APPLIED DISCRETE STRUCTURES



Al Doerr & Ken Levasseur
University of Massachusetts Lowell



Applied Discrete Structures

Al Doerr

Department of Mathematical Sciences
University of Massachusetts Lowell

Ken Levasseur

Department of Mathematical Sciences
University of Massachusetts Lowell
kenneth_levasseur@uml.edu

May 25, 2021

This text is disseminated via the Open Education Resource (OER) LibreTexts Project (https://LibreTexts.org) and like the thousands of other texts available within this powerful platform, it is freely available for reading, printing, and "consuming."

The LibreTexts mission is to bring together students, faculty, and scholars in a collaborative effort to provide and accessible, and comprehensive platform that empowers our community to develop, curate, adapt, and adopt openly licensed resources and technologies; through these efforts we can reduce the financial burden born from traditional educational resource costs, ensuring education is more accessible for students and communities worldwide.

Most, but not all, pages in the library have licenses that may allow individuals to make changes, save, and print this book. Carefully consult the applicable license(s) before pursuing such effects. Instructors can adopt existing LibreTexts texts or Remix them to quickly build course-specific resources to meet the needs of their students. Unlike traditional textbooks, LibreTexts' web based origins allow powerful integration of advanced features and new technologies to support learning.



LibreTexts is the adaptable, user-friendly non-profit open education resource platform that educators trust for creating, customizing, and sharing accessible, interactive textbooks, adaptive homework, and ancillary materials. We collaborate with individuals and organizations to champion open education initiatives, support institutional publishing programs, drive curriculum development projects, and more.

The LibreTexts libraries are Powered by NICE CXone Expert and was supported by the Department of Education Open Textbook Pilot Project, the California Education Learning Lab, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

Have questions or comments? For information about adoptions or adaptions contact info@LibreTexts.org or visit our main website at https://LibreTexts.org.

This text was compiled on 06/08/2025

TABLE OF CONTENTS

Colophon

Licensing

Dedication

Acknowledgements

Preface

1: Set Theory

- 1.1: Set Notation and Relations
- 1.2: Basic Set Operations
- 1.3: Cartesian Products and Power Sets
- 1.4: Binary Representation of Positive Integers
- 1.5: Summation Notation and Generalizations

2: Combinatorics

- 2.1: Basic Counting Techniques - The Rule of Products
- 2.2: Permutations
- 2.3: Partitions of Sets and the Law of Addition
- 2.4: Combinations and the Binomial Theorem

3: Logic

- 3.1: Propositions and Logical Operators
- 3.2: Truth Tables and Propositions Generated by a Set
- 3.3: Equivalence and Implication
- 3.4: The Laws of Logic
- 3.5: Mathematical Systems and Proofs
- 3.6: Propositions over a Universe
- 3.7: Mathematical Induction
- 3.8: Quantifiers
- 3.9: A Review of Methods of Proof

4: More on Sets

- 4.1: Methods of Proof for Sets
- 4.2: Laws of Set Theory
- 4.3: Minsets
- 4.4: The Duality Principle

5: Introduction to Matrix Algebra

- 5.1: Basic Definitions and Operations
- 5.2: Special Types of Matrices
- 5.3: Laws of Matrix Algebra
- 5.4: Matrix Oddities

6: Relations

- 6.1: Basic Definitions
- 6.2: Graphs of Relations on a Set
- 6.3: Properties of Relations
- 6.4: Matrices of Relations
- 6.5: Closure Operations on Relations

7: Functions

- 7.1: Definition and Notation
- 7.2: Properties of Functions
- 7.3: Function Composition

8: Recursion and Recurrence Relations

- 8.1: The Many Faces of Recursion
- 8.2: Sequences
- 8.3: Recurrence Relations
- 8.4: Some Common Recurrence Relations
- 8.5: Generating Functions

9: Graph Theory

- 9.1: Graphs - General Introduction
- 9.2: Data Structures for Graphs
- 9.3: Connectivity
- 9.4: Traversals- Eulerian and Hamiltonian Graphs
- 9.5: Graph Optimization
- 9.6: Planarity and Colorings

10: Trees

- 10.1: What is a Tree?
- 10.2: Spanning Trees
- 10.3: Rooted Trees
- 10.4: Binary Trees

11: Algebraic Structures

- 11.1: Operations
- 11.2: Algebraic Systems
- 11.3: Some General Properties of Groups
- 11.4: Greatest Common Divisors and the Integers Modulo n
- 11.5: Subsystems
- 11.6: Direct Products
- 11.7: Isomorphisms

12: More Matrix Algebra

- 12.1: Systems of Linear Equations
- 12.2: Matrix Inversion
- 12.3: An Introduction to Vector Spaces
- 12.4: The Diagonalization Process
- 12.5: Some Applications

- 12.6: Linear Equations over the Integers Mod 2

13: Boolean Algebra

- 13.1: Posets Revisited
- 13.2: Lattices
- 13.3: Boolean Algebras
- 13.4: Atoms of a Boolean Algebra
- 13.5: Finite Boolean Algebras as n-tuples of 0's and 1's
- 13.6: Boolean Expressions
- 13.7: A Brief Introduction to Switching Theory and Logic Design

14: Monoids and Automata

- 14.1: Monoids
- 14.2: Free Monoids and Languages
- 14.3: Automata, Finite-State Machines
- 14.4: The Monoid of a Finite-State Machine
- 14.5: The Machine of a Monoid

15: Group Theory and Applications

- 15.1: Cyclic Groups
- 15.2: Cosets and Factor Groups
- 15.3: Permutation Groups
- 15.4: Normal Subgroups and Group Homomorphisms
- 15.5: Coding Theory, Group Codes

16: An Introduction to Rings and Fields

- 16.1: Rings, Basic Definitions and Concepts
- 16.2: Fields
- 16.3: Polynomial Rings
- 16.4: Field Extensions
- 16.5: Power Series

17: Appendix

- 17.1: A - Algorithms
  - 17.1.1: An Introduction to Algorithms
  - 17.1.2: The Invariant Relation Theorem
- 17.2: B - Python and SageMath
  - 17.2.1: Python Iterators
  - 17.2.2: Dictionaries
- 17.3: C - Determinants
  - 17.3.1: Definition
  - 17.3.2: Computation
- 17.4: D - Notation
- 17.5: E - References

Index

Glossary

Detailed Licensing

Colophon

Edition 3rd Edition - version 8

Website faculty.uml.edu/klevasseur/ADS2

©2021 Al Doerr, Ken Levasseur

Applied Discrete Structures by Alan Doerr and Kenneth Levasseur is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 United States License. You are free to Share: copy and redistribute the material in any medium or format; Adapt: remix, transform, and build upon the material. You may not use the material for commercial purposes. The licensor cannot revoke these freedoms as long as you follow the license terms.

Licensing

*A detailed breakdown of this resource's licensing can be found in **Back Matter/Detailed Licensing**.*

Dedication

To our families

Donna, Christopher, Melissa, and Patrick Doerr

Karen, Joseph, Kathryn, and Matthew Levasseur

Acknowledgements

We would like to acknowledge the following instructors for their helpful comments and suggestions.

- Tibor Beke, UMass Lowell
- Alex DeCourcy, UMass Lowell
- Vince DiChiacchio
- Warren Grieff, UMass Lowell
- Matthew Haner, Mansfield University (PA)
- Dan Klain, UMass Lowell
- Sitansu Mittra, UMass Lowell
- Ravi Montenegro, UMass Lowell
- Tony Penta, UMass Lowell
- Jim Propp, UMass Lowell
- Thao Tran, UMass Lowell
- Richard Voss, Florida Atlantic U.

I'd like to particularly single out Jim Propp for his close scrutiny, along with that of his students, many of whom are listed below.

I would like to thank Rob Beezer, David Farmer, Karl-Dieter Crisman and other participants on the pretext-xml-support group for their guidance and work on MathBook XML, which has now been renamed PreTeXt. Thanks to the Pedagogy Subcommittee of the UMass Lowell Transformational Education Committee for their financial assistance in helping getting this project started.

Many students have provided feedback and pointed out typos in several editions of this book. They are listed below. Students with no affiliation listed are from UMass Lowell.

- Ryan Allen
- Rebecca Alves
- Anonymous student from Florida Atlantic U.
- Junaid Baig
- Anju Balaji
- Carlos Barrientos
- Chris Berns
- Raymond Berger, Eckerd College
- Brianne Bindas
- Nicholas Bishop
- Nathan Blood
- Cameron Bolduc
- Sam Bouchard
- Eric Breslau
- Rachel Bryan
- Courtney Caldwell
- Joseph Calles
- Rebecca Campbelli
- AJ Capone
- Eric Carey
- Emily Cashman
- Cora Casteel
- Rachel Chaiser, U. of Puget Sound

- Sam Chambers
- Vanessa Chen
- Hannah Chiodo
- Sofya Chow
- David Connolly
- Sean Cummings
- Alex DeCourcy
- Ryan Delosh
- Hillari Denny
- Matthew Edwards
- John El-Helou
- Adam Espinola
- Josh Everett
- Christian Franco
- Anthony Gaeta
- Lisa Gieng
- Holly Goodreau
- Lilia Heimold
- Kevin Holmes
- Benjamin Houle
- Alexa Hyde
- Michael Ingemi
- Eunji Jang
- Kyle Joaquim
- Devin Johnson
- Jeremy Joubert
- William Jozefczyk
- Antony Kellermann
- Yorgo A. Kennos
- Thomas Kiley
- Cody Kingman
- Leant Seu Kim
- Jessica Kramer
- John Kuczynski
- Auris Kveraga
- Justin LaGree
- Daven Lagu
- Kendra Lansing
- Gregory Lawrence
- Pearl Laxague
- Kevin Le
- Matt LeBlanc
- Maxwell Leduc
- Ariel Leva
- Robert Liana

- Tammy Liu
- Laura Lucaciu
- Kelly Ly
- Kevin Mackie, Learning Assistant
- Alexandra Mai
- Andrew Magee
- Matthew Malone
- Logan Mann
- Sam Marquis
- Amy Mazzucotelli
- Adam Melle
- Jason McAdam
- Nick McArdle
- Christine McCarthy
- Shelbylynn McCoy
- Conor McNierney
- Albara Mehene
- Joshua Michaud
- Max Mints
- Charles Mirabile
- Timothy Miskell
- Genevieve Moore
- Mike Morley
- Zach Mulcahy
- Tessa Munoz
- Zachary Murphy
- Logan Nadeau
- Carol Nguyen
- Hung Nguyen
- Shelly Noll
- Steven Oslan, the champion typo finder!
- Harsh Patel
- Beck Peterson
- Donna Petitti
- Paola Pevzner
- Samantha Poirier
- Roshan Ravi
- Ian Roberts
- John Raisbeck
- Adelia Reid
- Derek Ross
- Tyler Ross
- Jacob Rothmel
- Zach Rush
- Steve Sadler, Bellevue College (WA)

- Doug Salvati
- Chita Sano
- Noah Schultz
- Anna Sergienko
- Ben Shipman
- Florens Shosho
- Lorraine Sill
- Jonathan Silva
- Joshua Simard
- Mason Sirois
- Sana Shaikh
- Joel Slebodnick
- Greg Smelkov
- Andrew Somerville
- Alicia Stransky
- Brandon Swanberg
- Joshua Sullivan
- James Tan
- Bunchhoung Tiv
- Joanel Vasquez
- Rolando Vera
- Anh Vo
- Nick Wackowski
- Ryan Wallace
- Uriah Wardlaw
- Zachary Watkins
- Steve Werren
- Laura Wikoff
- Henry Zhu
- Several students at Luzurne County Community College (PA)

Preface

Applied Discrete Structures is designed for use in a university course in discrete mathematics spanning up to two semesters. Its original design was for computer science majors to be introduced to the mathematical topics that are useful in computer science. It can also serve the same purpose for mathematics majors, providing a first exposure to many essential topics.

We embarked on this open-source project in 2010, twenty-one years after the publication of the 2nd edition of *Applied Discrete Structures for Computer Science* in 1989. We had signed a contract for the second edition with Science Research Associates in 1988 but by the time the book was ready to print, SRA had been sold to MacMillan. Soon after, the rights had been passed on to Pearson Education, Inc. In 2010, the long-term future of printed textbooks was uncertain. In the meantime, textbook prices (both printed and e-books) had increased and a growing open source textbook movement had started. One of our objectives in revisiting this text is to make it available to our students in an affordable format. In its original form, the text was peer-reviewed and was adopted for use at several universities throughout the country. For this reason, we see *Applied Discrete Structures* as not only an inexpensive alternative, but a high quality alternative.

The current version of *Applied Discrete Structures* has been developed using *PreTeXt*, a lightweight XML application for authors of scientific articles, textbooks and monographs initiated by Rob Beezer, U. of Puget Sound. When the PreTeXt project was launched, it was the natural next step. The features of PreTeXt make it far more readable, with easy production of web, pdf and print formats.

The current computing landscape is very different from the 1980's and this accounts for the most significant changes in the text. One of the most common programming languages of the 1980's was Pascal. We used it to illustrate many of the concepts in the text. Although it isn't totally dead, Pascal is far from the mainstream of computing in the 21st century. The open source software movement was just starting in the late 1980's and in 2005, the first version of Sage (later renamed SageMath), an open-source computer algebra system, was first released. In Applied Discrete Structures we have replaced "Pascal Notes" with "SageMath Notes."

Many of the concepts introduced in this text are illustrated using SageMath code. SageMath (sagemath.org) is a free, open source, software system for advanced mathematics. Sage can be used either on your own computer, a local server, or on SageMathCloud (https://cloud.sagemath.com).*Ken Levasseur*
Lowell, MA

CHAPTER OVERVIEW

1: Set Theory

> ***empty set***
>
> *Betty's math teacher said, in a sweat:*
> *"I will teach you some set theory yet!"*
> *But his best efforts failed,*
> *And at Betty he railed:*
> *"Your insights? A true **empty set**!"*
>
> *SheilaB, The Omnificent English Dictionary In Limerick Form*

We begin this chapter with a brief description of discrete mathematics. We then cover some of the basic set language and notation that will be used throughout the text. Venn diagrams will be introduced in order to give the reader a clear picture of set operations. In addition, we will describe the binary representation of positive integers and introduce summation notation and its generalizations.

1.1: Set Notation and Relations
1.2: Basic Set Operations
1.3: Cartesian Products and Power Sets
1.4: Binary Representation of Positive Integers
1.5: Summation Notation and Generalizations

This page titled 1: Set Theory is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

1.1: Set Notation and Relations

1.1.1: Notion of a Set

The term set is intuitively understood by most people to mean a collection of objects that are called elements (of the set). This concept is the starting point on which we will build more complex ideas, much as in geometry where the concepts of point and line are left undefined. Because a set is such a simple notion, you may be surprised to learn that it is one of the most difficult concepts for mathematicians to define to their own liking. For example, the description above is not a proper definition because it requires the definition of a collection. (How would you define "collection"?) Even deeper problems arise when you consider the possibility that a set could contain itself. Although these problems are of real concern to some mathematicians, they will not be of any concern to us. Our first concern will be how to describe a set; that is, how do we most conveniently describe a set and the elements that are in it? If we are going to discuss a set for any length of time, we usually give it a name in the form of a capital letter (or occasionally some other symbol). In discussing set A, if x is an element of A, then we will write $x \in A$. On the other hand, if x is not an element of A, we write $x \notin A$. The most convenient way of describing the elements of a set will vary depending on the specific set.

Enumeration. When the elements of a set are enumerated (or listed) it is traditional to enclose them in braces. For example, the set of binary digits is $\{0, 1\}$ and the set of decimal digits is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.The choice of a name for these sets would be arbitrary; but it would be "logical" to call them B and D, respectively. The choice of a set name is much like the choice of an identifier name in programming. Some large sets can be enumerated without actually listing all the elements. For example, the letters of the alphabet and the integers from 1 to 100 could be described as $A = \{a, b, c, \ldots, x, y, z\}$, and $G = \{1, 2, \ldots, 99, 100\}$.The three consecutive "dots" are called an ellipsis. We use them when it is clear what elements are included but not listed. An ellipsis is used in two other situations. To enumerate the positive integers, we would write $\{1, 2, 3, \ldots\}$, indicating that the list goes on infinitely. If we want to list a more general set such as the integers between 1 and n, where n is some undetermined positive integer, we might write $\{1, \ldots, n\}$.

Standard Symbols. Sets that are frequently encountered are usually given symbols that are reserved for them alone. For example, since we will be referring to the positive integers throughout this book, we will use the symbol $\mathbb{P}$ instead of writing $\{1, 2, 3, \ldots\}$.A few of the other sets of numbers that we will use frequently are:

- $\mathbb{P}$: the positive integers, $\{1, 2, 3, 4, \ldots\}$
- $\mathbb{N}$: the natural numbers, $\{0, 1, 2, 3, \ldots\}$
- $\mathbb{Z}$: the integers, $\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$
- $\mathbb{Q}$: the rational numbers
- $\mathbb{R}$: the real numbers
- $\mathbb{C}$: the complex numbers

Caution: Some people (roughly half of the world?) call the set $\{1, 2, 3, 4, \ldots\}$the natural numbers. We are not among them. We take the *pythonic* approach that assumes that starting with zero is more natural than starting at one.

Set-Builder Notation. Another way of describing sets is to use set-builder notation. For example, we could define the rational numbers as

$$\mathbb{Q} = \{a/b \mid a, b \in \mathbb{Z}, b \neq 0\}.$$

Note that in the set-builder description for the rational numbers:

- a/b indicates that a typical element of the set is a "fraction."
- The vertical line, $\mid$, is read "such that" or "where," and is used interchangeably with a colon.
- $a, b \in \mathbb{Z}$ is an abbreviated way of saying a and b are integers.
- Commas in mathematics are read as "and."

The important fact to keep in mind in set notation, or in any mathematical notation, is that it is meant to be a help, not a hindrance. We hope that notation will assist us in a more complete understanding of the collection of objects under consideration and will enable us to describe it in a concise manner. However, brevity of notation is not the aim of sets. If you prefer to write $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ instead of $a, b \in \mathbb{Z}$, you should do so. Also, there are frequently many different, and equally good, ways of describing sets. For example, $\{x \in \mathbb{R} \mid x^2 - 5x + 6 = 0\}$ and $\{x \mid x \in \mathbb{R}, x^2 - 5x + 6 = 0\}$ both describe the solution set $\{2, 3\}$.

A proper definition of the real numbers is beyond the scope of this text. It is sufficient to think of the real numbers as the set of points on a number line. The complex numbers can be defined using set-builder notation as $\mathbb{C} = \{a + bi : a, b \in \mathbb{R}\}$, where $i^2 = -1$.

In the following definition we will leave the word "finite" undefined.

Definition 1.1.1: Finite Set

A set is a finite set if it has a finite number of elements. Any set that is not finite is an infinite set.

Definition 1.1.2: Cardinality

Let A be a finite set. The number of different elements in A is called its cardinality. The cardinality of a finite set A is denoted $|A|$.

As we will see later, there are different infinite cardinalities. We can't make this distinction until Chapter 7, so we will restrict cardinality to finite sets for now.

1.1.2: Subsets

Definition 1.1.3: Subset

Let A and B be sets. We say that A is a subset of B if and only if every element of A is an element of B. We write $A \subseteq B$ to denote the fact that A is a subset of B.

Example 1.1.1: Some Subsets

a. If $A = \{3, 5, 8\}$ and $B = \{5, 8, 3, 2, 6\}$, then $A \subseteq B$.
b. $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$
c. If $S = \{3, 5, 8\}$ and $T = \{5, 3, 8\}$, then $S \subseteq T$ and $T \subseteq S$.

Definition 1.1.4: Set Equality

Let A and B be sets. We say that A is equal to B (notation $A = B$) if and only if every element of A is an element of B and conversely every element of B is an element of A; that is, $A \subseteq B$ and $B \subseteq A$.

Example 1.1.2: Examples Illustrating Set Equality

a. In Example 1.1.1, $S = T$. Note that the ordering of the elements is unimportant.
b. The number of times that an element appears in an enumeration doesn't affect a set. For example, if $A = \{1, 5, 3, 5\}$ and $B = \{1, 5, 3\}$, then $A = B$. Warning to readers of other texts: Some books introduce the concept of a multiset, in which the number of occurrences of an element matters.

A few comments are in order about the expression "if and only if" as used in our definitions. This expression means "is equivalent to saying," or more exactly, that the word (or concept) being defined can at any time be replaced by the defining expression. Conversely, the expression that defines the word (or concept) can be replaced by the word.

Occasionally there is need to discuss the set that contains no elements, namely the empty set, which is denoted by $\emptyset$. This set is also called the null set.

It is clear, we hope, from the definition of a subset, that given any set A we have $A \subseteq A$ and $\emptyset \subseteq A$. If A is nonempty, then A is called an ***improper subset*** of A. All other subsets of A, including the empty set, are called ***proper subsets*** of A. The empty set is an improper subset of itself.

Note 1.1.1

Not everyone is in agreement on whether the empty set is a proper subset of any set. In fact earlier editions of this book sided with those who considered the empty set an improper subset. However, we bow to the emerging consensus at this time.

1.1.3: Exercises

Exercise 1.1.1

List four elements of each of the following sets:

a. $\{k \in \mathbb{P} \mid k-1 \text{ is a multiple of } 7\}$
b. $\{x \mid x \text{ is a fruit and its skin is normally eaten}\}$
c. $\{x \in \mathbb{Q} \mid \frac{1}{x} \in \mathbb{Z}\}$
d. $\{2n \mid n \in \mathbb{Z}, n < 0\}$
e. $\{s \mid s = 1+2+\cdots+n \text{ for some } n \in \mathbb{P}\}$

Answer

These answers are not unique.

a. $8, 15, 22, 29$
b. apple, pear, peach, plum
c. $1/2, 1/3, 1/4, 1/5$
d. $-8, -6, -4, -2$
e. $6, 10, 15, 21$

Exercise 1.1.2

List all elements of the following sets:

a. $\{\frac{1}{n} \mid n \in \{3,4,5,6\}\}$
b. $\{\alpha \in \text{ the alphabet } \mid \alpha \text{ precedes F}\}$
c. $\{x \in \mathbb{Z} \mid x = x+1\}$
d. $\{n^2 \mid n = -2, -1, 0, 1, 2\}$
e. $\{n \in \mathbb{P} \mid n \text{ is a factor of } 24\}$

Exercise 1.1.3

Describe the following sets using set-builder notation.

a. $\{5, 7, 9, \ldots, 77, 79\}$
b. the rational numbers that are strictly between -1 and 1
c. the even integers
d. $\{-18, -9, 0, 9, 18, 27, \ldots\}$

Answer

a. $\{2k+1 \mid k \in \mathbb{Z}, 2 \leqslant k \leqslant 39\}$
b. $\{x \in \mathbb{Q} \mid -1 < x < 1\}$
c. $\{2n \mid n \in \mathbb{Z}\}$
d. $\{9n \mid n \in \mathbb{Z}, -2 \leq n\}$

Exercise 1.1.4

Use set-builder notation to describe the following sets:

a. $\{1, 2, 3, 4, 5, 6, 7\}$
b. $\{1, 10, 100, 1000, 10000\}$
c. $\{1, 1/2, 1/3, 1/4, 1/5, \dots\}$
d. $\{0\}$

Exercise 1.1.5

Let $A = \{0, 2, 3\}$, $B = \{2, 3\}$, and $C = \{1, 5, 9\}$. Determine which of the following statements are true. Give reasons for your answers.

a. $3 \in A$
b. $\{3\} \in A$
c. $\{3\} \subseteq A$
d. $B \subseteq A$
e. $A \subseteq B$
f. $\emptyset \subseteq C$
g. $\emptyset \in A$
h. $A \subseteq A$

Answer

a. True
b. False
c. True
d. True
e. False
f. True
g. False
h. True

Exercise 1.1.6

One reason that we left the definition of a set vague is Russell's Paradox. Many mathematics and logic books contain an account of this paradox. Two references are [41] and [36]. Find one such reference and read it.

This page titled 1.1: Set Notation and Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

1.2: Basic Set Operations

1.2.1: Definitions

Definition 1.2.1: Intersection

Let A and B be sets. The intersection of A and B (denoted by $A \cap B$) is the set of all elements that are in both A and B. That is, $A \cap B = \{x : x \in A \text{ and } x \in B\}$.

Example 1.2.1: Some Intersections

- Let $A = \{1, 3, 8\}$ and $B = \{-9, 22, 3\}$. Then $A \cap B = \{3\}$.
- Solving a system of simultaneous equations such as $x + y = 7$ and $x - y = 3$ can be viewed as an intersection. Let $A = \{(x, y) : x + y = 7, x, y \in \mathbb{R}\}$ and $B = \{(x, y) : x - y = 3, x, y \in \mathbb{R}\}$. These two sets are lines in the plane and their intersection, $A \cap B = \{(5, 2)\}$, is the solution to the system.
- $\mathbb{Z} \cap \mathbb{Q} = \mathbb{Z}$.
- If $A = \{3, 5, 9\}$ and $B = \{-5, 8\}$, then $A \cap B = \emptyset$.

Definition 1.2.2: Disjoint Sets

Two sets are disjoint if they have no elements in common. That is, A and B are disjoint if $A \cap B = \emptyset$.

Definition 1.2.3: Union

Let A and B be sets. The union of A and B (denoted by $A \cup B$) is the set of all elements that are in A or in B or in both A and B. That is, $A \cup B = \{x : x \in A \text{ or } x \in B\}$.

It is important to note in the set-builder notation for $A \cup B$, the word "or" is used in the inclusive sense; it includes the case where x is in both A and B.

Example 1.2.2: Some Unions

- If $A = \{2, 5, 8\}$ and $B = \{7, 5, 22\}$, then $A \cup B = \{2, 5, 8, 7, 22\}$.
- $\mathbb{Z} \cup \mathbb{Q} = \mathbb{Q}$.
- $A \cup \emptyset = A$ for any set A.

Frequently, when doing mathematics, we need to establish a universe or set of elements under discussion. For example, the set $A = \{x : 81x^4 - 16 = 0\}$ contains different elements depending on what kinds of numbers we allow ourselves to use in solving the equation $81x^4 - 16 = 0$. This set of numbers would be our universe. For example, if the universe is the integers, then A is empty. If our universe is the rational numbers, then A is $\{2/3, -2/3\}$ and if the universe is the complex numbers, then A is $\{2/3, -2/3, 2i/3, -2i/3\}$.

Definition 1.2.4: Universe

The universe, or universal set, is the set of all elements under discussion for possible membership in a set. We normally reserve the letter U for a universe in general discussions.

1.2.2: Operations and their Venn Diagrams

When working with sets, as in other branches of mathematics, it is often quite useful to be able to draw a picture or diagram of the situation under consideration. A diagram of a set is called a Venn diagram. The universal set U is represented by the interior of a rectangle and the sets by disks inside the rectangle.

Example 1.2.3: Venn Diagram Examples

$A \cap B$ is illustrated in Figure 1.2.1 by shading the appropriate region.

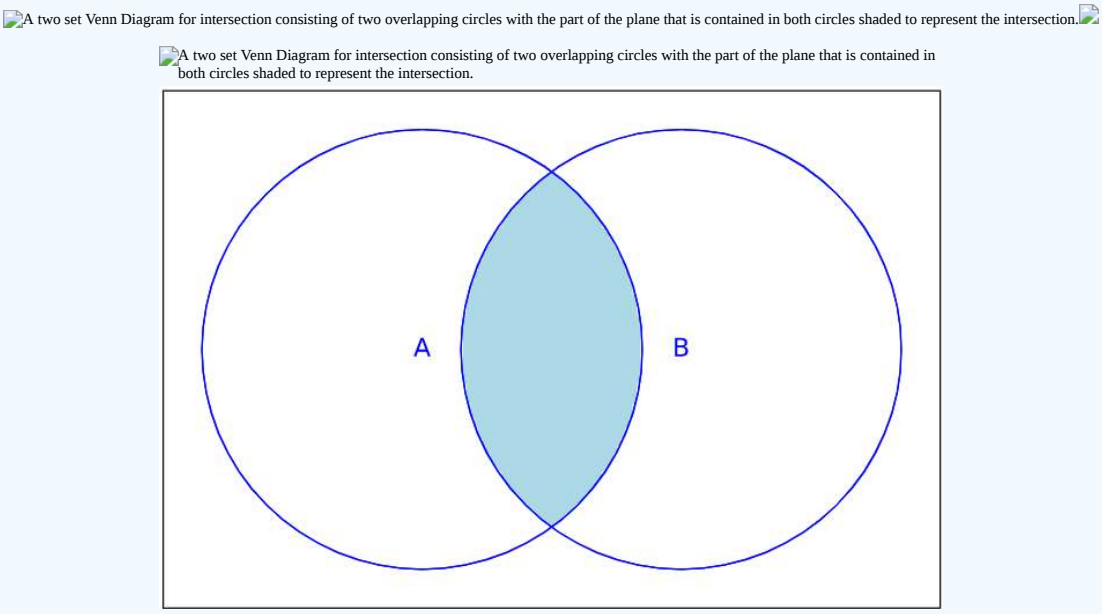


Figure 1.2.1: Venn Diagram for the Intersection of Two Sets

The union $A \cup B$ is illustrated in Figure 1.2.2.

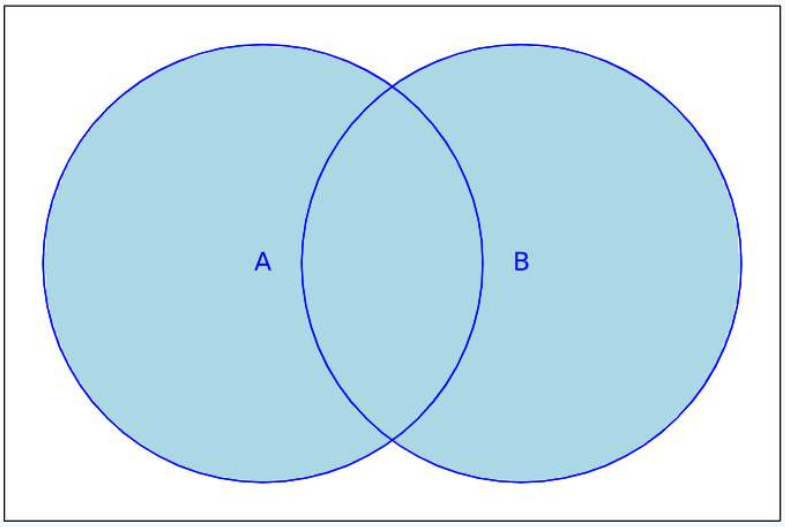


Figure 1.2.2: Venn Diagram for the Union $A \cup B$

In a Venn diagram, the region representing $A \cap B$ does not appear empty; however, in some instances it will represent the empty set. The same is true for any other region in a Venn diagram.

Definition 1.2.5: Complement of a Set

Let A and B be sets. The complement of A relative to B (notation $B - A$) is the set of elements that are in B and not in A. That is, $B - A = \{x : x \in B \text{ and } x \notin A\}$. If U is the universal set, then $U - A$ is denoted by A^c and is called simply the complement of A. $A^c = \{x \in U : x \notin A\}$.

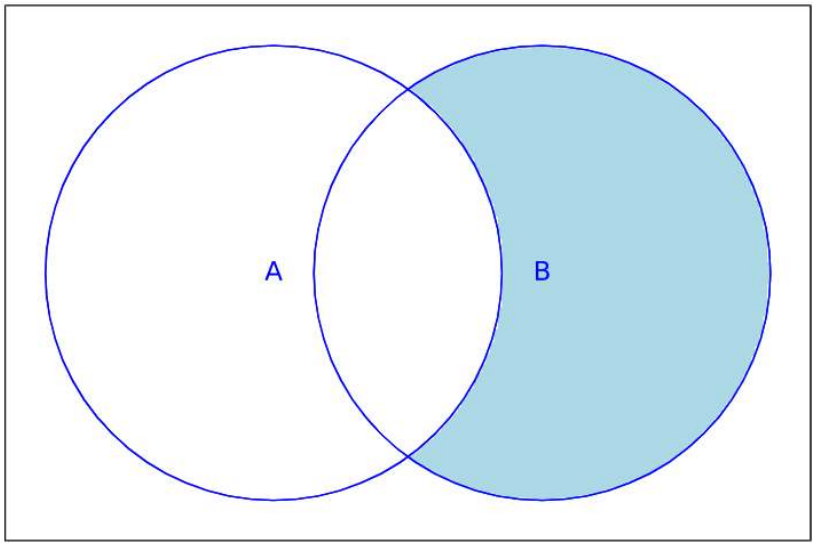


Figure 1.2.3: Venn Diagram for $B - A$

Example 1.2.4: Some Complements

a. Let $U = \{1, 2, 3, ..., 10\}$ and $A = \{2, 4, 6, 8, 10\}$. Then $U - A = \{1, 3, 5, 7, 9\}$ and $A - U = \emptyset$.
b. If $U = \mathbb{R}$, then the complement of the set of rational numbers is the set of irrational numbers.
c. $U^c = \emptyset$ and $\emptyset^c = U$.
d. The Venn diagram of $B - A$ is represented in Figure 1.2.3.
e. The Venn diagram of A^c is represented in Figure 1.2.4.
f. If $B \subseteq A$, then the Venn diagram of $A - B$ is as shown in Figure 1.2.5.
g. In the universe of integers, the set of even integers, $\{\ldots, -4, -2, 0, 2, 4, \ldots\}$, has the set of odd integers as its complement.

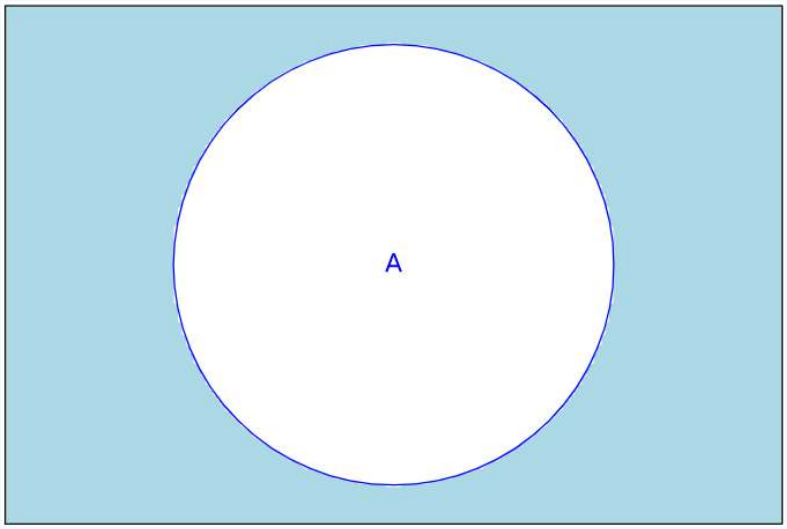


Figure 1.2.4: Venn Diagram for A^c

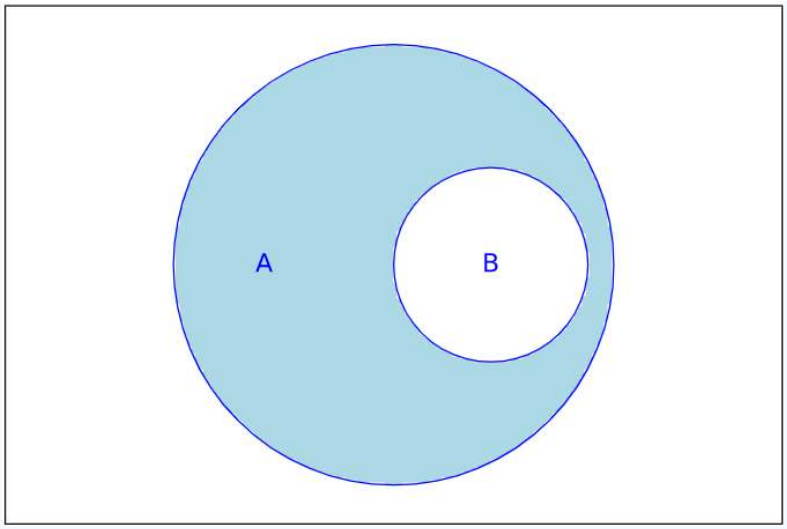


Figure 1.2.5: Venn Diagram for $A - B$ when B is a subset of A

Definition 1.2.6: Symmetric Difference

Let A and B be sets. The symmetric difference of A and B (denoted by $A \oplus B$) is the set of all elements that are in A and B but not in both. That is, $A \oplus B = (A \cup B) - (A \cap B)$.

Example 1.2.5: Some Symmetric Differences

a. Let $A = \{1, 3, 8\}$ and $B = \{2, 4, 8\}$. Then $A \oplus B = \{1, 2, 3, 4\}$.
b. $A \oplus \emptyset = A$ and $A \oplus A = \emptyset$ for any set A.
c. $\mathbb{R} \oplus \mathbb{Q}$ is the set of irrational numbers.
d. The Venn diagram of $A \oplus B$ is represented in Figure 1.2.6.

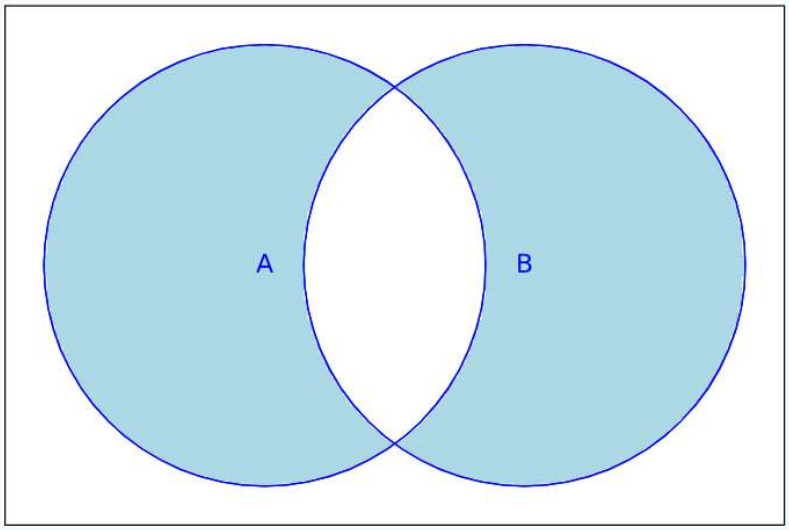


Figure 1.2.6: Venn Diagram for the symmetric difference \(A \oplus B\)

1.2.3: SageMath Note: Sets

To work with sets in Sage, a set is an expression of the form Set(*list*). By wrapping a list with `Set( )` , the order of elements appearing in the list and their duplication are ignored. For example, L1 and L2 are two different lists, but notice how as sets they are considered equal:

```
L1=[3,6,9,0,3]
L2=[9,6,3,0,9]
[L1==L2, Set(L1)==Set(L2) ]
```

The standard set operations are all methods and/or functions that can act on Sage sets. *You need to evaluate the following cell to use the subsequent cell.*

```
A=Set(srange(5,50,5))
B=Set(srange(6,50,6))
[A,B]
```

We can test membership, asking whether 10 is in each of the sets:

```
[10 in A, 10 in B]
```

The ampersand is used for the intersection of sets. Change it to the vertical bar, |, for union.

```
A & B
```

Symmetric difference and set complement are defined as “methods” in Sage. Here is how to compute the symmetric difference of A with B, followed by their differences.

```
[A.symmetric_difference(B),A.difference(B),B.difference(A)]
```

1.2.4: Exercises

Exercise 1.2.1

Let $A=\{0,2,3\}, B=\{2,3\}, C=\{1,5,9\}$, and let the universal set be $U=\{0,1,2,\ldots,9\}$. Determine:

a. $A\cap B$
b. $A\cup B$
c. $B\cup A$
d. $A\cup C$
e. $A-B$
f. $B-A$
g. A^c
h. C^c
i. $A\cap C$
j. $A\oplus B$

Answer

a. $\{2,3\}$
b. $\{0,2,3\}$
c. $\{0,2,3\}$
d. $\{0,1,2,3,5,9\}$
e. $\{0\}$

f. $\emptyset$
g. $\{1,4,5,6,7,8,9\}$
h. $\{0,2,3,4,6,7,8\}$
i. $\emptyset$
j. $\{0\}$

Exercise 1.2.2

Let A, B, and C be as in Exercise 1, let $D=\{3,2\}$, and let $E=\{2,3,2\}$. Determine which of the following are true. Give reasons for your decisions.

a. $A=B$
b. $B=C$
c. $B=D$
d. $E=D$
e. $A\cap B=B\cap A$
f. $A\cup B=B\cup A$
g. $A-B=B-A$
h. $A\oplus B=B\oplus A$

Exercise 1.2.3

Let $U=\{1,2,3,\ldots,9\}$. Give examples of sets A, B, and C for which:

a. $A\cap(B\cap C)=(A\cap B)\cap C$
b. $A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$
c. $(A\cup B)^c=A^c\cap B^c$
d. $A\cup A^c=U$
e. $A\subseteq A\cup B$
f. $A\cap B\subseteq A$

Answer

These are all true for any sets A, B, and C.

Exercise 1.2.4

Let $U=\{1,2,3,\ldots,9\}$. Give examples to illustrate the following facts:

a. If $A\subseteq B$ and $B\subseteq C$, then $A\subseteq C$.
b. There are sets A and B such that $A-B\neq B-A$
c. If $U=A\cup B$ and $A\cap B=\emptyset$, it always follows that $A=U-B$.

Exercise 1.2.5

What can you say about A if $U=\{1,2,3,4,5\}, B=\{2,3\}$, and (separately)

a. $A\cup B=\{1,2,3,4\}$
b. $A\cap B=\{2\}$
c. $A\oplus B=\{3,4,5\}$

Answer

a. $\{1,4\}\subseteq A\subseteq\{1,2,3,4\}$
b. $\{2\}\subseteq A\subseteq\{1,2,4,5\}$
c. $A=\{2,4,5\}$

Exercise 1.2.6

Suppose that U is an infinite universal set, and A and B are infinite subsets of U. Answer the following questions with a brief explanation.

a. Must A^c be finite?
b. Must $A \cup B$ be infinite?
c. Must $A \cap B$ be infinite?

Exercise 1.2.7

Given that U = all students at a university, D = day students, M = mathematics majors, and G = graduate students. Draw Venn diagrams illustrating this situation and shade in the following sets:

a. evening students
b. undergraduate mathematics majors
c. non-math graduate students
d. non-math undergraduate students

Answer

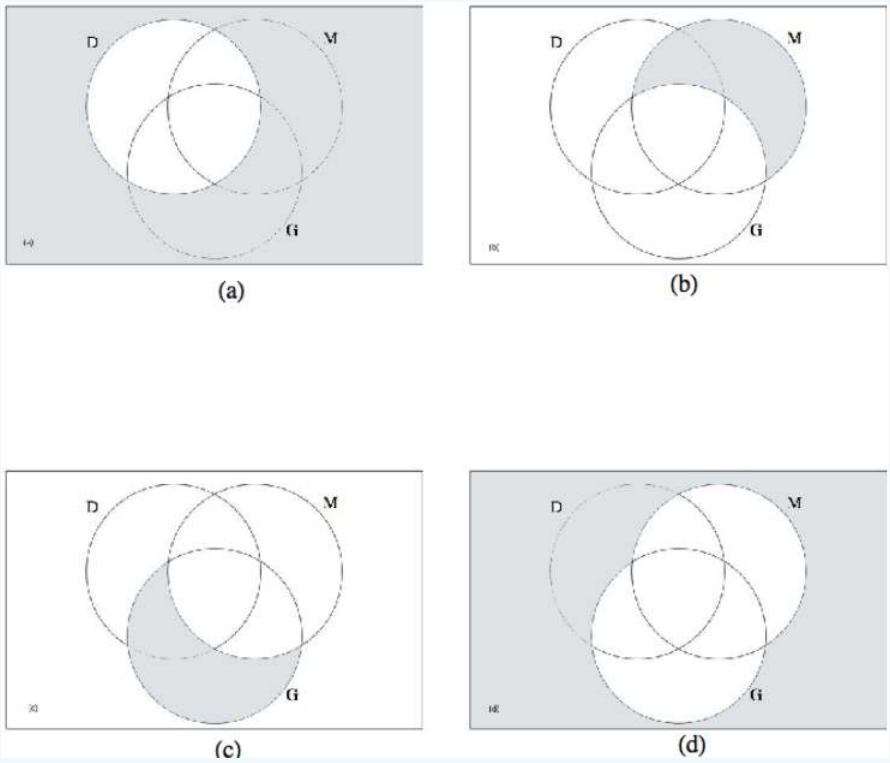


Figure 1.2.7 Solution to Exercise 1.2.7of Section 1.2.

Exercise 1.2.8

Let the sets D, M, G, and U be as in exercise 7. Let $|U| = 16,000$, $|D| = 9,000$, $|M| = 300$, and $|G| = 1,000$. Also assume that the number of day students who are mathematics majors is 250, 50 of whom are graduate students, that there are 95 graduate mathematics majors, and that the total number of day graduate students is 700. Determine the number of students who are:

a. evening students
b. nonmathematics majors
c. undergraduates (day or evening)

d. day graduate nonmathematics majors
e. evening graduate students
f. evening graduate mathematics majors
g. evening undergraduate nonmathematics majors

This page titled 1.2: Basic Set Operations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

1.3: Cartesian Products and Power Sets

1.3.1: Cartesian Products

Definition 1.3.1: Cartesian Product

Let A and B be sets. The Cartesian product of A and B, denoted by $A \times B$, is defined as follows: $A \times B = \{(a, b) \mid a \in A \quad \text{and} \quad b \in B\}$, that is, $A \times B$ is the set of all possible ordered pairs whose first component comes from A and whose second component comes from B.

Example 1.3.1: Cartesian Product

Notation in mathematics is often developed for good reason. In this case, a few examples will make clear why the symbol $\times$ is used for Cartesian products.

- Let $A = \{1, 2, 3\}$ and $B = \{4, 5\}$. Then $A \times B = \{(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}$.Note that $|A \times B| = 6 = |A| \times |B|$.
- $A \times A = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$Note that $|A \times A| = 9 = |A|^2$.

These two examples illustrate the general rule that if A and B are finite sets, then $|A \times B| = |A| \times |B|$.

We can define the Cartesian product of three (or more) sets similarly. For example, $A \times B \times C = \{(a, b, c) : a \in A, b \in B, c \in C\}$.

It is common to use exponents if the sets in a Cartesian product are the same:

$$A^2 = A \times A$$

$$A^3 = A \times A \times A$$

and in general,

$$A^n = \underbrace{A \times A \times \ldots \times A}_{n \text{ factors}}.$$

1.3.2: Power Sets

Definition 1.3.2: Power Set

If A is any set, the power set of A is the set of all subsets of A, denoted $\mathcal{P}(A)$.

The two extreme cases, the empty set and all of A, are both included in $\mathcal{P}(A)$.

Example 1.3.2: Some Power Sets

- $\mathcal{P}(\emptyset) = \{\emptyset\}$
- $\mathcal{P}(\{1\}) = \{\emptyset, \{1\}\}$
- $\mathcal{P}(\{1, 2\}) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

We will leave it to you to guess at a general formula for the number of elements in the power set of a finite set. In Chapter 2, we will discuss counting rules that will help us derive this formula.

1.3.3: SageMath Note: Cartesian Products and Power Sets

Here is a simple example of a cartesian product of two sets:

```
A=Set([0,1,2])
B=Set(['a','b'])
P=cartesian_product([A,B]);P
```

Here is the cardinality of the cartesian product.

```
P.cardinality()
```

The power set of a set is an iterable, as you can see from the output of this next cell

```
U=Set([0,1,2,3])
subsets(U)
```

You can iterate over a powerset. Here is a trivial example.

```
for a in subsets(U):
    print(str(a)+ " has " +str(len(a))+" elements.")
```

1.3.3.1: Exercises

Exercise 1.3.1

Let $A = \{0, 2, 3\}, B = \{2, 3\}, C = \{1, 4\}$, and let the universal set be $U = \{0, 1, 2, 3, 4\}$. List the elements of

a. $A \times B$
b. $B \times A$
c. $A \times B \times C$
d. $U \times \emptyset$
e. $A \times A^c$
f. B^2
g. B^3
h. $B \times \mathcal{P}(B)$

Answer

a. $\{(0, 2), (0, 3), (2, 2), (2, 3), (3, 2), (3, 3)\}$
b. $\{(2, 0), (2, 2), (2, 3), (3, 0), (3, 2), (3, 3)\}$
c. $\{(0, 2, 1), (0, 2, 4), (0, 3, 1), (0, 3, 4), (2, 2, 1), (2, 2, 4),$
$(2, 3, 1), (2, 3, 4), (3, 2, 1), (3, 2, 4), (3, 3, 1), (3, 3, 4)\}$
d. $\emptyset$
e. $\{(0, 1), (0, 4), (2, 1), (2, 4), (3, 1), (3, 4)\}$
f. $\{(2, 2), (2, 3), (3, 2), (3, 3)\}$
g. $\{(2, 2, 2), (2, 2, 3), (2, 3, 2), (2, 3, 3), (3, 2, 2), (3, 2, 3), (3, 3, 2), (3, 3, 3)\}$
h. $\{(2, \emptyset), (2, \{2\}), (2, \{3\}), (2, \{2, 3\}), (3, \emptyset), (3, \{2\}), (3, \{3\}), (3, \{2, 3\})\}$

Exercise 1.3.2

Suppose that you are about to flip a coin and then roll a die. Let $A = \{HEADS, TAILS\}$ and $B = \{1, 2, 3, 4, 5, 6\}$.

a. What is $|A \times B|$?
b. How could you interpret the set $A \times B$?

Exercise 1.3.3

List all two-element sets in $\mathcal{P}(\{a, b, c, d\})$

Answer

$\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}$ and $\{c, d\}$

Exercise 1.3.4

List all three-element sets in $\mathcal{P}(\{a, b, c, d\})$.

Exercise 1.3.5

How many singleton (one-element) sets are there in $\mathcal{P}(A)$ if $|A| = n$?

Answer

There are n singleton subsets, one for each element.

Exercise 1.3.6

A person has four coins in his pocket: a penny, a nickel, a dime, and a quarter. How many different sums of money can he take out if he removes 3 coins at a time?

Exercise 1.3.7

Let $A = \{+, -\}$ and $B = \{00, 01, 10, 11\}$.

a. List the elements of $A \times B$
b. How many elements do A^4 and $(A \times B)^3$ have?

Answer

a. $\{+00, +01, +10, +11, -00, -01, -10, -11\}$
b. 16 and 512

Exercise 1.3.8

Let $A = \{\bullet, \square, \otimes\}$ and $B = \{\square, \ominus, \bullet\}$.

a. List the elements of $A \times B$ and $B \times A$. The parentheses and comma in an ordered pair are not necessary in cases such as this where the elements of each set are individual symbols.
b. Identify the intersection of $A \times B$ and $B \times A$ for the case above, and then guess at a general rule for the intersection of $A \times B$ and $B \times A$, where A and B are any two sets.

Exercise 1.3.9

Let A and B be nonempty sets. When are $A \times B$ and $B \times A$ equal?

Answer

They are equal when $A = B$.

This page titled 1.3: Cartesian Products and Power Sets is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

1.4: Binary Representation of Positive Integers

1.4.1: Grouping by Twos

Recall that the set of positive integers, $\mathbb{P}$, is $\{1, 2, 3, \ldots\}$.Positive integers are naturally used to count things. There are many ways to count and many ways to record, or represent, the results of counting. For example, if we wanted to count five hundred twenty-three apples, we might group the apples by tens. There would be fifty-two groups of ten with three single apples left over. The fifty-two groups of ten could be put into five groups of ten tens (hundreds), with two tens left over. The five hundreds, two tens, and three units is recorded as 523. This system of counting is called the base ten positional system, or decimal system. It is quite natural for us to do grouping by tens, hundreds, thousands, ... since it is the method that all of us use in everyday life.

The term positional refers to the fact that each digit in the decimal representation of a number has a significance based on its position. Of course this means that rearranging digits will change the number being described. You may have learned of numeration systems in which the position of symbols does not have any significance (e.g., the ancient Egyptian system). Most of these systems are merely curiosities to us now.

The binary number system differs from the decimal number system in that units are grouped by twos, fours, eights, etc. That is, the group sizes are powers of two instead of powers of ten. For example, twenty-three can be grouped into eleven groups of two with one left over. The eleven twos can be grouped into five groups of four with one group of two left over. Continuing along the same lines, we find that twenty-three can be described as one sixteen, zero eights, one four, one two, and one one, which is abbreviated 10111_{two}, or simply 10111 if the context is clear.

1.4.2: Conversion Algorithm

The process that we used to determine the binary representation of 23 can be described in general terms to determine the binary representation of any positive integer n. A general description of a process such as this one is called an algorithm. Since this is the first algorithm in the book, we will first write it out using less formal language than usual, and then introduce some "algorithmic notation." If you are unfamiliar with algorithms, we refer you to Section 17.1

1. Start with an empty list of bits.
2. Assign the variable k the value n.
3. While k's value is positive, continue performing the following three steps until k becomes zero and then stop.
   a. divide k by 2, obtaining a quotient q (often denoted $k \operatorname{div} 2$) and a remainder r (denoted $(k \bmod 2)$).
   b. attach r to the left-hand side of the list of bits.
   c. assign the variable k the value q.

Example 1.4.1: An Example of Conversion to Binary

To determine the binary representation of 41 we take the following steps:

- $41 = 2 \times 20 + 1 \quad List = 1$
- $20 = 2 \times 10 + 0 \quad List = 01$
- $10 = 2 \times 5 + 0 \quad List = 001$
- $5 = 2 \times 2 + 1 \quad List = 1001$
- $2 = 2 \times 1 + 0 \quad List = 01001$
- $1 = 2 \times 0 + 1 \quad List = 101001$

Therefore, $41 = 101001_{\text{two}}$

The notation that we will use to describe this algorithm and all others is called pseudocode, an informal variation of the instructions that are commonly used in many computer languages. Read the following description carefully, comparing it with the informal description above. Appendix B, which contains a general discussion of the components of the algorithms in this book, should clear up any lingering questions. Anything after // are comments.

Algorithm 1.4.1: Binary Conversion Algorithm

An algorithm for determining the binary representation of a positive integer.

Input: a positive integer n.

Output: the binary representation of n in the form of a list of bits, with units bit last, twos bit next to last, etc.

1. k := n //initialize k
2. L := { } //initialize L to an empty list
3. While k > 0 do
   a. q := k div 2 //divide k by 2
   b. r:= k mod 2
   c. L: = prepend r to L //add r to the front of L
   d. k:= q //reassign k

Here is a Sage version of the algorithm with two alterations. It outputs the binary representation as a string, and it handles all integers, not just positive ones.

```
def binary_rep(n):
    if n==0:
        return '0'
    else:
        k=abs(n)
        s=''
        while k>0:
            s=str(k%2)+s
            k=k//2
        if n < 0:
            s='-'+s
        return s

binary_rep(41)
```

Now that you've read this section, you should get this joke.

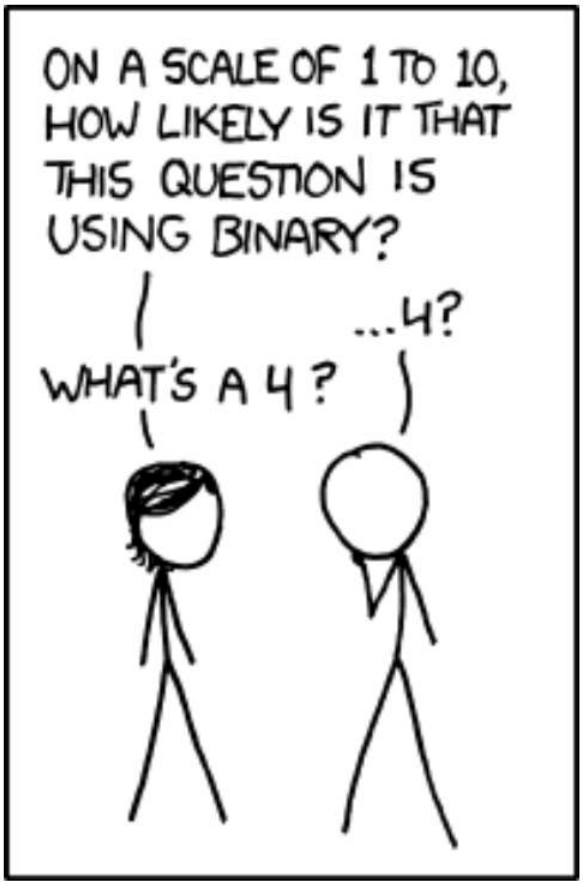


Figure 1.4.1: With permission from Randall Munroe, http://xkcd.com.

1.4.3: Exercises

Exercise 1.4.1

Find the binary representation of each of the following positive integers by working through the algorithm by hand. You can check your answer using the sage cell above.

a. 31
b. 32
c. 10
d. 100

Answer

a. 11111
b. 100000
c. 1010
d. 1100100

Exercise 1.4.2

Find the binary representation of each of the following positive integers by working through the algorithm by hand. You can check your answer using the sage cell above.

a. 64
b. 67
c. 28
d. 256

Exercise 1.4.3

What positive integers have the following binary representations?

a. 10010
b. 10011
c. 101010
d. 10011110000

Answer

a. 18
b. 19
c. 42
d. 1264

Exercise 1.4.4

What positive integers have the following binary representations?

a. 100001
b. 1001001
c. 1000000000
d. 1001110000

Exercise 1.4.5

The number of bits in the binary representations of integers increases by one as the numbers double. Using this fact, determine how many bits the binary representations of the following decimal numbers have without actually doing the full conversion.

a. 2017
b. 4000
c. 4500
d. 2^{50}

Answer

a. There is a bit for each power of 2 up to the largest one needed to represent an integer, and you start counting with the zeroth power. For example, 2017 is between $2^{10} = 1024$ and $2^{11} = 2048$, and so the largest power needed is 2^{10}. Therefore there are 11 bits in binary 2017.
b. 11
c. 12
d. 13
e. 51

Exercise 1.4.6

Let m be a positive integer with n-bit binary representation: $a_{n-1}a_{n-2}\cdots a_1a_0$ with $a_{n-1} = 1$ What are the smallest and largest values that m could have?

Exercise 1.4.7

If a positive integer is a multiple of 100, we can identify this fact from its decimal representation, since it will end with two zeros. What can you say about a positive integer if its binary representation ends with two zeros? What if it ends in k zeros?

Answer

A number must be a multiple of four if its binary representation ends in two zeros. If it ends in k zeros, it must be a multiple of 2^k.

Exercise 1.4.8

Can a multiple of ten be easily identified from its binary representation?

This page titled 1.4: Binary Representation of Positive Integers is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

1.5: Summation Notation and Generalizations

1.5.1: Sums

Most operations such as addition of numbers are introduced as binary operations. That is, we are taught that two numbers may be added together to give us a single number. Before long, we run into situations where more than two numbers are to be added. For example, if four numbers, a_1, a_2, a_3, and a_4 are to be added, their sum may be written down in several ways, such as $((a_1+a_2)+a_3)+a_4$ or $(a_1+a_2)+(a_3+a_4)$. In the first expression, the first two numbers are added, the result is added to the third number, and that result is added to the fourth number. In the second expression the first two numbers and the last two numbers are added and the results of these additions are added. Of course, we know that the final results will be the same. This is due to the fact that addition of numbers is an associative operation. For such operations, there is no need to describe how more than two objects will be operated on. A sum of numbers such as $a_1+a_2+a_3+a_4$ is called a series and is often written $\sum_{k=1}^{4} a_k$ in what is called *summation notation*.

We first recall some basic facts about series that you probably have seen before. A more formal treatment of sequences and series is covered in Chapter 8. The purpose here is to give the reader a working knowledge of summation notation and to carry this notation through to intersection and union of sets and other mathematical operations.

A *finite series* is an expression such as $a_1+a_2+a_3+\cdots+a_n=\sum_{k=1}^{n} a_k$

In the expression $\sum_{k=1}^{n} a_k$:

- The variable k is referred to as the *index*, or the index of summation.
- The expression a_k is the *general term* of the series. It defines the numbers that are being added together in the series.
- The value of k below the summation symbol is the *initial index* and the value above the summation symbol is the *terminal index*.
- It is understood that the series is a sum of the general terms where the index start with the initial index and increases by one up to and including the terminal index.

Example 1.5.1: Some Finite Series

a. $\displaystyle\sum_{i=1}^{4} a_i = a_1+a_2+a_3+a_4$

b. $\displaystyle\sum_{k=0}^{5} b_k = b_0+b_1+b_2+b_3+b_4+b_5$

c. $\displaystyle\sum_{i=-2}^{2} c_i = c_{-2}+c_{-1}+c_0+c_1+c_2$

Example 1.5.2: More Finite Series

If the general terms in a series are more specific, the sum can often be simplified. For example,

a. $\displaystyle\sum_{i=1}^{4} i^2 = 1^2+2^2+3^2+4^2 = 30$

b. $$\begin{aligned}\sum_{i=1}^{5}(2i-1) &= (2\cdot 1-1)+(2\cdot 2-1)+(2\cdot 3-1)+(2\cdot 4-1)+(2\cdot 5-1)\\ &= 1+3+5+7+9\\ &= 25\end{aligned}$$

.

1.5.2: Generalizations

Summation notation can be generalized to many mathematical operations, for example, $A_1\cap A_2\cap A_3\cap A_4=\bigcap_{i=1}^{4} A_i$

Definition 1.5.1: Generalized Set Operations

Let $A_1, A_2, \ldots, A_n$ be sets. Then:

a. $A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^{n} A_i$

b. $A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^{n} A_i$

c. $A_1 \times A_2 \times \cdots \times A_n = \bigtimes_{i=1}^{n} A_i$

d. $A_1 \oplus A_2 \oplus \cdots \oplus A_n = \bigoplus_{i=1}^{n} A_i$

Example 1.5.3: Some Generalized Options

If $A_1 = \{0, 2, 3\}$, $A_2 = \{1, 2, 3, 6\}$, and $A_3 = \{-1, 0, 3, 9\}$, then

$$\bigcap_{i=1}^{3} A_i = A_1 \cap A_2 \cap A_3 = \{3\}$$

and

$$\bigcup_{i=1}^{3} A_i = A_1 \cup A_2 \cup A_3 = \{-1, 0, 1, 2, 3, 6, 9\}.$$

With this notation it is quite easy to write lengthy expressions in a fairly compact form. For example, the statement

$$A \cap (B_1 \cup B_2 \cup \cdots \cup B_n) = (A \cap B_1) \cup (A \cap B_2) \cup \cdots \cup (A \cap B_n)$$

becomes

$$A \cap \left(\bigcup_{i=1}^{n} B_i \right) = \bigcup_{i=1}^{n} (A \cap B_i).$$

1.5.3: Exercises

Exercise 1.5.1

Calculate the following series:

a. $\sum_{i=1}^{3} (2 + 3i)$

b. $\sum_{i=-2}^{1} i^2$

c. $\sum_{j=0}^{n} 2^j$ for $n = 1, 2, 3, 4$

d. $\sum_{k=1}^{n} (2k - 1)$ for $n = 1, 2, 3, 4$

Answer

a. 24

b. 6

c. $3, 7, 15, 31$

d. $1, 4, 9, 16$

Exercise 1.5.2

Calculate the following series:

a. $\sum_{k=1}^{3} k^n$ for $n = 1, 2, 3, 4$

b. $\sum_{i=1}^{5} 20$
c. $\sum_{j=0}^{3} \left(n^j+1\right)$ for $n=1,2,3,4$
d. $\sum_{k=-n}^{n} k$ for $n=1,2,3,4$

Exercise 1.5.3

a. Express the formula $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$ without using summation notation.
b. Verify this formula for $n=3$.
c. Repeat parts (a) and (b) for $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$

Answer

a. $\frac{1}{1(1+1)}+\frac{1}{2(2+1)}+\frac{1}{3(3+1)}+\cdots+\frac{1}{n(n+1)}=\frac{n}{n+1}$
b. $\frac{1}{1(2)}+\frac{1}{2(3)}+\frac{1}{3(4)}=\frac{1}{2}+\frac{1}{6}+\frac{1}{12}=\frac{3}{4}=\frac{3}{3+1}$
c. $1+2^3+3^3+\cdots+n^3=\left(\frac{1}{4}\right)n^2(n+1)^2 \qquad 1+8+27=36=\left(\frac{1}{4}\right)(3)^2(3+1)^2$

Exercise 1.5.4

Verify the following properties for $n=3$.

a. $\sum_{i=1}^{n}(a_i+b_i)=\sum_{i=1}^{n}a_i+\sum_{i=1}^{n}b_i$
b. $c\left(\sum_{i=1}^{n}a_i\right)=\sum_{i=1}^{n}ca_i$

Exercise 1.5.5

Rewrite the following without summation sign for $n=3$. It is not necessary that you understand or expand the notation $\binom{n}{k}$ at this point. $(x+y)^n=\sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k$.

Answer

$(x+y)^3=\binom{3}{0}x^3+\binom{3}{1}x^2y+\binom{3}{2}xy^2+\binom{3}{3}y^n$

Exercise 1.5.6

a. Draw the Venn diagram for $\bigcap_{i=1}^{3} A_i$.
b. Express in "expanded format": $A\cup(\bigcap_{i=1}^{n}B_i)=\bigcap_{i=1}^{n}(A\cup B_n)$.

Exercise 1.5.7

For any positive integer k, let $A_k=\{x\in\mathbb{Q}: k-1<x\leq k\}$ and $B_k=\{x\in\mathbb{Q}:-k<x<k\}$. What are the following sets?

a. $\bigcup_{i=1}^{5} A_i$

b. $\bigcup_{i=1}^{5} B_i$
c. $\bigcap_{i=1}^{5} A_i$
d. $\bigcap_{i=1}^{5} B_i$

Answer

a. $\{x \in \mathbb{Q} \mid 0 < x \leq 5\}$
b. $\{x \in \mathbb{Q} \mid -5 < x < 5\} = B_5$
c. $\emptyset$
d. $\{x \in \mathbb{Q} \mid -1 < x < 1\} = B_1$

Exercise 1.5.8

For any positive integer k, let $A_k = \{x \in \mathbb{Q} : 0 < x < 1/k\}$ and $B_k = \{x \in \mathbb{Q} : 0 < x < k\}$. What are the following sets?

a. $\bigcup_{i=1}^{\infty} A_i$
b. $\bigcup_{i=1}^{\infty} B_i$
c. $\bigcap_{i=1}^{\infty} A_i$
d. $\bigcap_{i=1}^{\infty} B_i$

Exercise 1.5.9

The symbol Π is used for the product of numbers in the same way that Σ is used for sums. For example, $\prod_{i=1}^{5} x_i = x_1 x_2 x_3 x_4 x_5$. Evaluate the following:

a. $\prod_{i=1}^{3} i^2$
b. $\prod_{i=1}^{3} (2i+1)$

Answer

a. 36
b. 105

Exercise 1.5.10

Evaluate

a. $\prod_{k=0}^{3} 2^k$
b. $\prod_{k=1}^{100} \frac{k}{k+1}$

This page titled 1.5: Summation Notation and Generalizations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

2: Combinatorics

> ***Enumerative combinatorics***
>
> *Date back to the first prehistorics*
> *Who counted; relations*
> *Like sets' permutations*
> *To them were part cult, part folklorics*
>
> *Michael Toalster, The Omnificent English Dictionary in Limerick Form*

Throughout this book we will be counting things. In this chapter we will outline some of the tools that will help us count.

Counting occurs not only in highly sophisticated applications of mathematics to engineering and computer science but also in many basic applications. Like many other powerful and useful tools in mathematics, the concepts are simple; we only have to recognize when and how they can be applied.

2.1: Basic Counting Techniques - The Rule of Products
2.2: Permutations
2.3: Partitions of Sets and the Law of Addition
2.4: Combinations and the Binomial Theorem

This page titled 2: Combinatorics is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

2.1: Basic Counting Techniques - The Rule of Products

2.1.1: Combinatorics?

One of the first concepts our parents taught us was the "art of counting." We were taught to raise three fingers to indicate that we were three years old. The question of "how many" is a natural and frequently asked question. Combinatorics is the "art of counting." It is the study of techniques that will help us to count the number of objects in a set quickly. Highly sophisticated results can be obtained with this simple concept. The following examples will illustrate that many questions concerned with counting involve the same process.

Example 2.1.1: How Many Lunches Can You Have?

A snack bar serves five different sandwiches and three different beverages. How many different lunches can a person order? One way of determining the number of possible lunches is by listing or enumerating all the possibilities. One systematic way of doing this is by means of a tree, as in the following figure.

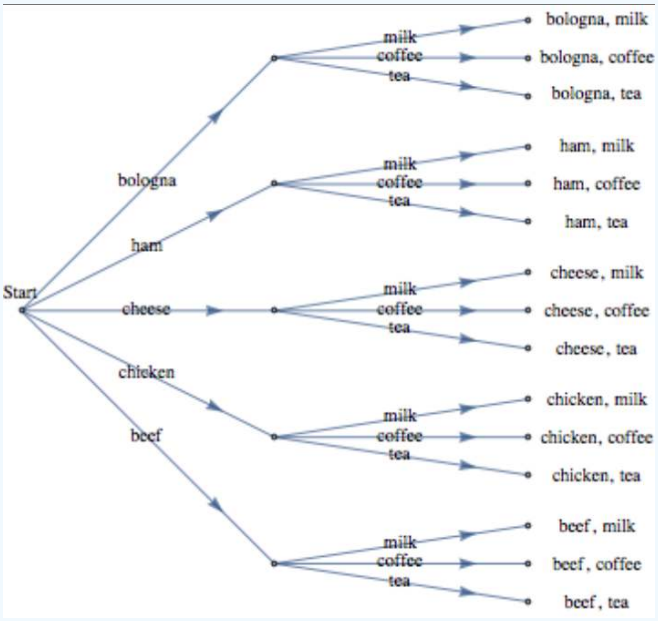


Figure 2.1.1: Tree diagram to enumerate the number of possible lunches.

Every path that begins at the position labeled START and goes to the right can be interpreted as a choice of one of the five sandwiches followed by a choice of one of the three beverages. Note that considerable work is required to arrive at the number fifteen this way; but we also get more than just a number. The result is a complete list of all possible lunches. If we need to answer a question that starts with "How many . . . ," enumeration would be done only as a last resort. In a later chapter we will examine more enumeration techniques.

An alternative method of solution for this example is to make the simple observation that there are five different choices for sandwiches and three different choices for beverages, so there are $5 \cdot 3 = 15$ different lunches that can be ordered.

Example 2.1.2: Counting Elements in a Cartesian Product

Let $A = \{a, b, c, d, e\}$ and $B = \{1, 2, 3\}$. From Chapter 1 we know how to list the elements in $A \times B = \{(a, 1), (a, 2), (a, 3), \ldots, (e, 3)\}$. Since the first entry of each pair can be any one of the five elements a, b, c, d, and e, and since the second can be any one of the three numbers 1, 2, and 3, it is quite clear there are $5 \cdot 3 = 15$ different elements in $A \times B$.

Example 2.1.3: A True-False Questionnaire

A person is to complete a true-false questionnaire consisting of ten questions. How many different ways are there to answer the questionnaire? Since each question can be answered in either of two ways (true or false), and there are ten questions, there are

$$2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10} = 1024$$

different ways of answering the questionnaire. The reader is encouraged to visualize the tree diagram of this example, but not to draw it!

We formalize the procedures developed in the previous examples with the following rule and its extension.

2.1.2: Rule Of Products

If two operations must be performed, and if the first operation can always be performed p_1 different ways and the second operation can always be performed p_2 different ways, then there are $p_1 p_2$ different ways that the two operations can be performed.

Note 2.1.1

It is important that p_2 does not depend on the option that is chosen in the first operation. Another way of saying this is that p_2 is independent of the first operation. If p_2 is dependent on the first operation, then the rule of products does not apply.

Example 2.1.4: Reduced Lunch Possibilities

Assume in Example 2.1.1, coffee is not served with a beef or chicken sandwiches. Then by inspection of Figure 2.1.1 we see that there are only thirteen different choices for lunch. The rule of products does not apply, since the choice of beverage depends on one's choice of a sandwich.

Extended Rule Of Products. The rule of products can be extended to include sequences of more than two operations. If n operations must be performed, and the number of options for each operation is p_1, $p_2, \ldots, p_n$ respectively, with each p_i independent of previous choices, then the n operations can be performed $p_1 \cdot p_2 \cdot \cdots \cdot p_n$ different ways.

Example 2.1.5: A Multiple Choice Questionnaire

A questionnaire contains four questions that have two possible answers and three questions with five possible answers. Since the answer to each question is independent of the answers to the other questions, the extended rule of products applies and there are $2 \cdot 2 \cdot 2 \cdot 2 \cdot 5 \cdot 5 \cdot 5 = 2^4 \cdot 5^3 = 2000$ different ways to answer the questionnaire.

In Chapter 1 we introduced the power set of a set A, $\mathcal{P}(A)$, which is the set of all subsets of A. Can we predict how many elements are in $\mathcal{P}(A)$ for a given finite set A? The answer is yes, and in fact if $|A| = n$, then $|\mathcal{P}(A)| = 2^n$. The ease with which we can prove this fact demonstrates the power and usefulness of the rule of products. Do not underestimate the usefulness of simple ideas.

Theorem 2.1.1: Power Set Cardinality Theorem

If A is a finite set, then $|\mathcal{P}(A)| = 2^{|A|}$.

Proof

Consider how we might determine any $B \in \mathcal{P}(A)$, where $|A| = n$. For each element $x \in A$ there are two choices, either $x \in B$ or $x \notin B$. Since there are n elements of A we have, by the rule of products,

$$\underbrace{2 \cdot 2 \cdot \cdots \cdot 2}_{n \text{ factors}} = 2^n$$

different subsets of A. Therefore, $\mathcal{P}(A) = 2^n$.

2.1.3: Exercises

Exercise 2.1.1

In horse racing, to bet the “daily double” is to select the winners of the first two races of the day. You win only if both selections are correct. In terms of the number of horses that are entered in the first two races, how many different daily double bets could be made?

Answer

If there are m horses in race 1 and n horses in race 2 then there are $m \cdot n$ possible daily doubles.

Exercise 2.1.2

Professor Shortcut records his grades using only his students' first and last initials. What is the smallest class size that will definitely force Prof. S. to use a different system?

Exercise 2.1.3

A certain shirt comes in four sizes and six colors. One also has the choice of a dragon, an alligator, or no emblem on the pocket. How many different shirts could you order?

Answer

$72 = 4 \cdot 6 \cdot 3$

Exercise 2.1.4

A builder of modular homes would like to impress his potential customers with the variety of styles of his houses. For each house there are blueprints for three different living rooms, four different bedroom configurations, and two different garage styles. In addition, the outside can be finished in cedar shingles or brick. How many different houses can be designed from these plans?

Exercise 2.1.5

The Pi Mu Epsilon mathematics honorary society of Outstanding University wishes to have a picture taken of its six officers. There will be two rows of three people. How many different way can the six officers be arranged?

Answer

$720 = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$

Exercise 2.1.6

An automobile dealer has several options available for each of three different packages of a particular model car: a choice of two styles of seats in three different colors, a choice of four different radios, and five different exteriors. How many choices of automobile does a customer have?

Exercise 2.1.7

A clothing manufacturer has put out a mix-and-match collection consisting of two blouses, two pairs of pants, a skirt, and a blazer. How many outfits can you make? Did you consider that the blazer is optional? How many outfits can you make if the manufacturer adds a sweater to the collection?

Answer

If we always include the blazer in the outfit we would have 6 outfits. If we consider the blazer optional then there would be 12 outfits. When we add a sweater we have the same type of choice. Considering the sweater optional produces 24 outfits.

Exercise 2.1.8

As a freshman, suppose you had to take two of four lab science courses, one of two literature courses, two of three math courses, and one of seven physical education courses. Disregarding possible time conflicts, how many different schedules do you have to choose from?

Exercise 2.1.9

a. Suppose each single character stored in a computer uses eight bits. Then each character is represented by a different sequence of eight 0's and 1's called a bit pattern. How many different bit patterns are there? (That is, how many different characters could be represented?)
b. How many bit patterns are palindromes (the same backwards as forwards)?
c. How many different bit patterns have an even number of 1's?

Answer

a. $2^8 = 256$
b. $2^4 = 16$. Here we are concerned only with the first four bits, since the last four must be the same.
c. $2^7 = 128$, you have no choice in the last bit.

Exercise 2.1.10

Automobile license plates in Massachusetts usually consist of three digits followed by three letters. The first digit is never zero. How many different plates of this type could be made?

Exercise 2.1.11

a. Let $A = \{1, 2, 3, 4\}$. Determine the number of different subsets of A.
b. Let $A = \{1, 2, 3, 4, 5\}$. Determine the number of proper subsets of A .

Answer

a. 16
b. 31

Exercise 2.1.12

How many integers from 100 to 999 can be written in base ten without using the digit 7?

Exercise 2.1.13

Consider three persons, A, B, and C, who are to be seated in a row of three chairs. Suppose A and B are identical twins. How many seating arrangements of these persons can there be

a. If you are a total stranger?
b. If you are A and B's mother?

This problem is designed to show you that different people can have different correct answers to the same problem.

Answer

a. 3
b. 6

Exercise 2.1.14

How many ways can a student do a ten-question true-false exam if he or she can choose not to answer any number of questions?

Exercise 2.1.15

Suppose you have a choice of fish, lamb, or beef for a main course, a choice of peas or carrots for a vegetable, and a choice of pie, cake, or ice cream for dessert. If you must order one item from each category, how many different dinners are possible?

Answer

18

Exercise 2.1.16

Suppose you have a choice of vanilla, chocolate, or strawberry for ice cream, a choice of peanuts or walnuts for chopped nuts, and a choice of hot fudge or marshmallow for topping. If you must order one item from each category, how many different sundaes are possible?

Exercise 2.1.17

A questionnaire contains six questions each having yes-no answers. For each yes response, there is a follow-up question with four possible responses.

a. Draw a tree diagram that illustrates how many ways a single question in the questionnaire can be answered.
b. How many ways can the questionnaire be answered?

Answer

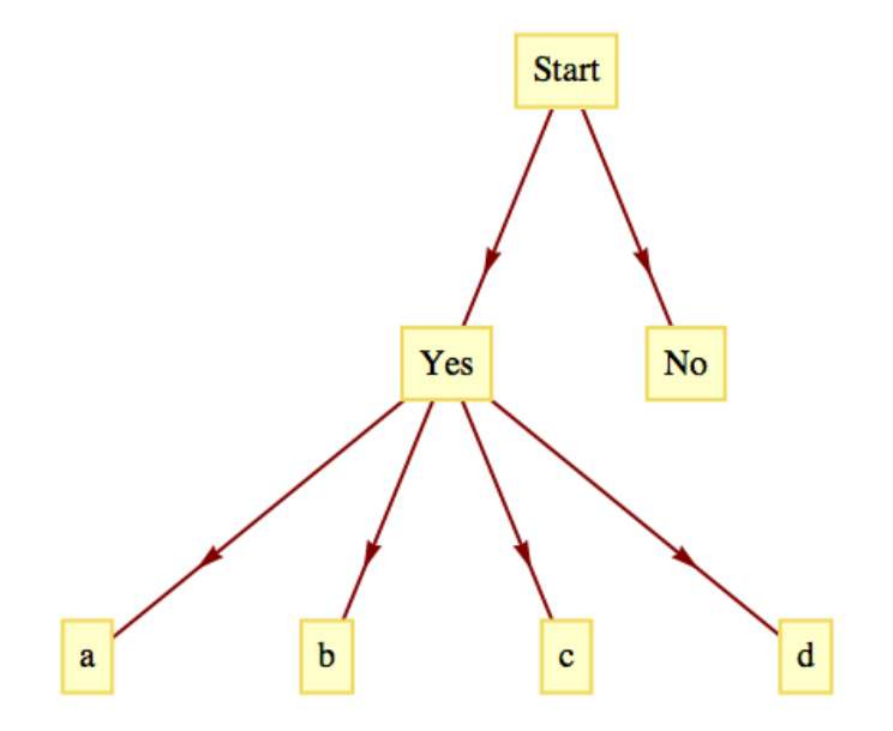


Figure 2.1.2 Solution to Exercise 2.1.17(a).

a. See Figure 2.1.2
b. 5^6

Exercise 2.1.18

Ten people are invited to a dinner party. How many ways are there of seating them at a round table? If the ten people consist of five men and five women, how many ways are there of seating them if each man must be surrounded by two women around the table?

Exercise 2.1.19

How many ways can you separate a set with n elements into two nonempty subsets if the order of the subsets is immaterial? What if the order of the subsets is important?

Answer

$2^{n-1}-1$ and 2^n-2

Exercise 2.1.20

A gardener has three flowering shrubs and four nonflowering shrubs, where all shrubs are distinguishable from one another. He must plant these shrubs in a row using an alternating pattern, that is, a shrub must be of a different type from that on either side. How many ways can he plant these shrubs? If he has to plant these shrubs in a circle using the same pattern, how many ways can he plant this circle? Note that one nonflowering shrub will be left out at the end.

This page titled 2.1: Basic Counting Techniques - The Rule of Products is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

2.2: Permutations

2.2.1: Ordering Things

A number of applications of the rule of products are of a specific type, and because of their frequent appearance they are given their own designation, permutations. Consider the following examples.

Example 2.2.1: Ordering the Elements of a Set

How many different ways can we order the three different elements of the set $A = \{a, b, c\}$? Since we have three choices for position one, two choices for position two, and one choice for the third position, we have, by the rule of products, $3 \cdot 2 \cdot 1 = 6$ different ways of ordering the three letters. We illustrate through a tree diagram.

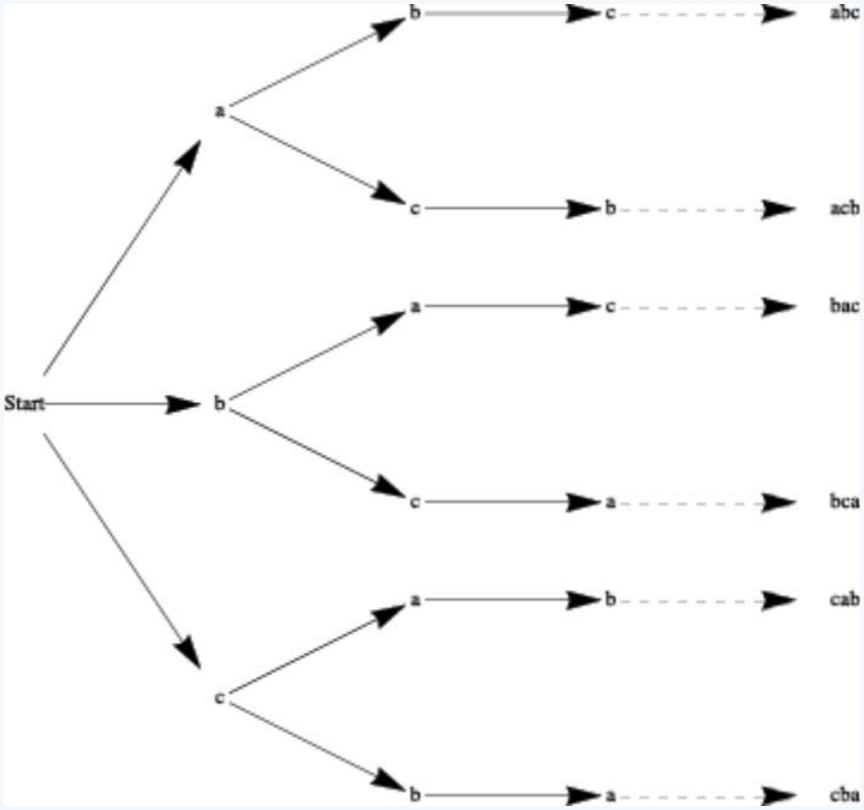


Figure 2.2.1: A tree to enumerate permutations of a three element set.

Each of the six orderings is called a permutation of the set A.

Example 2.2.2: Ordering a Schedule

A student is taking five courses in the fall semester. How many different ways can the five courses be listed? There are $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$ different permutations of the set of courses.

In each of the above examples of the rule of products we observe that:

1. We are asked to order or arrange elements from a single set.
2. Each element is listed exactly once in each list (permutation). So if there are n choices for position one in a list, there are $n - 1$ choices for position two, $n - 2$ choices for position three, etc.

Example 2.2.3: Some Orderings of a Baseball Team

The alphabetical ordering of the players of a baseball team is one permutation of the set of players. Other orderings of the players' names might be done by batting average, age, or height. The information that determines the ordering is called the key. We would expect that each key would give a different permutation of the names. If there are twenty-five players on the team, there are $25 \cdot 24 \cdot 23 \cdot \cdots \cdot 3 \cdot 2 \cdot 1$ different permutations of the players.

This number of permutations is huge. In fact it is 15511210043330985984000000, but writing it like this isn't all that instructive, while leaving it as a product as we originally had makes it easier to see where the number comes from. We just need to find a more compact way of writing these products.

We now develop notation that will be useful for permutation problems.

Definition 2.2.1: Factorial

If n is a positive integer then n factorial is the product of the first n positive integers and is denoted $n!$. Additionally, we define zero factorial, $0!$, to be 1.

The first few factorials are

$$\begin{array}{cccccccccc} n & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ n! & 1 & 1 & 2 & 6 & 24 & 120 & 720 & 5040 \end{array}.$$

Note that $4!$ is 4 times $3!$, or 24, and $5!$ is 5 times $4!$, or 120. In addition, note that as n grows in size, $n!$ grows extremely quickly. For example, $11! = 39916800$. If the answer to a problem happens to be $25!$, as in the previous example, you would never be expected to write that number out completely. However, a problem with an answer of $\frac{25!}{23!}$ can be reduced to $25 \cdot 24$, or 600.

If $|A| = n$, there are $n!$ ways of permuting all n elements of A . We next consider the more general situation where we would like to permute k elements out of a set of n objects, where $k \leq n$.

Example 2.2.4: Choosing Club Officers

A club of twenty-five members will hold an election for president, secretary, and treasurer in that order. Assume a person can hold only one position. How many ways are there of choosing these three officers? By the rule of products there are $25 \cdot 24 \cdot 23$ ways of making a selection.

Definition 2.2.2: Permutation

An ordered arrangement of k elements selected from a set of n elements, $0 \leq k \leq n$, where no two elements of the arrangement are the same, is called a permutation of n objects taken k at a time. The total number of such permutations is denoted by $P(n, k)$.

Theorem 2.2.1: Permutation Counting Formula

The number of possible permutations of k elements taken from a set of n elements is

$$P(n,k) = n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot (n-k+1) = \prod_{j=0}^{k-1} (n-j) = \frac{n!}{(n-k)!}.$$

Proof

Case I: If $k = n$ we have $P(n,n) = n! = \frac{n!}{(n-n)!}$.

Case II: If $0 \leq k < n$, then we have k positions to fill using n elements and

a. Position 1 can be filled by any one of $n - 0 = n$ elements
b. Position 2 can be filled by any one of $n - 1$ elements
c. $\cdots$

d. Position k can be filled by any one of $n-(k-1)=n-k+1$ elements

Hence, by the rule of products,

$$P(n,k)=n\cdot(n-1)\cdot(n-2)\cdots(n-k+1)=\frac{n!}{(n-k)!}.$$

It is important to note that the derivation of the permutation formula given above was done solely through the rule of products. This serves to reiterate our introductory remarks in this section that permutation problems are really rule-of-products problems. We close this section with several examples.

Example 2.2.5: Another Example of Choosing Officers

A club has eight members eligible to serve as president, vice-president, and treasurer. How many ways are there of choosing these officers?

Solution 1: Using the rule of products. There are eight possible choices for the presidency, seven for the vice-presidency, and six for the office of treasurer. By the rule of products there are $8\cdot7\cdot6=336$ ways of choosing these officers.

Solution 2: Using the permutation formula. We want the total number of permutations of eight objects taken three at a time:

$$P(8,3)=\frac{8!}{(8-3)!}=8\cdot7\cdot6=336$$

Example 2.2.6: Course Ordering, Revisited

To count the number of ways to order five courses, we can use the permutation formula. We want the number of permutations of five courses taken five at a time:

$$P(5,5)=\frac{5!}{(5-5)!}=5!=120.$$

Example 2.2.7: Ordering of Digits Under Different Conditions

Consider only the digits 1, 2, 3, 4, and 5.

a. How many three-digit numbers can be formed if no repetition of digits can occur?
b. How many three-digit numbers can be formed if repetition of digits is allowed?
c. How many three-digit numbers can be formed if only non-consecutive repetition of digits are allowed?

Solutions to (a): Solution 1: Using the rule of products. We have any one of five choices for digit one, any one of four choices for digit two, and three choices for digit three. Hence, $5\cdot4\cdot3=60$ different three-digit numbers can be formed.

Solution 2; Using the permutation formula. We want the total number of permutations of five digits taken three at a time:

$$P(5,3)=\frac{5!}{(5-3)!}=5\cdot4\cdot3=60.$$

Solution to (b): The definition of permutation indicates "...no two elements in each list are the same." Hence the permutation formula cannot be used. However, the rule of products still applies. We have any one of five choices for the first digit, five choices for the second, and five for the third. So there are $5\cdot5\cdot5=125$ possible different three-digit numbers if repetition is allowed.

Solution to (c): Again, the rule of products applies here. We have any one of five choices for the first digit, but then for the next two digits we have four choices since we are not allowed to repeat the previous digit So there are $5\cdot4\cdot4=80$ possible different three-digit numbers if only non-consecutive repetitions are allowed.

2.2.2: Exercises

Exercise 2.2.1

If a raffle has three different prizes and there are 1,000 raffle tickets sold, how many different ways can the prizes be distributed?

Answer

$P(1000, 3)$

Exercise 2.2.2

a. How many three-digit numbers can be formed from the digits 1, 2, 3 if no repetition of digits is allowed? List the three-digit numbers.
b. How many two-digit numbers can be formed if no repetition of digits is allowed? List them.
c. How many two-digit numbers can be obtained if repetition is allowed?

Exercise 2.2.3

How many eight-letter words can be formed from the 26 letters in the alphabet? Even without concerning ourselves about whether the words make sense, there are two interpretations of this problem. Answer both.

Answer

With repetition: $26^8 \approx 2.0883 \times 10^{11}$

Without repetition: $P(26, 8) \approx 6.2991\ 10^{10}$

Exercise 2.2.4

Let A be a set with $|A| = n$. Determine

a. $|A^3|$
b. $|\{(a, b, c) \mid a, b, c \in A \text{ and each coordinate is different}\}|$

Exercise 2.2.5

The state finals of a high school track meet involves fifteen schools. How many ways can these schools be listed in the program?

Answer

$15!$

Exercise 2.2.6

Consider the three-digit numbers that can be formed from the digits 1, 2, 3, 4, and 5 with no repetition of digits allowed.

a. How many of these are even numbers?
b. How many are greater than 250?

Exercise 2.2.7

All 15 players on the Tall U. basketball team are capable of playing any position.

a. How many ways can the coach at Tall U. fill the five starting positions in a game?
b. What is the answer if the center must be one of two players?

Answer

a. $P(15,5) = 360360$
b. $2 \cdot 14 \cdot 13 \cdot 12 \cdot 11 = 48048$

Exercise 2.2.8

a. How many ways can a gardener plant five different species of shrubs in a circle?
b. What is the answer if two of the shrubs are the same?
c. What is the answer if all the shrubs are identical?

Exercise 2.2.9

The president of the Math and Computer Club would like to arrange a meeting with six attendees, the president included. There will be three computer science majors and three math majors at the meeting. How many ways can the six people be seated at a circular table if the president does not want people with the same majors to sit next to one other?

Answer

$2 \cdot P(3,3) = 12$

Exercise 2.2.10

Six people apply for three identical jobs and all are qualified for the positions. Two will work in New York and the other one will work in San Diego. How many ways can the positions be filled?

Exercise 2.2.11

Let $A = \{1,2,3,4\}$. Determine the cardinality of

a. $\{(a_1, a_2) \mid a_1 \neq a_2\}$
b. What is the answer to the previous part if $|A| = n$
c. If $|A| = n$, determine the number of m-tuples in A, $m \leq n$, where each coordinate is different from the other coordinates.

Answer

a. $P(4,2) = 12$
b. $P(n;2) = n(n-1)$
c. Case 1: $m > n$. Since the coordinates must be different, this case is impossible.
Case 2: $m \leq n$. $P(n;m)$.

This page titled 2.2: Permutations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

2.3: Partitions of Sets and the Law of Addition

2.3.1: Partitions

One way of counting the number of students in your class would be to count the number in each row and to add these totals. Of course this problem is simple because there are no duplications, no person is sitting in two different rows. The basic counting technique that you used involves an extremely important first step, namely that of partitioning a set. The concept of a partition must be clearly understood before we proceed further.

Definition 2.3.1: Partition

A partition of set A is a set of one or more nonempty subsets of A: $A_1, A_2, A_3, \cdots$, such that every element of A is in exactly one set. Symbolically,

a. $A_1 \cup A_2 \cup A_3 \cup \cdots = A$
b. If $i \neq j$ then $A_i \cap A_j = \emptyset$

The subsets in a partition are often referred to as blocks. Note how our definition allows us to partition infinite sets, and to partition a set into an infinite number of subsets. Of course, if A is finite the number of subsets can be no larger than $|A|$.

Example 2.3.1: Some Partitions of a Four Element Set

Let $A = \{a, b, c, d\}$. Examples of partitions of A are:

- $\{\{a\}, \{b\}, \{c, d\}\}$
- $\{\{a, b\}, \{c, d\}\}$
- $\{\{a\}, \{b\}, \{c\}, \{d\}\}$

How many others are there, do you suppose?

There are 15 different partitions. The most efficient way to count them all is to classify them by the size of blocks. For example, the partition $\{\{a\}, \{b\}, \{c, d\}\}$has block sizes 1, 1, and 2.

Example 2.3.2: Some Integer Partitions

Two examples of partitions of set of integers $\mathbb{Z}$ are

- $\{\{n\} \mid n \in \mathbb{Z}\}$ and
- $\{\{n \in \mathbb{Z} \mid n < 0\}, \{0\}, \{n \in \mathbb{Z} \mid 0 < n\}\}$.

The set of subsets $\{\{n \in \mathbb{Z} \mid n \geq 0\}, \{n \in \mathbb{Z} \mid n \leq 0\}\}$ is not a partition because the two subsets have a nonempty intersection. A second example of a non-partition is $\{\{n \in \mathbb{Z} \mid |n| = k\} \mid k = -1, 0, 1, 2, \cdots\}$ because one of the blocks, when $k = -1$ is empty.

One could also think of the concept of partitioning a set as a "packaging problem." How can one "package" a carton of, say, twenty-four cans? We could use: four six-packs, three eight-packs, two twelve-packs, etc. In all cases: (a) the sum of all cans in all packs must be twenty-four, and (b) a can must be in one and only one pack.

2.3.2: Addition Laws

Theorem 2.3.1: The Basic Law of Addition

If A is a finite set, and if $\{A_1, A_2, \ldots, A_n\}$ is a partition of A , then

$$|A| = |A_1| + |A_2| + \cdots + |A_n| = \sum_{k=1}^{n} |A_k|$$

The basic law of addition can be rephrased as follows: If A is a finite set where $A_1 \cup A_2 \cup \cdots \cup A_n = A$ and where $A_i \cap A_j = \emptyset$ whenever $i \neq j$, then

$$|A| = |A_1 \cup A_2 \cup \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|$$

Example 2.3.3: Counting All Students

The number of students in a class could be determined by adding the numbers of students who are freshmen, sophomores, juniors, and seniors, and those who belong to none of these categories. However, you probably couldn't add the students by major, since some students may have double majors.

Example 2.3.4: Counting Students in Disjoint Classes

The sophomore computer science majors were told they must take one and only one of the following courses that are open only to them: Cryptography, Data Structures, or Javascript. The numbers in each course, respectively, for sophomore CS majors, were 75, 60, 55. How many sophomore CS majors are there? The Law of Addition applies here. There are exactly $75 + 60 + 55 = 190$ CS majors since the rosters of the three courses listed above would be a partition of the CS majors.

Example 2.3.5: Counting Students in Non-Disjoint Classes

It was determined that all junior computer science majors take at least one of the following courses: Algorithms, Logic Design, and Compiler Construction. Assume the number in each course was 75, 60 and 55, respectively for the three courses listed. Further investigation indicated ten juniors took all three courses, twenty-five took Algorithms and Logic Design, twelve took Algorithms and Compiler Construction, and fifteen took Logic Design and Compiler Construction. How many junior C.S. majors are there?

Example 2.3.4 was a simple application of the law of addition, however in this example some students are taking two or more courses, so a simple application of the law of addition would lead to double or triple counting. We rephrase information in the language of sets to describe the situation more explicitly.

A = the set of all junior computer science majors

A_1 = the set of all junior computer science majors who took Algorithms

A_2 = the set of all junior computer science majors who took Logic Design

A_3 = the set of all junior computer science majors who took Compiler Construction

Since all junior CS majors must take at least one of the courses, the number we want is:

$$|A| = |A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3| - \text{repeats}.$$

A Venn diagram is helpful to visualize the problem. In this case the universal set U can stand for all students in the university.

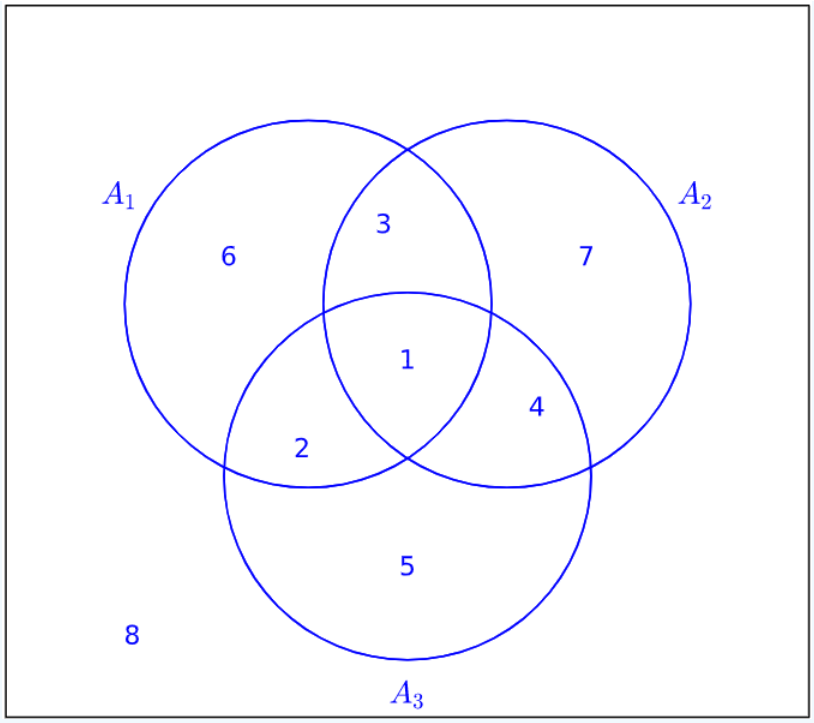


Figure 2.3.1: Venn Diagram

We see that the whole universal set is naturally partitioned into subsets that are labeled by the numbers 1 through 8, and the set A is partitioned into subsets labeled 1 through 7. The region labeled 8 represents all students who are not junior CS majors. Note also that students in the subsets labeled 2, 3, and 4 are double counted, and those in the subset labeled 1 are triple counted. To adjust, we must subtract the numbers in regions 2, 3 and 4. This can be done by subtracting the numbers in the intersections of each pair of sets. However, the individuals in region 1 will have been removed three times, just as they had been originally added three times. Therefore, we must finally add their number back in.

$$\begin{aligned}
|A| &= |A_1 \cup A_2 \cup A_3| \\
&= |A_1| + |A_2| + |A_3| - \text{repeats} \\
&= |A_1| + |A_2| + |A_3| - \text{duplicates} + \text{triplicates} \\
&= |A_1| + |A_2| + |A_3| - (|A_1 \cap A_2| + |A_1 \cap A_3| + |A_2 \cap A_3|) + |A_1 \cap A_2 \cap A_3| \\
&= 75 + 60 + 55 - 25 - 12 - 15 + 10 = 148
\end{aligned}$$

The ideas used in this latest example gives rise to a basic counting technique:

Theorem 2.3.2: Laws of Inclusion-Exclusion

Given finite sets A_1, A_2, A_3, then

a. The Two Set Inclusion-Exclusion Law:

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

b. The Three Set Inclusion-Exclusion Law:

$$\begin{aligned}
|A_1 \cup A_2 \cup A_3| &= |A_1| + |A_2| + |A_3| \\
&\quad - (|A_1 \cap A_2| + |A_1 \cap A_3| + |A_2 \cap A_3|) \\
&\quad + |A_1 \cap A_2 \cap A_3|
\end{aligned}$$

The inclusion-exclusion laws extend to more than three sets, as will be explored in the exercises.

In this section we saw that being able to partition a set into disjoint subsets gives rise to a handy counting technique. Given a set, there are many ways to partition depending on what one would wish to accomplish. One natural partitioning of sets is apparent when one draws a Venn diagram. This particular partitioning of a set will be discussed further in Chapters 4 and 13.

2.3.3: Exercises

Exercise 2.3.1

List all partitions of the set $A = \{a, b, c\}$.

Answer

$\{\{a\},\{b\},\{c\}\}, \{\{a,b\},\{c\}\}, \{\{a,c\},\{b\}\}, \{\{a\},\{b,c\}\}, \{\{a,b,c\}\}$

Exercise 2.3.2

Which of the following collections of subsets of the plane, $\mathbb{R}^2$, are partitions?

a. $\{\{(x,y) \mid x+y=c\} \mid c \in \mathbb{R}\}$
b. The set of all circles in $\mathbb{R}^2$
c. The set of all circles in $\mathbb{R}^2$ centered at the origin together with the set $\{(0,0)\}$
d. $\{\{(x,y)\} \mid (x,y) \in \mathbb{R}^2\}$

Exercise 2.3.3

A student, on an exam paper, defined the term partition the following way: "Let A be a set. A partition of A is any set of nonempty subsets $A_1, A_2, A_3, \ldots$ of A such that each element of A is in one of the subsets." Is this definition correct? Why?

Answer

No. By this definition it is possible that an element of A might belong to two of the subsets.

Exercise 2.3.4

Let A_1 and A_2 be subsets of a set U. Draw a Venn diagram of this situation and shade in the subsets $A_1 \cap A_2$, $A_1^c \cap A_2$, $A_1 \cap A_2^c$, and $A_1^c \cap A_2^c$. Use the resulting diagram and the definition of partition to convince yourself that the subset of these four subsets that are nonempty form a partition of U.

Exercise 2.3.5

Show that $\{\{2n \mid n \in \mathbb{Z}\}, \{2n+1 \mid n \in \mathbb{Z}\}\}$ is a partition of $\mathbb{Z}$. Describe this partition using only words.

Answer

The first subset is all the even integers and the second is all the odd integers. These two sets do not intersect and they cover the integers completely.

Exercise 2.3.6

a. A group of 30 students were surveyed and it was found that 18 of them took Calculus and 12 took Physics. If all students took at least one course, how many took both Calculus and Physics? Illustrate using a Venn diagram.
b. What is the answer to the question in part (a) if five students did not take either of the two courses? Illustrate using a Venn diagram.

Exercise 2.3.7

A survey of 90 people, 47 of them played tennis and 42 of them swam. If 17 of them participated in both activities, how many of them participated in neither?

Answer

Since 17 participated in both activities, 30 of the tennis players only played tennis and 25 of the swimmers only swam. Therefore, $17+30+25=72$ of those who were surveyed participated in an activity and so 18 did not.

Exercise 2.3.8

A survey of 300 people found that 60 owned an iPhone, 75 owned a Blackberry, and 30 owned an Android phone. Furthermore, 40 owned both an iPhone and a Blackberry, 12 owned both an iPhone and an Android phone, and 8 owned a Blackberry and an Android phone. Finally, 3 owned all three phones.

a. How many people surveyed owned none of the three phones?
b. How many people owned a Blackberry but not an iPhone?
c. How many owned a Blackberry but not an Android?

Exercise 2.3.9

Regarding Theorem 2.3.2,

a. Use the two set inclusion-exclusion law to derive the three set inclusion-exclusion law. Note: A knowledge of basic set laws is needed for this exercise.
b. State and derive the inclusion-exclusion law for four sets.

Answer

We assume that $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$.

$$\begin{aligned} |A_1 \cup A_2 \cup A_3| &= |(A_1 \cup A_2) \cup A_3| \quad Why? \\ &= |A_1 \cup A_2| + |A_3| - |(A_1 \cup A_2) \cap A_3| \quad Why? \\ &= |(A_1 \cup A_2| + |A_3| - |(A_1 \cap A_3) \cup (A_2 \cap A_3)| \quad Why? \\ &= |A_1| + |A_2| - |A_1 \cap A_2| + |A_3| \\ &\quad - (|A_1 \cap A_3| + |A_2 \cap A_3| - |(A_1 \cap A_3) \cap (A_2 \cap A_3)| \quad Why? \\ &= |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_1 \cap A_3| \\ &\quad - |A_2 \cap A_3| + |A_1 \cap A_2 \cap A_3| \quad Why? \end{aligned}$$

The law for four sets is

$$\begin{aligned} |A_1 \cup A_2 \cup A_3 \cup A_4| &= |A_1| + |A_2| + |A_3| + |A_4| \\ &\quad - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_1 \cap A_4| \\ &\quad - |A_2 \cap A_3| - |A_2 \cap A_4| - |A_3 \cap A_4| \\ &\quad + |A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| \\ &\quad + |A_1 \cap A_3 \cap A_4| + |A_2 \cap A_3 \cap A_4| \\ &\quad - |A_1 \cap A_2 \cap A_3 \cap A_4| \end{aligned}$$

Derivation:

$$
\begin{aligned}
|A_1 \cup A_2 \cup A_3 \cup A_4| &= |(A_1 \cup A_2 \cup A_3) \cup A_4| \\
&= |(A_1 \cup A_2 \cup A_3| + |A_4| - |(A_1 \cup A_2 \cup A_3) \cap A_4| \\
&= |(A_1 \cup A_2 \cup A_3| + |A_4| \\
&\quad - |(A_1 \cap A_4) \cup (A_2 \cap A_4) \cup (A_3 \cap A_4)| \\
&= |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_1 \cap A_3| \\
&\quad - |A_2 \cap A_3| + |A_1 \cap A_2 \cap A_3| + |A_4| - |A_1 \cap A_4| \\
&\quad + |A_2 \cap A_4| + |A_3 \cap A_4| - |(A_1 \cap A_4) \cap (A_2 \cap A_4)| \\
&\quad - |(A_1 \cap A_4) \cap (A_3 \cap A_4)| - |(A_2 \cap A_4) \cap (A_3 \cap A_4)| \\
&\quad + |(A_1 \cap A_4) \cap (A_2 \cap A_4) \cap (A_3 \cap A_4)| \\
&= |A_1| + |A_2| + |A_3| + |A_4| - |A_1 \cap A_2| - |A_1 \cap A_3| \\
&\quad - |A_2 \cap A_3| - |A_1 \cap A_4| - |A_2 \cap A_4| \quad - |A_3 \cap A_4| \\
&\quad + |A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| \\
&\quad + |A_1 \cap A_3 \cap A_4| + |A_2 \cap A_3 \cap A_4| \\
&\quad - |A_1 \cap A_2 \cap A_3 \cap A_4|
\end{aligned}
$$

Exercise 2.3.10

To complete your spring schedule, you must add Calculus and Physics. At 9:30, there are three Calculus sections and two Physics sections; while at 11:30, there are two Calculus sections and three Physics sections. How many ways can you complete your schedule if your only open periods are 9:30 and 11:30?

Exercise 2.3.11

The definition of $\mathbb{Q} = \{a/b \mid a, b \in \mathbb{Z}, b \neq 0\}$ given in Chapter 1 is awkward. If we use the definition to list elements in $\mathbb{Q}$, we will have duplications such as $\frac{1}{2}$, $\frac{-2}{-4}$ and $\frac{300}{600}$ Try to write a more precise definition of the rational numbers so that there is no duplication of elements.

Answer

Partition the set of fractions into blocks, where each block contains fractions that are numerically equivalent. Describe how you would determine whether two fractions belong to the same block. Redefine the rational numbers to be this partition. Each rational number is a set of fractions.

This page titled 2.3: Partitions of Sets and the Law of Addition is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

2.4: Combinations and the Binomial Theorem

2.4.1: Combinations

In Section 2.1 we investigated the most basic concept in combinatorics, namely, the rule of products. It is of paramount importance to keep this fundamental rule in mind. In Section 2.2 we saw a subclass of rule-of-products problems, permutations, and we derived a formula as a computational aid to assist us. In this section we will investigate another counting formula, one that is used to count combinations, which are subsets of a certain size.

In many rule-of-products applications the ordering is important, such as the batting order of a baseball team. In other cases it is not important, as in placing coins in a vending machine or in the listing of the elements of a set. Order is important in permutations. Order is not important in combinations.

Example 2.4.1: Counting Permutations

How many different ways are there to permute three letters from the set $A = \{a, b, c, d\}$? From the Permutation Counting Formula, Theorem 2.2.1, there are $P(4, 3) = \frac{4!}{(4-3)!} = 24$ different orderings of three letters from A

Example 2.4.2: Counting with No Order

How many ways can we select a set of three letters from $A = \{a, b, c, d\}$? Note here that we are not concerned with the order of the three letters. By trial and error, abc, abd, acd, and bcd are the only listings possible. To repeat, we were looking for all three-element subsets of the set A. Order is not important in sets. The notation for choosing 3 elements from 4 is most commonly $\binom{4}{3}$ or occasionally $C(4, 3)$, either of which is read "4 choose 3" or the number of combinations for four objects taken three at a time.

Definition 2.4.1: Binomial Coefficient

Let n and k be nonnegative integers. The binomial coefficient $\binom{n}{k}$ represents the number of combinations of n objects taken k at a time, and is read "n choose k."

We would now like to investigate the relationship between permutation and combination problems in order to derive a formula for $\binom{n}{k}$

Let us reconsider the Counting with No Order, Example 2.4.2. There are $3! = 6$ different orderings for each of the three-element subsets. The table below lists each subset of A and all permutations of each subset on the same line.

subset	permutations
$\{a, b, c\}$	$abc, acb, bca, bac, cab, cba$
$\{a, b, d\}$	$abd, adb, bda, bad, dab, dba.$
$\{a, c, d\}$	$acd, adc, cda, cad, dac, dca$
$\{b, c, d\}$	$bcd, bdc, cdb, cbd, dbc, dcb$

Hence, $\binom{4}{3} = \frac{P(4,3)}{3!} = \frac{4!}{(4-3)!\cdot 3!} = 4$

We generalize this result in the following theorem:

Theorem 2.4.1: Binomial Coefficient Formula

If n and k are nonnegative integers with $0 \leq k \leq n$, then the number k-element subsets of an n element set is equal to

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!}.$$

Proof

Proof 1: There are $k!$ ways of ordering the elements of any k element set. Therefore,

$$\binom{n}{k} = \frac{P(n,k)}{k!} = \frac{n!}{(n-k)!k!}..$$

Proof 2: To "construct" a permutation of k objects from a set of n elements, we can first choose one of the subsets of objects and second, choose one of the $k!$ permutations of those objects. By the rule of products,

$$P(n,k) = \binom{n}{k} \cdot k!$$

and solving for $\binom{n}{k}$ we get the desired formula.

Example 2.4.3: Flipping Coins

Assume an evenly balanced coin is tossed five times. In how many ways can three heads be obtained? This is a combination problem, because the order in which the heads appear does not matter. We can think of this as a situation involving sets by considering the set of flips of the coin, 1 through 5, in which heads comes up. The number of ways to get three heads is $\binom{5}{3} = \frac{5 \cdot 4}{2 \cdot 1} = 10$.

Example 2.4.4: Counting Five Ordered Flips Two Ways

We determine the total number of ordered ways a fair coin can land if tossed five consecutive times. The five tosses can produce any one of the following mutually exclusive, disjoint events: 5 heads, 4 heads, 3 heads, 2 heads, 1 head, or 0 heads. For example, by the previous example, there are $\binom{5}{3} = 10$ sequences in which three heads appear. Counting the other possibilities in the same way, by the law of addition we have:

$$\binom{5}{5} + \binom{5}{4} + \binom{5}{3} + \binom{5}{2} + \binom{5}{1} + \binom{5}{0} = 1 + 5 + 10 + 10 + 5 + 1 = 32$$

ways to observe the five flips.

Of course, we could also have applied the extended rule of products, and since there are two possible outcomes for each of the five tosses, we have $2^5 = 32$ ways.

You might think that counting something two ways is a waste of time but solving a problem two different ways often is instructive and leads to valuable insights. In this case, it suggests a general formula for the sum $\sum_{k=0}^{n} \binom{n}{k}$. In the case of $n = 5$, we get 2^5 so it is reasonable to expect that the general sum is 2^n, and it is. A logical argument to prove the general statment simply involves generalizing the previous example to n coin flips.

Example 2.4.5: A Committee of Five

A committee usually starts as an unstructured set of people selected from a larger membership. Therefore, a committee can be thought of as a combination. If a club of 25 members has a five-member social committee, there are $\binom{25}{5} = \frac{25 \cdot 24 \cdot 23 \cdot 22 \cdot 21}{5!} = 53130$ different possible social committees. If any structure or restriction is placed on the way the social committee is to be selected, the number of possible committees will probably change. For example, if the club has a rule that the treasurer must be on the social committee, then the number of possibilities is reduced to $\binom{24}{4} = \frac{24 \cdot 23 \cdot 22 \cdot 21}{4!} = 10626$.

If we further require that a chairperson other than the treasurer be selected for the social committee, we have $\binom{24}{4} \cdot 4 = 42504$ different possible social committees. The choice of the four non-treasurers accounts for the factor $\binom{24}{4}$ while the need to choose a chairperson accounts for the 4.

Example 2.4.6: Binomial Coefficients - Extreme Cases

By simply applying the definition of a Binomial Coefficient, Definition 2.4.1, as a number of subsets we see that there is $\binom{n}{0} = 1$ way of choosing a combination of zero elements from a set of n. In addition, we see that there is $\binom{n}{n} = 1$ way of choosing a combination of n elements from a set of n.

We could compute these values using the formula we have developed, but no arithmetic is really needed here. Other properties of binomial coefficients that can be derived using the subset definition will be seen in the exercises

2.4.2: Binomial Theorem

The binomial theorem gives us a formula for expanding $(x+y)^n$, where n is a nonnegative integer. The coefficients of this expansion are precisely the binomial coefficients that we have used to count combinations. Using high school algebra we can expand the expression for integers from 0 to 5:

n	$(x+y)^n$
0	1
1	$x+y$
2	$x^2+2xy+y^2$
3	$x^3+3x^2y+3xy^2+y^3$
4	$x^4+4x^3y+6x^2y^2+4xy^3+y^4$
5	$x^5+5x^4y+10x^3y^2+10x^2y^3+5xy^4+y^5$

In the expansion of $(x+y)^5$ we note that the coefficient of the third term is $\binom{5}{3}=10$, and that of the sixth term is $\binom{5}{5}=1$. We can rewrite the expansion as

$$\binom{5}{0}x^5+\binom{5}{1}x^4y+\binom{5}{2}x^3y^2+\binom{5}{3}x^2y^3+\binom{5}{4}xy^4+\binom{5}{5}y^5.$$

In summary, in the expansion of $(x+y)^n$ we note:

a. The first term is x^n and the last term is y^n.
b. With each successive term, exponents of x decrease by 1 as those of y increase by 1. For any term the sum of the exponents is n.
c. The coefficient of $x^{n-k}y^k$ is $\binom{n}{k}$.
d. The triangular array of binomial coefficients is called Pascal's triangle after the seventeenth-century French mathematician Blaise Pascal. Note that each number in the triangle other than the 1's at the ends of each row is the sum of the two numbers to the right and left of it in the row above.

Theorem 2.4.2: The Binomial Theorem

If $n \geq 0$, and x and y are numbers, then

$$(x+y)^n=\sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k.$$

Proof

This theorem will be proven using a logical procedure called mathematical induction, which will be introduced in Chapter 3.

Example 2.4.7: Identifying a Term in an Expansion

Find the third term in the expansion of $(x-y)^4=(x+(-y))^4$. The third term, when $k=2$, is $\binom{4}{2}x^{4-2}(-y)^2=6x^2y^2$.

Example 2.4.8: A Binomial Expansion

Expand $(3x-2)^3$. If we replace x and y in the Binomial Theorem with $3x$ and -2, respectively, we get

$$\begin{aligned}\sum_{k=0}^{3}\binom{3}{k}(3x)^{n-k}(-2)^k &= \binom{3}{0}(3x)^3(-2)^0+\binom{3}{1}(3x)^2(-2)^1+\binom{3}{2}(3x)^1(-2)^2+\binom{3}{3}(3x)^0(-2)^3 \\ &= 27x^3-54x^2+36x-8\end{aligned}.$$

2.4.3: SageMath Note

A bridge hand is a 13 element subset of a standard 52 card deck. The order in which the cards come to the player doesn't matter. From the point of view of a single player, the number of possible bridge hands is $\binom{52}{13}$, which can be easily computed with $Sage$.

```
1 | binomial(52,13)
```

In bridge, the location of a hand in relation to the dealer has some bearing on the game. An even truer indication of the number of possible hands takes into account $each$ player's possible hand. It is customary to refer to bridge positions as West, North, East and South. We can apply the rule of product to get the total number of bridge hands with the following logic. West can get any of the $\binom{52}{13}$ hands identified above. Then North get 13 of the remaining 39 cards and so has $\binom{39}{13}$ possible hands. East then gets 13 of the 26 remaining cards, which has $\binom{26}{13}$ possibilities. South gets the remaining cards. Therefore the number of bridge hands is computed using the Product Rule.

```
1 | binomial(52,13)*binomial(39,13)*binomial(26,13)
```

2.4.4: Exercises

Exercise 2.4.1

The judiciary committee at a college is made up of three faculty members and four students. If ten faculty members and 25 students have been nominated for the committee, how many judiciary committees could be formed at this point?

Answer

$\binom{10}{3} \cdot \binom{25}{4} = 1,518,000$

Exercise 2.4.2

Suppose that a single character is stored in a computer using eight bits.

a. How many bit patterns have exactly three 1's?
b. How many bit patterns have at least two 1's?

Hint

Think of the set of positions that contain a 1 to turn this is into a question about sets.

Answer

a. $\binom{8}{3}$
b. $2^8 - (\binom{8}{0} + \binom{8}{1})$

Exercise 2.4.3

How many subsets of $\{1, 2, 3, \ldots, 10\}$contain at least seven elements?

Answer

$\binom{10}{7} + \binom{10}{8} + \binom{10}{9} + \binom{10}{10} = 120 + 45 + 10 + 1 = 176$

Exercise 2.4.4

The congressional committees on mathematics and computer science are made up of five representatives each, and a congressional rule is that the two committees must be disjoint. If there are 385 members of congress, how many ways could the committees be selected?

Exercise 2.4.5

The image below shows a 6 by 6 grid and an example of a *lattice path* that could be taken from $(0,0)$ to $(6,6)$, which is a path taken by traveling along grid lines going only to the right and up. How many different lattice paths are there of this type? Generalize to the case of lattice paths from $(0,0)$ to (m,n) for any nonnegative integers m and n.

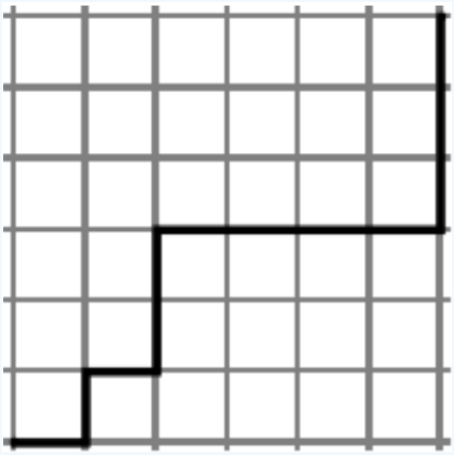


Figure 2.4.1: A lattice path

Hint

Think of each path as a sequence of instructions to go right (R) and up (U).

Answer

Each path can be described as a sequence or R's and U's with exactly six of each. The six positions in which R's could be placed can be selected from the twelve positions in the sequence $\binom{12}{6}$ ways. We can generalize this logic and see that there are $\binom{m+n}{m}$ paths from $(0,0)$ to (m,n).

Exercise 2.4.6

How many of the lattice paths from $(0,0)$ to $(6,6)$ pass through $(3,3)$ as the one in Figure 2.4.1 does?

Exercise 2.4.7

A poker game is played with 52 cards. At the start of a game, each player gets five of the cards. The order in which cards are dealt doesn't matter.

a. How many “hands” of five cards are possible?
b. If there are four people playing, how many initial five-card “hands” are possible, taking into account all players and their positions at the table? Position with respect to the dealer does matter.

Answer

a. $C(52,5) = 2,598,960$
b. $\binom{52}{5} \cdot \binom{47}{5} \cdot \binom{42}{5} \cdot \binom{37}{5}$

Exercise 2.4.8

A flush in a five-card poker hand is five cards of the same suit. The suits are spades, clubs, diamonds and hearts. How many spade flushes are possible in a 52-card deck? How many flushes are possible in any suit?

Exercise 2.4.9

How many five-card poker hands using 52 cards contain exactly two aces?

Answer

$\binom{4}{2} \cdot \binom{48}{3} = 6 \cdot 17296 = 103776$

Exercise 2.4.10

In poker, a full house is three-of-a-kind and a pair in one hand; for example, three fives and two queens. How many full houses are possible from a 52-card deck? You can use the sage cell in the SageMath Note to do this calculation, but also write your answer in terms of binomial coefficients.

Exercise 2.4.11

A class of twelve computer science students are to be divided into three groups of 3, 4, and 5 students to work on a project. How many ways can this be done if every student is to be in exactly one group?

Answer

$\binom{12}{3} \cdot \binom{9}{4} \cdot \binom{5}{5}$

Exercise 2.4.12

Explain in words why the following equalities are true based on number of subsets, and then verify the equalities using the formula for binomial coefficients.

a. $\binom{n}{1} = n$
b. $\binom{n}{k} = \binom{n}{n-k}, 0 \leq k \leq n$

Exercise 2.4.13

There are ten points, $P_1, P_2, \ldots, P_{10}$ on a plane, no three on the same line.

a. How many lines are determined by the points?
b. How many triangles are determined by the points?

Answer

a. $\binom{10}{2} = 45$
b. $\binom{10}{3} = 120$

Exercise 2.4.14

How many ways can n persons be grouped into pairs when n is even? Assume the order of the pairs matters, but not the order within the pairs. For example, if $n = 4$, the six different groupings would be

$\{1,2\}$ $\{3,4\}$
$\{1,3\}$ $\{2,4\}$
$\{1,4\}$ $\{2,3\}$
$\{2,3\}$ $\{1,4\}$
$\{2,4\}$ $\{1,3\}$
$\{3,4\}$ $\{1,2\}$

Exercise 2.4.15

Use the binomial theorem to prove that if A is a finite set, then $|P(A)| = 2^{|A|}$

Answer

Assume $|A| = n$. If we let $x = y = 1$ in the Binomial Theorem, we obtain $2^n = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n}$, with the right side of the equality counting all subsets of A containing $0, 1, 2, \ldots, n$elements. Hence $|P(A)| = 2^{|A|}$

Exercise 2.4.16

a. A state's lottery involves choosing six different numbers out of a possible 36. How many ways can a person choose six numbers?
b. What is the probability of a person winning with one bet?

Exercise 2.4.17

Use the binomial theorem to calculate 9998^3.

Hint

$9998 = 10000 - 2$

Answer

$10000^3 - 3 \cdot 2 \cdot 10000^2 + 3 \cdot 2^2 \cdot 10000 - 2^3 = 999,400,119,992.$

Exercise 2.4.18

In the card game Blackjack, there are one or more players and a dealer. Initially, each player is dealt two cards and the dealer is dealt one card down and one facing up. As in bridge, the order of the hands, but not the order of the cards in the hands, matters. Starting with a single 52 card deck, and three players, how many ways can the first two cards be dealt out? You can use the sage cell in the SageMath Note to do this calculation.

This page titled 2.4: Combinations and the Binomial Theorem is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

3: Logic

> ***formal logic***
>
> *You can write any letters you choose;*
> ***Formal logic**, though, like p's and q's*
> *To form statements — a lot,*
> *Using IF, OR, AND, NOT —*
> *To determine the falses and trues.*
>
> *Goldie, The Omnificent English Dictionary in Limerick Form*

In this chapter, we will introduce some of the basic concepts of mathematical logic. In order to fully understand some of the later concepts in this book, you must be able to recognize valid logical arguments. Although these arguments will usually be applied to mathematics, they employ the same techniques that are used by a lawyer in a courtroom or a physician examining a patient. An added reason for the importance of this chapter is that the circuits that make up digital computers are designed using the same algebra of propositions that we will be discussing.

3.1: Propositions and Logical Operators
3.2: Truth Tables and Propositions Generated by a Set
3.3: Equivalence and Implication
3.4: The Laws of Logic
3.5: Mathematical Systems and Proofs
3.6: Propositions over a Universe
3.7: Mathematical Induction
3.8: Quantifiers
3.9: A Review of Methods of Proof

This page titled 3: Logic is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.1: Propositions and Logical Operators

3.1.1: Propositions

Definition 3.1.1: Proposition

A proposition is a sentence to which one and only one of the terms *true* or *false* can be meaningfully applied.

Example 3.1.1: Some Propositions

"Four is even,", "$4 \in \{1, 3, 5\}$" and "$43 > 21$" are propositions.

In traditional logic, a declarative statement with a definite truth value is considered a proposition. Although our ultimate aim is to discuss mathematical logic, we won't separate ourselves completely from the traditional setting. This is natural because the basic assumptions, or postulates, of mathematical logic are modeled after the logic we use in everyday life. Since compound sentences are frequently used in everyday speech, we expect that logical propositions contain connectives like the word "and." The statement "Europa supports life or Mars supports life" is a proposition and, hence, must have a definite truth value. Whatever that truth value is, it should be the same as the truth value of "Mars supports life or Europa supports life."

3.1.2: Logical Operations

There are several ways in which we commonly combine simple statements into compound ones. The words/phrases *and*, *or*, *not*, *if ... then...*, and *...if and only if ...* can be added to one or more propositions to create a new proposition. To avoid any confusion, we will precisely define each one's meaning and introduce its standard symbol. With the exception of negation (*not*), all of the operations act on pairs of propositions. Since each proposition has two possible truth values, there are four ways that truth can be assigned to two propositions. In defining the effect that a logical operation has on two propositions, the result must be specified for all four cases. The most convenient way of doing this is with a truth table, which we will illustrate by defining the word *and*.

Definition 3.1.2: Logical Conjunction

If p and q are propositions, their conjunction, p and q (denoted $p \wedge q$), is defined by the truth table

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Note 3.1.1

a. To read this truth table, you must realize that any one line represents a case: one possible set of values for p and q.
b. The numbers 0 and 1 are used to denote false and true, respectively. This is consistent with the way that many programming languages treat logical, or Boolean, variables since a single bit, 0 or 1, can represent a truth value.
c. For each case, the symbol under p represents the truth value of p. The same is true for q. The symbol under $p \wedge q$ represents its truth value for that case. For example, the second row of the truth table represents the case in which p is false, q is true, and the resulting truth value for $p \wedge q$ is false. As in everyday speech, $p \wedge q$ is true only when both propositions are true.
d. Just as the letters x, y and z are frequently used in algebra to represent numeric variables, p, q and r seem to be the most commonly used symbols for logical variables. When we say that p is a logical variable, we mean that any proposition can take the place of p.
e. One final comment: The order in which we list the cases in a truth table is standardized in this book. If the truth table involves two simple propositions, the numbers under the simple propositions can be interpreted as the two-digit binary integers in increasing order, 00, 01, 10, and 11, for 0, 1, 2, and 3, respectively.

Definition 3.1.3: Logical Disjunction

If p and q are propositions, their disjunction, p or q (denoted $p \vee q$), is defined by the truth table

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Definition 3.1.4: Logical Negation

If p is a proposition, its negation, not p, denoted $\neg p$, and is defined by the truth table

p	$\neg p$
0	1
1	0

Note 3.1.2

Negation is the only standard operator that acts on a single proposition; hence only two cases are needed.

Consider the following propositions from everyday speech:

a. I'm going to quit if I don't get a raise.
b. If I pass the final, then I'll graduate.
c. I'll be going to the movies provided that my car starts.

All three propositions are conditional, they can all be restated to fit into the form "If *Condition*, then *Conclusion*." For example, the first statement can be rewritten as "If I don't get a raise, then I'm going to quit."

A conditional statement is meant to be interpreted as a guarantee; if the condition is true, then the conclusion is expected to be true. It says no more and no less.

Definition 3.1.5: Conditional Statement

The conditional statement "If p then q," denoted $p \rightarrow q$, is defined by the truth table

Table 3.1.1: Truth Table for $p \rightarrow q$

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Example 3.1.2: Analysis of a Conditional Proposition

Assume your instructor told you "If you receive a grade of 95 or better in the final examination, then you will receive an A in this course." Your instructor has made a promise to you. If you fulfill his condition, you expect the conclusion (getting an A) to be forthcoming. Suppose your graded final has been returned to you. Has your instructor told the truth or is your instructor guilty of a falsehood?

Case I: Your final exam score was less than 95 (the condition is false) and you did not receive an A (the conclusion is false). The instructor told the truth.

Case II: Your final exam score was less than 95, yet you received an A for the course. The instructor told the truth. (Perhaps your overall course average was excellent.)

Case III: Your final exam score was greater than 95, but you did not receive an A. The instructor lied.

Case IV: Your final exam score was greater than 95, and you received an A. The instructor told the truth.

To sum up, the only case in which a conditional proposition is false is when the condition is true and the conclusion is false.

The order of the condition and conclusion in a conditional proposition is important. If the condition and conclusion are exchanged, a different proposition is produced.

Definition 3.1.6: Converse

The converse of the proposition $p \to q$ is the proposition $q \to p$.

The converse of "If you receive a grade of 95 or better in the final exam, then you will receive an A in this course," is "If you receive an A in this course, then you received a grade of 95 or better in the final exam." It should be clear that these two statements say different things.

There *is* a proposition related to $p \to q$ that does have the same logical meaning. This is the contrapositive.

Definition 3.1.7: Contrapositive

The contrapositive of the proposition $p \to q$ is the proposition $\neg q \to \neg p$.

As we will see when we discuss logical proofs, we can prove a conditional proposition by proving its contrapositive, which may be somewhat easier.

Definition 3.1.8: Biconditional Proposition

If p and q are propositions, the biconditional statement "p if and only if q," denoted $p \leftrightarrow q$, is defined by the truth table

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Note that $p \leftrightarrow q$ is true when p and q have the same truth values. It is common to abbreviate "if and only if" to "iff."

Although "if ... then..." and "...if and only if ..." are frequently used in everyday speech, there are several alternate forms that you should be aware of. They are summarized in the following lists.

All of the following are equivalent to "If p then q":

- p implies q.
- q follows from p.
- p, only if q.
- q, if p.
- p is sufficient for q.
- q is necessary for p.

All of the following are equivalent to "p if and only if q":

- p is necessary and sufficient for q.
- p is equivalent to q.

- If p, then q, and if q, then p.
- If p, then q and conversely.

3.1.3: Exercises

Exercise 3.1.1

Let d = "I like discrete structures", c = "I will pass this course" and s = "I will do my assignments." Express each of the following propositions in symbolic form:

a. I like discrete structures and I will pass this course.
b. I will do my assignments or I will not pass this course.
c. It is not true that I both like discrete structures, and will do my assignments.
d. I will not do my assignment and I will not pass this course.

Answer

a. $d \wedge c$
b. $s \vee \neg c$
c. $\neg(d \wedge s)$
d. $\neg s \wedge \neg c$

Exercise 3.1.2

For each of the following propositions, identify simple propositions, express the compound proposition in symbolic form, and determine whether it is true or false:

a. The world is flat or zero is an even integer.
b. If 432,802 is a multiple of 4, then 432,802 is even.
c. 5 is a prime number and 6 is not divisible by 4.
d. $3 \in \mathbb{Z}$ and $3 \in \mathbb{Q}$.
e. $2/3 \in \mathbb{Z}$ and $2/3 \in \mathbb{Q}$.
f. The sum of two even integers is even and the sum of two odd integers is odd.

Exercise 3.1.3

Let p ="$2 \leq 5$", q = "8 is an even integer," and r = "11 is a prime number." Express the following as a statement in English and determine whether the statement is true or false:

a. $\neg p \wedge q$
b. $p \rightarrow q$
c. $(p \wedge q) \rightarrow r$
d. $p \rightarrow (q \vee (\neg r))$
e. $p \rightarrow ((\neg q) \vee (\neg r))$
f. $(\neg q) \rightarrow (\neg p)$

Answer

a. $2 > 5$ and 8 is an even integer. False.
b. If $2 \leqslant 5$ then 8 is an even integer. True.
c. If $2 \leqslant 5$ and 8 is an even integer then 11 is a prime number. True.
d. If $2 \leqslant 5$ then either 8 is an even integer or 11 is not a prime number. True.
e. If $2 \leqslant 5$ then either 8 is an odd integer or 11 is not a prime number. False.
f. If 8 is not an even integer then $2 > 5$. True.

Exercise 3.1.4

Rewrite each of the following statements using the other conditional forms:

a. If an integer is a multiple of 4, then it is even.
b. The fact that a polygon is a square is a sufficient condition that it is a rectangle.
c. If $x = 5$, then $x^2 = 25$.
d. If $x^2 - 5x + 6 = 0$, then $x = 2$ or $x = 3$.
e. $x^2 = y^2$ is a necessary condition for $x = y$.

Exercise 3.1.5

Write the converse of the propositions in Exercise 3.1.4. Compare the truth of each proposition and its converse.

Answer

Only the converse of d is true.

This page titled 3.1: Propositions and Logical Operators is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.2: Truth Tables and Propositions Generated by a Set

3.2.1: Truth Tables

Consider the compound proposition $c = (p \wedge q) \vee (\neg q \wedge r)$, where p, q, and r are propositions. This is an example of a proposition generated by p, q, and r. We will define this terminology later in the section. Since each of the three simple propositions has two possible truth values, it follows that there are eight different combinations of truth values that determine a value for c. These values can be obtained from a truth table for c. To construct the truth table, we build c from p, q, and r and from the logical operators. The result is the truth table below. Strictly speaking, the first three columns and the last column make up the truth table for c. The other columns are work space needed to build up to c.

Table 3.2.1: Truth Table for $c = (p \wedge q) \vee (\neg q \wedge r)$

p	q	r	$p \wedge q$	$\neg q$	$\neg q \wedge r$	$(p \wedge q) \vee (\neg q \wedge r)$
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	1	0	0
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	1	0	0	1

Note that the first three columns of the truth table are an enumeration of the eight three-digit binary integers. This standardizes the order in which the cases are listed. In general, if c is generated by n simple propositions, then the truth table for c will have 2^n rows with the first n columns being an enumeration of the n digit binary integers. In our example, we can see at a glance that for exactly four of the eight cases, c will be true. For example, if p and r are true and q is false (the sixth case), then c is true.

Let S be any set of propositions. We will give two definitions of a proposition generated by S. The first is a bit imprecise, but should be clear. The second definition is called a *recursive definition*. If you find it confusing, use the first definition and return to the second later.

3.2.2: Propositions Generated by a Set

Definition 3.2.1: Proposition Generated by a Set

Let S be any set of propositions. A proposition generated by S is any valid combination of propositions in S with conjunction, disjunction, and negation. Or, to be more precise,

a. If $p \in S$, then p is a proposition generated by S, and
b. If x and y are propositions generated by S, then so are (x), $\neg x$, $x \vee y$, and $x \wedge y$.

Note 3.2.1

We have not included the conditional and biconditional in the definition because they can both be generated from conjunction, disjunction, and negation, as we will see later.

If S is a finite set, then we may use slightly different terminology. For example, if $S = \{p, q, r\}$, we might say that a proposition is generated by $p, q,$ and r instead of from $\{p, q, r\}$.

It is customary to use the following hierarchy for interpreting propositions, with parentheses overriding this order:

- First: Negation

- Second: Conjunction
- Third: Disjunction
- Fourth: The conditional operation
- Fifth: The biconditional operation

Within any level of the hierarchy, work from left to right. Using these rules, $p \wedge q \vee r$ is taken to mean $(p \wedge q) \vee r$. These precedence rules are universal, and are exactly those used by computer languages to interpret logical expressions.

Example 3.2.1: Examples of the Hierarchy of Logical Operations

A few shortened expressions and their fully parenthesized versions:

a. $p \wedge q \wedge r$ is $(p \wedge q) \wedge r$.
b. $\neg p \vee \neg r$ is $(\neg p) \vee (\neg r)$.
c. $\neg \neg p$ is $\neg(\neg p)$.
d. $p \leftrightarrow q \wedge r \rightarrow s$ is $p \leftrightarrow ((q \wedge r) \rightarrow s)$.

A proposition generated by a set S need not include each element of S in its expression. For example, $\neg q \wedge r$ is a proposition generated by $p, q,$ and r.

3.2.3: Exercises

Exercise 3.2.1

Construct the truth tables of:

a. $p \vee p$
b. $p \wedge (\neg p)$
c. $p \vee (\neg p)$
d. $p \wedge p$

Answer

a.

p	$p \vee p$
0	0
1	1

b.

p	$\neg p$	$p \wedge (\neg p)$
0	1	0
1	0	0

c.

p	$\neg p$	$p \vee (\neg p)$
0	1	1
1	0	1

d.

p	$p \wedge p$
0	0
1	1

Exercise 3.2.2

Construct the truth tables of:

a. $\neg(p \wedge q)$
b. $p \wedge (\neg q)$
c. $(p \wedge q) \wedge r$
d. $(p \wedge q) \vee (q \wedge r) \vee (r \wedge p)$
e. $\neg p \vee \neg q$
f. $p \vee q \vee r \vee s$

Exercise 3.2.3

Rewrite the following with as few extraneous parentheses as possible:

a. $(\neg((p) \wedge (r))) \vee (s)$
b. $((p) \vee (q)) \wedge ((r) \vee (q))$

Answer

a. $\neg(p \wedge r) \vee s$
b. $(p \vee q) \wedge (r \vee q)$

Exercise 3.2.4

In what order are the operations in the following propositions performed?

a. $p \vee \neg q \vee r \wedge \neg p$
b. $p \wedge \neg q \wedge r \wedge \neg p$

Exercise 3.2.5

Determine the number of rows in the truth table of a proposition containing four variables $p, q, r,$ and s.

Answer

$2^4 = 16$ rows.

Exercise 3.2.6

If there are 45 lines on a sheet of paper, and you want to reserve one line for each line in a truth table, how large could $|S|$ be if you can write truth tables of propositions generated by S on the sheet of paper?

This page titled 3.2: Truth Tables and Propositions Generated by a Set is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.3: Equivalence and Implication

Consider two propositions generated by p and q: $\neg(p \wedge q)$ and $\neg p \vee \neg q$. At first glance, they are different propositions. In form, they are different, but they have the same meaning. One way to see this is to substitute actual propositions for p and q; such as p: I've been to Toronto; and q: I've been to Chicago.

Then $\neg(p \wedge q)$ translates to "I haven't been to both Toronto and Chicago," while $\neg p \vee \neg q$ is "I haven't been to Toronto or I haven't been to Chicago." Determine the truth values of these propositions. Naturally, they will be true for some people and false for others. What is important is that no matter what truth values they have, $\neg(p \wedge q)$ and $\neg p \vee \neg q$ will have the same truth value. The easiest way to see this is by examining the truth tables of these propositions.

Table 3.3.1: Truth Tables for $\neg(p \wedge q)$ and $\neg p \vee \neg q$

p	q	\(\neg (p\land q)\0	$\neg p \vee \neg q$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

In all four cases, $\neg(p \wedge q)$ and $\neg p \vee \neg q$ have the same truth value. Furthermore, when the biconditional operator is applied to them, the result is a value of true in all cases. A proposition such as this is called a tautology.

3.3.1: Tautologies and Contradictions

Definition 3.3.1: Tautology

An expression involving logical variables that is true in all cases is a tautology. The number 1 is used to symbolize a tautology.

Example 3.3.1: Some Tautologies

All of the following are tautologies because their truth tables consist of a column of 1's.

a. $(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$.
b. $p \vee \neg p$
c. $(p \wedge q) \rightarrow p$
d. $q \rightarrow (p \vee q)$
e. $(p \vee q) \leftrightarrow (q \vee p)$

Definition 3.3.2: Contradiction

An expression involving logical variables that is false for all cases is called a contradiction. The number 0 is used to symbolize a contradiction.

Example 3.3.2: Some Contradictions

$p \wedge \neg p$ and $(p \vee q) \wedge (\neg p) \wedge (\neg q)$ are contradictions.

3.3.2: Equivalence

Definition 3.3.3: Equivalence

Let S be a set of propositions and let r and s be propositions generated by S. r and s are equivalent if and only if $r \leftrightarrow s$ is a tautology. The equivalence of r and s is denoted $r \iff s$.

Equivalence is to logic as equality is to algebra. Just as there are many ways of writing an algebraic expression, the same logical meaning can be expressed in many different ways.

Example 3.3.3: Some Equivalences

The following are all equivalences:

a. $(p \wedge q) \vee (\neg p \wedge q) \iff q$.
b. $p \to q \iff \neg q \to \neg p$
c. $p \vee q \iff q \vee p$.

All tautologies are equivalent to one another.

Example 3.3.4: An Equivalence to 1

$p \vee \neg p \iff 1$.

All contradictions are equivalent to one another.

Example 3.3.5: An Equivalence to 0

$p \wedge \neg p \iff 0$.

3.3.3: Implication

Consider the two propositions:

Table 3.3.2

x: The money is behind Door A; and
y: The money is behind Door A or Door B

Imagine that you were told that there is a large sum of money behind one of two doors marked A and B, and that one of the two propositions x and y is true and the other is false. Which door would you choose? All that you need to realize is that if x is true, then y will also be true. Since we know that this can't be the case, y must be the true proposition and the money is behind Door B.

This is an example of a situation in which the truth of one proposition leads to the truth of another. Certainly, y can be true when x is false; but x can't be true when y is false. In this case, we say that x implies y.

Consider the truth table of $p \to q$, Table 3.1.1. If p implies q, then the third case can be ruled out, since it is the case that makes a conditional proposition false.

Definition 3.3.4: Implication

Let S be a set of propositions and let r and s be propositions generated by S. We say that r implies s if $r \to s$ is a tautology. We write $r \Rightarrow s$ to indicate this implication.

Example 3.3.6: Disjunctive Addition

A commonly used implication called "disjunctive addition" is $p \Rightarrow (p \vee q)$, which is verified by truth table Table 3.3.3.

Table 3.3.3: Truth Table to verify that $p \Rightarrow (p \vee q)$

p	q	$p \vee q$	$p \to o \vee q$
0	0	0	1
0	1	1	1

p	q	$p \vee q$	$p \rightarrow o \vee q$
1	0	1	1
1	1	1	1

If we let p represent "The money is behind Door A" and q represent "The money is behind Door B," $p \Rightarrow (p \vee q)$ is a formalized version of the reasoning used in Example 3.3.6. A common name for this implication is disjunctive addition. In the next section we will consider some of the most commonly used implications and equivalences.

When we defined what we mean by a Proposition Generated by a Set, Definition 3.2.1, we didn't include the conditional and biconditional operators. This was because of the two equivalences $p \rightarrow q \Leftrightarrow \neg p \vee q$ and $p \leftrightarrow q \Leftrightarrow (p \wedge q) \vee (\neg p \wedge \neg q)$. Therefore, any proposition that includes the conditional or biconditional operators can be written in an equivalent way using only conjunction, disjunction, and negation. We could even dispense with disjunction since $p \vee q$ is equivalent to a proposition that uses only conjunction and negation.

3.3.4: Universal Operation

We close this section with a final logical operation, the Sheffer Stroke, that has the interesting property that all other logical operations can be created from it. You can explore this operation in Exercise 3.3.8

Definition 3.3.5: The Sheffer Stroke

The Sheffer Stroke is the logical operator defined by the following truth table:

Table 3.3.4: Truth Table for the Sheffer Stroke

p	q	$p \mid q$
0	0	1
0	1	1
1	0	1
1	1	0

3.3.5: Exercises

Exercise 3.3.1

Given the following propositions generated by p, q, and r, which are equivalent to one another?

a. $(p \wedge r) \vee q$
b. $p \vee (r \vee q)$
c. $r \wedge p$
d. $\neg r \vee p$
e. $(p \vee q) \wedge (r \vee q)$
f. $r \rightarrow p$
g. $r \vee \neg p$
h. $p \rightarrow r$

Answer

$a \Leftrightarrow e, d \Leftrightarrow f, g \Leftrightarrow h$

Exercise 3.3.2

a. Construct the truth table for $x = (p \wedge \neg q) \vee (r \wedge p)$.
b. Give an example other than x itself of a proposition generated by p, q, and r that is equivalent to x.
c. Give an example of a proposition other than x that implies x.
d. Give an example of a proposition other than x that is implied by x.

Exercise 3.3.3

Is an implication equivalent to its converse? Verify your answer using a truth table.

Answer

No. In symbolic form the question is: Is $(p \to q) \Leftrightarrow (q \to p)$?

p	q	$p \to q$	$q \to p$	$(p \to q) \leftrightarrow (q \to p)$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

This table indicates that an implication is not always equivalent to its converse.

Exercise 3.3.4

Suppose that x is a proposition generated by p, q, and r that is equivalent to $p \vee \neg q$. Write out the truth table for x.

Exercise 3.3.5

How large is the largest set of propositions generated by p and q with the property that no two elements are equivalent?

Answer

Let x be any proposition generated by p and q. The truth table for x has 4 rows and there are 2 choices for a truth value for x for each row, so there are $2 \cdot 2 \cdot 2 \cdot 2 = 2^4$ possible propositions.

Exercise 3.3.6

Find a proposition that is equivalent to $p \vee q$ and uses only conjunction and negation.

Exercise 3.3.7

Explain why a contradiction implies any proposition and any proposition implies a tautology.

Answer

$0 \to p$ and $p \to 1$ are tautologies.

Exercise 3.3.8

The significance of the Sheffer Stroke is that it is a "universal" operation in that all other logical operations can be built from it.

a. Prove that $p|q$ is equivalent to $\neg(p \wedge q)$.
b. Prove that $\neg p \Leftrightarrow p|p$.
c. Build $\wedge$ using only the Sheffer Stroke.
d. Build $\vee$ using only the Sheffer Stroke.

This page titled 3.3: Equivalence and Implication is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.4: The Laws of Logic

3.4.1:

In this section, we will list the most basic equivalences and implications of logic. Most of the equivalences listed in Table 3.4.2 should be obvious to the reader. Remember, 0 stands for contradiction, 1 for tautology. Many logical laws are similar to algebraic laws. For example, there is a logical law corresponding to the associative law of addition, $a+(b+c)=(a+b)+c$. In fact, associativity of both conjunction and disjunction are among the laws of logic. Notice that with one exception, the laws are paired in such a way that exchanging the symbols $\wedge$, $\vee$, 1 and 0 for $\vee$, $\wedge$, 0, and 1, respectively, in any law gives you a second law. For example, $p \vee 0 \Leftrightarrow p$ results in $p \wedge 1 \Leftrightarrow p$. This is called a *duality principle*. For now, think of it as a way of remembering two laws for the price of one. We will leave it to the reader to verify a few of these laws with truth tables. However, the reader should be careful in applying duality to the conditional operator and implication since the dual involves taking the converse. For example, the dual of $p \wedge q \Rightarrow p$ is $p \vee q \Leftarrow p$, which is usually written $p \Rightarrow p \vee q$.

Example 3.4.1: Verification of an Identity Law

The Identity Law can be verified with this truth table. The fact that $(p \wedge 1) \leftrightarrow p$ is a tautology serves as a valid proof.

Table 3.4.1: Truth table to demonstrate the identity law for conjunction.

p	1	$p \wedge 1$	$(p \wedge 1) \leftrightarrow p$
0	1	0	1
1	1	1	1

Some of the logical laws in Table 3.4.3 might be less obvious to you. For any that you are not comfortable with, substitute actual propositions for the logical variables. For example, if p is "John owns a pet store" and q is "John likes pets," the detachment law should make sense.

Table 3.4.2: Basic Logical Laws - Equivalences

Commutative Laws	
$p \vee q \Leftrightarrow q \vee p$	$p \wedge q \Leftrightarrow q \wedge p$
Associative Laws	
$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
Distributive Laws	
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
Identity Laws	
$p \vee 0 \Leftrightarrow p$	$p \wedge 1 \Leftrightarrow p$
Negation Laws	
$p \wedge \neg p \Leftrightarrow 0$	$p \vee \neg p \Leftrightarrow 1$
Idempotent Laws	
$p \vee p \Leftrightarrow$	$p \wedge p \Leftrightarrow p$
Null Laws	
$p \wedge 0 \Leftrightarrow 0$	$p \vee 1 \Leftrightarrow 1$
Absorption Laws	
$p \wedge (p \vee q) \Leftrightarrow p$	$p \vee (p \wedge q) \Leftrightarrow p$

Commutative Laws	
DeMorgan's Laws	
$\neg(p \vee q) \Leftrightarrow (\neg p) \wedge (\neg q)$	$\neg(p \wedge q) \Leftrightarrow (\neg p) \vee (\neg q)$
Involution Laws	
$\neg(\neg p) \Leftrightarrow p$	

Table 3.4.3: Basic Logical Laws - Common Implications and Equivalences

Detachment (AKA Modus Ponens)	$(p \to q) \wedge p \Rightarrow q$
Indirect Reasoning (AKA Modus Tollens)	$(p \to q) \wedge \neg q \Rightarrow \neg p$
Disjunctive Addition	$p \Rightarrow (p \vee q)$
Conjunctive Simplification	$(p \wedge q) \Rightarrow p$ and $(p \wedge q) \Rightarrow q$
Disjunctive Simplification	$(p \vee q) \wedge \neg p \Rightarrow q$ and $(p \vee q) \wedge \neg q \Rightarrow p$
Chain Rule	$(p \to q) \wedge (q \to r) \Rightarrow (p \to r)$
Conditional Equivalence	$p \to q \Leftrightarrow \neg p \vee q$
Biconditional Equivalences	$(p \leftrightarrow q) \Leftrightarrow (p \to q) \wedge (q \to p) \Leftrightarrow (p \wedge q) \vee (\neg p \wedge \neg q)$
Contrapositive	$(p \to q) \Leftrightarrow (\neg q \to \neg p)$

3.4.2: Exercises

Exercise 3.4.1

Write the following in symbolic notation and determine whether it is a tautology: “If I study then I will learn. I will not learn. Therefore, I do not study.”

Answer

Let $s =$ I will study, $t =$ I will learn. The argument is: $((s \to t) \wedge (\neg t)) \to (\neg s)$, call the argument a.

s	t	$s \to t$	$(s \to t) \wedge (\neg t)$	a
0	0	1	1	1
0	1	1	0	1
1	0	0	0	1
1	1	1	0	1

.

Since a is a tautology, the argument is valid.

Exercise 3.4.2

Show that the common fallacy $(p \to q) \wedge \neg p \Rightarrow \neg q$ is not a law of logic.

Exercise 3.4.3

Describe, in general, how duality can be applied to implications if we introduce the relation $\Leftarrow$, read “is implied by.” We define this relation by

$$(p \Leftarrow q) \Leftrightarrow (q \Rightarrow p).$$

Answer

In any true statement S, replace; $\wedge$ with $\vee$, $\vee$ with $\wedge$, 0 with 1, 1 with 0, $\Leftarrow$ with $\Rightarrow$, and $\Rightarrow$ with $\Leftarrow$. Leave all other connectives unchanged.

Exercise 3.4.4

Write the dual of the following statements:

a. $(p \wedge q) \Rightarrow p$
b. $(p \vee q) \wedge \neg q \Rightarrow p$

This page titled 3.4: The Laws of Logic is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.5: Mathematical Systems and Proofs

3.5.1: Mathematical Systems

In this section, we present an overview of what a mathematical system is and how logic plays an important role in one. The axiomatic method that we will use here will not be duplicated with as much formality anywhere else in the book, but we hope an emphasis on how mathematical facts are developed and organized will help to unify the concepts we will present. The system of propositions and logical operators we have developed will serve as a model for our discussion. Roughly, a mathematical system can be defined as follows.

Definition 3.5.1: Mathematical System

A mathematical system consists of:

a. A set or universe, U.
b. Definitions: sentences that explain the meaning of concepts that relate to the universe. Any term used in describing the universe itself is said to be undefined. All definitions are given in terms of these undefined concepts of objects.
c. Axioms: assertions about the properties of the universe and rules for creating and justifying more assertions. These rules always include the system of logic that we have developed to this point.
d. Theorems: the additional assertions mentioned above.

Example 3.5.1: Euclidean Geometry

In Euclidean geometry the universe consists of points and lines (two undefined terms). Among the definitions is a definition of parallel lines and among the axioms is the axiom that two distinct parallel lines never meet.

Example 3.5.2: Propositional Calculus

Propositional calculus is a formal name for the logical system that we've been discussing. The universe consists of propositions. The axioms are the truth tables for the logical operators and the key definitions are those of equivalence and implication. We use propositions to describe any other mathematical system; therefore, this is the minimum amount of structure that a mathematical system can have.

Definition 3.5.2: Theorem

A true proposition derived from the axioms of a mathematical system is called a theorem.

Theorems are normally expressed in terms of a finite number of propositions, $p_1, p_2, \ldots, p_n$, called the *premises*, and a proposition,C, called the *conclusion*. These theorems take the form

$$p_1 \wedge p_2 \wedge \cdots \wedge p_n \Rightarrow C$$

or more informally,

$$p_1, p_2, \ldots, \text{ and } p_n \text{ imply } C$$

For a theorem of this type, we say that the premises imply the conclusion. When a theorem is stated, it is assumed that the axioms of the system are true. In addition, any previously proven theorem can be considered an extension of the axioms and can be used in demonstrating that the new theorem is true. When the proof is complete, the new theorem can be used to prove subsequent theorems. A mathematical system can be visualized as an inverted pyramid with the axioms at the base and the theorems expanding out in various directions.

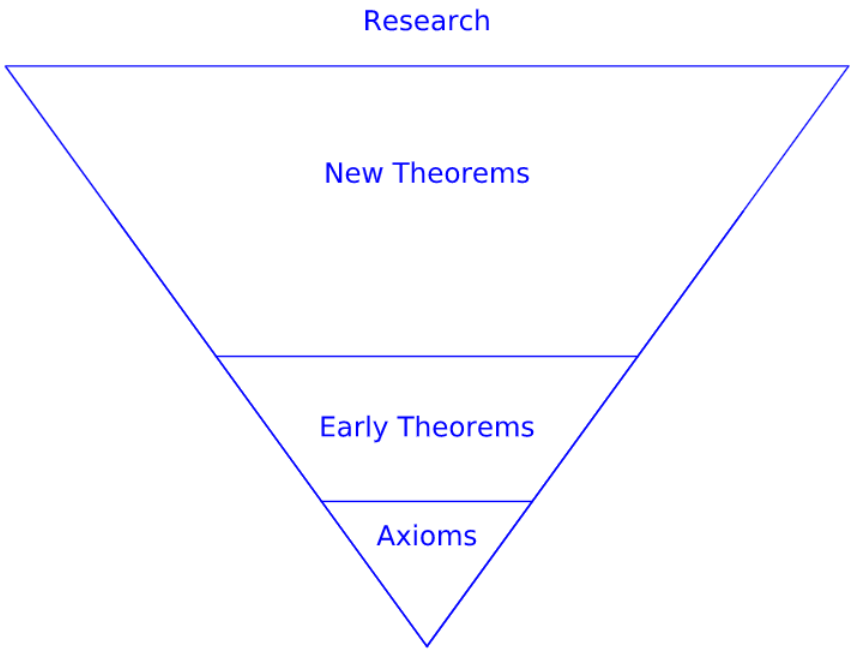


Figure 3.5.1: The body of knowledge in a mathematical system

Definition 3.5.3: Proof

A proof of a theorem is a finite sequence of logically valid steps that demonstrate that the premises of a theorem imply its conclusion.

Exactly what constitutes a proof is not always clear. For example, a research mathematician might require only a few steps to prove a theorem to a colleague, but might take an hour to give an effective proof to a class of students. Therefore, what constitutes a proof often depends on the audience. But the audience is not the only factor. One of the most famous theorems in graph theory, The Four-Color Theorem, Theorem 9.6.1, was proven in 1976, after over a century of effort by many mathematicians. Part of the proof consisted of having a computer check many different graphs for a certain property. Without the aid of the computer, this checking would have taken years. In the eyes of some mathematicians, this proof was considered questionable. Shorter proofs have been developed since 1976 and there is no controversy associated with The Four Color Theorem at this time.

3.5.2: Direct Proof

Theoretically, you can prove anything in propositional calculus with truth tables. In fact, the laws of logic stated in Section 3.4 are all theorems. Propositional calculus is one of the few mathematical systems for which any valid sentence can be determined true or false by mechanical means. A program to write truth tables is not too difficult to write; however, what can be done theoretically is not always practical. For example,

$$a, a \to b, b \to c, \ldots, y \to z \Rightarrow z$$

is a theorem in propositional calculus. However, suppose that you wrote such a program and you had it write the truth table for

$$(a \wedge (a \to b) \wedge (b \to c) \wedge \cdots \wedge (y \to z)) \to z$$

The truth table will have 2^{26} cases. At one million cases per second, it would take approximately one hour to verify the theorem. Now if you decided to check a similar theorem,

$$p_1, p_1 \to p_2, \ldots, p_{99} \to p_{100} \Rightarrow p_{100}$$

you would really have time trouble. There would be $2^{100} \approx 1.26765 \times 10^{30}$ cases to check in the truth table. At one million cases per second it would take approximately 1.46719×10^{19} days to check all cases. For most of the remainder of this section, we will discuss an alternate method for proving theorems in propositional calculus. It is the same method that we will use in a less formal way for proofs in other systems. Formal axiomatic methods would be too unwieldy to actually use in later sections. However, none of the theorems in later chapters would be stated if they couldn't be proven by the axiomatic method.

We will introduce two types of proof here, direct and indirect.

Example 3.5.3: A Typical Direct Proof

This is a theorem: $p \to r, q \to s, p \vee q \Rightarrow s \vee r$. A direct proof of this theorem is:

Table 3.5.1: Direct prrof of $p \to r$, $q \to s$, $p \vee q \Rightarrow s \vee r$

Step	Proposition	Justification
1.	$p \vee q$	Premise
2.	$\neg p \to q$	(1), conditional rule
3.	$q \to s$	Premise
4.	$\neg p \to s$	(2), (3), chain rule
5.	$\neg s \to p$	(4), contrapositive
6.	$p \to r$	Premise
7.	$\neg s \to r$	(5), (6), chain rule
8.	$s \vee r$	(7), conditional rule □

Note that □ marks the end of a proof.

Example 3.5.3 illustrates the usual method of formal proof in a formal mathematical system. The rules governing these proofs are:

1. A proof must end in a finite number of steps.
2. Each step must be either a premise or a proposition that is implied from previous steps using any valid equivalence or implication.
3. For a direct proof, the last step must be the conclusion of the theorem. For an indirect proof (see below), the last step must be contradiction.
4. Justification Column. The column labeled “justification” is analogous to the comments that appear in most good computer programs. They simply make the proof more readable.

Example 3.5.4: Two Proofs of the Same Theorem

Here are two direct proofs of $\neg p \vee q, s \vee p, \neg q \Rightarrow s$:

Table 3.5.2: Direct proof of $\neg p \vee q$, $s \vee p$, $\neg q \Rightarrow s$

1.	$\neg p \vee q$	Premise
2.	$\neg q$	Premise
3.	$\neg p$	Disjunctive simplification, (1), (2)
4.	$s \vee p$	Premise
5.	s	Disjunctive simplification, (3), (4), □

You are invited to justify the steps in this second proof:

Table 3.5.3: Alternate proof of $\neg p \vee q,\ s \vee p,\ \neg q \Rightarrow s$

1.	$\neg p \vee q$
2.	$\neg q \rightarrow \neg p$
3.	$s \vee p$
4.	$p \vee s$
5.	$\neg p \rightarrow s$
6.	$\neg q \rightarrow s$
7.	$\neg q$
8.	s □

The conclusion of a theorem is often a conditional proposition. The condition of the conclusion can be included as a premise in the proof of the theorem. The object of the proof is then to prove the consequence of the conclusion. This rule is justified by the logical law

$$p \rightarrow (h \rightarrow c) \Leftrightarrow (p \wedge h) \rightarrow c$$

Example 3.5.5: Example of a Proof with a Conditional Conclusion

The following proof of $p \rightarrow (q \rightarrow s), \neg r \vee p, q \Rightarrow r \rightarrow s$ includes r as a fourth premise. Inference of truth of s completes the proof.

Table 3.5.4: Proof of a theorem with a conditional conclusion.

1.	$\neg r \vee p$	Premise
2.	r	Added premise
3.	p	(1), (2), disjunction simplification
4.	$p \rightarrow (q \rightarrow s)$	Premise
5.	$q \rightarrow s$	(3), (4), detachment
6.	q	Premise
7.	s	(5), (6), detachment. □

3.5.3: Indirect Proof

Consider a theorem $P \Rightarrow C$, where P represents $p_1, p_2, \ldots,$ and p_n, the premises. The method of ***indirect proof*** is based on the equivalence $P \rightarrow C \Leftrightarrow \neg(P \wedge \neg C)$. In words, this logical law states that if $P \Rightarrow C$, then $P \wedge \neg C$ is always false; that is, $P \wedge \neg C$ is a contradiction. This means that a valid method of proof is to negate the conclusion of a theorem and add this negation to the premises. If a contradiction can be implied from this set of propositions, the proof is complete. For the proofs in this section, a contradiction will often take the form $t \wedge \neg t$.

For proofs involving numbers, a contradiction might be $1 = 0$ or $0 < 0$. Indirect proofs involving sets might conclude with $x \in \emptyset$ or ($x \in A$ and $x \in A^c$). Indirect proofs are often more convenient than direct proofs in certain situations. Indirect proofs are often called *proofs by contradiction*.

Example 3.5.6: An Indirect Proof

Here is an example of an indirect proof of the theorem in Example 3.5.3.

Table 3.5.5: An Indirect proof of $p \rightarrow r,\ q \rightarrow s,\ p \vee q \Rightarrow s \vee r$

1.	$\neg(s \vee r)$	Negated conclusion
2.	$\neg s \wedge \neg r$	DeMorgan's Law, (1)
3.	$\neg s$	Conjunctive simplification, (2)
4.	$q \rightarrow s$	Premise
5.	$\neg q$	Indirect reasoning, (3), (4)
6.	$\neg r$	Conjunctive simplification, (2)
7.	$p \rightarrow r$	Premise
8.	$\neg p$	Indirect reasoning, (6), (7)
9.	$(\neg p) \wedge (\neg q)$	Conjunctive, (5), (8)
10.	$\neg(p \vee q)$	DeMorgan's Law, (9)
11.	$p \vee q$	Premise
12.	0	(10), (11) □

Note 3.5.1: Proof Style

The rules allow you to list the premises of a theorem immediately; however, a proof is much easier to follow if the premises are only listed when they are needed.

Example 3.5.7: Yet Another Indirect Proof

Here is an indirect proof of $a \rightarrow b, \neg(b \vee c) \Rightarrow \neg a$.

Table 3.5.6: Indirect proof of $a \rightarrow b, \neg(b \vee c) \Rightarrow \neg a$

1.	a	Negation of the conclusion
2.	$a \rightarrow b$	Premise
3.	b	(1), (2), detachment
4.	$b \vee c$	(3), disjunctive addition
5.	$\neg(b \vee c)$	Premise
6.	0	(4), (5) □

As we mentioned at the outset of this section, we are only presenting an overview of what a mathematical system is. For greater detail on axiomatic theories, see Stoll (1961). An excellent description of how propositional calculus plays a part in artificial intelligence is contained in Hofstadter (1980). If you enjoy the challenge of constructing proofs in propositional calculus, you should enjoy the game WFF'N PROOF (1962), by L.E. Allen.

3.5.4: Exercises

Exercise 3.5.1

Prove with truth tables:

a. $p \vee q, \neg q \Rightarrow p$
b. $p \rightarrow q, \neg q \Rightarrow \neg p$

Answer

a.

p	q	$(p \vee q) \wedge \neg q$	$((p \vee q) \wedge \neg q) \rightarrow p$
0	0	0	1
0	1	0	1
1	0	1	1
1	1	0	1

b.

p	q	$(p \rightarrow q) \wedge \neg q$	$\neg p$	$(p \rightarrow q) \wedge (\neg q)$
0	0	1	1	1
0	1	0	1	1
1	0	0	0	1
1	1	0	0	1

Exercise 3.5.2

Prove with truth tables:

a. $q, \neg q \Rightarrow p$
b. $p \rightarrow q \Rightarrow \neg p \vee q$

Exercise 3.5.3

Give direct and indirect proofs of:

a. $a \rightarrow b, c \rightarrow b, d \rightarrow (a \vee c), d \Rightarrow b$.
b. $(p \rightarrow q) \wedge (r \rightarrow s), (q \rightarrow t) \wedge (s \rightarrow u), \neg(t \wedge u), p \rightarrow r \Rightarrow \neg p$.
c. $p \rightarrow (q \rightarrow r), \neg s \vee p, q \Rightarrow s \rightarrow r$.
d. $p \rightarrow q, q \rightarrow r, \neg(p \wedge r), p \vee r \Rightarrow r$.
e. $\neg q, p \rightarrow q, p \vee t \Rightarrow t$

Answer

a. Direct proof:

1. $\neg b$ Negated conclusion
2. $a \rightarrow b$ Premise
3. $\neg a$ Indirect Reasoning (1), (2)
4. $c \rightarrow b$ Premise
5. $\neg c$ Indirect Reasoning (1), (4)
6. $(\neg a \wedge \neg c)$ Conjunctive (3), (5)
7. $\neg(a \vee c)$ DeMorgan's law (6)
8. $d \rightarrow (a \vee c)$ Premise
9. $\neg d$ Indirect Reasoning (7), (8)
10. d Premise
11. 0 (9), (10) □

a. Indirect proof:

1. $d \rightarrow (a \vee c)$
2. d
3. $a \vee c$
4. $a \rightarrow b$
5. $\neg a \vee b$
6. $c \rightarrow b$
7. $\neg c \vee b$
8. $(\neg a \vee b) \wedge (\neg c \vee b)$
9. $(\neg a \wedge \neg c) \vee b$
10. $\neg(a \vee c) \vee b$

11. b □

b. Direct proof:

1. $(p \to q) \wedge (r \to s)$
2. $p \to q$
3. $(p \to t) \wedge (s \to u)$
4. $q \to t$
5. $p \to t$
6. $r \to s$
7. $s \to u$
8. $r \to u$
9. $p \to r$
10. $p \to u$
11. $p \to (t \wedge u)$ Use $(x \to y) \wedge (x \to z) \Leftrightarrow x \to (y \wedge z)$
12. $\neg(t \wedge u) \to \neg p$
13. $\neg(t \wedge u)$
14. $\neg p$ □

b. Indirect proof:

1. p
2. $p \to q$
3. q
4. $q \to t$
5. t
6. $\neg(t \wedge u)$
7. $\neg t \vee \neg u$
8. $\neg u$
9. $s \to u$
10. $\neg s$
11. $r \to s$
12. $\neg r$
13. $p \to r$
14. r
15. 0 □

c. Direct proof:

1. $\neg s \vee p$ Premise
2. s Added premise (conditional conclusion)
3. $\neg(\neg s)$ Involution (2)
4. p Disjunctive simplification (1), (3)
5. $p \to (q \to r)$ Premise
6. $q \to r$ Detachment (4), (5)
7. q Premise
8. r Detachment (6), (7) □

c. Indirect proof:

1. $\neg(s \to r)$ Negated conclusion
2. $\neg(\neg s \vee r)$ Conditional equivalence (I)
3. $s \wedge \neg r$ DeMorgan (2)
4. s Conjunctive simplification (3)
5. $\neg s \vee p$ Premise
6. $s \to p$ Conditional equivalence (5)
7. p Detachment (4), (6)
8. $p \to (q \to r)$ Premise

9. $q \to r$ Detachment (7), (8)
10. q Premise
11. r Detachment (9), (10)
12. $\neg r$ Conjunctive simplification (3)
13. 0 Conjunction (11), (12) □

d. Direct proof:

1. $p \to q$
2. $q \to r$
3. $p \to r$
4. $p \vee r$
5. $\neg p \vee r$
6. $(p \vee r) \wedge (\neg p \vee r)$
7. $(p \wedge \neg p) \vee r$
8. $0 \vee r$
9. r□

d. Indirect proof:

1. $\neg rv$ Negated conclusion
2. $p \vee r$ Premise
3. p (1), (2)
4. $p \to q$ Premise
5. q Detachment (3), (4)
6. $q \to r$ Premise
7. r Detachment (5), (6)
8. 0 (1), (7) □

Exercise 3.5.4

Give direct and indirect proofs of:

a. $p \to q, \neg r \to \neg q, \neg r \Rightarrow \neg p$.
b. $p \to \neg q, \neg r \to q, p \Rightarrow r$.
c. $a \vee b, c \wedge d, a \to \neg c \Rightarrow b$.

Exercise 3.5.5

Are the following arguments valid? If they are valid, construct formal proofs; if they aren't valid, explain why not.

a. If wages increase, then there will be inflation. The cost of living will not increase if there is no inflation. Wages will increase. Therefore, the cost of living will increase.
b. If the races are fixed or the casinos are crooked, then the tourist trade will decline. If the tourist trade decreases, then the police will be happy. The police force is never happy. Therefore, the races are not fixed.

Answer

1. Let W stand for "Wages will increase," I stand for "there will be inflation," and C stand for "cost of living will increase." Therefore the argument is: $W \to I,\ \neg I \to \neg C,\ W \Rightarrow C$. The argument is invalid. The easiest way to see this is through a truth table, which has one case, the seventh, that this false. Let x be the conjunction of all premises.

W	I	C	$\neg I$	$\neg C$	$W \to I$	$\neg I \to \neg C$	x	$x \to C$
0	0	0	1	1	1	0	0	1
0	0	1	1	0	1	1	0	1
0	1	0	0	1	1	1	0	1
0	1	1	0	0	1	1	0	1
1	0	0	1	1	0	0	0	1
1	0	1	1	0	0	1	0	1
1	1	0	0	1	1	1	1	0
1	1	1	0	0	1	1	1	1

2. Let r stand for "the races are fixed," c stand for "casinos are crooked," t stand for "the tourist trade will decline," and p stand for "the police will be happy." Therefore, the argument is:

$$(r \vee c) \to t, t \to p, \neg p \to \neg r.$$

The argument is valid. Proof:

1. $t \to p$ Premise
2. $\neg p$ Premise
3. $\neg t$ Indirect Reasoning (1), (2)
4. $(r \vee c) \to t$ Premise
5. $\neg(r \vee c)$ Indirect Reasoning (3), (4)
6. $(\neg r) \wedge (\neg c)$ DeMorgan (5)
7. $\neg r$ Conjunction simplification (6) □

Exercise 3.5.6

Determine the validity of the following argument: For students to do well in a discrete mathematics course, it is necessary that they study hard. Students who do well in courses do not skip classes. Students who study hard do well in courses. Therefore students who do well in a discrete mathematics course do not skip class.

Exercise 3.5.7

Describe how $p_1, p_1 \to p_2, \ldots, p_{99} \to p_{100} \Rightarrow p_{100}$ could be proved in 199 steps.

Answer

$p_1 \to p_k$ and $p_k \to p_{k+1}$ implies $p_1 \to p_{k+1}$. It takes two steps to get to $p_1 \to p_{k+1}$ from $p_1 \to p_k$ This means it takes $2(100-1)$ steps to get to $p_1 \to p_{100}$ (subtract 1 because $p_1 \to p_2$ is stated as a premise). A final step is needed to apply detachment to imply p_{100}

This page titled 3.5: Mathematical Systems and Proofs is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.6: Propositions over a Universe

3.6.1: Propositions over a Universe

Consider the sentence “He was a member of the Boston Red Sox.” There is no way that we can assign a truth value to this sentence unless “he” is specified. For that reason, we would not consider it a proposition. However, “he” can be considered a variable that holds a place for any name. We might want to restrict the value of “he” to all names in the major-league baseball record books. If that is the case, we say that the sentence is a proposition over the set of major-league baseball players, past and present.

Definition 3.6.1: Proposition over a Universe

Let U be a nonempty set. A proposition over U is a sentence that contains a variable that can take on any value in U and that has a definite truth value as a result of any such substitution.

Example 3.6.1: Some Propositions over a Variety of Universes

a. A few propositions over the integers are $4x^2 - 3x = 0,\ 0 \le n \le 5,$ and “k is a multiple of 3.”
b. A few propositions over the rational numbers are $4x^2 - 3x = 0,\ y^2 = 2,$ and $(s-1)(s+1) = s^2 - 1.$
c. A few propositions over the subsets of $\mathbb{P}$ are $(A = \emptyset) \vee (A = \mathbb{P}),\ \ 3 \in A,$ and $A \cap \{1, 2, 3\} \neq \emptyset.$

All of the laws of logic that we listed in Section 3.4 are valid for propositions over a universe. For example, if p and q are propositions over the integers, we can be certain that $p \wedge q \Rightarrow p,$ because $(p \wedge q) \to p$ is a tautology and is true no matter what values the variables in p and q are given. If we specify p and q to be $p(n) : n < 4$ and $q(n) : n < 8,$ we can also say that p implies $p \wedge q.$ This is not a usual implication, but for the propositions under discussion, it is true. One way of describing this situation in general is with truth sets.

3.6.2: Truth Sets

Definition 3.6.2: Truth Set

If p is a proposition over U, the truth set of p is $T_p = \{a \in U \mid p(a) \text{ is true}\}.$

Example 3.6.2: Truth Set Example

The truth set of the proposition $\{1, 2\} \cap A = \emptyset,$ taken as a proposition over the power set of $\{1, 2, 3, 4\}$ is $\{\emptyset, \{3\}, \{4\}, \{3, 4\}\}.$

Example 3.6.3: Truth Sets Depend on the Universe

Over the universe $\mathbb{Z}$ (the integers), the truth set of $4x^2 - 3x = 0$ is $\{0\}.$ If the universe is expanded to the rational numbers, the truth set becomes $\{0, 3/4\}.$ The term *solution set* is often used for the truth set of an equation such as the one in this example.

Definition 3.6.3: Tautologies and Contradictions over a Universe

A proposition over U is a tautology if its truth set is $U.$ It is a contradiction if its truth set is empty.

Example 3.6.4: Tautology, Contradiction over $\mathbb{Q}$

$(s-1)(s+1) = s^2 - 1$ is a tautology over the rational numbers. $x^2 - 2 = 0$ is a contradiction over the rationals.

The truth sets of compound propositions can be expressed in terms of the truth sets of simple propositions. For example, if $a \in T_{p \wedge q}$ if and only if a makes $p \wedge q$ true. This is true if and only if a makes both p and q true, which, in turn, is true if and only if $a \in T_p \cap T_q.$ This explains why the truth set of the conjunction of two propositions equals the intersection of the truth sets of the two propositions. The following list summarizes the connection between compound and simple truth sets

Table 3.6.1: Truth Sets of Compound Statements

$T_{p\wedge q} = T_p \cap T_q$
$T_{p\vee q} = T_p \cup T_q$
$T_{\neg p} = {T_p}^c$
$T_{p\leftrightarrow q} = (T_p \cap T_q) \cup ({T_p}^c \cap {T_q}^c)$
$T_{p\to q} = {T_p}^c \cup T_q$

Definition 3.6.4: Equivalence of Propositions over a Universe

Two propositions, p and q, are equivalent if $p \leftrightarrow q$ is a tautology. In terms of truth sets, this means that p and q are equivalent if $T_p = T_q$.

Example 3.6.5: Some Pairs of Equivalent Propositions

a. $n+4=9$ and $n=5$ are equivalent propositions over the integers.
b. $A \cap \{4\} \neq \emptyset$ and $4 \in A$ are equivalent propositions over the power set of the natural numbers.

Definition 3.6.5: Implication for Propositions over a Universe

If p and q are propositions over U, p implies q if $p \to q$ is a tautology.

Since the truth set of $p \to q$ is ${T_p}^c \cup T_q$, the Venn diagram for $T_{p\to q}$ in Figure 3.6.1 shows that $p \Rightarrow q$ when $T_p \subseteq T_q$.

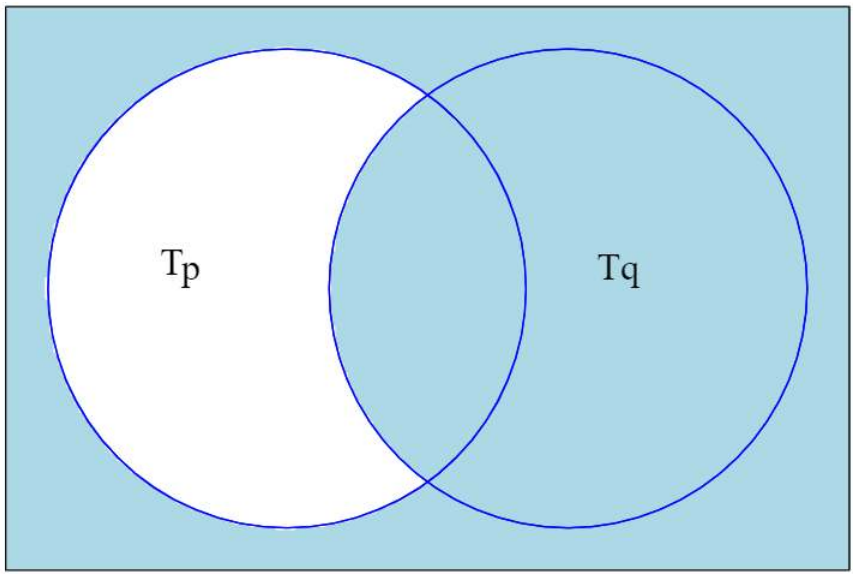


Figure 3.6.1: Venn Diagram for $T_{p\to q}$

Example 3.6.6: Examples of Implications

a. Over the natural numbers: $n \leq 4 \Rightarrow n \leq 8$ since $\{0,1,2,3,4\} \subseteq \{0,1,2,3,4,5,6,7,8\}$
b. Over the power set of the integers: $|A^c| = 1$ implies $A \cap \{0,1\} \neq \emptyset$
c. Over the power set of the integers, $A \subseteq$ even integers $\Rightarrow A \cap$ odd integers $= \emptyset$

3.6.3: Exercises

Exercise 3.6.1

If $U = \mathcal{P}(\{1,2,3,4\})$, what are the truth sets of the following propositions?

a. $A \cap \{2,4\} = \emptyset$.
b. $3 \in A$ and $1 \notin A$.
c. $A \cup \{1\} = A$.
d. A is a proper subset of $\{2,3,4\}$.
e. $|A| = |A^c|$.

Answer

a. $\{\{1\},\{3\},\{1,3\},\emptyset\}$
b. $\{\{3\},\{3,4\},\{3,2\},\{2,3,4\}\}$
c. $\{\{1\},\{1,2\},\{1,3\},\{1,4\},\{1,2,3\},\{1,2,4\},\{1,3,4\},\{1,2,3,4\}\}$
d. $\{\{2\},\{3\},\{4\},\{2,3\},\{2,4\},\{3,4\}\}$
e. $\{A \subseteq U : |A| = 2\}$

Exercise 3.6.2

Over the universe of positive integers, define

Table 3.6.2

$p(n)$:	n is prime and $n < 32$.
$q(n)$:	n is a power of 3.
$r(n)$:	n is a divisor of 27.

1. What are the truth sets of these propositions?
2. Which of the three propositions implies one of the others?

Answer

a.
i. $T_p = \{2,3,5,7,11,13,17,19,23,29,31\}$
ii. $T_q = \{1,3,9,27,81,\ldots\}$
iii. $T_r = \{1,3,9,27\}$

b. $r \Rightarrow q$

Exercise 3.6.3

If $U = \{0,1,2\}$, how many propositions over U could you list without listing two that are equivalent?

Answer

There are $2^3 = 8$ subsets of U, allowing for the possibility of 2^8 nonequivalent propositions over U.

Exercise 3.6.4

Given the propositions over the natural numbers:

Table 3.6.3

$p : n < 4$,	$q : 2n > 17$, and	$r : n$ is a divisor of 18

What are the truth sets of:

a. q
b. $p \wedge q$
c. r
d. $q \rightarrow r$

Exercise 3.6.5

Suppose that s is a proposition over $\{1, 2, \ldots, 8\}$.If $T_s = \{1, 3, 5, 7\}$, give two examples of propositions that are equivalent to s.

Answer

Two possible answers: s is odd and $(s-1)(s-3)(s-5)(s-7) = 0$

Exercise 3.6.6

a. Determine the truth sets of the following propositions over the positive integers:

$$p(n) : n \text{ is a perfect square and } n < 100$$

$$q(n) : n = |\mathcal{P}(A)| \text{ for some set } A.$$

b. Determine $T_{p \wedge q}$ for p and q above.

Exercise 3.6.7

Let the universe be $\mathbb{Z}$, the set of integers. Which of the following propositions are equivalent over $\mathbb{Z}$?

Table 3.6.4

a:	$0 < n^2 < 9$
b:	$0 < n^3 < 27$
c:	$0 < n < 3$

Answer

b and c

This page titled 3.6: Propositions over a Universe is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.7: Mathematical Induction

3.7.1: Introduction, First Example

In this section, we will examine mathematical induction, a technique for proving propositions over the positive integers. Mathematical induction reduces the proof that all of the positive integers belong to a truth set to a finite number of steps.

Example 3.7.1: Formula for Triangular Numbers

Consider the following proposition over the positive integers, which we will label $p(n)$: The sum of the positive integers from 1 to n is $\frac{n(n+1)}{2}$. This is a well-known formula that is quite simple to verify for a given value of n. For example, $p(5)$ is: The sum of the positive integers from 1 to 5 is $\frac{5(5+1)}{2}$. Indeed, $1+2+3+4+5=15=\frac{5(5+1)}{2}$. However, this doesn't serve as a proof that $p(n)$ is a tautology. All that we've established is that 5 is in the truth set of p. Since the positive integers are infinite, we certainly can't use this approach to prove the formula.

An Analogy: A proof by mathematical induction is similar to knocking over a row of closely spaced dominos that are standing on end. To knock over the dominos in Figure 3.7.1, all you need to do is push the first domino over. To be assured that they all will be knocked over, some work must be done ahead of time. The dominos must be positioned so that if any domino is pushed is knocked over, it will push the next domino in the line.



Figure 3.7.1: An analogy for Mathematical Induction, Creative Commons photo by Ranveig Thattai

Returning to Example 3.7.1 imagine the propositions $p(1), p(2), p(3), \ldots$ to be an infinite line of dominos. Let's see if these propositions are in the same formation as the dominos were. First, we will focus on one specific point of the line: $p(99)$ and $p(100)$. We are not going to prove that either of these propositions is true, just that the truth of $p(99)$ implies the truth of $p(100)$. In terms of our analogy, if $p(99)$ is knocked over, it will knock over $p(100)$.

In proving $p(99) \Rightarrow p(100)$, we will use $p(99)$ as our premise. We must prove: The sum of the positive integers from 1 to 100 is $\frac{100(100+1)}{2}$. We start by observing that the sum of the positive integers from 1 to 100 is $(1+2+\cdots+99)+100$. That is, the sum of the positive integers from 1 to 100 equals the sum of the first ninety-nine plus the final number, 100. We can now apply our premise, $p(99)$, to the sum $1+2+\cdots+99$. After rearranging our numbers, we obtain the desired expression for $1+2+\cdots+100$:

$$\begin{aligned} 1+2+\cdots+99+100 &= (1+2+\cdots+99)+100 \\ &= \frac{99\cdot(99+1)}{2}+100 \text{ by our assumption of } p(99) \\ &= \frac{99\cdot 100}{2}+\frac{2\cdot 100}{2} \\ &= \frac{100\cdot 101}{2} \\ &= \frac{100\cdot(100+1)}{2} \end{aligned}$$

What we've just done is analogous to checking two dominos in a line and finding that they are properly positioned. Since we are dealing with an infinite line, we must check all pairs at once. This is accomplished by proving that $p(n) \Rightarrow p(n+1)$ for all $n \geq 1$:

$$\begin{aligned} 1+2+\cdots+n+(n+1) &= (1+2+\cdots+n)+(n+1) \\ &= \frac{n(n+1)}{2}+(n+1) \text{ by } p(n) \\ &= \frac{n(n+1)}{2}+\frac{2(n+1)}{2} \\ &= \frac{(n+1)(n+2)}{2} \\ &= \frac{(n+1)((n+1)+1)}{2} \end{aligned}$$

They are all lined up! Now look at $p(1)$: The sum of the positive integers from 1 to 1 is $\frac{1\cdot(1+1)}{2}$. Clearly, $p(1)$ is true. This sets off a chain reaction. Since $p(1) \Rightarrow p(2)$, $p(2)$ is true. Since $p(2) \Rightarrow p(3)$, $p(3)$ is true; and so on. ■

Theorem 3.7.1: The Principle of Mathematical Induction

Let $p(n)$ be a proposition over the positive integers. If

1. $p(1)$ is true, and
2. for all $n \geq 1$, $p(n) \Rightarrow p(n+1)$,

then $p(n)$ is a tautology.

Note 3.7.1

The truth of $p(1)$ is called the *basis* for the induction proof. The premise that $p(n)$ is true in the second part is called the *induction hypothesis*. The proof that $p(n)$ implies $p(n+1)$ is called the *induction step* of the proof. Despite our analogy, the basis is usually done first in an induction proof. However, order doesn't really matter.

3.7.2: Examples

Example 3.7.2: Generalized Detachment

Consider the implication over the positive integers.

$$p(n): q_0 \rightarrow q_1, q_1 \rightarrow q_2, \ldots, q_{n-1} \rightarrow q_n, q_0 \Rightarrow q_n$$

A proof that $p(n)$ is a tautology follows. Basis: $p(1)$ is $q_0 \rightarrow q_1, q_0 \Rightarrow q_1$. This is the logical law of detachment which we know is true. If you haven't done so yet, write out the truth table of $((q_0 \rightarrow q_1) \wedge q_0) \rightarrow q_1$ to verify this step.

Induction: Assume that $p(n)$ is true for some $n \geq 1$. We want to prove that $p(n+1)$ must be true. That is:

$$q_0 \rightarrow q_1, q_1 \rightarrow q_2, \ldots, q_{n-1} \rightarrow q_n, q_n \rightarrow q_{n+1}, q_0 \Rightarrow q_{n+1}$$

Here is a direct proof of $p(n+1)$:

Table 3.7.1

Step	Proposition	Justification
$\mathbf{1-(n+1)}$	$q_0 \rightarrow q_1, q_1 \rightarrow q_2, \ldots, q_{n-1} \rightarrow q_n, q_0$	Premises
$\boldsymbol{n+2}$	q_n	$(1)-(n+1)$, $p(n)$
$\boldsymbol{n+3}$	$q_n \rightarrow q_{n+1}$	Premise
$\boldsymbol{n+4}$	q_{n+1}	$(n+2)$, $(n+3)$, detachment □

Example 3.7.3: An Example from Number Theory

For all $n \geq 1$, n^3+2n is a multiple of 3. An inductive proof follows:

Basis: $1^3+2(1)=3$ is a multiple of 3. The basis is almost always this easy!

Induction: Assume that $n \geq 1$ and n^3+2n is a multiple of 3. Consider $(n+1)^3+2(n+1)$. Is it a multiple of 3?

$$\begin{aligned}(n+1)^3+2(n+1) &= n^3+3n^2+3n+1+(2n+2)\\ &= n^3+2n+3n^2+3n+3\\ &= (n^3+2n)+3(n^2+n+1)\end{aligned} \quad .$$

Yes, $(n+1)^3+2(n+1)$ is the sum of two multiples of 3; therefore, it is also a multiple of 3. □

Now we will discuss some of the variations of the principle of mathematical induction. The first simply allows for universes that are similar to $\mathbb{P}$ such as $\{-2,-1,0,1,\ldots\}$or $\{5,6,7,8,\ldots\}$.

Theorem 3.7.2: Principle of Mathematical Induction (Generalized)

If $p(n)$ is a proposition over $\{k_0, k_0+1, k_0+2, \ldots\}$, where k_0 is any integer, then $p(n)$ is a tautology if

1. $p(k_0)$ is true, and
2. for all $n \geq k_0$, $p(n) \Rightarrow p(n+1)$.

Example 3.7.4: A Proof of the Permutations Formula

In Chapter 2, we stated that the number of different permutations of k elements taken from an n element set, $P(n;k)$, can be computed with the formula $\frac{n!}{(n-k)!}$. We can prove this statement by induction on n. For $n \geq 0$, let $q(n)$ be the proposition

$$P(n;k) = \frac{n!}{(n-k)!} \text{ for all } k,\ 0 \leq k \leq n.$$

Basis: $q(0)$ states that $P(0;0)$ if is the number of ways that 0 elements can be selected from the empty set and arranged in order, then $P(0;0) = \frac{0!}{0!} = 1$. This is true. A general law in combinatorics is that there is exactly one way of doing nothing.

Induction: Assume that $q(n)$ is true for some natural number n. It is left for us to prove that this assumption implies that $q(n+1)$ is true. Suppose that we have a set of cardinality $n+1$ and want to select and arrange k of its elements. There are two cases to consider, the first of which is easy. If $k=0$, then there is one way of selecting zero elements from the set; hence

$$P(n+1;0) = 1 = \frac{(n+1)!}{(n+1+0)!}$$

and the formula works in this case.

The more challenging case is to verify the formula when k is positive and less than or equal to $n+1$. Here we count the value of $P(n+1;k)$ by counting the number of ways that the first element in the arrangement can be filled and then counting the number of ways that the remaining $k-1$ elements can be filled in using the induction hypothesis.

There are $n+1$ possible choices for the first element. Since that leaves n elements to fill in the remaining $k-1$ positions, there are $P(n;k-1)$ ways of completing the arrangement. By the rule of products,

$$\begin{aligned} P(n+1;k) &= (n+1)P(n;k-1) \\ &= (n+1)\frac{n!}{(n-(k-1))!} \\ &= \frac{(n+1)n!}{(n-k+1)!} \\ &= \frac{(n+1)!}{((n+1)-k)!} \end{aligned}$$

■

3.7.3: Course of Values Induction

A second variation allows for the expansion of the induction hypothesis. The course-of-values principle includes the previous generalization. It is also sometimes called *strong induction*.

Theorem 3.7.3: The Course-of-Values Principle of Mathematical Induction

If $p(n)$ is a proposition over $\{k_0, k_0+1, k_0+2, \ldots\}$, where k_0 is any integer, then $p(n)$ is a tautology if

1. $p(k_0)$ is true, and
2. for all $n \geq k_0$, $p(k_0), p(k_0+1), \ldots, p(n) \Rightarrow p(n+1)$.

A ***prime number*** is defined as a positive integer that has exactly two positive divisors, 1 and itself. There are an infinite number of primes. The list of primes starts with $2, 3, 5, 7, 11, \ldots$.

The proposition over $\{2, 3, 4, \ldots\}$that we will prove here is $p(n)$: n can be written as the product of one or more primes. In most texts, the assertion that $p(n)$ is a tautology would appear as

Theorem 3.7.4: Existence of Prime Factorizations

Every positive integer greater than or equal to 2 has a prime decomposition.

Proof

If you were to encounter this theorem outside the context of a discussion of mathematical induction, it might not be obvious that the proof can be done by induction. Recognizing when an induction proof is appropriate is mostly a matter of experience. Now on to the proof!

Basis: Since 2 is a prime, it is already decomposed into primes (one of them).

Induction: Suppose that for some $n \geq 2$ all of the integers $2, 3, \ldots, n$ have a prime decomposition. Notice the course-of-value hypothesis. Consider $n+1$. Either $n+1$ is prime or it isn't. If $n+1$ is prime, it is already decomposed into primes. If not, then $n+1$ has a divisor, d, other than 1 and $n+1$. Hence, $n+1 = cd$ where both c and d are between 2 and n. By the induction hypothesis, c and d have prime decompositions, $c_1 c_2 \cdots c_s$ and $d_1 d_2 \cdots d_t$, respectively. Therefore, $n+1$ has the prime decomposition $c_1 c_2 \cdots c_s d_1 d_2 \cdots d_t$.

Note 3.7.2: Peano Postulates and Induction

Mathematical induction originated in the late nineteenth century. Two mathematicians who were prominent in its development were Richard Dedekind and Giuseppe Peano. Dedekind developed a set of axioms that describe the positive integers. Peano refined these axioms and gave a logical interpretation to them. The axioms are usually called the Peano Postulates.

Axiom 3.7.1: Peano Postulates

The system of positive integers consists of a nonempty set, $\mathbb{P}$; a least element of $\mathbb{P}$, denoted 1; and a "successor function," s, with the properties

1. If $k \in \mathbb{P}$, then there is an element of $\mathbb{P}$ called the successor of k, denoted $s(k)$.
2. No two elements of $\mathbb{P}$ have the same successor.
3. No element of $\mathbb{P}$ has 1 as its successor.
4. If $S \subseteq \mathbb{P}$, $1 \in S$, and $k \in S \Rightarrow s(k) \in S$, then $S = \mathbb{P}$.

Note 3.7.3

- You might recognize $s(k)$ as simply being $k+1$.
- Axiom 4 is the one that makes mathematical induction possible. In an induction proof, we simply apply that axiom to the truth set of a proposition.

3.7.4: Exercises

Exercise 3.7.1

Prove that the sum of the first n odd integers equals n^2.

Answer

We wish to prove that $P(n): 1+3+5+\cdots+(2n-1)=n^2$ is true for $n \geqslant 1$. Recall that the nth odd positive integer is $2n-1$.

Basis: for $n=1$, $P(n)$ is $1=1^2$, which is true

Induction: Assume that for some $n \geqslant 1$, $P(n)$ is true. Then we infer that $P(n+1)$ follows:

$$\begin{aligned} 1+3+\cdots+(2(n+1)-1) &= (1+3+\cdots+(2n-1))+(2(n+1)-1) \\ &= n^2+(2n+1) \quad \text{by } P(n) \text{ and basic algebra} \\ &= (n+1)^2 \quad \blacksquare \end{aligned}$$

Exercise 3.7.2

Prove that if $n \geq 1$, then $1(1!)+2(2!)+\cdots+n(n!)=(n+1)!-1$.

Exercise 3.7.3

Prove that for $n \geq 1$: $\sum_{k=1}^{n} k^2 = \frac{1}{6}n(n+1)(2n+1)$.

Answer

Proof:

- Basis: We note the the proposition is true when $n=1$: $\sum_{k=1}^{1} k^2 = 1 = \frac{1(2)(3)}{6}$.
- Induction: Assume that the proposition is true for some $n \geq 1$. This is the induction hypothesis.

$$\begin{aligned} \sum_{k=1}^{n+1} k^2 &= \sum_{k=1}^{n} k^2 + (n+1)^2 \\ &= \frac{n(n+1)(2n+1)}{6} + (n+1)^2 \quad \text{by the induction hypothesis} \\ &= \frac{(n+1)(2n^2+7n+6)}{6} \\ &= \frac{(n+1)(n+2)(2n+3)}{6} \quad \blacksquare \end{aligned}$$

Therefore, the truth of the proposition for n implies the truth of the proposition for $n+1$.

Exercise 3.7.4

Prove that for $n \geq 0$: $\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$.

Exercise 3.7.5

Use mathematical induction to show that for $n \geq 1$,

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

Answer

Basis: For $n = 1$, we observe that $\frac{1}{(1 \cdot 2)} = \frac{1}{(1+1)}$

Induction: Assume that for some $n \geqslant 1$, the formula is true.

Then:

$$\begin{aligned} \frac{1}{(1 \cdot 2)} + \cdots + \frac{1}{n(n+1)} + \frac{1}{(n+1)(n+2)} &= \frac{n}{n+1} + \frac{1}{(n+1)(n+2)} \\ &= \frac{(n+2)n}{(n+1)(n+2)} + \frac{1}{(n+1)(n+2)} \\ &= \frac{(n+1)^2}{(n+1)(n+2)} \\ &= \frac{n+1}{n+2} \quad \blacksquare \end{aligned}$$

Exercise 3.7.6

Prove that if $n \geq 2$, the generalized DeMorgan's Law is true:

$$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \Leftrightarrow (\neg p_1) \vee (\neg p_2) \vee \cdots \vee (\neg p_n).$$

Answer

Basis: ($n = 2$) Proven with a truth table already.

Induction: Assume the generalized DeMorgan's Law with n propositions is true for some $n \geq 2$.

$$\begin{aligned} \neg\left(p_1 \wedge p_2 \wedge \cdots \wedge p_n \wedge p_{n+1}\right) &\Leftrightarrow \neg\left(\left(p_1 \wedge p_2 \wedge \cdots \wedge p_n\right) \wedge p_{n+1}\right) \\ &\Leftrightarrow \neg\left(p_1 \wedge p_2 \wedge \cdots \wedge p_n\right) \vee \left(\neg p_{n+1}\right) \\ &\Leftrightarrow \left(\left(\neg p_1\right) \vee \left(\neg p_2\right) \vee \cdots \vee \left(\neg p_n\right)\right) \vee \left(\neg p_{n+1}\right) \\ &\Leftrightarrow \left(\neg p_1\right) \vee \left(\neg p_2\right) \vee \cdots \vee \left(\neg p_n\right) \vee \left(\neg p_{n+1}\right) \end{aligned}$$

Exercise 3.7.7

The number of strings of n zeros and ones that contain an even number of ones is 2^{n-1}. Prove this fact by induction for $n \geq 1$.

Answer

Let A_n be the set of strings of zeros and ones of length n (we assume that $|A_n| = 2^n$ is known). Let E_n be the set of the "even" strings, and $E_n^c =$ the odd strings. The problem is to prove that for $n \geqslant 1$, $|E_n| = 2^{n-1}$. Clearly, $|E_1| = 1$, and, if for some $n \geqslant 1$, $|E_n| = 2^{n-1}$, it follows that $|E_{n+1}| = 2^n$ by the following reasoning.

We partition E_{n+1} according to the first bit: $E_{n+1} = \{1s \mid s \in E_n^c\} \cup \{0s \mid s \in E_n\}$

Since $\{1s \mid s \in E_n^c\}$ and $\{0s \mid s \in E_n\}$ are disjoint, we can apply the addition law. Therefore,

$$\begin{aligned} |E_{n+1}| &= |E_n^c| + |E_n| \\ &= 2^{n-1} + (2^n - 2^{n-1}) = 2^n. \quad \blacksquare \end{aligned}$$

Exercise 3.7.8

Let $p(n)$ be $8^n - 3^n$ is a multiple of 5. Prove that $p(n)$ is a tautology over $\mathbb{N}$.

Exercise 3.7.9

Suppose that there are n people in a room, $n \geq 1$, and that they all shake hands with one another. Prove that $\frac{n(n-1)}{2}$ handshakes will have occurred.

Answer

Assume that for n persons $(n \geqslant 1)$, $\frac{(n-1)n}{2}$ handshakes take place. If one more person enters the room, he or she will shake hands with n people,

$$\begin{aligned} \frac{(n-1)n}{2} + n &= \frac{n^2 - n + 2n}{2} \\ &= \frac{n^2 + n}{2} = \frac{n(n+1)}{2} \\ &= \frac{((n+1)-1)(n+1)}{2} \end{aligned}$$

Also, for $n = 1$, there are no handshakes, which matches the conjectured formula:

$$\frac{(1-1)(1)}{2} = 0 \quad \blacksquare.$$

Exercise 3.7.10

Prove that it is possible to make up any postage of eight cents or more using only three- and five-cent stamps.

Exercise 3.7.11

Generalized associativity. It is well known that if a_1, a_2, and a_3 are numbers, then no matter what order the sums in the expression $a_1 + a_2 + a_3$ are taken in, the result is always the same. Call this fact $p(3)$ and assume it is true. Prove using course-of-values induction that if a_1, a_2, ..., and a_n are numbers, then no matter what order the sums in the expression $a_1 + a_2 + \cdots + a_n$ are taken in, the result is always the same.

Answer

Let $p(n)$ be "$a_1 + a_2 + \cdots + a_n$ has the same value no matter how it is evaluated."

Basis: $a_1 + a_2 + a_3$ may be evaluated only two ways. Since + is associative, $(a_1 + a_2) + a_3 = a_1 + (a_2 + a_3)$. Hence, $p(3)$ is true.

Induction: Assume that for some $n \geq 3$, $p(3), p(4), \ldots, p(n)$ are all true. Now consider the sum $a_1 + a_2 + \cdots + a_n + a_{n+1}$. Any of the n additions in this expression can be applied last. If the jth addition is applied last, we have $c_j = (a_1 + a_2 + \cdots + a_j) + (a_{j+1} + \cdots + a_{n+1})$. No matter how the expression to the left and right of the jth addition are evaluated, the result will always be the same by the induction hypothesis, specifically $p(j)$ and $p(n+1-j)$. We now can prove that $c_1 = c_2 = \cdots = c_n$. If $i < j$,

$$\begin{aligned} c_i &= (a_1 + a_2 + \cdots + a_i) + (a_{i+1} + \cdots + a_{n+1}) \\ &= (a_1 + a_2 + \cdots + a_i) + ((a_{i+1} + \cdots + a_j) + (a_{j+1} + \cdots + a_{n+1}) \\ &= ((a_1 + a_2 + \cdots + a_i) + ((a_{i+1} + \cdots + a_j)) + (a_{j+1} + \cdots + a_{n+1}) \\ &= ((a_1 + a_2 + \cdots + a_j)) + (a_{j+1} + \cdots + a_{n+1}) \\ &= c_j \qquad \square \end{aligned}$$

Exercise 3.7.12

Let S be the set of all numbers that can be produced by applying any of the rules below in any order a finite number of times.

- Rule 1: $\frac{1}{2} \in S$
- Rule 2: $1 \in S$
- Rule 3: If a and b have been produced by the rules, then $ab \in S$.
- Rule 4: If a and b have been produced by the rules, then $\frac{a+b}{2} \in S$.

Prove that $a \in S \Rightarrow 0 \leq a \leq 1$.

Hint

The number of times the rules are applied should be the integer that you do the induction on.

Exercise 3.7.13

Proofs involving objects that are defined recursively are often inductive. A recursive definition is similar to an inductive proof. It consists of a basis, usually the simple part of the definition, and the recursion, which defines complex objects in terms of simpler ones. For example, if x is a real number and n is a positive integer, we can define x^n as follows:

- Basis: $x^1 = x$.
- Recursion: if $n \geq 2$, $x^n = x^{n-1}x$.

For example, $x^3 = x^2x = (x^1x)x = (xx)x$.

Prove that if $n, m \in \mathbb{P}$, $x^{m+n} = x^m x^n$. There is much more on recursion in Chapter 8.

Hint

Let $p(m)$ be the proposition that $x^{m+n} = x^m x^n$ for all $n \geq 1$.

Answer

For $m \geqslant 1$, let $p(m)$ be $x^{n+m} = x^n x^m$ for all $n \geqslant 1$. The basis for this proof follows directly from the basis for the definition of exponentiation.

Induction: Assume that for some $m \geqslant 1, p(m)$ is true. Then

$$\begin{aligned} x^{n+(m+1)} &= x^{(n+m)+1} && \text{by associativity of integer addition} \\ &= x^{n+m}x^1 && \text{by recursive definition} \\ &= x^n x^m x^1 && \text{induction hypothesis} \\ &= x^n x^{m+1} && \text{recursive definition} \quad \square \end{aligned}$$

Exercise 3.7.14

Let S be a finite set and let P_n be defined recursively by $P_1 = S$ and $P_n = S \times P_{n-1}$ for $n \geq 2$.

- List the elements of P_3 for the case $S = \{a, b\}$.
- Determine the formula for $|P_n|$, given that $|S| = k$, and prove your formula by induction.

This page titled 3.7: Mathematical Induction is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.8: Quantifiers

As we saw in Section 3.6, if $p(n)$ is a proposition over a universe U, its truth set T_p is equal to a subset of U. In many cases, such as when $p(n)$ is an equation, we are most concerned with whether T_p is empty or not. In other cases, we might be interested in whether $T_p = U$; that is, whether $p(n)$ is a tautology. Since the conditions $T_p \neq \emptyset$ and $T_p = U$ are so often an issue, we have a special system of notation for them.

3.8.1: Existential Quantifier

Definition 3.8.1: The Existential Quantifier

If $p(n)$ is a proposition over U with $T_p \neq \emptyset$, we commonly say "There exists an n in U such that $p(n)$ (is true)." We abbreviate this with the symbols $(\exists n)_U(p(n))$. The symbol $\exists$ is called the existential quantifier. If the context is clear, the mention of U is dropped: $(\exists n)(p(n))$.

Example 3.8.1: Some Examples of Existential Quantifiers

a. $(\exists k)_{\mathbb{Z}}(k^2 - k - 12 = 0)$ is another way of saying that there is an integer that solves the equation $k^2 - k - 12 = 0$. The fact that two such integers exist doesn't affect the truth of this proposition in any way.
b. $(\exists k)_{\mathbb{Z}}(3k = 102)$ simply states that 102 is a multiple of 3, which is true. On the other hand, $(\exists k)_{\mathbb{Z}}(3k = 100)$ states that 100 is a multiple of 3, which is false.
c. $(\exists x)_{\mathbb{R}}(x^2 + 1 = 0)$ is false since the solution set of the equation $x^2 + 1 = 0$ in the real numbers is empty. It is common to write $(\nexists x)_{\mathbb{R}}(x^2 + 1 = 0)$ in this case.

There are a wide variety of ways that you can write a proposition with an existential quantifier. Table 3.8.1 contains a list of different variations that could be used for both the existential and universal quantifiers.

3.8.2: Universal Quantifier

Definition 3.8.2: The Universal Quantifier

If $p(n)$ is a proposition over U with $T_p = U$, we commonly say "For all n in U, $p(n)$ (is true)." We abbreviate this with the symbols $(\forall n)_U(p(n))$. The symbol $\forall$ is called the universal quantifier. If the context is clear, the mention of U is dropped: $(\forall n)(p(n))$.

Example 3.8.2: Some Universal Quantifiers

a. We can say that the square of every real number is non-negative symbolically with a universal quantifier: $(\forall x)_{\mathbb{R}}(x^2 \geq 0)$.
b. $(\forall n)_{\mathbb{Z}}(n + 0 = 0 + n = n)$ says that the sum of zero and any integer n is n. This fact is called the identity property of zero for addition.

Table 3.8.1: Notational Variations with Quantifier Expressions

Universal Quantifier	Existential Quantifier
$(\forall n)_U(p(n))$	$(\exists n)_U(p(n))$
$(\forall n \in U)(p(n))$	$(\exists n \in U)(p(n))$
$\forall n \in U, p(n)$	$\exists n \in U$ such that $p(n)$
$p(n), \forall n \in U$	$p(n)$ is true for some $n \in U$
$p(n)$ is true for all $n \in U$	

3.8.3: 3.8.3: Negation of Quantified Propositions

When you negate a quantified proposition, the existential and universal quantifiers complement one another.

Example 3.8.3: Negation of an Existential Quantifier

Over the universe of animals, define $F(x)$: x is a fish and $W(x)$: x lives in the water. We know that the proposition $W(x) \to F(x)$ is not always true. In other words, $(\forall x)(W(x) \to F(x))$ is false. Another way of stating this fact is that there exists an animal that lives in the water and is not a fish; that is,

$$\begin{aligned} \neg(\forall x)(W(x) \to F(x)) &\Leftrightarrow (\exists x)(\neg(W(x) \to F(x))) \\ &\Leftrightarrow (\exists x)(W(x) \wedge \neg F(x)) \end{aligned} .$$

Note that the negation of a universally quantified proposition is an existentially quantified proposition. In addition, when you negate an existentially quantified proposition, you get a universally quantified proposition. Symbolically,

Table 3.8.2: Negation of Quantified Expressions

$\neg((\forall n)_U(p(n))) \Leftrightarrow (\exists n)_U(\neg p(n))$
$\neg((\exists n)_U(p(n))) \Leftrightarrow (\forall n)_U(\neg p(n))$

Example 3.8.4: More Negations of Quantified Expressions

a. The ancient Greeks first discovered that $\sqrt{2}$ is an irrational number; that is, $\sqrt{2}$ is not a rational number. $\neg((\exists r)_{\mathbb{Q}}(r^2 = 2))$ and $(\forall r)_{\mathbb{Q}}(r^2 \neq 2)$ both state this fact symbolically.
b. $\neg((\forall n)_{\mathbb{P}}(n^2 - n + 41 \text{ is prime}))$ is equivalent to $(\exists n)_{\mathbb{P}}(n^2 - n + 41 \text{ is composite})$. They are either both true or both false.

3.8.4: Multiple Quantifiers

If a proposition has more than one variable, then you can quantify it more than once. For example, $p(x, y) : x^2 - y^2 = (x + y)(x - y)$ is a tautology over the set of all pairs of real numbers because it is true for each pair (x, y) in $\mathbb{R} \times \mathbb{R}$. Another way to look at this proposition is as a proposition with two variables. The assertion that $p(x, y)$ is a tautology could be quantified as $(\forall x)_{\mathbb{R}}((\forall y)_{\mathbb{R}}(p(x, y)))$ or $(\forall y)_{\mathbb{R}}((\forall x)_{\mathbb{R}}(p(x, y)))$

In general, multiple universal quantifiers can be arranged in any order without logically changing the meaning of the resulting proposition. The same is true for multiple existential quantifiers. For example, $p(x, y) : x + y = 4$ and $x - y = 2$ is a proposition over $\mathbb{R} \times \mathbb{R}$. $(\exists x)_{\mathbb{R}}((\exists y)_{\mathbb{R}}(x + y = 4 \text{ and } x - y = 2))$ and $(\exists y)_{\mathbb{R}}((\exists x)_{\mathbb{R}}(x + y = 4 \text{ and } x - y = 2))$ are equivalent. A proposition with multiple existential quantifiers such as this one says that there are simultaneous values for the quantified variables that make the proposition true. A similar example is $q(x, y) : 2x - y = 2 \text{ and } 4x - 2y = 5$, which is always false; and the following are all equivalent:

$$\begin{aligned} \neg((\exists x)_{\mathbb{R}}((\exists y)_{\mathbb{R}}(q(x, y)))) &\Leftrightarrow \neg(\exists y)_{\mathbb{R}}((\exists x)_{\mathbb{R}}(q(x, y)))) \\ &\Leftrightarrow (\forall y)_{\mathbb{R}}(\neg((\exists x)_{\mathbb{R}}(q(x, y)))) \\ &\Leftrightarrow ((\forall y)_{\mathbb{R}}((\forall x)_{\mathbb{R}}(\neg q(x, y)))) \\ &\Leftrightarrow ((\forall x)_{\mathbb{R}}((\forall y)_{\mathbb{R}}(\neg q(x, y)))) \end{aligned}$$

When existential and universal quantifiers are mixed, the order cannot be exchanged without possibly changing the meaning of the proposition. For example, let $\mathbb{R}^+$ be the positive real numbers, $x : (\forall a)_{\mathbb{R}^+}((\exists b)_{\mathbb{R}^+}(ab = 1))$ and $y : (\exists b)_{\mathbb{R}^+}((\forall a)_{\mathbb{R}^+}(ab = 1))$ have different logical values; x is true, while y is false.

Tips on Reading Multiply-Quantified Propositions. It is understandable that you would find propositions such as x difficult to read. The trick to deciphering these expressions is to "peel" one quantifier off the proposition just as you would peel off the layers of an onion (but quantifiers shouldn't make you cry!). Since the outermost quantifier in x is universal, x says that $z(a) : (\exists b)_{\mathbb{R}^+}(ab = 1)$ is true for each value that a can take on. Now take the time to select a value for a, like 6. For the value that we selected, we get $z(6) : (\exists b)_{\mathbb{R}^+}(6b = 1)$, which is obviously true since $6b = 1$ has a solution in the positive real numbers. We will get that same truth value no matter which positive real number we choose for a; therefore, $z(a)$ is a tautology over $\mathbb{R}^+$ and we are justified in

saying that x is true. The key to understanding propositions like x on your own is to experiment with actual values for the outermost variables as we did above.

Now consider y. To see that y is false, we peel off the outer quantifier. Since it is an existential quantifier, all that y says is that some positive real number makes $w(b)$: $(\forall a)_{\mathbb{R}^+}(ab = 1)$ true. Choose a few values of b to see if you can find one that makes $w(b)$ true. For example, if we pick $b = 2$, we get $(\forall a)_{\mathbb{R}^+}(2a = 1)$, which is false, since $2a$ is almost always different from 1. You should be able to convince yourself that no value of b will make $w(b)$ true. Therefore, y is false.

Another way of convincing yourself that y is false is to convince yourself that $\neg y$ is true:

$$\begin{aligned}\neg((\exists b)_{\mathbb{R}^+}((\forall a)_{\mathbb{R}^+}(ab = 1))) &\Leftrightarrow (\forall b)_{\mathbb{R}^+}\neg((\forall a)_{\mathbb{R}^+}(ab = 1)) \\ &\Leftrightarrow (\forall b)_{\mathbb{R}^+}((\exists a)_{\mathbb{R}^+}(ab \neq 1))\end{aligned}$$

In words, for each value of b, there is a value for a that makes $ab \neq 1$. One such value is $a = \frac{1}{b} + 1$. Therefore, $\neg y$ is true.

3.8.5: Exercises

Exercise 3.8.1

Let $C(x)$ be "x is cold-blooded," let $F(x)$ be "x is a fish," and let $S(x)$ be "x lives in the sea."

a. Translate into a formula: Every fish is cold-blooded.
b. Translate into English: $(\exists x)(S(x) \wedge \neg F(x))$.
c. Translate into English: $(\forall x)(F(x) \rightarrow S(x))$.

Answer

1. $(\forall x)(F(x) \rightarrow C(x))$
2. There are objects in the sea which are not fish.
3. Every fish lives in the sea.

Exercise 3.8.2

Let $M(x)$ be "x is a mammal," let $A(x)$ be "x is an animal," and let $W(x)$ be "x is warm-blooded."

a. Translate into a formula: Every mammal is warm-blooded.
b. Translate into English: $(\exists x)(A(x) \wedge (\neg M(x)))$.

Exercise 3.8.3

Over the universe of books, define the propositions $B(x)$: x has a blue cover, $M(x)$: x is a mathematics book, $U(x)$: x is published in the United States, and $R(x, y)$: The bibliography of x includes y.

Translate into words:

a. $(\exists x)(\neg B(x))$.
b. $(\forall x)(M(x) \wedge U(x) \rightarrow B(x))$.
c. $(\exists x)(M(x) \wedge \neg B(x))$.
d. $(\exists y)((\forall x)(M(x) \rightarrow R(x, y)))$.
e. Express using quantifiers: Every book with a blue cover is a mathematics book.
f. Express using quantifiers: There are mathematics books that are published outside the United States.
g. Express using quantifiers: Not all books have bibliographies.

Answer

a. There is a book with a cover that is not blue.
b. Every mathematics book that is published in the United States has a blue cover.
c. There exists a mathematics book with a cover that is not blue.
d. There exists a book that appears in the bibliography of every mathematics book.
e. $(\forall x)(B(x) \rightarrow M(x))$

f. $(\exists x)(M(x) \wedge \neg U(x))$
g. $(\exists x)((\forall y)(\neg R(x, y))$

Exercise 3.8.4

Let the universe of discourse, U, be the set of all people, and let $M(x, y)$ be "x is the mother of y."

Which of the following is a true statement? Translate it into English.

a. $(\exists x)_U((\forall y)_U(M(x, y)))$
b. $(\forall y)_U((\exists x)_U(M(x, y)))$
c. Translate the following statement into logical notation using quantifiers and the proposition $M(x, y)$: "Everyone has a maternal grandmother."

Exercise 3.8.5

Translate into your own words and indicate whether it is true or false that $(\exists u)_{\mathbb{Z}}(4u^2 - 9 = 0)$.

Answer

The equation $4u^2 - 9 = 0$ has a solution in the integers. (False)

Exercise 3.8.6

Use quantifiers to say that $\sqrt{3}$ is an irrational number.

Hint

Your answer will depend on your choice of a universe

Exercise 3.8.7

What do the following propositions say, where U is the power set of $\{1, 2, \ldots, 9\}$?Which of these propositions are true?

a. $(\forall A)_U|A| \neq |A^c|$.
b. $(\exists A)_U(\exists B)_U(|A| = 5, |B| = 5, \text{ and } A \cap B = \emptyset)$.
c. $(\forall A)_U(\forall B)_U(A - B = B^c - A^c)$.

Answer

a. Every subset of U has a cardinality different from its complement. (True)
b. There is a pair of disjoint subsets of U both having cardinality 5. (False)
c. $A - B = B^c - A^c$ is a tautology. (True)

Exercise 3.8.8

Use quantifiers to state that for every positive integer, there is a larger positive integer.

Exercise 3.8.9

Use quantifiers to state that the sum of any two rational numbers is rational.

Answer

$(\forall a)_{\mathbb{Q}}(\forall b)_{\mathbb{Q}}(a + b$ is a rational number.)

Exercise 3.8.10

Over the universe of real numbers, use quantifiers to say that the equation $a + x = b$ has a solution for all values of a and b.

Hint

You will need three quantifiers.

Exercise 3.8.11

Let n be a positive integer. Describe using quantifiers:

a. $x \in \bigcup_{k=1}^{n} A_k$

b. $x \in \bigcap_{k=1}^{n} A_k$

Answer

Let $I = \{1, 2, 3, \ldots, n\}$

a. $(\exists x)_I \, (x \in A_i)$

b. $(\forall x)_I \, (x \in A_i)$

Exercise 3.8.12

Prove that $(\exists x)(\forall y)(p(x, y)) \Rightarrow (\forall y)(\exists x)(p(x, y))$, but that converse is not true.

This page titled 3.8: Quantifiers is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

3.9: A Review of Methods of Proof

One of the major goals of this chapter is to acquaint the reader with the key concepts in the nature of proof in logic, which of course carries over into all areas of mathematics and its applications. In this section we will stop, reflect, and "smell the roses," so that these key ideas are not lost in the many concepts covered in logic. In Chapter 4 we will use set theory as a vehicle for further practice and insights into methods of proof.

3.9.1: Concepts in Proof

All theorems in mathematics can be expressed in form "If P then C" ($P \Rightarrow C$), or in the form "C_1 if and only if C_2" ($C_1 \Leftrightarrow C_2$). The latter is equivalent to "If C_1 then C_2," and "If C_2 then C_1."

In "If P then C," P is the premise (or hypothesis) and C is the conclusion. It is important to realize that a theorem makes a statement that is dependent on the premise being true.

There are two basic methods for proving $P \Rightarrow C$:

- *Directly:* Assume P is true and prove C is true.
- *Indirectly (or by contradiction):* Assume P is true and C is false and prove that this leads to a contradiction of some premise, theorem, or basic truth.

The method of proof for "If and only if" theorems is found in the law $(P \leftrightarrow C) \Leftrightarrow ((P \rightarrow C) \wedge (C \rightarrow P))$. Hence to prove an "If and only if" statement one must prove an "if . . . then ..." statement and its converse.

The initial response of most people when confronted with the task of being told they must be able to read and do proofs is often "Why?" or "I can't do proofs." To answer the first question, doing proofs or problem solving, even on the most trivial level, involves being able to read statements. First we must understand the problem and know the hypothesis; second, we must realize when we are done and we must understand the conclusion. To apply theorems or algorithms we must be able to read theorems and their proofs intelligently.

To be able to do the actual proofs of theorems we are forced to learn:

- the actual meaning of the theorems, and
- the basic definitions and concepts of the topic discussed.

For example, when we discuss rational numbers and refer to a number x as being rational, this means we can substitute a fraction $\frac{p}{q}$ in place of x, with the understanding that p and q are integers and $q \neq 0$. Therefore, to prove a theorem about rational numbers it is absolutely necessary that you know what a rational number "looks like."

It's easy to comment on the response, "I cannot do proofs." Have you tried? As elementary school students we may have been awe of anyone who could handle algebraic expressions, especially complicated ones. We learned by trying and applying ourselves. Maybe we cannot solve all problems in algebra or calculus, but we are comfortable enough with these subjects to know that we can solve many and can express ourselves intelligently in these areas. The same remarks hold true for proofs.

3.9.2: Proving $P \Rightarrow C$

First one must completely realize what is given, the hypothesis. The importance of this is usually overlooked by beginners. It makes sense, whenever you begin any task, to spend considerable time thinking about the tools at your disposal. Write down the premise in precise language. Similarly, you have to know when the task is finished. Write down the conclusion in precise language. Then you usually start with P and attempt to show that C follows logically. How do you begin? Basically you attack the proof the same way you solve a complicated equation in elementary algebra. You may not know exactly what each and every step is but you must try something. If we are lucky, C follows naturally; if it doesn't, try something else. Often what is helpful is to work backward from C. Finally, we have all learned, possibly the hard way, that mathematics is a participating sport, not a spectator sport. One learns proofs by doing them, not by watching others do them. We give several illustrations of how to set up the proofs of several examples. Our aim here is not to prove the statements given, but to concentrate on the logical procedure.

Example 3.9.1: The Sum of Odd Integers

We will outline a proof that the sum of any two odd integers is even. Our first step will be to write the theorem in the familiar conditional form: If x and y are odd integers, then $x+y$ is even. The premise and conclusion of this theorem should be clear now. Notice that if x and y are not both odd, then the conclusion may or may not be true. Our only objective is to show that the truth of the premise forces the conclusion to be true. Therefore, we can express the integers x and y in the form that all odd integers take; that is:

$$n \in \mathbb{Z} \text{ is odd implies that } (\exists m \in \mathbb{Z})(n = 2m+1)$$

This observation allows us to examine the sum $x+y$ and to verify that it must be even.

One final important point: This example involves two odd integers that may or may not be equal. If we use the fact that x is odd and infer that $x = 2m+1$ for some integer m, we can do a similar thing with y. However, in this context we cannot write $y = 2m+1$ since we have already linked m to x. We need to use a different variable, maybe q or m' - any other symbol that is not already used in our discussion.

Example 3.9.2: The Square of an Even Integer

Let $n \in \mathbb{Z}$. We will outline a proof that n^2 is even if and only if n is even.

Outline of a proof: Since this is an "If and only if" theorem we must prove two things:

i. ($\Rightarrow$) If n^2 is even, then n is even. To do this directly, assume that n^2 is even and prove that n is even. To do this indirectly, assume n^2 is even and that n is odd, and reach a contradiction. It turns out that the latter of the two approaches is easiest here.
ii. ($\Leftarrow$) If n is even, then n^2 is even. To do this directly, assume that n is even and prove that n^2 is even.

Now that we have broken the theorem down into two parts and know what to prove, we proceed to prove the two implications. The final ingredient that we need is a convenient way of describing even integers. When we refer to an integer n (or m, or k,. . .) as even, we can always replace it with a product of the form $2q$, where q is an integer (more precisely, $(\exists q)_{\mathbb{Z}}(n = 2q)$). In other words, for an integer to be even it must have a factor of two in its prime decomposition.

Example 3.9.3: $\sqrt{2}$ is Irrational

Our final example will be an outline of the proof that the square root of 2 is irrational (not an element of $\mathbb{Q}$). This is an example of the theorem that does not appear to be in the standard $P \Rightarrow C$ form. One way to rephrase the theorem is: If x is a rational number, then $x^2 \neq 2$. A direct proof of this theorem would require that we verify that the square of every rational number is not equal to 2. There is no convenient way of doing this, so we must turn to the indirect method of proof. In such a proof, we assume that x is a rational number and that $x^2 = 2$. This will lead to a contradiction. In order to reach this contradiction, we need to use the following facts:

- A rational number is a quotient of two integers.
- Every fraction can be reduced to lowest terms, so that the numerator and denominator have no common factor greater than 1.
- If n is an integer, n^2 is even if and only if n is even.

3.9.3: Exercises

Exercise 3.9.1

Prove that the sum of two odd positive integers is an even positive integer. You might want to read Example 3.9.1 before attempting this.

Answer

The given statement can be written in if . . . , then . . . format as: If x and y are two odd positive integers, then $x+y$ is an even integer.

Proof: Assume x and y are two positive odd integers. It can be shown that $x+y=2\cdot(\text{some positive integer})$.

x odd and positive $\Rightarrow x=2m+1$ for some $m\geq 0$,

y odd and positive $\Rightarrow y=2n+1$ for some $n\geq 0$.

Then,

$$x+y=(2m+1)+(2n+1)=2((m+n)+1)=2\cdot(\text{some positive integer})$$

Therefore, $x+y$ is an even positive integer. □

Exercise 3.9.2

Write out a complete proof that if n is an integer, n^2 is even if and only if n is even.

Exercise 3.9.3

Write out a complete proof that $\sqrt{2}$ is irrational.

Answer

Proof: (Indirect) Assume to the contrary, that $\sqrt{2}$ is a rational number. Then there exists $p,q\in\mathbb{Z},(q\neq 0)$ where $\frac{p}{q}=\sqrt{2}$ and where $\frac{p}{q}$ is in lowest terms, that is, p and q have no common factor other than 1.

$$\begin{aligned}\frac{p}{q}=\sqrt{2}\Rightarrow\frac{p^2}{q^2}&=2\\ \Rightarrow p^2&=2q^2\\ \Rightarrow p^2&\text{ is an even integer}\\ \Rightarrow p&\text{ is an even integer (see Exercise 2)}\\ \Rightarrow 4&\text{ is a factor of }p^2\\ \Rightarrow q^2&\text{ is even}\\ \Rightarrow q&\text{ is even}\end{aligned}$$

Hence both p and q have a common factor, namely 2, which is a contradiction. □

Exercise 3.9.4

Prove that the cube root of 2 is an irrational number.

Exercise 3.9.5

Prove that if x and y are real numbers such that $x+y\leq 1$, then $x\leq\frac{1}{2}$ or $y\leq\frac{1}{2}$.

Answer

Proof: (Indirect) Assume $x,y\in\mathbb{R}$ and $x+y\leqslant 1$. Assume to the contrary that $\left(x\leqslant\frac{1}{2}\text{ or }y\leqslant\frac{1}{2}\right)$ is false, which is equivalent to $x>\frac{1}{2}$ and $y>\frac{1}{2}$. Hence $x+y>\frac{1}{2}+\frac{1}{2}=1$. This contradicts the assumption that $x+y\leqslant 1$. □

Exercise 3.9.6

Use the following definition of absolute value to prove the given statements: If x is a real number, then the absolute value of x, $|x|$, is defined by:

$$|x|=\begin{cases}x & \text{if } x\geq 0\\ -x & \text{if } x<0\end{cases}$$

a. For any real number x, $|x|\geq 0$. Moreover, $|x|=0$ implies $x=0$.

b. For any two real numbers x and y, $|x| \cdot |y| = |xy|$.
c. For any two real numbers x and y, $|x + y| \leq |x| + |y|$.

This page titled 3.9: A Review of Methods of Proof is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

4: More on Sets

In this chapter we shall look more closely at some basic facts about sets. One question we could ask ourselves is: Can we manipulate sets similarly to the way we manipulated expressions in basic algebra, or to the way we manipulated propositions in logic? In basic algebra we are aware that $a \cdot (b + c) = a \cdot b + a \cdot c$ for all real numbers a, b, and c. In logic we verified an analogue of this statement, namely, $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r))$, where $p, q,$ and r were arbitrary propositions. If A, B, and C are arbitrary sets, is $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$? How do we convince ourselves of it is truth, or discover that it is false? Let us consider some approaches to this problem, look at their pros and cons, and determine their validity. Later in this chapter, we introduce partitions of sets and minsets.

4.1: Methods of Proof for Sets

4.2: Laws of Set Theory

4.3: Minsets

4.4: The Duality Principle

This page titled 4: More on Sets is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

4.1: Methods of Proof for Sets

If A, B, and C are arbitrary sets, is it always true that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$? There are a variety of ways that we could attempt to prove that this distributive law for intersection over union is indeed true. We start with a common "non-proof" and then work toward more acceptable methods.

4.1.1: Examples and Counterexamples

We could, for example, let $A = \{1, 2\}$, $B = \{5, 8, 10\}$, and $C = \{3, 2, 5\}$, and determine whether the distributive law is true for these values of A, B, and C. In doing this we will have only determined that the distributive law is true for this one example. It does not prove the distributive law for all possible sets A, B, and C and hence is an invalid method of proof. However, trying a few examples has considerable merit insofar as it makes us more comfortable with the statement in question. Indeed, if the statement is not true for the example, we have disproved the statement.

Definition 4.1.1: Counterexample

An example that disproves a statement is called a counterexample.

Example 4.1.1: Disproving Distributivity of Addition over Multiplication

From basic algebra we learned that multiplication is distributive over addition. Is addition distributive over multiplication? That is, is $a + (b \cdot c) = (a + b) \cdot (a + c)$ always true? If we choose the values $a = 3$, $b = 4$, and $c = 1$, we find that $3 + (4 \cdot 1) \neq (3 + 4) \cdot (3 + 1)$. Therefore, this set of values serves as a counterexample to a distributive law of addition over multiplication.

4.1.2: Proof Using Venn Diagrams

In this method, we illustrate both sides of the statement via a Venn diagram and determine whether both Venn diagrams give us the same "picture," For example, the left side of the distributive law is developed in Figure 4.1.1 and the right side in Figure 4.1.2. Note that the final results give you the same shaded area.

The advantage of this method is that it is relatively quick and mechanical. The disadvantage is that it is workable only if there are a small number of sets under consideration. In addition, it doesn't work very well in a static environment like a book or test paper. Venn diagrams tend to work well if you have a potentially dynamic environment like a blackboard or video.

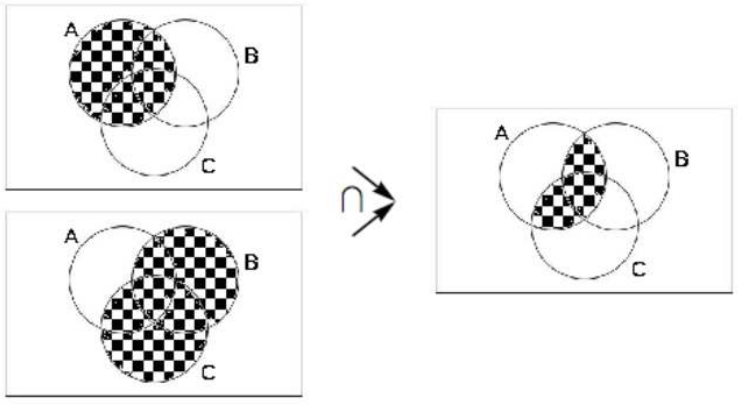


Figure 4.1.1: Development of the left side of the distributive law for sets.

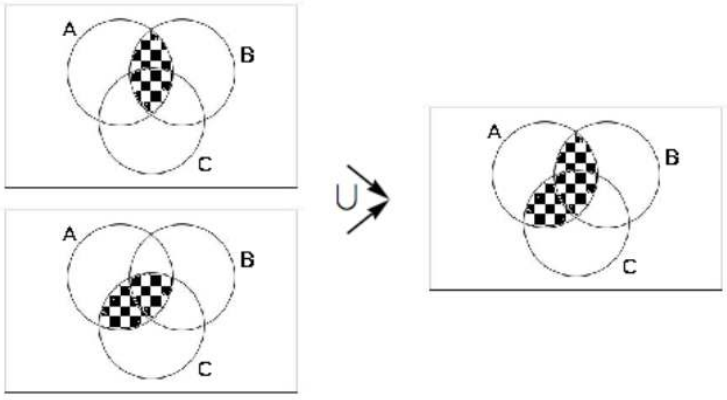


Figure 4.1.2: Development of the right side of the distributive law for sets

4.1.3: Proof using Set-membership Tables

Let A be a subset of a universal set U and let $u \in U$. To use this method we note that exactly one of the following is true: $u \in A$ or $u \notin A$. Denote the situation where $u \in A$ by 1 and that where $u \notin A$ by 0. Working with two sets, A and B, and if $u \in U$, there are four possible outcomes of "where u can be." What are they? The set-membership table for $A \cup B$ is:

Table 4.1.1: Membership Table for $A \cup B$

A	B	$A \cup B$
0	0	0
0	1	1
1	0	1
1	1	1

This table illustrates that $u \in A \cup B$ if and only if $u \in A$ or $u \in B$.

In order to prove the distributive law via a set-membership table, write out the table for each side of the set statement to be proved and note that if S and T are two columns in a table, then the set statement S is equal to the set statement T if and only if corresponding entries in each row are the same.

To prove $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, first note that the statement involves three sets, A, B, and C, so there are $2^3 = 8$ possibilities for the membership of an element in the sets.

Table 4.1.2: Membership table to prove the distributive law of intersection over union

A	B	C	$B \cup C$	$A \cap B$	$A \cap C$	$A \cap (B \cup C)$	$(A \cap B) \cup (A \cap$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

Since each entry in Column 7 is the same as the corresponding entry in Column 8, we have shown that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ for any sets A, B, and C. The main advantage of this method is that it is mechanical. The main

disadvantage is that it is reasonable to use only for a relatively small number of sets. If we are trying to prove a statement involving five sets, there are $2^5 = 32$ rows, which would test anyone's patience doing the work by hand.

4.1.4: Proof Using Definitions

This method involves using definitions and basic concepts to prove the given statement. This procedure forces one to learn, relearn, and understand basic definitions and concepts. It helps individuals to focus their attention on the main ideas of each topic and therefore is the most useful method of proof. One does not learn a topic by memorizing or occasionally glancing at core topics, but by using them in a variety of contexts. The word proof panics most people; however, everyone can become comfortable with proofs. Do not expect to prove every statement immediately. In fact, it is not our purpose to prove every theorem or fact encountered, only those that illustrate methods and/or basic concepts. Throughout the text we will focus in on main techniques of proofs. Let's illustrate by proving the distributive law.

Proof Technique 1. State or restate the theorem so you understand what is given (the hypothesis) and what you are trying to prove (the conclusion).

Theorem 4.1.1: The Distributive Law of Intersection over Union

If A, B, and C are sets, then $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Proof

What we can assume: A, B, and C are sets.

What we are to prove: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Commentary: What types of objects am I working with: sets? real numbers? propositions? The answer is sets: sets of elements that can be anything you care to imagine. The universe from which we draw our elements plays no part in the proof of this theorem.

We need to show that the two sets are equal. Let's call them the left-hand set (LHS) and the right-hand set (RHS). To prove that $LHS = RHS$, we must prove two things: (a) $LHS \subseteq RHS$, and (b) $RHS \subseteq LHS$.

To prove part a and, similarly, part b, we must show that each element of LHS is an element of RHS. Once we have diagnosed the problem we are ready to begin.

We must prove: (a) $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.

Let $x \in A \cap (B \cup C)$:

$$\begin{aligned} x \in A \cap (B \cup C) &\Rightarrow x \in A \text{ and } (x \in B \text{ or } x \in C) \\ &\quad \text{def. of union and intersection} \\ &\Rightarrow (x \in A \text{ and } x \in B) \text{ or } (x \in A \text{ and } x \in C) \\ &\quad \text{distributive law of logic} \\ &\Rightarrow (x \in A \cap B) \text{ or } (x \in A \cap C) \\ &\quad \text{def. of intersection} \\ &\Rightarrow x \in (A \cap B) \cup (A \cap C) \\ &\quad \text{def. of union} \end{aligned}$$

We must also prove (b) $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$.

$$\begin{aligned} x \in (A \cap B) \cup (A \cap C) &\Rightarrow (x \in A \cap B) \text{or } (x \in A \cap C) \\ &\quad \text{Why?} \\ &\Rightarrow (x \in A \text{ and } x \in B) \text{ or } (x \in A \text{ and } x \in C) \\ &\quad \text{Why?} \\ &\Rightarrow x \in A \text{ and } (x \in B \text{ or } x \in C) \\ &\quad \text{Why?} \\ &\Rightarrow x \in A \cap (B \cup C) \\ &\quad \text{Why?} \ \square \end{aligned}$$

Proof Technique 2

1. To prove that $A \subseteq B$, we must show that if $x \in A$, then $x \in B$.
2. To prove that $A = B$, we must show:
   a. $A \subseteq B$ and
   b. $B \subseteq A$.

To further illustrate the Proof-by-Definition technique, let's prove the following theorem.

Theorem 4.1.2: Another Proof using Definitions

If A, B, and C are any sets, then $A \times (B \cap C) = (A \times B) \cap (A \times C)$.

Proof

Commentary; We again ask ourselves: What are we trying to prove? What types of objects are we dealing with? We realize that we wish to prove two facts: (a) $LHS \subseteq RHS$, and (b) $RHS \subseteq LHS$.

To prove part (a), and similarly part (b), we'll begin the same way. Let ___ $\in LHS$ to show ___ $\in RHS$. What should ___ be? What does a typical object in the LHS look like?

Now, on to the actual proof.

a. $A \times (B \cap C) \subseteq (A \times B) \cap (A \times C)$.
Let $(x, y) \in A \times (B \cap C)$.

$$\begin{aligned}(x,y) \in A \times (B \cap C) &\Rightarrow x \in A \text{ and } y \in (B \cap C)\\ &\quad \text{Why?}\\ &\Rightarrow x \in A \text{ and } (y \in B \text{ and } y \in C)\\ &\quad \text{Why?}\\ &\Rightarrow (x \in A \text{ and } y \in B) \text{ and } (x \in A \text{ and } y \in C)\\ &\quad \text{Why?}\\ &\Rightarrow (x,y) \in (A \times B) \text{ and } (x,y) \in (A \times C)\\ &\quad \text{Why?}\\ &\Rightarrow (x,y) \in (A \times B) \cap (A \times C)\\ &\quad \text{Why?}\end{aligned}$$

b. $(A \times B) \cap (A \times C) \subseteq A \times (B \cap C)$.
Let $(x, y) \in (A \times B) \cap (A \times C)$.

$$\begin{aligned}(x,y) \in (A \times B) \cap (A \times C) &\Rightarrow (x,y) \in A \times B \text{ and } (x,y) \in A \times C\\ &\quad \text{Why?}\\ &\Rightarrow (x \in A \text{ and } y \in B) \text{ and } (x \in A \text{ and } y \in C)\\ &\quad \text{Why?}\\ &\Rightarrow x \in A \text{ and } (y \in B \text{ and } y \in C)\\ &\quad \text{Why?}\\ &\Rightarrow x \in A \text{ and } y \in (B \cap C)\\ &\quad \text{Why?}\\ &\Rightarrow (x,y) \in A \times (B \cap C)\\ &\quad \text{Why?}\end{aligned}$$

4.1.5: Exercises

Exercise 4.1.1

Prove the following:

a. Let A, B, and C be sets. If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.
b. Let A and B be sets. Then $A - B = A \cap B^c$.
c. Let A, B, and C be sets. If ($A \subseteq B$ and $A \subseteq C$) then $A \subseteq B \cap C$.
d. Let A and B be sets. $A \subseteq B$ if and only if $B^c \subseteq A^c$.

e. Let $A, B,$ and C be sets. If $A \subseteq B$ then $A \times C \subseteq B \times C$.

Answer

a. Assume that $x \in A$ (condition of the conditional conclusion $A \subseteq C$). Since $A \subseteq B$, $x \in B$ by the definition of $\subseteq$. $B \subseteq C$ and $x \in B$ implies that $x \in C$. Therefore, if $x \in A$, then $x \in C$. $\square$

b. (Proof that $A - B \subseteq A \cap B^c$) Let x be in $A - B$. Therefore, x is in A, but it is not in B; that is, $x \in A$ and $x \in B^c \Rightarrow x \in A \cap B^c$. $\square$

c. $(\Rightarrow)$ Assume that $A \subseteq B$ and $A \subseteq C$. Let $x \in A$. By the two premises,$x \in B$ and $x \in C$. Therefore, by the definition of intersection, $x \in B \cap C$. $\square$

d. $(\Rightarrow)$ (Indirect) Assume that B^c is not a subset of A^c . Therefore, there exists $x \in B^c$ that does not belong to A^c. $x \notin A^c \Rightarrow x \in A$. Therefore, $x \in A$ and $x \notin B$, a contradiction to the assumption that $A \subseteq B$. $\square$

e. There are two cases to consider. The first is when C is empty. Then the conclusion follows since both Cartesian products are empty.
If C isn't empty, we have two subcases, if A is empty, $A \times C = \emptyset$, which is a subset of every set. Finally, the interesting subcase is when A is not empty. Now we pick any pair $(a, c) \in A \times C$. This means that a is in A and c is in C. Since A is a subset of B, a is in B and so $(a, c) \in B \times C$. Therefore $A \times C \subseteq B \times C$. $\square$

Exercise 4.1.2

Write the converse of parts (a), (c), and (e) of Exercise 1 and prove or disprove them.

Exercise 4.1.3

Disprove the following, assuming $A, B,$ and C are sets:

a. $A - B = B - A$.
b. $A \times B = B \times A$.
c. $A \cap B = A \cap C$ implies $B = C$.
d. $A \oplus (B \cap C) = (A \oplus B) \cap (A \oplus C)$

Answer

a. If $A = \mathbb{Z}$ and $B = \emptyset$, $A - B = \mathbb{Z}$, while $B - A = \emptyset$.
b. If $A = \{0\}$ and $B = \{1\}$, $(0, 1) \in A \times B$, but $(0, 1)$ is not in $B \times A$.
c. Let $A = \emptyset$, $B = \{0\}$, and $C = \{1\}$.
d. If $A = \{1\}$, $B = \{1\}$, and $C = \emptyset$, then the left hand side of the identity is $\{1\}$while the right hand side is the empty set. Another example is $A = \{1, 2\}, B = \{1\}$, and $C = \{2\}$.

Exercise 4.1.4

Let $A, B,$ and C be sets. Write the following in "if . . . then . . ." language and prove:

a. $x \in B$ is a sufficient condition for $x \in A \cup B$.
b. $A \cap B \cap C = \emptyset$ is a necessary condition for $A \cap B = \emptyset$.
c. $A \cup B = B$ is a necessary and sufficient condition for $A \subseteq B$.

Exercise 4.1.5

Prove by induction that if A, B_1, B_2, ... , B_n are sets, $n \geq 2$, then $A \cap (B_1 \cup B_2 \cup \cdots \cup B_n) = (A \cap B_1) \cup (A \cap B_2) \cup \cdots \cup (A \cap B_n)$.

Answer

Proof: Let $p(n)$ be

$$A \cap (B_1 \cup B_2 \cup \cdots \cup B_n) = (A \cap B_1) \cup (A \cap B_2) \cup \cdots \cup (A \cap B_n).$$

Basis: We must show that $p(2): A \cap (B_1 \cup B_2) = (A \cap B_1) \cup (A \cap B_2)$ is true. This was done by several methods in Section 4.1.

Induction: Assume for some $n \geq 2$ that $p(n)$ is true. Then

$$\begin{aligned} A \cap (B_1 \cup B_2 \cup \cdots \cup B_{n+1}) &= A \cap ((B_1 \cup B_2 \cup \cdots \cup B_n) \cup B_{n+1}) \\ &= (A \cap (B_1 \cup B_2 \cup \cdots \cup B_n)) \cup (A \cap B_{n+1}) \quad \text{by } p(2) \\ &= ((A \cap B_1) \cup \cdots \cup (A \cap B_n)) \cup (A \cap B_{n+1}) \quad \text{by the induction hypothesis} \\ &= (A \cap B_1) \cup \cdots \cup (A \cap B_n) \cup (A \cap B_{n+1}) \quad \square \end{aligned}$$

Exercise 4.1.6

Let A, B and C be sets. Prove or disprove:

$$A \cap B \neq \emptyset, B \cap C \neq \emptyset \Rightarrow A \cap C \neq \emptyset$$

Answer

The statement is false. The sets $A = \{1, 2\}$, $B = \{2, 3\}$ and $C = \{3, 4\}$ provide a counterexample. Looking ahead to Chapter 6, we would say that the relation of being non-disjoint is not transitive, Definition 6.3.3.

This page titled 4.1: Methods of Proof for Sets is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

4.2: Laws of Set Theory

4.2.1: Tables of Laws

The following basic set laws can be derived using either the Basic Definition or the Set-Membership approach and can be illustrated by Venn diagrams.

Table 4.2.1: Basic Laws of Set Theory

Commutative Laws	
(1) $A \cup B = B \cup A$	(1′) $A \cap B = B \cap A$
Associative Laws	
(2) $A \cup (B \cup C) = (A \cup B) \cup C$	(2′) $A \cap (B \cap C) = (A \cap B) \cap C$
Distributive Laws	
(3) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	(3′) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Identity Laws	
(4) $A \cup \emptyset = \emptyset \cup A = A$	(4′) $A \cap U = U \cap A = A$
Complement Laws	
(5) $A \cup A^c = U$	(5′) $A \cap A^c = \emptyset$
Idempotent Laws	
(6) $A \cup A = A$	(6′) $A \cap A = A$
Null Laws	
(7) $A \cup U = U$	(7′) $A \cap \emptyset = \emptyset$
Absorption Laws	
(8) $A \cup (A \cap B) = A$	(8′) $A \cap (A \cup B) = A$
DeMorgan's Laws	
(9) $(A \cup B)^c = A^c \cap B^c$	(9′) $(A \cap B)^c = A^c \cup B^c$
Involution Law	
(10) $(A^c)^c = A$	

It is quite clear that most of these laws resemble or, in fact, are analogous of laws in basic algebra and the algebra of propositions.

4.2.2: Proof Using Previously Proven Theorems

Once a few basic laws or theorems have been established, we frequently use them to prove additional theorems. This method of proof is usually more efficient than that of proof by Definition. To illustrate, let us prove the following Corollary to the Distributive Law. The term "corollary" is used for theorems that can be proven with relative ease from previously proven theorems.

Corollary 4.2.1: A Corollary to the Distributive Law of Sets

Let A and B be sets. Then $(A \cap B) \cup (A \cap B^c) = A$.

Proof

$$\begin{aligned}(A\cap B)\cup(A\cap B^c) &= A\cap(B\cup B^c)\\ &\quad \text{Why?}\\ &= A\cap U\\ &\quad \text{Why?}\\ &= A\\ &\quad \text{Why?}\end{aligned}.$$

4.2.3: Proof Using the Indirect Method/Contradiction

The procedure one most frequently uses to prove a theorem in mathematics is the Direct Method, as illustrated in Theorem 4.1.1 and Theorem 4.1.2. Occasionally there are situations where this method is not applicable. Consider the following:

Theorem 4.2.1: An Indirect Proof in Set Theory

Let A, B, C be sets. If $A\subseteq B$ and $B\cap C=\emptyset,$ then $A\cap C=\emptyset$.

Proof

Commentary: The usual and first approach would be to assume $A\subseteq B$ and $B\cap C=\emptyset$ is true and to attempt to prove $A\cap C=\emptyset$ is true. To do this you would need to show that nothing is contained in the set $A\cap C$. Think about how you would show that something doesn't exist. It is very difficult to do directly.

The Indirect Method is much easier: If we assume the conclusion is false and we obtain a contradiction --- then the theorem must be true. This approach is on sound logical footing since it is exactly the same method of indirect proof that we discussed in Subsection 3.5.3.

Assume $A\subseteq B$ and $B\cap C=\emptyset,$ and $A\cap C\neq\emptyset$. To prove that this cannot occur, let $x\in A\cap C$.

$$\begin{aligned}x\in A\cap C &\Rightarrow x\in A \text{ and } x\in C\\ &\Rightarrow x\in B \text{ and } x\in C\\ &\Rightarrow x\in B\cap C\end{aligned}.$$

But this contradicts the second premise. Hence, the theorem is proven.

4.2.4: Exercises

In the exercises that follow it is most important that you outline the logical procedures or methods you use.

Exercise 4.2.1

a. Prove the associative law for intersection (Law $2'$) with a Venn diagram.
b. Prove DeMorgan's Law (Law 9) with a membership table.
c. Prove the Idempotent Law (Law 6) using basic definitions.

Answer

a.

b.

A	B	A^c	B^c	$A\cup B$	$(A\cup B)^c$	$A^c\cap B^c$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

The last two columns are the same so the two sets must be equal.

c.
$$\begin{aligned}x\in A\cup A &\Rightarrow (x\in A)\vee(x\in A) \quad \text{by the definition of } \cap\\ &\Rightarrow x\in A \quad \text{by the idempotent law of logic}\end{aligned}$$

Therefore, $A \cup A \subseteq A$.

$$\begin{aligned} x \in A &\Rightarrow (x \in A) \vee (x \in A) \quad \text{by conjunctive addition} \\ &\Rightarrow x \in A \cup A \end{aligned}$$

Therefore, $A \subseteq A \cup A$ and so we have $A \cup A = A$.

Exercise 4.2.2

a. Prove the Absorption Law (Law 8′) with a Venn diagram.
b. Prove the Identity Law (Law 4) with a membership table.
c. Prove the Involution Law (Law 10) using basic definitions.

Exercise 4.2.3

Prove the following using the set theory laws, as well as any other theorems proved so far.

a. $A \cup (B - A) = A \cup B$
b. $A - B = B^c - A^c$
c. $A \subseteq B, A \cap C \neq \emptyset \Rightarrow B \cap C \neq \emptyset$
d. $A \cap (B - C) = (A \cap B) - (A \cap C)$
e. $A - (B \cup C) = (A - B) \cap (A - C)$

Answer

For all parts of this exercise, a reason should be supplied for each step. We have supplied reasons only for part a and left them out of the other parts to give you further practice.

a. Here:

$$\begin{aligned} A \cup (B - A) &= A \cup (B \cap A^c) \text{ by Exercise 4.1.1 of Section 4.1} \\ &= (A \cup B) \cap (A \cup A^c) \text{ by the distributive law} \\ &= (A \cup B) \cap U \text{ by the null law} \\ &= (A \cup B) \text{ by the identity law. } \square \end{aligned}$$

b. Now:

$$\begin{aligned} A - B &= A \cap B^c \\ &= B^c \cap A \\ &= B^c \cap (A^c)^c \\ &= B^c - A^c. \end{aligned}$$

c. Select any element, $x \in A \cap C$. One such element exists since $A \cap C$ is not empty.

$$\begin{aligned} x \in A \cap C &\Rightarrow x \in A \wedge x \in C \\ &\Rightarrow x \in B \wedge x \in C \\ &\Rightarrow x \in B \cap C \\ &\Rightarrow B \cap C \neq \emptyset. \quad \square \end{aligned}$$

d. We'll start with the right side of this identity.

$$\begin{aligned} (A \cap B) - (A \cap C) &= A \cap B \cap (A \cap C)^c \\ &= A \cap B \cap (A^c \cup C^c) \\ &= (A \cap B \cap A^c) \cup (A \cap B \cap C^c) \\ &= \emptyset \cup A \cap (B \cap C^c) \\ &= A \cap (B - C). \quad \square \end{aligned}$$

e. Here:

$$\begin{aligned} A-(B\cup C) &= A\cap(B\cup C)^c \\ &= A\cap(B^c\cap C^c) \\ &= (A\cap B^c)\cap(A\cap C^c) \\ &= (A-B)\cap(A-C). \quad \square \end{aligned}$$

Exercise 4.2.4

Use previously proven theorems to prove the following.

a. $A\cap(B\cap C)^c=(A\cap B^c)\cup(A\cap C^c)$
b. $A\cap(B\cap(A\cap B)^c)=\emptyset$
c. $(A\cap B)\cup B^c=A\cup B^c$
d. $A\cup(B-C)=(A\cup B)-(C-A)$.

Exercise 4.2.5: Hierarchy of Set Operations

The rules that determine the order of evaluation in a set expression that involves more than one operation are similar to the rules for logic. In the absence of parentheses, complementations are done first, intersections second, and unions third. Parentheses are used to override this order. If the same operation appears two or more consecutive times, evaluate from left to right. In what order are the following expressions performed?

a. $A\cup B^c\cap C$.
b. $A\cap B\cup C\cap B$.
c. $A\cup B\cup C^c$

Answer

a. $A\cup((B^c)\cap C)$
b. $(A\cap B)\cup(C\cap B)$
c. $(A\cup B)\cup(C^c)$

Exercise 4.2.6

There are several ways that we can use to format the proofs in this chapter. One that should be familiar to you from Chapter 3 is illustrated with the following alternate proof of part (a) in Theorem 4.1.1:

Table 4.2.2: An alternate format for the proof of Theorem 4.1.1

(1)	$x\in A\cap(B\cup C)$	Premise
(2)	$(x\in A)\wedge(x\in B\cup C)$	(1), definition of intersection
(3)	$(x\in A)\wedge((x\in B)\vee(x\in C))$	(2), definition of union
(4)	$(x\in A)\wedge(x\in B)\vee(x\in A)\wedge(x\in C)$	(3), distribute $\wedge$ over $\vee$
(5)	$(x\in A\cap B)\vee(x\in A\cap C)$	(4), definition of intersection
(6)	$x\in(A\cap B)\cup(A\cap C)$	(5), definition of union ■

Prove part (b) of Theorem 4.1.2 and Theorem 4.2.1 using this format.

This page titled 4.2: Laws of Set Theory is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

4.3: Minsets

4.3.1: Definition of Minsets

Let B_1 and B_2 be subsets of a set A. Notice that the Venn diagram of Figure 4.3.1 is naturally partitioned into the subsets A_1, A_2, A_3, and A_4. Further we observe that A_1, A_2, A_3, and A_4 can be described in terms of B_1 and B_2 as follows:

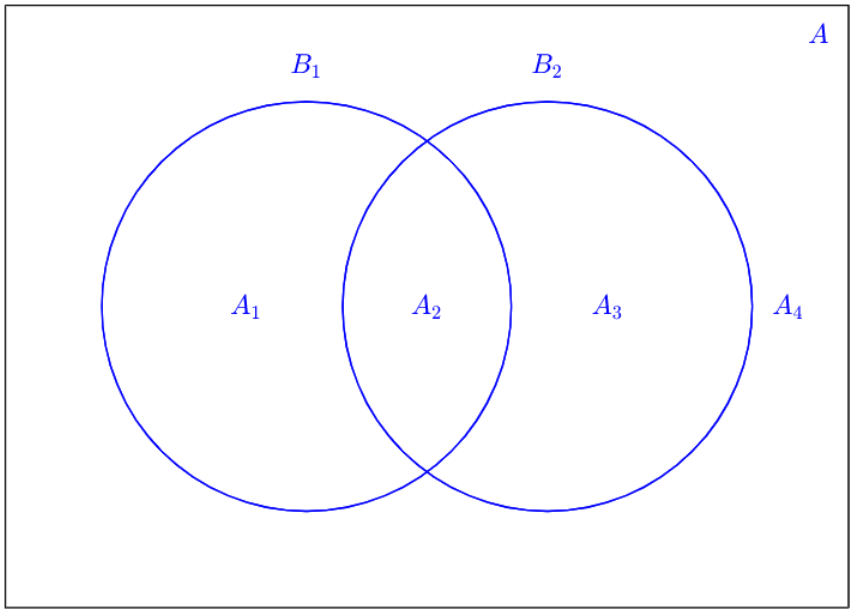


Figure 4.3.1: Venn Diagram of Minsets

Table 4.3.1: Minsets generated by two sets

$A_1 = B_1 \cap B_2^c$
$A_2 = B_1 \cap B_2$
$A_3 = B_1^C \cap B_2$
$A_4 = B_1^c \cap B_2^c$

Each A_i is called a minset generated by B_1 and B_2. We note that each minset is formed by taking the intersection of two sets where each may be either B_k or its complement, B_k^c. Note also, given two sets, there are $2^2 = 4$ minsets.

Minsets are occasionally called *minterms*.

The reader should note that if we apply all possible combinations of the operations intersection, union, and complementation to the sets B_1 and B_2 of Figure 4.3.1, the smallest sets generated will be exactly the minsets, the minimum sets. Hence the derivation of the term minset.

Next, consider the Venn diagram containing three sets, B_1, B_2, and B_3. Draw it right now and count the regions! What are the minsets generated by B_1, B_2, and B_3? How many are there? Following the procedures outlined above, we note that the following are three of the $2^3 = 8$ minsets. What are the others?

Table 4.3.2: Three of the minsets generated by B_1, B_2, and B_3

$B_1 \cap B_2 \cap B_3^c$
$B_1 \cap B_2^c \cap B_3$
$B_1 \cap B_2^c \cap B_3^c$

Definition 4.3.1: Minset

Let $\{B_1, B_2, \ldots, B_n\}$ be a set of subsets of set A. Sets of the form $D_1 \cap D_2 \cap \cdots \cap D_n$, where each D_i may be either B_i or B_i^c, is called a minset generated by B_1, B_2,... and B_n.

Example 4.3.1: A Concrete Example of Some Minsets

Consider the following concrete example. Let $A = \{1, 2, 3, 4, 5, 6\}$with subsets $B_1 = \{1, 3, 5\}$ and $B_2 = \{1, 2, 3\}$. How can we use set operations applied to and produce a partition of A? As a first attempt, we might try these three sets:

Table 4.3.3

$B_1 \cap B_2 = \{1, 3\}$
$B_1^c = \{2, 4, 6\}$
$B_2^c = \{4, 5, 6\}.$

We have produced all elements of A but we have 4 and 6 repeated in two sets. In place of B_1^c and B_2^c, let's try $B_1^c \cap B_2$ and $B_1 \cap B_2^c$, respectively:

Table 4.3.4

$B_1^c \cap B_2 = \{2\}$ and
$B_1 \cap B_2^c = \{5\}.$

We have now produced the elements 1, 2, 3, and 5 using $B_1 \cap B_2$, $B_1^c \cap B_2$ and $B_1 \cap B_2^c$ yet we have not listed the elements 4 and 6. Most ways that we could combine B_1 and B_2 such as $B_1 \cup B_2$ or $B_1 \cup B_2^c$ will produce duplications of listed elements and will not produce both 4 and 6. However we note that $B_1^c \cap B_2^c = \{4, 6\}$, exactly the elements we need.

After more experimenting, we might reach a conclusion that each element of A appears exactly once in one of the four minsets $B_1 \cap B_2$, $B_1^c \cap B_2$, $B_1 \cap B_2^c$ and $B_1^c \cap B_2^c$. Hence, we have a partition of A. In fact this is the finest partition of A in that all other partitions we could generate consist of selected unions of these minsets.

At this point, we might ask and be able to answer the question "How many different subsets of our universe can we generate from B_1 and B_2?" The answer is $2^{\text{number of nonempty minsets}}$, which is $2^4 = 16$ in this case. Notice that in general, it would be impossible to find two sets from which we could generate all subsets of $A = \{1, 2, 3, 4, 5, 6\}$since there will never be more than four nonempty minsets. If we allowed ourselves three subsets and tried to generat all sets from them, then the number of minsets would be $2^3 = 8$. With only six elements in A, there could be six minsets, each containing a single element. In that case we could generate the whole power set of A.

4.3.2: Properties of Minsets

Theorem 4.3.1: Minset Partition Theorem

Let A be a set and let B_1, B_2 ... , B_n be subsets of A. The set of nonempty minsets generated by B_1, B_2 ... , B_n is a partition of A.

Proof

The proof of this theorem is left to the reader.

One of the most significant facts about minsets is that any subset of A that can be obtained from B_1, B_2 ... , B_n, using the standard set operations can be obtained in a standard form by taking the union of selected minsets.

Definition 4.3.2: Minset Normal Form

A set is said to be in minset normal form when it is expressed as the union of zero or more distinct nonempty minsets.

Note 4.3.1

- The union of zero sets is the empty set, $\emptyset$.
- Minset normal form is also called ***canonical form***.

Example 4.3.2: Another Concrete Example of Minsets

Let $U = \{-2, -1, 0, 1, 2\}$, $B_1 = \{0, 1, 2\}$, and $B_2 = \{0, 2\}$. Then

Table 4.3.5

$B_1 \cap B_2 = \{0, 2\}$
$B_1^c \cap B_2 = \emptyset$
$B_1 \cap B_2^c = \{1\}$
$B_1^c \cap B_2^c = \{-2, -1\}$

In this case, there are only three nonempty minsets, producing the partition $\{\{0, 2\}, \{1\}, \{-2, -1\}\}$ An example of a set that could not be produced from just B_1 and B_2 is the set of even elements of U, $\{-2, 0, 2\}$. This is because -2 and -1 cannot be separated. They are in the same minset and any union of minsets either includes or excludes them both. In general, there are $2^3 = 8$ different minset normal forms because there are three nonempty minsets. This means that only 8 of the $2^5 = 32$ subsets of U could be generated from any two sets B_1 and B_2.

4.3.3: Exercises

Exercise 4.3.1

Consider the subsets $A = \{1, 7, 8\}$, $B = \{1, 6, 9, 10\}$, and $C = \{1, 9, 10\}$, where $U = \{1, 2, \ldots, 10\}$.

a. List the nonempty minsets generated by A, B, and C.
b. How many elements of the power set of U can be generated by A, B, and C? Compare this number with $|\mathcal{P}(U)|$. Give an example of one subset that cannot be generated by A, B, and C.

Answer

a. $\{1\}, \{2, 3, 4, 5\}, \{6\}, \{7, 8\}, \{9, 10\}$
b. 2^5 , as compared with 2^{10}. $\{1, 2\}$ is one of the 992 sets that can't be generated.

Exercise 4.3.2

a. Partition $\{1, 2, \ldots . 9\}$ into the minsets generated by $B_1 = \{5, 6, 7\}$, $B_2 = \{2, 4, 5, 9\}$, and $B_3 = \{3, 4, 5, 6, 8, 9\}$.
b. How many different subsets of $\{1, 2, \ldots, 9\}$ can you create using B_1, B_2, and B_3 with the standard set operations?
c. Do there exist subsets C_1, C_2, C_3 whose minsets will generate every subset of $\{1, 2, \ldots, 9\}$?

Exercise 4.3.3

Partition the set of strings of 0's and 1's of length two or less, using the minsets generated by $B_1 = \{s \mid s \text{ has length } 2\}$, and $B_2 = \{s \mid s \text{ starts with a } 0\}$.

Answer

$B_1 = \{00, 01, 10, 11\}$ and $B_2 = \{0, 00, 01\}$ generate minsets $\{00, 01\}, \{0\}, \{10, 11\}$, and $\{\lambda, 1\}$. Note: λ is the null string, which has length zero.

Exercise 4.3.4

Let B_1, B_2, and B_3 be subsets of a universal set U,

a. Symbolically list all minsets generated by B_1, B_2, and B_3.
b. Illustrate with a Venn diagram all minsets obtained in part (a).
c. Express the following sets in minset normal form: B_1^c, $B_1 \cap B_2$, $B_1 \cup B_2^c$.

Exercise 4.3.5

a. Partition $A = \{0, 1, 2, 3, 4, 5\}$with the minsets generated by $B_1 = \{0, 2, 4\}$ and $B_2 = \{1, 5\}$.
b. How many different subsets of A can you generate from B_1 and B_2?

Answer

a. $B_1 \cap B_2 = \emptyset,\ B_1 \cap B_2^c = \{0, 2, 4\}, B_1^c \cap B_2 = \{1, 5\}, B_1^c \cap B_2^c = \{3\}$
b. 2^3, since there are 3 nonempty minsets.

Exercise 4.3.6

If $\{B_1, B_2, \ldots, B_n\}$ is a partition of A, how many minsets are generated by $B_1, B_2, \ldots, B_n$?

Exercise 4.3.7

Prove Theorem 4.3.1.

Answer

Let $a \in A$. For each i, $a \in B_i$, or $a \in {B_i}^c$, since $B_i \cup {B_i}^c = A$ by the complement law. Let $D_i = B_i$ if $a \in B_i$, and $D_i = {B_i}^c$ otherwise. Since a is in each D_i, it must be in the minset $D_1 \cap D_2 \cdots \cap D_n$. Now consider two different minsets $M_1 = D_1 \cap D_2 \cdots \cap D_n$, and $M_2 = G_1 \cap G_2 \cdots \cap G_n$, where each D_i and G_i is either B_i or ${B_i}^c$. Since these minsets are not equal, $D_i \neq G_i$, for some i. Therefore, $M_1 \cap M_2 = D_1 \cap D_2 \cdots \cap D_n \cap G_1 \cap G_2 \cdots \cap G_n = \emptyset$, since two of the sets in the intersection are disjoint. Since every element of A is in a minset and the minsets are disjoint, the nonempty minsets must form a partition of A. □

Exercise 4.3.8

Let S be a finite set of n elements. Let B_i, $i = 1, 2, \ldots, k$ be nonempty subsets of S. There are 2^{2^k} minset normal forms generated by the k subsets. The number of subsets of S is 2^n. Since we can make $2^{2^k} > 2^n$ by choosing $k \geq \log_2 n$, it is clear that two distinct minset normal-form expressions do not always equal distinct subsets of S. Even for $k < \log_2 n$, it may happen that two distinct minset normal-form expressions equal the same subset of S. Determine necessary and sufficient conditions for distinct normal-form expressions to equal distinct subsets of S.

This page titled 4.3: Minsets is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

4.4: The Duality Principle

4.4.1:

In Section 4.2, we observed that each of the Table 4.2.1 labeled 1 through 9 had an analogue $1'$ through $9'$. We notice that each of the laws in one column can be obtained from the corresponding law in the other column by replacing $\cup$ by $\cap$, $\cap$ by $\cup$, $\emptyset$ by U, U by $\emptyset$, and leaving the complement unchanged.

Definition 4.4.1: Duality Principle for Sets

Let S be any identity involving sets and the operations complement, intersection and union. If $S*$ is obtained from S by making the substitutions $\cup \to \cap$, $\cap \to \cup$, $\emptyset \to U$, and $U \to \emptyset$, then the statement $S*$ is also true and it is called the dual of the statement S.

Example 4.4.1: Example of a Dual

The dual of $(A \cap B) \cup (A \cap B^c) = A$ is $(A \cup B) \cap (A \cup B^c) = A$.

One should not underestimate the importance of this concept. It gives us a whole second set of identities, theorems, and concepts. For example, we can consider the dual of *minsets* and *minset normal form* to obtain what is called *maxsets* and *maxset normal form*.

4.4.2: Exercises

Exercise 4.4.1

State the dual of each of the following:

a. $A \cup (B \cap A) = A$.
b. $A \cup ((B^c \cup A) \cap B)^{\,c} = U$
c. $(A \cup B^c)^{\,c} \cap B = A^c \cap B$

Answer

a. $A \cap (B \cup A) = A$
b. $A \cap ((B^c \cap A) \cup B)^{\,c} = \emptyset$
c. $(A \cap B^c)^{\,c} \cup B = A^c \cup B$

Exercise 4.4.2

Examine Table 3.4.2 and then write a description of the principle of duality for logic.

Exercise 4.4.3

Write the dual of each of the following:

a. $p \vee \neg((\neg q \vee p) \wedge q) \Leftrightarrow 1$
b. $(\neg(p \wedge (\neg q))) \vee q \Leftrightarrow (\neg p \vee q)$.

Answer

a. $(p \wedge \neg(\neg q \wedge p) \vee q) \Leftrightarrow 0$
b. $(\neg(p \vee (\neg q))) \wedge q \Leftrightarrow ((\neg p) \wedge q)$

Exercise 4.4.4

Use the principle of duality and the definition of minset to write the definition of maxset.

Exercise 4.4.5

Let $A = \{1, 2, 3, 4, 5, 6\}$and let $B_1 = \{1, 3, 5\}$ and $B_2 = \{1, 2, 3\}$.

a. Find the maxsets generated by B_1 and B_2. Note the set of maxsets does not constitute a partition of A. Can you explain why?
b. Write out the definition of maxset normal form.
c. Repeat Exercise 4.3.4 for maxsets.

Answer

The maxsets are:

- $B_1 \cup B_2 = \{1, 2, 3, 5\}$
- $B_1 \cup {B_2}^c = \{1, 3, 4, 5, 6\}$
- ${B_1}^c \cup B_2 = \{1, 2, 3, 4, 6\}$
- ${B_1}^c \cup {B_2}^c = \{2, 4, 5, 6\}$

They do not form a partition of A since it is not true that the intersection of any two of them is empty. A set is said to be in *maxset normal form* when it is expressed as the intersection of distinct nonempty maxsets or it is the universal set U.

Exercise 4.4.6

What is the dual of the expression in Exercise 4.1.5?

This page titled 4.4: The Duality Principle is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

5: Introduction to Matrix Algebra

> ***diagonal matrix***
>
> *"It's totally right what you say, Trix,*
> *That in a* ***diagonal matrix***
> *You'll find," Al confirms,*
> *"Off-diagonal terms*
> *Are all zero. Now bug off and play, Trix!"*
>
> *Bob Egg, The Omnificent English Dictionary in Limerick Form*

The purpose of this chapter is to introduce you to matrix algebra, which has many applications. You are already familiar with several algebras: elementary algebra, the algebra of logic, the algebra of sets. We hope that as you studied the algebra of logic and the algebra of sets, you compared them with elementary algebra and noted that the basic laws of each are similar. We will see that matrix algebra is also similar. As in previous discussions, we begin by defining the objects in question and the basic operations.

5.1: Basic Definitions and Operations
5.2: Special Types of Matrices
5.3: Laws of Matrix Algebra
5.4: Matrix Oddities

This page titled 5: Introduction to Matrix Algebra is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

5.1: Basic Definitions and Operations

5.1.1: Matrix Order and Equality

Definition 5.1.1: Matrix

A matrix is a rectangular array of elements of the form

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix}$$

A convenient way of describing a matrix in general is to designate each entry via its position in the array. That is, the entry a_{34} is the entry in the third row and fourth column of the matrix A. Depending on the situation, we will decide in advance to which set the entries in a matrix will belong. For example, we might assume that each entry a_{ij} ($1 \leq i \leq m,\ 1 \leq j \leq n$) is a real number. In that case we would use $M_{m\times n}(\mathbb{R})$ to stand for the set of all m by n matrices whose entries are real numbers. If we decide that the entries in a matrix must come from a set S, we use $M_{m\times n}(S)$ to denote all such matrices.

Definition 5.1.2: The Order of a Matrix

A matrix A that has m rows and n columns is called an $m \times n$ (read "m by n") matrix, and is said to have order $m \times n$.

Since it is rather cumbersome to write out the large rectangular array above each time we wish to discuss the generalized form of a matrix, it is common practice to replace the above by $A = (a_{ij})$. In general, matrices are often given names that are capital letters and the corresponding lower case letter is used for individual entries. For example the entry in the third row, second column of a matrix called C would be c_{32}.

Example 5.1.1: Orders of Some Matrices

$A = \begin{pmatrix} 2 & 3 \\ 0 & -5 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 15 \end{pmatrix}$, and $D = \begin{pmatrix} 1 & 2 & 5 \\ 6 & -2 & 3 \\ 4 & 2 & 8 \end{pmatrix}$ are 2×2, 3×1, and 3×3 matrices, respectively.

Since we now understand what a matrix looks like, we are in a position to investigate the operations of matrix algebra for which users have found the most applications.

First we ask ourselves: Is the matrix $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ equal to the matrix $B = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}$? No, they are not because the corresponding entries in the second row, second column of the two matrices are not equal.

Next, is $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ equal to $B = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$? No, although the corresponding entries in the first two columns are identical, B doesn't have a third column to compare to that of A. We formalize these observations in the following definition.

Definition 5.1.3: Equality of Matrices

A matrix A is said to be equal to matrix B (written $A = B$) if and only if:

1. A and B have the same order, and
2. all corresponding entries are equal: that is, $a_{ij} = b_{ij}$ for all appropriate i and j.

5.1.2: Matrix Addition and Scalar Multiplication

The first two operations we introduce are very natural and are not likely cause much confusion. The first is matrix addition. It seems natural that if $A = \begin{pmatrix} 1 & 0 \\ 2 & -1 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & 4 \\ -5 & 2 \end{pmatrix}$, then

$$A + B = \begin{pmatrix} 1+3 & 0+4 \\ 2-5 & -1+2 \end{pmatrix} = \begin{pmatrix} 4 & 4 \\ -3 & 1 \end{pmatrix}.$$

However, if $A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & 0 \\ 2 & 8 \end{pmatrix}$, is there a natural way to add them to give us $A + B$? No, the orders of the two matrices must be identical.

Definition 5.1.4: Matrix Addition

Let A and B be $m \times n$ matrices. Then $A + B$ is an $m \times n$ matrix where $(A+B)_{ij} = a_{ij} + b_{ij}$ (read "The ith jth entry of the matrix $A + B$ is obtained by adding the ith jth entry of A to the ith jth entry of B"). If the orders of A and B are not identical, $A + B$ is not defined.

In short, $A + B$ is defined if and only if A and B are of the same order.

Another frequently used operation is that of multiplying a matrix by a number, commonly called a scalar in this context. Scalars normally come from the same set as the entries in a matrix. For example, if $A \in M_{m\times n}(\mathbb{R})$, a scalar can be any real number.

Example 5.1.2: A Scalar Product

If $c = 3$ and if $A = \begin{pmatrix} 1 & -2 \\ 3 & 5 \end{pmatrix}$ and we wish to find cA, it seems natural to multiply each entry of A by 3 so that $3A = \begin{pmatrix} 3 & -6 \\ 9 & 15 \end{pmatrix}$, and this is precisely the way scalar multiplication is defined.

Definition 5.1.5: Scalar Multiplication

Let A be an $m \times n$ matrix and c a scalar. Then cA is the $m \times n$ matrix obtained by multiplying c times each entry of A; that is $(cA)_{ij} = ca_{ij}$.

5.1.3: Matrix Multiplication

A definition that is more awkward to motivate is the product of two matrices. See Exercise 5.1.8 for an attempt to do so. In time, the reader will see that the following definition of the product of matrices will be very useful, and will provide an algebraic system that is quite similar to elementary algebra.

Here is a video introduction to matrix multiplication.

Definition 5.1.6: Matrix Multiplication

Let A be an $m \times n$ matrix and let B be an $n \times p$ matrix. The product of A and B, denoted by AB, is an $m \times p$ matrix whose ith row jth column entry is

$$\begin{aligned}(AB)_{ij} &= a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} \\ &= \sum_{k=1}^{n} a_{ik}b_{kj}\end{aligned}$$

for $1 \le i \le m$ and $1 \le j \le p$.

The mechanics of computing one entry in the product of two matrices is illustrated in Figure 5.1.1.

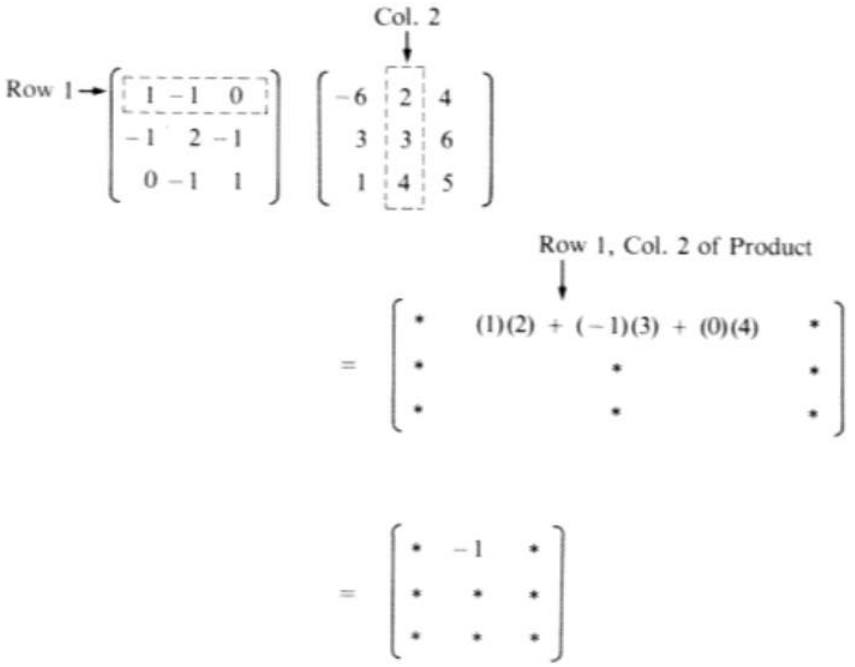


Figure 5.1.1: Computation of one entry in the product of two 3 by 3 matrices

The computation of a product can take a considerable amount of time in comparison to the time required to add two matrices. Suppose that A and B are $n \times n$ matrices; then $(AB)_{ij}$ is determined performing n multiplications and $n-1$ additions. The full product takes n^3 multiplications and $n^3 - n^2$ additions. This compares with n^2 additions for the sum of two $n \times n$ matrices. The product of two 10 by 10 matrices will require 1,000 multiplications and 900 additions, clearly a job that you would assign to a computer. The sum of two matrices requires a more modest 100 additions. This analysis is based on the assumption that matrix multiplication will be done using the formula that is given in the definition. There are more advanced methods that, in theory, reduce operation counts. For example, Strassen's algorithm (https://en.wikipedia.org/wiki/Strassen_algorithm) computes the product of two n by n matrices in $7 \cdot 7^{\log_2 n} - 6 \cdot 4^{\log_2 n} \approx 7n^{2.808}$ operations. There are practical issues involved in actually using the algorithm in many situations. For example, round-off error can be more of a problem than with the standard formula.

Example 5.1.3: A Matrix Product

Let $A = \begin{pmatrix} 1 & 0 \\ 3 & 2 \\ -5 & 1 \end{pmatrix}$, a 3×2 matrix, and let $B = \begin{pmatrix} 6 \\ 1 \end{pmatrix}$, a 2×1 matrix. Then AB is a 3×1 matrix:

$$AB = \begin{pmatrix} 1 & 0 \\ 3 & 2 \\ -5 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot 6 + 0 \cdot 1 \\ 3 \cdot 6 + 2 \cdot 1 \\ -5 \cdot 6 + 1 \cdot 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 20 \\ -29 \end{pmatrix}$$

Remarks:

1. The product AB is defined only if A is an $m \times n$ matrix and B is an $n \times p$ matrix; that is, the two "inner" numbers must be equal. Furthermore, the order of the product matrix AB is the "outer" numbers, in this case $m \times p$.
2. It is wise to first determine the order of a product matrix. For example, if A is a 3×2 matrix and B is a 2×2 matrix, then AB is a 3×2 matrix of the form

$$AB = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix}$$

Then to obtain, for example, c_{31}, we multiply corresponding entries in the third row of A times the first column of B and add the results.

Example 5.1.4: Multiplication with a Diagonal Matrix

Let $A = \begin{pmatrix} -1 & 0 \\ 0 & 3 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & 10 \\ 2 & 1 \end{pmatrix}$. Then $AB = \begin{pmatrix} -1\cdot 3+0\cdot 2 & -1\cdot 10+0\cdot 1 \\ 0\cdot 3+3\cdot 2 & 0\cdot 10+3\cdot 1 \end{pmatrix} = \begin{pmatrix} -3 & -10 \\ 6 & 3 \end{pmatrix}$

The net effect is to multiply the first row of B by -1 and the second row of B by 3.

Note: $BA = \begin{pmatrix} -3 & 30 \\ -2 & 3 \end{pmatrix} \neq AB$. The columns of B are multiplied by -1 and 3 when the order is switched.

Remarks:

- An $n \times n$ matrix is called a *square matrix*.
- If A is a square matrix, AA is defined and is denoted by A^2, and $AAA = A^3$, etc.
- The $m \times n$ matrices whose entries are all 0 are denoted by $\mathbf{0}_{m\times n}$, or simply $\mathbf{0}$, when no confusion arises regarding the order.

5.1.4: Exercises

Exercise 5.1.1

Let $A = \begin{pmatrix} 1 & -1 \\ 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 0 & 1 \\ 3 & -5 \end{pmatrix}$, and $C = \begin{pmatrix} 0 & 1 & -1 \\ 3 & -2 & 2 \end{pmatrix}$

a. Compute AB and BA.
b. Compute $A+B$ and $B+A$.
c. If $c = 3$, show that $c(A+B) = cA+cB$.
d. Show that $(AB)C = A(BC)$.
e. Compute A^2C.
f. Compute $B+\mathbf{0}$.
g. Compute $A\mathbf{0}_{2\times 2}$ and $\mathbf{0}_{2\times 2}A$, where $\mathbf{0}_{2\times 2}$ is the 2×2 zero matrix.
h. Compute $0A$, where 0 is the real number (scalar) zero.
i. Let $c = 2$ and $d = 3$. Show that $(c+d)A = cA+dA$.

Answer

For parts c, d and i of this exercise, only a verification is needed. Here, we supply the result that will appear on both sides of the equality.

a. $AB = \begin{pmatrix} -3 & 6 \\ 9 & -13 \end{pmatrix}$ $BA = \begin{pmatrix} 2 & 3 \\ -7 & -18 \end{pmatrix}$
b. $\begin{pmatrix} 1 & 0 \\ 5 & -2 \end{pmatrix}$
c. $\begin{pmatrix} 3 & 0 \\ 15 & -6 \end{pmatrix}$
d. $\begin{pmatrix} 18 & -15 & 15 \\ -39 & 35 & -35 \end{pmatrix}$
e. $\begin{pmatrix} -12 & 7 & -7 \\ 21 & -6 & 6 \end{pmatrix}$
f. $B+0 = B$
g. $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
h. $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

i. $\begin{pmatrix} 5 & -5 \\ 10 & 15 \end{pmatrix}$

Exercise 5.1.2

Let $A = \begin{pmatrix} 1 & 0 & 2 \\ 2 & -1 & 5 \\ 3 & 2 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 0 & 2 & 3 \\ 1 & 1 & 2 \\ -1 & 3 & -2 \end{pmatrix}$, and $C = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 4 & 0 & 1 & 1 \\ 3 & -1 & 4 & 1 \end{pmatrix}$ Compute, if possible;

a. $A - B$
b. AB
c. $AC - BC$
d. $A(BC)$
e. $CA - CB$
f. $C \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$

Exercise 5.1.3

Let $A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$. Find a matrix B such that $AB = I$ and $BA = I$, where $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Answer

$\begin{pmatrix} 1/2 & 0 \\ 0 & 1/3 \end{pmatrix}$

Exercise 5.1.4

Find AI and BI where I is as in Exercise 5.1.3, where $A = \begin{pmatrix} 1 & 8 \\ 9 & 5 \end{pmatrix}$ and $B = \begin{pmatrix} -2 & 3 \\ 5 & -7 \end{pmatrix}$. What do you notice?

Exercise 5.1.5

Find A^3 if $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$. What is A^{15} equal to?

Answer

$A^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 27 \end{pmatrix}$ $A^{15} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 32768 & 0 \\ 0 & 0 & 14348907 \end{pmatrix}$

Exercise 5.1.6

a. Determine I^2 and I^3 if $I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.
b. What is I^n equal to for any $n \geq 1$?
c. Prove your answer to part (b) by induction.

Exercise 5.1.7

a. If

$$A=\begin{pmatrix}2 & 1\\ 1 & -1\end{pmatrix}, X=\begin{pmatrix}x_1\\ x_2\end{pmatrix}, \text{ and } B=\begin{pmatrix}3\\ 1\end{pmatrix},$$

show that $AX=B$ is a way of expressing the system $\begin{matrix}2x_1+x_2=3\\ x_1-x_2=1\end{matrix}$ using matrices.

b. Express the following systems of equations using matrices:

i. $\begin{matrix}2x_1-x_2=4\\ x_1+x_2=0\\ x_1+x_2+2x_3=1\end{matrix}$

ii. $\begin{matrix}x_1+2x_2-x_3=-1\\ x_1+3x_2+x_3=5\\ x_1+x_2=3\end{matrix}$

iii. $\begin{matrix}x_2=5\\ x_1+3x_3=6\end{matrix}$

Answer

a. $Ax=\begin{pmatrix}2x_1+1x_2\\ 1x_1-1x_2\end{pmatrix}$ equals $\begin{pmatrix}3\\ 1\end{pmatrix}$ if and only if both of the equalities $2x_1+x_2=3$ and $x_1-x_2=1$ are true.

b. (i) $A=\begin{pmatrix}2 & -1\\ 1 & 1\end{pmatrix}$ $x=\begin{pmatrix}x_1\\ x_2\end{pmatrix}$ $B=\begin{pmatrix}4\\ 0\end{pmatrix}$

c. $A=\begin{pmatrix}1 & 1 & 2\\ 1 & 2 & -1\\ 1 & 3 & 1\end{pmatrix}$ $x=\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}$ $B=\begin{pmatrix}1\\ -1\\ 5\end{pmatrix}$

d. $A=\begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 0\\ 1 & 0 & 3\end{pmatrix}$ $x=\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}$ $B=\begin{pmatrix}3\\ 5\\ 6\end{pmatrix}$

Exercise 5.1.8

In this exercise, we propose to show how matrix multiplication is a natural operation. Suppose a bakery produces bread, cakes and pies every weekday, Monday through Friday. Based on past sales history, the bakery produces various numbers of each product each day, summarized in the 5×3 matrix D. It should be noted that the order could be described as "number of days by number of products." For example, on Wednesday (the third day) the number of cakes (second product in our list) that are produced is $d_{3,2}=4$.

$$D=\begin{pmatrix}25 & 5 & 5\\ 14 & 5 & 8\\ 20 & 4 & 15\\ 18 & 5 & 7\\ 35 & 10 & 9\end{pmatrix}$$

The main ingredients of these products are flour, sugar and eggs. We assume that other ingredients are always in ample supply, but we need to be sure to have the three main ones available. For each of the three products, The amount of each ingredient that is needed is summarized in the 3×3, or "number of products by number of ingredients" matrix P. For example, to bake a cake (second product) we need $P_{2,1}=1.5$ cups of flour (first ingredient). Regarding units: flour and sugar are given in cups per unit of each product, while eggs are given in individual eggs per unit of each product.

$$P = \begin{pmatrix} 2 & 0.5 & 0 \\ 1.5 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

These amounts are “made up”, so don't used them to do your own baking!

a. How many cups of flour will the bakery need every Monday? Pay close attention to how you compute your answer and the units of each number.
b. How many eggs will the bakery need every Wednesday?
c. Compute the matrix product DP. What do you notice?
d. Suppose the costs of ingredients are $\$0.12$ for a cup of flour, $\$0.15$ for a cup of sugar and $\$0.19$ for one egg. How can this information be put into a matrix that can meaningfully be multiplied by one of the other matrices in this problem?

This page titled 5.1: Basic Definitions and Operations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

5.2: Special Types of Matrices

5.2.1: Diagonal Matrices

We have already investigated, in exercises in the previous section, one special type of matrix. That was the zero matrix, and found that it behaves in matrix algebra in an analogous fashion to the real number 0; that is, as the additive identity. We will now investigate the properties of a few other special matrices.

Definition 5.2.1: Diagonal Matrix

A square matrix D is called a diagonal matrix if $d_{ij} = 0$ whenever $i \neq j$.

Example 5.2.1: Some Diagonal Matrices

$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$, $B = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -5 \end{pmatrix}$, and $I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ are all diagonal matrices.

5.2.2: Identity Matrix and Matrix Inverses

In the example above, the 3×3 diagonal matrix I whose diagonal entries are all 1's has the distinctive property that for any other 3×3 matrix A we have $AI = IA = A$. For example:

Example 5.2.2: Multiplying by the Identity Matrix

If $A = \begin{pmatrix} 1 & 2 & 5 \\ 6 & 7 & -2 \\ 3 & -3 & 0 \end{pmatrix}$, then $AI = \begin{pmatrix} 1 & 2 & 5 \\ 6 & 7 & -2 \\ 3 & -3 & 0 \end{pmatrix}$ and $IA = \begin{pmatrix} 1 & 2 & 5 \\ 6 & 7 & -2 \\ 3 & -3 & 0 \end{pmatrix}$.

In other words, the matrix I behaves in matrix algebra like the real number 1; that is, as a multiplicative identity. In matrix algebra, the matrix I is called simply the identity matrix. Convince yourself that if A is any $n \times n$ matrix $AI = IA = A$.

Definition 5.2.2: Identity Matrix

The $n \times n$ diagonal matrix I_n whose diagonal components are all 1's is called the identity matrix. If the context is clear, we simply use I.

In the set of real numbers we recall that, given a nonzero real number x, there exists a real number y such that $xy = yx = 1$. We know that real numbers commute under multiplication so that the two equations can be summarized as $xy = 1$. Further we know that $y = x^{-1} = \frac{1}{x}$. Do we have an analogous situation in $M_{n\times n}(\mathbb{R})$? Can we define the multiplicative inverse of an $n \times n$ matrix A? It seems natural to imitate the definition of multiplicative inverse in the real numbers.

Definition 5.2.3: Matrix Inverse

Let A be an $n \times n$ matrix. If there exists an $n \times n$ matrix B such that $AB = BA = I$, then B is a multiplicative inverse of A (called simply an inverse of A) and is denoted by A^{-1}

When we are doing computations involving matrices, it would be helpful to know that when we find A^{-1}, the answer we obtain is the only inverse of the given matrix. This would let us refer to *the* inverse of a matrix. We refrained from saying that in the definition, but the theorem below justifies it.

Remark: Those unfamiliar with the laws of matrix algebra should return to the following proof after they have familiarized themselves with the Laws of Matrix Algebra in Section 5.5.

Theorem 5.2.1: Inverses are Unique

The inverse of an $n \times n$ matrix A, when it exists, is unique.

Proof

Let A be an $n \times n$ matrix. Assume to the contrary, that A has two (different) inverses, say B and C. Then

$$\begin{aligned} B &= BI && \text{Identity property of } I \\ &= B(AC) && \text{Assumption that } C \text{ is an inverse of } A \\ &= (BA)C && \text{Associativity of matrix multiplication} \\ &= IC && \text{Assumption that } B \text{ is an inverse of } A \\ &= C && \text{Identity property of } I \end{aligned}$$

Let $A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$. What is A^{-1} ? Without too much difficulty, by trial and error, we determine that $A^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}$. This might lead us to guess that the inverse is found by taking the reciprocal of all nonzero entries of a matrix. Alas, it isn't that easy!

If $A = \begin{pmatrix} 1 & 2 \\ -3 & 5 \end{pmatrix}$, the "reciprocal rule" would tell us that the inverse of A is $B = \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{-1}{3} & \frac{1}{5} \end{pmatrix}$. Try computing AB and you will see that you don't get the identity matrix. So, what *is* A^{-1}? In order to understand more completely the notion of the inverse of a matrix, it would be beneficial to have a formula that would enable us to compute the inverse of at least a 2×2 matrix. To do this, we introduce the definition of the determinant of a 2×2 matrix.

Definition 5.2.4: Determinant of a 2 by 2 Matrix

Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. The determinant of A is the number $\det A = ad - bc$.

In addition to $\det A$, common notation for the determinant of matrix A is $|A|$. This is particularly common when writing out the whole matrix, which case we would write $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ for the determinant of the general 2×2 matrix.

Example 5.2.3: Some Determinants of Two by Two Matrices

If $A = \begin{pmatrix} 1 & 2 \\ -3 & 5 \end{pmatrix}$ then $\det A = 1 \cdot 5 - 2 \cdot (-3) = 11$. If $B = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$ then $\det B = 1 \cdot 4 - 2 \cdot 2 = 0$.

Theorem 5.2.2: Inverse of 2 by 2 Matrix

Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. If $\det A \neq 0$, then $A^{-1} = \frac{1}{\det A} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.

Proof

See Exercise 5.2.4at the end of this section.

Example 5.2.4: Finding Inverses

Can we find the inverses of the matrices in Example 5.2.3? If $A = \begin{pmatrix} 1 & 2 \\ -3 & 5 \end{pmatrix}$ then

$$A^{-1} = \frac{1}{11} \begin{pmatrix} 5 & -2 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} \frac{5}{11} & -\frac{2}{11} \\ \frac{3}{11} & \frac{1}{11} \end{pmatrix}$$

The reader should verify that $AA^{-1} = A^{-1}A = I$.

The second matrix, B, has a determinant equal to zero. If we tried to apply the formula in Theorem 5.2.2, we would be dividing by zero. For this reason, the formula can't be applied and in fact B^{-1} does not exist.

Remarks:

- In general, if A is a 2×2 matrix and if $\det A = 0$, then A^{-1} does not exist.
- A formula for the inverse of $n \times n$ matrices $n \geq 3$ can be derived that also involves $\det A$. Hence, in general, if the determinant of a matrix is zero, the matrix does not have an inverse. However the formula for even a 3×3 matrix is very long and is not the most efficient way to compute the inverse of a matrix.
- In Chapter 12 we will develop a technique to compute the inverse of a higher-order matrix, if it exists.
- Matrix inversion comes first in the hierarchy of matrix operations; therefore, AB^{-1} is $A(B^{-1})$.

5.2.3: Exercises

Exercise 5.2.1

For the given matrices A find A^{-1} if it exists and verify that $AA^{-1} = A^{-1}A = I$. If A^{-1} does not exist explain why.

a. $A = \begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix}$

b. $A = \begin{pmatrix} 6 & -3 \\ 8 & -4 \end{pmatrix}$

c. $A = \begin{pmatrix} 1 & -3 \\ 0 & 1 \end{pmatrix}$

d. $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

e. Use the definition of the inverse of a matrix to find A^{-1}: $A = \begin{pmatrix} 3 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & -5 \end{pmatrix}$

Answer

a. $\begin{pmatrix} -1/5 & 3/5 \\ 2/5 & -1/5 \end{pmatrix}$

b. No inverse exists.

c. $\begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}$

d. $A^{-1} = A$

e. $\begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1/5 \end{pmatrix}$

Exercise 5.2.2

For the given matrices A find A^{-1} if it exists and verify that $AA^{-1} = A^{-1}A = I$. If A^{-1} does not exist explain why.

a. $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$

b. $A = \begin{pmatrix} 0 & 1 \\ 0 & 2 \end{pmatrix}$

c. $A = \begin{pmatrix} 1 & c \\ 0 & 1 \end{pmatrix}$

d. $A = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$, where $|a| \neq |b|$.

Exercise 5.2.3

a. Let $A = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & -3 \\ 2 & 1 \end{pmatrix}$. Verify that $(AB)^{-1} = B^{-1}A^{-1}$.

b. Let A and B be $n \times n$ invertible matrices. Prove that $(AB)^{-1} = B^{-1}A^{-1}$. Why is the right side of the above statement written "backwards"? Is this necessary? Hint: Use Theorem 5.2.1

Answer

Let A and B be n by n invertible matrices.

$$\begin{aligned} \left(B^{-1}A^{-1}\right)(AB) &= \left(B^{-1}\right)\left(A^{-1}(AB)\right) \\ &= \left(B^{-1}\right)\left(\left(A^{-1}A\right)B\right) \\ &= \left(\left(B^{-1}\right)IB\right) \\ &= B^{-1}(B) \\ &= I \end{aligned}$$

Similarly, $(AB)\left(B^{-1}A^{-1}\right) = I$.

By Theorem 5.2.1, $B^{-1}A^{-1}$ is the only inverse of AB. If we tried to invert AB with $A^{-1}B^{-1}$, we would be unsuccessful since we cannot rearrange the order of the matrices.

Exercise 5.2.4

Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Derive the formula for A^{-1}.

Exercise 5.2.5: Linearity of Determinants

a. Let A and B be 2-by-2 matrices. Show that $\det(AB) = (\det A)(\det B)$.

b. It can be shown that the statement in part (a) is true for all $n \times n$ matrices. Let A be any invertible $n \times n$ matrix. Prove that $\det\left(A^{-1}\right) = (\det A)^{-1}$. Note: The determinant of the identity matrix I_n is 1 for all n.

c. Verify that the equation in part (b) is true for the matrix in Exercise 5.2.1(a) of this section.

Answer

$1 = \det I = \det\left(AA^{-1}\right) = \det A \ \det A^{-1}$. Now solve for $\det A^{-1}$.

Exercise 5.2.6

Prove by induction that for $n \geq 1$, $\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}^n = \begin{pmatrix} a^n & 0 \\ 0 & b^n \end{pmatrix}$.

Exercise 5.2.7

Use the assumptions in Exercise 5.2.5 to prove by induction that if $n \geq 1$, $\det\left(A^n\right) = (\det A)^n$.

Answer

Basis: $(n = 1) : \det A^1 = \det A = (\det A)^1$

Induction: Assume $\det A^n = (\det A)^n$ for some $n \geq 1$.

$$\begin{aligned} \det A^{n+1} &= \det\left(A^n A\right) && \text{by the definition of exponents} \\ &= \det\left(A^n\right)\det(A) && \text{by exercise 5} \\ &= (detA)^n(\det A) && \text{by the induction hypothesis} \\ &= (\det A)^{n+1} \end{aligned}$$

Exercise 5.2.8

Prove: If the determinant of a matrix A is zero, then A does not have an inverse. Hint: Use the indirect method of proof and exercise 5.2.5.

Exercise 5.2.9

a. Let $A, B,$ and D be $n \times n$ matrices. Assume that B is invertible. If $A = BDB^{-1}$, prove by induction that $A^m = BD^m B^{-1}$ is true for $m \geq 1$.

b. Given that $A = \begin{pmatrix} -8 & 15 \\ -6 & 11 \end{pmatrix} = B \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} B^{-1}$ where $B = \begin{pmatrix} 5 & 3 \\ 3 & 2 \end{pmatrix}$ what is A^{10}?

Answer

a. Assume $A = BDB^{-1}$

Basis: $(m = 1)$: $A^1 = A = BD^1 B^{-1}$ is given.

Induction: Assume that for some positive integer m, $A^m = BD^m B^{-1}$

$$\begin{aligned} A^{m+1} &= A^m A \\ &= (BD^m B^{-1})(BDB^{-1}) \quad \text{by the induction hypothesis} \\ &= (BD^m (B^{-1}B)(DB^{-1}) \quad \text{by associativity} \\ &= BD^m DB^{-1} \quad \text{by the definition of inverse} \\ &= BD^{m+1} B^{-1} \quad \square \end{aligned}$$

b. $A^{10} = BD^{10} B^{-1} = \begin{pmatrix} -9206 & 15345 \\ -6138 & 10231 \end{pmatrix}$

This page titled 5.2: Special Types of Matrices is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

5.3: Laws of Matrix Algebra

5.3.1: Laws

The following is a summary of the basic laws of matrix operations. Assume that the indicated operations are defined; that is, that the orders of the matrices A, B and C are such that the operations make sense.

Table 5.3.1: Laws of Matrix Algebra

(1) Commutative Law of Addition	$A+B=B+A$
(2) Associative Law of Addition	$A+(B+C)=(A+B)+C$
(3) Distributive Law of a Scalar over Matrices	$c(A+B)=cA+cB$, where $c\in\mathbb{R}$.
(4) Distributive Law of Scalars over a Matrix	$(c_1+c_2)\,A=c_1A+c_2A$, where $c_1,c_2\in\mathbb{R}$.
(5) Associative Law of Scalar Multiplication	$c_1\,(c_2A)=(c_1\cdot c_2)\,A$, where $c_1,c_2\in\mathbb{R}$.
(6) Zero Matrix Annihilates all Products	$\mathbf{0}A=\mathbf{0}$, where $\mathbf{0}$ is the zero matrix.
(7) Zero Scalar Annihilates all Products	$0A=\mathbf{0}$, where 0 on the left is the scalar zero.
(8) Zero Matrix is an identity for Addition	$A+\mathbf{0}=A$.
(9) Negation produces additive inverses	$A+(-1)A=\mathbf{0}$.
(10) Right Distributive Law of Matrix Multiplication	$(B+C)A=BA+CA$.
(11) Left Distributive Law of Matrix Multiplication	$A(B+C)=AB+AC$.
(12) Associative Law of Multiplication	$A(BC)=(AB)C$.
(13) Identity Matrix is a Multiplicative Identity	$IA=A$ and $AI=A$.
(14) Involution Property of Inverses	If A^{-1} exists, $\left(A^{-1}\right)^{-1}=A$.
(15) Inverse of Product Rule	If A^{-1} and B^{-1} exist, $(AB)^{-1}=B^{-1}A^{-1}$

5.3.2: Commentary

Example 5.3.1: More Precise Statement of One Law

If we wished to write out each of the above laws more completely, we would specify the orders of the matrices. For example, Law 10 should read:

> *Let A, B, and C be $m\times n$, $n\times p$, and $n\times p$ matrices, respectively, then* $A(B+C)=AB+AC$

Remarks:

- Notice the absence of the "law" $AB=BA$. Why?
- Is it really necessary to have both a right (No. 11) and a left (No. 10) distributive law? Why?

5.3.3: Exercises

Exercise 5.3.1

Rewrite the above laws specifying as in Example 5.3.1 the orders of the matrices.

Answer

a. Let A and B be m by n matrices. Then $A+B=B+A$,
b. Let A, B, and C be m by n matrices. Then $A+(B+C)=(A+B)+C$.

c. Let A and B be m by n matrices, and let $c \in \mathbb{R}$. Then $c(A+B) = cA + cB$,
d. Let A be an m by n matrix, and let $c_1, c_2 \in \mathbb{R}$. Then $(c_1 + c_2)\, A = c_1 A + c_2 A$.
e. Let A be an m by n matrix, and let $c_1, c_2 \in \mathbb{R}$. Then $c_1\,(c_2 A) = (c_1 c_2)\, A$
f. Let $\mathbf{0}$ be the zero matrix, of size m by n, and let A be a matrix of size n by r. Then $\mathbf{0}A = \mathbf{0} =$ the m by r zero matrix.
g. Let A be an m by n matrix, and $0 =$ the number zero. Then $0A = 0 =$ the m by n zero matrix.
h. Let A be an m by n matrix, and let $\mathbf{0}$ be the m by n zero matrix. Then $A + \mathbf{0} = A$.
i. Let A be an m by n matrix. Then $A + (-1)A = \mathbf{0}$, where $\mathbf{0}$ is the m by n zero matrix.
j. Let A, B, and C be m by n, n by r, and n by r matrices respectively. Then $A(B+C) = AB + AC$.
k. Let A, B, and C be m by n, r by m, and r by m matrices respectively. Then $(B+C)A = BA + CA$.
l. Let A, B, and C be m by n, n by r, and r by p matrices respectively. Then $A(BC) = (AB)C$.
m. Let A be an m by n matrix, I_m the m by m identity matrix, and I_n the n by n identity matrix. Then $I_m A = A I_n = A$
n. Let A be an n by n matrix. Then if A^{-1} exists, $\left(A^{-1}\right)^{-1} = A$.
o. Let A and B be n by n matrices. Then if A^{-1} and B^{-1} exist, $(AB)^{-1} = B^{-1}A^{-1}$.

Exercise 5.3.2

Verify each of the Laws of Matrix Algebra using examples.

Exercise 5.3.3

Let $A = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix}$, $B = \begin{pmatrix} 3 & 7 & 6 \\ 2 & -1 & 5 \end{pmatrix}$, and $C = \begin{pmatrix} 0 & -2 & 4 \\ 7 & 1 & 1 \end{pmatrix}$. Compute the following as efficiently as possible by using any of the Laws of Matrix Algebra:

a. $AB + AC$
b. A^{-1}
c. $A(B+C)$
d. $\left(A^2\right)^{-1}$
e. $(C+B)^{-1}A^{-1}$

Answer

a. $AB + AC = \begin{pmatrix} 21 & 5 & 22 \\ -9 & 0 & -6 \end{pmatrix}$
b. $A^{-1} = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix} = A$
c. $A(B+C) = AB + BC$, which is given in part (a).
d. $\left(A^2\right)^{-1} = (AA)^{-1} = (A^{-1}A) = I^{-1} = I$ by part c

Exercise 5.3.4

Let $A = \begin{pmatrix} 7 & 4 \\ 2 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & 5 \\ 2 & 4 \end{pmatrix}$. Compute the following as efficiently as possible by using any of the Laws of Matrix Algebra:

a. AB
b. $A + B$
c. $A^2 + AB + BA + B^2$
d. $B^{-1}A^{-1}$
e. $A^2 + AB$

Exercise 5.3.5

Let A and B be $n \times n$ matrices of real numbers. Is $A^2 - B^2 = (A - B)(A + B)$? Explain.

This page titled 5.3: Laws of Matrix Algebra is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

5.4: Matrix Oddities

5.4.1: Dissimilarities with Elementary Algebra

We have seen that matrix algebra is similar in many ways to elementary algebra. Indeed, if we want to solve the matrix equation $AX = B$ for the unknown X, we imitate the procedure used in elementary algebra for solving the equation $ax = b$. One assumption we need is that A is a square matrix that has an inverse. Notice how exactly the same properties are used in the following detailed solutions of both equations.

Table 5.4.1

Equation in the Algebra of Real Numbers		Equation in Matrix Algebra
$ax = b$		$AX = B$
$a^{-1}(ax) = a^{-1}b$ if $a \neq 0$		$A^{-1}(AX) = A^{-1}B$ if A^{-1} exists
$(a^{-1}a)x = a^{-1}b$	Associative Property	$(A^{-1}A)X = A^{-1}B$
$1x = a^{-1}b$	Inverse Property	$IX = A^{-1}B$
$x = a^{-1}b$	Identity Property	$X = A^{-1}B$

Certainly the solution process for solving $AX = B$ is the same as that of solving $ax = b$.

The solution of $xa = b$ is $x = ba^{-1} = a^{-1}b$. In fact, we usually write the solution of both equations as $x = \frac{b}{a}$. In matrix algebra, the solution of $XA = B$ is $X = BA^{-1}$, which is not necessarily equal to $A^{-1}B$. So in matrix algebra, since the commutative law (under multiplication) is not true, we have to be more careful in the methods we use to solve equations.

It is clear from the above that if we wrote the solution of $AX = B$ as $X = \frac{B}{A}$, we would not know how to interpret $\frac{B}{A}$. Does it mean $A^{-1}B$ or BA^{-1}? Because of this, A^{-1} is never written as $\frac{I}{A}$.

Observation 5.4.1: Matrix Oddities

Some of the main dissimilarities between matrix algebra and elementary algebra are that in matrix algebra:

1. AB may be different from BA.
2. There exist matrices A and B such that $AB = \mathbf{0}$, and yet $A \neq \mathbf{0}$ and $B \neq \mathbf{0}$.
3. There exist matrices A where $A \neq \mathbf{0}$, and yet $A^2 = \mathbf{0}$.
4. There exist matrices A where $A^2 = A$ with $A \neq I$ and $A \neq \mathbf{0}$
5. There exist matrices A where $A^2 = I$, where $A \neq I$ and $A \neq -I$

5.4.2: Exercises

Exercise 5.4.1

Discuss each of the “Matrix Oddities” with respect to elementary algebra.

Answer

In elementary algebra (the algebra of real numbers), each of the given oddities does not exist.

- AB may be different from BA. Not so in elementary algebra, since $ab = ba$ by the commutative law of multiplication.
- There exist matrices A and B such that $AB = \mathbf{0}$, yet $A \neq \mathbf{0}$ and $B \neq \mathbf{0}$. In elementary algebra, the only way $ab = 0$ is if either a or b is zero. There are no exceptions.
- There exist matrices A, $A \neq \mathbf{0}$, yet $A^2 = \mathbf{0}$. In elementary algebra, $a^2 = 0 \Leftrightarrow a = 0$.
- There exist matrices $A^2 = A$. where $A \neq \mathbf{0}$ and $A \neq I$. In elementary algebra, $a^2 = a \Leftrightarrow a = 0$ or 1.
- There exist matrices A where $A^2 = I$ but $A \neq I$ and $A \neq -I$. In elementary algebra, $a^2 = 1 \Leftrightarrow a = 1$ or -1.

Exercise 5.4.2

Determine 2×2 matrices which show that each of the "Matrix Oddities" are true.

Exercise 5.4.3

Prove or disprove the following implications.

a. $A^2 = A$ and $\det A \neq 0 \Rightarrow A = I$
b. $A^2 = I$ and $\det A \neq 0 \Rightarrow A = I$ or $A = -I$.

Answer

a. $\det A \neq 0 \Rightarrow A^{-1}$ exists, and if you multiply the equation $A^2 = A$ on both sides by A^{-1}, you obtain $A = I$.
b. Counterexample: $A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Exercise 5.4.4

Let $M_{n\times n}(\mathbb{R})$ be the set of real $n \times n$ matrices. Let $P \subseteq M_{n\times n}(\mathbb{R})$ be the subset of matrices defined by $A \in P$ if and only if $A^2 = A$. Let $Q \subseteq P$ be defined by $A \in Q$ if and only if $\det A \neq 0$.

a. Determine the cardinality of Q.
b. Consider the special case $n = 2$ and prove that a sufficient condition for $A \in P \subseteq M_{2\times 2}(\mathbb{R})$ is that A has a zero determinant (i.e., A is singular) and $tr(A) = 1$ where $tr(A) = a_{11} + a_{22}$ is the sum of the main diagonal elements of A.
c. Is the condition of part b a necessary condition?

Exercise 5.4.5

Write each of the following systems in the form $AX = B$, and then solve the systems using matrices.

a. $\begin{aligned} 2x_1 + x_2 &= 3 \\ x_1 - x_2 &= 1 \end{aligned}$
b. $\begin{aligned} 2x_1 - x_2 &= 4 \\ x_1 - x_2 &= 0 \end{aligned}$
c. $\begin{aligned} 2x_1 + x_2 &= 1 \\ x_1 - x_2 &= 1 \end{aligned}$
d. $\begin{aligned} 2x_1 + x_2 &= 1 \\ x_1 - x_2 &= -1 \end{aligned}$
e. $\begin{aligned} 3x_1 + 2x_2 &= 1 \\ 6x_1 + 4x_2 &= -1 \end{aligned}$

Answer

a. $A^{-1} = \begin{pmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{pmatrix}$ $x_1 = 4/3$, and $x_2 = 1/3$
b. $A^{-1} = \begin{pmatrix} 1 & -1 \\ 1 & -2 \end{pmatrix}$ $x_1 = 4$, and $x_2 = 4$
c. $A^{-1} = \begin{pmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{pmatrix}$ $x_1 = 2/3$, and $x_2 = -1/3$
d. $A^{-1} = \begin{pmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{pmatrix}$ $x_1 = 0$, and $x_2 = 1$
e. The matrix of coefficients for this system has a zero determinant; therefore, it has no inverse. The system cannot be solved by this method. In fact, the system has no solution.

Exercise 5.4.6

Recall that $p(x) = x^2 - 5x + 6$ is called a polynomial, or more specifically, a polynomial over $\mathbb{R}$, where the coefficients are elements of $\mathbb{R}$ and $x \in \mathbb{R}$. Also, think of the method of solving, and solutions of, $x^2 - 5x + 6 = 0$. We would like to define the analogous situation for 2×2 matrices. First define where A is a 2×2 matrix $p(A) = A^2 - 5A + 6I$. Discuss the method of solving and the solutions of $A^2 - 5A + 6I = \mathbf{0}$.

Exercise 5.4.7

For those who know calculus:

a. Write the series expansion for e^a centered around $a = 0$.
b. Use the idea of Exercise 5.4.6 to write what would be a plausible definition of e^A where A is an $n \times n$ matrix.
c. If $A = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}$, use the series in part (b) to show that $e^A = \begin{pmatrix} e & e-1 \\ 0 & 1 \end{pmatrix}$ and $e^B = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$.
d. Show that $e^A e^B \neq e^B e^A$.
e. Show that $e^{A+B} = \begin{pmatrix} e & 0 \\ 0 & 1 \end{pmatrix}$.
f. Is $e^A e^B = e^{A+B}$?

This page titled 5.4: Matrix Oddities is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

6: Relations

> ***adjacency matrix***
>
> *An **adjacency matrix** will show*
> *Where the edges 'tween vertices go.*
> *For a nice simple graph*
> *You can cut through the chaff:*
> *Noughts and ones in symmetrical flow.*
>
> *psheil, The Omnificent English Dictionary in Limerick Form*

One understands a set of objects completely only if the structure of that set is made clear by the interrelationships between its elements. For example, the individuals in a crowd can be compared by height, by age, or through any number of other criteria. In mathematics, such comparisons are called relations. The goal of this chapter is to develop the language, tools, and concepts of relations.

6.1: Basic Definitions
6.2: Graphs of Relations on a Set
6.3: Properties of Relations
6.4: Matrices of Relations
6.5: Closure Operations on Relations

This page titled 6: Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

6.1: Basic Definitions

In Chapter 1 we introduced the concept of the Cartesian product of sets. Let's assume that a person owns three shirts and two pairs of slacks. More precisely, let $A = \{\text{blue shirt, tan shirt, mint green shirt}\}$ and $B = \{\text{grey slacks, tan slacks}\}$. Then $A \times B$ is the set of all six possible combinations of shirts and slacks that the individual could wear. However, an individual may wish to restrict himself or herself to combinations which are color coordinated, or "related." This may not be all possible pairs in $A \times B$ but will certainly be a subset of $A \times B$. For example, one such subset may be

$$\{(\text{blue shirt, grey slacks}), (\text{blue shirt, tan slacks}), (\text{mint green shirt, tan slacks})\}.$$

6.1.1: Relations between Two Sets

Definition 6.1.1: Relation

Let A and B be sets. A relation from A into B is any subset of $A \times B$.

Example 6.1.1: A Simple Example

Let $A = \{1, 2, 3\}$ and $B = \{4, 5\}$. Then $\{(1, 4), (2, 4), (3, 5)\}$is a relation from A into B. Of course, there are many others we could describe; 64, to be exact.

Example 6.1.2: Divisibility Example

Let $A = \{2, 3, 5, 6\}$ and define a relation r from A into A by $(a, b) \in r$ if and only if a divides evenly into b. The set of pairs that qualify for membership is $r = \{(2, 2), (3, 3), (5, 5), (6, 6), (2, 6), (3, 6)\}$.

6.1.2: Relations on a Set

Definition 6.1.2: Relation on a Set

A relation from a set A into itself is called a relation on A.

The relation "divides" in Example 6.1.2 will appear throughout the book. Here is a general definition on the whole set of integers.

Definition 6.1.3: Divides

Let $a, b \in \mathbb{Z}$, $a \neq 0$. We say that a divides b, denoted $a \mid b$, if and only if there exists an integer k such that $ak = b$.

Be very careful in writing about the relation "divides." The vertical line symbol use for this relation, if written carelessly, can look like division. While $a \mid b$ is either true or false, a/b is a number.

Based on the equation $ak = b$, we can say that $a|b$ is equivalent to $k = \frac{b}{a}$, or a divides evenly into b. In fact the "divides" is short for "divides evenly into." You might find the equation $k = \frac{b}{a}$ initially easier to understand, but in the long run we will find the equation $ak = b$ more convenient.

Sometimes it is helpful to illustrate a relation with a graph. Consider Example 6.1.1. A graph of r can be drawn as in Figure 6.1.1. The arrows indicate that 1 is related to 4 under r. Also, 2 is related to 4 under r, and 3 is related to 5, while the upper arrow denotes that r is a relation from the whole set A into the set B.

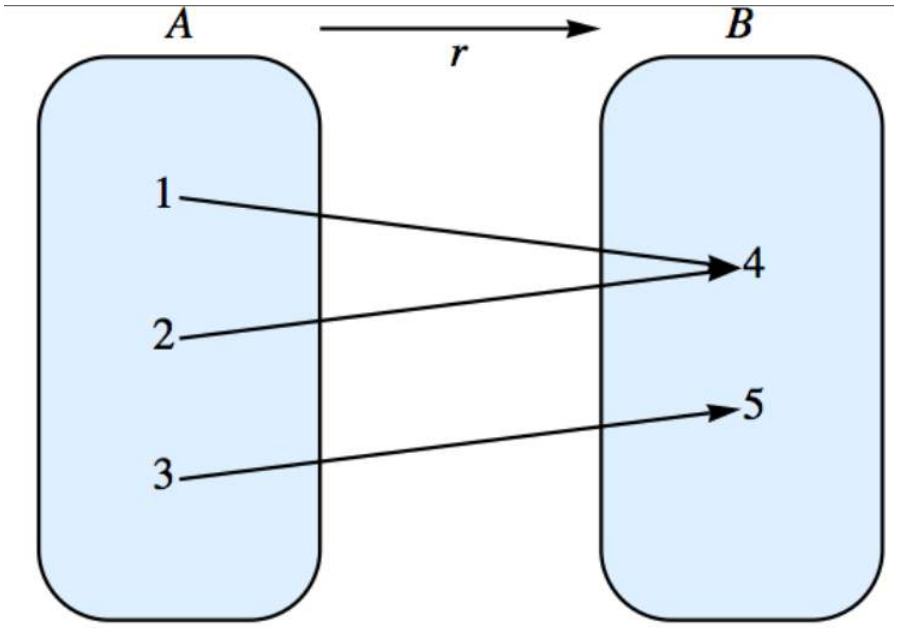


Figure 6.1.1: The graph of a relation

A typical element in a relation r is an ordered pair (x, y). In some cases, r can be described by actually listing the pairs which are in r, as in the previous examples. This may not be convenient if r is relatively large. Other notations are used with certain well-known relations. Consider the "less than or equal" relation on the real numbers. We could define it as a set of ordered pairs this way:

$$\leq = \{(x, y) | x \leq y\}$$

However, the notation $x \leq y$ is clear and self-explanatory; it is a more natural, and hence preferred, notation to use than $(x, y) \in \leq$.

Many of the relations we will work with "resemble" the relation $\leq$, so xsy is a common way to express the fact that x is related to y through the relation s.

Relation Notation Let s be a relation from a set A into a set B. Then the fact that $(x, y) \in s$ is frequently written xsy.

6.1.3: Composition of Relations

With $A = \{2, 3, 5, 8\}$, $B = \{4, 6, 16\}$, and $C = \{1, 4, 5, 7\}$, let r be the relation "divides," from A into B, and let s be the relation $\leq$ from B into C. So $r = \{(2, 4), (2, 6), (2, 16), (3, 6), (8, 16)\}$and $s = \{(4, 4), (4, 5), (4, 7), (6, 7)\}$.

Notice that in Figure 6.1.2 that we can, for certain elements of A, go through elements in B to results in C. That is:

Table 6.1.1

$2 \mid 4$ and $4 \leq 4$
$2 \mid 4$ and $4 \leq 5$
$2 \mid 4$ and $4 \leq 7$

$2 \mid 6$ and $6 \leq 7$
$3 \mid 6$ and $6 \leq 7$

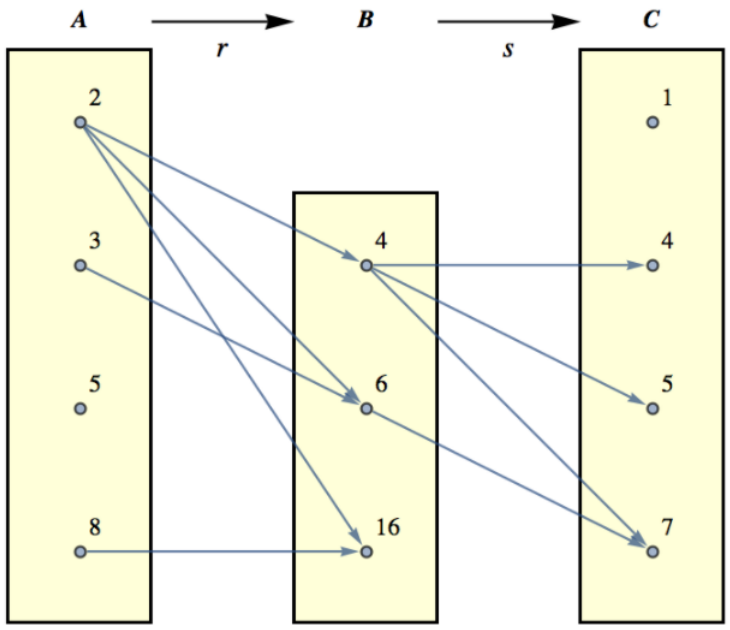


Figure 6.1.2: Relation Composition - a graphical view

Based on this observation, we can define a new relation, call it rs, from A into C. In order for (a, c) to be in rs, it must be possible to travel along a path in Figure 6.1.2 from a to c. In other words, $(a, c) \in rs$ if and only if $(\exists b)_B(arb \text{ and } bsc)$. The name rs was chosen because it reminds us that this new relation was formed by the two previous relations r and s. The complete listing of all elements in rs is $\{(2, 4), (2, 5), (2, 7), (3, 7)\}$.We summarize in a definition.

Definition 6.1.4: Composition of Relations

Let r be a relation from a set A into a set B, and let s be a relation from B into a set C. The composition of r with s, written rs, is the set of pairs of the form $(a, c) \in A \times C$, where $(a, c) \in rs$ if and only if there exists $b \in B$ such that $(a, b) \in r$ and $(b, c) \in s$.

Remark: A word of warning to those readers familiar with composition of functions. (For those who are not, disregard this remark. It will be repeated at an appropriate place in the next chapter.) As indicated above, the traditional way of describing a composition of two relations is rs where r is the first relation and s the second. However, function composition is traditionally expressed in the opposite order: $s \circ r$, where r is the first function and s is the second.

6.1.4: Exercises

Exercise 6.1.1

For each of the following relations r defined on $\mathbb{P}$, determine which of the given ordered pairs belong to r

a. xry iff $x \mid y$; (2, 3), (2, 4), (2, 8), (2, 17)
b. xry iff $x \leq y$; (2, 3), (3, 2), (2, 4), (5, 8)
c. xry iff $y = x^2$; (1,1), (2, 3), (2, 4), (2, 6)

Answer

a. $(2, 4), (2, 8)$
b. $(2, 3), (2, 4), (5, 8)$
c. $(1, 1), (2, 4)$

Exercise 6.1.2

The following relations are on $\{1,3,5\}$.Let r be the relation xry iff $y = x+2$ and s the relation xsy iff $x \leq y$.

a. List all elements in rs.
b. List all elements in sr.
c. Illustrate rs and sr via a diagram.
d. Is the relation rs equal to the relation sr?

Exercise 6.1.3

Let $A = \{1,2,3,4,5\}$ and define r on A by xry iff $x+1 = y$. We define $r^2 = rr$ and $r^3 = r^2 r$. Find:

a. r
b. r^2
c. r^3

Answer

a. $r = \{(1,2),(2,3),(3,4),(4,5)\}$
b. $r^2 = \{(1,3),(2,4),(3,5)\} = \{(x,y) : y = x+2, x, y \in A\}$
c. $r^3 = \{(1,4),(2,5)\} = \{(x,y) : y = x+3, x, y \in A\}$

Exercise 6.1.4

Given s and t, relations on $\mathbb{Z}$, $s = \{(1,n) : n \in \mathbb{Z}\}$ and $t = \{(n,1) : n \in \mathbb{Z}\}$, what are st and ts? Hint: Even when a relation involves infinite sets, you can often get insights into them by drawing partial graphs.

Exercise 6.1.5

Let ρ be the relation on the power set, $\mathcal{P}(S)$, of a finite set S of cardinality n defined ρ by $(A,B) \in \rho$ iff $A \cap B = \emptyset$.

a. Consider the specific case $n = 3$, and determine the cardinality of the set ρ.
b. What is the cardinality of ρ for an arbitrary n? Express your answer in terms of n. (Hint: There are three places that each element of S can go in building an element of ρ.)

Answer

a. When $n = 3$, there are 27 pairs in the relation.
b. Imagine building a pair of disjoint subsets of S. For each element of S there are three places that it can go: into the first set of the ordered pair, into the second set, or into neither set. Therefore the number of pairs in the relation is 3^n, by the product rule.

Exercise 6.1.6

Consider the two relations on people: M, where aMb if a's mother is b; and S, where aSb if a and b are siblings. Describe, in words, the two relations MS and SM in simple English terms.

Exercise 6.1.7

Let r_1, r_2, and r_3 be relations on any set A. Prove that if $r_1 \subseteq r_2$ then $r_1 r_3 \subseteq r_2 r_3$.

Answer

Assume $(x,y) \in r_1 r_3$. This implies that there exist $z \in A$ such that $(x,z) \in r_1$ and $(z,y) \in r_3$. We are given that , $r_1 \subseteq r_2$, which implies that $(x,z) \in r_2$. Combining this with $(z,y) \in r_3$ implies that $(x,y) \in r_2 r_3$, which proves that $r_1 r_3 \subseteq r_2 r_3$.

This page titled 6.1: Basic Definitions is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

6.2: Graphs of Relations on a Set

In this section we introduce directed graphs as a way to visualize relations on a set.

6.2.1: Digraphs

Let $A = \{0, 1, 2, 3\}$, and let

$$r = \{(0, 0), (0, 3), (1, 2), (2, 1), (3, 2), (2, 0)\}$$

In representing this relation as a graph, elements of A are called the vertices of the graph. They are typically represented by labeled points or small circles. We connect vertex a to vertex b with an arrow, called an edge, going from vertex a to vertex b if and only if arb. This type of graph of a relation r is called a ***directed graph*** or ***digraph***. Figure $6.2.1$ is a digraph for r. Notice that since 0 is related to itself, we draw a "self-loop" at 0.

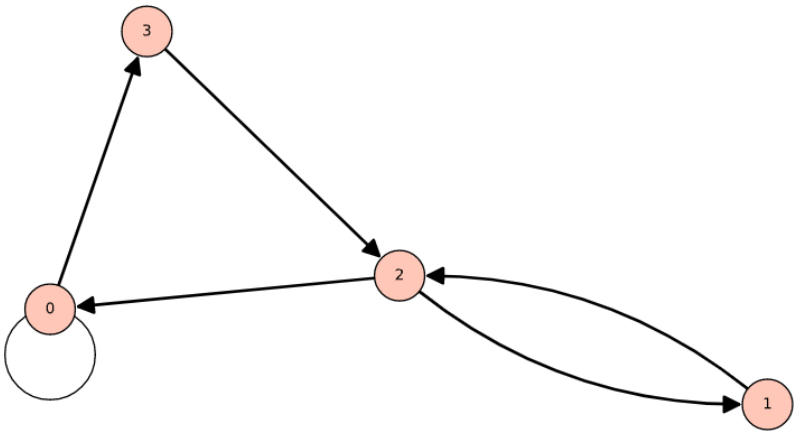


Figure $6.2.1$: Digraph of a relation

The actual location of the vertices in a digraph is immaterial. The actual location of vertices we choose is called an ***embedding of a graph***. The main idea is to place the vertices in such a way that the graph is easy to read. After drawing a rough-draft graph of a relation, we may decide to relocate the vertices so that the final result will be neater. Figure $6.2.1$ could also be presented as in Figure $6.2.2$.

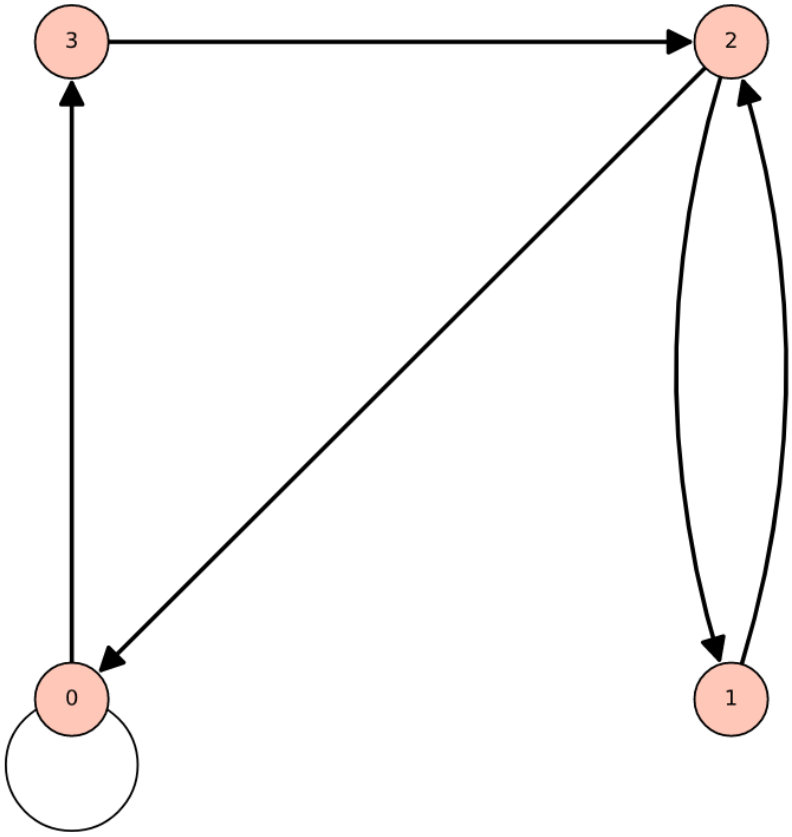


Figure 6.2.2: Alternate embedding of the previous directed graph

A vertex of a graph is also called a node, point, or a junction. An edge of a graph is also referred to as an arc, a line, or a branch. Do not be concerned if two graphs of a given relation look different as long as the connections between vertices are the same in the two graphs.

Example 6.2.1: Another Directed Graph

Consider the relation s whose digraph is Figure 6.2.3. What information does this give us? The graph tells us that s is a relation on $A = \{1, 2, 3\}$ and that $s = \{(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 3)\}$.

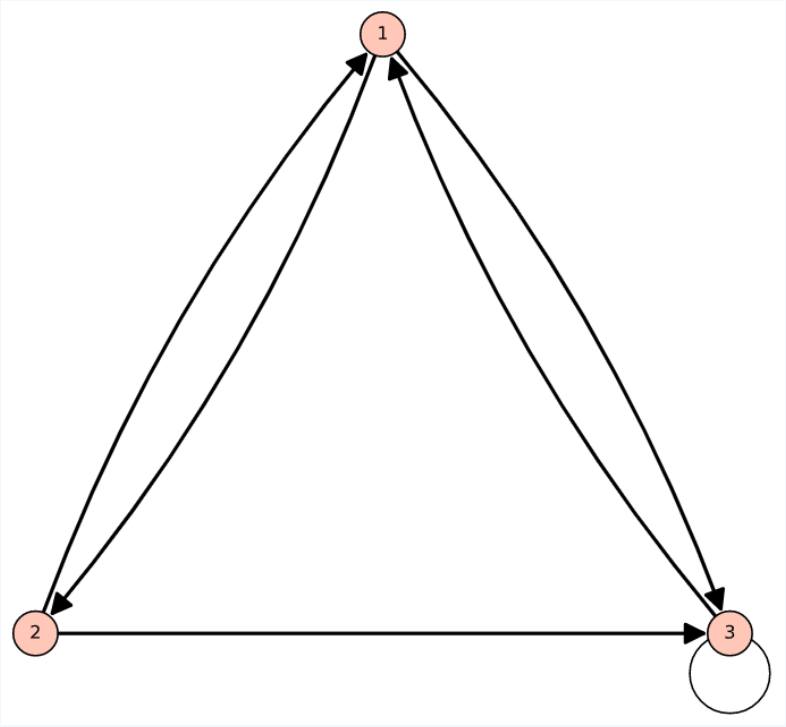


Figure 6.2.3: Digraph of the relation s

We will be building on the next example in the following section.

Example 6.2.2: Ordering Subsets of a Two Element Universe

Let $B = \{1, 2\}$, and let $A = \mathcal{P}(B) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$. Then $\subseteq$ is a relation on A whose digraph is Figure 6.2.4.

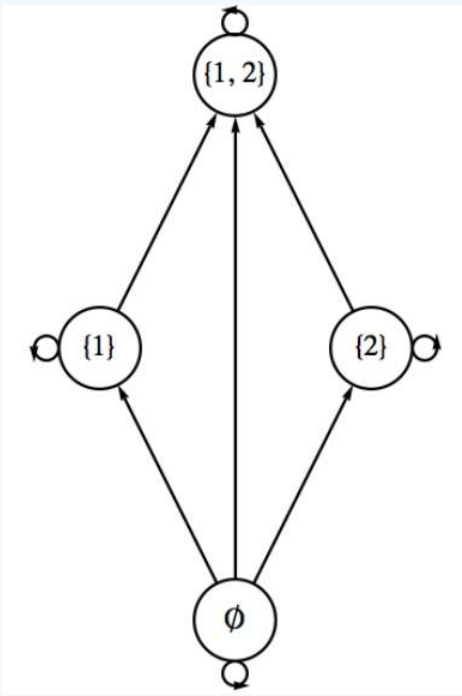


Figure 6.2.4: Graph for set containment on subsets of $\{1, 2\}$

We will see in the next section that since $\subseteq$ has certain structural properties that describe "partial orderings." We will be able to draw a much simpler type graph than this one, but for now the graph above serves our purposes.

6.2.2: Exercises

Exercise 6.2.1

Let $A = \{1, 2, 3, 4\}$, and let r be the relation $\leq$ on A. Draw a digraph for r.

Answer

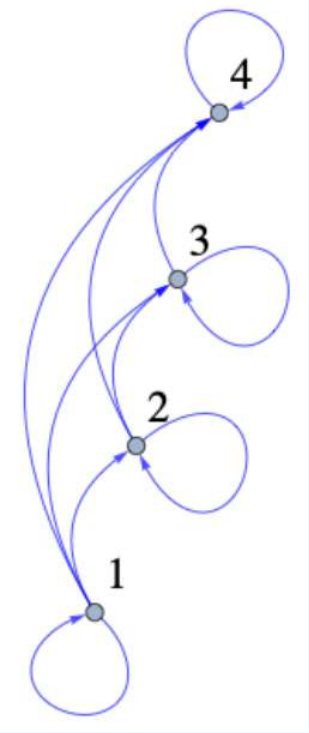


Figure 6.2.5 Digraph for $\leq$



Figure 6.2.6 Hasse diagram for $\leq$

Exercise 6.2.2

Let $B = \{1, 2, 3, 4, 6, 8, 12, 24\}$, and let s be the relation "divides" on B. Draw a digraph for s.

Exercise 6.2.3

Let $A = \{1, 2, 3, 4, 5\}$. Define t on A by atb if and only if $b - a$ is even. Draw a digraph for t.

Answer

See Figure 6.2.7

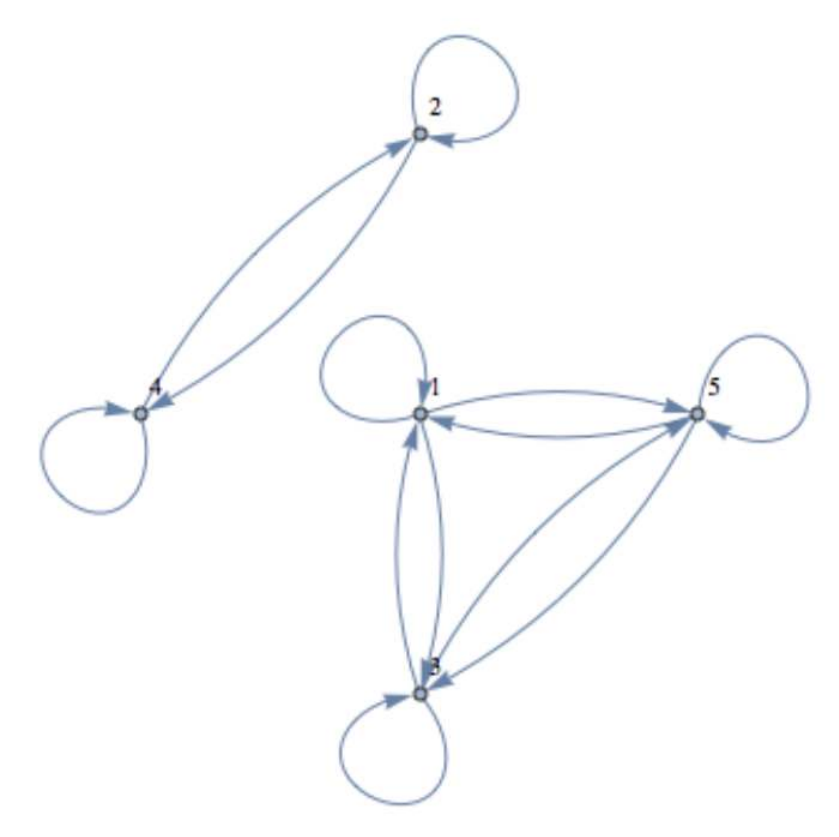


Figure 6.2.7: Digraph of the relation t

Exercise 6.2.4

Let A be the set of strings of 0's and 1's of length 3 or less. This includes the empty string, λ, which is the only string of length zero.

a. Define the relation of d on A by xdy if x is contained within y. For example, $01d101$. Draw a digraph for this relation.
b. Do the same for the relation p defined by xpy if x is a prefix of y. For example, $10p101$, but $01p101$ is false.

Exercise 6.2.5

Recall the relation in Exercise 6.1.5 of Section 6.1, ρ defined on the power set, $\mathcal{P}(S)$, of a set S. The definition was $(A, B) \in \rho$ iff $A \cap B = \emptyset$. Draw the digraph for ρ where $S = \{a, b\}$.

Exercise 6.2.6

Let $C = \{1, 2, 3, 4, 6, 8, 12, 24\}$ and define t on C by atb if and only if a and b share a common divisor greater than 1. Draw a digraph for t.

This page titled 6.2: Graphs of Relations on a Set is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

6.3: Properties of Relations

6.3.1: Individual Properties

Consider the set $B = \{1, 2, 3, 4, 6, 12, 36, 48\}$and the relations "divides" and $\leq$ on B. We notice that these two relations on B have three properties in common:

- Every element in B divides itself and is less than or equal to itself. This is called the reflexive property.
- If we search for two elements from B where the first divides the second and the second divides the first, then we are forced to choose the two numbers to be the same. In other words, no two *different* numbers are related in both directions. The reader can verify that a similar fact is true for the relation $\leq$ on B. This is called the antisymmetric property.
- Next if we choose three values (not necessarily distinct) from B such that the first divides the second and the second divides the third, then we always find that the first number divides the third. Again, the same is true if we replace "divides" with "is less than or equal to." This is called the transitive property.

Relations that satisfy these properties are of special interest to us. Formal definitions of the properties follow.

Definition 6.3.1: Reflexive Relation

Let A be a set and let r be a relation on A. Then r is ***reflexive*** if and only if ara for all $a \in A$.

Definition 6.3.2: Antisymmetric Relation

Let A be a set and let r be a relation on A. Then r is ***antisymmetric*** if and only if whenever arb and $a \neq b$ then bra is false.

An equivalent condition for antisymmetry is that if arb and bra then $a = b$. You are encouraged to convince yourself that this is true. This condition is often more convenient to prove than the definition, even though the definition is probably easier to understand.

A word of warning about antisymmetry: Students frequently find it difficult to understand this definition. Keep in mind that this term is defined through an "If...then..." statement. The question that you must ask is: Is it true that whenever there are elements a and b from A where arb and $a \neq b$, it follows that b is not related to a? If so, then the relation is antisymmetric.

Another way to determine whether a relation is antisymmetric is to examine (or imagine) its digraph. The relation is not antisymmetric if there exists a pair of vertices that are connected by edges in both directions.

Definition 6.3.3: Transitive Relation

Let A be a set and let r be a relation on A. r is ***transitive*** if and only if whenever arb and brc then arc.

6.3.2: Partial Orderings

Not all relations have all three of the properties discussed above, but those that do are a special type of relation.

Definition 6.3.4: Partial Ordering

A relation on a set A that is reflexive, antisymmetric, and transitive is called a ***partial ordering*** on A. A set on which there is a partial ordering relation defined is called a ***partially ordered set*** or ***poset***.

Example 6.3.1: Set Containment as a Partial Ordering

Let A be a set. Then $\mathcal{P}(A)$ together with the relation $\subseteq$ (set containment) is a poset. To prove this we observe that the three properties hold, as discussed in Chapter 4.

- Let $B \in \mathcal{P}(A)$. The fact that $B \subseteq B$ follows from the definition of subset. Hence, set containment is reflexive.
- Let $B_1, B_2 \in \mathcal{P}(A)$ and assume that $B_1 \subseteq B_2$ and $B_1 \neq B_2$. Could it be that $B_2 \subseteq B_1$? No. There must be some element $a \in A$ such that $a \notin B_1$, but $a \in B_2$. This is exactly what we need to conclude that B_2 is not contained in B_1. Hence, set containment is antisymmetric.

- Let $B_1, B_2, B_3 \in \mathcal{P}(A)$ and assume that $B_1 \subseteq B_2$ and $B_2 \subseteq B_3$. Does it follow that $B_1 \subseteq B_3$? Yes, if $a \in B_1$, then $a \in B_2$ because $B_1 \subseteq B_2$. Now that we have $a \in B_2$ and we have assumed $B_2 \subseteq B_3$, we conclude that $a \in B_3$. Therefore, $B_1 \subseteq B_3$ and so set containment is transitive.

Figure 6.2.4 is the graph for the "set containment" relation on the power set of $\{1, 2\}$.

Figure 6.2.4 is helpful insofar as it reminds us that each set is a subset of itself and shows us at a glance the relationship between the various subsets in $\mathcal{P}(\{1, 2\})$. However, when a relation is a partial ordering, we can streamline a graph like this one. The streamlined form of a graph is called a ***Hasse diagram*** or ***ordering diagram***. A Hasse diagram takes into account the following facts.

- By the reflexive property, each vertex must be related to itself, so the arrows from a vertex to itself (called "self-loops") are not drawn in a Hasse diagram. They are simply assumed.
- By the antisymmetry property, connections between two distinct elements in a directed graph can only go one way, if at all. When there is a connection, we agree to always place the second element above the first (as we do above with the connection from $\{1\}$ to $\{1, 2\}$). For this reason, we can just draw a connection without an arrow, just a line.
- By the transitive property, if there are edges connecting one element up to a second element and the second element up to a third element, then there will be a direct connection from the first to the third. We see this in Figure 6.2.4 with $\emptyset$ connected to $\{1\}$and then $\{1\}$ connected to $\{1, 2\}$. Notice the edge connecting $\emptyset$ to $\{1, 2\}$. Whenever we identify this situation, remove the connection from the first to the third in a Hasse diagram and simply observe that an upward path of any length implies that the lower element is related to the upper one.

Using these observations as a guide, we can draw a Hasse diagram for $\subseteq$ on $\{1, 2\}$ as in Figure 6.3.1.

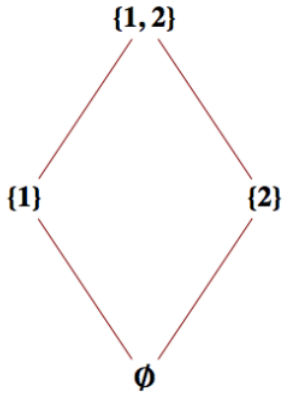


Figure 6.3.1: Hasse diagram for set containment on subsets of $\{1, 2\}$

Example 6.3.2: Definition of a Relation using a Hasse Diagram

Consider the partial ordering relation s whose Hasse diagram is Figure 6.3.2.

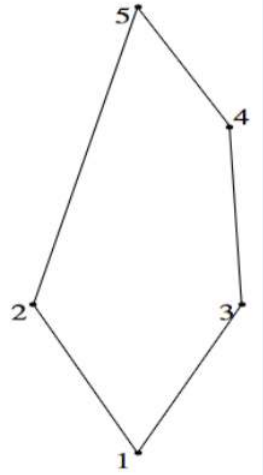


Figure 6.3.2 Hasse diagram for the pentagonal poset

How do we read this diagram? What is A? What is s? What does the digraph of s look like? Certainly $A = \{1, 2, 3, 4, 5\}$ and $1s2$, $3s4$, $1s4$, $1s5$, etc., Notice that $1s5$ is implied by the fact that there is a path of length three upward from 1 to 5. This follows from the edges that are shown and the transitive property that is presumed in a poset. Since $1s3$ and $3s4$, we know that $1s4$. We then combine $1s4$ with $4s5$ to infer $1s5$. Without going into details why, here is a complete list of pairs defined by s.

$$s = \{(1,1), (2,2), (3,3), (4,4), (5,5), (1,3), (1,4), (1,5), (1,2), (3,4), (3,5), (4,5), (2,5)\}$$

A digraph for s is Figure 6.3.3. It is certainly more complicated to read and difficult to draw than the Hasse diagram.

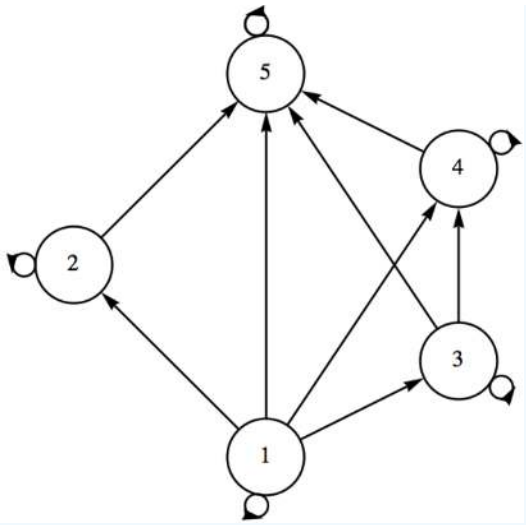


Figure 6.3.3 Digraph for the pentagonal poset

A classic example of a partial ordering relation is $\leq$ on the real numbers, $\mathbb{R}$. Indeed, when graphing partial ordering relations, it is natural to "plot" the elements from the given poset starting with the "least" element to the "greatest" and to use terms like "least," "greatest," etc. Because of this the reader should be forewarned that some texts use the symbol $\leq$ for arbitrary partial orderings. This can be quite confusing for the novice, so we continue to use generic letters r, s, etc.

6.3.3: Equivalence Relations

Another common property of relations is symmetry.

Definition 6.3.5: Symmetric Relation

Let r be a relation on a set A. r is ***symmetric*** if and only if whenever arb, it follows that bra.

Consider the relation of equality defined on any set A. Certainly $a = b$ implies that $b = a$ so equality is a symmetric relation on A.

Surprisingly, equality is also an antisymmetric relation on A. This is due to the fact that the condition that defines the antisymmetry property, $a = b$ and $a \neq b$, is a contradiction. Remember, a conditional proposition is always true when the condition is false. So a relation can be both symmetric and antisymmetric on a set! Again recall that these terms are *not* negatives of one other. That said, there are very few important relations other than equality that are both symmetric and antisymmetric.

Definition 6.3.6: Equivalence Relation

A relation r on a set A is called an equivalence relation if and only if it is reflexive, symmetric, and transitive.

The classic example of an equivalence relation is equality on a set A. In fact, the term equivalence relation is used because those relations which satisfy the definition behave quite like the equality relation. Here is another important equivalence relation.

Example 6.3.3: Equivalent Fractions

Let $\mathbb{Z}^*$ be the set of nonzero integers. One of the most basic equivalence relations in mathematics is the relation q on $\mathbb{Z} \times \mathbb{Z}^*$ defined by $(a, b)q(c, d)$ if and only if $ad = bc$. We will leave it to the reader to, verify that q is indeed an equivalence relation. Be aware that since the elements of $\mathbb{Z} \times \mathbb{Z}^*$ are ordered pairs, proving symmetry involves four numbers and transitivity involves six numbers. Two ordered pairs, (a, b) and (c, d), are related if the fractions $\frac{a}{b}$ and $\frac{c}{d}$ are numerically equal.

Our next example involves the following fundamental relations on the set of integers.

Definition 6.3.7: Congruence Modulo n

Let n be a positive integer, $n \geq 2$. We define ***congruence modulo n*** to be the relation $\equiv_n$ defined on the integers by

$$a \equiv_n b \Leftrightarrow n \mid (a - b)$$

We observe the following about congruence modulo n:

- This relation is reflexive, for if $a \in \mathbb{Z}$, $n \mid (a - a) \Rightarrow a \equiv_n a$.
- This relation is symmetric. We can prove this through the following chain of implications.

$$\begin{aligned} a \equiv_n b &\Rightarrow n \mid (a - b) \\ &\Rightarrow \text{For some } k \in \mathbb{Z}, a - b = nk \\ &\Rightarrow b - a = n(-k) \\ &\Rightarrow n \mid (b - a) \\ &\Rightarrow b \equiv_n a \end{aligned} \quad .$$

- Finally, this relation is transitive. We leave it to the reader to prove that if $a \equiv_n b$ and $b \equiv_n c$, then $a \equiv_n c$.

Frequently, you will see the equivalent notation $a \equiv b (\text{mod } n)$ for congruence modulo n.

Example 6.3.4: Random Relations Usually have No Properties

Consider the relation s described by the digraph in Figure 6.3.4. This was created by randomly selecting whether or not two elements from $\{a, b, c\}$ were related or not. Convince yourself that the following are true:

- This relation is not reflexive.
- It is not antisymmetric.
- Also, it is not symmetric.
- It is not transitive.
- Is s an equivalence relation or a partial ordering?

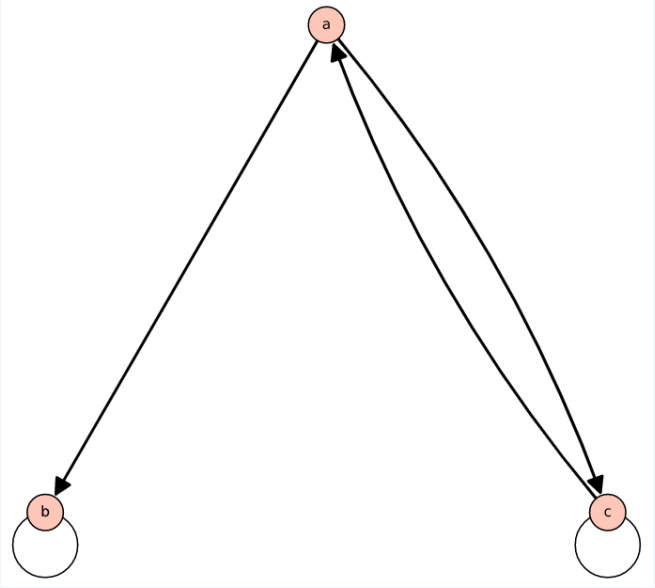


Figure 6.3.4 Digraph of a random relation r

Not every random choice of a relation will be so totally negative, but as the underlying set increases, the likelihood any of the properties are true begins to vanish.

6.3.4: Exercises

Exercise 6.3.1

a. Let $B = \{a, b\}$ and $U = \mathcal{P}(B)$. Draw a Hasse diagram for $\subseteq$ on U.
b. Let $A = \{1, 2, 3, 6\}$. Show that divides, $|$, is a partial ordering on A.
c. Draw a Hasse diagram for divides on A.
d. Compare the graphs of parts a and c.

Answer

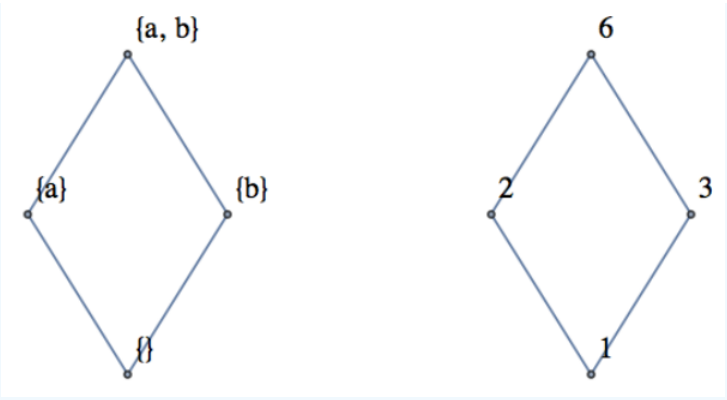


Figure 6.3.5

a. See Figure 6.3.5
b. The graphs are the same if we disregard the names of the vertices.

Exercise 6.3.2

Repeat Exercise 6.3.1 with $B = \{a, b, c\}$ and $A = \{1, 2, 3, 5, 6, 10, 15, 30\}$.

Hint

Here is a Hasse diagram for the part (a).

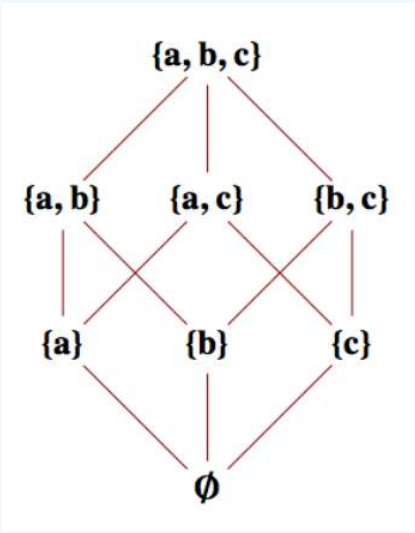


Figure 6.3.6 Hasse Diagram for a set containment for subsets of a three element set.

Exercise 6.3.3

Consider the relations defined by the digraphs in Figure 6.3.7.

a. Determine whether the given relations are reflexive, symmetric, antisymmetric, or transitive. Try to develop procedures for determining the validity of these properties from the graphs,
b. Which of the graphs are of equivalence relations or of partial orderings?

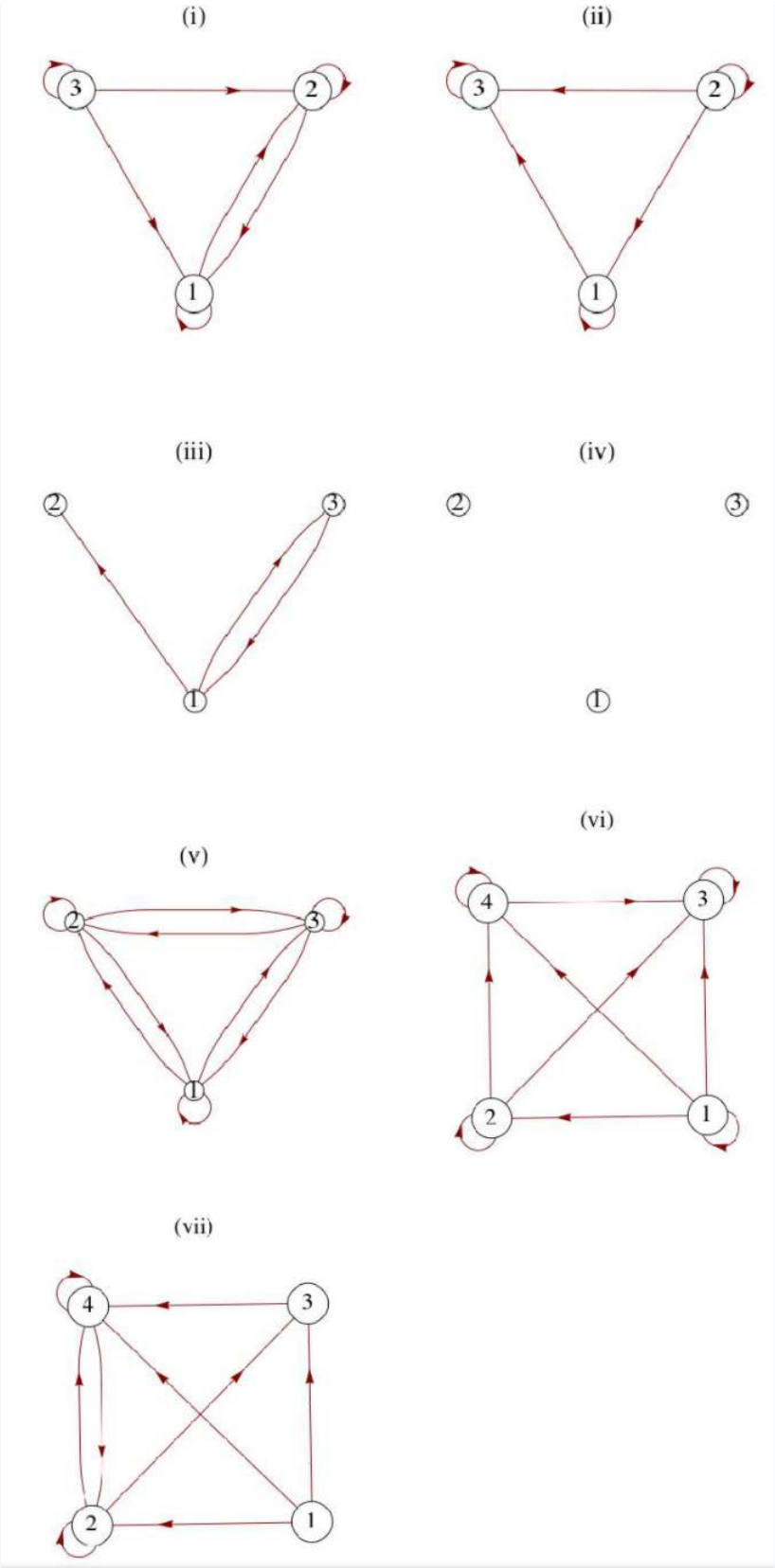


Figure 6.3.7: Some digraphs of relations

Answer

https://math.libretexts.org/@go/page/80523

Table 6.3.1: Properties of relations defined by digraphs

Part	reflexive?	symmetric?	antisymmetric?	transitive?
i	yes	no	no	yes
ii	yes	no	yes	yes
iii	no	no	no	no
iv	no	yes	yes	yes
v	yes	yes	no	yes
vi	yes	no	yes	yes
vii	no	no	no	no

i. See Table 6.3.1
ii. Graphs ii and vi show partial ordering relations. Graph v is of an equivalence relation.

Exercise 6.3.4

Determine which of the following are equivalence relations and/or partial ordering relations for the given sets:

a. $A = \{\text{ lines in the plane}\}$, and r defined by xry if and only if x is parallel to y. Assume every line is parallel to itself.
b. $A = \mathbb{R}$ and r defined by xry if and only if $|x - y| \leq 7$.

Exercise 6.3.5

Consider the relation on $\{1, 2, 3, 4, 5, 6\}$defined by $r = \{(i, j) : \ \ |i - j| = 2\}$.

a. Is r reflexive?
b. Is r symmetric?
c. Is r transitive?
d. Draw a graph of r.

Answer

1. No, since $|1 - 1 | = 0 \neq 2$, for example
2. Yes, because $|i - j | = |j - i|$.
3. No, since $|2 - 4 | = 2$ and $|4 - 6 | = 2$, but $|2 - 6 | = 4 \neq 2$, for example.
4. See Figure 6.3.8

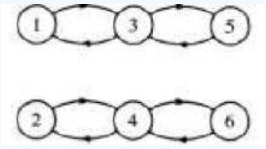


Figure 6.3.8

Exercise 6.3.6

For the set of cities on a map, consider the relation xry if and only if city x is connected by a road to city y. A city is considered to be connected to itself, and two cities are connected even though there are cities on the road between them. Is this an equivalence relation or a partial ordering? Explain.

Exercise 6.3.7: Equivalence Classes

Let $A = \{0, 1, 2, 3\}$ and let

$$r = \{(0,0), (1,1), (2,2), (3,3), (1,2), (2,1), (3,2), (2,3), (3,1), (1,3)\}$$

1. Verify that r is an equivalence relation on A.
2. Let $a \in A$ and define $c(a) = \{b \in A \mid arb\}$. $c(a)$ is called the ***equivalence class of*** a ***under*** r. Find $c(a)$ for each element $a \in A$.
3. Show that $\{c(a) \mid a \in A\}$ forms a partition of A for this set A.
4. Let r be an equivalence relation on an arbitrary set A. Prove that the set of all equivalence classes under r constitutes a partition of A.

Answer

a.

b. $c(0) = \{0\}, c(1) = \{1, 2, 3\} = c(2) = c(3)$

c. $c(0) \cup c(1) = A$ and $c(0) \cap c(1) = \emptyset$

d. Let A be any set and let r be an equivalence relation on A. Let a be any element of A. $a \in c(a)$ since r is reflexive, so each element of A is in some equivalence class. Therefore, the union of all equivalence classes equals A. Next we show that any two equivalence classes are either identical or disjoint and we are done. Let $c(a)$ and $c(b)$ be two equivalence classes, and assume that $c(a) \cap c(b) \neq \emptyset$. We want to show that $c(a) = c(b)$. To show that $c(a) \subseteq c(b)$, let $x \in c(a)$. $x \in c(a) \Rightarrow arx$. Also, there exists an element, y, of A that is in the intersection of $c(a)$ and $c(b)$ by our assumption. Therefore,

$$\begin{aligned} ary \wedge bry &\Rightarrow ary \wedge yrb \quad r \text{ is symmetric} \\ &\Rightarrow arb \quad \text{transitivity of } r \end{aligned}$$

Next,

$$\begin{aligned} arx \wedge arb &\Rightarrow xra \wedge arb \\ &\Rightarrow xrb \\ &\Rightarrow brx \\ &\Rightarrow x \in c(b) \end{aligned}$$

Similarly, $c(b) \subseteq c(a)$. $\square$

Exercise 6.3.8

Define r on the power set of $\{1, 2, 3\}$ by $ArB \Leftrightarrow |A| = |B|$. Prove that r is an equivalence relation. What are the equivalence classes under r?

Exercise 6.3.9

Consider the following relations on $\mathbb{Z}_8 = \{0, 1, \ldots, 7\}$. Which are equivalence relations? For the equivalence relations, list the equivalence classes.

a. arb iff the English spellings of a and b begin with the same letter.
b. asb iff $a - b$ is a positive integer.
c. atb iff $a - b$ is an even integer.

Answer

a. Equivalence Relation, $c(0) = \{0\}, c(1) = \{1\}, c(2) = \{2, 3\} = c(3), c(4) = \{4, 5\} = c(5)$,and $c(6) = \{6, 7\} = c(7)$
b. Not an Equivalence Relation.
c. Equivalence Relation, $c(0) = \{0, 2, 4, 6\} = c(2) = c(4) = c(6)$ and $c(1) = \{1, 3, 5, 7\} = c(3) = c(5) = c(7)$

Exercise 6.3.10

Building on Exercise 6.3.7:

a. Prove that congruence modulo m is transitive.
b. What are the equivalence classes under congruence modulo 2?
c. What are the equivalence classes under congruence modulo 10?

Exercise 6.3.11

In this exercise, we prove that implication is a partial ordering. Let A be any set of propositions.

a. Verify that $q \to q$ is a tautology, thereby showing that $\Rightarrow$ is a reflexive relation on A.
b. Prove that $\Rightarrow$ is antisymmetric on A. Note: we do not use = when speaking of propositions, but rather equivalence, $\Leftrightarrow$.
c. Prove that $\Rightarrow$ is transitive on A.
d. Given that q_i is the proposition $n < i$ on $\mathbb{N}$, draw the Hasse diagram for the relation $\Rightarrow$ on $\{q_1, q_2, q_3, \ldots\}$.

Answer

1. The proof follows from the biconditional equivalence in Table 3.4.3.
2. Apply the chain rule.
3. See Figure 6.3.9

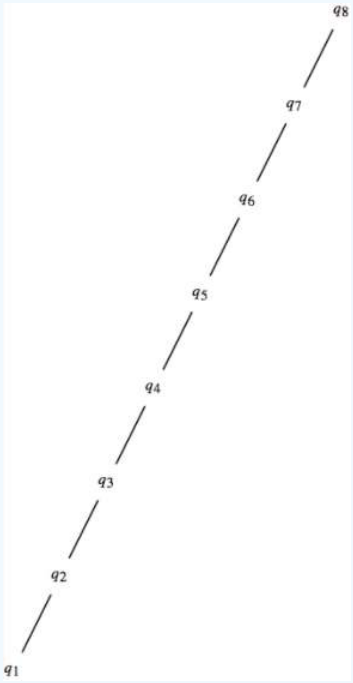


Figure 6.3.9

Exercise 6.3.12

Let $S = \{1, 2, 3, 4, 5, 6, 7\}$be a poset $(S, \leq)$ with the Hasse diagram shown below. Another relation $r \subseteq S \times S$ is defined as follows: $(x, y) \in r$ if and only if there exists $z \in S$ such that $z < x$ and $z < y$ in the poset $(S, \leq)$.

a. Prove that r is reflexive.
b. Prove that r is symmetric.

c. A compatible with respect to relation r is any subset Q of set S such that $x \in Q$ and $y \in Q \Rightarrow (x, y) \in r$. A compatible g is a maximal compatible if Q is not a proper subset of another compatible. Give all maximal compatibles with respect to relation r defined above.

d. Discuss a characterization of the set of maximal compatibles for relation r when $(S, \leq)$ is a general finite poset. What conditions, if any, on a general finite poset $(S, \leq)$ will make r an equivalence relation?

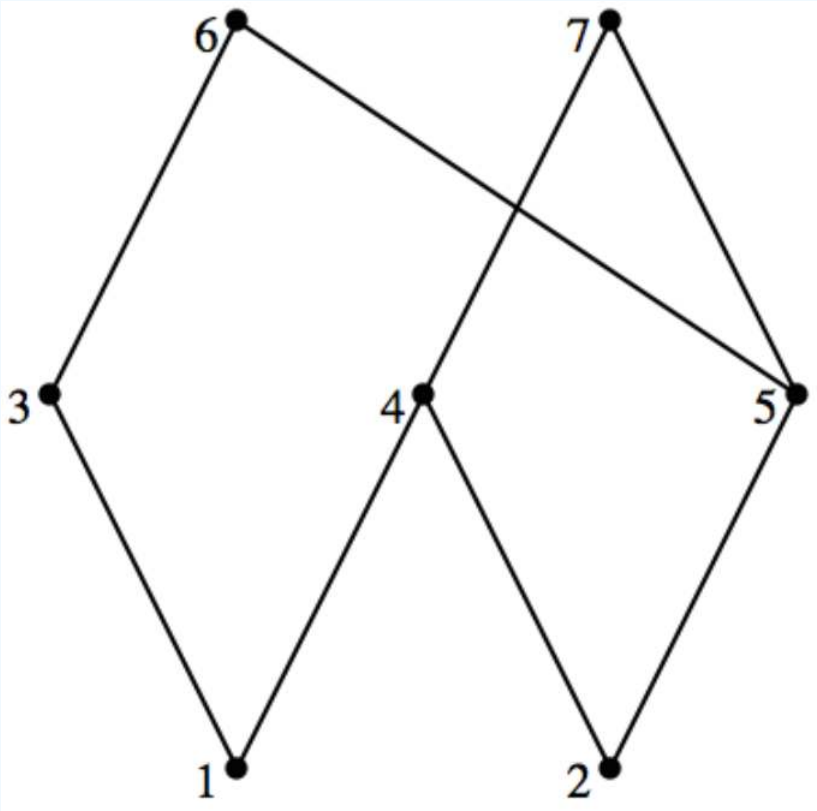


Figure 6.3.10: Hasse diagram for r in Exercise 6.3.12

This page titled 6.3: Properties of Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

6.4: Matrices of Relations

We have discussed two of the many possible ways of representing a relation, namely as a digraph or as a set of ordered pairs. In this section we will discuss the representation of relations by matrices.

6.4.1: Representing a Relation with a Matrix

Definition 6.4.1: Adjacency Matrix

Let $A=\{a_1,a_2,\ldots,a_m\}$ and $B=\{b_1,b_2,\ldots,b_n\}$ be finite sets of cardinality m and n, respectively. Let r be a relation from A into B. Then r can be represented by the $m\times n$ matrix R defined by

$$R_{ij}=\begin{cases}1 & \text{if } a_i r b_j\\ 0 & \text{otherwise}\end{cases}$$

R is called the ***adjacency matrix*** (or the relation matrix) of r.

Example 6.4.1: A Simple Example

Let $A=\{2,5,6\}$ and let r be the relation $\{(2,2),(2,5),(5,6),(6,6)\}$on A. Since r is a relation from A into the same set A (the B of the definition), we have $a_1=2,\ a_2=5,$ and $a_3=6,$ while $b_1=2,\ b_2=5,$ and $b_3=6$. Next, since

- $2r2$, we have $R_{11}=1$
- $2r5$, we have $R_{12}=1$
- $5r6$, we have $R_{23}=1$
- $6r6$, we have $R_{33}=1$

All other entries of R are zero, so

$$R=\begin{pmatrix}1&1&0\\0&0&1\\0&0&1\end{pmatrix}$$

6.4.2: Composition as Matrix Multiplication

From the definition of r and of composition, we note that

$$r^2=\{(2,2),(2,5),(2,6),(5,6),(6,6)\}$$

The adjacency matrix of r^2 is

$$R^2=\begin{pmatrix}1&1&1\\0&0&1\\0&0&1\end{pmatrix}.$$

We do not write R^2 only for notational purposes. In fact, R^2 can be obtained from the matrix product RR; however, we must use a slightly different form of arithmetic.

Definition 6.4.2: Boolean Arithmetic

Boolean arithmetic is the arithmetic defined on $\{0,1\}$ using Boolean addition and Boolean multiplication, defined by

Table 6.4.1

$0+0=0$	$0+1=1+0=1$	$1+1=1$
$0\cdot 0=0$	$0\cdot 1=1\cdot 0=0$	$1\cdot 1=1$

Notice that from Chapter 3, this is the "arithmetic of logic," where $+$ replaces "or" and $\cdot$ replaces "and."

Example 6.4.2: Composition by Multiplication

Suppose that $R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ and $S = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$. Then using Boolean arithmetic, $RS = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ and $SR = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$.

Theorem 6.4.1: Composition is Matrix Multiplication

Let A_1, A_2, and A_3 be finite sets where r_1 is a relation from A_1 into A_2 and r_2 is a relation from A_2 into A_3. If R_1 and R_2 are the adjacency matrices of r_1 and r_2, respectively, then the product R_1R_2 using Boolean arithmetic is the adjacency matrix of the composition r_1r_2.

Remark: A convenient help in constructing the adjacency matrix of a relation from a set A into a set B is to write the elements from A in a column preceding the first column of the adjacency matrix, and the elements of B in a row above the first row. Initially, R in Example 6.4.1 would be

$$\begin{array}{c} \\ 2 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 5 & 6 \end{array} \\ \begin{pmatrix} & & \end{pmatrix} \end{array}$$

To fill in the matrix, R_{ij} is 1 if and only if $(a_i, b_j) \in r$. So that, since the pair $(2, 5) \in r$, the entry of R corresponding to the row labeled 2 and the column labeled 5 in the matrix is a 1.

Example 6.4.3: Relations and Information

This final example gives an insight into how relational data base programs can systematically answer questions pertaining to large masses of information. Matrices R (on the left) and S (on the right) define the relations r and s where arb if software a can be run with operating system b, and bsc if operating system b can run on computer c.

$$\begin{array}{c} \\ \text{P1} \\ \text{P2} \\ \text{P3} \\ \text{P4} \end{array} \begin{array}{c} \begin{array}{cccc} \text{OS1} & \text{OS2} & \text{OS3} & \text{OS4} \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ \text{OS1} \\ \text{OS2} \\ \text{OS3} \\ \text{OS4} \end{array} \begin{array}{c} \begin{array}{ccc} \text{C1} & \text{C2} & \text{C3} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

Although the relation between the software and computers is not implicit from the data given, we can easily compute this information. The matrix of rs is RS, which is

$$\begin{array}{c} \\ \text{P1} \\ \text{P2} \\ \text{P3} \\ \text{P4} \end{array} \begin{array}{c} \begin{array}{ccc} \text{C1} & \text{C2} & \text{C3} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

This matrix tells us at a glance which software will run on the computers listed. In this case, all software will run on all computers with the exception of program P2, which will not run on the computer C3, and programs P3 and P4, which will not run on the computer C1.

6.4.3: Exercises

Exercise 6.4.1

Let $A_1 = \{1, 2, 3, 4\}$, $A_2 = \{4, 5, 6\}$, and $A_3 = \{6, 7, 8\}$. Let r_1 be the relation from A_1 into A_2 defined by $r_1 = \{(x, y) \mid y - x = 2\}$, and let r_2 be the relation from A_2 into A_3 defined by $r_2 = \{(x, y) \mid y - x = 1\}$.

a. Determine the adjacency matrices of r_1 and r_2.
b. Use the definition of composition to find $r_1 r_2$.
c. Verify the result in part b by finding the product of the adjacency matrices of r_1 and r_2.

Answer

a.
$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccc} 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \text{ and } \begin{array}{c} \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{ccc} 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

b. $r_1 r_2 = \{(3, 6), (4, 7)\}$

c.
$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccc} 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

Exercise 6.4.2

a. Determine the adjacency matrix of each relation given via the digraphs in Exercise 6.3.3 of Section 6.3.
b. Using the matrices found in part (a) above, find r^2 of each relation in Exercise 6.3.3 of Section 6.3.
c. Find the digraph of r^2 directly from the given digraph and compare your results with those of part (b).

Exercise 6.4.3

Suppose that the matrices in Example 6.4.2 are relations on $\{1, 2, 3, 4\}$. What relations do R and S describe?

Answer

Table 6.4.2

R : xry if and only if $\lvert x - y \rvert = 1$
S : xsy if and only if x is less than y.

Exercise 6.4.4

Let D be the set of weekdays, Monday through Friday, let W be a set of employees $\{1, 2, 3\}$ of a tutoring center, and let V be a set of computer languages for which tutoring is offered, $\{A(PL), B(asic), C(++), J(ava), L(isp), P(ython)\}$. We define s (schedule) from D into W by dsw if w is scheduled to work on day d. We also define r from W into V by wrl if w can tutor students in language l. If s and r are defined by matrices

$$S = \begin{array}{c} \\ M \\ T \\ W \\ R \\ F \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \end{array} \text{ and } R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cccccc} A & B & C & J & L & P \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

a. compute SR using Boolean arithmetic and give an interpretation of the relation it defines, and
b. compute SR using regular arithmetic and give an interpretation of what the result describes.

Exercise 6.4.5

How many different reflexive, symmetric relations are there on a set with three elements?

Hint

Consider the possible matrices.

Answer

The diagonal entries of the matrix for such a relation must be 1. When the three entries above the diagonal are determined, the entries below are also determined. Therefore, there are 2^3 fitting the description.

Exercise 6.4.6

Let $A = \{a, b, c, d\}$. Let r be the relation on A with adjacency matrix $\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$

a. Explain why r is a partial ordering on A.
b. Draw its Hasse diagram.

Exercise 6.4.7

Define relations p and q on $\{1, 2, 3, 4\}$ by $p = \{(a, b) \mid |a - b| = 1\}$ and $q = \{(a, b) \mid a - b \text{ is even}\}$.

a. Represent p and q as both graphs and matrices.
b. Determine pq, p^2, and q^2; and represent them clearly in any way.

Answer

a. $\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$ and $\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$

b. $PQ = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$ $P^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} = Q^2$

Exercise 6.4.8

a. Prove that if r is a transitive relation on a set A, then $r^2 \subseteq r$.
b. Find an example of a transitive relation for which $r^2 \neq r$.

Exercise 6.4.9

We define $\leq$ on the set of all $n \times n$ relation matrices by the rule that if R and S are any two $n \times n$ relation matrices, $R \leq S$ if and only if $R_{ij} \leq S_{ij}$ for all $1 \leq i, j \leq n$.

a. Prove that $\leq$ is a partial ordering on all $n \times n$ relation matrices.
b. Prove that $R \leq S \Rightarrow R^2 \leq S^2$, but the converse is not true.
c. If R and S are matrices of equivalence relations and $R \leq S$, how are the equivalence classes defined by R related to the equivalence classes defined by S?

Answer

a. Reflexive: $R_{ij} = R_{ij}$ for all i, j, therefore $R_{ij} \leq R_{ij}$
Antisymmetric: Assume $R_{ij} \leq S_{ij}$ and $S_{ij} \leq R_{ij}$ for all $1 \leq i, j \leq n$. Therefore, $R_{ij} = S_{ij}$ for all $1 \leq i, j \leq n$ and so $R = S$
Transitive: Assume R, S, and T are matrices where $R_{ij} \leq S_{ij}$ and $S_{ij} \leq T_{ij}$, for all $1 \leq i, j \leq n$. Then $R_{ij} \leq T_{ij}$ for all $1 \leq i, j \leq n$, and so $R \leq T$.

b.
$$\begin{aligned}(R^2)_{ij} &= R_{i1}R_{1j} + R_{i2}R_{2j} + \cdots + R_{in}R_{nj} \\ &\leq S_{i1}S_{1j} + S_{i2}S_{2j} + \cdots + S_{in}S_{nj} \\ &= (S^2)_{ij} \Rightarrow R^2 \leq S^2\end{aligned}$$

To verify that the converse is not true we need only one example. For $n = 2$, let $R_{12} = 1$ and all other entries equal 0, and let S be the zero matrix. Since R^2 and S^2 are both the zero matrix, $R^2 \leq S^2$, but since $R_{12} > S_{12}$, $R \leq S$ is false.

c. The matrices are defined on the same set $A = \{a_1,\ a_2, \cdots, a_n\}$. Let $c(a_i)$, $i = 1,\ 2, \cdots, n$ be the equivalence classes defined by R and let $d(a_i)$ be those defined by S. Claim: $c(a_i) \subseteq d(a_i)$.

$$\begin{aligned}a_j \in c(a_i) &\Rightarrow a_i r a_j \\ &\Rightarrow R_{ij} = 1 \Rightarrow S_{ij} = 1 \\ &\Rightarrow a_i s a_j \\ &\Rightarrow a_j \in d(a_i)\end{aligned}$$

This page titled 6.4: Matrices of Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

6.5: Closure Operations on Relations

In Section 6.1, we studied relations and one important operation on relations, namely composition. This operation enables us to generate new relations from previously known relations. In Section 6.3, we discussed some key properties of relations. We now wish to consider the situation of constructing a new relation r^+ from an existing relation r where, first, r^+ contains r and, second, r^+ satisfies the transitive property.

6.5.1: Transitive Closure

Consider a telephone network in which the main office a is connected to, and can communicate to, individuals b and c. Both b and c can communicate to another person, d; however, the main office cannot communicate with d. Assume communication is only one way, as indicated. This situation can be described by the relation $r = \{(a,b),(a,c),(b,d),(c,d)\}$. We would like to change the system so that the main office a can communicate with person d and still maintain the previous system. We, of course, want the most economical system.

This can be rephrased as follows; Find the smallest relation r^+ which contains r as a subset and which is transitive; $r^+ = \{(a,b),(a,c),(b,d),(c,d),(a,d)\}$.

Definition 6.5.1: Transitive Closure

Let A be a set and r be a relation on A. The transitive closure of r, denoted by r^+, is the smallest transitive relation that contains r as a subset.

Let $A = \{1,2,3,4\}$, and let $\mathcal{S} = \{(1,2),(2,3),(3,4)\}$ be a relation on A. This relation is called the successor relation on A since each element is related to its successor. How do we compute $\mathcal{S}^+$? By inspection we note that $(1,3)$ must be in $\mathcal{S}^+$. Let's analyze why. This is so because $(1,2) \in \mathcal{S}$ and $(2,3) \in \mathcal{S}$, and the transitive property forces $(1,3)$ to be in $\mathcal{S}^+$.

In general, it follows that if $(a,b) \in \mathcal{S}$ and $(b,c) \in S$, then $(a,c) \in \mathcal{S}^+$. This condition is exactly the membership requirement for the pair (a,c) to be in the composition $\mathcal{S}\mathcal{S} = \mathcal{S}^2$. So every element in $\mathcal{S}^2$ must be an element in $\mathcal{S}^+$. So we now know that, $\mathcal{S}^+$ contains at least $\mathcal{S} \cup \mathcal{S}^2$. In particular, for this example, since $\mathcal{S} = \{(1,2),(2,3),(3,4)\}$ and $\mathcal{S}^2 = \{(1,3),(2,4)\}$, we have

$$\mathcal{S} \cup \mathcal{S}^2 = \{(1,2),(2,3),(3,4),(1,3),(2,4)\}$$

Is the relation $\mathcal{S} \cup \mathcal{S}^2$ transitive? Again, by inspection, $(1,4)$ is not an element of $\mathcal{S} \cup \mathcal{S}^2$, but $(1,3) \in \mathcal{S}^2$ and $(3,4) \in \mathcal{S}$. Therefore, the composition $\mathcal{S}^2\mathcal{S} = \mathcal{S}^3$ produces $(1,4)$, and it must be an element of $\mathcal{S}^+$ since $(1,3)$ and $(3,4)$ are required to be in $\mathcal{S}^+$. This shows that $\mathcal{S}^3 \subseteq \mathcal{S}^+$. This process must be continued until the resulting relation is transitive. If A is finite, as is true in this example, the transitive closure will be obtained in a finite number of steps. For this example,

$$\mathcal{S}^+ = \mathcal{S} \cup \mathcal{S}^2 \cup \mathcal{S}^3 = \{(1,2),(2,3),(3,4),(1,3),(2,4),(1,4)\}$$

Theorem 6.5.1: Transitive Closure on a Finite Set

If r is a relation on a set A and $|A| = n$, then the transitive closure of r is the union of the first n powers of r. That is,

$$r^+ = r \cup r^2 \cup r^3 \cup \cdots \cup r^n.$$

Let's now consider the matrix analogue of the transitive closure.

Consider the relation

$$r = \{(1,4),(2,1),(2,2),(2,3),(3,2),(4,3),(4,5),(5,1)\}$$

on the set $A = \{1,2,3,4,5\}$. The matrix of r is

$$R = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Recall that $r^2, r^3, \ldots$ can be determined through computing the matrix powers $R^2, R^3, \ldots$. For our example,

Table 6.5.1

$R^2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	$R^3 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$
$R^4 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$	$R^5 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$

How do we relate $\bigcup_{i=1}^{5} r^i$ to the powers of R?

Theorem 6.5.2: Matrix of a Transitive Closure

Let r be a relation on a finite set and R its matrix. Let R^+ be the matrix of r^+, the transitive closure of r. Then $R^+ = R + R^2 + \cdots + R^n$, using Boolean arithmetic.

Using this theorem, we find R^+ is the 5×5 matrix consisting of all $1's$, thus, r^+ is all of $A \times A$.

6.5.2: Algorithms for computing transitive closure

Let r be a relation on the set $\{1, 2, \ldots, n\}$with relation matrix R. The matrix of the transitive closure R^+, can be computed by the equation $R^+ = R + R^2 + \cdots + R^n$. By using ordinary polynomial evaluation methods, you can compute R^+ with $n-1$ matrix multiplications:

$$R^+ = R(I + R(I + (\cdots R(I + R)\cdots)))$$

For example, if $n = 3$, $R^+ = R(I + R(I + R))$.

We can make use of the fact that if T is a relation matrix, $T + T = T$ due to the fact that $1 + 1 = 1$ in Boolean arithmetic. Let $S_k = R + R^2 + \cdots + R^k$. Then

$$\begin{aligned} R &= S_1 \\ S_1(I + S_1) &= R(I + R) = R + R^2 = S_2 \\ S_2(I + S_2) &= (R + R^2)(I + R + R^2) \\ &= (R + R^2) + (R^2 + R^3) + (R^3 + R^4) \\ &= R + R^2 + R^3 + R^4 = S_4 \end{aligned} .$$

Similarly,

$$S_4(I + S_4) = S_8$$

and by induction we can prove

$$S_{2^k}(I + S_{2^k}) = S_{2^{k+1}}$$

Notice how each matrix multiplication doubles the number of terms that have been added to the sum that you currently have computed. In algorithmic form, we can compute R^+ as follows.

Algorithm 6.5.1: Transitive Closure Algorithm

Let R be a relation matrix and let R^+ be its transitive closure matrix, which is to be computed as matrix T

```
1.0. S=R
2.0  T=S*(I+S)
3.0 While T != S
        3.1 S = T
        3.2 T= S*(I+S) // using Boolean arithmetic
4.0 Return T
```

Listing 6.5.1

Note 6.5.1

- Often the higher-powered terms in S_n do not contribute anything to R^+. When the condition $T = S$ becomes true in Step 3, this is an indication that no higher-powered terms are needed.
- To compute R^+ using this algorithm, you need to perform no more than $\lceil \log_2 n \rceil$ matrix multiplications, where $\lceil x \rceil$ is the least integer that is greater than or equal to x. For example, if r is a relation on 25 elements, no more than $\lceil \log_2 25 \rceil = 5$ matrix multiplications are needed.

A second algorithm, Warshall's Algorithm, reduces computation time to the time that it takes to multiply two square matrices with the same order as the relation matrix in question.

Algorithm 6.5.2: Warshall's Algorithm

Let R be an $n \times n$ relation matrix and let R^+ be its transitive closure matrix, which is to be computed as matrix T using Boolean arithmetic

```
1.0 T = R
2.0 for k = 1 to n:
     for i = 1 to n:
      for j = 1 to n:
        T[i,j]= T[i,j] + T[i,k] * T[k,j]
3.0 Return T
```

Listing 6.5.2

6.5.3: Exercises

Exercise 6.5.1

Let $A = \{0, 1, 2, 3, 4\}$ and $\mathcal{S} = \{(0, 1), (1, 3), (2, 3), (3, 4), (4, 1)\}$.Compute $\mathcal{S}^+$ using the matrix representation of $\mathcal{S}$. Verify your results by checking against the result obtained directly from the definition of transitive closure.

Exercise 6.5.2

Let $A = \{1, 2, 3, 4, 6, 12\}$and $t = \{(a, b) \mid b/a \text{ is a prime number}\}$. Determine t^+ by any means. Represent your answer as a matrix.

Exercise 6.5.3

a. Draw digraphs of the relations $\mathcal{S}$, $\mathcal{S}^2$, $\mathcal{S}^3$, and $\mathcal{S}^+$ where $\mathcal{S}$ is defined in the first exercise above.
b. Verify that in terms of the graph of $\mathcal{S}$, $a\mathcal{S}^+b$ if and only if b is reachable from a along a path of any finite nonzero length.

Answer

1. See graphs below.
2. For example, $0s^+4$and using S one can go from 0 to 4 using a path of length 3.

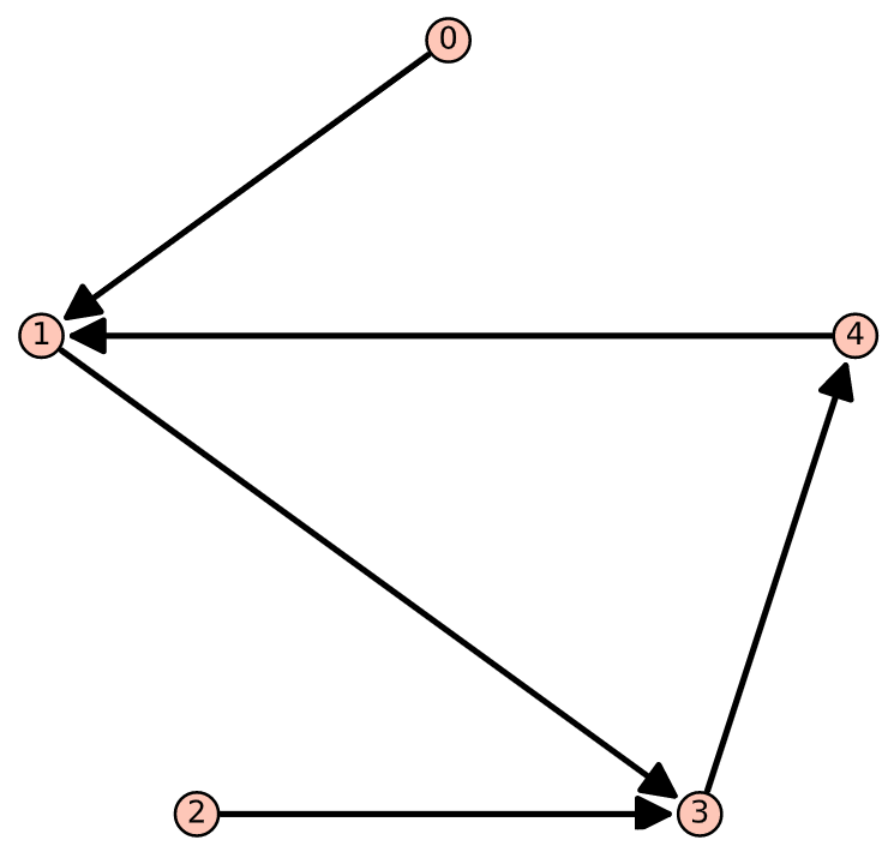


Figure 6.5.1: Digraph of S

https://math.libretexts.org/@go/page/80525

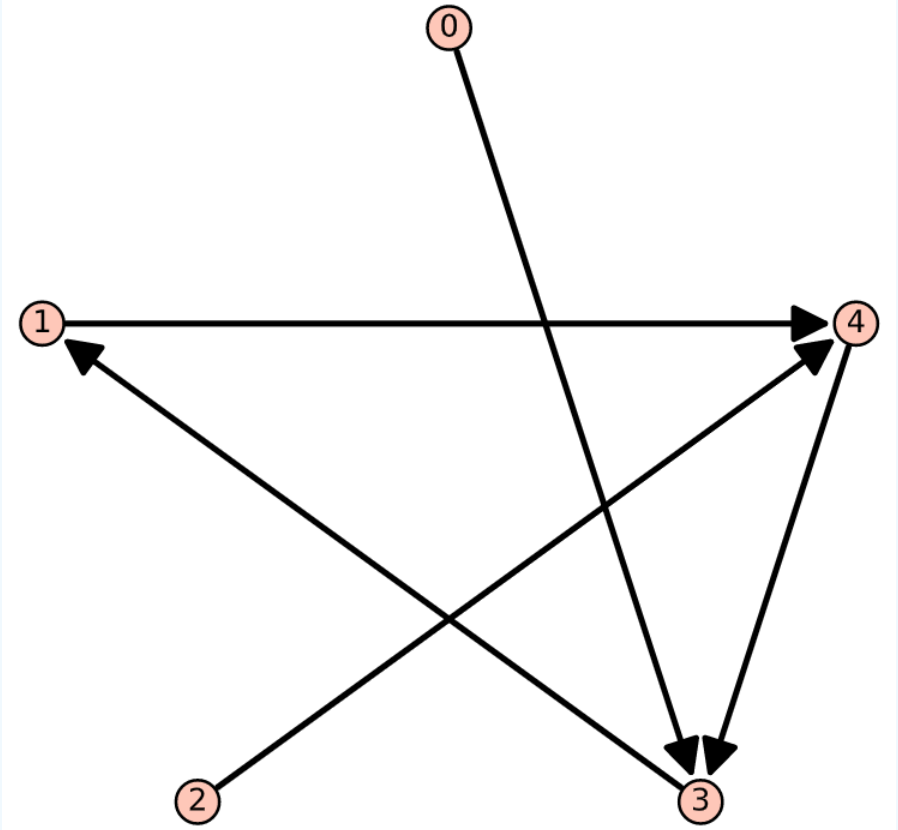


Figure 6.5.2 Digraph of S^2

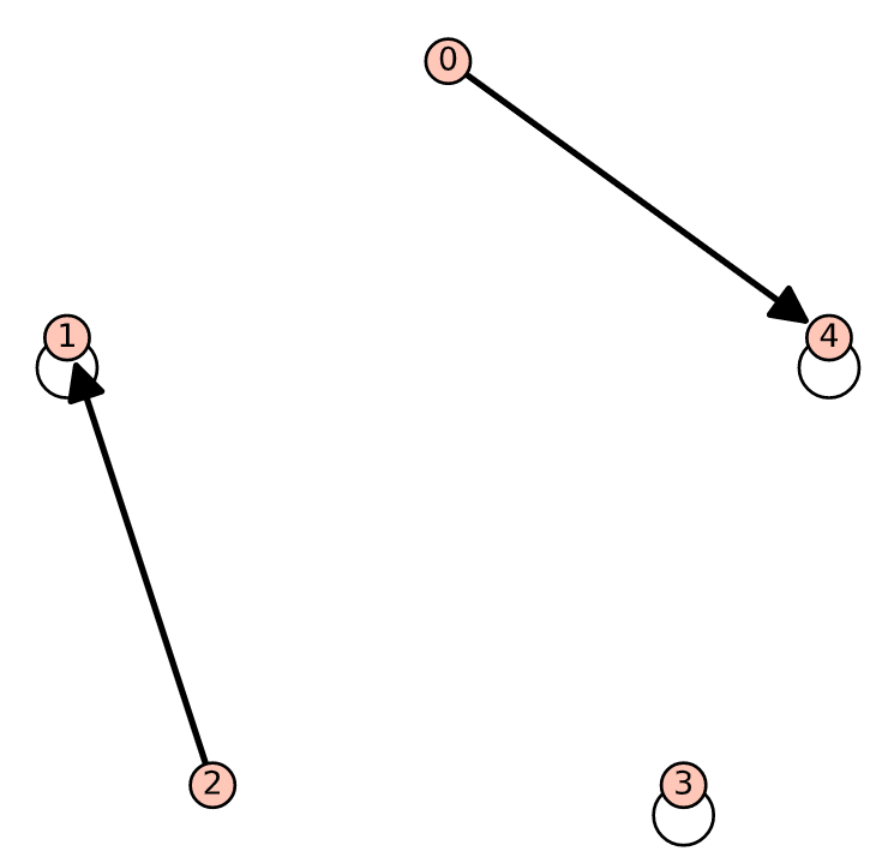


Figure 6.5.3 Digraph of S^3

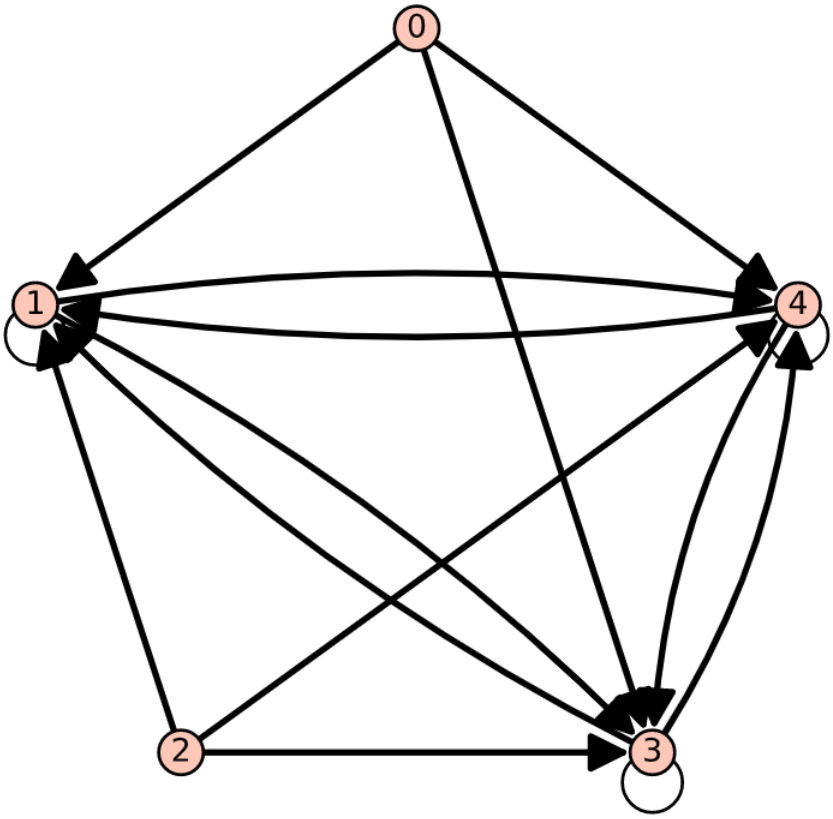


Figure 6.5.4 Digraph of S^+

Exercise 6.5.4

Let r be the relation represented by the following digraph.

a. Find r^+ using the definition based on order pairs.
b. Determine the digraph of r^+ directly from the digraph of r.
c. Verify your result in part (b) by computing the digraph from your result in part (a).

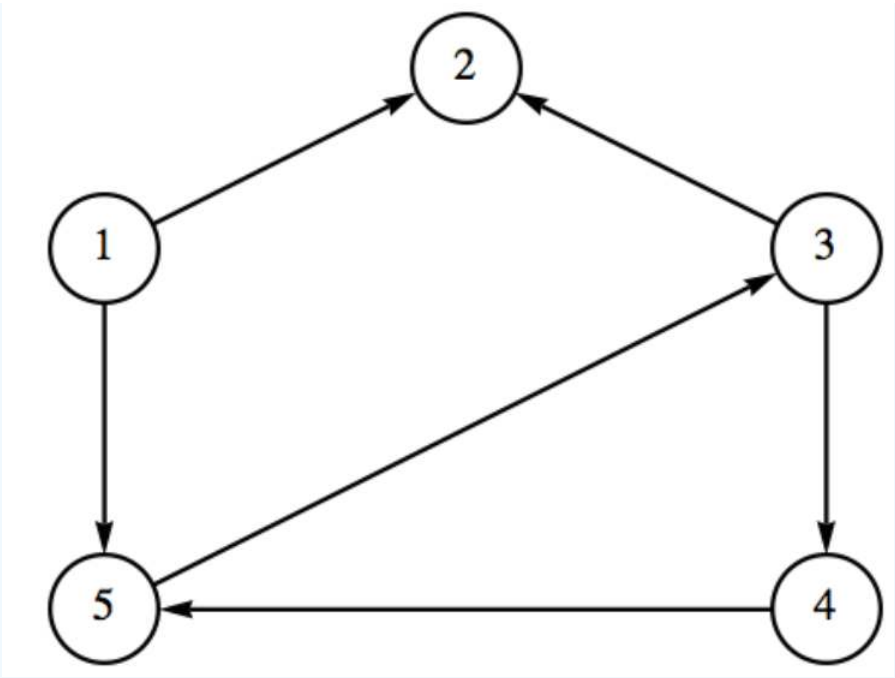


Figure 6.5.5: Digraph of r in Exercise 6.5.4

Exercise 6.5.5

a. Define reflexive closure and symmetric closure by imitating the definition of transitive closure.
b. Use your definitions to compute the reflexive and symmetric closures of examples in the text.
c. What are the transitive reflexive closures of these examples?
d. Convince yourself that the reflexive closure of the relation $<$ on the set of positive integers $\mathbb{P}$ is $\leq$.

Answer

Definition: Reflexive Closure. Let r be a relation on A. The reflexive closure of r is the smallest reflexive relation that contains r.

Theorem: The reflexive closure of r is the union of r with $\{(x, x) : x \in A\}$

Exercise 6.5.6

What common relations on $\mathbb{Z}$ are the transitive closures of the following relations?

a. aSb if and only if $a+1=b$.
b. aRb if and only if $|a-b|=2$.

Exercise 6.5.7

a. Let A be any set and r a relation on A, prove that $(r^+)^+ = r^+$.
b. Is the transitive closure of a symmetric relation always both symmetric and reflexive? Explain.

Answer

a. By the definition of transitive closure, r^+ is the smallest relation which contains r; therefore, it is transitive. The transitive closure of r^+, $(r^+)^+$, is the smallest transitive relation that contains r^+. Since r^+ is transitive, $(r^+)^+ = r^+$.
b. The transitive closure of a symmetric relation is symmetric, but it may not be reflexive. If one element is not related to any elements, then the transitive closure will not relate that element to others.

Exercise 6.5.8

The definition of the Transitive Closure, Definition 6.5.1, of r refers to the "smallest transitive relation that contains r as a subset." Show that the intersection of all transitive relations on A containing r is a transitive relation containing r and is precisely r^+.

This page titled 6.5: Closure Operations on Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

7: Functions

> ***countably infinite***
>
> *A **countably infinite** set*
> *Is as simple as things like this get.*
> *Just start counting at 1,*
> *Then continue—it's fun!*
> *I'll check back when you're done, so don't sweat.*
>
> *Chris Doyle, The Omnificent English Dictionary in Limerick Form*

In this chapter we will consider some basic concepts of the relations that are called functions. A large variety of mathematical ideas and applications can be more completely understood when expressed through the function concept.

7.1: Definition and Notation
7.2: Properties of Functions
7.3: Function Composition

This page titled 7: Functions is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

7.1: Definition and Notation

7.1.1: Fundamentals

Definition 7.1.1: Function

A function from a set A into a set B is a relation from A into B such that each element of A is related to exactly one element of the set B. The set A is called the ***domain*** of the function and the set B is called the ***codomain***.

The reader should note that a function f is a relation from A into B with two important restrictions:

- Each element in the set A, the domain of f, must be related to some element of B, the codomain.
- The phrase "is related to exactly one element of the set B" means that if $(a, b) \in f$ and $(a, c) \in f$, then $b = c$.

Example 7.1.1: A Function as a List of Ordered Pairs

Let $A = \{-2, -1, 0, 1, 2\}$ and $B = \{0, 1, 2, 3, 4\}$, and if $s = \{(-2, 4), (-1, 1), (0, 0), (1, 1), (2, 4)\}$, then s is a function from A into B.

Example 7.1.2: A Function as a Set of Ordered Pairs in Set-Builder Notation

Let $\mathbb{R}$ be the real numbers. Then $L = \{(x, 3x) \mid x \in \mathbb{R}\}$ is a function from $\mathbb{R}$ into $\mathbb{R}$, or, more simply, L is a function on $\mathbb{R}$.

It is customary to use a different system of notation for functions than the one we used for relations. If f is a function from the set A into the set B, we will write $f : A \to B$.

The reader is probably more familiar with the notation for describing functions that is used in basic algebra or calculus courses. For example, $y = \frac{1}{x}$ or $f(x) = \frac{1}{x}$ both define the function $\left\{\left(x, \frac{1}{x}\right) \middle| x \in \mathbb{R}, x \neq 0\right\}$. Here the domain was assumed to be those elements of $\mathbb{R}$ whose substitutions for x make sense, the nonzero real numbers, and the codomain was assumed to be $\mathbb{R}$. In most cases, we will make a point of listing the domain and codomain in addition to describing what the function does in order to define a function.

The terms ***mapping***, ***map***, and ***transformation*** are also used for functions.

Definition 7.1.2: The Set of Functions Between Two Sets

Given two sets, A and B, the set of all functions from A into B is denoted B^A.

The notation used for sets of functions makes sense in light of Exercise 7.1.5.

One way to imagine a function and what it does is to think of it as a machine. The machine could be mechanical, electronic, hydraulic, or abstract. Imagine that the machine only accepts certain objects as raw materials or input. The possible raw materials make up the domain. Given some input, the machine produces a finished product that depends on the input. The possible finished products that we imagine could come out of this process make up the codomain.

Example 7.1.3: A Definition Based on Images

We can define a function based on specifying the codomain element to which each domain element is related. For example, $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2$ is an alternate description of $f = \left\{\left(x, x^2\right) \middle| x \in \mathbb{R}\right\}$.

Definition 7.1.3: Image of an Element Under a Function

Let $f : A \to B$, read "Let f be a function from the set A into the set B." If $a \in A$, then $f(a)$ is used to denote that element of B to which a is related. $f(a)$ is called the *image* of a, or, more precisely, the image of a under f. We write $f(a) = b$ to indicate that the image of a is b.

In Example 7.1.3, the image of 2 under f is 4; that is, $f(2)=4$. In Example 7.1.1, the image of -1 under s is 1; that is, $s(-1)=1$.

Definition 7.1.4: Range of a Function

The range of a function is the set of images of its domain. If $f: X \to Y$, then the range of f is denoted $f(X)$, and

$$f(X)=\{f(a) \mid a \in X\}=\{b \in Y \mid \exists a \in X \text{ such that } f(a)=b\}.$$

Note that the range of a function is a subset of its codomain. $f(X)$ is also read as "the image of the set X under the function f" or simply "the image of f."

In Example 7.1.1, $s(A)=\{0,1,4\}$. Notice that 2 and 3 are not images of any element of A. In addition, note that both 1 and 4 are related to more than one element of the domain: $s(1)=s(-1)=1$ and $s(2)=s(-2)=4$. This does not violate the definition of a function. Go back and read the definition if this isn't clear to you.

In Example 7.1.2, the range of L is equal to its codomain, $\mathbb{R}$. If b is any real number, we can demonstrate that it belongs to $L(\mathbb{R})$ by finding a real number x for which $L(x)=b$. By the definition of L, $L(x)=3x$, which leads us to the equation $3x=b$. This equation always has a solution, $\frac{b}{3}$; thus $L(\mathbb{R})=\mathbb{R}$.

The formula that we used to describe the image of a real number under L, $L(x)=3x$, is preferred over the set notation for L due to its brevity. Any time a function can be described with a rule or formula, we will use this form of description. In Example 7.1.1, the image of each element of A is its square. To describe that fact, we write $s(a)=a^2 \quad (a \in A)$, or $S: A \to B$ defined by $S(a)=a^2$.

There are many ways that a function can be described. Many factors, such as the complexity of the function, dictate its representation.

Example 7.1.4: Data as a Function

Suppose a survey of 1,000 persons is done asking how many hours of television each watches per day. Consider the function $W: \{0,1,\ldots,24\} \to \{0,1,2,\ldots,1000\}$ defined by

$$W(t)=\text{the number of persons who gave a response of } t \text{ hours}$$

This function will probably have no formula such as the ones for s and L above.

Example 7.1.5: Conditional Definition of a Function

Consider the function $m: \mathbb{P} \to \mathbb{Q}$ defined by the set

$$m=\{(1,1),(2,1/2),(3,9),(4,1/4),(5,25),\ldots\}$$

No simple single formula could describe m, but if we assume that the pattern given continues, we can write

$$m(x)=\begin{cases} x^2 & \text{if } x \text{ is odd} \\ 1/x & \text{if } x \text{ is even} \end{cases}$$

7.1.2: Functions of Two Variables

If the domain of a function is the Cartesian product of two sets, then our notation and terminology changes slightly. For example, consider the function $G: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined by $G((n_1,n_2))=n_1^2+n_2^2-n_1n_2+10$. For this function, we would drop one set of parentheses and write $G(4,2)=22$, not $G((4,2))=22$. We call G a function of two variables. From one point of view, this function is no different from any others that we have seen. The elements of the domain happen to be slightly more complicated. On the other hand, we can look at the individual components of the ordered pairs as being separate. If we interpret G as giving us the cost of producing quantities of two products, we can imagine varying n_1 while n_2 is fixed, or vice versa.

The same observations can be made for function of three or more variables.

7.1.3: SageMath Note

There are several ways to define a function in Sage. The simplest way to implement f is as follows.

```
f(x)=x^2
f
```

```
[f(4), f(1.2)]
```

Sage is built upon the programming language Python, which is a *strongly typed language* and so you can't evaluate expressions such as `f('Hello')`. However a function such as f, as defined above, will accept any type of number, so a bit more work is needed to restrict the inputs of f to the integers.

A second way to define a function in Sage is based on Python syntax.

```
def fa(x)
    return x^2

#end of definition - now we test it:
[fa(2),fa(1.2)]
```

7.1.4: Non-Functions

We close this section with two examples of relations that are not functions.

Example 7.1.6: A Non-Function

Let $A = B = \{1, 2, 3\}$ and let $f = \{(1, 2), (2, 3)\}$. Here f is not a function from A into B since f does not act on, or "use," all elements of A.

Example 7.1.7: Another Non-Function

Let $A = B = \{1, 2, 3\}$ and let $g = \{(1, 2), (2, 3), (2, 1), (3, 2)\}$. We note that g acts on all of A. However, g is still not a function since $(2, 3) \in g$ and $(2, 1) \in g$ and the condition on each domain being related to exactly one element of the codomain is violated.

7.1.5: Exercises

Exercise 7.1.1

Let $A = \{1, 2, 3, 4\}$ and $B = \{a, b, c, d\}$. Determine which of the following are functions. Explain.

a. $f \subseteq A \times B$, where $f = \{(1, a), (2, b), (3, c), (4, d)\}$.
b. $g \subseteq A \times B$, where $g = \{(1, a), (2, a), (3, b), (4, d)\}$.
c. $h \subseteq A \times B$, where $h = \{(1, a), (2, b), (3, c)\}$.
d. $k \subseteq A \times B$, where $k = \{(1, a), (2, b), (2, c), (3, a), (4, a)\}$.
e. $L \subseteq A \times A$, where $L = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$.

Answer

a. Yes
b. Yes
c. No
d. No
e. Yes

Exercise 7.1.2

Find the ranges of the following functions on $\mathbb{Z}$:

a. $g = \{(x, 4x+1) | x \in \mathbb{Z}\}$.
b. $h(x) =$ the least integer that is greater than or equal to $\sqrt{|x|}$.
c. $P(x) = x + 10$.

Exercise 7.1.3

Find the ranges of each of the relations that are functions in Exercise 7.1.1.

Answer

a. Range of $f = f(A) = \{a, b, c, d\} = B$
b. Range of $g = g(A) = \{a, b, d\}$
c. h is not a function.
d. k is not a function.
e. Range of $L = L(A) = \{1\}$

Exercise 7.1.4

Let A be a set and let S be any subset of A. Let $\chi_S : A \to \{0, 1\}$ be defined by

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S \end{cases}$$

The function χ_S is called the ***characteristic function* of S**.

a. If $A = \{a, b, c\}$ and $S = \{a, b\}$, list the elements of χ_S .
b. If $A = \{a, b, c, d, e\}$ and $S = \{a, c, e\}$, list the elements of χ_S.
c. If $A = \{a, b, c\}$, what are $\chi_\emptyset$ and χ_A?

Exercise 7.1.5

If A and B are finite sets, how many different functions are there from A into B?

Answer

For each of the $|A|$ elements of A, there are $|B|$ possible images, so there are $|B| \cdot |B| \cdot \ldots \cdot |B| = |B|^{|A|}$ functions from A into B.

Exercise 7.1.6

Let U be a set with subsets A and B.

a. Show that $g : U \to \{0, 1\}$ defined by $g(a) = \min(C_A(a), C_B(a))$ is the characteristic function of $A \cap B$.
b. What characteristic function is $h : U \to \{0, 1\}$ defined by $h(a) = \max(C_A(a), C_B(a))$?
c. How are the characteristic functions of A and A^c related?

Exercise 7.1.7

Let f be a function with domain A and codomain B. Consider the relation $K \subseteq A \times A$ defined on the domain of f by $(x, y) \in K$ if and only if $f(x) = f(y)$. The relation K is called the ***kernel*** of f.

a. Prove that K is an equivalence relation.
b. For the specific case of $A = \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers, let $f : \mathbb{Z} \to \mathbb{Z}$ be defined by $f(x) = x^2$. Describe the equivalence classes of the kernel for this specific function.

Exercise 7.1.8

Let $f : \mathbb{P} \to \mathbb{P}$, where $f(a)$ is the largest power of two that evenly divides a; for example, $f(12) = 4, f(9) = 1,$ and $f(8) = 8$. Describe the equivalence classes of the kernel of f.

This page titled 7.1: Definition and Notation is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

7.2: Properties of Functions

7.2.1: Properties

Consider the following functions:

Let $A = \{1, 2, 3, 4\}$ and $B = \{a, b, c, d\}$, and define $f : A \to B$ by

$$f(1) = a, f(2) = b, f(3) = c \text{ and } f(4) = d$$

Let $A = \{1, 2, 3, 4\}$ and $B = \{a, b, c, d\}$, and define $g : A \to B$ by

$$g(1) = a, g(2) = b, g(3) = a \text{ and } g(4) = b.$$

The first function, f, gives us more information about the set B than the second function, g. Since A clearly has four elements, f tells us that B contains at least four elements since each element of A is mapped onto a different element of B. The properties that f has, and g does not have, are the most basic properties that we look for in a function. The following definitions summarize the basic vocabulary for function properties.

Definition 7.2.1: Injective Function, Injection

A function $f : A \to B$ is injective if

$$\forall a, b \in A, a \neq b \Rightarrow f(a) \neq f(b)$$

An injective function is called an injection, or a one-to-one function.

Notice that the condition for an injective function is logically equivalent to

$$f(a) = f(b) \Rightarrow a = b.$$

for all $a, b \in A$. This is often a more convenient condition to prove than what is given in the definition.

Definition 7.2.2: Surjective Function, Surjection

A function $f : A \to B$ is surjective if its range, $f(A)$, is equal to its codomain, B. A surjective function is called a surjection, or an onto function.

Notice that the condition for a surjective function is equivalent to

$$\text{For all } b \in B, \text{ there exists } a \in A \text{ such that } f(a) = b.$$

Definition 7.2.3: Bijective Function, Bijection

A function $f : A \to B$ is bijective if it is both injective and surjective. Bijective functions are also called one-to-one, onto functions.

The function f that we opened this section with is bijective. The function g is neither injective nor surjective.

Example 7.2.1: Injective but Not Surjective Function

Let $A = \{1, 2, 3\}$ and $B = \{a, b, c, d\}$, and define $f : A \to B$ by $f(1) = b$, $f(2) = c$, and $f(3) = a$. Then f is injective but not surjective.

Example 7.2.2: Characteristic Functions

The characteristic function, χ_S, in Exercise 7.1.4 is surjective if S is a proper subset of A, but never injective if $|A| > 2$.

7.2.2: Counting

Example 7.2.3: Seating Students

Let A be the set of students who are sitting in a classroom, let B be the set of seats in the classroom, and let s be the function which maps each student into the chair he or she is sitting in. When is s one to one? When is it onto? Under normal circumstances, s would always be injective since no two different students would be in the same seat. In order for s to be surjective, we need all seats to be used, so s is a surjection if the classroom is filled to capacity.

Functions can also be used for counting the elements in large finite sets or in infinite sets. Let's say we wished to count the occupants in an auditorium containing 1,500 seats. If each seat is occupied, the answer is obvious, 1,500 people. What we have done is to set up a one-to-one correspondence, or bijection, from seats to people. We formalize in a definition.

Definition 7.2.4: Cardinality

Two sets are said to have the same cardinality if there exists a bijection between them. If a set has the same cardinality as the set $\{1, 2, 3, \ldots, n\}$,then we say its cardinality is n.

The function f that opened this section serves to show that the two sets $A = \{1, 2, 3, 4\}$ and $B = \{a, b, c, d\}$ have the same cardinality. Notice in applying the definition of cardinality, we don't actually appear to count either set, we just match up the elements. However, matching the letters in B with the numbers 1, 2, 3, and 4 is precisely how we count the letters.

Definition 7.2.5: Countable Set

If a set is finite or has the same cardinality as the set of positive integers, it is called a countable set.

Example 7.2.4: Counting the Alphabet

The alphabet $\{A, B, C, \ldots, Z\}$ has cardinality 26 through the following bijection into the set $\{1, 2, 3, \ldots, 26\}$.

$$\begin{array}{ccccc} A & B & C & \cdots & Z \\ \downarrow & \downarrow & \downarrow & \cdots & \downarrow \\ 1 & 2 & 3 & \cdots & 26 \end{array}.$$

Example 7.2.5: As Many Evens as All Positive Integers

Recall that $2\mathbb{P} = \{b \in \mathbb{P} \mid b = 2k \text{ for some } k \in \mathbb{P}\}$. Paradoxically, $2\mathbb{P}$ has the same cardinality as the set $\mathbb{P}$ of positive integers. To prove this, we must find a bijection from $\mathbb{P}$ to $2\mathbb{P}$. Such a function isn't unique, but this one is the simplest: $f : \mathbb{P} \to 2\mathbb{P}$ where $f(m) = 2m$. Two statements must be proven to justify our claim that f is a bijection:

- f is one-to-one.
  Proof: Let $a, b \in \mathbb{P}$ and assume that $f(a) = f(b)$. We must prove that $a = b$.
  $$f(a) = f(b) \Longrightarrow 2a = 2b \Longrightarrow a = b.$$
- f is onto.
  Proof: Let $b \in 2\mathbb{P}$. We want to show that there exists an element $a \in \mathbb{P}$ such that $f(a) = b$. If $b \in 2\mathbb{P}$, $b = 2k$ for some $k \in \mathbb{P}$ by the definition of $2\mathbb{P}$. So we have $f(k) = 2k = b$. Hence, each element of $2\mathbb{P}$ is the image of some element of $\mathbb{P}$.

Another way to look at any function with $\mathbb{P}$ as its domain is creating a list of the form $f(1), f(2), f(3), \ldots$. In the previous example, the list is $2, 4, 6, \ldots$. This infinite list clearly has no duplicate entries and every even positive integer appears in the list eventually.

A function $f : \mathbb{P} \to A$ is a bijection if the infinite list $f(1), f(2), f(3), \ldots$ contains no duplicates, and every element of A appears on in the list. In this case, we say the A is ***countably infinite***, or simply ***countable***.

Readers who have studied real analysis should recall that the set of rational numbers is a countable set, while the set of real numbers is not a countable set. See the exercises at the end of this section for an another example of such a set.

We close this section with a theorem called the Pigeonhole Principle, which has numerous applications even though it is an obvious, common-sense statement. Never underestimate the importance of simple ideas. The Pigeonhole Principle states that if there are more pigeons than pigeonholes, then two or more pigeons must share the same pigeonhole. A more rigorous mathematical statement of the principle follows.

Theorem 7.2.1: The Pigeonhole Principle

Let f be a function from a finite set X into a finite set Y. If $n \geq 1$ and $|X| > n|Y|$, then there exists an element of Y that is the image under f of at least $n+1$ elements of X.

Proof

Assume no such element exists. For each $y \in Y$, let $A_y = \{x \in X \mid f(x) = y\}$. Then it must be that $|A_y| \leq n$. Furthermore, the set of nonempty A_y form a partition of X. Therefore,

$$|X| = \sum_{y \in Y} |A_y| \leq n|Y|$$

which is a contradiction.

Example 7.2.6: A Duplicate Name is Assured

Assume that a room contains four students with the first names John, James, and Mary. Prove that two students have the same first name. We can visualize a mapping from the set of students to the set of first names; each student has a first name. The pigeonhole principle applies with $n = 1$, and we can conclude that at least two of the students have the same first name.

7.2.3: Exercises

Exercise 7.2.1

Determine which of the functions in Exercise 7.1.1 of Section 7.1 are one- to-one and which are onto.

Answer

The only one-to-one function and the only onto function is f.

Exercise 7.2.2

a. Determine all bijections from $\{1, 2, 3\}$ into $\{a, b, c\}$.
b. Determine all bijections from $\{1, 2, 3\}$ into $\{a, b, c, d\}$.

Exercise 7.2.3

Which of the following are one-to-one, onto, or both?

a. $f_1 : \mathbb{R} \to \mathbb{R}$ defined by $f_1(x) = x^3 - x$.
b. $f_2 : \mathbb{Z} \to \mathbb{Z}$ defined by $f_2(x) = -x + 2$.
c. $f_3 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined by $f_3(j, k) = 2^j 3^k$.
d. $f_4 : \mathbb{P} \to \mathbb{P}$ defined by $f_4(n) = \lceil n/2 \rceil$, where $\lceil x \rceil$ is the ceiling of x, the smallest integer greater than or equal to x.
e. $f_5 : \mathbb{N} \to \mathbb{N}$ defined by $f_5(n) = n^2 + n$.
f. $f_6 : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ defined by $f_6(n) = (2n, 2n + 1)$.

Answer

a. f_1 is onto but not one-to-one: $f_1(0) = f_1(1)$.
b. f_2 is one-to-one and onto.

c. f_3 is one-to-one but not onto.
d. f_4 is onto but not one-to-one.
e. f_5 is one-to-one but not onto.
f. f_6 is one-to-one but not onto.

Exercise 7.2.4

Which of the following are injections, surjections, or bijections on $\mathbb{R}$, the set of real numbers?

a. $f(x) = -2x$.
b. $g(x) = x^2 - 1$.
c. $h(x) = \begin{cases} x & x < 0 \\ x^2 & x \geq 0 \end{cases}$
d. $q(x) = 2^x$
e. $r(x) = x^3$
f. $s(x) = x^3 - x$

Exercise 7.2.5

Suppose that m pairs of socks are mixed up in your sock drawer. Use the Pigeonhole Principle to explain why, if you pick $m+1$ socks at random, at least two will make up a matching pair.

Answer

Let $X = \{\text{socks selected}\}$ and $Y = \{\text{pairs of socks}\}$ and define $f : X \to Y$ where $f(x) =$the pair of socks that x belongs to . By the Pigeonhole principle, there exist two socks that were selected from the same pair.

Exercise 7.2.6

In your own words explain the statement “The sets of integers and even integers have the same cardinality.”

Exercise 7.2.7

Let $A = \{1, 2, 3, 4, 5\}$. Find functions, if they exist that have the properties specified below.

a. A function that is one-to-one and onto.
b. A function that is neither one-to-one nor onto.
c. A function that is one-to-one but not onto.
d. A function that is onto but not one-to-one.

Answer

a. $f(n) = n,$ for example
b. $f(n) = 1,$ for example
c. None exist.
d. None exist.

Exercise 7.2.8

a. Define functions, if they exist, on the positive integers, $\mathbb{P}$, with the same properties as in Exercise $7.2.7$ (if possible).
b. Let A and B be finite sets where $|A| = |B|$. Is it possible to define a function $f : A \to B$ that is one-to-one but not onto? Is it possible to find a function $g : A \to B$ that is onto but not one-to-one?

Exercise 7.2.9

a. Prove that the set of natural numbers is countable.
b. Prove that the set of integers is countable.
c. Prove that the set of rational numbers is countable.

Answer

1. Use $s : \mathbb{N} \to \mathbb{P}$ defined by $s(x) = x + 1$.
2. Use the function $f : \mathbb{N} \to \mathbb{Z}$ defined by $f(\text{x0} = x/2$ if x is even and $f(x) = -(x+1)/2$ if x is odd.
3. The proof is due to Georg Cantor (1845-1918), and involves listing the rationals through a definite procedure so that none are omitted and duplications are avoided. In the first row list all nonnegative rationals with denominator 1, in the second all nonnegative rationals with denominator 2, etc. In this listing, of course, there are duplications, for example, $0/1 = 0/2 = 0, 1/1 = 3/3 = 1, 6/4 = 9/6 = 3/2$, etc. To obtain a list without duplications follow the arrows in Figure 7.2.1, listing only the circled numbers.
   We obtain: $0, 1, 1/2, 2, 3, 1/3, 1/4, 2/3, 3/2, 4/1$, Each nonnegative rational appears in this list exactly once. We now must insert in this list the negative rationals, and follow the same scheme to obtain:

$$0, 1, -1, 1/2, -1/2, 2, -2, 3, -3, 1/3, -1/3, \ldots$$

   which can be paired off with the elements of $\mathbb{N}$.

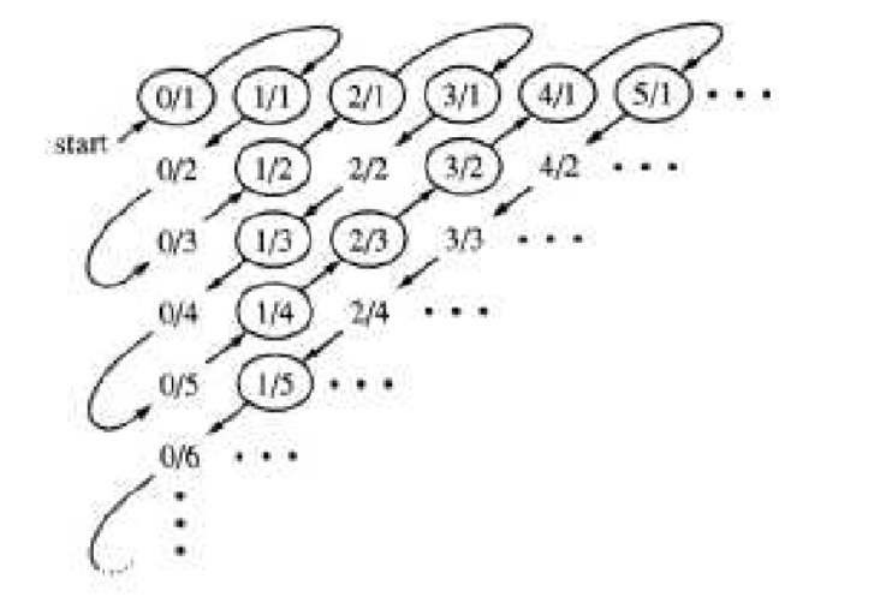


Figure 7.2.1: Enumeration of the rational numbers.

Exercise 7.2.10

a. Prove that the set of finite strings of 0's and 1's is countable.
b. Prove that the set of odd integers is countable.
c. Prove that the set $\mathbb{N} \times \mathbb{N}$ is countable.

Exercise 7.2.11

Use the Pigeonhole Principle to prove that an injection cannot exist between a finite set A and a finite set B if the cardinality of A is greater than the cardinality of B.

Answer

Let f be any function from A into B. By the Pigeonhole principle with $n = 1$, there exists an element of B that is the image of at least two elements of A. Therefore, f is not an injection.

Exercise 7.2.12

The important properties of relations are not generally of interest for functions. Most functions are not reflexive, symmetric, antisymmetric, or transitive. Can you give examples of functions that do have these properties?

Exercise 7.2.13

Prove that the set of all infinite sequences of 0's and 1's is not a countable set.

Answer

The proof is indirect and follows a technique called the Cantor diagonal process. Assume to the contrary that the set is countable, then the elements can be listed: $n_1, n_2, n_3, \ldots$ where each n_i is an infinite sequence of 0s and 1s. Consider the array:

$$\begin{aligned} n_1 &= n_{11}n_{12}n_{13}\cdots \\ n_2 &= n_{21}n_{22}n_{23}\cdots \\ n_3 &= n_{31}n_{32}n_{33}\cdots \\ &\vdots \end{aligned}$$

We assume that this array contains all infinite sequences of 0s and 1s. Consider the sequence s defined by $s_i = \begin{cases} 0 & \text{if } n_{\text{ii}} = 1 \\ 1 & \text{if } n_{\text{ii}} = 0 \end{cases}$

Notice that s differs from each n_i in the ith position and so cannot be in the list. This is a contradiction, which completes our proof.

Exercise 7.2.14

Prove that the set of all functions on the integers is an uncountable set.

Exercise 7.2.15

Given five points on the unit square, $\{(x, y) \mid 0 \leq x, y \leq 1\}$, prove that there are two of the points a distance of no more than $\frac{\sqrt{2}}{2}$ from one another.

This page titled 7.2: Properties of Functions is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

7.3: Function Composition

Now that we have a good understanding of what a function is, our next step is to consider an important operation on functions. Our purpose is not to develop the algebra of functions as completely as we did for the algebras of logic, matrices, and sets, but the reader should be aware of the similarities between the algebra of functions and that of matrices. We first define equality of functions.

7.3.1: Function Equality

Definition 7.3.1: Equality of Functions

Let $f, g : A \to B$; that is, let f and g both be functions from A into B. Then f is equal to g (denoted $f = g$) if and only if $f(x) = g(x)$ for all $x \in A$.

Two functions that have different domains cannot be equal. For example, $f : \mathbb{Z} \to \mathbb{Z}$ defined by $f(x) = x^2$ and $g : \mathbb{R} \to \mathbb{R}$ defined by $g(x) = x^2$ are not equal even though the formula that defines them is the same.

On the other hand, it is not uncommon for two functions to be equal even though they are defined differently. For example consider the functions h and k, where $h : \{-1, 0, 1, 2\} \to \{0, 1, 2\}$ is defined by $h(x) = |x|$ and $k : \{-1, 0, 1, 2\} \to \{0, 1, 2\}$ is defined by $k(x) = -\frac{x^3}{3} + x^2 + \frac{x}{3}$ appear to be very different functions. However, they are equal because $h(x) = k(x)$ for $x = -1, 0, 1,$ and 2.

7.3.2: Function Composition

One of the most important operations on functions is that of composition.

Definition 7.3.2: Composition of Functions

Let $f : A \to B$ and $g : B \to C$. Then the composition of f followed by g, written $g \circ f$, is a function from A into C defined by $(g \circ f)(x) = g(f(x))$, which is read "g of f of x."

The reader should note that it is traditional to write the composition of functions from right to left. Thus, in the above definition, the first function performed in computing $g \circ f$ is f. On the other hand, for relations, the composition rs is read from left to right, so that the first relation is r.

Example 7.3.1: A Basic Example

Let $f : \{1, 2, 3\} \to \{a, b\}$ be defined by $f(1) = a$, $f(2) = a$, and $f(3) = b$. Let $g : \{a, b\} \to \{5, 6, 7\}$ be defined by $g(a) = 5$ and $g(b) = 7$. Then $g \circ f : \{1, 2, 3\} \to \{5, 6, 7\}$ is defined by $(g \circ f)(1) = 5$, $(g \circ f)(2) = 5$, and $(g \circ f)(3) = 7$. For example, $(g \circ f)(1) = g(f(l)) = g(a) = 5$. Note that $f \circ g$ is not defined. Why?

Let $f : \mathbb{R} \to \mathbb{R}$ be defined by $f(x) = x^3$ and let $g : \mathbb{R} \to \mathbb{R}$ be defined by $g(x) = 3x + 1$. Then, since

$$(g \circ f)(x) = g(f(x)) = g\left(x^3\right) = 3x^3 + 1$$

we have $g \circ f : \mathbb{R} \to \mathbb{R}$ is defined by $(g \circ f)(x) = 3x^3 + 1$. Here $f \circ g$ is also defined and $f \circ g : \mathbb{R} \to \mathbb{R}$ is defined by $(f \circ g)(x) = (3x + 1)^3$. Moreover, since $3x^3 + 1 \neq (3x + 1)^3$ for at least one real number, $g \circ f \neq f \circ g$. Therefore, the commutative law is not true for functions under the operation of composition. However, the associative law is true for functions under the operation of composition.

Theorem 7.3.1: Function Composition is Associative

If $f : A \to B$, $g : B \to C$, and $h : C \to D$, then $h \circ (g \circ f) = (h \circ g) \circ f$.

Proof

Note: In order to prove that two functions are equal, we must use the definition of equality of functions. Assuming that the functions have the same domain, they are equal if, for each domain element, the images of that element under the two

functions are equal.

We wish to prove that $(h \circ (g \circ f))(x) = ((h \circ g) \circ f)(x)$ for all $x \in A$, which is the domain of both functions.

$$\begin{aligned}(h \circ (g \circ f))(x) &= h((g \circ f)(x)) \text{ by the definition of composition} \\ &= h(g(f(x))) \text{ by the definition of composition}\end{aligned}$$

Similarly,

$$\begin{aligned}((h \circ g) \circ f)(x) &= (h \circ g)(f(x)) \text{ by the definition of composition} \\ &= h(g(f(x))) \text{ by the definition of composition}\end{aligned}.$$

Notice that no matter how the functions the expression $h \circ g \circ f$ is grouped, the final image of any element of $x \in A$ is $h(g(f(x)))$ and so $h \circ (g \circ f) = (h \circ g) \circ f$.

If f is a function on a set A, then the compositions $f \circ f$, $f \circ f \circ f, \ldots$ are valid, and we denote them as f^2 , $f^3, \ldots$. Repeated compositions of f with itself can be defined recursively. We will discuss this form of definition in detail in Section 8.1.

Definition 7.3.3: Powers of Functions

Let $f : A \to A$.

- $f^1 = f$; that is, $f^1(a) = f(a)$, for $a \in A$.
- For $n \geq 1$, $f^{n+1} = f \circ f^n$; that is, $f^{n+1}(a) = f\left(f^n(a)\right)$ for $a \in A$.

Two useful theorems concerning composition are given below. The proofs are left for the exercises.

Theorem 7.3.2: The Composition of Injections is an Injection

If $f : A \to B$ and $g : B \to C$ are injections, then $g \circ f : A \to C$ is an injection.

Theorem 7.3.3: The Composition of Surjections is a Surjection

If $f : A \to B$ and $g : B \to C$ are surjections, then $g \circ f : A \to C$ is a surjection.

We would now like to define the concepts of identity and inverse for functions under composition. The motivation and descriptions of the definitions of these terms come from the definitions of the terms in the set of real numbers and for matrices. For real numbers, the numbers 0 and 1 play the unique role that $x + 0 = 0 + x = x$ and $x \cdot 1 = 1 \cdot x = x$ for any real number x. 0 and 1 are the identity elements for the reals under the operations of addition and multiplication, respectively. Similarly, the $n \times n$ zero matrix 0 and the $n \times n$ identity matrix I are such that for any $n \times n$ matrix A, $A + 0 = 0 + A = A$ and $AI = IA = A$. Hence, an elegant way of defining the identity function under the operation of composition would be to imitate the above well-known facts.

Definition 7.3.4: Identity Function

For any set A, the identity function on A is a function from A onto A, denoted by i (or, more specifically, i_A) such that $i(a) = a$ for all $a \in A$.

Based on the definition of i, we can show that for all functions $f : A \to A$, $f \circ i = i \circ f = f$.

Example 7.3.2: The Identity Function on $\{1, 2, 3\}$

If $A = \{1, 2, 3\}$, then the identity function $i : A \to A$ is defined by $i(1) = 1$, $i(2) = 2$, and $i(3) = 3$.

Example 7.3.3: The Identity Function on $\mathbb{R}$

The identity function on $\mathbb{R}$ is $i : \mathbb{R} \to \mathbb{R}$ defined by $i(x) = x$.

7.3.3: Inverse Functions

We will introduce the inverse of a function with a special case: the inverse of a function on a set. After you've taken the time to understand this concept, you can read about the inverse of a function from one set into another. The reader is encouraged to reread the definition of the inverse of a matrix in Section 5.2 (Definition 5.2.3) to see that the following definition of the inverse function is a direct analogue of that definition.

Definition 7.3.5: Inverse of a Function on a Set

Let $f : A \to A$. If there exists a function $g : A \to A$ such that $g \circ f = f \circ g = i$, then g is called the inverse of f and is denoted by f^{-1}, read "f inverse."

Notice that in the definition we refer to "the inverse" as opposed to "an inverse." It can be proven that a function can never have more than one inverse (see exercises).

An alternate description of the inverse of a function, which can be proven from the definition, is as follows: Let $f : A \to A$ be such that $f(a) = b$. Then when it exists, f^{-1} is a function from A to A such that $f^{-1}(b) = a$. Note that f^{-1} "undoes" what f does.

Example 7.3.4: The Inverse of a Function on $\{1, 2, 3\}$

Let $A = \{1, 2, 3\}$ and let f be the function defined on A such that $f(1) = 2$, $f(2) = 3$, and $f(3) = 1$. Then $f^{-1} : A \to A$ is defined by $f^{-1}(1) = 3$, $f^{-1}(2) = 1$, and $f^{-1}(3) = 2$.

Example 7.3.5: Inverse of a Real Function

If $g : \mathbb{R} \to \mathbb{R}$ is defined by $g(x) = x^3$, then g^{-1} is the function that undoes what g does. Since g cubes real numbers, g^{-1} must be the "reverse" process, namely, takes cube roots. Therefore, $g^{-1} : \mathbb{R} \to \mathbb{R}$ is defined by $g^{-1}(x) = \sqrt[3]{x}$. We should show that $g^{-1} \circ g = i$ and $g \circ g^{-1} = i$. We will do the first, and the reader is encouraged to do the second.

$$\begin{aligned}
\left(g^{-1} \circ g\right)(x) &= g^{-1}(g(x)) \quad \text{Definition of composition} \\
&= g^{-1}\left(x^3\right) \quad \text{Definition of } g \\
&= \sqrt[3]{x^3} \quad \text{Definition of } g^{-1} \\
&= x \quad \text{Definition of cube root} \\
&= i(x) \quad \text{Definition of the identity function}
\end{aligned}$$

Therefore, $g^{-1} \circ g = i$. Why?

The definition of the inverse of a function alludes to the fact that not all functions have inverses. How do we determine when the inverse of a function exists?

Theorem 7.3.4: Bijections Have Inverses

Let $f : A \to A$. f^{-1} exists if and only if f is a bijection; i. e. f is one-to-one and onto.

Proof

($\Rightarrow$) In this half of the proof, assume that f^{-1} exists and we must prove that f is one-to-one and onto. To do so, it is convenient for us to use the relation notation, where $f(s) = t$ is equivalent to $(s, t) \in f$. To prove that f is one-to-one, assume that $f(a) = f(b) = c$. Alternatively, that means (a, c) and (b, c) are elements of f. We must show that $a = b$. Since $(a, b), (c, b) \in f$, (c, a) and (c, b) are in f^{-1}. By the fact that f^{-1} is a function and c cannot have two images, a and b must be equal, so f is one-to-one.

Next, to prove that f is onto, observe that for f^{-1} to be a function, it must use all of its domain, namely A. Let b be any element of A. Then b has an image under f^{-1} , $f^{-1}(b)$. Another way of writing this is $\left(b, f^{-1}(b)\right) \in f^{-1}$, By the definition of the inverse, this is equivalent to $\left(f^{-1}(b), b\right) \in f$. Hence, b is in the range of f. Since b was chosen arbitrarily, this shows that the range of f must be all of A.

($\Leftarrow$) Assume f is one-to-one and onto and we are to prove f^{-1} exists. We leave this half of the proof to the reader. □

Definition 7.3.6: Permutation

A bijection of a set A into itself is called a permutation of A.

Next, we will consider the functions for which the domain and codomain are not necessarily equal. How do we define the inverse in this case?

Definition 7.3.7: Inverse of a Function (General Case)

Let $f: A \rightarrow B,$ If there exists a function $g: B \rightarrow A$ such that $g \circ f=i_A$ and $f \circ g=i_B$, then g is called the inverse of f and is denoted by f^{-1} , read "f inverse."

Note the slightly more complicated condition for the inverse in this case because the domains of $f \circ g$ and $g \circ f$ are different if A and B are different. The proof of the following theorem isn't really very different from the special case where $A=B$.

Theorem 7.3.5: When Does a Function Have an Inverse?

Let $f: A \rightarrow B$. f^{-1} exists if and only if f is a bijection.

Example 7.3.6: Another Inverse

Let $A=\{1,2,3\}$ and $B=\{a, b, c\}$. Define $f: A \rightarrow B$ by $f(1)=a$, $f(2)=b$, and $f(3)=c$. Then $g: B \rightarrow A$ defined by $g(a)=1,\ g(b)=2,$ and $g(c)=3$ is the inverse of f.

$$\left.\begin{array}{l}(g \circ f)(1)=1 \\ (g \circ f)(2)=2 \\ (g \circ f)(3)=3\end{array}\right\} \Rightarrow\ g \circ f=i_A \text{ and } \left.\begin{array}{l}(f \circ g)(a)=a \\ (f \circ g)(b)=b \\ (f \circ g)(c)=c\end{array}\right\} \Rightarrow\ f \circ g=i_B$$

7.3.4: Exercises

Exercise 7.3.1

Let $A=\{1,2,3,4,5\}$, $B=\{a, b, c, d, e, f\},$ and $C=\{+,-\}$. Define $f: A \rightarrow B$ by $f(k)$ equal to the kth letter in the alphabet, and define $g: B \rightarrow C$ by $g(\alpha)=+$ if α is a vowel and $g(\alpha)=-$ if α is a consonant.

a. Find $g \circ f$.
b. Does it make sense to discuss $f \circ g$? If not, why not?
c. Does f^{-1} exist? Why?
d. Does g^{-1} exist? Why?

Answer

a. $g \circ f: A \rightarrow C$ is defined by $(g \circ f)(k)=\begin{cases}+ & \text{if } k=1 \text{ or } k=5 \\ - & \text{otherwise}\end{cases}$
b. No, since the domain of f is not equal to the codomain of g.
c. No, since f is not surjective.
d. No, since g is not injective.

Exercise 7.3.2

Let $A = \{1, 2, 3\}$. Define $f : A \to A$ by $f(1) = 2$, $f(2) = 1$, and $f(3) = 3$. Find f^2, f^3, f^4 and f^{-1}.

Exercise 7.3.3

Let $A = \{1, 2, 3\}$.

a. List all permutations of A.
b. Find the inverse and square of each of the permutations of part a, where the square of a permutation, f, is the composition $f \circ f$.
c. Show that the composition of any two permutations of A is a permutation of A.
d. Prove that if A is any set where $|A| = n$, then the number of permutations of A is $n!$.

Answer

1. The permutations of A are i, r_1, r_2, f_1, f_2, and f_3, defined in Table 15.3.1.
2.

g	g^{-1}	g^2
i	i	i
r_1	r_2	r_2
r_2	r_1	r_1
f_1	f_1	i
f_2	f_2	i
f_3	f_3	i

3. If f and g are permutations of A, then they are both injections and their composition, $f \circ g$, is a injection, by Theorem 7.3.2 By Theorem 7.3.3 $f \circ g$ is also a surjection; therefore, $f \circ g$ is a bijection on A, a permutation.
4. Proof by induction: Basis: $(n = 1)$. The number of permutations of A is one, the identity function, and $1! = 1$.

   Induction: Assume that the number of permutations on a set with n elements, $n \geq 1$, is $n!$. Furthermore, assume that $|A| = n + 1$ and that A contains an element called σ. Let $A' = A - \{\sigma\}$. We can reduce the definition of a permutation, f, on A to two steps. First, we select any one of the $n!$ permutations on A'. (Note the use of the induction hypothesis.) Call it g. This permutation almost completely defines a permutation on A that we will call f. For all a in A', we start by defining $f(a)$ to be $g(a)$. We may be making some adjustments, but define it that way for now. Next, we select the image of σ, which can be done $n + 1$ different ways, allowing for any value in A. To keep our function bijective, we must adjust f as follows: If we select $f(\sigma) = y \neq \sigma$, then we must find the element, z, of A such that $g(z) = y$, and redefine the image of z to $f(z) = \sigma$. If we had selected $f(\sigma) = \sigma$, then there is no adjustment needed. By the rule of products, the number of ways that we can define f is $n!(n + 1) = (n + 1)!$ □

Exercise 7.3.4

Define s, u, and d, all functions on the integers, by $s(n) = n^2$, $u(n) = n + 1$, and $d(n) = n - 1$. Determine:

a. $u \circ s \circ d$
b. $s \circ u \circ d$
c. $d \circ s \circ u$

Exercise 7.3.5

Consider the following functions on the set of bit strings of length 4. In these definitions, addition is done modulo 2, so that $1 + 1 = 0$. Which of these functions has an inverse? For those that have an inverse, what is it? For those that don't explain why.

a. $f_1(b_1b_2b_3b_4) = b_2b_3b_4b_1$
b. $f_2(b_1b_2b_3b_4) = b_4b_3b_2b_1$

c. $f_3(b_1b_2b_3b_4) = (b_1 + b_2)(b_2 + b_3)(b_3 + b_4)(b_4 + b_1)$
d. $f_4(b_1b_2b_3b_4) = b_1(b_1 + b_2)(b_1 + b_2 + b_3)(b_1 + b_2 + b_3 + b_4)$

Answer

a. f_1 has an inverse. $f_1^{-1} = f_1^3$.
b. f_2 has an inverse. $f_2^{-1} = f_2$.
c. f_3 does not have an inverse. One way to verify this is to note that f_3 is not one-to-one because $f_3(0000) = 0000 = f_3(1111)$.
d. f_4 has an inverse. $f_4^{-1} = f_4^3$.

Exercise 7.3.6: Inverse Images

If f is any function from A into B, we can describe the inverse image as a function from B into $\mathcal{P}(A)$, which is also commonly denoted f^{-1}. If $b \in B$, $f^{-1}(b) = \{a \in A \mid f(a) = b\}$. If f does have an inverse, the inverse image of b is $\left\{f^{-1}(b)\right\}$.

a. Let $g : \mathbb{R} \to \mathbb{R}$ be defined by $g(x) = x^2$. What are $g^{-1}(4)$, $g^{-1}(0)$ and $g^{-1}(-1)$?
b. If $r : \mathbb{R} \to \mathbb{Z}$, where $r(x) = \lceil x \rceil$, what is $r^{-1}(1)$?

Exercise 7.3.7

Let f, g, and h all be functions from $\mathbb{Z}$ into $\mathbb{Z}$ defined by $f(n) = n + 5$, $g(n) = n - 2$, and $h(n) = n^2$. Define:

a. $f \circ g$
b. f^3
c. $f \circ h$

Answer

a. $f \circ g(n) = n + 3$
b. $f^3(n) = n + 15$
c. $f \circ h(n) = n^2 + 5$

Exercise 7.3.8

Define the following functions on the integers by $f(k) = k + 1$, $g(k) = 2k$, and $h(k) = \lceil k/2 \rceil$

a. Which of these functions are one-to-one?
b. Which of these functions are onto?
c. Express in simplest terms the compositions $f \circ g$, $g \circ f$, $g \circ h$, $h \circ g$, and h^2.

Exercise 7.3.9

Let A be a nonempty set. Prove that if f is a bijection on A and $f \circ f = f$, then f is the identity function, i

Hint

You have seen a similar proof in matrix algebra.

Exercise 7.3.10

For the real matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $\det(A) = ad - bc$.

Recall that a ***bijection*** from a set to itself is also referred to as a ***permutation*** of the set. Let π be a permutation of $\{a, b, c, d\}$ such that a becomes $\pi(a)$, b becomes $\pi(b)$, etc.

Let $B = \begin{pmatrix} \pi(a) & \pi(b) \\ \pi(c) & \pi(d) \end{pmatrix}$. How many permutations of π leave the determinant of A invariant, that is, $\det A = \det B$?

Exercise 7.3.11

State and prove a theorem on inverse functions analogous to the one that says that if a matrix has an inverse, that inverse is unique.

Answer

If $f : A \to B$ and f has an inverse, then that inverse is unique.

Proof: Suppose that g and h are both inverses of f, both having domain B and codomain A.

$$\begin{aligned} g &= g \circ i_B \\ &= g \circ (f \circ h) \\ &= (g \circ f) \circ h \\ &= i_A \circ h \\ &= h \quad \Rightarrow g = h \quad \square \end{aligned}$$

Exercise 7.3.12

Let f and g be functions whose inverses exist. Prove that $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

Hint

See Exercise 5.4.3 of Section 5.4.

Exercise 7.3.13

Prove Theorem 7.3.2 and Theorem 7.3.3.

Answer

Let x, x' be elements of A such that $g \circ f(x) = g \circ f(x')$; that is, $g(f(x)) = g(f(x'))$. Since g is injective, $f(x) = f(x')$ and since f is injective, $x = x'$. $\square$

Let x be an element of C. We must show that there exists an element of A whose image under $g \circ f$ is x. Since g is surjective, there exists an element of B, y, such that $g(y) = x$. Also, since f is a surjection, there exists an element of A, z, such that $f(z) = y$, $g \circ f(z) = g(f(z)) = g(y) = x$. $\square$

Exercise 7.3.14

Prove the second half of Theorem 7.3.4.

Exercise 7.3.15

Prove by induction that if $n \geq 2$ and $f_1, f_2, \ldots, f_n$ are invertible functions on some nonempty set A, then $(f_1 \circ f_2 \circ \cdots \circ f_n)^{-1} = f_n^{-1} \circ \cdots \circ f_2^{-1} \circ f_1^{-1}$. The basis has been taken care of in Exercise 7.3.12

Answer

Basis: $(n = 2)$: $(f_1 \circ f_2)^{-1} = {f_2}^{-1} \circ {f_1}^{-2}$ by Exercise 7.3.12

Induction: Assume $n \geq 2$ and

$$(f_1 \circ f_2 \circ \cdots \circ f_n)^{-1} = {f_n}^{-1} \circ \cdots \circ {f_2}^{-1} \circ {f_1}^{-1}$$

and consider $(f_1 \circ f_2 \circ \cdots \circ f_{n+1})^{-1}$.

$$
\begin{aligned}
(f_1 \circ f_2 \circ \cdots \circ f_{n+1})^{-1} &= ((f_1 \circ f_2 \circ \cdots \circ f_n) \circ f_{n+1})^{-1} \\
&= {f_{n+1}}^{-1} \circ (f_1 \circ f_2 \circ \cdots \circ f_n)^{-1} \\
&\qquad \text{by the basis} \\
&= {f_{n+1}}^{-1} \circ ({f_n}^{-1} \circ \cdots \circ {f_2}^{-1} \circ {f_1}^{-1}) \\
&\qquad \text{by the induction hypothesis} \\
&= {f_{n+1}}^{-1} \circ \cdots \circ {f_2}^{-1} \circ {f_1}^{-1} \quad .\square
\end{aligned}
$$

Exercise 7.3.16

a. Our definition of cardinality states that two sets, A and B, have the same cardinality if there exists a bijection between the two sets. Why does it not matter whether the bijection is from A into B or B into A?
b. Prove that "has the same cardinality as" is an equivalence relation on sets.

Exercise 7.3.17

Construct a table listing as many "Laws of Function Composition" as you can identify. Use previous lists of laws as a guide.

Exercise 7.3.18

Based on the definition of the identity function, show that for all functions $f : A \to A$, $f \circ i = i \circ f = f$.

This page titled 7.3: Function Composition is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

8: Recursion and Recurrence Relations

> ***Fibonacci Sequence***
>
> *Zero, one! One, two, three! Five and eight!*
> *Then thirteen, twenty-one! At this rate*
> ***Fibonacci** appears;*
> *The man's sequence for years*
> *Has kept math students studying late.*
>
> *Goldie, The Omnificent English Dictionary in Limerick Form*

An essential tool that anyone interested in computer science must master is how to think recursively. The ability to understand definitions, concepts, algorithms, etc., that are presented recursively and the ability to put thoughts into a recursive framework are essential in computer science. One of our goals in this chapter is to help the reader become more comfortable with recursion in its commonly encountered forms.

A second goal is to discuss recurrence relations. We will concentrate on methods of solving recurrence relations, including an introduction to generating functions.

8.1: The Many Faces of Recursion
8.2: Sequences
8.3: Recurrence Relations
8.4: Some Common Recurrence Relations
8.5: Generating Functions

This page titled 8: Recursion and Recurrence Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

8.1: The Many Faces of Recursion

Consider the following definitions, all of which should be somewhat familiar to you. When reading them, concentrate on how they are similar.

8.1.1: Binomial Coefficients

Here is a recursive definition of binomial coefficients, which we introduced in Chapter 2.

Definition 8.1.1: Binomial Coefficient - Recursion Definition

Assume $n \geq 0$ and $n \geq k \geq 0$. We define $\binom{n}{k}$ by

- $\binom{n}{0} = 1$
- $\binom{n}{n} = 1$ and
- $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ if $n > k > 0$

Observation 8.1.1

A word about definitions: Strictly speaking, when mathematical objects such as binomial coefficients are defined, they should be defined just once. Since we defined binomial coefficients earlier, in Definition 2.4.1, other statements describing them should be theorems. The theorem, in this case, would be that the "definition" above is consistent with the original definition. Our point in this chapter in discussing recursion is to observe alternative definitions that have a recursive nature. In the exercises, you will have the opportunity to prove that the two definitions are indeed equivalent.

Here is how we can apply the recursive definition to compute $\binom{5}{2}$.

$$\begin{aligned}
\binom{5}{2} &= \binom{4}{2} + \binom{4}{1} \\
&= (\binom{3}{2} + \binom{3}{1}) + (\binom{3}{1} + \binom{3}{0}) \\
&= \binom{3}{2} + 2\binom{3}{1} + 1 \\
&= (\binom{2}{2} + \binom{2}{1}) + 2(\binom{2}{1} + \binom{2}{0}) + 1 \\
&= (1 + \binom{2}{1}) + 2(\binom{2}{1} + 1) + 1 \\
&= 3\binom{2}{1} + 4 \\
&= 3(\binom{1}{1} + \binom{1}{0}) + 4 \\
&= 3(1+1) + 4 = 10
\end{aligned}$$

8.1.2: Polynomials and Their Evaluation

Definition 8.1.2: Polynomial Expression in x over S (Non-Recursive)

Let n be an integer, $n \geq 0$. An nth degree polynomial in x is an expression of the form $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$, where $a_n, a_{n-1}, \ldots, a_1, a_0$ are elements of some designated set of numbers, S, called the set of coefficients and $a_n \neq 0$.

We refer to x as a variable here, although the more precise term for x is an *indeterminate*. There is a distinction between the terms indeterminate and variable, but that distinction will not come into play in our discussions.

Zeroth degree polynomials are called constant polynomials and are simply elements of the set of coefficients.

This definition is often introduced in algebra courses to describe expressions such as $f(n) = 4n^3 + 2n^2 - 8n + 9$, a third-degree, or cubic, polynomial in n. This definition has a drawback when the variable is given a value and the expression must be evaluated. For example, suppose that $n = 7$. Your first impulse is likely to do this:

$$\begin{aligned} f(7) &= 4 \cdot 7^3 + 2 \cdot 7^2 - 8 \cdot 7 + 9 \\ &= 4 \cdot 343 + 2 \cdot 49 - 8 \cdot 7 + 9 \\ &= 1423 \end{aligned}$$

A count of the number of operations performed shows that five multiplications and three additions/subtractions were performed. The first two multiplications compute 7^2 and 7^3, and the last three multiply the powers of 7 times the coefficients. This gives you the four terms; and adding/subtracting a list of k numbers requires $k-1$ addition/subtractions. The following definition of a polynomial expression suggests another more efficient method of evaluation.

Definition 8.1.3: Polynomial Expression in x over S (Recursive)

Let S be a set of coefficients and x a variable.

a. A zeroth degree polynomial expression in x over S is a nonzero element of S.
b. For $n \geq 1$, an nth degree polynomial expression in x over S is an expression of the form $p(x)x + a$ where $p(x)$ is an $(n-1)^{st}$ degree polynomial expression in x and $a \in S$.

We can easily verify that $f(n) = 4n^3 + 2n^2 - 8n + 9$ is a third-degree polynomial expression in n over $\mathbb{Z}$ based on this definition:

$$f(n) = 4n^3 + 2n^2 - 8n + 9 = ((4n + 2)n - 8)n + 9$$

Notice that 4 is a zeroth degree polynomial since it is an integer. Therefore $4n + 2$ is a first-degree polynomial; therefore, $(4n + 2)n - 8$ is a second-degree polynomial in n over $\mathbb{Z}$; therefore, $f(n)$ is a third-degree polynomial in n over $\mathbb{Z}$. The final expression for $f(n)$ is called its ***telescoping form***. If we use it to calculate $f(7)$, we need only three multiplications and three additions/subtractions. This is called ***Horner's method*** for evaluating a polynomial expression.

Example 8.1.1: More Telescoping Polynomials

a. The telescoping form of $p(x) = 5x^4 + 12x^3 - 6x^2 + x + 6$ is $(((5x + 12)x - 6)x + 1)x + 6$. Using Horner's method, computing the value of $p(c)$ requires four multiplications and four additions/subtractions for any real number c.
b. $g(x) = -x^5 + 3x^4 + 2x^2 + x$ has the telescoping form $((((-x + 3)x)x + 2)x + 1)x$.

Many computer languages represent polynomials as lists of coefficients, usually starting with the constant term. For example, $g(x) = -x^5 + 3x^4 + 2x^2 + x$ would be represented with the list $\{0, 1, 2, 0, 3, -1\}$.In both Mathematica and Sage, polynomial expressions can be entered and manipulated, so the list representation is only internal. Some programming languages require users to program polynomial operations with lists. We will leave these programming issues to another source.

8.1.3: Recursive Searching - The Binary Search

Next, we consider a recursive algorithm for a binary search within a sorted list of items. Suppose $r = \{r(1), r(2), \ldots, r(n)\}$ represent a list of n items sorted by a numeric key in descending order. The jth item is denoted $r(j)$ and its key value by $r(j).\texttt{key}$. For example, each item might contain data on the buildings in a city and the key value might be the height of the building. Then $r(1)$ would be the item for the tallest building and $r(1).\texttt{key}$ would be its height. The algorithm $\texttt{BinarySearch}(j, k)$ can be applied to search for an item in r with key value C. This would be accomplished by the execution of $\texttt{BinarySearch}(1, n)$. When the algorithm is completed, the variable `Found` will have a value of `true` if an item with the desired key value was found, and the value of location will be the index of an item whose key is C. If `Found` keeps the value `false`, no such item exists in the list. The general idea behind the algorithm is illustrated in Figure 8.1.1

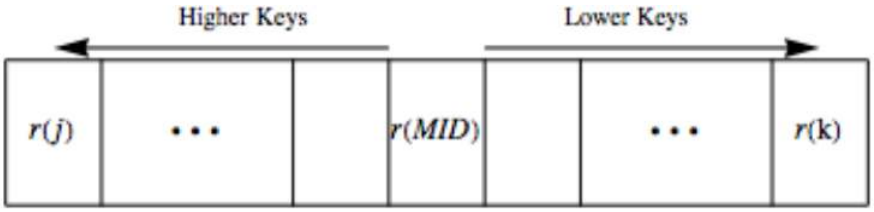


Figure 8.1.1: General Scheme for a Binary Search

In the following implementation of the Binary Search in SageMath, we search within a sorted list of integers. Therefore, the items themselves are the keys.

```
def BinarySearch(r,j,k,C):
    found = False
    if j <= k:
        mid = floor((j+k)/2)
        print('probing at position '+str(mid))
        if r[mid] == C:
            location = mid
            found = True
```

8.1.4: Recursively Defined Sequences

For the next two examples, consider a sequence of numbers to be a list of numbers consisting of a zeroth number, first number, second number, ... If a sequence is given the name S, the kth number of S is usually written S_k or $S(k)$.

Example 8.1.2: Geometric Growth Sequence

Define the sequence of numbers B by

$$B_0 = 100 \text{ and}$$

$$B_k = 1.08B_{k-1} \text{ for } k \geq 1.$$

These rules stipulate that each number in the list is 1.08 times the previous number, with the starting number equal to 100. For example

$$\begin{aligned} B_3 &= 1.08B_2 \\ &= 1.08\,(1.08B_1) \\ &= 1.08\,(1.08\,(1.08B_0)) \\ &= 1.08(1.08(1.08 \cdot 100)) \\ &= 1.08^3 \cdot 100 = 125.971 \end{aligned}$$

Example 8.1.3: The Fibonacci Sequence

The Fibonacci sequence is the sequence F defined by

$$F_0 = 1, F_1 = 1 \text{ and}$$

$$F_k = F_{k-2} + F_{k-1} \text{ for } k \geq 2$$

8.1.5: Recursion

All of the previous examples were presented recursively. That is, every “object” is described in one of two forms. One form is by a simple definition, which is usually called the basis for the recursion. The second form is by a recursive description in which objects are described in terms of themselves, with the following qualification. What is essential for a proper use of recursion is that the objects can be expressed in terms of simpler objects, where “simpler” means closer to the basis of the recursion. To avoid what might be considered a circular definition, the basis must be reached after a finite number of applications of the recursion.

To determine, for example, the fourth item in the Fibonacci sequence we repeatedly apply the recursive rule for F until we are left with an expression involving F_0 and F_1:

$$\begin{aligned} F_4 &= F_2 + F_3 \\ &= (F_0 + F_1) + (F_1 + F_2) \\ &= (F_0 + F_1) + (F_1 + (F_0 + F_1)) \\ &= (1 + 1) + (1 + (1 + 1)) \\ &= 5 \end{aligned}$$

8.1.6: Iteration

On the other hand, we could compute a term in the Fibonacci sequence such as F_5 by starting with the basis terms and working forward as follows:

Table 8.1.1

$F_2 = F_0 + F_1 = 1 + 1 = 2$
$F_3 = F_1 + F_2 = 1 + 2 = 3$
$F_4 = F_2 + F_3 = 2 + 3 = 5$
$F_5 = F_3 + F_4 = 3 + 5 = 8$

This is called an iterative computation of the Fibonacci sequence. Here we start with the basis and work our way forward to a less simple number, such as 5. Try to compute F_5 using the recursive definition for F as we did for F_4. It will take much more time than it would have taken to do the computations above. Iterative computations usually tend to be faster than computations that apply recursion. Therefore, one useful skill is being able to convert a recursive formula into a nonrecursive formula, such as one that requires only iteration or a faster method, if possible.

An iterative formula for $\binom{n}{k}$ is also much more efficient than an application of the recursive definition. The recursive definition is not without its merits, however. First, the recursive equation is often useful in manipulating algebraic expressions involving binomial coefficients. Second, it gives us an insight into the combinatoric interpretation of $\binom{n}{k}$. In choosing k elements from $\{1, 2, \ldots, n\}$,there are $\binom{n-1}{k}$ ways of choosing all k from $\{1, 2, \ldots, n-1\}$, and there are $\binom{n-1}{k-1}$ ways of choosing the k elements if n is to be selected and the remaining $k-1$ elements come from $\{1, 2, \ldots, n-1\}$. Note how we used the Law of Addition from Chapter 2 in our reasoning.

BinarySearch Revisited. In the binary search algorithm, the place where recursion is used is easy to pick out. When an item is examined and the key is not the one you want, the search is cut down to a sublist of no more than half the number of items that you were searching in before. Obviously, this is a simpler search. The basis is hidden in the algorithm. The two cases that complete the search can be thought of as the basis. Either you find an item that you want, or the sublist that you have been left to search in is empty, when $j > k$.

BinarySearch can be translated without much difficulty into any language that allows recursive calls to its subprograms. The advantage to such a program is that its coding would be much shorter than a nonrecursive program that does a binary search. However, in most cases the recursive version will be slower and require more memory at execution time.

8.1.7: Induction and Recursion

The definition of the positive integers in terms of Peano's Postulates is a recursive definition. The basis element is the number 1 and the recursion is that if n is a positive integer, then so is its successor. In this case, n is the simple object and the recursion is of a

forward type. Of course, the validity of an induction proof is based on our acceptance of this definition. Therefore, the appearance of induction proofs when recursion is used is no coincidence.

Example 8.1.4: Proof of a Formula for B

A formula for the sequence B in Example 8.1.2 is $B = 100(1.08)^k$ for $k \geq 0$. A proof by induction follow.

If $k = 0$, then $B = 100(1.08)^0 = 100$, as defined. Now assume that for some $k \geq 1$, the formula for B_k is true.

$$\begin{aligned} B_{k+1} &= 1.08 B_k \text{ by the recursive definition} \\ &= 1.08\left(100(1.08)^k\right) \text{ by the induction hypothesis} \\ &= 100(1.08)^{k+1} \end{aligned}$$

hence the formula is true for $k+1$

The formula that we have just proven for B is called a closed form expression. It involves no recursion or summation signs.

Definition 8.1.4: Closed Form Expression

Let $E = E\left(x_1, x_2, \ldots, x_n\right)$ be an algebraic expression involving variables $x_1, x_2, \ldots, x_n$ which are allowed to take on values from some predetermined set. E is a ***closed form expression*** if there exists a number T such that the evaluation of E with any allowed values of the variables will take no more than T operations (alternatively, T time units).

Example 8.1.5: Reducing a Summation to Closed Form

The sum $E(n) = \sum_{k=1}^{n} k$ is not a closed form expression because the number of additions needed to evaluate $E(n)$ grows indefinitely with n. A closed form expression that computes the value of $E(n)$ is $\frac{n(n+1)}{2}$, which only requires $T = 3$ operations.

8.1.8: Exercises

Exercise 8.1.1

By the recursive definition of binomial coefficients, $\binom{7}{2} = \binom{6}{2} + \binom{6}{1}$. Continue expanding $\binom{7}{2}$ to express it in terms of quantities defined by the basis. Check your result by applying the factorial definition of $\binom{n}{k}$.

Answer

$$
\begin{aligned}
\binom{7}{2} &= \binom{6}{2} + \binom{6}{1} \\
&= \binom{5}{2} + \binom{5}{1} + \binom{5}{1} + \binom{5}{0} \\
&= \binom{5}{2} + 2\binom{5}{1} + 1 \\
&= \binom{4}{2} + \binom{4}{1} + 2(\binom{4}{1} + \binom{4}{0}) + 1 \\
&= \binom{4}{2} + 3\binom{4}{1} + 3 \\
&= \binom{3}{2} + \binom{3}{1} + 3(\binom{3}{1} + \binom{3}{0}) + 3 \\
&= \binom{3}{2} + 4\binom{3}{1} + 6 \\
&= \binom{2}{2} + \binom{2}{1} + 4(\binom{2}{1} + \binom{2}{0}) + 6 \\
&= 5\binom{2}{1} + 11 \\
&= 5(\binom{1}{1} + \binom{1}{0}) + 11 \\
&= 21
\end{aligned}
$$

Exercise 8.1.2

Define the sequence L by $L_0 = 5$ and for $k \geq 1$, $L_k = 2L_{k-1} - 7$. Determine L_4 and prove by induction that $L_k = 7 - 2^{k+1}$.

Exercise 8.1.3

Let $p(x) = x^5 + 3x^4 - 15x^3 + x - 10$.

a. Write $p(x)$ in telescoping form.
b. Use a calculator to compute $p(3)$ using the original form of $p(x)$.
c. Use a calculator to compute $p(3)$ using the telescoping form of $p(x)$.
d. Compare your speed in parts b and c.

Answer

a. $p(x)$ in telescoping form: $((((x+3)x-15)x+0)x+1)x-10$
b. $p(3) = ((((3+3)3-15)3-0)3+1)3-10 = 74$

Exercise 8.1.4

Suppose that a list of nine items, $(r(l), r(2), \ldots, r(9))$, is sorted by key in decending order so that $r(3).\texttt{key} = 12$ and $r(4).\texttt{key} = 10$. List the executions of the BinarySearch algorithms that would be needed to complete BinarySearch(1,9) when:

a. The search key is C = 12
b. The search key is C = 11

Assume that distinct items have distinct keys.

Exercise 8.1.5

What is wrong with the following definition of $f : \mathbb{R} \to \mathbb{R}$? $f(0) = 1$ and $f(x) = f(x/2)/2$if $x \neq 0$.

Answer

The basis is not reached in a finite number of steps if you try to compute $f(x)$ for a nonzero value of x.

Exercise 8.1.6

Prove the two definitions of binomials coefficients, Definition 2.4.1 and Definition 8.1.1, are equivalent.

Exercise 8.1.7

Prove by induction that if $n \geq 0$, $\sum_{k=0}^{n} \binom{n}{k} = 2^n$

This page titled 8.1: The Many Faces of Recursion is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

8.2: Sequences

8.2.1: Sequences and Ways They Are Defined

Definition 8.2.1: Sequence

A sequence is a function from the natural numbers into some predetermined set. The image of any natural number k can be written as $S(k)$ or S_k and is called the kth ***term*** of S. The variable k is called the ***index*** or ***argument*** of the sequence.

For example, a sequence of integers would be a function $S : \mathbb{N} \to \mathbb{Z}$.

Example 8.2.1: Three Sequences Defined in Different Ways

a. The sequence A defined by $A(k) = k^2 - k,\ \ k \geq 0,$ is a sequence of integers.

b. The sequence B defined recursively by $B(0) = 2$ and $B(k) = B(k-1) + 3\ \ $ for $k \geq 1$ is a sequence of integers. The terms of B can be computed either by applying the recursion formula or by iteration. For example,

$$\begin{aligned} B(3) &= B(2) + 3 \\ &= (B(1) + 3) + 3 \\ &= ((B(0) + 3) + 3) + 3 \\ &= ((2 + 3) + 3) + 3 = 11 \end{aligned}$$

or

$$\begin{aligned} B(1) &= B(0) + 3 = 2 + 3 = 5 \\ B(2) &= B(1) + 3 = 5 + 3 = 8 \\ B(3) &= B(2) + 3 = 8 + 3 = 11 \end{aligned}$$

c. Let C_r be the number of strings of 0's and 1's of length r having no consecutive zeros. These terms define a sequence C of integers.

Remarks:

1. A sequence is often called a *discrete function*.
2. Although it is important to keep in mind that a sequence is a function, another useful way of visualizing a sequence is as a list. For example, the sequence A in the previous example could be written as $(0, 0, 2, 6, 12, 20, \ldots)$.Finite sequences can appear much the same way when they are the input to or output from a computer. The index of a sequence can be thought of as a time variable. Imagine the terms of a sequence flashing on a screen every second. Then s_k would be what you see in the kth second. It is convenient to use terminology like this in describing sequences. For example, the terms that precede the kth term of A would be $A(0), A(1), \ldots, A(k-1)$. They might be called the earlier terms.

8.2.2: Fundamental Problem

Given the definition of any sequence, a fundamental problem that we will concern ourselves with is to devise a method for determining any specific term in a minimum amount of time. Generally, time can be equated with the number of operations needed. In counting operations, the application of a recursive formula would be considered an operation.

a. The terms of A in Example $8.2.1$ are very easy to compute because of the closed form expression. No matter what term you decide to compute, only three operations need to be performed.

b. How to compute the terms of B is not so clear. Suppose that you wanted to know $B(100)$. One approach would be to apply the definition recursively:

$$B(100) = B(99) + 3 = (B(98) + 3) + 3 = \cdots$$

The recursion equation for B would be applied 100 times and 100 additions would then follow. To compute $B(k)$ by this method, $2k$ operations are needed. An iterative computation of $B(k)$ is an improvement: $B(1) = B(0) + 3 = 2 + 3 = 5$

$B(2) = B(1) + 3 = 5 + 3 = 8$ etc. Only k additions are needed. This still isn't a good situation. As k gets large, we take more and more time to compute $B(k)$. The formula $B(k) = B(k-1) + 3$ is called a recurrence relation on B. The process of finding a closed form expression for $B(k)$, one that requires no more than some fixed number of operations, is called solving the recurrence relation.

c. The determination of C_k is a standard kind of problem in combinatorics. One solution is by way of a recurrence relation. In fact, many problems in combinatorics are most easily solved by first searching for a recurrence relation and then solving it. The following observation will suggest the recurrence relation that we need to determine C_k. If $k \geq 2$, then every string of 0's and 1's with length k and no two consecutive 0's is either $1s_{k-1}$ or $01s_{k-2}$, where s_{k-1} and s_{k-2} are strings with no two consecutive 0's of length $k-1$ and $k-2$ respectively. From this observation we can see that $C_k = C_{k-2} + C_{k-1}$ for $k \geq 2$. The terms $C_0 = 1$ and $C_1 = 2$ are easy to determine by enumeration. Now, by iteration, any C_k can be easily determined. For example, $C_5 = 21$ can be computed with five additions. A closed form expression for C_k would be an improvement. Note that the recurrence relation for C_k is identical to the one for The Fibonacci Sequence, Example 8.1.3. Only the basis is different.

8.2.3: Exercises

Exercise 8.2.1

Prove by induction that $B(k) = 3k + 2,\ k \geq 0$, is a closed form expression for the sequence B in Example 8.2.1

Answer

Basis: $B(0) = 3 \cdot 0 + 2 = 2$, as defined.

Induction: Assume: $B(k) = 3k + 2$ for some $k \geq 0$.

$$\begin{aligned} B(k+1) &= B(k) + 3 \\ &= (3k+2) + 3 \quad \text{by the induction hypothesis} \\ &= (3k+3) + 2 \\ &= 3(k+1) + 2 \quad \text{as desired} \end{aligned}$$

Exercise 8.2.2

a. Consider sequence Q defined by $Q(k) = 2k + 9,\ k \geq 1$. Complete the table below and determine a recurrence relation that describes Q.

k	$Q(k)$	$Q(k) - Q(k-1)$
2		
3		
4		
5		
6		
7		

b. Let $A(k) = k^2 - k,\ k \geq 0$. Complete the table below and determine a recurrence relation for A.

k	$A(k)$	$A(k) - A(k-1)$	$A(k) - 2A(k-1) + A(k-2)$
2			
3			
4			
5			

Exercise 8.2.3

Given k lines ($k \geq 0$) on a plane such that no two lines are parallel and no three lines meet at the same point, let $P(k)$ be the number of regions into which the lines divide the plane (including the infinite ones (see Figure 8.2.1). Describe how the recurrence relation $P(k) = P(k-1) + k$ can be derived. Given that $P(0) = 1$, determine $P(5)$.

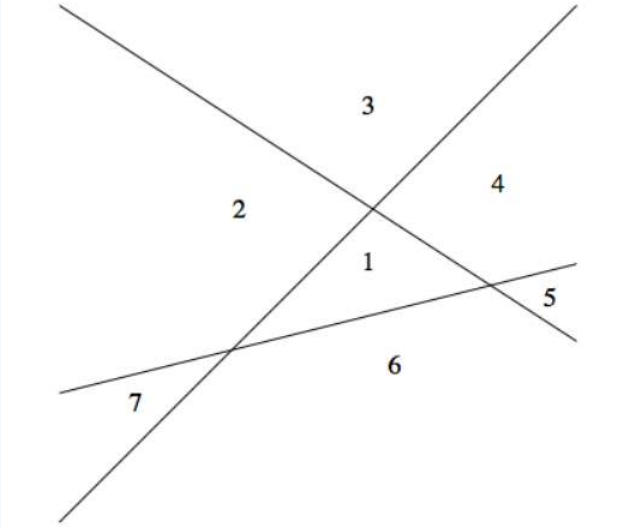


Figure 8.2.1: A general configuration of three lines

Answer

Imagine drawing line k in one of the infinite regions that it passes through. That infinite region is divided into two infinite regions by line k. As line k is drawn through every one of the $k-1$ previous lines, you enter another region that line k divides. Therefore k regions are divided and the number of regions is increased by k. From this observation we get $P(5)=16$.

Exercise 8.2.4

A sample of a radioactive substance is expected to decay by 0.15 percent each hour. If w_t, $t \geq 0$, is the weight of the sample t hours into an experiment, write a recurrence relation for w.

Exercise 8.2.5

Let $M(n)$ be the number of multiplications needed to evaluate an nth degree polynomial. Use the recursive definition of a polynomial expression to define M recursively.

Answer

For n greater than zero, $M(n)=M(n-1)+1$, and $M(0)=0$.

Exercise 8.2.6

Let S be sequence of integers. Using short English sentences, not symbols, describe what the following propositions say about S. Are the two propositions equivalent?

a. $(\forall M)_{\mathbb{N}}((\exists n)_{\mathbb{N}}(S(n) \geq M))$
b. $(\forall M)_{\mathbb{N}}((\exists N)_{\mathbb{N}}((\forall n)_{\mathbb{N}}(n \geq N \rightarrow S(n) \geq M)))$

This page titled 8.2: Sequences is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

8.3: Recurrence Relations

In this section we will begin our study of recurrence relations and their solutions. Our primary focus will be on the class of finite order linear recurrence relations with constant coefficients (shortened to finite order linear relations). First, we will examine closed form expressions from which these relations arise. Second, we will present an algorithm for solving them. In later sections we will consider some other common relations (8.4) and introduce two additional tools for studying recurrence relations: generating functions (8.5) and matrix methods (Chapter 12).

8.3.1: Definition and Terminology

Definition 8.3.1: Recurrence Relation

Let S be a sequence of numbers. A recurrence relation on S is a formula that relates all but a finite number of terms of S to previous terms of S. That is, there is a k_0 in the domain of S such that if $k \geq k_0$, then $S(k)$ is expressed in terms of some (and possibly all) of the terms that precede $S(k)$. If the domain of S is $\{0, 1, 2, \ldots\}$,the terms $S(0), S(1), \ldots, S(k_0 - 1)$ are not defined by the recurrence formula. Their values are the initial conditions (or boundary conditions, or basis) that complete the definition of S.

Example 8.3.1: Some Examples of Recurrence Relations

a. The Fibonacci sequence is defined by the recurrence relation $F_k = F_{k-2} + F_{k-1},\ k \geq 2$, with the initial conditions $F_0 = 1$ and $F_1 = 1$. The recurrence relation is called a second-order relation because F_k depends on the two previous terms of F. Recall that the sequence C in Section 8.2, Example 8.2.1, can be defined with the same recurrence relation, but with different initial conditions.
b. The relation $T(k) = 2T(k-1)^2 - kT(k-3)$ is a third-order recurrence relation. If values of $T(0)$, $T(1)$, and $T(2)$ are specified, then T is completely defined.
c. The recurrence relation $S(n) = S(\lfloor n/2 \rfloor) + 5,\ n > 0$, with $S(0) = 0$ has infinite order. To determine $S(n)$ when n is even, you must go back $n/2$ terms. Since $n/2$ grows unbounded with n, no finite order can be given to S.

8.3.2: Solving Recurrence Relations

Sequences are often most easily defined with a recurrence relation; however, the calculation of terms by directly applying a recurrence relation can be time-consuming. The process of determining a closed form expression for the terms of a sequence from its recurrence relation is called solving the relation. There is no single technique or algorithm that can be used to solve all recurrence relations. In fact, some recurrence relations cannot be solved. The relation that defines T above is one such example. Most of the recurrence relations that you are likely to encounter in the future are classified as finite order linear recurrence relations with constant coefficients. This class is the one that we will spend most of our time with in this chapter.

Definition 8.3.2: n^{th} Order Linear Recurrence Relation

Let S be a sequence of numbers with domain $k \geq 0$. An nth order linear recurrence relation on S with constant coefficients is a recurrence relation that can be written in the form

$$S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = f(k) \text{ for } k \geq n$$

where $C_1, C_2, \ldots, C_n$ are constants and f is a numeric function that is defined for $k \geq n$.

Note 8.3.1

We will shorten the name of this class of relations to n^{th} order linear relations. Therefore, in further discussions, $S(k) + 2kS(k-1) = 0$ would not be considered a first-order linear relation.

Example 8.3.2: Some Finite Order Linear Relations

a. The Fibonacci sequence is defined by the second-order linear relation because $F_k - F_{k-1} - F_{k-2} = 0$
b. The relation $P(j) + 2P(j-3) = j^2$ is a third-order linear relation. In this case, $C_1 = C_2 = 0$.
c. The relation $A(k) = 2(A(k-1) + k)$ can be written as $A(k) - 2A(k-1) = 2k$. Therefore, it is a first-order linear relation.

8.3.3: Recurrence Relations Obtained from "Solutions"

Before giving an algorithm for solving finite order linear relations, we will examine recurrence relations that arise from certain closed form expressions. The closed form expressions are selected so that we will obtain finite order linear relations from them. This approach may seem a bit contrived, but if you were to write down a few simple algebraic expressions, chances are that most of them would be similar to the ones we are about to examine.

For our first example, consider D, defined by $D(k) = 5 \cdot 2^k$, $k \geq 0$. If $k \geq 1$, $D(k) = 5 \cdot 2^k = 2 \cdot 5 \cdot 2^{k-1} = 2D(k-1)$. Therefore, D satisfies the first order linear relation $D(k) - 2D(k-1) = 0$ and the initial condition $D(0) = 5$ serves as an initial condition for D.

As a second example, consider $C(k) = 3^{k-1} + 2^{k+1} + k$, $k \geq 0$. Quite a bit more algebraic manipulation is required to get our result:

Table 8.3.1

$C(k) = 3^{k-1} + 2^{k+1} + k$	Original equation
$3C(k-1) = 3^{k-1} + 3 \cdot 2^k + 3(k-1)$	Substitute $k-1$ for k and multiply by 3
	Subtract the second equation from the first.
$C(k) - 3C(k-1) = -2^k - 2k + 3$	3^{k-1} term is eliminated. This is a first order relation.
$2C(k-1) - 6C(k-2) = -2^k - 2(2(k-1)) + 6$	Substitute $k-1$ for k in the third equation, multiply by 2.
$C(k) - 5C(k-1) + 6C(k-2) = 2k - 7$	Subtract the 4th equation from the 3rd. 2^{k+1} term is eliminated. This is 2nd order relation.

The recurrence relation that we have just obtained, defined for $k \geq 2$, together with the initial conditions $C(0) = 7/3$ and $C(1) = 6$, define C.

Table 8.3.2 summarizes our results together with a few other examples that we will let the reader derive. Based on these results, we might conjecture that any closed form expression for a sequence that combines exponential expressions and polynomial expressions will be solutions of finite order linear relations. Not only is this true, but the converse is true: a finite order linear relation defines a closed form expression that is similar to the ones that were just examined. The only additional information that is needed is a set of initial conditions.

Table 8.3.2: Recurrence relations obtained from given sequences

Closed Form Expression	Recurrence Relation
$D(k) = 5 \cdot 2^k$	$D(k) - 2D(k-1) = 0$
$C(k) = 3^{k-1} + 2^{k+1} + k$	$C(k) - 2C(k-1) - 6C(k-2) = 2k - 7$
$Q(k) = 2k + 9$	$Q(k) - Q(k-1) = 2$
$A(k) = k^2 - k$	$A(k) - 2A(k-1) + A(k-2) = 2$
$B(k) = 2k^2 + 1$	$B(k) - 2B(k-1) + B(k-2) = 4$
$G(k) = 2 \cdot 4^k - 5(-3)^k$	$G(k) - G(k-1) + 12G(k-2) = 0$

Closed Form Expression	Recurrence Relation
$J(k) = (3+k)2^k$	$J(k) - 4J(k-1) + 4J(k-2) = 0$

Definition 8.3.3: Homogeneous Recurrence Relation

An nth order linear relation is homogeneous if $f(k) = 0$ for all k. For each recurrence relation $S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = f(k)$, the associated homogeneous relation is $S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = 0$

Example 8.3.3: First Order Homogeneous Recurrence Relations

$D(k) - 2D(k-1) = 0$ is a first-order homogeneous relation. Since it can also be written as $D(k) = 2D(k-1)$, it should be no surprise that it arose from an expression that involves powers of 2. More generally, you would expect that the solution of $L(k) - aL(k-1)$ would involve a^k. Actually, the solution is $L(k) = L(0)a^k$, where the value of $L(0)$ is given by the initial condition.

Example 8.3.4: A Second Order Example

Consider the second-order homogeneous relation $S(k) - 7S(k-1) + 12S(k-2) = 0$ together with the initial conditions $S(0) = 4$ and $S(1) = 4$. From our discussion above, we can predict that the solution to this relation involves terms of the form ba^k, where b and a are nonzero constants that must be determined. If the solution were to equal this quantity exactly, then

$$S(k) = ba^k$$
$$S(k-1) = ba^{k-1}$$
$$S(k-2) = ba^{k-2}$$

Substitute these expressions into the recurrence relation to get

$$ba^k - 7ba^{k-1} + 12ba^{k-2} = 0$$

Each term on the left-hand side of this equation has a factor of ba^{k-2}, which is nonzero. Dividing through by this common factor yields

$$a^2 - 7a + 12 = (a-3)(a-4) = 0 \tag{8.3.1}$$

Therefore, the only possible values of a are 3 and 4. Equation (8.3.1) is called the characteristic equation of the recurrence relation. The fact is that our original recurrence relation is true for any sequence of the form $S(k) = b_1 3^k + b_2 4^k$, where b_1 and b_2 are real numbers. This set of sequences is called the general solution of the recurrence relation. If we didn't have initial conditions for S, we would stop here. The initial conditions make it possible for us to find definite values for b_1 and b_2.

$$\left\{ \begin{array}{c} S(0) = 4 \\ S(1) = 4 \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} b_1 3^0 + b_2 4^0 = 4 \\ b_1 3^1 + b_2 4^1 = 4 \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} b_1 + b_2 = 4 \\ 3b_1 + 4b_2 = 4 \end{array} \right\}$$

The solution of this set of simultaneous equations is $b_1 = 12$ and $b_2 = -8$ and so the solution is $S(k) = 12 \cdot 3^k - 8 \cdot 4^k$.

Definition 8.3.4: Characteristic Equation

The characteristic equation of the homogeneous n^{th} order linear relation $S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = 0$ is the nth degree polynomial equation

$$a^n + \sum_{j=1}^{n} C_j a^{n-j} = a^n + C_1 a^{n-1} + \cdots + C_{n-1} a + C_n = 0$$

The left-hand side of this equation is called the characteristic polynomial. The roots of the characteristic polynomial are called the characteristic roots of the equation.

Example 8.3.5: Some Characteristic Equations

a. The characteristic equation of $F(k)-F(k-1)-F(k-2)=0$ is $a^2-a-1=0$.
b. The characteristic equation of $Q(k)+2Q(k-1)-3Q(k-2)-6Q(k-4)=0$ is $a^4+2a^3-3a^2-6=0$. Note that the absence of a $Q(k-3)$ term means that there is not an $x^{4-3}=x$ term appearing in the characteristic equation.

Algorithm 8.3.1: Algorithm for Solving Homogeneous Finite Order Linear Relations

a. Write out the characteristic equation of the relation $S(k)+C_1S(k-1)+\ldots+C_nS(k-n)=0,$ which is $a^n+C_1a^{n-1}+\cdots+C_{n-1}a+C_n=0$.
b. Find all roots of the characteristic equation, the characteristic roots.
c. If there are n distinct characteristic roots, $a_1, a_2, \ldots a_n$, then the general solution of the recurrence relation is $S(k)=b_1{a_1}^k+b_2{a_2}^k+\cdots+b_n{a_n}^k$. If there are fewer than n characteristic roots, then at least one root is a multiple root. If a_j is a double root, then the $b_j{a_j}^k$ term is replaced with $(b_{j0}+b_{j1}k)\,a_j^k$. In general, if a_j is a root of multiplicity p, then the $b_j{a_j}^k$ term is replaced with $\left(b_{j0}+b_{j1}k+\cdots+b_{j(p-1)}k^{p-1}\right)a_j^k$.
d. If n initial conditions are given, we get n linear equations in n unknowns (the $b_j's$ from Step 3) by substitution. If possible, solve these equations to determine a final form for $S(k)$.

Although this algorithm is valid for all values of n, there are limits to the size of n for which the algorithm is feasible. Using just a pencil and paper, we can always solve second-order equations. The quadratic formula for the roots of $ax^2+bx+c=0$ is

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

The solutions of $a^2+C_1a+C_2=0$ are then

$$\frac{1}{2}\left(-C_1+\sqrt{{C_1}^2-4C_2}\right) \text{ and } \frac{1}{2}\left(-C_1-\sqrt{{C_1}^2-4C_2}\right)$$

Although cubic and quartic formulas exist, they are too lengthy to introduce here. For this reason, the only higher-order relations ($n\geq 3$) that you could be expected to solve by hand are ones for which there is an easy factorization of the characteristic polynomial.

Example 8.3.6: A Solution Using the Algorithm

Suppose that T is defined by $T(k)=7T(k-1)-10T(k-2),$ with $T(0)=4$ and $T(1)=17$. We can solve this recurrence relation with Algorithm 8.3.1:

a. Note that we have written the recurrence relation in "nonstandard" form. To avoid errors in this easy step, you might consider a rearrangement of the equation to, in this case, $T(k)-7T(k-1)+10T(k-2)=0$. Therefore, the characteristic equation is $a^2-7a+10=0$.
b. The characteristic roots are $\frac{1}{2}\left(7+\sqrt{49-40}\right)=5$ and $\frac{1}{2}\left(7-\sqrt{49-40}\right)=2$. These roots can be just as easily obtained by factoring the characteristic polynomial into $(a-5)(a-2)$.
c. The general solution of the recurrence relation is $T(k)=b_1{2}^k+b_2{5}^k$.
d. $\left\{\begin{array}{c}T(0)=4\\T(1)=17\end{array}\right\}\Rightarrow\left\{\begin{array}{c}b_1{2}^0+b_2{5}^0=4\\b_1{2}^1+b_2{5}^1=17\end{array}\right\}\Rightarrow\left\{\begin{array}{c}b_1+b_2=4\\2b_1+5b_2=17\end{array}\right\}$ The simultaneous equations have the solution $b_1=1$ and $b_2=3$. Therefore, $T(k)=2^k+3\cdot 5^k$.

Here is one rule that might come in handy: If the coefficients of the characteristic polynomial are all integers, with the constant term equal to m, then the only possible rational characteristic roots are divisors of m (both positive and negative).

With the aid of a computer (or possibly only a calculator), we can increase n. Approximations of the characteristic roots can be obtained by any of several well-known methods, some of which are part of standard software packages. There is no general rule that specifies the values of n for which numerical approximations will be feasible. The accuracy that you get will depend on the relation that you try to solve. (See Exercise 8.3.17of this section.)

Example 8.3.7: Solution of a Third Order Recurrence Relation

Solve $S(k)-7S(k-2)+6S(k-3)=0$, where $S(0)=8$, $S(1)=6$, and $S(2)=22$.

a. The characteristic equation is $a^3-7a+6=0$.
b. The only rational roots that we can attempt are $\pm1, \pm2, \pm3,$ and ±6. By checking these, we obtain the three roots 1, 2, and -3.
c. The general solution is $S(k)=b_1 1^k+b_2 2^k+b_3(-3)^k$. The first term can simply be written b_1.
d. $\left\{\begin{array}{c} S(0)=8 \\ S(1)=6 \\ S(2)=22 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1+b_2+b_3=8 \\ b_1+2b_2-3b_3=6 \\ b_1+4b_2+9b_3=22 \end{array}\right\}$ You can solve this system by elimination to obtain $b_1=5$, $b_2=2$, and $b_3=1$. Therefore, $S(k)=5+2\cdot 2^k+(-3)^k=5+2^{k+1}+(-3)^k$

Example 8.3.8: Solution with a Double Characteristic Root

Solve $D(k)-8D(k-1)+16D(k-2)=0$, where $D(2)=16$ and $D(3)=80$.

a. Characteristic equation: $a^2-8a+16=0$.
b. $a^2-8a+16=(a-4)^2$. Therefore, there is a double characteristic root, 4.
c. General solution: $D(k)=(b_{10}+b_{11}k)\,4^k$.
d. $\left\{\begin{array}{c} D(2)=16 \\ D(3)=80 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} (b_{10}+b_{11}2)\,4^2=16 \\ (b_{10}+b_{11}3)\,4^3=80 \end{array}\right\}$
$\Rightarrow \left\{\begin{array}{c} 16b_{10}+32b_{11}=16 \\ 64b_{10}+192b_{11}=80 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_{10}=\frac{1}{2} \\ b_{11}=\frac{1}{4} \end{array}\right\}$
Therefore $D(k)=(1/2+(1/4)k)4^k=(2+k)4^{k-1}$.

8.3.4: Solution of Nonhomogeneous Finite Order Linear Relations

Our algorithm for nonhomogeneous relations will not be as complete as for the homogeneous case. This is due to the fact that different right-hand sides ($f(k)$'s) call for different rules in obtaining a particular solution.

Algorithm 8.3.2: Algorithm for Solving Nonhomogeneous Finite Order Linear Relations

To solve the recurrence relation $S(k)+C_1S(k-1)+\ldots+C_nS(k-n)=f(k)$

a. Write the associated homogeneous relation and find its general solution (Steps (a) through (c) of Algorithm 8.3.1). Call this the homogeneous solution, $S^{(h)}(k)$.
b. Start to obtain what is called a particular solution, $S^{(p)}(k)$ of the recurrence relation by taking an educated guess at the form of a particular solution. For a large class of right-hand sides, this is not really a guess, since the particular solution is often the same type of function as $f(k)$ (see Table 8.3.3).
c. Substitute your guess from Step 2 into the recurrence relation. If you made a good guess, you should be able to determine the unknown coefficients of your guess. If you made a wrong guess, it should be apparent from the result of this substitution, so go back to Step 2.
d. The general solution of the recurrence relation is the sum of the homogeneous and particular solutions. If no conditions are given, then you are finished. If n initial conditions are given, they will translate to n linear equations in n unknowns and solve the system to get a complete solution.

Table 8.3.3: Particular solutions for given right-hand sides

Right Hand Side, $f(k)$	Form of Particular Solution, $S^{(p)}(k)$
Constant, q	Constant, d
Linear Function, q_0+q_1k	Linear Function, d_0+d_1k
m^{th} degree polynomial, $q_0+q_1k+\cdots+q_mk^m$	m^{th} degree polynomial, $d_0+d_1k+\cdots+d_mk^m$

exponential function, qa^k	exponential function, da^k

Example 8.3.9: Solution of a Nonhomogeneous First Order Recurrence Relation

Solve $S(k)+5S(k-1)=9,$ with $S(0)=6.$

a. The associated homogeneous relation, $S(k)+5S(k-1)=0$ has the characteristic equation $a+5=0;$ therefore, $a=-5.$ The homogeneous solution is $S^{(h)}(k)=b(-5)^k.$
b. Since the right-hand side is a constant, we guess that the particular solution will be a constant, $d.$
c. If we substitute $S^{(p)}(k)=d$ into the recurrence relation, we get $d+5d=9,$ or $6d=9.$ Therefore, $S^{(p)}(k)=1.5.$
d. The general solution of the recurrence relation is $S(k)=S^{(h)}(k)+S^{(p)}(k)=b(-5)^k+1.5$ The initial condition will give us one equation to solve in order to determine $b.$ $S(0)=6 \Rightarrow b(-5)^0+1.5=6 \Rightarrow b+1.5=6$ Therefore, $b=4.5$ and $S(k)=4.5(-5)^k+1.5.$

Example 8.3.10: Solution of a Nonhomogeneous Second Order Recurrence Relation

Consider $T(k)-7T(k-1)+10T(k-2)=6+8k$ with $T(0)=1$ and $T(1)=2.$

a. From Example 8.3.6, we know that $T^{(h)}(k)=b_1 2^k+b_2 5^k.$ Caution:Don't apply the initial conditions to $T^{(h)}$ until you add $T^{(p)}$!
b. Since the right-hand side is a linear polynomial, $T^{(p)}$ is linear; that is, $T^{(p)}(k)=d_0+d_1 k.$
c. Substitution into the recurrence relation yields: $(d_0+d_1 k)-7\left(d_0+d_1(k-1)\right)+10\left(d_0+d_1(k-2)\right)=6+8k$
$\Rightarrow (4d_0-13d_1)+(4d_1)\,k=6+8k$ Two polynomials are equal only if their coefficients are equal. Therefore,
$\left\{\begin{array}{c} 4d_0-13d_1=6 \\ 4d_1=8 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} d_0=8 \\ d_1=2 \end{array}\right\}$
d. Use the general solution $T(k)=b_1 2^k+b_2 5^k+8+2k$ and the initial conditions to get a final solution:
$\left\{\begin{array}{c} T(0)=1 \\ T(1)=2 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1+b_2+8=1 \\ 2b_1+5b_2+10=2 \end{array}\right\}$
$\Rightarrow \left\{\begin{array}{c} b_1+b_2=-7 \\ 2b_1+5b_2=-8 \end{array}\right\}$
$\Rightarrow \left\{\begin{array}{c} b_1=-9 \\ b_2=2 \end{array}\right\}$
Therefore, $T(k)=-9\cdot 2^k+2\cdot 5^k+8+2k.$

Note 8.3.2: A Quick Note on Interest Rates

When a quantity, such as a savings account balance, is increased by some fixed percent, it is most easily computed with a multiplier. In the case of an 8% increase, the multiplier is 1.08 because any original amount $A,$ has $0.08A$ added to it, so that the new balance is $A+0.08A=(1+0.08)A=1.08A.$

Another example is that if the interest rate is $3.5\%,$ the multiplier would be 1.035. This presumes that the interest is applied at the end of year for 3.5% annual interest, often called ***simple interest***. If the interest is applied monthly, and we assume a simplifed case where each month has the same length, the multiplier after every month would be $\left(1+\frac{0.035}{12}\right)\approx 1.00292.$ After a year passes, this multiplier would be applied 12 times, which is the same as multiplying by $1.00292^{12}\approx 1.03557.$ That increase from 1.035 to 1.03557 is the effect of ***compound interest***.

Example 8.3.11: A Sort of Annuity

Suppose you open a savings account that pays an annual interest rate of $8\%.$ In addition, suppose you decide to deposit one dollar when you open the account, and you intend to double your deposit each year. Let $B(k)$ be your balance after k years. B can be described by the relation $B(k)=1.08B(k-1)+2^k,$ with $S(0)=1.$ If, instead of doubling the deposit each year, you deposited a constant amount, $q,$ the 2^k term would be replaced with $q.$ A sequence of regular deposits such as this is called a simple annuity.

Returning to the original situation,

a. $B^{(h)}(k) = b_1(1.08)^k$
b. $B^{(p)}(k)$ should be of the form $d2^k$.
c.
$$\begin{aligned} d2^k = 1.08d2^{k-1} + 2^k &\Rightarrow (2d)2^{k-1} = 1.08d2^{k-1} + 2\cdot 2^{k-1} \\ &\Rightarrow 2d = 1.08d + 2 \\ &\Rightarrow .92d = 2 \\ &\Rightarrow d = 2.174 \text{ to the nearest thousandth)} \end{aligned}$$

Therefore $B^{(p)}(k) = 2.174 \cdot 2^k$.
d. $B(0) = 1 \Rightarrow b_1 + 2.174 = 1$
$\Rightarrow b_1 = -1.174$
Therefore, $B(k) = -1.174 \cdot 1.08^k + 2.174 \cdot 2^k$.

Example 8.3.12: Matching Roots

Find the general solution to $S(k) - 3S(k-1) - 4S(k-2) = 4^k$.

a. The characteristic roots of the associated homogeneous relation are -1 and 4. Therefore, $S^{(h)}(k) = b_1(-1)^k + b_2 4^k$.
b. A function of the form $d4^k$ will not be a particular solution of the nonhomogeneous relation since it solves the associated homogeneous relation. When the right-hand side involves an exponential function with a base that equals a characteristic root,you should multiply your guess at a particular solution by k. Our guess at $S^{(p)}(k)$ would then be $dk4^k$. See Observation 8.3.1 for a more complete description of this rule.
c. Substitute $dk4^k$ into the recurrence relation for $S(k)$:
$$dk4^k - 3d(k-1)4^{k-1} - 4d(k-2)4^{k-2} = 4^k$$
$$16dk4^{k-2} - 12d(k-1)4^{k-2} - 4d(k-2)4^{k-2} = 4^k$$

Each term on the left-hand side has a factor of 4^{k-2}
$$16dk - 12d(k-1) - 4d(k-2) = 4^2 20d = 16 \Rightarrow d = 0.8$$

Therefore, $S^{(p)}(k) = 0.8k4^k$.
d. The general solution to the recurrence relation is
$$S(k) = b_1(-1)^k + b_2 4^k + 0.8k4^k$$

Observation 8.3.1: When the Base of Right-Hand Side is Equal to a Characteristic Root

If the right-hand side of a nonhomogeneous relation involves an exponential with base a, and a is also a characteristic root of multiplicity p, then multiply your guess at a particular solution as prescribed in Table 8.3.3 by k^p, where k is the index of the sequence.

Example 8.3.13: Examples of Matching Bases

a. If $S(k) - 9S(k-1) + 20S(k-2) = 2\cdot 5^k$, the characteristic roots are 4 and 5. Since 5 matches the base of the right side, $S^{(p)}(k)$ will take the form $dk5^k$.
b. If $S(n) - 6S(n-1) + 9S(n-2) = 3^{n+1}$ the only characteristic root is 3, but it is a double root (multiplicity 2). Therefore, the form of the particular solution is $dn^2 3^n$.
c. If $Q(j) - Q(j-1) - 12Q(j-2) = (-3)^j + 6\cdot 4^j$, the characteristic roots are -3 and 4. The form of the particular solution will be $d_1 j(-3)^j + d_2 j\cdot 4^j$.

d. If $S(k)-9S(k-1)+8S(k-2)=9k+1=(9k+1)1^k$, the characteristic roots are 1 and 8. If the right-hand side is a polynomial, as it is in this case, then the exponential factor 1^k can be introduced. The particular solution will take the form $k\left(d_0+d_1k\right)$.

We conclude this section with a comment on the situation in which the characteristic equation gives rise to complex roots. If we restrict the coefficients of our finite order linear relations to real numbers, or even to integers, we can still encounter characteristic equations whose roots are complex. Here, we will simply take the time to point out that our algorithms are still valid with complex characteristic roots, but the customary method for expressing the solutions of these relations is different. Since an understanding of these representations requires some background in complex numbers, we will simply suggest that an interested reader can refer to a more advanced treatment of recurrence relations (see also difference equations).

8.3.5: Exercises

Solve the following sets of recurrence relations and initial conditions:

Exercise 8.3.1

$S(k)-10S(k-1)+9S(k-2)=0, \quad S(0)=3,\ S(1)=11$

Answer

$S(k)=2+9^k$

Exercise 8.3.2

$S(k)-9S(k-1)+18S(k-2)=0, \quad S(0)=0,\ S(1)=3$

Exercise 8.3.3

$S(k)-0.25S(k-1)=0, \ S(0)=6$

Answer

$S(k)=6(1/4)^k$

Exercise 8.3.4

$S(k)-20S(k-1)+100S(k-2)=0, \quad S(0)=2,\ S(1)=50$

Exercise 8.3.5

$S(k)-2S(k-1)+S(k-2)=2, S(0)=25, \ S(1)=16$

Answer

$S(k)=k^2-10k+25$

Exercise 8.3.6

$S(k)-S(k-1)-6S(k-2)=-30, \quad S(0)=7,\ S(1)=6$

Exercise 8.3.7

$S(k)-5S(k-1)=5^k, \ S(0)=3$

Answer

$S(k) = (3+k)5^k$

Exercise 8.3.8

$S(k) - 5S(k-1) + 6S(k-2) = 2,\ S(0) = -1,\ S(1) = 0$

Exercise 8.3.9

$S(k) - 4S(k-1) + 4S(k-2) = 3k + 2^k,\ S(0) = 1, S(1) = 1$

Answer

$S(k) = (12+3k) + \left(k^2 + 7k - 22\right) 2^{k-1}$

Exercise 8.3.10

$S(k) = rS(k-1) + a, \quad S(0) = 0,\ r, a \geq 0, r \neq 1$

Exercise 8.3.11

$S(k) - 4S(k-1) - 11S(k-2) + 30S(k-3) = 0, \quad S(0) = 0, S(1) = -35,\ S(2) = -85$

Answer

$P(k) = 4(-3)^k + 2^k - 5^{k+1}$

Exercise 8.3.12

Find a closed form expression for $P(k)$ in Exercise 8.2.3 of Section 8.2.

Exercise 8.3.13

a. Find a closed form expression for the terms of the Fibonacci sequence (see Example 8.1.3).
b. The sequence C was defined by C_r = the number of strings of zeros and ones with length r having no consecutive zeros (Example 8.2.1(c)). Its recurrence relation is the same as that of the Fibonacci sequence. Determine a closed form expression for C_r, $r \geq 1$.

Answer

a. The characteristic equation is $a^2 - a - 1 = 0,$ which has solutions $\alpha = \left(1+\sqrt{5}\right)/2$ and $\beta = \left(1-\sqrt{5}\right)/2,$ It is useful to point out that $\alpha + \beta = 1$ and $\alpha - \beta = \sqrt{5}.$ The general solution is $F(k) = b_1\alpha^k + b_2\beta^k.$ Using the initial conditions, we obtain the system: $b_1 + b_2 = 1$ and $b_1\alpha + b_2\beta = 1.$ The solution to this system is $b_1 = \alpha/(\alpha-\beta) = \left(5+\sqrt{5}\right)/2\sqrt{5}$ and $b_2 = \beta/(\alpha-\beta) = \left(5-\sqrt{5}\right)/2\sqrt{5}$
Therefore the final solution is

$$\begin{aligned} F(n) &= \frac{\alpha^{n+1} - \beta^{n+1}}{\alpha - \beta} \\ &= \frac{\left(\left(1+\sqrt{5}\right)/2\right)^{n+1} - \left(\left(1-\sqrt{5}\right)/2\right)^{n+1}}{\sqrt{5}} \end{aligned}$$

b. $C_r = F(r+1)$

Exercise 8.3.14

If $S(n)=\sum_{j=1}^{n} g(j),\ n\geq 1$, then S can be described with the recurrence relation $S(n)=S(n-1)+g(n)$. For each of the following sequences that are defined using a summation, find a closed form expression:

a. $S(n)=\sum_{j=1}^{n} j,\ n\geq 1$
b. $Q(n)=\sum_{j=1}^{n} j^2,\ n\geq 1$
c. $P(n)=\sum_{j=1}^{n} \left(\frac{1}{2}\right)^j,\ n\geq 0$
d. $T(n)=\sum_{j=1}^{n} j^3,\ n\geq 1$

Exercise 8.3.15

Let $D(n)$ be the number of ways that the set $\{1,2,\ldots,n\}$,$n\geq 1$, can be partitioned into two nonempty subsets.

a. Find a recurrence relation for D. (Hint: It will be a first-order linear relation.)
b. Solve the recurrence relation.

Answer

a. For each two-block partition of $\{1,2,\ldots,n-1\}$, there are two partitions we can create when we add n, but there is one additional two-block partition to count for which one block is $\{n\}$. Therefore, $D(n)=2D(n-1)+1$ for $n\geq 2$ and $D(1)=0$.
b. $D(n)=2^{n-1}-1$

Exercise 8.3.16

If you were to deposit a certain amount of money at the end of each year for a number of years, this sequence of payments would be called an annuity (see Example 8.3.11).

a. Find a closed form expression for the balance or value of an annuity that consists of payments of q dollars at a rate of interest of i. Note that for a normal annuity, the first payment is made after one year.
b. With an interest rate of 5.5 percent, how much would you need to deposit into an annuity to have a value of one million dollars after 18 years?
c. The payment of a loan is a form of annuity in which the initial value is some negative amount (the amount of the loan) and the annuity ends when the value is raised to zero. How much could you borrow if you can afford to pay $5,000 per year for 25 years at 11 percent interest?

Exercise 8.3.17

Suppose that C is a small positive number. Consider the recurrence relation $B(k)-2B(k-1)+\left(1-C^2\right)B(k-2)=C^2$, with initial conditions $B(0)=1$ and $B(1)=1$. If C is small enough, we might consider approximating the relation by replacing $1-C^2$ with 1 and C^2 with 0. Solve the original relation and its approximation. Let B_a a be the solution of the approximation. Compare closed form expressions for $B(k)$ and $B_a(k)$. Their forms are very different because the characteristic roots of the original relation were close together and the approximation resulted in one double characteristic root. If characteristic roots of a relation are relatively far apart, this problem will not occur. For example, compare the general solutions of $S(k)+1.001S(k-1)-2.004002S(k-2)=0.0001$ and $S_a(k)+S_a(k-1)-2S_a(k-2)=0$.

This page titled 8.3: Recurrence Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

8.4: Some Common Recurrence Relations

In this section we intend to examine a variety of recurrence relations that are not finite-order linear with constant coefficients. For each part of this section, we will consider a concrete example, present a solution, and, if possible, examine a more general form of the original relation.

8.4.1: First Basic Example

Consider the homogeneous first-order linear relation without constant coefficients, $S(n)-nS(n-1)=0,\ \ n\geq 1$, with initial condition $S(0)=1$. Upon close examination of this relation, we see that the nth term is n times the $(n-1)^{st}$ term, which is a property of n factorial. $S(n)=n!$ is a solution of this relation, for if $n\geq 1$,

$$S(n)=n!=n\cdot(n-1)!=n\cdot S(n-1)$$

In addition, since $0!=1$, the initial condition is satisfied. It should be pointed out that from a computational point of view, our "solution" really isn't much of an improvement since the exact calculation of $n!$ takes $n-1$ multiplications.

If we examine a similar relation, $G(k)-2^kG(k-1),\ \ k\geq 1$ with $G(0)=1$, a table of values for G suggests a possible solution:

k	0	1	2	3	4	5
$G(k)$	1	2	2^3	2^6	2^{10}	2^{15}

The exponent of 2 in $G(k)$ is growing according to the relation $E(k)=E(k-1)+k,\ \ $ with $E(0)=0$. Thus $E(k)=\frac{k(k+1)}{2}$ and $G(k)=2^{k(k+1)/2}$. Note that $G(k)$ could also be written as $2^0 2^1 2^2\cdots 2^k$, for $k\geq 0$, but this is not a closed form expression.

In general, the relation $P(n)=f(n)P(n-1)$ for $n\geq 1$ with $P(0)=f(0)$, where f is a function that is defined for all $n\geq 0$, has the "solution"

$$P(n)=\prod_{k=0}^{n} f(k)$$

This product form of $P(n)$ is not a closed form expression because as n grows, the number of multiplications grow. Thus, it is really not a true solution. Often, as for $G(k)$ above, a closed form expression can be derived from the product form.

8.4.2: Analysis of the Binary Search Algorithm

8.4.2.1:

Suppose you intend to use a binary search algorithm (see Subsection 8.1.3) on lists of zero or more sorted items, and that the items are stored in an array, so that you have easy access to each item. A natural question to ask is "How much time will it take to complete the search?" When a question like this is asked, the time we refer to is often the so-called worst-case time. That is, if we were to search through n items, what is the longest amount of time that we will need to complete the search? In order to make an analysis such as this independent of the computer to be used, time is measured by counting the number of steps that are executed. Each step (or sequence of steps) is assigned an absolute time, or weight; therefore, our answer will not be in seconds, but in absolute time units. If the steps in two different algorithms are assigned weights that are consistent, then analyses of the algorithms can be used to compare their relative efficiencies. There are two major steps that must be executed in a call of the binary search algorithm:

1. If the lower index is less than or equal to the upper index, then the middle of the list is located and its key is compared to the value that you are searching for.
2. In the worst case, the algorithm must be executed with a list that is roughly half as large as in the previous execution. If we assume that Step 1 takes one time unit and $T(n)$ is the worst-case time for a list of n items, then

$$T(n)=1+T(\lfloor n/2\rfloor),\quad n>0 \tag{8.4.1}$$

For simplicity, we will assume that

$$T(0)=0 \tag{8.4.2}$$

even though the conditions of Step 1 must be evaluated as false if $n = 0$. You might wonder why $n/2$ is truncated in (8.4.1). If n is odd, then $n = 2k+1$ for some $k \geq 0$, the middle of the list will be the $(k+1)^{st}$ item, and no matter what half of the list the search is directed to, the reduced list will have $k = \lfloor n/2 \rfloor$ items. On the other hand, if n is even, then $n = 2k$ for $k > 0$. The middle of the list will be the kth item, and the worst case will occur if we are directed to the k items that come after the middle (the $(k+1)^{st}$ through $(2k)^{th}$ items). Again the reduced list has $\lfloor n/2 \rfloor$ items.

Solution to (8.4.1) *and* (8.4.2). To determine $T(n)$, the easiest case is when n is a power of two. If we compute $T\left(2^m\right)$, $m \geq 0$, by iteration, our results are

$$\begin{aligned} T(1) &= 1 + T(0) = 1 \\ T(2) &= 1 + T(1) = 2 \\ T(4) &= 1 + T(2) = 3 \\ T(8) &= 1 + T(4) = 4 \end{aligned}$$

The pattern that is established makes it clear that $T\left(2^m\right) = m + 1$. This result would seem to indicate that every time you double the size of your list, the search time increases by only one unit.

A more complete solution can be obtained if we represent n in binary form. For each $n \geq 1$, there exists a non-negative integer r such that

$$2^{r-1} \leq n < 2^r \tag{8.4.3}$$

For example, if $n = 21$, $2^4 \leq 21 < 2^5$; therefore, $r = 5$. If n satisfies (8.4c), its binary representation requires r digits. For example, $21_{\text{ten}} = 10101_{\text{two}}$.

In general, $n = (a_1 a_2 \ldots a_r)_{\text{two}}$. where $a_1 = 1$. Note that in this form, $\lfloor n/2 \rfloor$ is easy to describe: it is the $r-1$ digit binary number $(a_1 a_2 \ldots a_{r-1})_{\text{two}}$

Therefore,

$$\begin{aligned} T(n) &= T\left(a_1 a_2 \ldots a_r\right) \\ &= 1 + T\left(a_1 a_2 \ldots a_{r-1}\right) \\ &= 1 + \left(1 + T\left(a_1 a_2 \ldots a_{r-2}\right)\right) \\ &= 2 + T\left(a_1 a_2 \ldots a_{r-2}\right) \\ &\vdots \\ &= (r-1) + T\left(a_1\right) \\ &= (r-1) + 1 \quad \text{since } T(1) = 1 \\ &= r \end{aligned}$$

From the pattern that we've just established, $T(n)$ reduces to r. A formal inductive proof of this statement is possible. However, we expect that most readers would be satisfied with the argument above. Any skeptics are invited to provide the inductive proof.

For those who prefer to see a numeric example, suppose $n = 21$.

$$\begin{aligned} T(21) &= T(10101) \\ &= 1 + T(1010) \\ &= 1 + (1 + T(101)) \\ &= 1 + (1 + (1 + T(10))) \\ &= 1 + (1 + (1 + (1 + T(1)))) \\ &= 1 + (1 + (1 + (1 + (1 + T(0))))) \\ &= 5 \end{aligned}$$

Our general conclusion is that the solution to (8.4.1) and (8.4.2) is that for $n \geq 1$, $T(n) = r$, where $2^{r-1} \leq n < 2^r$.

A less cumbersome statement of this fact is that $T(n) = \lfloor \log_2 n \rfloor + 1$. For example, $T(21) = \lfloor \log_2 21 \rfloor + 1 = 4 + 1 = 5$.

8.4.2.2: Review of Logarithms

Any discussion of logarithms must start by establishing a base, which can be any positive number other than 1. With the exception of Theorem 8.4.2, our base will be 2. We will see that the use of a different base (10 and $e \approx 2.171828$ are the other common ones) only has the effect of multiplying each logarithm by a constant. Therefore, the base that you use really isn't very important. Our choice of base 2 logarithms is convenient for the problems that we are considering.

Definition 8.4.1: Base 2 Logarithm

The base 2 logarithm of a positive number represents an exponent and is defined by the following equivalence for any positive real numbers a.

$$\log_2 a = x \quad \Leftrightarrow \quad 2^x = a.$$

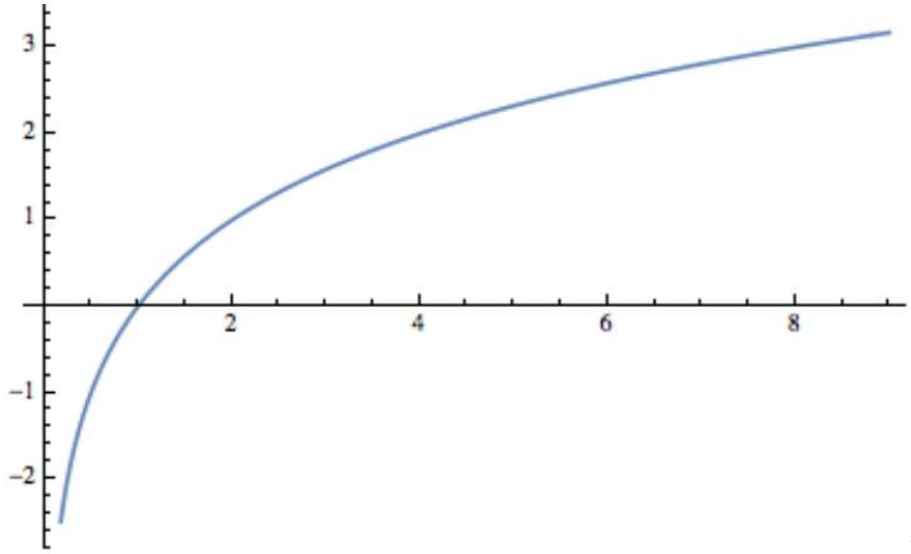


Figure 8.4.1: Plot of the logarithm, bases 2, function

For example, $\log_2 8 = 3$ because $2^3 = 8$ and $\log_2 1.414 \approx 0.5$ because $2^{0.5} \approx 1.414$. A graph of the function $f(x) = \log_2 x$ in Figure 8.4.1 shows that if $a < b$, the $\log_2 a < \log_2 b$; that is, when x increases, $\log_2 x$ also increases. However, if we move x from $2^{10} = 1024$ to $2^{11} = 2048$, $\log_2 x$ only increases from 10 to 11. This slow rate of increase of the logarithm function is an important point to remember. An algorithm acting on n pieces of data that can be executed in $\log_2 n$ time units can handle significantly larger sets of data than an algorithm that can be executed in $n/100$ or $\sqrt{n}$ time units. The graph of $T(n) = \lfloor \log_2 n \rfloor + 1$ would show the same behavior.

A few more properties that we will use in subsequent discussions involving logarithms are summarized in the following theorem.

Theorem 8.4.1: Fundamental Properties of Logarithms

Let a and b be positive real numbers, and r a real number.

$$\log_2 1 = 0 \tag{8.4.4}$$
$$\log_2 ab = \log_2 a + \log_2 b \tag{8.4.5}$$
$$\log_2 \frac{a}{b} = \log_2 a - \log_2 b \tag{8.4.6}$$
$$\log_2 a^r = r \log_2 a \tag{8.4.7}$$
$$2^{\log_2 a} = a \tag{8.4.8}$$

Definition 8.4.2: Logarithms Base b

If $b > 0$, $b \neq 1$, then for $a > 0$,

$$\log_b a = x \Leftrightarrow b^x = a$$

Theorem 8.4.2: How Logarithms with Different Bases are Related

Let $b > 0$, $b \neq 1$. Then for all $a > 0$, $\log_b a = \frac{\log_2 a}{\log_2 b}$. Therefore, if $b > 1$, base b logarithms can be computed from base 2 logarithms by dividing by the positive scaling factor $\log_2 b$. If $b < 1$, this scaling factor is negative.

Proof

By an analogue of (8.4.8), $a = b^{\log_b a}$. Therefore, if we take the base 2 logarithm of both sides of this equality we get:

$$\log_2 a = \log_2 \left(b^{\log_b a}\right) \Rightarrow \log_2 a = \log_b a \cdot \log_2 b$$

Finally, divide both sides of the last equation by $\log_2 b$.

Note 8.4.1

$\log_2 10 \approx 3.32192$ and $\log_2 e \approx 1.4427$.

8.4.2.3:

Returning to the binary search algorithm, we can derive the final expression for $T(n)$ using the properties of logarithms, including that the logarithm function is increasing so that inequalities are maintained when taking logarithms of numbers.

$$\begin{aligned} T(n) = r &\Leftrightarrow 2^{r-1} \leq n < 2^r \\ &\Leftrightarrow \log_2 2^{r-1} \leq \log_2 n < \log_2 2^r \\ &\Leftrightarrow r - 1 \leq \log_2 n < r \\ &\Leftrightarrow r - 1 = \lfloor \log_2 n \rfloor \\ &\Leftrightarrow T(n) = r = \lfloor \log_2 n \rfloor + 1 \end{aligned}$$

We can apply several of these properties of logarithms to get an alternate expression for $T(n)$:

$$\begin{aligned} \lfloor \log_2 n \rfloor + 1 &= \lfloor \log_2 n + 1 \rfloor \\ &= \lfloor \log_2 n + \log_2 2 \rfloor \\ &= \lfloor \log_2 2n \rfloor \end{aligned}$$

If the time that was assigned to Step 1 of the binary search algorithm is changed, we wouldn't expect the form of the solution to be very different. If $T(n) = a + T(\lfloor n/2 \rfloor)$ with $T(0) = c$, then $T(n) = c + a \lfloor \log_2 2n \rfloor$.

A further generalization would be to add a coefficient to $T(\lfloor n/2 \rfloor)$: $T(n) = a + bT(\lfloor n/2 \rfloor)$ with $T(0) = c$, where $a, b, c \in \mathbb{R}$, and $b \neq 0$ is not quite as simple to derive. First, if we consider values of n that are powers of 2:

$$\begin{aligned} T(1) &= a + bT(0) = a + bc \\ T(2) &= a + b(a + bc) = a + ab + cb^2 \\ T(4) &= a + b\left(a + ab + cb^2\right) = a + ab + ab^2 + cb^3 \\ &\vdots \\ T\left(2^r\right) &= a + ab + ab^2 + \cdots + ab^r + cb^{r+1} \end{aligned}$$

If n is not a power of 2, by reasoning that is identical to what we used to (8.4.1) and (8.4.2),

$$T(n) = \sum_{k=0}^{r} ab^k + cb^{r+1}$$

where $r = \lfloor \log_2 n \rfloor$.

The first term of this expression is a geometric sum, which can be written in closed form. Let x be that sum:

$$x = a + ab + ab^2 + \cdots + ab^r$$
$$bx = \ ab + ab^2 + \cdots + ab^r + ab^{r+1}$$

We've multiplied each term of x by b and aligned the identical terms in x and bx. Now if we subtract the two equations,

$$x - bx = a - ab^{r+1} \Rightarrow x(1-b) = a\left(1 - b^{r+1}\right)$$

Therefore, $x = a\frac{b^{r+1}-1}{b-1}$.

A closed form expression for $T(n)$ is

$$T(n) = a\frac{b^{r+1}-1}{b-1} + \ cb^{r+1} \text{ where } r = \lfloor \log_2 n \rfloor$$

8.4.3: Analysis of Bubble Sort and Merge Sort

The efficiency of any search algorithm such as the binary search relies on fact that the search list is sorted according to a key value and that the search is based on the key value. There are several methods for sorting a list. One example is the bubble sort. You might be familiar with this one since it is a popular "first sorting algorithm." A time analysis of the algorithm shows that if $B(n)$ is the worst-case time needed to complete the bubble sort on n items, then $B(n) = (n-1) + B(n-1)$ and $B(1) = 0$. The solution of this relation is a quadratic function $B(n) = \frac{1}{2}\left(n^2 - n\right)$. The growth rate of a quadratic function such as this one is controlled by its squared term. Any other terms are dwarfed by it as n gets large. For the bubble sort, this means that if we double the size of the list that we are to sort, n changes to $2n$ and so n^2 becomes $4n^2$. Therefore, the time needed to do a bubble sort is quadrupled. One alternative to bubble sort is the merge sort. Here is a simple version of this algorithm for sorting $F = \{r(1), r(2), \ldots, r(n)\}$, $n \geq 1$. If $n = 1$, the list is sorted trivially. If $n \geq 2$ then:

1. Divide F into $F_1 = \{r(1), \ldots, r(\lfloor n/2 \rfloor)\}$ and $F_2 = \{r(\lfloor n/2 \rfloor + 1), \ldots, r(n)\}$.
2. Sort F_1 and F_2 using a merge sort.
3. Merge the sorted lists F_1 and F_2 into one sorted list. If the sort is to be done in descending order of key values, you continue to choose the higher key value from the fronts of F_1 and F_2 and place them in the back of F.

Note that F_1 will always have $\lfloor n/2 \rfloor$ items and F_2 will have $\lceil n/2 \rceil$ items; thus, if n is odd, F_2 gets one more item than F_1. We will assume that the time required to perform Step 1 of the algorithm is insignificant compared to the other steps; therefore, we will assign a time value of zero to this step. Step 3 requires roughly n comparisons and n movements of items from F_1 and F_2 to F; thus, its time is proportional to n. For this reason, we will assume that Step 3 takes n time units. Since Step 2 requires $T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)$ time units,

$$T(n) = n + T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) \tag{8.4.9}$$

with the initial condition

$$T(1) = 0 \tag{8.4.10}$$

Instead of an exact solution of these equations, we will be content with an estimate for $T(n)$. First, consider the case of $n = 2^r$, $r \geq 1$:

$$T\left(2^1\right) = T(2) = 2 + T(1) + T(1) = 2 = 1 \cdot 2$$
$$T\left(2^2\right) = T(4) = 4 + \ T(2) + T(2) = 8 = 2 \cdot 4$$
$$T\left(2^3\right) = T(8) = 8 + T(4) + T(4) = 24 = 3 \cdot 8$$
$$\vdots$$
$$T\left(2^r\right) = r2^r = 2^r \log_2 2^r$$

Thus, if n is a power of 2, $T(n) = n \log_2 n$. Now if, for some $r \geq 2$, $2^{r-1} \leq n \leq 2^r$, then $(r-1)2^{r-1} \leq T(n) < r2^r$. This can be proved by induction on r. As n increases from 2^{r-1} to 2^r, $T(n)$ increases from $(r-1)2^{r-1}$ to $r2^r$ and is slightly larger than $\lfloor n \log_2 n \rfloor$. The discrepancy is small enough so that $T_e(n) = \lfloor n \log_2 n \rfloor$ can be considered a solution of (8.4.9) and (8.4.10) for the purposes of comparing the merge sort with other algorithms. Table 8.4.1 compares $B(n)$ with $T_e(n)$ for selected values of n.

Table 8.4.1: Comparison of Times for Bubble Sort and Merge Sort

n	$B(n)$	$T_e(n)$
10	45	34
50	1225	283
100	4950	665
500	124750	4483
1000	499500	9966

8.4.4: Derangements

A derangement is a permutation on a set that has no "fixed points". Here is a formal definition:

Definition 8.4.3: Derangement

A derangement of a nonempty set A is a permutation of A (i.e., a bijection from A into A) such that $f(a) \neq a$ for all $a \in A$.

If $A = \{1, 2, \ldots, n\}$, an interesting question might be "How many derangements are there of A?" We know that our answer is bounded above by $n!$. We can also expect our answer to be quite a bit smaller than $n!$ since n is the image of itself for $(n-1)!$ of the permutations of A.

Let $D(n)$ be the number of derangements of $\{1, 2, \ldots, n\}$. Our answer will come from discovering a recurrence relation on D. Suppose that $n \geq 3$. If we are to construct a derangement of $\{1, 2, \ldots, n\}, f$, then $f(n) = k \neq n$. Thus, the image of n can be selected in $n-1$ different ways. No matter which of the $n-1$ choices we make, we can complete the definition of f in one of two ways. First, we can decide to make $f(k) = n$, leaving $D(n-2)$ ways of completing the definition of f, since f will be a derangement of $\{1, 2, \ldots, n\} - \{n, k\}$. Second, if we decide to select $f(k) \neq n$, each of the $D(n-1)$ derangements of $\{1, 2, \ldots, n-1\}$ can be used to define f. If g is a derangement of $\{1, 2, \ldots, n-1\}$ such that $g(p) = k$, then define f by

$$f(j) = \begin{cases} n & \text{if } j = p \\ k & \text{if } j = n \\ g(j) & \text{otherwise} \end{cases}$$

Note that with our second construction of f, $f(f(n)) = f(k) \neq n$, while in the first construction, $f(f(n)) = f(k) = n$. Therefore, no derangement of $\{1, 2, \ldots, n\}$with $f(n) = k$ can be constructed by both methods.

To recap our result, we see that f is determined by first choosing one of $n-1$ images of n and then constructing the remainder of f in one of $D(n-2) + D(n-1)$ ways. Therefore,

$$D(n) = (n-1)(D(n-2) + D(n-1)) \tag{8.4.11}$$

This homogeneous second-order linear relation with variable coefficients, together with the initial conditions $D(1) = 0$ and $D(2) = 1$, completely defines D. Instead of deriving a solution of this relation by analytical methods, we will give an empirical derivation of an approximation of $D(n)$. Since the derangements of $\{1, 2 \ldots, n\}$ are drawn from a pool of $n!$ permutations, we will see what percentage of these permutations are derangements by listing the values of $n!$, $D(n)$, and $\frac{D(n)}{n!}$. The results we observe will indicate that as n grows, $\frac{D(n)}{n!}$ hardly changes at all. If this quotient is computed to eight decimal places, for $n \geq 12$, $D(n)/n! = 0.36787944$. The reciprocal of this number, which $D(n)/n!$ seems to be tending toward, is, to eight places, 2.7182818. This number appears in so many places in mathematics that it has its own name, e. An approximate solution of our recurrence relation on D is then $D(n) \approx \frac{n!}{e}$.

```
def D(n):
    if n<=2:
        return n-1
    else:
        return (n-1)*(D(n-2)+D(n-1))

list(map(lambda k:[k,D(k),(D(k)/factorial(k)).n(digits=8)],range(1,16)))
```

8.4.5: Exercises

Exercise 8.4.1

Solve the following recurrence relations. Indicate whether your solution is an improvement over iteration.

a. $nS(n)-S(n-1)=0,\ \ S(0)=1.$
b. $T(k)+3kT(k-1)=0,\ \ T(0)=1.$
c. $U(k)-\frac{k-1}{k}U(k-1)=0,\ \ k\geq 2,\ U(1)=1.$

Answer

a. $S(n)=1/n!$
b. $U(k)=1/k,$ an improvement.
c. $T(k)=(-3)^k k!,$ no improvement.

Exercise 8.4.2

Prove that if $n\geq 0,\ \lfloor n/2\rfloor+\lceil n/2\rceil=n.$ (Hint: Consider the cases of n odd and n even separately.)

Exercise 8.4.3

Solve as completely as possible:

a. $T(n)=3+T(\lfloor n/2\rfloor),\ T(0)=0.$
b. $T(n)=1+\frac{1}{2}T(\lfloor n/2\rfloor),\ T(0)=2.$
c. $V(n)=1+V\lfloor n/8\rfloor),\ V(0)=0.$ (Hint: Write n in octal form.)

Answer

a. $T(n)=3\left(\lfloor\log_2 n\rfloor+1\right)$
b. $T(n)=2$
c. $V(n)=\lfloor\log_8 n\rfloor+1$

Exercise 8.4.4

Prove by induction that if $T(n)=1+T(\lfloor n/2\rfloor),\ T(0)=0,$ and $2^{r-1}\leq n<2^r\ \ , r\geq 1,$ then $T(n)=r.$

Hint

Prove by induction on r.

Exercise 8.4.5

Use the substitution $S(n)=T(n+1)/T(n)$ to solve $T(n)T(n-2)=T(n-1)^2$ for $n\geq 2,$ with $T(0)=1,\ T(1)=6,$ and $T(n)\geq 0.$

Answer

The indicated substitution yields $S(n)=S(n+1).$ Since $S(0)=T(1)/T(0)=6,\ \ S(n)=6$ for all $n.$ Therefore $T(n+1)=6T(n)\Rightarrow T(n)=6^n.$

Exercise 8.4.6

Use the substitution $G(n)=T(n)^2$ to solve $T(n)^2-T(n-1)^2=1$ for $n\geq 1,$ with $T(0)=10.$

Exercise 8.4.7

Solve as completely as possible:

a. $Q(n) = 1 + Q\left(\lfloor \sqrt{n} \rfloor\right),\ n \geq 2,\ Q(1) = 0.$
b. $R(n) = n + R(\lfloor n/2 \rfloor),\ n \geq 1,\ R(0) = 0.$

Answer

a. A good approximation to the solution of this recurrence relation is based on the following observation: n is a power of a power of two; that is, n is 2^m, where $m = 2^k$, then $Q(n) = 1 + Q\left(2^{m/2}\right)$. By applying this recurrence relation k times we obtain $Q(n) = k$. Going back to the original form of n, $\log_2 n = 2^k$ or $\log_2(\log_2 n) = k$. We would expect that in general, $Q(n)$ is $\lfloor \log_2(\log_2 n) \rfloor$. We do not see any elementary method for arriving at an exact solution.
b. Suppose that n is a positive integer with $2^{k-1} \leq n < 2^k$. Then n can be written in binary form, $(a_{k-1}a_{k-2} \cdots a_2 a_1 a_0)_{\text{two}}$ with $a_{k-1} = 1$ and $R(n)$ is equal to the sum $\sum_{i=0}^{k-1} (a_{k-1}a_{k-2} \cdots a_i)_{\text{two}}$. If $2^{k-1} \leq n < 2^k$, then we can estimate this sum to be between $2n - 1$ and $2n + 1$. Therefore, $R(n) \approx 2n$.

Exercise 8.4.8

Suppose Step 1 of the merge sort algorithm did take a significant amount of time. Assume it takes 0.1 time unit, independent of the value of n.

a. Write out a new recurrence relation for $T(n)$ that takes this factor into account.
b. Solve for $T\left(2^r\right),\ r \geq 0$.
c. Assuming the solution for powers of 2 is a good estimate for all n, compare your result to the solution in the text. As gets large, is there really much difference?

This page titled 8.4: Some Common Recurrence Relations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

8.5: Generating Functions

This section contains an introduction to the topic of generating functions and how they are used to solve recurrence relations, among other problems. Methods that employ generating functions are based on the concept that you can take a problem involving sequences and translate it into a problem involving generating functions. Once you've solved the new problem, a translation back to sequences gives you a solution of the original problem.

This section covers:

1. The definition of a generating function.
2. Solution of a recurrence relation using generating functions to identify the skills needed to use generating functions.
3. An introduction and/or review of the skills identified in point 2.
4. Some applications of generating functions.

8.5.1: Definition

Definition 8.5.1: Generating Function of a Sequence

The generating function of a sequence S with terms $S_0, S_1, S_2, \ldots$, is the infinite sum

$$G(S;z) = \sum_{n=0}^{\infty} S_n z^n = S_0 + S_1 z + S_2 z^2 + S_3 z^3 + \cdots$$

The domain and codomain of generating functions will not be of any concern to us since we will only be performing algebraic operations on them.

Example 8.5.1: First Examples

a. If $S_n = 3^n, n \geq 0$, then

$$\begin{aligned} G(S;z) &= 1 + 3z + 9z^2 + 27z^3 + \cdots \\ &= \sum_{n=0}^{\infty} 3^n z^n \\ &= \sum_{n=0}^{\infty} (3z)^n \end{aligned}$$

We can get a closed form expression for $G(S;z)$ by observing that $G(S;z) - 3zG(S;z) = 1$. Therefore, $G(S;z) = \frac{1}{1-3z}$.

b. Finite sequences have generating functions. For example, the sequence of binomial coefficients $\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}, n \geq 1$ has generating function

$$\begin{aligned} G(\binom{n}{\cdot};z) &= \binom{n}{0} + \binom{n}{1} z + \cdots + \binom{n}{n} z^n \\ &= \sum_{k=0}^{\infty} \binom{n}{k} z^k \\ &= (1+z)^n \end{aligned}$$

by application of the binomial formula.

c. If $Q(n) = n^2$, $G(Q;z) = \sum_{n=0}^{\infty} n^2 z^n = \sum_{k=0}^{\infty} k^2 z^k$. Note that the index that is used in the summation has no significance. Also, note that the lower limit of the summation could start at 1 since $Q(0) = 0$.

8.5.2: Solution of a Recurrence Relation Using Generating Functions

We illustrate the use of generating functions by solving $S(n) - 2S(n-1) - 3S(n-2) = 0, \quad n \geq 2$, with $S(0) = 3$ and $S(1) = 1$.

1. Translate the recurrence relation into an equation about generating functions.
   Let $V(n) = S(n) - 2S(n-1) - 3S(n-2), \quad n \geq 2$, with $V(0) = 0$ and $V(1) = 0$. Therefore,

$$G(V; z) = 0 + 0z + \sum_{n=2}^{\infty} (S(n) - 2S(n-1) - 3S(n-2))z^n = 0$$

2. Solve for the generating function of the unknown sequence, $G(S; z) = \sum_{n=0}^{\infty} S_n z^n$.

$$\begin{aligned} 0 &= \sum_{n=2}^{\infty} (S(n) - 2S(n-1) - 3S(n-2))z^n \\ &= \sum_{n=2}^{\infty} S(n)z^n - 2\left(\sum_{n=2}^{\infty} S(n-1)z^n\right) - 3\left(\sum_{n=2}^{\infty} S(n-2)z^n\right) \end{aligned}$$

   Close examination of the three sums above shows:

   a.
$$\begin{aligned} \sum_{n=2}^{\infty} S_n z^n &= \sum_{n=0}^{\infty} S_n z^n - S(0) - S(1)z \\ &= G(S; z) - 3 - z \end{aligned}$$

   since $S(0) = 3$ and $S(1) = 1$.

   b.
$$\begin{aligned} \sum_{n=2}^{\infty} S(n-1)z^n &= z\left(\sum_{n=2}^{\infty} S(n-1)z^{n-1}\right) \\ &= z\left(\sum_{n=1}^{\infty} S(n)z^n\right) \\ &= z\left(\sum_{n=0}^{\infty} S(n)z^n - S(0)\right) \\ &= z(G(S; z) - 3) \end{aligned}$$

   c.
$$\begin{aligned} \sum_{n=2}^{\infty} S(n-2)z^n &= z^2\left(\sum_{n=2}^{\infty} S(n-2)z^{n-2}\right) \\ &= z^2 G(S; z) \end{aligned}$$

   Therefore,

$$\begin{aligned} &(G(S; z) - 3 - z) - 2z(G(S; z) - 3) - 3z^2 G(S; z) = 0 \\ &\Rightarrow G(S; z) - 2zG(S; z) - 3z^2 G(S; z) = 3 - 5z \\ &\Rightarrow G(S; z) = \frac{3 - 5z}{1 - 2z - 3z^2} \end{aligned}$$

3. Determine the sequence whose generating function is the one we got in Step 2.

   For our example, we need to know one general fact about the closed form expression of an exponential sequence (a proof will be given later):

$$T(n) = ba^n, \; n \geq 0 \Leftrightarrow G(T; z) = \frac{b}{1 - az} \tag{8.5.1}$$

   Now, in order to recognize S in our example, we must write our closed form expression for $G(S; z)$ as a sum of terms like $G(T; z)$ above. Note that the denominator of $G(S; z)$ can be factored:

$$G(S;z) = \frac{3-5z}{1-2z-3z^2} = \frac{3-5z}{(1-3z)(1+z)}$$

If you look at this last expression for $G(S;z)$ closely, you can imagine how it could be the result of addition of two fractions,

$$\frac{3-5z}{(1-3z)(1+z)} = \frac{A}{1-3z} + \frac{B}{1+z} \tag{8.5.2}$$

where A and B are two real numbers that must be determined. Starting on the right of (8.5.2), it should be clear that the sum, for any A and B, would look like the left-hand side. The process of finding values of A and B that make (8.5.2) true is called the ***partial fractions decomposition*** of the left-hand side:

$$\begin{aligned}\frac{A}{1-3z} + \frac{B}{1+z} &= \frac{A(1+z)}{(1-3z)(1+z)} + \frac{B(1-3z)}{(1-3z)(1+z)} \\ &= \frac{(A+B)+(A-3B)z}{(1-3z)(1+z)}\end{aligned}$$

Therefore,

$$\left\{ \begin{array}{c} A+B=3 \\ A-3B=-5 \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} A=1 \\ B=2 \end{array} \right\}$$

and

$$G(S;z) = \frac{1}{1-3z} + \frac{2}{1+z}$$

We can apply (8.5.1) to each term of $G(S;z)$:

- $\frac{1}{1-3z}$ is the generating function for $S_1(n) = 1 \cdot 3^n = 3^n$
- $\frac{2}{1+z}$ is the generating function for $S_2(n) = 2(-1)^n$.

Therefore, $S(n) = 3^n + 2(-1)^n$.

From this example, we see that there are several skills that must be mastered in order to work with generating functions. You must be able to:

a. Manipulate summation expressions and their indices (in Step 2).
b. Solve algebraic equations and manipulate algebraic expressions, including partial function decompositions (Steps 2 and 3).
c. Identify sequences with their generating functions (Steps 1 and 3).

We will concentrate on the last skill first, a proficiency in the other skills is a product of doing as many exercises and reading as many examples as possible.

First, we will identify the operations on sequences and on generating functions.

8.5.3: Operations on Sequences

Definition 8.5.2: Operations on Sequences

Let S and T be sequences of numbers and let c be a real number. Define the sum $S+T$, the scalar product cS, the product ST, the convolution $S*T$, the pop operation $S\uparrow$ (read "S pop"), and the push operation $S\downarrow$ (read "S push") term-wise for $k \geq 0$ by

$$(S+T)(k) = S(k)+T(k) \tag{8.5.3}$$

$$(cS)(k) = cS(k) \tag{8.5.4}$$

$$(S \cdot T)(k) = S(k)T(k) \tag{8.5.5}$$

$$(S*T)(k) = \sum_{j=0}^{k} S(j)T(k-j) \tag{8.5.6}$$

$$(S\uparrow)(k) = S(k+1) \tag{8.5.7}$$

$$(S\downarrow)(k)=\begin{cases}0 & \text{if } k=0\\ S(k-1) & \text{if } k>0\end{cases} \tag{8.5.8}$$

If one imagines a sequence to be a matrix with one row and an infinite number of columns, $S+T$ and cS are exactly as in matrix addition and scalar multiplication. There is no obvious similarity between the other operations and matrix operations.

The pop and push operations can be understood by imagining a sequence to be an infinite stack of numbers with $S(0)$ at the top, $S(1)$ next, etc., as in Figure 8.5.1a. The sequence $S\uparrow$ is obtained by "popping" S(0) from the stack, leaving a stack as in Figure 8.5.1b, with S(1) at the top, S(2) next, etc. The sequence $S\downarrow$ is obtained by placing a zero at the top of the stack, resulting in a stack as in Figure 8.5.1c. Keep these figures in mind when we discuss the pop and push operations.

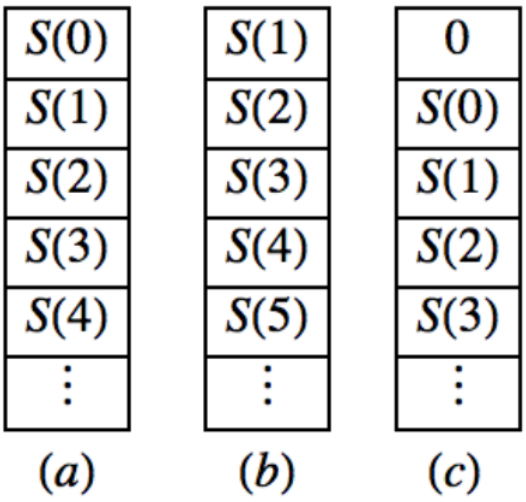


Figure 8.5.1: Stack interpretation of pop and push orientation

Example 8.5.2: Some Sequence Operations

If $S(n)=n$, $T(n)=n^2$, $U(n)=2^n$, and $R(n)=n2^n$:

a. $(S+T)(n)=n+n^2$
b. $(U+R)(n)=2^n+n2^n=(1+n)2^n$
c. $(2U)(n)=2\cdot 2^n=2^{n+1}$
d. $\left(\frac{1}{2}R\right)(n)=\frac{1}{2}n2^n=n2^{n-1}$
e. $(S\cdot T)(n)=nn^2=n^3$
f. $$\begin{aligned}(S*T)(n)&=\sum_{j=0}^{n}S(j)T(n-j)=\sum_{j=0}^{n}j(n-j)^2\\ &=\textstyle\sum_{j=0}^{n}\left(jn^2-2nj^2+j^3\right)\\ &=n^2\textstyle\sum_{j=0}^{n}j-2n\sum_{j=0}^{n}j^2+\sum_{j=0}^{n}j^3\\ &=n^2\left(\frac{n(n+1)}{2}\right)-2n\left(\frac{(2n+1)(n+1)n}{6}\right)+\frac{1}{4}n^2(n+1)^2\\ &=\frac{n^2(n+1)(n-1)}{12}\end{aligned}$$
g. $$\begin{aligned}(U*U)(n)&=\sum_{j=0}^{n}U(j)U(n-j)\\ &=\textstyle\sum_{j=0}^{n}2^j2^{n-j}\\ &=(n+1)2^n\end{aligned}$$
h. $(S\uparrow)(n)=n+1$
i. $(S\downarrow)(n)=\max(0,n-1)$
j. $((S\downarrow)\downarrow)(n)=\max(0,n-2)$

k. $(U\downarrow)(n)=\begin{cases}2^{n-1} & \text{if } n>0\\ 0 & \text{if } n=0\end{cases}$

l. $((U\downarrow)\uparrow)(n)=(U\downarrow)(n+1)=2^n=U(n)$

m. $((U\uparrow)\downarrow)(n)=\begin{cases}0 & \text{if } n=0\\ U(n) & \text{if } n>0\end{cases}$

Note that $(U\downarrow)\uparrow\neq(U\uparrow)\downarrow$.

Definition 8.5.3: Multiple Pop and Push

If S is a sequence of numbers and p a positive integer greater than 1, define

$$S\uparrow p=(S\uparrow(p-1))\uparrow \quad \text{if } p\geq 2 \text{ and } S\uparrow 1=S\uparrow$$

Similarly, define

$$S\downarrow p=(S\downarrow(p-1))\downarrow \quad \text{if } p\geq 2 \text{ and } S\downarrow 1=S\downarrow$$

In general, $(S\uparrow p)(k)=S(k+p)$, and

$$(S\downarrow p)(k)=\begin{cases}0 & \text{if } k<p\\ S(k-p) & \text{if } k\geq p\end{cases}$$

8.5.4: Operations on Generating Functions

Definition 8.5.4: Operations on Generating Functions

If $G(z)=\sum_{k=0}^{\infty}a_kz^k$ and $H(z)=\sum_{k=0}^{\infty}b_kz^k$ are generating functions and c is a real number, then the sum $G+H$, scalar product cG, product GH, and monomial product z^pG, $p\geq 1$ are generating functions, where

$$(G+H)(z)=\sum_{k=0}^{\infty}(a_k+b_k)z^k \tag{8.5.9}$$

$$(cG)(z)=\sum_{k=0}^{\infty}ca_kz^k \tag{8.5.10}$$

$$(GH)(z)=\sum_{k=0}^{\infty}cz^k \text{ where } c_k=\sum_{j=0}^{k}a_jb_{k-j} \tag{8.5.11}$$

$$(z^pG)(z)=z^p\sum_{k=0}^{\infty}a_kz^k=\sum_{k=0}^{\infty}a_kz^{k+p}=\sum_{n=p}^{\infty}a_{n-p}z^n \tag{8.5.12}$$

The last sum is obtained by substituting $n-p$ for k in the previous sum.

Example 8.5.3: Some Operations on Generating Functions

If $D(z)=\sum_{k=0}^{\infty}kz^k$ and $H(z)=\sum_{k=0}^{\infty}2^kz^k$ then

$$(D+H)(z)=\sum_{k=0}^{\infty}\left(k+2^k\right)z^k$$

$$(2H)(z)=\sum_{k=0}^{\infty}2\cdot 2^kz^k=\sum_{k=0}^{\infty}2^{k+1}z^k$$

$$
\begin{aligned}
(zD)(z) &= z\sum_{k=0}^{\infty} kz^k = \sum_{k=0}^{\infty} kz^{k+1} \\
&= \sum_{k=1}^{\infty}(k-1)z^k = D(z) - \sum_{k=1}^{\infty} z^k
\end{aligned}
$$

$$
(DH)(z) = \sum_{k=0}^{\infty}\left(\sum_{j=0}^{k} j2^{k-j}\right) z^k
$$

$$
(HH)(z) = \sum_{k=0}^{\infty}\left(\sum_{j=0}^{k} 2^j 2^{k-j}\right) z^k = \sum_{k=0}^{\infty}(k+1)2^k z^k
$$

Note: $D(z) = G(S; z)$, and $H(z) = G(U; z)$ from Example 8.5.2.

Now we establish the connection between the operations on sequences and generating functions. Let S and T be sequences and let c be a real number.

$$G(S+T; z) = G(S; z) + G(T; z) \tag{8.5.13}$$

$$G(cS; z) = cG(S; z) \tag{8.5.14}$$

$$G(S * T; z) = G(S; z)G(T; z) \tag{8.5.15}$$

$$G(S\uparrow; z) = (G(S; z) - S(0))/z \tag{8.5.16}$$

$$G(S\downarrow; z) = zG(S; z) \tag{8.5.17}$$

In words, (8.5.13) says that the generating function of the sum of two sequences equals the sum of the generating functions of those sequences. Take the time to write out the other four identities in your own words. From the previous examples, these identities should be fairly obvious, with the possible exception of the last two. We will prove (8.5.16) as part of the next theorem and leave the proof of (8.5.17) to the interested reader. Note that there is no operation on generating functions that is related to sequence multiplication; that is, $G(S \cdot T; z)$ cannot be simplified.

Theorem 8.5.1: Generating Functions Related to Pop and Push

If $p > 1$,

a. $G(S\uparrow p; z) = \left(G(S; z) - \sum_{k=0}^{p-1} S(k)z^k\right)/z^k$

b. $G(S\downarrow p; z) = z^p G(S; z)$.

Proof

We prove (a) by induction and leave the proof of (b) to the reader.

Basis:

$$
\begin{aligned}
G(S\uparrow; z) &= \sum_{k=0}^{\infty} S(k+1)z^k \\
&= \sum_{k=1}^{\infty} S(k)z^{k-1} \\
&= \left(\sum_{k=1}^{\infty} S(k)z^k\right) \Big/ z \\
&= \left(S(0) + \sum_{k=1}^{\infty} S(k)z^k - S(0)\right) \Big/ z \\
&= (G(S; z) - S(0))/z
\end{aligned}
$$

Therefore, part (a) is true for $p = 1$.

Induction: Suppose that for some $p \geq 1$, the statement in part (a) is true:

$$\begin{aligned} G(S \uparrow (p+1); z) &= G((S \uparrow p) \uparrow; z) \\ &= (G(S \uparrow p; z) - (S \uparrow p)(0))/z \text{ by the basis} \\ &= \frac{\left(G(S;z) - \sum_{k=0}^{p-1} S(k) z^k\right)}{z^p} - S(p) \\ &\qquad z \end{aligned}$$

by the induction hypothesis. Now write $S(p)$ in the last expression above as $\left(S(p)z^p\right)/z^p$ so that it fits into the finite summation:

$$\begin{aligned} G(S \uparrow (p+1); z) &= \left(\frac{G(S;z) - \sum_{k=0}^{p} S(k) z^k}{z^p} \right) \Bigg/ z \\ &= \left(G(S;z) - \sum_{k=0}^{p} S(k) z^k \right) / z^{p+1} \end{aligned}$$

Therefore the statement is true for $p+1$.

8.5.5: Closed Form Expressions for Generating Functions

The most basic tool used to express generating functions in closed form is the closed form expression for the geometric series, which is an expression of the form $a + ar + ar^2 + \cdots$. It can either be terminated or extended infinitely.

Finite Geometric Series:

$$a + ar + ar^2 + \cdots + ar^n = a\left(\frac{1 - r^{n+1}}{1 - r}\right) \tag{8.5.18}$$

Infinite Geometric Series:

$$a + ar + ar^2 + \cdots = \frac{a}{1 - r} \tag{8.5.19}$$

Restrictions: a and r represent constants and the right sides of the two equations apply under the following conditions:

1. r must not equal 1 in the finite case. Note that $a + ar + \cdots ar^n = (n+1)a \quad \text{if } r = 1$.
2. In the infinite case, the absolute value of r must be less than 1.

These restrictions don't come into play with generating functions. We could derive (8.5.18) by noting that if $S(n) = a + ar + \cdots + ar^n, \quad n > 0$, then $S(n) = rS(n-1) + a$ (See Exercise 8.3.10 of Section 8.3). An alternative derivation was used in Section 8.4. We will take the same steps to derive (8.5.19). Let $x = a + ar + ar^2 + \cdots$. Then

$$rx = ar + ar^2 + \cdots = x - a \Rightarrow x - rx = a \Rightarrow x = \frac{a}{1 - r}$$

Example 8.5.4: Generating Functions Involving Geometric Sums

a. If $S(n) = 9 \cdot 5^n, \ n \geq 0$, $G(S;z)$ is an infinite geometric series with $a = 9$ and $r = 5z$. Therefore, $G(S;z) = \frac{9}{1-5z}$.
b. If $T(n) = 4, \ n \geq 0$, then $G(T;z) = 4/(1-z)$.
c. If $U(n) = 3(-1)^n$, then $G(U;z) = 3/(1+z)$.
d. Let $C(n) = S(n) + T(n) + U(n) = 9 \cdot 5^n + 4 + 3(-1)^n$. Then

$$\begin{aligned} G(C;z) &= G(S;z) + G(T;z) + G(U;z) \\ &= \frac{9}{1-5z} + \frac{4}{1-z} + \frac{3}{1+z} \\ &= -\frac{14z^2 + 34z - 16}{5z^3 - z^2 - 5z + 1} \end{aligned} \quad .$$

Given a choice between the last form of $G(C;z)$ and the previous sum of three fractions, we would prefer leaving it as a sum of three functions. As we saw in an earlier example, a partial fractions decomposition of a fraction such as the last expression requires some effort to produce.

e. If $G(Q;z) = 34/(2-3z)$, then Q can be determined by multiplying the numerator and denominator by 1/2 to obtain $\frac{17}{1-\frac{3}{2}z}$. We recognize this fraction as the sum of the infinite geometric series with $a = 17$ and $r = \frac{3}{2}z$. Therefore $Q(n) = 17(3/2)^n$.

f. If $G(A;z) = (1+z)^3$, then we expand $(1+z)^3$ to $1+3z+3z^2+z^3$. Therefore $A(0) = 1$, $A(1) = 3$ $A(2) = 3$, $A(3) = 1$, and, since there are no higher-powered terms, $A(n) = 0$, $n \geq 4$. A more concise way of describing A is $A(k) = \binom{3}{k}$, since $\binom{n}{k}$ is interpreted as 0 of $k > n$.

Table 8.5.1: Closed Form Expressions of some Generating Functions

Sequence	Generating Function
$S(k) = ba^k$	$G(S;z) = \frac{b}{1-az}$
$S(k) = k$	$G(S;z) = \frac{z}{(1-z)^2}$
$S(k) = bka^k$	$G(S;z) = \frac{abz}{(1-az)^2}$
$S(k) = \frac{1}{k!}$	$G(S;z) = e^z$
$S(k) = \begin{cases} \binom{n}{k} & 0 \leq k \leq n \\ 0 & k > n \end{cases}$	$G(S;z) = (1+z)^n$

Example 8.5.5: Another Complete Solution

Solve $S(k)+3S(k-1)-4S(k-2) = 0, \quad k \geq 2$, with $S(0) = 3$ and $S(1) = -2$. The solution will be derived using the same steps that were used earlier in this section, with one variation.

1. Translate to an equation about generating functions. First, we change the index of the recurrence relation by substituting $n+2$ for k. The result is $S(n+2)+3S(n+1)-4S(n) = 0, \quad n \geq 0$. Now, if $V(n) = S(n+2)+3S(n+1)-4S(n)$, then V is the zero sequence, which has a zero generating function. Furthermore, $V = S\uparrow 2+3(S\uparrow)-4S$. Therefore,

$$\begin{aligned} 0 &= G(V;z) \\ &= G(S\uparrow 2;z)+3G(S\uparrow;z)-4G(S;z) \\ &= \frac{G(S;z)-S(0)-S(1)z}{z^2}+4\frac{(G(S;z)-S(0))}{z}-4G(S;z) \end{aligned}$$

2. We want to now solve the following equation for $G(S;z)$:

$$\frac{G(S;z)-S(0)-S(1)z}{z^2}+4\frac{(G(S;z)-S(0))}{z}-4G(S;z) = 0$$

Multiply by z^2 :

$$G(S;z)-3+2z+3z(G(S;z)-3)-4z^2G(S;z) = 0$$

Expand and collect all terms involving $G(S;z)$ on one side of the equation:

$$\begin{aligned} G(S;z)+3zG(S;z)-4z^2G(S;z) &= 3+7z \\ \left(1+3z-4z^2\right)G(S;z) &= 3+7z \end{aligned}$$

Therefore,

$$G(S;z)=\frac{3+7z}{1+3z-4z^2}$$

3. Determine S from its generating function. $1+3z-4z^2=(1+4z)(1-z)$ thus a partial fraction decomposition of $G(S;z)$ would be:

$$\frac{A}{1+4z}+\frac{B}{1-z}=\frac{Az-A-4Bz-B}{(z-1)(4z+1)}=\frac{(A+B)+(4B-A)z}{(z-1)(4z+1)}$$

Therefore, $A+B=3$ and $4B-A=7$. The solution of this set of equations is $A=1$ and $B=2$.
$G(S;z)=\frac{1}{1+4z}+\frac{2}{1-z}$.

$\frac{1}{1+4z}$ is the generating function of $S_1(n)=(-4)^n$, and
$\frac{2}{1-z}$ is the generating function of $S_2(n)=2(1)^n=2$

In conclusion, since $G(S;z)=G(S_1;z)+G(S_2;z)$, $S(n)=2+(-4)^n$.

Example 8.5.6: An Application to Counting

Let $A=\{a,b,c,d,e\}$ and let A^* be the set of all strings of length zero or more that can be made using each of the elements of A zero or more times. By the generalized rule of products, there are 5^n such strings that have length n, $n\geq 0$, Suppose that X_n is the set of strings of length n with the property that all of the a's and b's precede all of the c's, d's, and e's. Thus $aaabde\in X_6$, but $abcabc\notin X_6$. Let $R(n)=|X_n|$. A closed form expression for R can be obtained by recognizing R as the convolution of two sequences. To illustrate our point, we will consider the calculation of $R(6)$.

Note that if a string belongs to X_6, it starts with k characters from $\{a,b\}$ and is followed by $6-k$ characters from $\{c,d,e\}$. Let $S(k)$ be the number of strings of a's and b's with length k and let $T(k)$ be the number of strings of c's, d's, and e's with length k. By the generalized rule of products, $S(k)=2^k$ and $T(k)=3^k$. Among the strings in X_6 are the ones that start with two a's and b's and end with c's, d's, and e's. There are $S(2)T(4)$ such strings. By the law of addition,

$$|X_6|=R(6)=S(0)T(6)+S(1)T(5)+\cdots+S(5)T(1)+S(6)T(0)$$

Note that the sixth term of R is the sixth term of the convolution of S with T, $S*T$. Think about the general situation for a while and it should be clear that $R=S*T$. Now, our course of action will be to:

a. Determine the generating functions of S and T,
b. Multiply $G(S;z)$ and $G(T;z)$ to obtain $G(S*T;z)=G(R;z)$, and
c. Determine R on the basis of $G(R;z)$.

a. $G(S;z)=\sum_{k=0}^{\infty}2^kz^k=\frac{1}{1-2z}$, and $G(T;z)=\sum_{k=0}^{\infty}3^kz^k=\frac{1}{1-3z}$
b. $G(R;z)=G(S;z)G(T;z)=\dfrac{1}{(1-2z)(1-3z)}$
c. To recognize R from $G(R;z)$, we must do a partial fractions decomposition:

$$\frac{1}{(1-2z)(1-3z)}=\frac{A}{1-2z}+\frac{B}{1-3z}=\frac{-3Az+A-2Bz+B}{(2z-1)(3z-1)}=\frac{(A+B)+(-3A-2B)z}{(2z-1)(3z-1)}$$

Therefore, $A+B=1$ and $-3A-2B=0$. The solution of this pair of equations is $A=-2$ and $B=3$. Since
$G(R;z)=\frac{-2}{1-2z}+\frac{3}{1-3z}$, which is the sum of the generating functions of $-2(2)^k$ and $3(3)^k$,
$R(k)=-2(2)^k+3(3)^k=3^{k+1}-2^{k+1}$
For example, $R(6)=3^7-2^7=2187-128=2059$. Naturally, this equals the sum that we get from $(S*T)(6)$. To put this number in perspective, the total number of strings of length 6 with no restrictions is $5^6=15625$, and
$\frac{2059}{15625}\approx 0.131776$. Therefore approximately 13 percent of the strings of length 6 satisfy the conditions of the problem.

8.5.6: Extra for Experts

The remainder of this section is intended for readers who have had, or who intend to take, a course in combinatorics. We do not advise that it be included in a typical course. The method that was used in the previous example is a very powerful one and can be used to solve many problems in combinatorics. We close this section with a general description of the problems that can be solved in this way, followed by some examples.

Consider the situation in which $P_1, P_2, \ldots, P_m$ are m actions that must be taken, each of which results in a well-defined outcome. For each $k = 1, 2, \ldots, m$ define X_k to be the set of possible outcomes of P_k . We will assume that each outcome can be quantified in some way and that the quantification of the elements of X_k is defined by the function $Q_k : X_k \to \{0, 1, 2, \ldots\}$. Thus, each outcome has a non-negative integer associated with it. Finally, define a frequency function $F_k : \{0, 1, 2, \ldots\} \to \{0, 1, 2, \ldots\}$such that $F_k(n)$ is the number of elements of X_k that have a quantification of n.

Now, based on these assumptions, we can define the problems that can be solved. If a process P is defined as a sequence of actions $P_1, P_2, \ldots, P_m$ as above, and if the outcome of P, which would be an element of $X_1 \times X_2 \times \cdots \times X_m$, is quantified by

$$Q(a_1, a_2, \ldots, a_m) = \sum_{k=1}^{m} Q_k(a_k)$$

then the frequency function, F, for P is the convolution of the frequency functions forP_1, P_2, $\ldots$, P_m, which has a generating function equal to the product of the generating functions of the frequency functions F_1, F_2, $\ldots$, F_m. That is,

$$G(F; z) = G(F_1; z)\, G(F_2; z) \cdots (F_m; z)$$

Example 8.5.7: Rolling Two Dice

Suppose that you roll a die two times and add up the numbers on the top face for each roll. Since the faces on the die represent the integers 1 through 6, the sum must be between 2 and 12. How many ways can any one of these sums be obtained? Obviously, 2 can be obtained only one way, with two 1's. There are two sequences that yield a sum of 3: 1-2 and 2-1. To obtain all of the frequencies with which the numbers 2 through 12 can be obtained, we set up the situation as follows. For $j = 1, 2$; P_j is the rolling of the die for the jth time. $X_j = \{1, 2, \ldots, 6\}$ and $Q_j : X_j \to \{0, 1, 2, 3, \ldots\}$ is defined by $Q_j(x) = x$. Since each number appears on a die exactly once, the frequency function is $F_j(k) = 1$ if $1 \le k \le 6$, and $F_j(k) = 0$ otherwise. The process of rolling the die two times is quantified by adding up the $Q_j's$; that is, $Q(a_1, a_2) = Q_1(a_1) + Q_2(a_2)$. The generating function for the frequency function of rolling the die two times is then

$$\begin{aligned} G(F; z) &= G(F_1; z)\, G(F_2; z) \\ &= (z^6 + z^5 + z^4 + z^3 + z^2 + z)^2 \\ &= z^{12} + 2z^{11} + 3z^{10} + 4z^9 + 5z^8 + 6z^7 + 5z^6 + 4z^5 + 3z^4 + 2z^3 + z^2 \end{aligned}$$

Now, to get $F(k)$, just read the coefficient of z^k. For example, the coefficient of z^5 is 4, so there are four ways to roll a total of 5.

To apply this method, the crucial step is to decompose a large process in the proper way so that it fits into the general situation that we've described.

Example 8.5.8: Distribution of a Committee

Suppose that an organization is divided into three geographic sections, A, B, and C. Suppose that an executive committee of 11 members must be selected so that no more than 5 members from any one section are on the committee and that Sections A, B, and C must have minimums of 3, 2, and 2 members, respectively, on the committee. Looking only at the number of members from each section on the committee, how many ways can the committee be made up? One example of a valid committee would be 4 A's, 4 B's, and 3 C's.

Let P_A be the action of deciding how many members (not who) from Section A will serve on the committee. $X_A = \{3, 4, 5\}$ and $Q_A(k) = k$. The frequency function, F_A , is defined by $F_A(k) = 1$ if $k \in X_k$, with $F_A(k) = 0$ otherwise. $G(F_A; z)$ is then $z^3 + z^4 + z^5$. Similarly, $G(F_B; z) = z^2 + z^3 + z^4 + z^5 = G(F_C; z)$. Since the committee must have 11 members, our answer will be the coefficient of z^{11} in $G(F_A; z)\, G(F_B; z)\, G(F_C; z)$, which is 10.

8.5.7: Exercises

Exercise 8.5.1

What sequences have the following generating functions?

a. 1
b. $\dfrac{10}{2-z}$
c. $1+z$
d. $\dfrac{3}{1+2z}+\dfrac{3}{1-3z}$

Answer

a. $1,0,0,0,0,\ldots$
b. $5(1/2)^k$
c. $1,1,0,0,0,\ldots$
d. $3(-2)^k+3\cdot 3^k$

Exercise 8.5.2

What sequences have the following generating functions?

a. $\dfrac{1}{1+z}$
b. $\dfrac{1}{4-3z}$
c. $\dfrac{2}{1-z}+\dfrac{1}{1+z}$
d. $\dfrac{z+2}{z+3}$

Exercise 8.5.3

Find closed form expressions for the generating functions of the following sequences:

a. $V(n)=9^n$
b. P, where $P(k)-6P(k-1)+5P(k-2)=0 \quad$ for $k\geq 2$, with $P(0)=2$ and $P(1)=2$.
c. The Fibonacci sequence: $F(k+2)=F(k+1)+F(k), \quad k\geq 0$, with $F(0)=F(1)=1$.

Answer

a. $1/(1-9z)$
b. $(2-10z)\,/\,(1-6z+5z^2)$
c. $1\,/\,(1-z-z^2)$

Exercise 8.5.4

Find closed form expressions for the generating functions of the following sequences:

a. $W(n)=\binom{5}{n}2^n$ for $0\leq n\leq 5$ and $W(n)=0$ for $n>5$.
b. Q, where $Q(k)+Q(k-1)-42Q(k-2)=0 \quad$ for $k\geq 2$, with $Q(0)=2$ and $Q(1)=2$.
c. G, where $G(k+3)=G(k+2)+G(k+1)+G(k) \quad$ for $k\geq 0$, with $G(0)=G(1)=G(2)=1$.

Exercise 8.5.5

For each of the following expressions, find the partial fraction decomposition and identify the sequence having the expression as a generating function.

a. $\dfrac{5+2z}{1-4z^2}$
b. $\dfrac{32-22z}{2-3z+z^2}$
c. $\dfrac{6-29z}{1-11z+30z^2}$

Answer

a. $3/(1-2z)+2/(1+2z), 3\cdot 2^k+2(-2)^k$
b. $10/(1-z)+12/(2-z), 10+6(1/2)^k$
c. $-1/(1-5z)+7/(1-6z), 7\cdot 6^k-5^k$

Exercise 8.5.6

Find the partial fraction decompositions and identify the sequence having the following expressions:

a. $\dfrac{1}{1-9z^2}$
b. $\dfrac{1+3z}{16-8z+z^2}$
c. $\dfrac{2z}{1-6z-7z^2}$

Exercise 8.5.7

Given that $S(k)=k$ and $T(k)=10k$, what is the kth term of the generating function of each of the following sequences:

a. $S+T$
b. $S\uparrow *T$
c. $S*T$
d. $S\uparrow *S\uparrow$

Answer

a. $11k$
b. $(5/3)k(k+1)(2k+1)+5k(k+1)$
c. $\sum_{j=0}^{k}(j)(10(k-j))=10k\sum_{j=0}^{k}j-10\sum_{j=0}^{k}j^2 \quad =5k^2(k+1)-(5k(k+1)(2k+1)/6)=(5/3)k(k+1)(2k+1)$
d. $k(k+1)(2k+7)/12$

Exercise 8.5.8

Given that $P(k)=\binom{10}{k}$ and $Q(k)=k!$, what is the kth term of the generating function of each of the following sequences:

a. $P*P$
b. $P+P\uparrow$
c. $P*Q$
d. $Q*Q$

Exercise 8.5.9

A game is played by rolling a die five times. For the kth roll, one point is added to your score if you roll a number higher than k. Otherwise, your score is zero for that roll. For example, the sequence of rolls $2, 3, 4, 1, 2$ gives you a total score of three; while a sequence of 1,2,3,4,5 gives you a score of zero. Of the $6^5 = 7776$ possible sequences of rolls, how many give you a score of zero?, of one? . . . of five?

Answer

Coefficients of z^0 through z^5 in $(1+5z)(2+4z)(3+3z)(4+2z)(5+z)$

k	Number of ways of getting a score of k
0	120
1	1044
2	2724
3	2724
4	1044
5	120

Exercise 8.5.10

Suppose that you roll a die ten times in a row and record the square of each number that you roll. How many ways could the sum of the squares of your rolls equal 40? What is the most common outcome?

This page titled 8.5: Generating Functions is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

9: Graph Theory

> ***Bipartite***
>
> *Draw some lines joining dots in set A*
> *To some dots in set B. Then we say*
> *It's **bipartite** if we*
> *Have no "B" joined to "B"*
> *And no "A" joined to "A". That okay?*
>
> *Chris Howlett, The Omnificent English Dictionary in Limerick Form*

This chapter has three principal goals. First, we will identify the basic components of a graph and some of the features that many graphs have. Second, we will discuss some of the questions that are most commonly asked of graphs. Third, we want to make the reader aware of how graphs are used. In Section 9.1, we will discuss these topics in general, and in later sections we will take a closer look at selected topics in graph theory.

Chapter 10 will continue our discussion with an examination of trees, a special type of graph.

9.1: Graphs - General Introduction
9.2: Data Structures for Graphs
9.3: Connectivity
9.4: Traversals- Eulerian and Hamiltonian Graphs
9.5: Graph Optimization
9.6: Planarity and Colorings

This page titled 9: Graph Theory is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

9.1: Graphs - General Introduction

9.1.1: Definitions

Recall that we introduced directed graphs in Chapter 6 as a tool to visualize relations on a set. Here is a formal definition.

Definition 9.1.1: Simple Directed Graph

A simple directed graph consists of a nonempty ***set of vertices***, V, and a ***set of edges***, E, that is a subset of the set $V \times V$.

Note 9.1.1: Some Terminology and Comments

Each edge is an ordered pair of elements from the vertex set. The first entry is the ***initial vertex*** of the edge and the second entry is the ***terminal vertex***. Despite the set terminology in this definition, we often think of a graph as a picture, an aid in visualizing a situation. In Chapter 6, we introduced this concept to help understand relations on sets. Although those relations were principally of a mathematical nature, it remains true that when we see a graph, it tells us how the elements of a set are related to one another. We have chosen not to allow a graph with an empty vertex set, the so-called empty graph. There are both advantages and disadvantages to allowing the empty graph, so you may encounter it in other references.

Example 9.1.1: A Simple Directed Graph

Figure 9.1.1 is an example of a simple directed graph. In set terms, this graph is (V, E), where $V = \{s, a, b\}$ and $E = \{(s, a), (s, b), (a, b), (b, a), (b, b)\}$. Note how each edge is labeled either 0 or 1. There are often reasons for labeling even simple graphs. Some labels are to help make a graph easier to discuss; others are more significant. We will discuss the significance of the labels on this graph later.

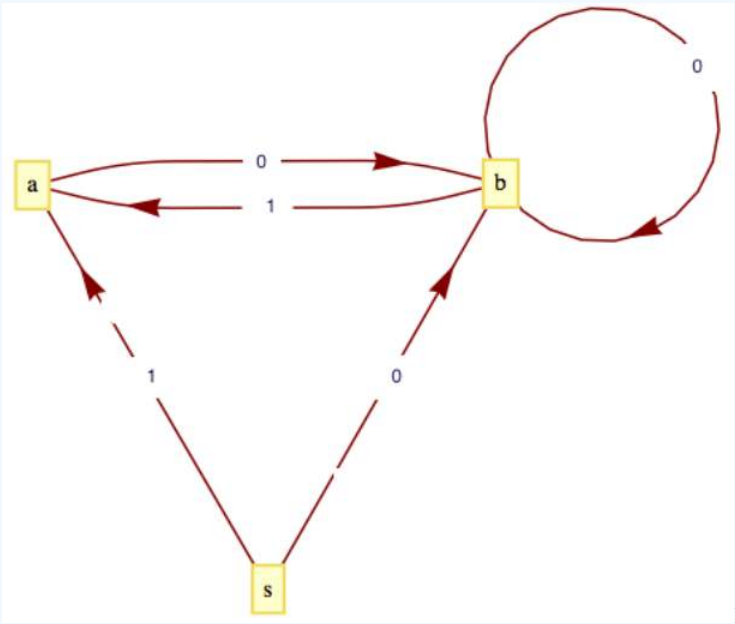


Figure 9.1.1: A directed graph

In certain cases there may be a need for more than one edge between two vertices, and we need to expand the class of directed graphs.

Definition 9.1.2: Multigraph

A multigraph is a set of vertices V with a set of edges that can contain more than one edge between the vertices.

One important point to keep in mind is that if we identify a graph as being a multigraph, it isn't necessary that there are two or more edges between some of the vertices. It is only just *allowed*. In other words, every simple graph is a multigraph. This is analogous to

how a rectangle is a more general geometric figure than a square, but a square is still considered a rectangle.

Example 9.1.2: A Multigraph

A common occurrence of a multigraph is a road map. The cities and towns on the map can be thought of as vertices, while the roads are the edges. It is not uncommon to have more than one road connecting two cities. In order to give clear travel directions, we name or number roads so that there is no ambiguity. We use the same method to describe the edges of the multigraph in Figure 9.1.2. There is no question what $e3$ is; however, referring to the edge $(2, 3)$ would be ambiguous.

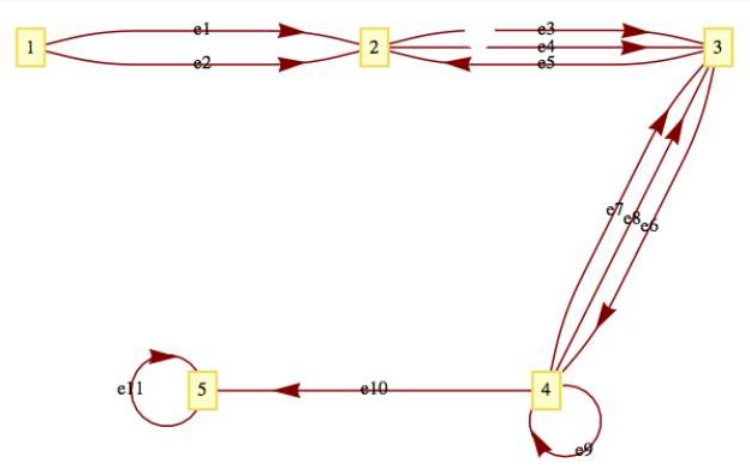


Figure 9.1.2: A directed multigraph

There are cases where the order of the vertices is not significant and so we use a different mathematical model for this situation:

Definition 9.1.3: Undirected Graph

An undirected graph consists of a nonempty set V, called a vertex set, and a set E of two-element subsets of V, called the edge set. The two-element subsets are drawn as lines connecting the vertices. It is customary to not allow "self loops" in undirected graphs.

Note 9.1.2: On Empty Graphs

It may occur to some readers that a graph could be empty, in the sense that it has empty vertex and edge sets. We might refer to this graph as the ***empty graph***. However, there doesn't seem to be a universally agreed upon definition of an empty graph. In some works, a graph with any number of vertices and no edges is called an empty graph. To avoid this dilemma, we have defined both directed and undirected graphs to have nonempty vertex sets. For convenience, we've relaxed this rule in our definition of a Rooted Tree, Definition 10.3.1, and allowed for an empty tree.

Example 9.1.3: An Undirected Graph

A network of computers can be described easily using a graph. Figure 9.1.3 describes a network of five computers, a, b, c, d, and e. An edge between any two vertices indicates that direct two-way communication is possible between the two computers. Note that the edges of this graph are not directed. This is due to the fact that the relation that is being displayed is symmetric (i.e., if X can communicate with Y, then Y can communicate with X). Although directed edges could be used here, it would simply clutter the graph.

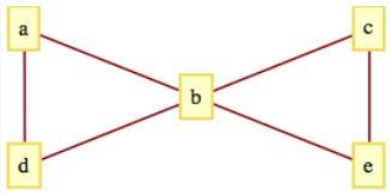


Figure 9.1.3: Communications Map

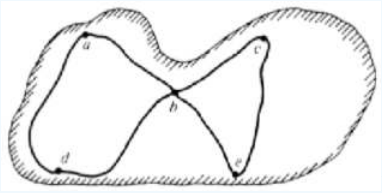


Figure 9.1.4: Island Road Map

This undirected graph, in set terms, is $V = \{a, b, c, d, e\}$ and $E = \{\{a, b\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, e\}, \{b, e\}\}$

There are several other situations for which this graph can serve as a model. One of them is to interpret the vertices as cities and the edges as roads, an abstraction of a map such as the one in Figure 9.1.4. Another interpretation is as an abstraction of the floor plan of a house. See Exercise 9.1.11. Vertex a represents the outside of the house; all others represent rooms. Two vertices are connected if there is a door between them.

Definition 9.1.4: Complete Undirected Graph

A complete undirected graph on n vertices is an undirected graph with the property that each pair of distinct vertices are connected to one another. Such a graph is usually denoted by K_n.

Example 9.1.4: A Labeled Graph

A flowchart is a common example of a simple graph that requires labels for its vertices and some of its edges. Figure 9.1.5 is one such example that illustrates how many problems are solved.

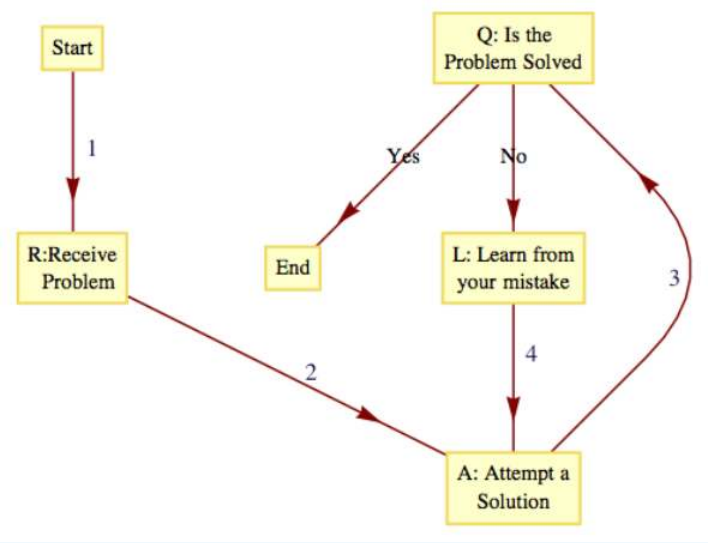


Figure 9.1.5: A flow chart - an example of a labeled graph

At the start of the problem-solving process, we are at the vertex labeled “Start” and at the end (if we are lucky enough to have solved the problem) we will be at the vertex labeled “End.” The sequence of vertices that we pass through as we move from “Start” to “End” is called a path. The “Start” vertex is called the initial vertex of the path, while the “End” is called the final, or terminal, vertex. Suppose that the problem is solved after two attempts; then the path that was taken is Start, R, A, Q, L, A, Q, End. An alternate path description would be to list the edges that were used: $1, 2, 3, \text{No}, 4, 3, \text{Yes}$. This second method of describing a path has the advantage of being applicable for multigraphs. On the graph in Figure 9.1.2, the vertex list $1, 2, 3, 4, 3$does not clearly describe a path between 1 and 3, but e_1, e_4, e_6, e_7 is unambiguous.

Note 9.1.3: A Summary of Path Notation and Terminology

If x and y are two vertices of a graph, then a ***path*** between x and y describes a motion from x to y along edges of the graph. Vertex x is called the initial vertex of the path and y is called the terminal vertex. A path between x and y can always be described by its edge list, the list of edges that were used: $(e_1, e_2, \ldots, e_n)$, where: (1) the initial vertex of e_1 is x; (2) the terminal vertex of e_i is the initial vertex of e_{i+1}, $i = 1, 2, \ldots, n-1$; and (3) the terminal vertex of e_n is y. The number of edges in the edge list is the ***path length***. A path on a simple graph can also be described by a vertex list. A path of length n will have a list of $n+1$ vertices $v_0 = x$, $v_1, v_2, \ldots, v_n = y$, where, for $k = 0, 1, 2, \ldots, n-1$, (v_k, v_{k+1}) is an edge on the graph. A ***circuit*** is a path that terminates at its initial vertex.

Suppose that a path between two vertices has an edge list $(e_1, e_2, \ldots, e_n)$. A ***subpath*** of this graph is any portion of the path described by one or more consecutive edges in the edge list. For example, $(3, \text{No}, 4)$ is a subpath of $(1, 2, 3, \text{No}, 4, 3, \text{Yes})$. Any path is its own subpath; however, we call it an improper subpath of itself. All other nonempty subpaths are called proper subpaths.

A path or circuit is simple if it contains no proper subpath that is a circuit. This is the same as saying that a path or circuit is simple if it does not visit any vertex more than once except for the common initial and terminal vertex in the circuit. In the problem-solving method described in Figure 9.1.5, the path that you take is simple only if you reach a solution on the first try.

9.1.2: Subgraphs

Intuitively, you could probably predict what the term “subgraph” means. A graph contained within a graph, right? But since a graph involves two sets, vertices and edges, does it involve a subset of both of these sets, or just one of them? The answer is it could be either. There are different types of subgraphs. The two that we will define below will meet most of our future needs in discussing the theory of graphs.

Definition 9.1.5: Subgraph

Let $G = (V, E)$ be a graph of any kind: directed, directed multigraph, or undirected. $G' = (V', E')$ is a subgraph of G if $V' \neq \emptyset$, $V' \subseteq V$ and $e \in E'$ only if $e \in E$ and the vertices of e are in V'. You create a subgraph of G by removing zero or more vertices and all edges that include the removed vertices and then you possibly remove some other edges.

If the only removed edges are those that include the removed vertices, then we say that G' is an ***induced subgraph***. Finally, G' is a ***spanning subgraph*** of G if $V' = V$, or, in other words, no vertices are removed from G, only edges.

Example 9.1.5: Some Subgraphs

Consider the graph, G, in the top left of Figure 9.1.6. The other three graphs in that figure are all subgraphs of G. The graph in the top right was created by first removing vertex 5 and all edges connecting it. In addition, we have removed the edge $\{1, 4\}$. That removed edge disqualifies the graph from being an induced subgraph. The graphs in the bottom left and right are both spanning subgraphs. The one on the bottom right is a tree, and is referred to as a spanning subtree. Spanning subtrees will be a discussed in Chapter 10.

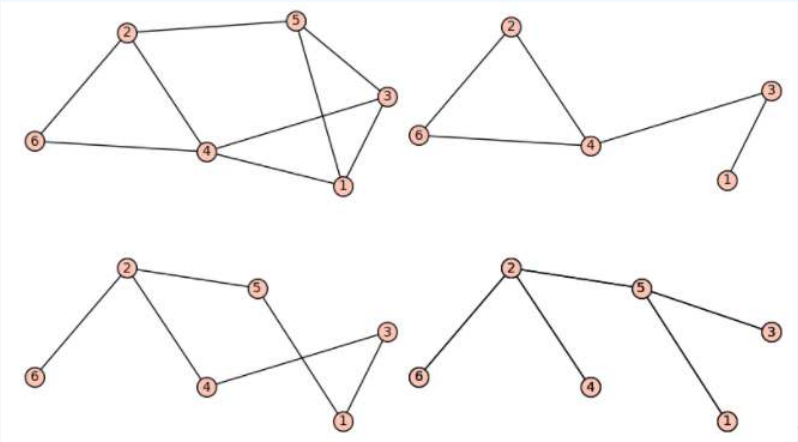


Figure 9.1.6: A few subgraphs

https://math.libretexts.org/@go/page/80534

One set of subgraphs of any graph is the connected components of a graph. For simplicity, we will define them for undirected graphs. Given a graph $G = (V, E)$, consider the relation "is connected to" on V. We interpret this relation so that each vertex is connected to itself, and any two distinct vertices are related if there is a path along edges of the graph from one to the other. It shouldn't be too difficult to convince yourself that this is an equivalence relation on V.

Definition 9.1.6: Connected Component

Given a graph $G = (V, E)$, let C be the relation "is connected to" on V. Then the connected components of G are the induced subgraphs of G each with a vertex set that is an equivalence class with respect to C.

Example 9.1.6

If you ignore the duplicate names of vertices in the four graphs of Figure 9.1.6, and consider the whole figure as one large graph, then there are four connected components in that graph. It's as simple as that! It's harder to describe precisely than to understand the concept.

From the examples we've seen so far, we can see that although a graph can be defined, in short, as a collection of vertices and edges, an integral part of most graphs is the labeling of the vertices and edges that allows us to interpret the graph as a model for some situation. We continue with a few more examples to illustrate this point.

Example 9.1.7: A Graph as a Model for a Set of Strings

Suppose that you would like to mechanically describe the set of strings of 0's and 1's having no consecutive 1's. One way to visualize a string of this kind is with the graph in Figure 9.1.1. Consider any path starting at vertex s. If the label on each graph is considered to be the output to a printer, then the output will have no consecutive 1's. For example, the path that is described by the vertex list $(s, a, b, b, a, b, b, a, b)$ would result in an output of 10010010. Conversely, any string with no consecutive 1's determines a path starting at s.

Example 9.1.8: A Tournament Graph

Suppose that four teams compete in a round-robin sporting event; that is, each team meets every other team once, and each game is played until a winner is determined. If the teams are named A, B, C, and D, we can define the relation β on the set of teams by $X \beta Y$ if X beat Y. For one set of results, the graph of β might look like Figure 9.1.7.

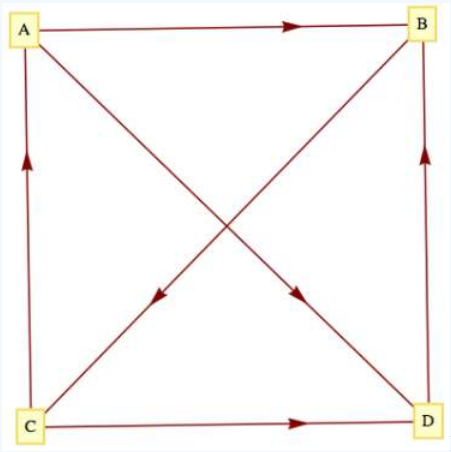


Figure 9.1.7: Round-robin tournament graph with four vertices

There are many types of tournaments and they all can be modeled by different types of graphs.

Definition 9.1.7: Tournament Graph

a. A tournament graph is a directed graph with the property that no edge connects a vertex to itself, and between any two vertices there is at most one edge.

b. A complete (or round-robin) tournament graph is a tournament graph with the property that between any two distinct vertices there is exactly one edge.
c. A single-elimination tournament graph is a tournament graph with the properties that: (i) one vertex (the champion) has no edge terminating at it and at least one edge initiating from it; (ii) every other vertex is the terminal vertex of exactly one edge; and (iii) there is a path from the champion vertex to every other vertex.

Example 9.1.9: Graph of a Single Elimination Tournament

The major league baseball championship is decided with a single-elimination tournament, where each "game" is actually a series of games. From 1969 to 1994, the two divisional champions in the American League (East and West) competed in a series of games. The loser is eliminated and the winner competed against the winner of the National League series (which is decided as in the American League). The tournament graph of the 1983 championship is in Figure 9.1.8

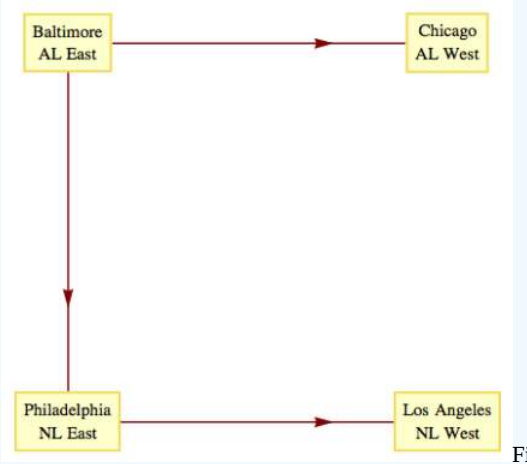


Figure 9.1.8: A single elimination tournament graph

9.1.3: Graph Isomorphisms

Next, we establish the relation "is isomorphic to," a form of equality on graphs. The graphs in Figure 9.1.9 obviously share some similarities, such as the number of vertices and the number of edges. It happens that they are even more similar than just that. If the letters a, b, c, and d in the left graph are replaced with the numbers 1,3,4, and 2, respectively, and the vertices are moved around so that they have the same position as the graph on the right, you get the graph on the right.

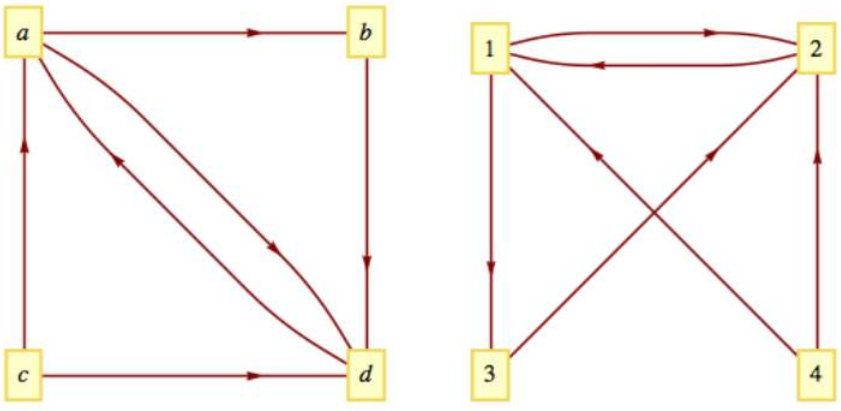


Figure 9.1.9: Isomorphic Graphs

Here is a more precise definition that reflects the fact that the actual positioning (or embedding) of vertices isn't an essential part of a graph.

Definition 9.1.8: Isomorphic Graphs

Two graphs (V, E) and (V', E') are isomorphic if there exists a bijection $f : V \to V'$ such that $(v_i, v_j) \in E$ if and only if $(f(v_i), f(v_j)) \in E'$. For multigraphs, we add that the number of edges connecting v_i to v_j must equal the number of edges from $f(v_i)$ to $f(v_j)$.

The most significant local characteristic of a vertex within a graph is its degree. Collectively, the degrees can partially characterize a graph.

Definition 9.1.9: Degree of a Vertex

a. Let v be a vertex of an undirected graph. The degree of v, denoted $deg(v)$, is the number of edges that connect v to the other vertices in the graph.
b. If v is a vertex of a directed graph, then the outdegree of v, denoted $outdeg(v)$, is the number of edges of the graph that initiate at v. The indegree of v, denoted $indeg(v)$, is the number of edges that terminate at v.

Definition 9.1.10: Degree Sequence of a Graph

The degree sequence of an undirected graph is the non-increasing sequence of its vertex degrees.

Example 9.1.10: Some Degrees

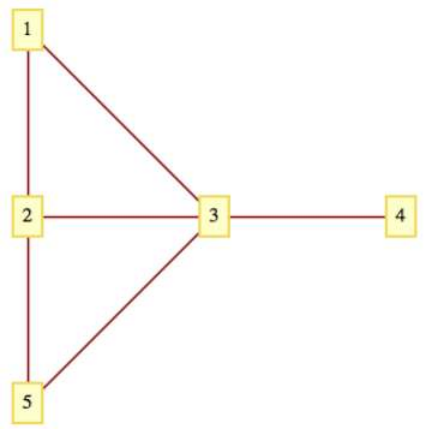


Figure 9.1.10: An undirected graph

a. The degrees of vertices 1 through 5 in Figure 9.1.10 are 2, 3, 4, 1, and 2, respectively. The degree sequence of the graph is $(4, 3, 2, 2, 1)$.
b. In a tournament graph, $outdeg(v)$ is the number of wins for v and $indeg(v)$ is the number of losses. In a complete (round-robin) tournament graph with n vertices, $outdeg(v) + indeg(v) = n - 1$ for each vertex.

Definition 9.1.11: Graphic Sequence

A finite nonincreasing sequence of integers $d_1, d_2, \ldots, d_n$ is graphic if there exists an undirected graph with n vertices having the sequence as its degree sequence.

For example, $4, 2, 1, 1, 1, 1$ is graphic because the degrees of the graph in Figure 9.1.11 match these numbers. There is no connection between the vertex number and its degree in this graph.

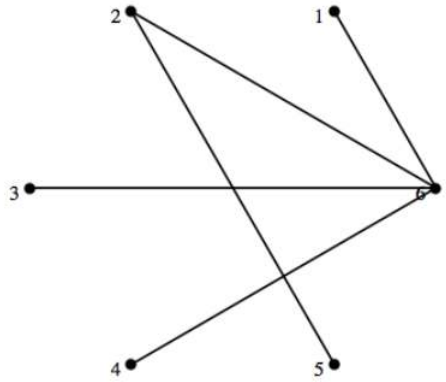


Figure 9.1.11: A graph that shows that $4, 2, 1, 1, 1, 1$ is a graphic sequence.

See [26] for more details on what are also referred to as ***graphical degree sequences***, including an algorithm for determining whether or not a sequence is graphic.

9.1.4: Steps

List 9.1.1: A Prospectus for the Rest of the Chapter

The question "Once you have a graph, what do you do with it?" might come to mind. The following list of common questions and comments about graphs is a partial list that will give you an overview of the remainder of the chapter.

1. How can a graph be represented as a data structure for use on a computer? We will discuss some common data structures that are used to represent graphs in Section 9.2.
2. Given two vertices in a graph, does there exist a path between them? The existence of a path between any or all pairs of vertices in a graph will be discussed in Section 9.3. A related question is: How many paths of a certain type or length are there between two vertices?
3. Is there a path (or circuit) that passes through every vertex (or uses every edge) exactly once? Paths of this kind are called traversals. We will discuss traversals in Section 9.4.
4. Suppose that a cost is associated with the use of each vertex and/or edge in a path. What is the "cheapest" path, circuit, or traversal of a given kind? Problems of this kind will be discussed in Section 9.5.
5. Given the specifications of a graph, or the graph itself, what is the best way to draw the graph? The desire for neatness alone makes this a reasonable question, but there are other motivations. Another goal might be to avoid having edges of the graph cross one another. This is discussed in Section 9.6.

9.1.5: Exercises

Exercise 9.1.1

What is the significance of the fact that there is a path connecting vertex b with every other vertex in Figure 9.1.3, as it applies to various situations that it models?

Answer

In Figure 9.1.3 computer b can communicate with all other computers. In Figure 9.1.4 there are direct roads to and from city b to all other cities.

Exercise 9.1.2

Draw a graph similar to Figure 9.1.1 that represents the set of strings of 0's and 1's containing no more than two consecutive 1's in any part of the string.

Exercise 9.1.3

Draw a directed graph that models the set of strings of 0's and 1's (zero or more of each) where all of the 1's must appear consecutively.

Answer

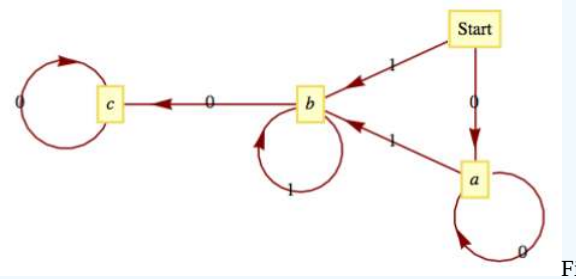


Figure 9.1.12: Solution to Exercise 9.1.3 of Section 9.1

Exercise 9.1.4

In the NCAA final-four basketball tournament, the East champion plays the West champion, and the champions from the Mideast and Midwest play. The winners of the two games play for the national championship. Draw the eight different single-elimination tournament graphs that could occur.

Exercise 9.1.5

What is the maximum number of edges in an undirected graph with eight vertices?

Answer

The maximum number of edges would be $\binom{8}{2} = \frac{(7)(8)}{2} = 28$.

Exercise 9.1.6

Which of the graphs in Figure 9.1.13 are isomorphic? What is the correspondence between their vertices?

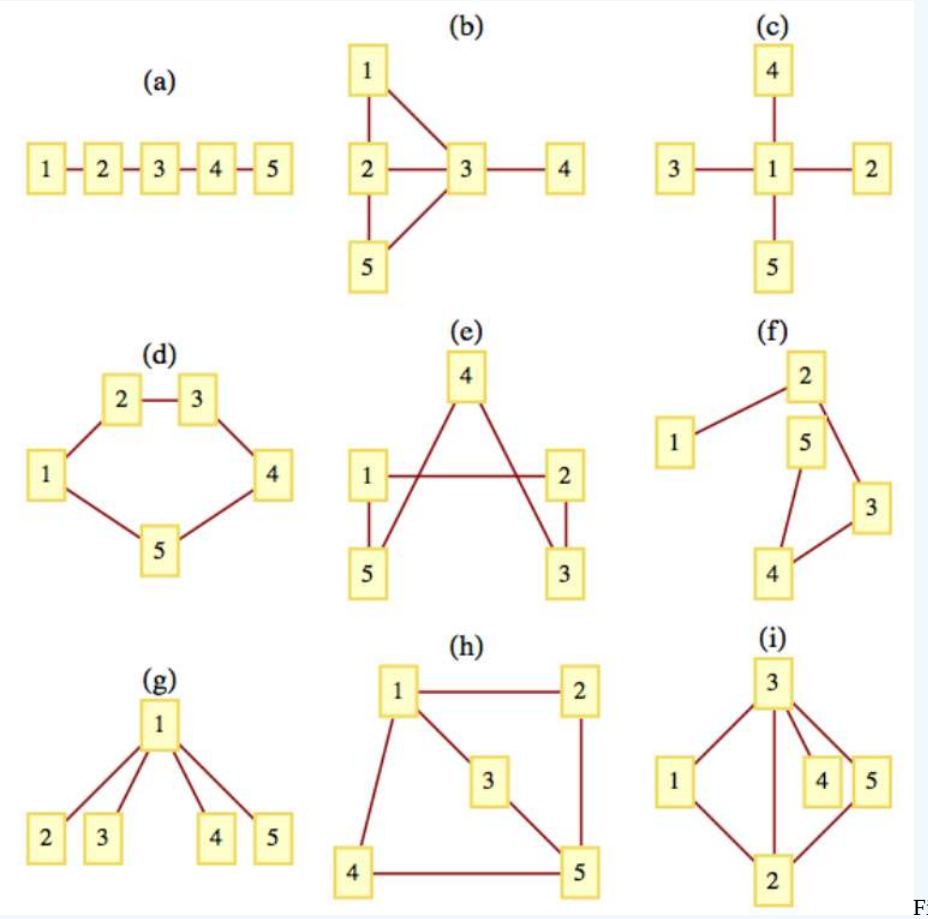


Figure 9.1.13: Which graphs are isomorphic to one another?

Exercise 9.1.7

a. How many edges does a complete tournament graph with n vertices have?
b. How many edges does a single-elimination tournament graph with n vertices have?

Answer

a. $\binom{n}{2} = \frac{(n-1)n}{2}$

b. $n-1$, each vertex except the champion vertex has an indegree of 1 and the champion vertex has an indegree of zero.

Exercise 9.1.8

Draw complete undirected graphs with 1, 2, 3, 4, and 5 vertices. How many edges does a K_n, a complete undirected graph with n vertices, have?

Exercise 9.1.9

Determine whether the following sequences are graphic. Explain your logic.

a. $(6,5,4,3,2,1,0)$
b. $(2,2,2,2,2,2)$
c. $(3,2,2,2,2,2)$
d. $(5,3,3,3,3,3)$
e. $(1,1,1,1,1,1)$
f. $(5,5,4,3,2,1)$

Answer

a. Not graphic - if the degree of a graph with seven vertices is 6, it is connected to all other vertices and so there cannot be a vertex with degree zero.
b. Graphic. One graph with this degree sequence is a cycle of length 6.
c. Not Graphic. The number of vertices with odd degree is odd, which is impossible.
d. Graphic. A "wheel graph" with one vertex connected to all other and the others connected to one another in a cycle has this degree sequence.
e. Graphic. Pairs of vertices connected only to one another.
f. Not Graphic. With two vertices having maximal degree, 5, every vertex would need to have a degree of 2 or more, so the 1 in this sequence makes it non-graphic.

Exercise 9.1.10

a. Based on observations you might have made in Exercise 9.1.9, describe as many characteristics as you can about graphic sequences of length n.
b. Consider the two graphs in Figure 9.1.14 Notice that they have the same degree sequences, $(2,2,2,2,2,2)$.Explain why the two graphs are not isomorphic.

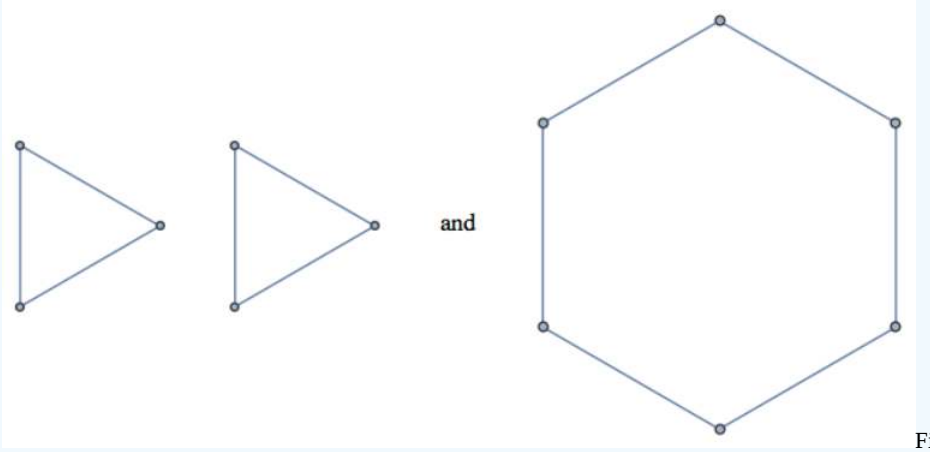


Figure 9.1.14: Two graphs with the same degree sequences

Exercise 9.1.11

Draw a plan for the rooms of a house so that Figure $9.1.3$ models connectedness of the rooms. That is, (a, b) is an edge if and only if a door connects rooms a and b.

Exercise 9.1.12

How many subgraphs are there of a K_n, $n \geq 1$. How many of them are spanning graphs?

This page titled 9.1: Graphs - General Introduction is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

9.2: Data Structures for Graphs

In this section, we will describe data structures that are commonly used to represent graphs. In addition we will introduce the basic syntax for graphs in Sage.

9.2.1: Basic Data Structures

List 9.2.1: Data Structures for Graphs

Assume that we have a graph with n vertices that can be indexed by the integers $1, 2, \ldots, n$. Here are three different data structures that can be employed to represent graphs.

a. Adjacency Matrix: As we saw in Chapter 6, the information about edges in a graph can be summarized with an adjacency matrix, G, where $G_{ij} = 1$ if and only if vertex i is connected to vertex j in the graph. Note that this is the same as the adjacency matrix for a relation.
b. Edge Dictionary: For each vertex in our graph, we maintain a list of edges that initiate at that vertex. If G represents the graph's edge information, then we denote by G_i the list of vertices that are terminal vertices of edges initiating at vertex i. The exact syntax that would be used can vary. We will use Sage/Python syntax in our examples.
c. Edge List: Note that in creating either of the first two data structures, we would presume that a list of edges for the graph exists. A simple way to represent the edges is to maintain this list of ordered pairs, or two element sets, depending on whether the graph is intended to be directed or undirected. We will not work with this data structure here, other than in the first example.

Example 9.2.1: A Very Small Example

We consider the representation of the following graph:

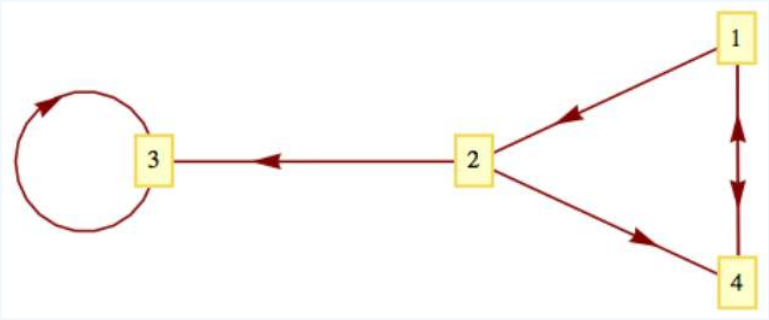


Figure 9.2.1: Graph for a Very Small Example

The adjacency matrix that represents the graph would be

$$G = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

The same graph could be represented with the edge dictionary

```
{1:[2,4],2:[3,4],3:[3],4:[1]}
```

.

Notice the general form of each item in the dictionary: `vertex:[list of vertices]` .

Finally, a list of edges `[(1,2),(1,4),(2,3),(2,4),(3,3),(4,1)]` also describes the same graph.

A natural question to ask is: Which data structure should be used in a given situation? For small graphs, it really doesn't make much difference. For larger matrices the edge count would be a consideration. If n is large and the number of edges is relatively small, it might use less memory to maintain an edge dictionary or list of edges instead of building an $n \times n$ matrix. Some software for working with graphs will make the decision for you.

Example 9.2.2: NCAA Basketball

Consider the tournament graph representing a NCAA Division 1 men's (or women's) college basketball season in the United States. There are approximately 350 teams in Division 1. Suppose we constructed the graph with an edge from team A to team B if A beat B at least once in the season; and we label the edge with the number of wins. Since the average team plays around 30 games in a season, most of which will be against other Division I teams, we could expect around $\frac{30 \cdot 350}{2} = 5,250$ edges in the graph. This would be somewhat reduced by games with lower division teams and cases where two or more wins over the same team produces one edge. Since 5,250 is much smaller than $350^2 = 122,500$ entries in an adjacency matrix, an edge dictionary or edge list would be more compact than an adjacency matrix. Even if we were to use software to create an adjacency matrix, many programs will identify the fact that a matrix such as the one in this example would be "sparse" and would leave data in list form and use sparse array methods to work with it.

9.2.2: Graphs

The most common way to define a graph in Sage is to use an edge dictionary. Here is how the graph in Example 9.2.1 is generated and then displayed. Notice that we simply wrap the function `DiGraph()` around the same dictionary expression we identified earlier.

```
G1 = DiGraph( {1 : [4, 2], 2 : [3, 4], 3 : [3], 4 : [1]})
G1.show()
```

You can get the adjacency matrix of a graph with the `adjacency_matrix` method.

```
G1.adjacency_matrix()
```

You can also define a graph based on its adjacency matrix.

```
M = Matrix([[0,1,0,0,0],[0,0,1,0,0],[0,0,0,1,0],
            [0,0,0,0,1],[1,0,0,0,0]])
DiGraph(M).show()
```

The edge list of any directed graph can be easily retrieved. If you replace `edges` with `edge_iterator`, you can iterate through the edge list. The third coordinate of the items in the edge is the label of the edge, which is `None` in this case.

```
DiGraph(M).edges()
```

Replacing the wrapper `DiGraph()` with `Graph()` creates an undirected graph.

```
G2 = Graph( {1 : [4, 2], 2 : [3, 4], 3 : [3], 4 : [1]})
G2.show()
```

There are many special graphs and graph families that are available in Sage through the `graphs` module. They are referenced with the prefix `graphs.` followed by the name and zero or more parameters inside parentheses. Here are a couple of them, first a complete graph with five vertices.

```
graphs.CompleteGraph(5).show()
```

Here is a wheel graph, named for an obvious pattern of vertices and edges. We assign a name to it first and then show the graph without labeling the vertices.

```
w=graphs.WheelGraph(20)
w.show(vertex_labels=false)
```

There are dozens of graph methods, one of which determines the degree sequence of a graph. In this case, it's the wheel graph above.

```
1 w.degree_sequence()
```

The degree sequence method is defined within the graphs module, but the prefix `graphs.` is not needed because the value of `w` inherits the graphs methods.

9.2.3: Exercises

Exercise 9.2.1

Estimate the number of vertices and edges in each of the following graphs. Would the graph be considered sparse, so that an adjacency matrix would be inefficient?

a. Vertices: Cities of the world that are served by at least one airline. Edges: Pairs of cities that are connected by a regular direct flight.
b. Vertices: ASCII characters. Edges: connect characters that differ in their binary code by exactly two bits.
c. Vertices: All English words. Edges: An edge connects word x to word y if x is a prefix of y.

Answer

a. A rough estimate of the number of vertices in the "world airline graph" would be the number of cities with population greater than or equal to 100,000. This is estimated to be around 4,100. There are many smaller cities that have airports, but some of the metropolitan areas with clusters of large cities are served by only a few airports. 4,000-5,000 is probably a good guess. As for edges, that's a bit more difficult to estimate. It's certainly not a complete graph. Looking at some medium sized airports such as Manchester, NH, the average number of cities that you can go to directly is in the 50-100 range. So a very rough estimate would be $\frac{75 \cdot 4500}{2} = 168,750$. This is far less than $4,500^2$, so an edge list or dictionary of some kind would be more efficient.
b. The number of ASCII characters is 128. Each character would be connected to $\binom{8}{2} = 28$ others and so there are $\frac{128 \cdot 28}{2} = 3,584$ edges. Comparing this to the $128^2 = 16,384$, an array is probably the best choice.
c. The Oxford English Dictionary as approximately a half-million words, although many are obsolete. The number of edges is probably of the same order of magnitude as the number of words, so an edge list or dictionary is probably the best choice.

Exercise 9.2.2

Each edge of a graph is colored with one of the four colors red, blue, yellow, or green. How could you represent the edges in this graph using a variation of the adjacency matrix structure?

Exercise 9.2.3

Directed graphs $G_1, \ldots, G_6$, each with vertex set $\{1, 2, 3, 4, 5\}$ are represented by the matrices below. Which graphs are isomorphic to one another?

$$G_1: \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad G_2: \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad G_3: \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad G_4: \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$G_5: \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad G_6: \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Answer

Each graph is isomorphic to itself. In addition, G_2 and G_4 are isomorphic; and $G_3, G_5,$ and G_6 are isomorphic to one another.

Exercise 9.2.4

The following Sage command verifies that the wheel graph with four vertices is isomorphic to the complete graph with four vertices.

```
graphs.WheelGraph(4).is_isomorphic(graphs.CompleteGraph(4))
```

A list of all graphs in this the `graphs` database is available via tab completion. Type "graphs." and then hit the tab key to see which graphs are available. This can be done using the Sage application or SageMathCloud, but not sage cells. Find some other pairs of isomorphic graphs in the database.

This page titled 9.2: Data Structures for Graphs is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

9.3: Connectivity

This section is devoted to a question that, when posed in relation to the graphs that we have examined, seems trivial. That question is: Given two vertices, s and t, of a graph, is there a path from s to t? If $s = t$, this question is interpreted as asking whether there is a circuit of positive length starting at s. Of course, for the graphs we have seen up to now, this question can be answered after a brief examination.

9.3.1: Preliminaries

There are two situations under which a question of this kind is nontrivial. One is where the graph is very large and an "examination" of the graph could take a considerable amount of time. Anyone who has tried to solve a maze may have run into a similar problem. The second interesting situation is when we want to pose the question to a machine. If only the information on the edges between the vertices is part of the data structure for the graph, how can you put that information together to determine whether two vertices can be connected by a path?

Note 9.3.1: Connectivity Terminology

Let v and w be vertices of a directed graph. Vertex v is connected to vertex w if there is a path from v to w. Two vertices are strongly connected if they are connected in both directions to one another. A graph is connected if, for each pair of distinct vertices, v and w, v is connected to w or w is connected to v. A graph is strongly connected if every pair of its vertices is strongly connected. For an undirected graph, in which edges can be used in either direction, the notions of strongly connected and connected are the same.

Theorem 9.3.1: Maximal Path Theorem

If a graph has n vertices and vertex u is connected to vertex w, then there exists a path from u to w of length no more than n.

Proof

(Indirect): Suppose u is connected to w, but the shortest path from u to w has length m, where $m > n$. A vertex list for a path of length m will have $m+1$ vertices. This path can be represented as $(v_0, v_1, \ldots, v_m)$, where $v_0 = u$ and $v_m = w$. Note that since there are only n vertices in the graph and m vertices are listed in the path after v_0, we can apply the pigeonhole principle and be assured that there must be some duplication in the last m vertices of the vertex list, which represents a circuit in the path. This means that our path of minimum length can be reduced, which is a contradiction.

9.3.2: Adjacency Matrix Method

Algorithm 9.3.1: Adjacency Matrix Method

Suppose that the information about edges in a graph is stored in an adjacency matrix, G. The relation, r, that G defines is vrw if there is an edge connecting v to w. Recall that the composition of r with itself, r^2, is defined by vr^2w if there exists a vertex y such that vry and yrw; that is, v is connected to w by a path of length 2. We could prove by induction that the relation r^k, $k \geq 1$, is defined by vr^kw if and only if there is a path of length k from v to w. Since the transitive closure, r^+, is the union of r, r^2 , $r^3, \ldots$, we can answer our connectivity question by determining the transitive closure of r, which can be done most easily by keeping our relation in matrix form. Theorem 9.3.1 is significant in our calculations because it tells us that we need only go as far as G^n to determine the matrix of the transitive closure.

The main advantage of the adjacency matrix method is that the transitive closure matrix can answer all questions about the existence of paths between any vertices. If G^+ is the matrix of the transitive closure, v_i is connected to v_j if and only if $(G^+)_{ij} = 1$. A directed graph is connected if $(G^+)_{ij} = 1$ or $(G^+)_{ji} = 1$ for each $i \neq j$. A directed graph is strongly connected if its transitive closure matrix has no zeros.

A disadvantage of the adjacency matrix method is that the transitive closure matrix tells us whether a path exists, but not what the path is. The next algorithm will solve this problem.

9.3.3: Breadth-First Search

We will describe the Breadth-First Search Algorithm first with an example.

The football team at Mediocre State University (MSU) has had a bad year, 2 wins and 9 losses. Thirty days after the end of the football season, the university trustees are meeting to decide whether to rehire the head coach; things look bad for him. However, on the day of the meeting, the coach issues the following press release with results from the past year:

List 9.3.1: Press Release: MSU Completes Successful Season

The Mediocre State University football team compared favorably with national champion Enormous State University this season.

- Mediocre State defeated Local A and M.
- Local A and M defeated City College.
- City College defeated Corn State U.
- ... (25 results later)
- Tough Tech defeated Enormous State University (ESU).

...and ESU went on to win the national championship!

The trustees were so impressed that they rehired the coach with a raise! How did the coach come up with such a list?

In reality, such lists exist occasionally and have appeared in newspapers from time to time. Of course they really don't prove anything since each team that defeated MSU in our example above can produce a similar, shorter chain of results. Since college football records are readily available, the coach could have found this list by trial and error. All that he needed to start with was that his team won at least one game. Since ESU lost one game, there was some hope of producing the chain.

The problem of finding this list is equivalent to finding a path in the tournament graph for last year's football season that initiates at MSU and ends at ESU. Such a graph is far from complete and is likely to be represented using edge lists. To make the coach's problem interesting, let's imagine that only the winner of any game remembers the result of the game. The coach's problem has now taken on the flavor of a maze. To reach ESU, he must communicate with the various teams along the path. One way that the coach could have discovered his list in time is by sending the following messages to the coaches of the two teams that MSU defeated during the season:

Note 9.3.2

When this example was first written, we commented that ties should be ignored. Most recent NCAA rules call for a tiebreaker in college football and so ties are no longer an issue. Email was also not common and we described the process in terms of letters, not email messages. Another change is that the coach could also have asked the MSU math department to use Mathematica or Sage to find the path!

List 9.3.2: The Coach's Letter

Dear Football Coach:

Please follow these directions exactly.

1. If you are the coach at ESU, contact the coach at MSU now and tell him who sent you this message.
2. If you are not the coach at ESU and this is the first message of this type that you have received, then:
    - Remember from whom you received this message.
    - Forward a copy of this message, signed by you, to each of the coaches whose teams you defeated during the past year.
    - Ignore this message if you have received one like it already.

Signed,

Coach of MSU

List 9.3.3: Observations

From the conditions of this message, it should be clear that if everyone cooperates and if coaches participate within a day of receiving the message:

1. If a path of length n exists from MSU to ESU, then the coach will know about it in n days.
2. By making a series of phone calls, the coach can construct a path that he wants by first calling the coach who defeated ESU (the person who sent ESU's coach that message). This coach will know who sent him a letter, and so on. Therefore, the vertex list of the desired path is constructed in reverse order.
3. If a total of M football games were played, no more than M messages will be sent out.
4. If a day passes without any message being sent out, no path from MSU to ESU exists.
5. This method could be extended to construct a list of all teams that a given team can be connected to. Simply imagine a series of letters like the one above sent by each football coach and targeted at every other coach.

The general problem of finding a path between two vertices in a graph, if one exists, can be solved exactly as we solved the problem above. The following algorithm, commonly called a breadth-first search, uses a stack.

Note 9.3.3: Stacks

A stack is a fundamental data structure in computer science. A common analogy used to describe stacks is a stack of plates. If you put a plate on the top of a stack and then want to use a plate, it's natural to use that top plate. So the last plate in is the first plate out. “Last in, first out” is the short description of the rule for stacks. This is contrast with a queue which uses a “First in, first out” rule.

Algorithm 9.3.2: Breadth-first Search

A broadcasting algorithm for finding a path between vertex i and vertex j of a graph having n vertices. Each item V_k of a list $V = \{V_1, V_2, \ldots, V_n\}$, consists of a Boolean field $V_k.\text{found}$ and an integer field $V_k.\text{from}$. The sets D_1, $D_2, \ldots$, called depth sets, have the property that if $k \in D_r$, then the shortest path from vertex i to vertex k is of length r. In Step 5, a stack is used to put the vertex list for the path from the vertex i to vertex j in the proper order. That stack is the output of the algorithm.

1. Set the value $V_k.\text{found}$ equal to False, $k = 1, 2, \ldots, n$
2. $r = 0$
3. $D_0 = \{i\}$
4. while $(\neg V_j.\text{found})$ and $(D_r \neq \emptyset)$
   - $D_{r+1} = \emptyset$
   - for each k in D_r:
     for each edge (k,t):
     If $V_t.\text{found}$ == False:
     $V_t.\text{found} = \text{True}$
     $V_t.\text{from} = k$
     $D_{r+1} = D_{r+1} \cup \{t\}$
   - $r = r + 1$
5. if $V_j.\text{found}$:
   - $S = EmptyStack$
   - $k = j$
   - while $V_k.\text{from} \neq i$:
     Push k onto S
     $k = V_k.\text{from}$
   - Push k onto S
   - Push i onto S

List 9.3.4: Notes on Breadth-first Search

- This algorithm will produce one path from vertex i to vertex j, if one exists, and that path will be as short as possible. If more than one path of this length exists, then the one that is produced depends on the order in which the edges are examined and the order in which the elements of D_r are examined in Step 4.
- The condition $D_r \neq \emptyset$ is analogous to the condition that no mail is sent in a given stage of the process, in which case MSU cannot be connected to ESU.
- This algorithm can be easily revised to find paths to all vertices that can be reached from vertex i. Step 5 would be put off until a specific path to a vertex is needed since the information in V contains an efficient list of all paths. The algorithm can also be extended further to find paths between any two vertices.

Example 9.3.1: A Simple Example

Consider the graph below. The existence of a path from vertex 2 to vertex 3 is not difficult to determine by examination. After a few seconds, you should be able to find two paths of length four. Algorithm 9.3.2 will produce one of them.

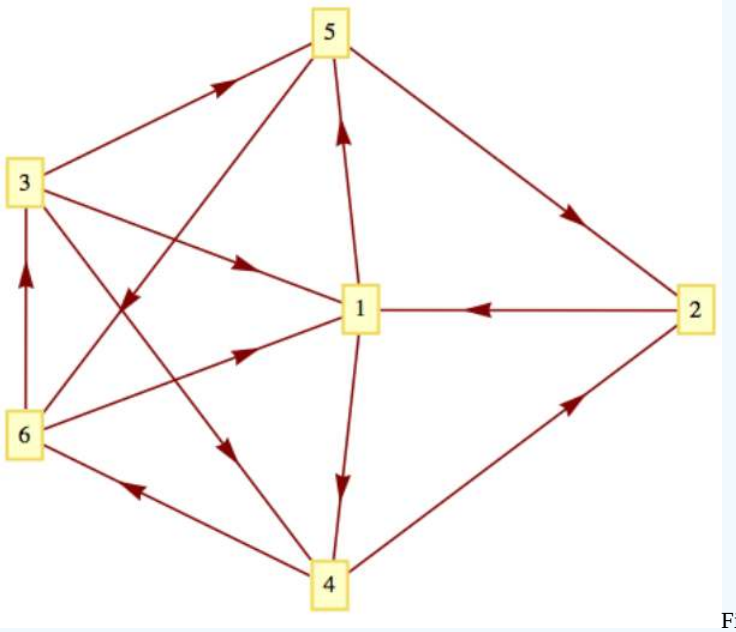


Figure 9.3.1: A simple example of breadth-first search

Suppose that the edges from each vertex are sorted in ascending order by terminal vertex. For example, the edges from vertex 3 would be in the order $(3,1),(3,4),(3,5)$. In addition, assume that in the body of Step 4 of the algorithm, the elements of D_r are used in ascending order. Then at the end of Step 4, the value of V will be

k	1	2	3	4	5	6
V_k. found	T	T	T	T	T	T
V_k. from	2	4	6	1	1	4
Depthset	1	3	4	2	2	3

Therefore, the path $(2,1,4,6,3)$ is produced by the algorithm. Note that if we wanted a path from 2 to 5, the information in V produces the path (2, 1, 5) since $V_k.\text{from} = 1$ and $V_1.\text{from} = 2$. A shortest circuit that initiates at vertex 2 is also available by noting that $V_2.\text{from} = 4$, $V_4.\text{from} = 1$, and $V_1.\text{from} = 2$; thus the circuit $(2,1,4,2)$ is the output of the algorithm.

9.3.4: Graph Measurements

If we were to perform a breadth first search from each vertex in a graph, we could proceed to determine several key measurements relating to the general connectivity of that graph. From each vertex v, the distance from v to any other vertex w, $d(v,w)$, is number of edges in the shortest path from v to w. This number is also the index of the depth set to which w belongs in a breath-first search starting at v.

$$d(v,w) = i \iff w \in D_v(i)$$

where D_v is the family of depth sets starting at v.

If the vector of "from-values" is known from the breath-first search, then the distance can be determined recursively as follows:

$$d(v,v) = 0$$

$$d(v,w) = 1 + d(v, w.\,from) \text{ if } w \neq v$$

Example 9.3.2: Computing Distances

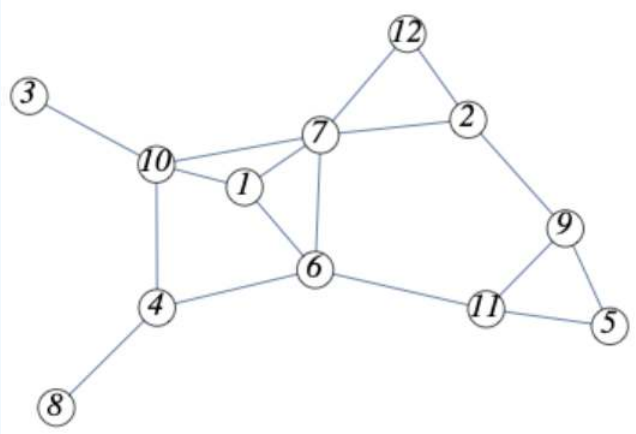


Figure 9.3.2: Graph Measurements Example

Consider Figure 9.3.2. If we perform a breadth first search of this graph starting at vertex 2, for example, we get the following "from data" telling us from what vertex each vertex is reached.

vertex	1	2	3	4	5	6	7	8	9	10	11	12
vertex.from	7	2	10	6	9	7	2	4	2	7	9	2

For example, 4.from has a value of 6. We can compute $d(2,4)$:

$$\begin{aligned} d(2,4) &= 1 + d(2, 4.from) = 1 + d(2,6) \\ &= 2 + d(2, 6.from) = 2 + d(2,7) \\ &= 3 + d(2, 7.from) = 3 + d(2,2) \\ &= 3 \end{aligned}$$

Once we know distances between any two vertices, we can determine the eccentricity of each vertex; and the graph's diameter, radius and center. First, we define these terms precisely.

Definition 9.3.1: Eccentricity of a Vertex

The maximum distance from a vertex to all other vertices, $e(v) = max_w d(v,w)$.

Definition 9.3.2: Diameter of a Graph

The maximum eccentricity of vertices in a graph, denoted $d(G)$.

Definition 9.3.3: Radius of a Graph

The minimum eccentricity of vertices in a graph, denoted $r(G)$.

Definition 9.3.4: Center of a Graph

The set of vertices with minimal eccentricity, $C(G) = \{v \in V \mid e(v) = r(G)\}$

Example 9.3.3: Measurements from Distance Matrices

If we compute all distances between vertices, we can summarize the results in a distance matrix, where the entry in row i, column j is the distance from vertex i to vertex j. For the graph in Example 9.3.2, that matrix is

$$\begin{pmatrix} 0 & 2 & 2 & 2 & 3 & 1 & 1 & 3 & 3 & 1 & 2 & 2 \\ 2 & 0 & 3 & 3 & 2 & 2 & 1 & 4 & 1 & 2 & 2 & 1 \\ 2 & 3 & 0 & 2 & 5 & 3 & 2 & 3 & 4 & 1 & 4 & 3 \\ 2 & 3 & 2 & 0 & 3 & 1 & 2 & 1 & 3 & 1 & 2 & 3 \\ 3 & 2 & 5 & 3 & 0 & 2 & 3 & 4 & 1 & 4 & 1 & 3 \\ 1 & 2 & 3 & 1 & 2 & 0 & 1 & 2 & 2 & 2 & 1 & 2 \\ 1 & 1 & 2 & 2 & 3 & 1 & 0 & 3 & 2 & 1 & 2 & 1 \\ 3 & 4 & 3 & 1 & 4 & 2 & 3 & 0 & 4 & 2 & 3 & 4 \\ 3 & 1 & 4 & 3 & 1 & 2 & 2 & 4 & 0 & 3 & 1 & 2 \\ 1 & 2 & 1 & 1 & 4 & 2 & 1 & 2 & 3 & 0 & 3 & 2 \\ 2 & 2 & 4 & 2 & 1 & 1 & 2 & 3 & 1 & 3 & 0 & 3 \\ 2 & 1 & 3 & 3 & 3 & 2 & 1 & 4 & 2 & 2 & 3 & 0 \end{pmatrix}$$

If we scan the matrix, we can see that the maximum distance is the distance between vertices 3 and 5, which is 5 and is the diameter of the graph. If we focus on individual rows and identify the maximum values, which are the eccentricities, their minimum is 3, which the graph's radius. This eccentricity value is attained by vertices in the set $\{1, 4, 6, 7\}$,which is the center of the graph.

9.3.4.1: SageMath Note - Graph Searching

The following sequence of Sage cells illustrates how searching can be done in graphs.

Generate a random undirected graph with 18 vertices. For each pair of vertices, an edge is included between them with probability 0.2. Since there are $\binom{18}{2} = 153$ potential edges, we expect that there will be approximately $0.2 \cdot 153 \approx 31$ edges. The random number generation is seeded first so that the result will always be the same in spite of the random graph function. Changing or removing that first line will let you experiment with different graphs.

```
set_random_seed(2002)
Gr=graphs.RandomGNP(18,0.2)
Gr.show()
```

Count the number of edges. In this case the number is a bit less than expected.

```
len(Gr.edges(labels=False))
```

Find a shortest path from vertex 0 to vertex 8.

```
Gr.shortest_path(0, 8)
```

Generate a list of vertices that would be reached in a breadth-first search. The expression `Gr.breadth_first_search(0)` creates an iterator that is convenient for programming. Wrapping `list( )` around the expression shows the order in which the vertices are visited with the depth set indicated in the second coordinates.

```
list(Gr.breadth_first_search(0,report_distance='True'))
```

9.3.5: Exercises

Exercise 9.3.1

Apply Algorithm 9.3.2 to find a path from 5 to 1 in Figure 9.3.1. What would be the final value of V? Assume that the terminal vertices in edge lists and elements of the depth sets are put into ascending order, as we assumed in Example 9.3.1.

Answer

k	1	2	3	4	5	6
$V[k]$. found	T	T	T	F	F	T
$V[k]$. from	2	5	6	$*$	$*$	5
DepthSet	2	1	2	$*$	$*$	1

($* =$ undefined)

Exercise 9.3.2

Apply Algorithm 9.3.2 to find a path from d to c in the road graph in Example 9.1.3 using the edge list in that example. Assume that the elements of the depth sets are put into ascending order.

Exercise 9.3.3

In a simple undirected graph with no self-loops, what is the maximum number of edges you can have, keeping the graph unconnected? What is the minimum number of edges that will assure that the graph is connected?

Answer

If the number of vertices is n, there can be $\frac{(n-1)(n-2)}{2}$ vertices with one vertex not connected to any of the others. One more edge and connectivity is assured.

Exercise 9.3.4

Use a broadcasting algorithm to determine the shortest path from vertex a to vertex i in the graphs shown in the Figure 9.3.3 below. List the depth sets and the stack that is created.

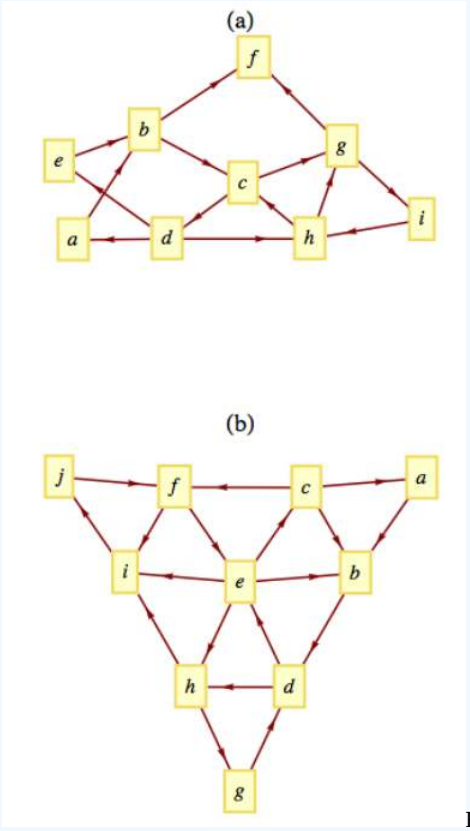


Figure 9.3.3: Shortest paths from a to i?

Exercise 9.3.5

For each of the following graphs, determine the eccentricities of each vertex, and the diameter, radius, and center of the graph.

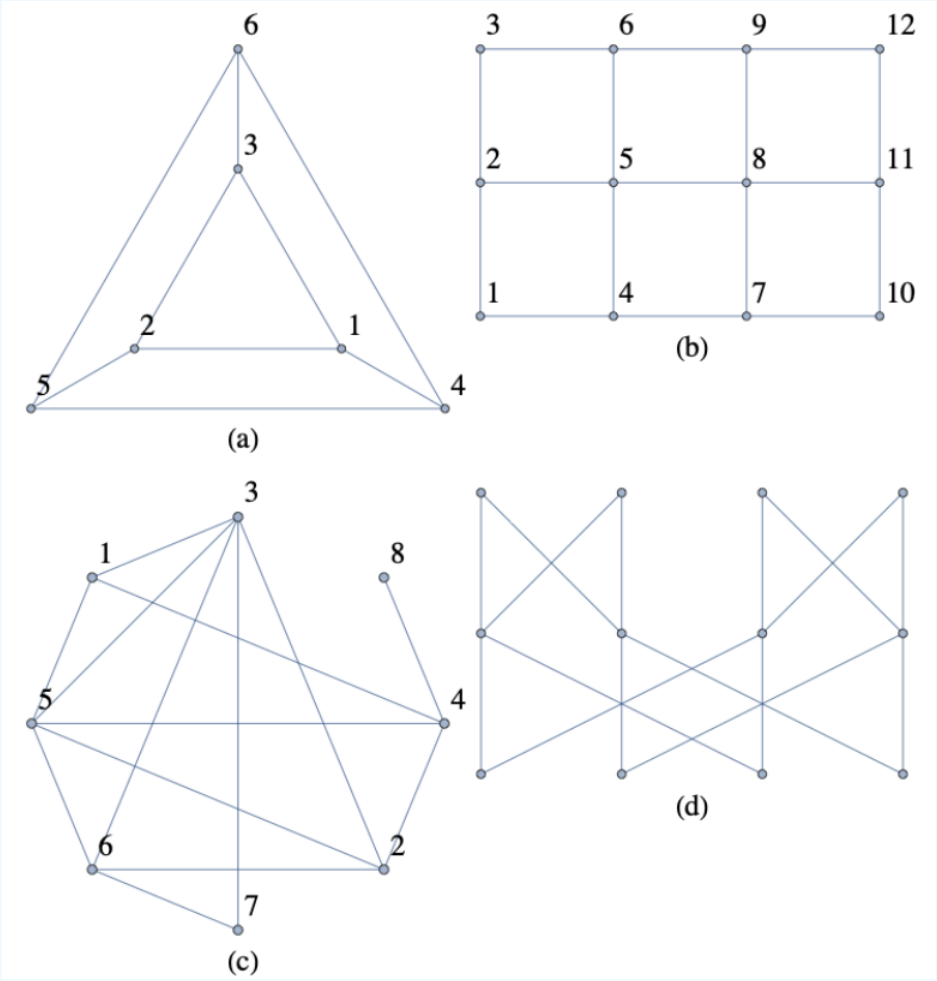


Figure 9.3.4: An undirected graph with 6 vertices.

Answer

a. The eccentricity of each vertex is 2; and the diameter and radius are both 2 as well. All vertices are part of the center.
b. The corners (1,3,10 and 10) have eccentricities 5. The two central vertices, 5 and 8, which are in the center of the graph have eccentricity 3. All other vertices have eccentricity 4. The diameter is 5. The radius is 3.
c. Vertices 1, 2 and 5 have eccentricity 2 and make up the center of this graph. Verticies 7 and 8 have eccentricity 4, and all other vertices have eccentricity 3. The diameter is 4. The radius is 2.
d. The eccentricity of each vertex is 4; and the diameter and radius are both 4 as well. All vertices are part of the center.

Exercise 9.3.6

a. The terms diameter, radius and center are familiar ones in the context of circles. Compare their usage in circles and graphs. How are they similar and how are they different?
b. "Eccentricity" might be less familiar. How is is used in geometry, and does it have a compatible use in graph theory?

Exercise 9.3.7

Prove (by induction on k) that if the relation r on vertices of a graph is defined by vrw if there is an edge connecting v to w, then r^k, $k \geq 1$, is defined by vr^kw if there is a path of length k from v to w.

Answer

Basis: $(k=1)$ Is the relation r^1, defined by vr^1w if there is a path of length 1 from v to w? Yes, since vrw if and only if an edge, which is a path of length 1, connects v to w.

Induction: Assume that vr^kw if and only if there is a path of length k from v to w. We must show that $vr^{k+1}w$ if and only if there is a path of length $k+1$ from v to w.

$$vr^{k+1}w \Rightarrow vr^k y \text{ and } yrw \text{ for some vertex } y$$

By the induction hypothesis, there is a path of length k from v to y. And by the basis, there is a path of length one from y to w. If we combine these two paths, we obtain a path of length $k+1$ from v to w. Of course, if we start with a path of length $k+1$ from v to w, we have a path of length k from v to some vertex y and a path of length 1 from y to w. Therefore, $vr^k y$ and $yrw \Rightarrow vr^{k+1}w$.

Exercise 9.3.8

For each of the following distance matrices of graphs, identify the diameter, radius and center. Assume the graphs vertices are the numbers 1 through n for an $n \times n$ matrix.

a.
$$\begin{pmatrix} 0 & 2 & 1 & 2 & 2 & 3 & 3 & 2 & 1 & 1 \\ 2 & 0 & 1 & 2 & 3 & 3 & 3 & 2 & 3 & 2 \\ 1 & 1 & 0 & 1 & 2 & 2 & 2 & 1 & 2 & 1 \\ 2 & 2 & 1 & 0 & 3 & 3 & 3 & 2 & 3 & 2 \\ 2 & 3 & 2 & 3 & 0 & 2 & 1 & 1 & 2 & 1 \\ 3 & 3 & 2 & 3 & 2 & 0 & 1 & 1 & 3 & 2 \\ 3 & 3 & 2 & 3 & 1 & 1 & 0 & 1 & 3 & 2 \\ 2 & 2 & 1 & 2 & 1 & 1 & 1 & 0 & 2 & 1 \\ 1 & 3 & 2 & 3 & 2 & 3 & 3 & 2 & 0 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}$$

b.
$$\begin{pmatrix} 0 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 2 & 3 & 1 & 1 \\ 2 & 0 & 2 & 2 & 1 & 1 & 1 & 3 & 2 & 1 & 1 & 3 \\ 2 & 2 & 0 & 1 & 3 & 2 & 1 & 2 & 2 & 3 & 1 & 1 \\ 2 & 2 & 1 & 0 & 3 & 1 & 2 & 1 & 2 & 3 & 2 & 1 \\ 3 & 1 & 3 & 3 & 0 & 2 & 2 & 4 & 3 & 2 & 2 & 4 \\ 3 & 1 & 2 & 1 & 2 & 0 & 2 & 2 & 3 & 2 & 2 & 2 \\ 3 & 1 & 1 & 2 & 2 & 2 & 0 & 3 & 3 & 2 & 2 & 2 \\ 1 & 3 & 2 & 1 & 4 & 2 & 3 & 0 & 3 & 4 & 2 & 2 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 0 & 1 & 3 & 1 \\ 3 & 1 & 3 & 3 & 2 & 2 & 2 & 4 & 1 & 0 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 2 & 0 & 2 \\ 1 & 3 & 1 & 1 & 4 & 2 & 2 & 2 & 1 & 2 & 2 & 0 \end{pmatrix}$$

This page titled 9.3: Connectivity is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

9.4: Traversals- Eulerian and Hamiltonian Graphs

The subject of graph traversals has a long history. In fact, the solution by Leonhard Euler (Switzerland, 1707-83) of the Koenigsberg Bridge Problem is considered by many to represent the birth of graph theory.

9.4.1: Eulerian Graphs

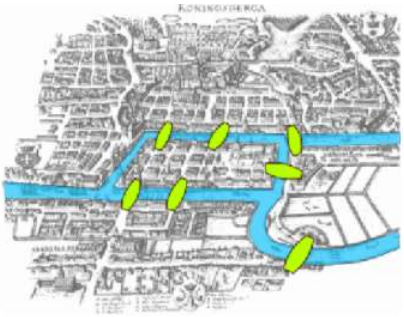


Figure 9.4.1: A map of Koenigsberg, circa 1735

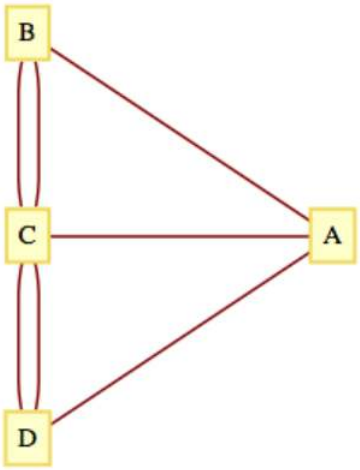


Figure 9.4.2: A multigraph for the bridges of Koenigsberg

A map of the Prussian city of Koenigsberg (circa 1735) in Figure 9.4.1 shows that there were seven bridges connecting the four land masses that made up the city. The legend of this problem states that the citizens of Koenigsberg searched in vain for a walking tour that passed over each bridge exactly once. No one could design such a tour and the search was abruptly abandoned with the publication of Euler's Theorem.

Theorem 9.4.1: Euler's Theorem: Koenigsberg Case

No walking tour of Koenigsberg can be designed so that each bridge is used exactly once.

Proof

The map of Koenigsberg can be represented as an undirected multigraph, as in Figure 9.4.2 The four land masses are the vertices and each edge represents a bridge.

The desired tour is then a path that uses each edge once and only once. Since the path can start and end at two different vertices, there are two remaining vertices that must be intermediate vertices in the path. If x is an intermediate vertex, then every time that you visit x, you must use two of its incident edges, one to enter and one to exit. Therefore, there must be an even number of edges connecting x to the other vertices. Since every vertex in the Koenigsberg graph has an odd number of edges, no tour of the type that is desired is possible.

As is typical of most mathematicians, Euler wasn't satisfied with solving only the Koenigsberg problem. His original theorem, which is paraphrased below, concerned the existence of paths and circuits like those sought in Koenigsberg. These paths and circuits have become associated with Euler's name.

Definition 9.4.1: Eulerian Paths, Circuits, Graphs

An Eulerian path through a graph is a path whose edge list contains each edge of the graph exactly once. If the path is a circuit, then it is called an Eulerian circuit. An Eulerian graph is a graph that possesses an Eulerian circuit.

Example 9.4.1: An Eulerian Graph

Without tracing any paths, we can be sure that the graph below has an Eulerian circuit because all vertices have an even degree. This follows from the following theorem.

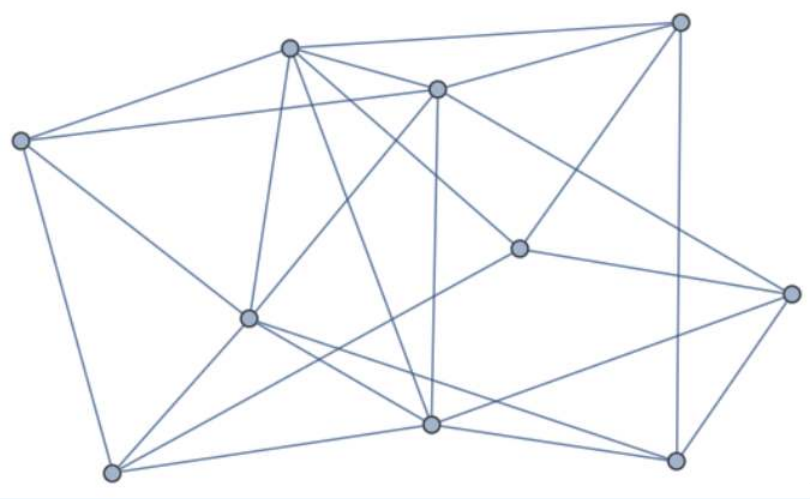


Figure 9.4.3: An Eulerian graph

Theorem 9.4.2: Euler's Theorem: General Case

An undirected graph has an Eulerian path if and only if it is connected and has either zero or two vertices with an odd degree. If no vertex has an odd degree, then the graph is Eulerian.

Proof

It can be proven by induction that the number of vertices in an undirected graph that have an odd degree must be even. We will leave the proof of this fact to the reader as an exercise. The necessity of having either zero or two vertices of odd degree is clear from the proof of the Koenigsberg case of this theorem. Therefore, we will concentrate on proving that this condition is sufficient to ensure that a graph has an Eulerian path. Let k be the number of vertices with odd degree.

Phase 1. If $k = 0$, start at any vertex, v_0, and travel along any path, not using any edge twice. Since each vertex has an even degree, this path can always be continued past each vertex that you reach except v_0. The result is a circuit that includes v_0. If $k = 2$, let v_0 be either one of the vertices of odd degree. Trace any path starting at v_0 using up edges until you can go no further, as in the $k = 0$ case. This time, the path that you obtain must end at the other vertex of odd degree that we will call v_1. At the end of Phase 1, we have an initial path that may or may not be Eulerian. If it is not Eulerian, Phase 2 can be repeated until all of the edges have been used. Since the number of unused edges is decreased in any use of Phase 2, an Eulerian path must be obtained in a finite number of steps.

Phase 2. As we enter this phase, we have constructed a path that uses a proper subset of the edges in our graph. We will refer to this path as the current path. Let V be the vertices of our graph, E the edges, and E_u the edges that have been used in the current path. Consider the graph $G' = (V, E - E_u)$. Note that every vertex in G' has an even degree. Select any edge, e, from G'. Let v_a and v_b be the vertices that e connects. Trace a new path starting at v_a whose first edge is e. We can be sure that at least one vertex of the new path is also in the current path since (V, E) is connected. Starting at v_a, there exists a path in (V, E) to any vertex in the current path. At some point along this path, which we can consider the start of the new path, we will have intersected the current path. Since the degree of each vertex in G' is even, any path that we start at v_a can be continued until it is a circuit. Now, we simply augment the current path with this circuit. As we travel along the current path, the first time that we intersect the new path, we travel along it (see Figure 9.4.4). Once we complete the circuit that is the new path, we resume the traversal of the current path.

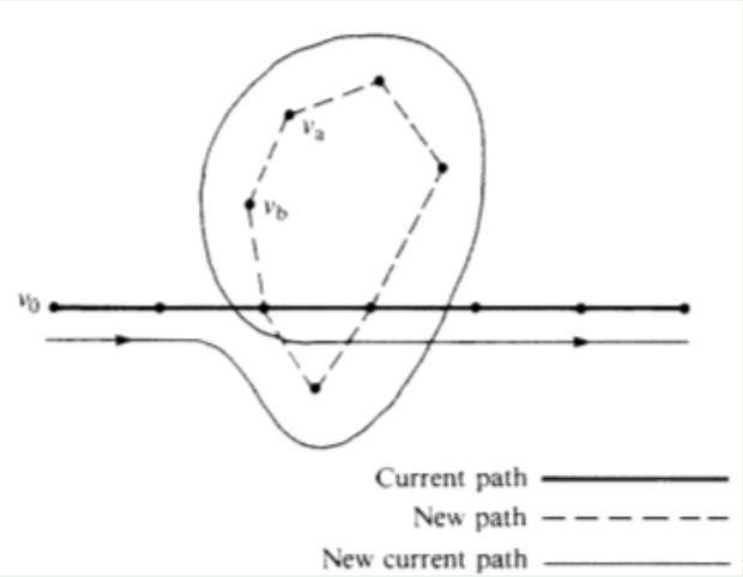


Figure 9.4.4: Path Augmentation Plan

If the result of this phase is an Eulerian path, then we are finished; otherwise, repeat this phase.

Example 9.4.2: Complete Eulerian Graphs

The complete undirected graphs K_{2n+1}, $n = 1, 2, 3, \ldots..$, are Eulerian. If $n \geq 1$, then K_{2n} is not Eulerian.

9.4.2: Hamiltonian Graphs

To search for a path that uses every vertex of a graph exactly once seems to be a natural next problem after you have considered Eulerian graphs.The Irish mathematician Sir William Rowan Hamilton (1805-65) is given credit for first defining such paths. He is also credited with discovering the quaternions, for which he was honored by the Irish government with a postage stamp in 2005.

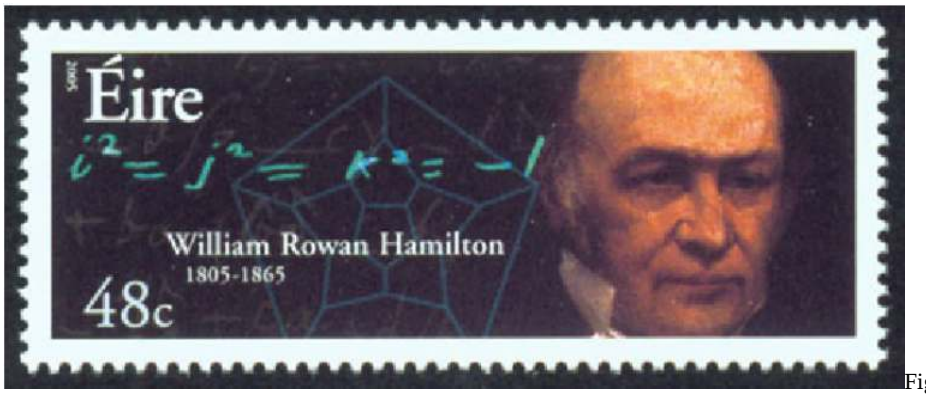


Figure 9.4.5: Irish stamp honoring Sir William Rowan Hamilton

Definition 9.4.2: Hamiltonian Path, Circuit, and Graphs

A Hamiltonian path through a graph is a path whose vertex list contains each vertex of the graph exactly once, except if the path is a circuit, in which case the initial vertex appears a second time as the terminal vertex. If the path is a circuit, then it is called a Hamiltonian circuit. A Hamiltonian graph is a graph that possesses a Hamiltonian circuit.

Example 9.4.3: The Original Hamiltonian Graph

Figure 9.4.7 shows a graph that is Hamiltonian. In fact, it is the graph that Hamilton used as an example to pose the question of existence of Hamiltonian paths in 1859. In its original form, the puzzle that was posed to readers was called “Around the World.” The vertices were labeled with names of major cities of the world and the object was to complete a tour of these cities.

The graph is also referred to as the dodecahedron graph, where vertices correspond with the corners of a dodecahedron and the edges are the edges of the solid that connect the corners.

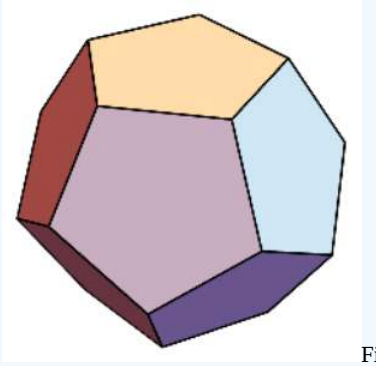


Figure 9.4.6: A Dodecahedron

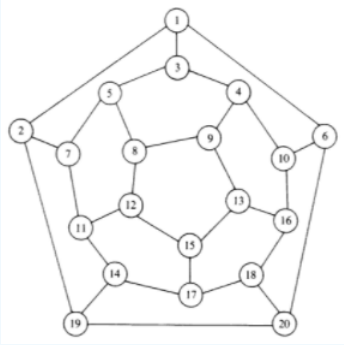


Figure 9.4.7: The Dodecahedron Graph

Problem 9.4.1

Unfortunately, a simple condition doesn't exist that characterizes a Hamiltonian graph. An obvious necessary condition is that the graph be connected; however, there is a connected undirected graph with four vertices that is not Hamiltonian. Can you draw such a graph?

Note 9.4.1: What is Possible and What is Impossible?

The search for a Hamiltonian path in a graph is typical of many simple-sounding problems in graph theory that have proven to be very difficult to solve. Although there are simple algorithms for conducting the search, they are impractical for large problems because they take such a long time to complete as graph size increases. Currently, every algorithm to search for a Hamiltonian path in a graph takes a time that grows at a rate that is greater than any polynomial as a function of the number of vertices. Rates of this type are called "***super-polynomial***." That is, if $T(n)$ is the time it takes to search a graph of n vertices, and $p(n)$ is any polynomial, then $T(n) > p(n)$ for all but possibly a finite number of positive values for n.

It is an unproven but widely held belief that no faster algorithm exists to search for Hamiltonian paths in general graphs. To sum up, the problem of determining whether a graph is Hamiltonian is theoretically possible; however, for large graphs we consider it a practical impossibility. Many of the problems we will discuss in the next section, particularly the Traveling Salesman Problem, are thought to be impossible in the same sense.

Definition 9.4.3: The n-cube

Let $n \geq 1$, and let B^n be the set of strings of 0's and 1's with length n. The n-cube is the undirected graph with a vertex for each string in B^n and an edge connecting each pair of strings that differ in exactly one position. The n-cube is normally denoted Q_n.

The n-cube is among the graphs that are defined within the `graphs` package of SageMath and is created with the expression `graphs.CubeGraph(n)` .

```
graphs.CubeGraph(4).show(layout="spring")
```

Example 9.4.4: Analog-to-Digital Conversion and the Gray Code

A common problem encountered in engineering is that of analog-to-digital (a-d) conversion, where the reading on a dial, for example, must be converted to a numerical value. In order for this conversion to be done reliably and quickly, one must solve an interesting problem in graph theory. Before this problem is posed, we will make the connection between a-d conversion and the graph problem using a simple example. Suppose a dial can be turned in any direction, and that the positions will be converted to one of the numbers zero through seven as depicted in Figure 9.4.8. The angles from 0 to 360 are divided into eight equal parts, and each part is assigned a number starting with 0 and increasing clockwise. If the dial points in any of these sectors the conversion is to the number of that sector. If the dial is on the boundary, then we will be satisfied with the conversion to either of the numbers in the bordering sectors. This conversion can be thought of as giving an approximate angle of the dial, for if the dial is in sector k, then the angle that the dial makes with east is approximately $45k^{\circ}$.

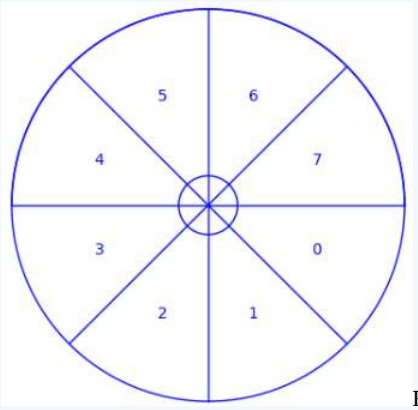


Figure 9.4.8: Analog-Digital Dial

Now that the desired conversion has been identified, we will describe a “solution” that has one major error in it, and then identify how this problem can be rectified. All digital computers represent numbers in binary form, as a sequence of 0's and 1's called bits, short for binary digits. The binary representations of numbers 0 through 7 are:

$$
\begin{aligned}
0 &= 000_{two} = 0\cdot 4 + 0\cdot 2 + 0\cdot 1\\
1 &= 001_{two} = 0\cdot 4 + 0\cdot 2 + 1\cdot 1\\
2 &= 010_{two} = 0\cdot 4 + 1\cdot 2 + 0\cdot 1\\
3 &= 011_{two} = 0\cdot 4 + 1\cdot 2 + 1\cdot 1\\
4 &= 100_{two} = 1\cdot 4 + 0\cdot 2 + 0\cdot 1\\
5 &= 101_{two} = 1\cdot 4 + 0\cdot 2 + 1\cdot 1\\
6 &= 110_{two} = 1\cdot 4 + 1\cdot 2 + 0\cdot 1\\
7 &= 111_{two} = 1\cdot 4 + 1\cdot 2 + 1\cdot 1
\end{aligned}
$$

The way that we could send those bits to a computer is by coating parts of the back of the dial with a metallic substance, as in Figure 9.4.9. For each of the three concentric circles on the dial there is a small magnet. If a magnet lies under a part of the dial that has been coated with metal, then it will turn a switch ON, whereas the switch stays OFF when no metal is detected above a magnet. Notice how every ON/OFF combination of the three switches is possible given the way the back of the dial is coated.

If the dial is placed so that the magnets are in the middle of a sector, we expect this method to work well. There is a problem on certain boundaries, however. If the dial is turned so that the magnets are between sectors three and four, for example, then it is unclear what the result will be. This is due to the fact that each magnet will have only a fraction of the required metal above it to turn its switch ON. Due to expected irregularities in the coating of the dial, we can be safe in saying that for each switch either ON or OFF could be the result, and so if the dial is between sectors three and four, any number could be indicated. This problem does not occur between every sector. For example, between sectors 0 and 1, there is only one switch that cannot be predicted. No matter what the outcome is for the units switch in this case, the indicated sector must be either 0 or 1. This consistent with the original objective that a positioning of the dial on a boundary of two sectors should produce the number of either sector.

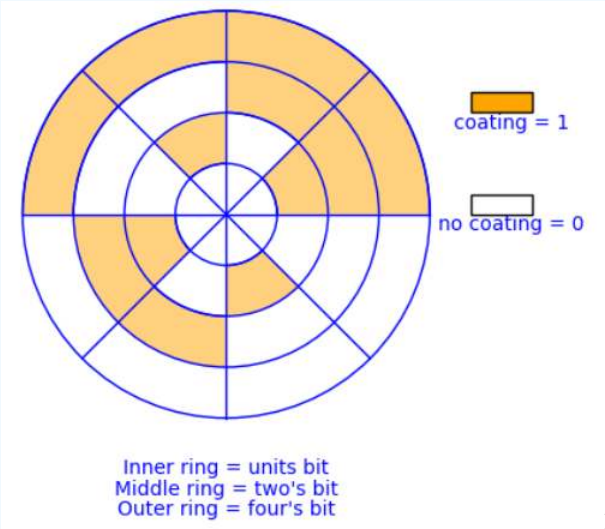


Figure 9.4.9: Coating scheme for the Analog-Digital Dial

Is there a way to coat the sectors on the back of the dial so that each of the eight patterns corresponding to the numbers 0 to 7 appears once, and so that between any two adjacent sectors there is only one switch that will have a questionable setting? What we are describing here is a Hamiltonian circuit of the (Figure 9.4.10). If one can draw a path along the edges in the 3-cube that starts at any vertex, passes through every other vertex once, and returns to the start, then that sequence of bit patterns can be used to coat the back of the dial so that between every sector there is only one questionable switch. Such a path is not difficult to find, as we will see below.

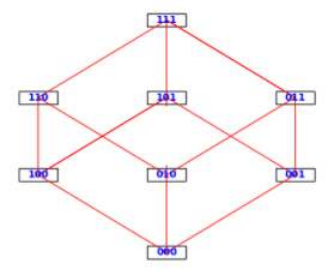


Figure 9.4.10: The 3-Cube

Many A-D conversion problems require many more sectors and switches than this example, and the same kinds of problems can occur. The solution would be to find a path within a much larger yet similar graph. For example, there might be 1,024 sectors with 10 switches, resulting in a graph with 1,024 vertices. Fortunately, our solution will apply to the n-cube for any positive value of n.

A Hamiltonian circuit of the n-cube can be described recursively. The circuit itself, called the Gray Code, is not the only Hamiltonian circuit of the n-cube, but it is the easiest to describe. The standard way to write the Gray Code is as a column of strings, where the last string is followed by the first string to complete the circuit.

Basis for the Gray Code ($n = 1$): The Gray Code for the 1-cube is $G_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Note that the edge between 0 and 1 is used twice in this circuit. That doesn't violate any rules for Hamiltonian circuits, but can only happen if a graph has two vertices.

Recursive definition of the Gray Code: Given the Gray Code for the n-cube, $n \geq 1$, then G_{n+1} is obtained by (1) listing G_n with each string prefixed with 0, and then (2) reversing the list of strings in G_n with each string prefixed with 1. Symbolically, the recursion can be expressed as follows, where G_n^r is the reverse of list G_n.

$$G_{n+1} = \begin{pmatrix} 0G_n \\ 1G_n^r \end{pmatrix}$$

The Gray Codes for the 2-cube and 3-cube are

$$G_2 = \begin{pmatrix} 00 \\ 01 \\ 11 \\ 10 \end{pmatrix} \text{ and } G_3 = \begin{pmatrix} 000 \\ 001 \\ 011 \\ 010 \\ 110 \\ 111 \\ 101 \\ 100 \end{pmatrix}$$

One question might come to mind at this point. If the coatings of the dial no longer in the sequence from 0 to 7, how would you interpret the patterns that are on the back of the dial as numbers from 0 to 7? In Chapter 14 we will see that if the Gray Code is used, this "decoding" is quite easy.

Example 9.4.5: Applications of the Gray Code

One application of the Gray code was discussed in the Introduction to this book. Another application is in statistics. In a statistical analysis, there is often a variable that depends on several factors, but exactly which factors are significant may not be obvious. For each subset of factors, there would be certain quantities to be calculated. One such quantity is the multiple correlation coefficient for a subset. If the correlation coefficient for a given subset, A, is known, then the value for any subset that is obtained by either deleting or adding an element to A can be obtained quickly. To calculate the correlation coefficient for each set, we simply travel along G_n, where n is the number of factors being studied. The first vertex will always be the string of 0's, which represents the empty set. For each vertex that you visit, the set that it corresponds to contains the kth factor if the kth character is a 1.

The 3-cube and its generalization, the n-cube, play a role in the design of a multiprocessor called a hypercube. A multiprocessor is a computer that consists of several independent processors that can operate simultaneously and are connected to one another by a network of connections. In a hypercube with $M = 2^n$ processors, the processors are numbered 0 to $M - 1$. Two processors are connected if their binary representations differ in exactly one bit. The hypercube has proven to be the best possible network for certain problems requiring the use of a "supercomputer."

9.4.3: Exercises

Exercise 9.4.1

Locate a map of New York City and draw a graph that represents its land masses, bridges and tunnels. Is there an Eulerian path through New York? You can do the same with any other city that has at least two land masses.

Answer

Using a recent road map, it appears that an Eulerian circuit exists in New York City, not including the small islands that belong to the city. Lowell, Massachusetts, is located at the confluence of the Merrimack and Concord rivers and has several canals flowing through it. No Eulerian path exists for Lowell.

Exercise 9.4.2

Which of the drawings in Figure 9.4.11 can be drawn without removing your pencil from the paper and without drawing any line twice?

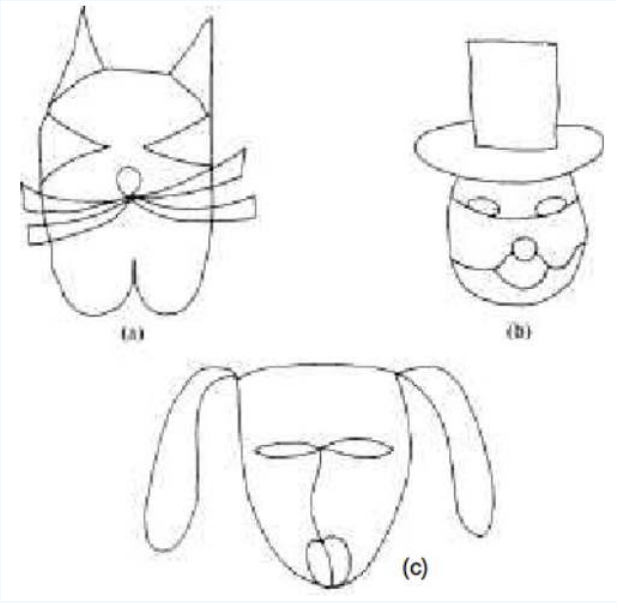


Figure 9.4.11

Exercise 9.4.3

Write out the Gray Code for the 4-cube.

Answer

Gray Code for the 4-cube:

$$G_4 = \begin{pmatrix} 0000 \\ 0001 \\ 0011 \\ 0010 \\ 0110 \\ 0111 \\ 0101 \\ 0100 \\ 1100 \\ 1101 \\ 1111 \\ 1110 \\ 1010 \\ 1011 \\ 1001 \\ 1000 \end{pmatrix}$$

Exercise 9.4.4

Find a Hamiltonian circuit for the dodecahedron graph in Figure 9.4.7.

Exercise 9.4.5

The Euler Construction Company has been contracted to construct an extra bridge in Koenigsberg so that an Eulerian path through the town exists. Can this be done, and if so, where should the bridge be built?

Answer

Any bridge between two land masses will be sufficient. To get an Eulerian circuit, you must add a second bridge that connects the two land masses that were not connected by the first bridge.

Exercise 9.4.6

Consider the graphs in Figure 9.4.12 Determine which of the graphs have an Eulerian path, and find an Eulerian path for the graphs that have one.

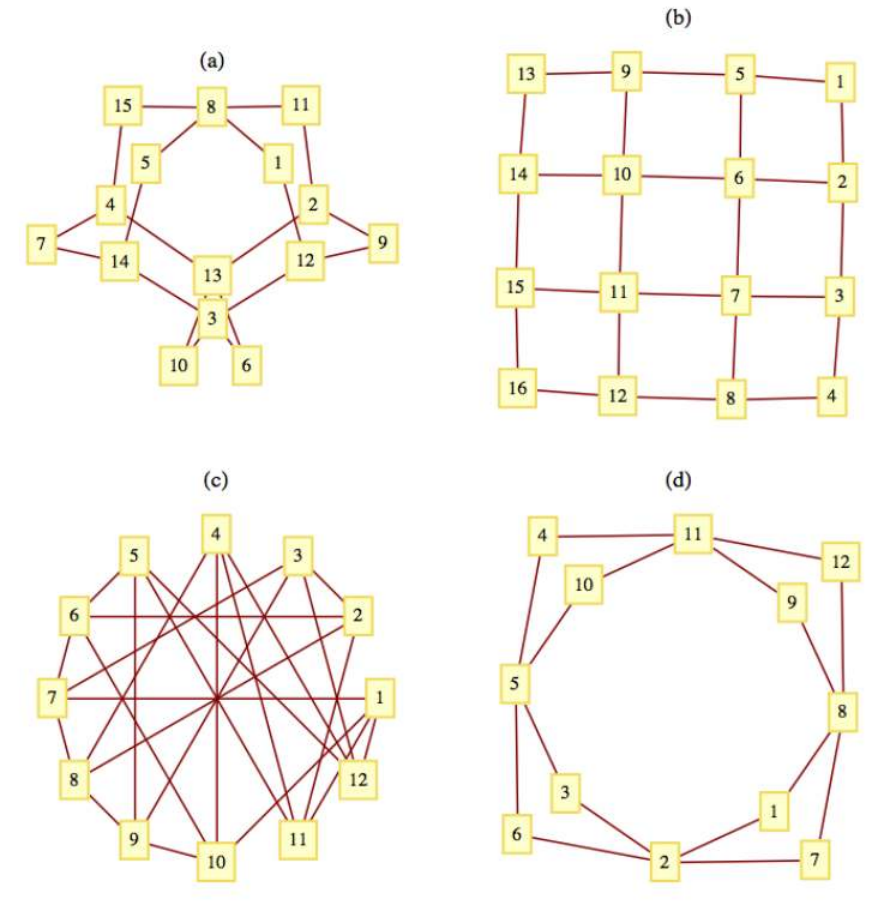


Figure 9.4.12: Graphs for Exercise 9.4.6

Exercise 9.4.7

Formulate Euler's theorem for directed graphs.

Answer

Let $G = (V, E)$ be a directed graph. G has an Eulerian circuit if and only if G is connected and $indeg(v) = outdeg(v)$ for all $v \in V$. There exists an Eulerian path from v_1 to v_2 if and only if G is connected, $indeg(v_1) = outdeg(v_1) - 1$, $indeg(v_2) = outdeg(v_2) + 1$, and for all other vertices in V the indegree and outdegree are equal.

Exercise 9.4.8

Prove that the number of vertices in an undirected graph with odd degree must be even.

Hint

Prove by induction on the number of edges.

Exercise 9.4.9

a. Under what conditions will a round-robin tournament graph be Eulerian?
b. Prove that every round-robin tournament graph is Hamiltonian.

Answer

A round-robin tournament graph is rarely Eulerian. It will be Eulerian if it has an odd number of vertices and each vertex (team) wins exactly as many times as it loses. Every round-robin tournament graph has a Hamiltonian path. This can be proven by induction on the number of vertices.

Exercise 9.4.10

For what values of n is the n-cube Eulerian?

Exercise 9.4.11

A particular set of dominoes has 21 tiles: $(1,1), (1,2), \ldots (1,6), (2,2), (2,3), \ldots, (6,6)$Is it possible to lay all 21 tiles in a line so that each adjacent pair of tile ends matches (that is, each 1 abuts a 1, and so on)?

Answer

No, such a line does not exist. The dominoes with two different numbers correspond with edges in a K_6. See corresponding dominos and edges in Figure 9.4.13 Dominos with two equal numbers could be held back and inserted into the line created with the other dominoes if such a line exists. For example, if $(2,5)$, $(5,4)$ were part of the line, $(5,5)$ could be inserted between those two dominoes. The line we want exists if and only if there exists an Eulerian path in a K_6. Since all six vertices of a K_6 have odd degree no such path exists.

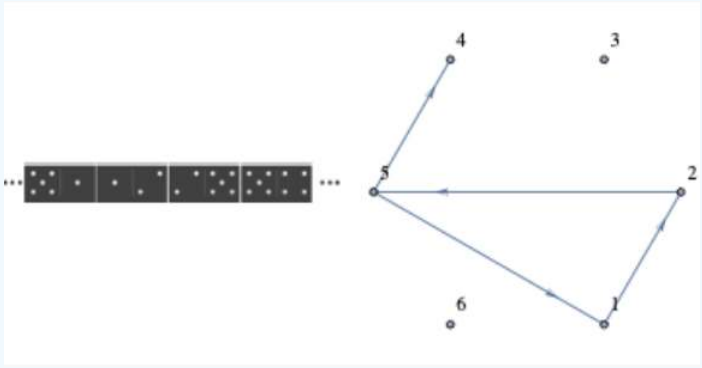


Figure 9.4.13 Correspondence between a line of dominos and a path in K_6

This page titled 9.4: Traversals- Eulerian and Hamiltonian Graphs is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

9.5: Graph Optimization

The common thread that connects all of the problems in this section is the desire to optimize (maximize or minimize) a quantity that is associated with a graph. We will concentrate most of our attention on two of these problems, the Traveling Salesman Problem and the Maximum Flow Problem. At the close of this section, we will discuss some other common optimization problems.

9.5.1: Weighted Graphs

Definition 9.5.1: Weighted Graph

A weighted graph, (V, E, w), is a graph (V, E) together with a weight function $w : E \to \mathbb{R}$. If $e \in E$, $w(e)$ is the weight on edge e.

As you will see in our examples, $w(e)$ is often a cost associated with the edge e; therefore, most weights will be positive.

Example 9.5.1: A Distance Graph

Let V be the set of six capital cities in New England: Boston, Augusta, Hartford, Providence, Concord, and Montpelier. Let E be $\{\{a, b\} \in V \times V \mid a \neq b\}$; that is, (V, E) is a complete unordered graph. An example of a weight function on this graph is $w(c_1, c_2) =$ the distance, in miles, from c_1 to c_2.

Many road maps define distance functions as in the following table.

Table 9.5.1: Distances between capital cities in New England

--	Augusta	Boston	Concord	Hartford	Montpelier	Providence
Augusta, ME	--	165	148	266	190	208
Boston, MA	165	--	75	103	192	43
Concord, NH	148	75	--	142	117	109
Hartford, CT	266	103	142	--	204	70
Montpelier, VT	190	192	117	204	--	223
Providence, RI	208	43	109	70	223	--

9.5.2: Traveling Salesman Problem

The Traveling Salesman Problem is, given a weighted graph, to find a circuit $(e_1, e_2, \ldots, e_n)$ that visits every vertex at least once and minimizes the sum of the weights, $\sum_{i=1}^{n} w(e_i)$. Any such circuit is called an optimal path.

Some statements of the Traveling Salesman Problem require that the circuit be Hamiltonian. In many applications, the graph in question will be complete and this restriction presents no problem. If the weight on each edge is constant, for example, $w(e) = 1$, then an optimal path would be any Hamiltonian circuit.

Example 9.5.2: The Problem of a Boston Salesman

The Traveling Salesman Problem gets its name from the situation of a salesman who wants to minimize the number of miles that he travels in visiting his customers. For example, if a salesman from Boston must visit the other capital cities of New England, then the problem is to find a circuit in the weighted graph of Example 9.5.1. Note that distance and cost are clearly related in this case. In addition, tolls and traffic congestion might also be taken into account.

The search for an efficient algorithm that solves the Traveling Salesman has occupied researchers for years. If the graph in question is complete, there are $(n-1)!$ different circuits. As n gets large, it is impossible to check every possible circuit. The most efficient algorithms for solving the Traveling Salesman Problem take an amount of time that is proportional to $n2^n$. Since this quantity grows so quickly, we can't expect to have the time to solve the Traveling Salesman Problem for large values of n. Most of the

useful algorithms that have been developed have to be heuristic; that is, they find a circuit that should be close to the optimal one. One such algorithm is the “closest neighbor” algorithm, one of the earliest attempts at solving the Traveling Salesman Problem. The general idea behind this algorithm is, starting at any vertex, to visit the closest neighbor to the starting point. At each vertex, the next vertex that is visited is the closest one that has not been reached. This shortsighted approach typifies heuristic algorithms called greedy algorithms, which attempt to solve a minimization (maximization) problem by minimizing (maximizing) the quantity associated with only the first step.

Algorithm 9.5.1: The Closest Neighbor Algorithm

Let $G=(V,E,w)$ be a complete weighted graph with $|V|=n$. The closest neighbor circuit through G starting at v_1 is $(v_1,v_2,\ldots,v_n)$, defined by the steps:

1. $V_1=V-\{v_1\}$.
2. For $k=2$ to $n-1$
   a. $v_k=$ the closest vertex in V_{k-1} to v_{k-1}:

$$w(v_{k-1},v_k)=\min\left(w(v_{k-1},v)\mid v\in V_{k-1}\right)$$

   In case of a tie for closest, v_k may be chosen arbitrarily.
   b. $V_k=V_{k-1}-\{v_k\}$
3. $v_n=$ the only element of V_n

The cost of the closest neighbor circuit is $\sum_{k=1}^{n-1} w(v_k,v_{k+1})+w(v_n,v_1)$

Example 9.5.3: A Small Example

The closest neighbor circuit starting at A in Figure 9.5.1 is $(1,3,2,4,1)$, with a cost of 29. The optimal path is $(1,2,3,4,1)$, with a cost of 27.

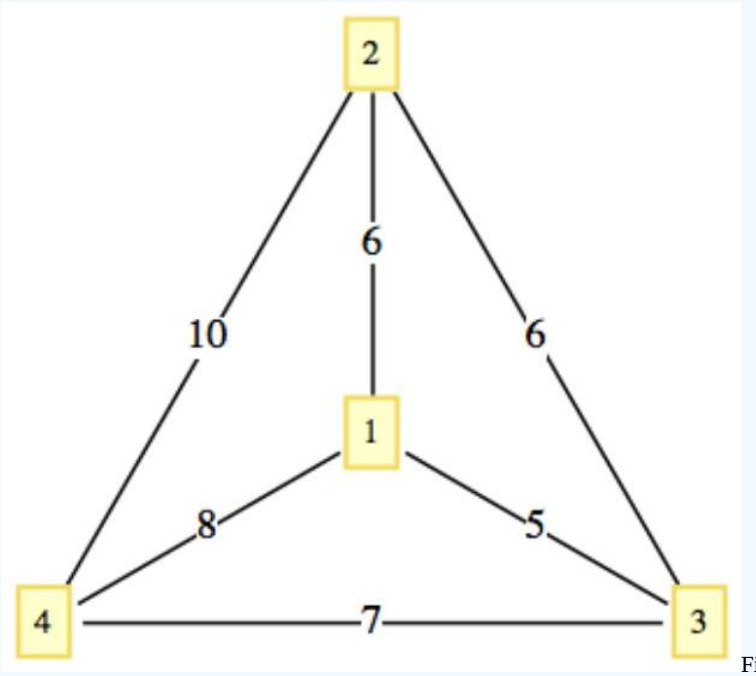


Figure 9.5.1: A small example

Although the closest neighbor circuit is often not optimal, we may be satisfied if it is close to optimal. If C_{opt} and C_{cn} are the costs of optimal and closest neighbor circuits in a graph, then it is always the case that $C_{opt}\le C_{cn}$ or $\frac{C_{cn}}{C_{opt}}\ge 1$. We can assess how good the closest neighbor algorithm is by determining how small the quantity $\frac{C_{cn}}{C_{opt}}$ gets. If it is always near 1, then the algorithm is good.

However, if there are graphs for which it is large, then the algorithm may be discarded. Note that in Example $9.5.3$, $\frac{C_{cn}}{C_{opt}} = \frac{29}{27} \approx 1.074$. A 7 percent increase in cost may or may not be considered significant, depending on the situation.

Example 9.5.4: The One-Way Street

A salesman must make stops at vertices A, B, and C, which are all on the same one-way street. The graph in Figure $9.5.2$ is weighted by the function $w(i, j)$ equal to the time it takes to drive from vertex i to vertex j.

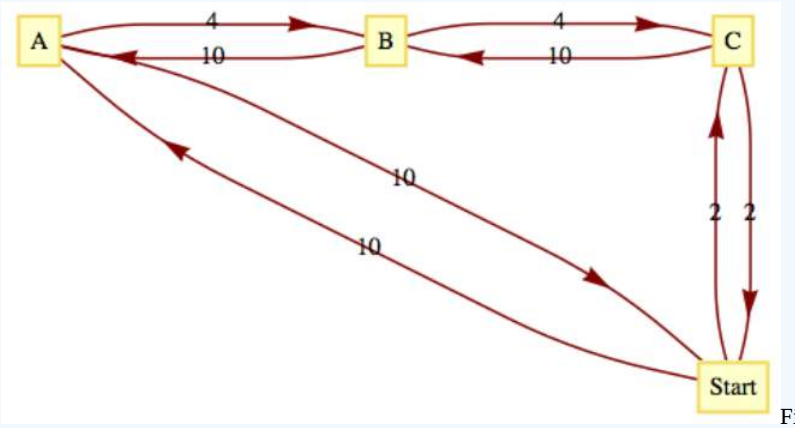


Figure $9.5.2$: Traveling a one-way street

Note that if j is down the one-way street from i, then $w(i, j) < w(j, i)$. The values of C_{opt}, and C_{cn} are 20 and 32, respectively. Verify that C_{cn} is 32 by using the closest neighbor algorithm. The value of $\frac{C_{cn}}{C_{opt}} = 1.6$ is significant in this case since our salesman would spend 60 percent more time on the road if he used the closest neighbor algorithm.

A more general result relating to the closest neighbor algorithm presumes that the graph in question is complete and that the weight function satisfies the conditions

- $w(x, y) = w(y, x)$ for all x, y in the vertex set, and
- $w(x, y) + w(y, z) \geq w(x, z)$ for all x, y, z in the vertex set.

The first condition is called the symmetry condition and the second is the triangle inequality.

Theorem 9.5.1

If (V, E, w) is a complete weighted graph that satisfies the symmetry and triangle inequality conditions, then

$$\frac{C_{cn}}{C_{opt}} \leq \frac{\lceil \log_2(2n) \rceil}{2}$$

Observation 9.5.1

If $|V| = 8$, then this theorem says that C_{cn} can be no larger than twice the size of C_{opt}; however, it doesn't say that the closest neighbor circuit will necessarily be that far from an optimal circuit. The quantity $\frac{\lceil \log_2(2n) \rceil}{2}$ is called an upper bound for the ratio $\frac{C_{cn}}{C_{opt}}$. It tells us only that things can't be any worse than the upper bound. Certainly, there are many graphs with eight vertices such that the optimal and closest neighbor circuits are the same. What is left unstated in this theorem is whether there are graphs for which the quantities are equal. If there are such graphs, we say that the upper bound is sharp.

The value of $\frac{C_{cn}}{C_{opt}}$ in Example Example $9.5.4$ is 1.6, which is greater than $\frac{\lceil \log_2(2 \cdot 4) \rceil}{2} = 1.5$; however, the weight function in this example does not satisfy the conditions of the theorem.

Example 9.5.5: The Unit Square Problem

Suppose a robot is programmed to weld joints on square metal plates. Each plate must be welded at prescribed points on the square. To minimize the time it takes to complete the job, the total distance that a robot's arm moves should be minimized. Let $d(P,Q)$ be the distance between P and Q. Assume that before each plate can be welded, the arm must be positioned at a certain point P_0 . Given a list of n points, we want to put them in order so that

$$d(P_0,P_1)+d(P_1,P_2)+\cdots+d(P_{n-1},P_n)+d(P_n,P_0)$$

is as small as possible.

The type of problem that is outlined in the example above is of such importance that it is one of the most studied version of the Traveling Salesman Problem. What follows is the usual statement of the problem. Let $[0,1]=\{x\in\mathbb{R}\mid\ 0\le x\le 1\}$, and let $S=[0,1]^2$, the unit square. Given n pairs of real numbers $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ in S that represent the n vertices of a K_n, find a circuit of the graph that minimizes the sum of the distances traveled in traversing the circuit.

Since the problem calls for a circuit, it doesn't matter which vertex we start at; assume that we will start at (x_1,y_1). Once the problem is solved, we can always change our starting position. A function can most efficiently describe a circuit in this problem. Every bijection $f:\{1,\ldots,n\}\to\{1,\ldots,n\}$ with $f(1)=1$ describes a circuit

$$(x_1,y_1),\left(x_{f(2)},y_{f(2)}\right),\ldots,\left(x_{f(n)},y_{f(n)}\right)$$

There are $(n-1)!$ such bijections. Since a circuit and its reversal have the same associated cost, there are $\frac{(n-1)!}{2}$ cases to consider. An examination of all possible cases is not feasible for large values of n.

One popular heuristic algorithm is the strip algorithm:

Heuristic Algorithm 9.5.2: The Strip Algorithm

Given n points in the unit square:

Phase 1:

1. Divide the square into $\left\lceil\sqrt{n/2}\right\rceil$ vertical strips, as in Figure 9.5.3. Let d be the width of each strip. If a point lies on a boundary between two strips, consider it part of the left-hand strip.
2. Starting from the left, find the first strip that contains one of the points. Locate the starting point by selecting the first point that is encountered in that strip as you travel from bottom to top. We will assume that the first point is (x_1,y_1)
3. Alternate traveling up and down the strips that contain vertices until all of the vertices have been reached.
4. Return to the starting point.

Phase 2:

1. Shift all strips $d/2$ units to the right (creating a small strip on the left).
2. Repeat Steps 1.2 through 1.4 of Phase 1 with the new strips.

When the two phases are complete, choose the shorter of the two circuits obtained.

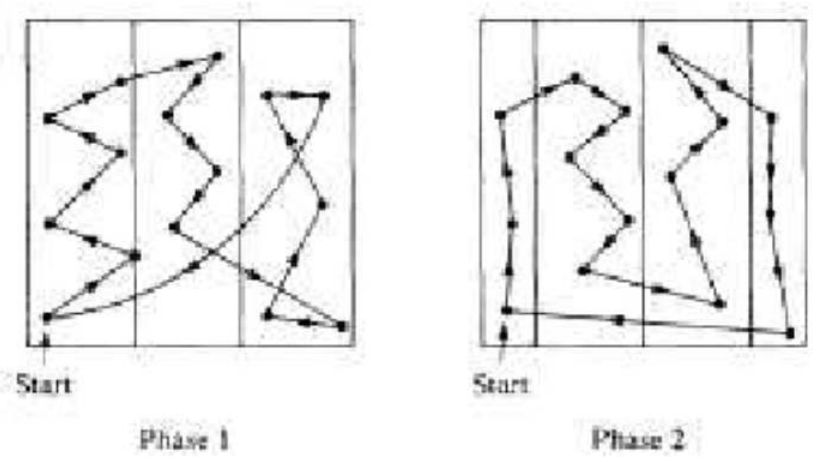


Figure 9.5.3: The Strip Algorithm

Step Item 3 may need a bit more explanation. How do you travel up or down a strip? In most cases, the vertices in a strip will be vertically distributed so that the order in which they are visited is obvious. In some cases, however, the order might not be clear, as in the third strip in Phase I of Figure 9.5.3. Within a strip, the order in which you visit the points (if you are going up the strip) is determined thusly: (x_i, y_i) precedes (x_j, y_j) if $y_i < y_j$ or if $y_i = y_j$ and $x_i < x_j$. In traveling down a strip, replace $y_i < y_j$ with $y_i > y_j$.

The selection of $\lceil \sqrt{n/2} \rceil$ strips was made in a 1959 paper by Beardwood, Halton, and Hammersley. It balances the problems that arise if the number of strips is too small or too large. If the square is divided into too few strips, some strips may be packed with vertices so that visiting them would require excessive horizontal motion. If too many strips are used, excessive vertical motion tends to be the result. An update on what is known about this algorithm is contained in [39].

Since the construction of a circuit in the square consists of sorting the given points, it should come as no surprise that the strip algorithm requires a time that is roughly a multiple of $n \log n$ time units when n points are to be visited.

The worst case that has been encountered with this algorithm is one in which the circuit obtained has a total distance of approximately $\sqrt{2n}$ (see Sopowit et al.).

9.5.3: Networks and the Maximum Flow Problem

Definition 9.5.2: Network

A network is a simple weighted directed graph that contains two distinguished vertices called the source and the sink with the properties that the indegree of the source and outdegree of the sink are both zero, and source is connected to sink. The weight function on a network is the capacity function, which has positive weights.

An example of a real situation that can be represented by a network is a city's water system. A reservoir would be the source, while a distribution point in the city to all of the users would be the sink. The system of pumps and pipes that carries the water from source to sink makes up the remaining network. We can assume that the water that passes through a pipe in one minute is

controlled by a pump and the maximum rate is determined by the size of the pipe and the strength of the pump. This maximum rate of flow through a pipe is called its capacity and is the information that the weight function of a network contains.

Example 9.5.6: A City Water System

Consider the system that is illustrated in Figure 9.5.4. The numbers that appear next to each pipe indicate the capacity of that pipe in thousands of gallons per minute. This map can be drawn in the form of a network, as in Figure 9.5.5.

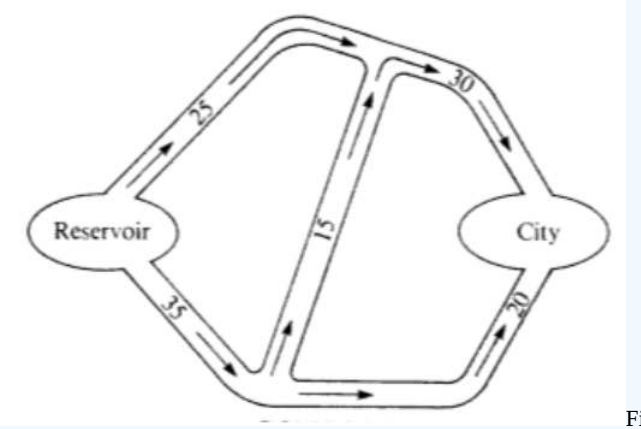


Figure 9.5.4: City Water System

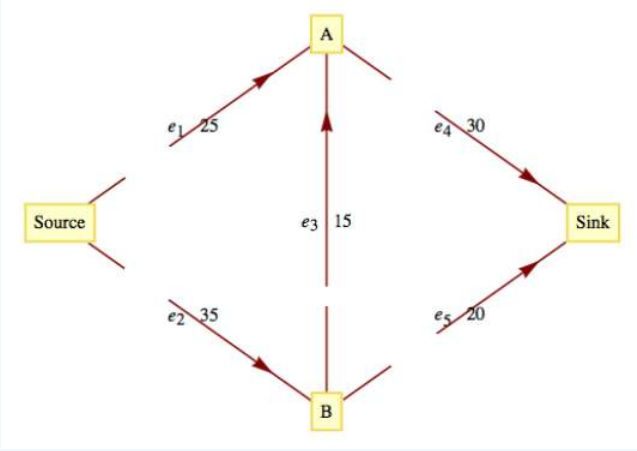


Figure 9.5.5: Flow Diagram for a City's Water Network

Although the material passing through this network is water, networks can also represent the flow of other materials, such as automobiles, electricity, bits, telephone calls, or patients in a health system.

Problem 9.5.1: The Maximum Flow Problem

The Maximum Flow Problem is derived from the objective of moving the maximum amount of water or other material from the source to the sink. To measure this amount, we define a flow as a function $f : E \to \mathbb{R}$ such that (1) the flow of material through any edge is nonnegative and no larger than its capacity: $0 \leq f(e) \leq w(e),$ for all $e \in E;$ and (2) for each vertex other than the source and sink, the total amount of material that is directed into a vertex is equal to the total amount that is directed out:

$$\begin{array}{rcl} \sum_{(x,v)\in E} f(x,v) & = & \sum_{(v,y)\in E} f(v,y) \\ \text{Flow into } v & = & \text{Flow out of } v \end{array} \tag{9.5.1}$$

The summation on the left of (9.5.1) represents the sum of the flows through each edge in E that has v as a terminal vertex. The right-hand side indicates that you should add all of the flows through edges that initiate at v.

Theorem 9.5.2: Flow out of Source equals Flow in Sink

If f is a flow, then $\sum_{(\text{source},v)\in E} f(\text{source},v) = \sum_{(v,\text{sink})\in E} f(v,\text{sink})$

Proof

Subtract the right-hand side of (9.5.1) from the left-hand side. The result is:

$$\text{Flow into } v - \text{ Flow out of } v = 0$$

Now sum up these differences for each vertex in $V' = V - \{\text{source}, \text{sink}\}$. The result is

$$\sum_{v\in V'} \left(\sum_{(x,v)\in E} f(x,v) - \sum_{(v,y)\in E} f(v,y) \right) = 0 \tag{9.5.2}$$

Now observe that if an edge connects two vertices in V', its flow appears as both a positive and a negative term in (9.5.2). This means that the only positive terms that are not cancelled out are the flows into the sink. In addition, the only negative terms that remain are the flows out of the source. Therefore,

$$\sum_{(v,\text{sink})\in E} f(v,\text{sink}) - \sum_{(\text{source},v)\in E} f(\text{source},v) = 0$$

Definition 9.5.3: The Value of a Flow

The two values ***flow into the sink*** and ***flow out of the source*** were proved to be equal in Theorem 9.5.2 and this common value is called the ***value of the flow***. It is denoted by $V(f)$. The value of a flow represents the amount of material that passes through the network with that flow.

Since the Maximum Flow Problem consists of maximizing the amount of material that passes through a given network, it is equivalent to finding a flow with the largest possible value. Any such flow is called a ***maximal flow***.

For the network in Figure 9.5.5, one flow is f_1, defined by $f_1(e_1) = 25$, $f_1(e_2) = 20$, $f_1(e_3) = 0$, $f_1(e_4) = 25$, and $f_1(e_5) = 20$. The value of f_1, $V(f_1)$, is 45. Since the total flow into the sink can be no larger than 50 ($w(e_4) + w(e_5) = 30 + 20$), we can tell that f_1 is not very far from the solution. Can you improve on f_1 at all? The sum of the capacities into the sink can't always be obtained by a flow. The same is true for the sum of the capacities out of the source. In this case, the sum of the capacities out of the source is 60, which obviously can't be reached in this network.

A solution of the Maximum Flow Problem for this network is the maximal flow f_2, where $f_2(e_1) = 25$, $f_2(e_2) = 25$, $f_2(e_3) = 5$, $f_2(e_4) = 30$, and $f_2(e_5) = 20$, with $V(f_2) = 50$. This solution is not unique. In fact, there is an infinite number of maximal flows for this problem.

There have been several algorithms developed to solve the Maximal Flow Problem. One of these is the Ford and Fulkerson Algorithm (FFA). The FFA consists of repeatedly finding paths in a network called flow augmenting paths until no improvement can be made in the flow that has been obtained.

Definition 9.5.4: Flow Augmenting Path

Given a flow f in a network (V, E), a flow augmenting path with respect to f is a simple path from the source to the sink using edges both in their forward and their reverse directions such that for each edge e in the path, $w(e) - f(e) > 0$ if e is used in its forward direction and $f(e) > 0$ if e is used in the reverse direction.

Example 9.5.7: Augmenting City Water Flow

For f_1 in Figure 9.5.5, a flow augmenting path would be (e_2, e_3, e_4) since $w(e_2) - f_1(e_2) = 15$, $w(e_3) - f_1(e_3) = 5$, and $w(e_4) - f_1(e_4) = 5$.

These positive differences represent unused capacities, and the smallest value represents the amount of flow that can be added to each edge in the path. Note that by adding 5 to each edge in our path, we obtain f_2, which is maximal. If an edge with a

positive flow is used in its reverse direction, it is contributing a movement of material that is counterproductive to the objective of maximizing flow. This is why the algorithm directs us to decrease the flow through that edge.

Algorithm 9.5.3: The Ford and Fulkerson Algorithm

1. Define the flow function f_0 by $f_0(e) = 0$ for each edge $e \in E$.
2. i = 0.
3. Repeat:
   a. If possible, find a flow augmenting path with respect to f_i.
   b. If a flow augmenting path exists, then:
      i. Determine

$$d = \min\{\{w(e) - f_i(e) \mid e \text{ is used in the forward direction}\}, \{f_i(e) \mid e \text{ is used in the reverse direction}\}\}$$

      ii. Define f_{i+1} by

$$\begin{aligned} f_{i+1}(e) &= f_i(e) && \text{if } e \text{ is not part of the flow augmenting path} \\ f_{i+1}(e) &= f_i(e) + d && \text{if } e \text{ is used in the forward direction} \\ f_{i+1}(e) &= f_i(e) - d && \text{if } e \text{ is used in the reverse direction} \end{aligned}$$

      iii. $i = i + 1$

   until no flow augmenting path exists.
4. Terminate with a maximal flow f_i

List 9.5.1: Notes on the Ford and Fulkerson Algorithm

1. It should be clear that every flow augmenting path leads to a flow of increased value and that none of the capacities of the network can be violated.
2. The depth-first search should be used to find flow augmenting paths since it is far more efficient than the breadth-first search in this situation. The depth-first search differs from the breadth-first algorithm in that you sequentially visit vertices until you reach a “dead end” and then backtrack.
3. There have been networks discovered for which the FFA does not terminate in a finite number of steps. These examples all have irrational capacities. It has been proven that if all capacities are positive integers, the FFA terminates in a finite number of steps. See Ford and Fulkerson, Even, or Berge for details.
4. When you use the FFA to solve the Maximum Flow Problem by hand it is convenient to label each edge of the network with the fraction $\frac{f_i(e)}{w(e)}$.

Algorithm 9.5.4: Depth-First Search for a Flow Augmenting Path

This is a depth-first search for the Sink Initiating at the Source. Let E' be the set of directed edges that can be used in producing a flow augmenting path. Add to the network a vertex called start and the edge $(\text{start}, \text{source})$.

1. $S =$ vertex set of the network.
2. $p =$ source Move p along the edge $(start, source)$
3. while p is not equal to start or sink:
   a. if an edge in E' exists that takes you from p to another vertex in S:

$$\text{then set } p \text{ to be that next vertex and delete the edge from } E'$$

$$\text{else reassign } p \text{ to be the vertex that } p \text{ was reached from (i.e., backtrack).}$$

4. if $p =$ start:

then no flow augmenting path exists.

else $p =$ sink and you have found a flow augmenting path.

Example 9.5.8: A Flow Augmenting Path Going Against the Flow

Consider the network in Figure 9.5.6, where the current flow, f, is indicated by a labeling of the edges.

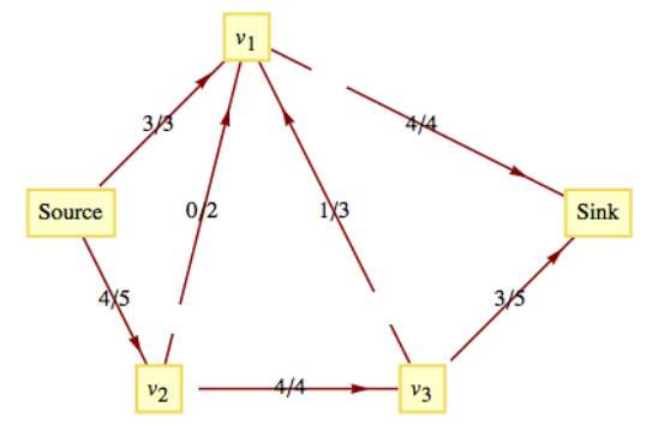


Figure 9.5.6: Current Flow

The path $(Source, v_2, v_1, v_3, Sink)$ is a flow augmenting path that allows us to increase the flow by one unit. Note that (v_1, v_3) is used in the reverse direction, which is allowed because $f(v_1, v_3) > 0$. The value of the new flow that we obtain is 8. This flow must be maximal since the capacities out of the source add up to 8. This maximal flow is defined by Figure 9.5.7.

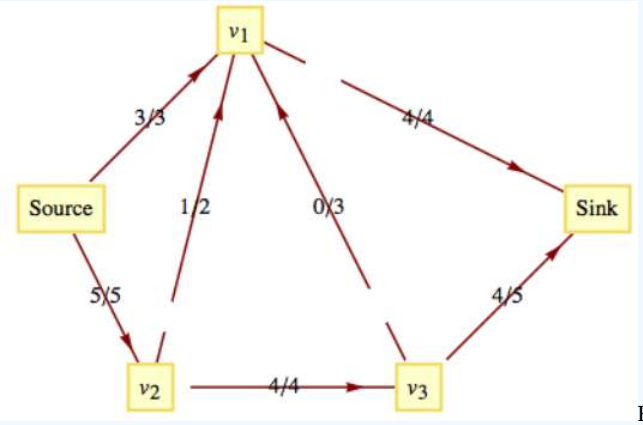


Figure 9.5.7: Updated Flow

9.5.4: Other Graph Optimization Problems

1. The Minimum Spanning Tree Problem: Given a weighted graph, (V, E, w), find a subset E' of E with the properties that (V, E') is connected and the sum of the weights of edges in E' is as small as possible. We will discuss this problem in Chapter 10.
2. The Minimum Matching Problem: Given an undirected weighted graph, (K, E, w), with an even number of vertices, pair up the vertices so that each pair is connected by an edge and the sum of these edges is as small as possible. A unit square version of this problem has been studied extensively. See [39] for details on what is known about this version of the problem.
3. The Graph Center Problem: Given a connected, undirected, weighted graph, find a vertex (called a center) in the graph with the property that the distance from the center to every other vertex is as small as possible. "As small as possible" is normally interpreted as minimizing the maximum distance from the center to a vertex.

9.5.5: Exercises

Exercise 9.5.1

Find the closest neighbor circuit through the six capitals of New England starting at Boston. If you start at a different city, will you get a different circuit?

Answer

The circuit would be Boston, Providence, Hartford, Concord, Montpelier, Augusta, Boston. It does matter where you start. If you start in Concord, for example, your mileage will be higher.

Exercise 9.5.2

Is the estimate in Theorem $9.5.1$ sharp for $n = 3$? For $n = 4$?

Exercise 9.5.3

Given the following sets of points in the unit square, find the shortest circuit that visits all the points and find the circuit that is obtained with the strip algorithm.

a. $\{(0.1k, 0.1k) : k = 0, 1, 2, \ldots, 10\}$
b. $\{(0.1, 0.3), (0.3, 0.8), (0.5, 0.3), (0.7, 0.9), (0.9, 0.1)\}$
c. $\{(0.0, 0.5), (0.5, 0.0), (0.5, 1.0), (1.0, 0.5)\}$
d. $\{(0, 0), (0.2, 0.6), (0.4, 0.1), (0.6, 0.8), (0.7, 0.5)\}$

Answer

a. Optimal cost $= 2\sqrt{2}$. Phase 1 cost $= 2.4\sqrt{2}$. Phase 2 cost $= 2.6\sqrt{2}$.
b. Optimal cost $= 2.60$. Phase 1 cost $= 3.00$. Phase 2 cost $2\sqrt{2}$.
c. $A = (0.0, 0.5), B = (0.5, 0.0), C = (0.5, 1.0), D = (1.0, 0.5)$
There are 4 points; so we will divide the unit square into two strips.
   - Optimal Path: (B, A, C, D) $\quad$ Distance $= 2\sqrt{2}$
   - Phase I Path: (B, A, C, D) $\quad$ Distance $= 2\sqrt{2}$
   - Phase II Path: (A, C, B, D) $\quad$ Distance $= 2 + \sqrt{2}$

d. $A = (0, 0), B = (0.2, 0.6), C = (0.4, 0.1), D = (0.6, 0.8), E = (0.7, 0.5)$
There are 5 points; so we will divide the unit square into three strips.
   - Optimal Path: (A, B, D, E, C) $\quad$ Distance $= 2.31$
   - Phase I Path: (A, C, B, C, E) $\quad$ Distance $= 2.57$
   - Phase II Path: (A, B, D, E, C) $\quad$ Distance $= 2.31$

Exercise 9.5.4

For $n = 4, 5,$ and 6, locate n points in the unit square for which the strip algorithm works poorly.

Exercise 9.5.5

Consider the network whose maximum capacities are shown on the following graph.

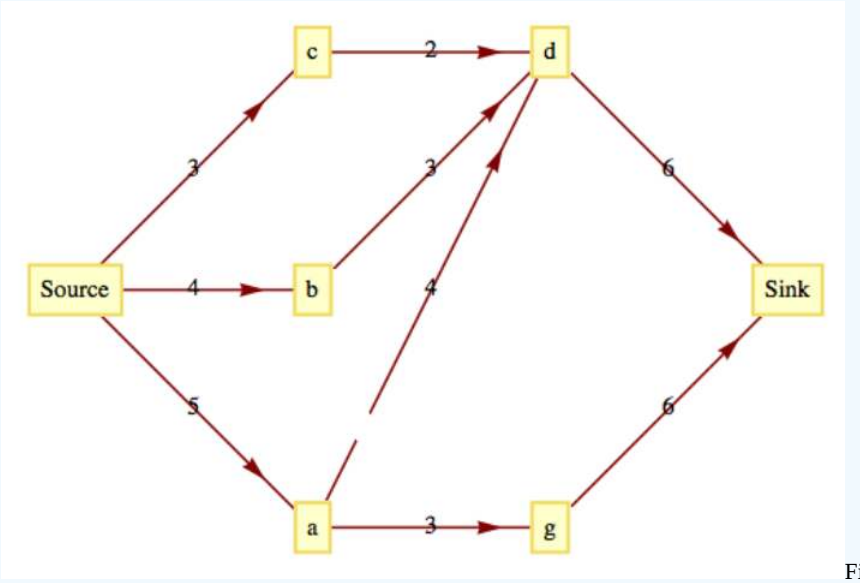


Figure 9.5.8

a. A function f is partially defined on the edges of this network by: $f(\text{Source}, c) = 2$, $f(\text{Source}, b) = 2$, $f(\text{Source}, a) = 2$, and $f(a, d) = 1$. Define f on the rest of the other edges so that f is a flow. What is the value of f ?
b. Find a flow augmenting path with respect to f for this network. What is the value of the augmented flow?
c. Is the augmented flow a maximum flow? Explain.

Answer

a. $f(c, d) = 2$, $f(b, d) = 2$, $f(d, k) = 5$, $f(a, g) = 1$, and $f(g, k) = 1$.
b. There are three possible flow-augmenting paths. s, b, d, k with flow increase of 1. s, a, d, k with flow increase of 1, and s, a, g, k with flow increase of 2.
c. The new flow is never maximal, since another flow-augmenting path will always exist. For example, if s, b, d, k is used above, the new flow can be augmented by 2 units with s, a, g, k.

Exercise 9.5.6

Given the following network with capacity function c and flow function f, find a maximal flow function. The labels on the edges of the network are of the form $f(e)/c(e)$, where $c(e)$ is the capacity of edge e and $f(e)$ is the used capacity for flow f.

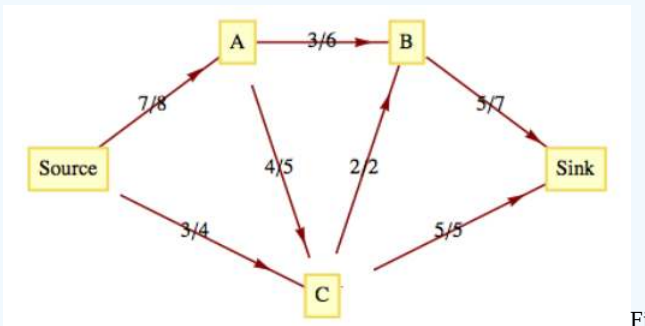


Figure 9.5.9

Exercise 9.5.7

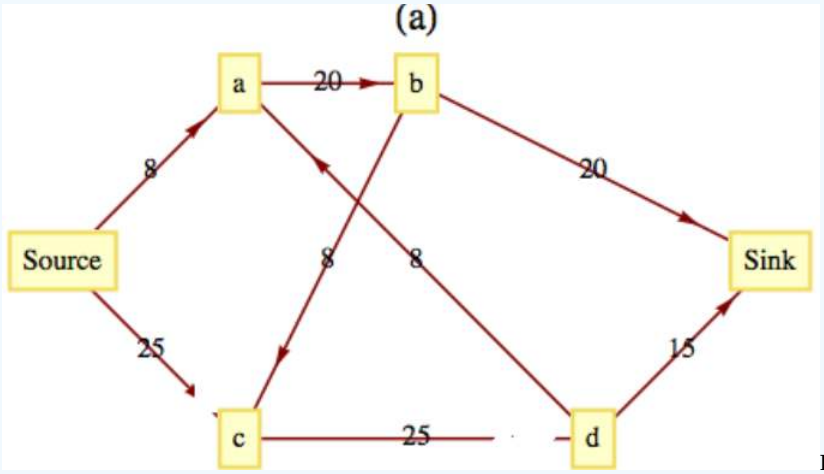


Figure 9.5.10

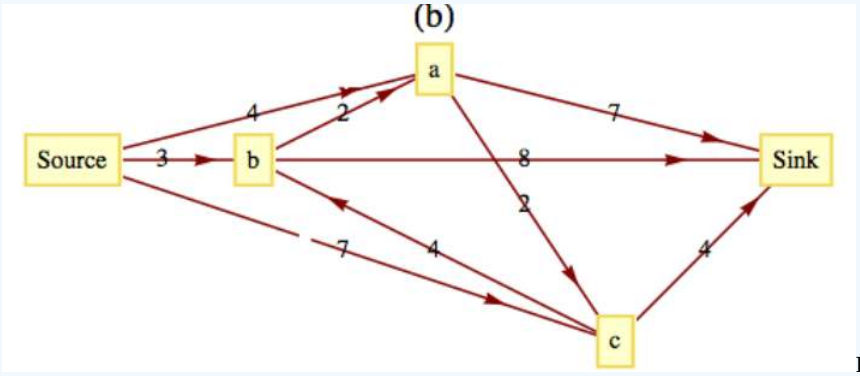


Figure 9.5.11

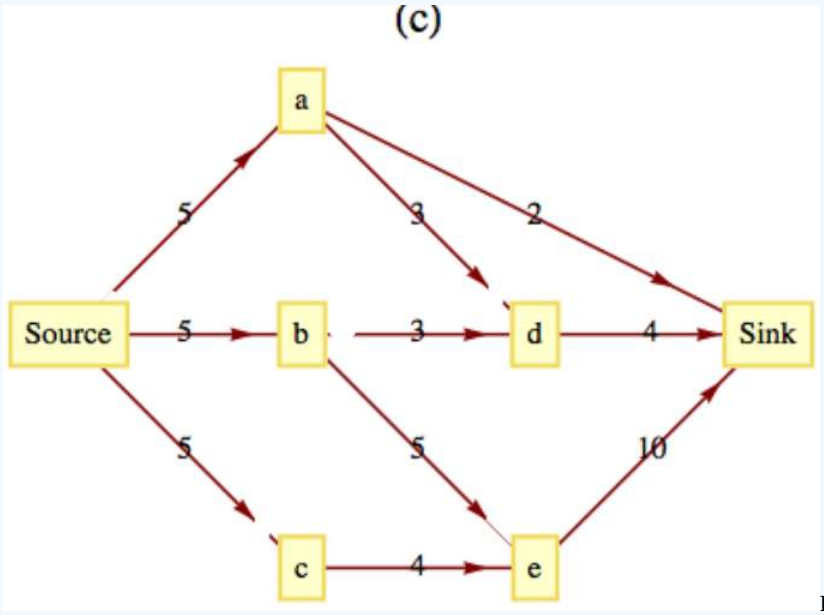


Figure 9.5.12

Answer

1. Value of maximal flow $= 31$.
2. Value of maximal flow $= 14$.
3. Value of maximal flow $= 14$. See Table $9.5.2$for one way to got this flow.

Table $9.5.2$

Step	Flow-Augmenting Path	Flow Added
1	Source, A, Sink	2
2	Source, C, B, Sink	3
3	Source, E, D, Sink	4
4	Source, A, B, Sink	1
5	Source, C, D, Sink	2
6	Source, A, B, C, D, Sink	2

Exercise 9.5.8

a. Find two maximal flows for the network in Figure $9.5.6$ other than the one found in the text.
b. Describe the set of all maximal flows for the same network.
c. Prove that if a network has two maximal flows, then it has an infinite number of maximal flows.

Exercise 9.5.9

Discuss reasons that the closest neighbor algorithm is not used in the unit square version of the Traveling Salesman Problem.

Hint

Count the number of comparisons of distances that must be done.

Answer

To locate the closest neighbor among the list of k other points on the unit square requires a time proportional to k. Therefore the time required for the closest-neighbor algorithm with n points is proportional to $(n-1)+(n-2)+\cdots+2+1,$ which is proportional to n^2. Since the strip algorithm takes a time proportional to $n(\log n)$, it is much faster for large values of n.

Exercise 9.5.10

Explore the possibility of solving the Traveling Salesman Problem in the "unit box": $[0,1]^3$.

Exercise 9.5.11

Devise a "closest neighbor" algorithm for matching points in the unit square.

This page titled 9.5: Graph Optimization is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

9.6: Planarity and Colorings

The topics in this section are related to how graphs are drawn.

Planarity: Can a given graph be drawn in a plane so that no edges intersect? Certainly, it is natural to avoid intersections, but up to now we haven't gone out of our way to do so.

Colorings: Suppose that each vertex in an undirected graph is to be colored so that no two vertices that are connected by an edge have the same color. How many colors are needed? This question is motivated by the problem of drawing a map so that no two bordering countries are colored the same. A similar question can be asked for coloring edges.

9.6.1: Planar Graphs

Definition 9.6.1: Planar Graph/Plane Graph

A graph is planar if it can be drawn in a plane so that no edges cross. If a graph is drawn so that no edges intersect, it is a plane graph, and such a drawing is a planar embedding of the graph.

Example 9.6.1: A Planar Graph

The graph in Figure 9.6.1(a) is planar but not a plane graph. The same graph is drawn as a plane graph in Figure 9.6.1(b).

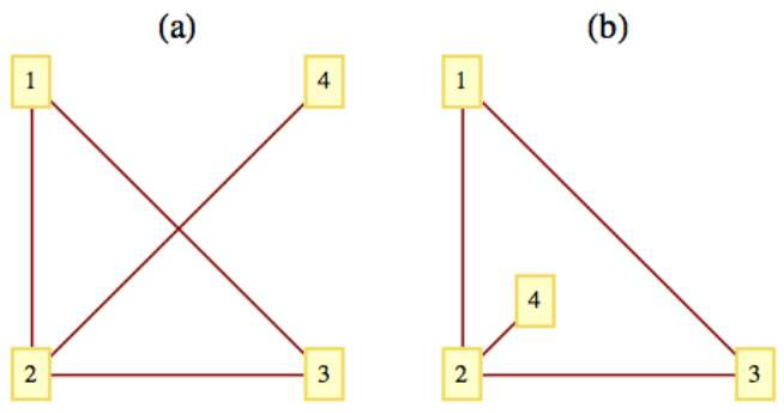


Figure 9.6.1: A Planar Graph

a. In discussing planarity, we need only consider simple undirected graphs with no self-loops. All other graphs can be treated as such since all of the edges that relate any two vertices can be considered as one “package” that clearly can be drawn in a plane.
b. Can you think of a graph that is not planar? How would you prove that it isn't planar? Proving the nonexistence of something is usually more difficult than proving its existence. This case is no exception. Intuitively, we would expect that sparse graphs would be planar and dense graphs would be nonplanar. Theorem 9.6.2 will verify that dense graphs are indeed nonplanar.
c. The topic of planarity is a result of trying to restrict a graph to two dimensions. Is there an analogous topic for three dimensions? What graphs can be drawn in one dimension?

Definition 9.6.2: Path Graph

A path graph of length n, denoted P_n, is an undirected graph with $n+1$ vertices $v_0, v_1, \ldots, v_n$ having n edges $\{v_i, v_{i+1}\}$, $i = 0, 1, \ldots, n-1$.

Observation 9.6.1: Graphs in Other Dimensions

If a graph has only a finite number of vertices, it can always be drawn in three dimensions with no edge crossings. Is this also true for all graphs with an infinite number of vertices? The only “one-dimensional” graphs are graphs consisting of a single vertex, and path graphs, as shown in Figure 9.6.2.



Figure 9.6.2: One dimensional graphs

A discussion of planarity is not complete without mentioning the famous Three Utilities Puzzle. The object of the puzzle is to supply three houses, A, B, and C, with the three utilities, gas, electric, and water. The constraint that makes this puzzle impossible to solve is that no utility lines may intersect. There is no planar embedding of the graph in Figure 9.6.3, which is commonly denoted $K_{3,3}$. This graph is one of two fundamental nonplanar graphs. The Kuratowski Reduction Theorem states that if a graph is nonplanar then it "contains" either a $K_{3,3}$ or a K_5. Containment is in the sense that if you start with a nonplanar graph you can always perform a sequence of edge deletions and contractions (shrinking an edge so that the two vertices connecting it coincide) to produce one of the two graphs.

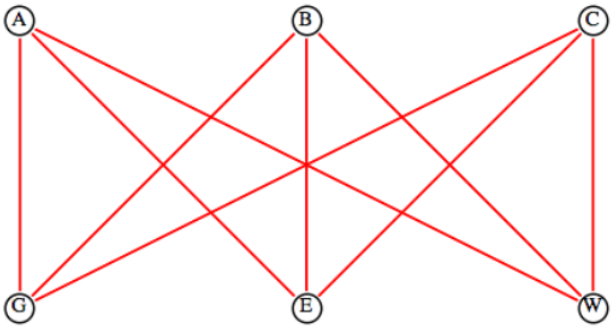


Figure 9.6.3: The Three Utilities Puzzle

A planar graph divides the plane into one or more regions. Two points on the plane lie in the same region if you can draw a curve connecting the two points that does not pass through an edge. One of these regions will be of infinite area. Each point on the plane is either a vertex, a point on an edge, or a point in a region. A remarkable fact about the geography of planar graphs is the following theorem that is attributed to Euler.

Activity 9.6.1

a. Experiment: Jot down a graph right now and count the number of vertices, regions, and edges that you have. If $v+r-e$ is not 2, then your graph is either nonplanar or not connected.

Theorem 9.6.1: Euler's Formula

If $G=(V,E)$ is a connected planar graph with r regions, v vertices, and e edges, then

$$v+r-e=2 \tag{9.6.1}$$

Proof

We prove Euler's Formula by Induction on e, for $e \geq 0$.

Basis: If $e=0$, then G must be a graph with one vertex, $v=1$; and there is one infinite region, $r=1$. Therefore, $v+r-e=1+1-0=2$, and the basis is true.

Induction: Suppose that G has k edges, $k \geq 1$, and that all connected planar graphs with less than k edges satisfy (9.6.1). Select any edge that is part of the boundary of the infinite region and call it e_1. Let G' be the graph obtained from G by deleting e_1. Figure 9.6.4illustrates the two different possibilities we need to consider: either G' is connected or it has two connected components, G_1 and G_2.

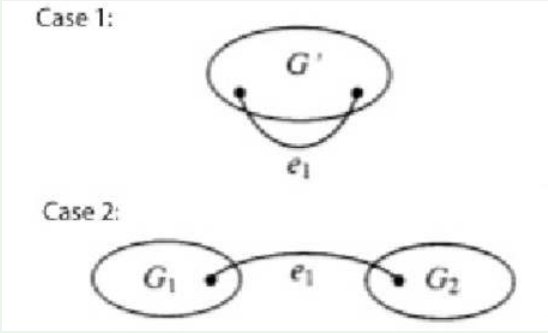


Figure 9.6.4: Two cases in the proof of Euler's Formula

If G' is connected, the induction hypothesis can be applied to it. If G' has v' vertices, r' edges and e' edges, then $v'+r'-e'=2$ and in terms of the corresponding numbers for G,

$v'=v$	No vertices were removed to form G'
$r'=r-1$	One region of G was merged with the infinite region when e_1 was removed
$e'=k-1$	We assumed that G had k edges.

For the case where G' is connected,

$$\begin{aligned} v+r-e &= v+r-k \\ &= v'+(r'+1)-(e'+1) \\ &= v'+r'-e' \\ &= 2 \end{aligned}$$

If G' is not connected, it must consist of two connected components, G_1 and G_2, since we started with a connected graph, G. We can apply the induction hypothesis to each of the two components to complete the proof. We leave it to the students to do this, with the reminder that in counting regions, G_1 and G_2 will share the same infinite region.

Theorem 9.6.2: A Bound on Edges of a Planar Graph

If $G=(V,E)$ is a connected planar graph with v vertices, $v\geq 3$, and e edges, then

$$e\leq 3v-6 \tag{9.6.2}$$

Proof

(Outline of a Proof)

a. Let r be the number of regions in G. For each region, count the number of edges that comprise its border. The sum of these counts must be at least $3r$. Recall that we are working with simple graphs here, so a region made by two edges connecting the same two vertices is not possible.
b. Based on (a), infer that the number of edges in G must be at least $\frac{3r}{2}$.
c. $e\geq \dfrac{3r}{2} \Rightarrow r\leq \dfrac{2e}{3}$
d. Substitute $\frac{2e}{3}$ for r in Euler's Formula to obtain an inequality that is equivalent to (9.6.2)

Remark 9.6.1

One implication of (9.6.2) is that the number of edges in a connected planar graph will never be larger than three times its number of vertices (as long as it has at least three vertices). Since the maximum number of edges in a graph with v vertices is a quadratic function of v, as v increases, planar graphs are more and more sparse.

The following theorem will be useful as we turn to graph coloring.

Theorem 9.6.3: A Vertex of Degree Five

If G is a connected planar graph, then it has a vertex with degree 5 or less.

Proof

(by contradiction): We can assume that G has at least seven vertices, for otherwise the degree of any vertex is at most 5. Suppose that G is a connected planar graph and each vertex has a degree of 6 or more. Then, since each edge contributes to the degree of two vertices, $e \geq \frac{6v}{2} = 3v$. However, Theorem 9.6.2 states that the $e \leq 3v - 6 < 3v$, which is a contradiction.

9.6.2: Graph Coloring

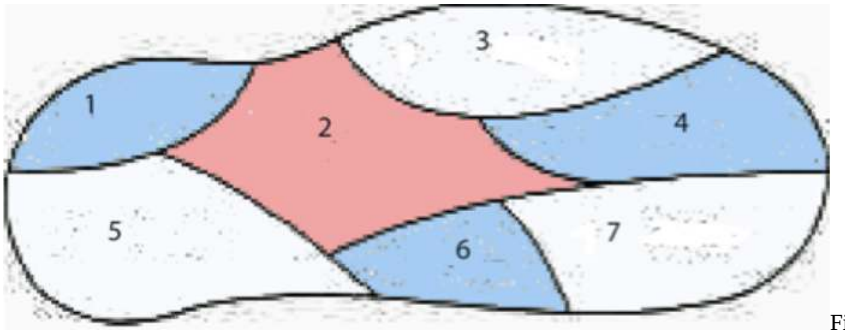


Figure 9.6.5: A 3-coloring of Euler Island

The map of Euler Island in Figure 9.6.5 shows that there are seven towns on the island. Suppose that a cartographer must produce a colored map in which no two towns that share a boundary have the same color. To keep costs down, she wants to minimize the number of different colors that appear on the map. How many colors are sufficient? For Euler Island, the answer is three. Although it might not be obvious, this is a graph problem. We can represent the map with a graph, where the vertices are countries and an edge between two vertices indicates that the two corresponding countries share a boundary of positive length. This problem motivates a more general problem.

Definition 9.6.3: Graph Coloring

Given an undirected graph $G = (V, E)$, find a "coloring function" f from V into a set of colors H such that $(v_i, v_j) \in E \Rightarrow f(v_i) \neq f(v_j)$ and H has the smallest possible cardinality. The cardinality of H is called the chromatic number of G, $\chi(G)$.

- A coloring function onto an n-element set is called an n-coloring.
- In terms of this general problem, the chromatic number of the graph of Euler Island is three. To see that no more than three colors are needed, we need only display a 3-coloring: $f(1) = f(4) = f(6) = \text{blue}$, $f(2) = \text{red}$, and $f(3) = f(5) = f(7) = \text{white}$. This coloring is not unique. The next smallest set of colors would be of two colors, and you should be able to convince yourself that no 2-coloring exists for this graph.

In the mid-nineteenth century, it became clear that the typical planar graph had a chromatic number of no more than 4. At that point, mathematicians attacked the Four-Color Conjecture, which is that if G is any planar graph, then its chromatic number is no more than 4. Although the conjecture is quite easy to state, it took over 100 years, until 1976, to prove the conjecture in the affirmative.

Theorem 9.6.4: The Four-Color Theorem

If G is a planar graph, then $\chi(G) \leq 4$.

A proof of the Four-Color Theorem is beyond the scope of this text, but we can prove a theorem that is only 25 percent inferior.

Theorem 9.6.5: The Five-Color Theorem

If G is a planar graph, then $\chi(G) \leq 5$.

Proof

The number 5 is not a sharp upper bound for $\chi(G)$ because of the Four-Color Theorem.

This is a proof by Induction on the Number of Vertices in the Graph.

Basis: Clearly, a graph with one vertex has a chromatic number of 1.

Induction: Assume that all planar graphs with $n-1$ vertices have a chromatic number of 5 or less. Let G be a planar graph. By Theorem 9.6.3 there exists a vertex v with $\deg v \leq 5$. Let $G-v$ be the planar graph obtained by deleting v and all edges that connect v to other vertices in G. By the induction hypothesis, $G-v$ has a 5-coloring. Assume that the colors used are red, white, blue, green, and yellow.

If $\deg v < 5$, then we can produce a 5-coloring of G by selecting a color that is not used in coloring the vertices that are connected to v with an edge in G.

If $\deg v = 5$, then we can use the same approach if the five vertices that are adjacent to v are not all colored differently. We are now left with the possibility that v_1, v_2, v_3, v_4, and v_5 are all connected to v by an edge and they are all colored differently. Assume that they are colored red, white blue, yellow, and green, respectively, as in Figure 9.6.6

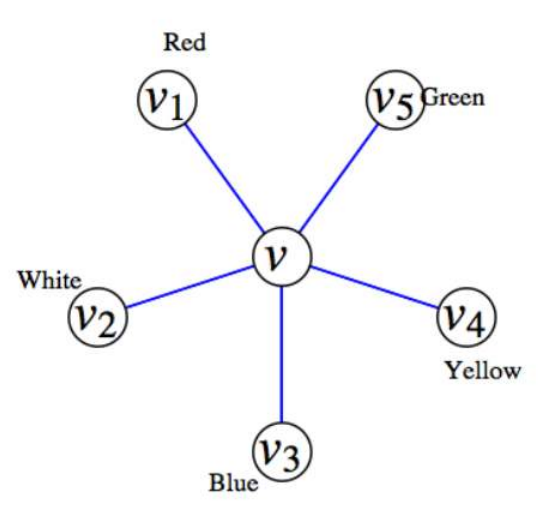


Figure 9.6.6

Starting at v_1 in $G-v$, suppose we try to construct a path v_3 that passes through only red and blue vertices. This can either be accomplished or it can't be accomplished. If it can't be done, consider all paths that start at v_1, and go through only red and blue vertices. If we exchange the colors of the vertices in these paths, including v_1 we still have a 5-coloring of $G-v$. Since v_1 is now blue, we can color the central vertex, v, red.

Finally, suppose that v_1 is connected to v_3 using only red and blue vertices. Then a path from v_1 to v_3 by using red and blue vertices followed by the edges (v_3, v) and (v, v_1) completes a circuit that either encloses v_2 or encloses v_4 and v_5. Therefore, no path from v_2 to v_4 exists using only white and yellow vertices. We can then repeat the same process as in the previous paragraph with v_2 and v_4, which will allow us to color v white.

Definition 9.6.4: Bipartite Graph

A bipartite graph is a graph that has a 2-coloring. Equivalently, a graph is bipartite if its vertices can be partitioned into two nonempty subsets so that no edge connects vertices from the same subset.

Example 9.6.2: A Few Examples

a. The graph of the Three Utilities Puzzle is bipartite. The vertices are partitioned into the utilities and the homes. Of course a 2-coloring of the graph is to color the utilities red and the homes blue.
b. For $n \geq 1$, the n-cube is bipartite. A coloring would be to color all strings with an even number of 1's red and the strings with an odd number of 1's blue. By the definition of the n-cube, two strings that have the same color couldn't be connected since they would need to differ in at least two positions.
c. Let V be a set of 64 vertices, one for each square on a chess board. We can index the elements of V by v_{ij} = the square on the row i, column j. Connect vertices in V according to whether or not you can move a knight from one square to another. Using our indexing of V, $(v_{ij}, v_{kl}) \in E$ if and only if $\begin{array}{l} |i-k|+|j-l|=3 \\ \text{and } |i-k| \cdot |j-l|=2 \end{array}$ (V, E) is a bipartite graph. The usual coloring of a chessboard is valid 2-coloring.

How can you recognize whether a graph is bipartite? Unlike planarity, there is a nice equivalent condition for a graph to be bipartite.

Theorem 9.6.6: No Odd Circuits in a Bipartite Graph

An undirected graph is bipartite if and only if it has no circuit of odd length.

Proof

($\Rightarrow$) Let $G=(V, E)$ be a bipartite graph that is partitioned into two sets, R(ed) and B(lue) that define a 2-coloring. Consider any circuit in V. If we specify a direction in the circuit and define f on the vertices of the circuit by

$$f(u) = \text{the next vertex in the circuit after } v$$

Note that f is a bijection. Hence the number of red vertices in the circuit equals the number of blue vertices, and so the length of the circuit must be even.

($\Leftarrow$) Assume that G has no circuit of odd length. For each component of G, select any vertex w and color it red. Then for every other vertex v in the component, find the path of shortest distance from w to v. If the length of the path is odd, color v blue, and if it is even, color v red. We claim that this method defines a 2-coloring of G. Suppose that it does not define a 2-coloring. Then let v_a and v_b be two vertices with identical colors that are connected with an edge. By the way that we colored G, neither v_a nor v_b could equal w. We can now construct a circuit with an odd length in G. First, we start at w and follow the shortest path to v_a . Then follow the edge (v_a, v_b) , and finally, follow the reverse of a shortest path from w to v_b. Since v_a and v_b have the same color, the first and third segments of this circuit have lengths that are both odd or even, and the sum of their lengths must be even. The addition of the single edge (v_a, v_b) shows us that this circuit has an odd length. This contradicts our premise.

9.6.3: Exercises

Exercise 9.6.1

Apply Theorem 9.6.3 to prove that once n gets to a certain size, a K_n is nonplanar. What is the largest complete planar graph?

Answer

Theorem 9.6.3can be applied to infer that if $n \geqslant 5$, then K_n is nonplanar. A K_4 is the largest complete planar graph.

Exercise 9.6.2

Can you apply Theorem 9.6.3 to prove that the Three Utilities Puzzle can't be solved?

Exercise 9.6.3

What are the chromatic numbers of the following graphs?

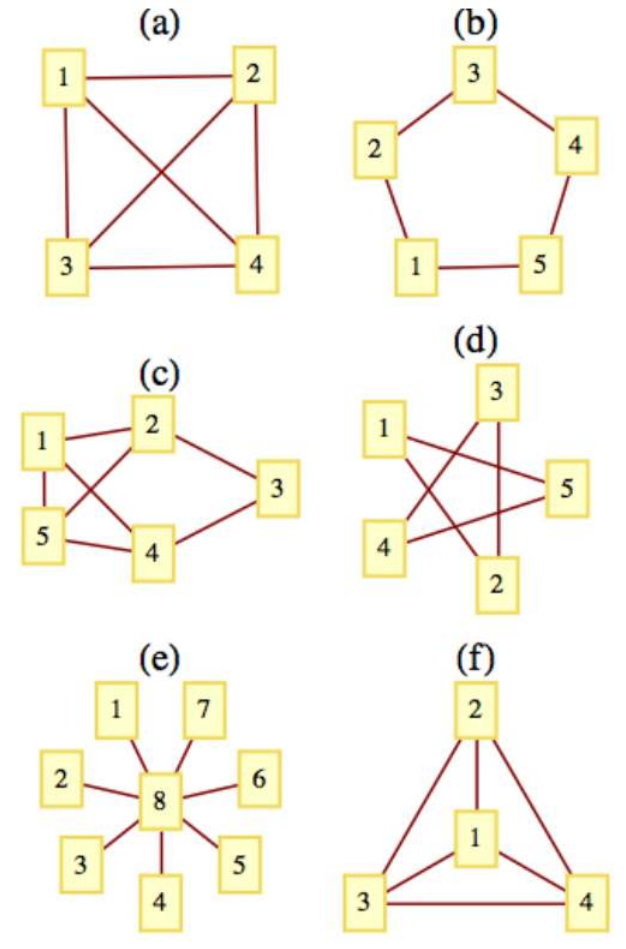


Figure 9.6.7: What are the chromatic numbers?

Answer

a. 4
b. 3
c. 3
d. 3
e. 2
f. 4

Exercise 9.6.4

Prove that if an undirected graph has a subgraph that is a K_3 it then its chromatic number is at least 3.

Exercise 9.6.5

What is $\chi(K_n)$, $n \geq 1$?

Answer

The chromatic number is n since every vertex is connected to every other vertex.

Exercise 9.6.6

What is the chromatic number of the United States?

Exercise 9.6.7

Complete the proof of Euler's Formula, Theorem 9.6.1.

Answer

Suppose that G' is not connected. Then G' is made up of 2 components that are planar graphs with less than k edges, G_1 and G_2. For $i=1,2$ let $v_i, r_i,$ ande_i be the number of vertices, regions and edges in G_i. By the induction hypothesis, $v_i + r_i - e_i = 2 \quad \text{for } i = 1,2.$

One of the regions, the infinite one, is common to both graphs. Therefore, when we add edge e back to the graph, we have $r = r_1 + r_2 - 1, \quad v = v_1 + v_2, \quad \text{and } e = e_1 + e_2 + 1.$

$$\begin{aligned} v + r - e &= (v_1 + v_2) + (r_1 + r_2 - 1) - (e_1 + e_2 + 1) \\ &= (v_1 + r_1 - e_1) + (v_2 + r_2 - e_2) - 2 \\ &= 2 + 2 - 2 \\ &= 2 \end{aligned}$$

Exercise 9.6.8

Use the outline of a proof of Theorem 9.6.2 to write a complete proof. Be sure to point out where the premise $v \geq 3$ is essential.

Exercise 9.6.9

Let $G = (V, E)$ with $|V| \geq 11$, and let U be the set of all undirected edges between distinct vertices in V. Prove that either G or $G' = (V, E^c)$ is nonplanar.

Answer

Since $|E| + E^c = \frac{n(n-1)}{2}$, either E or E^c has at least $\frac{n(n-1)}{4}$ elements. Assume that it is E that is larger. Since $\frac{n(n-1)}{4}$ is greater than $3n-6$ for $n \geqslant 11$, G would be nonplanar. Of course, if E^c is larger, then G' would be nonplanar by the same reasoning. Can you find a graph with ten vertices such that it is planar and its complement is also planar?

Exercise 9.6.10

Design an algorithm to determine whether a graph is bipartite.

Exercise 9.6.11

Prove that a bipartite graph with an odd number of vertices greater than or equal to 3 has no Hamiltonian circuit.

Answer

Suppose that (V, E) is bipartite (with colors red and blue), $|E|$ is odd, and $(v_1, v_2, \ldots, v_{2n+1}, v_1)$ is a Hamiltonian circuit. If v_1 is red, then v_{2n+1} would also be red. But then $\{v_{2n+1}, v_1\}$ would not be in E, a contradiction.

Exercise 9.6.12

Prove that any graph with a finite number of vertices can be drawn in three dimensions so that no edges intersect.

Exercise 9.6.13

Suppose you had to color the edges of an undirected graph so that for each vertex, the edges that it is connected to have different colors. How can this problem be transformed into a vertex coloring problem?

Answer

Draw a graph with one vertex for each edge, If two edges in the original graph meet at the same vertex, then draw an edge connecting the corresponding vertices in the new graph.

Exercise 9.6.14

a. Suppose the edges of a K_6 are colored either red or blue. Prove that there will be either a "red K_3" (a subset of the vertex set with three vertices connected by red edges) or a "blue K_3" or both.
b. Suppose six people are selected at random. Prove that either there exists a subset of three of them with the property that any two people in the subset can communicate in a common language, or there exist three people, no two of whom can communicate in a common language.

Exercise 9.6.15

Let d be a positive integer, and let $a_1, a_2, \ldots a_d$ be positive integers greater than or equal to two. The ***mesh graph*** $M(a_1, a_2, \ldots, a_d)$ has vertices of the form $x = (x_1, x_2, \ldots, x_d)$ where $1 \leq x_i \leq a_i$. Two vertices x and y are adjacent if and only if $\sum_{i=1}^{d} |x_i - y_i| = 1$. In other words, two adjacent vertices must differ in only one coordinate and by a difference of 1.

a. What is the chromatic number of $M(a_1, a_2, \ldots, a_d)$?
b. For what pairs (a_1, a_2) does $M(a_1, a_2)$ have a Hamiltonian circuit?
c. For what triples (a_1, a_2, a_3) does $M(a_1, a_2, a_3)$ have a Hamiltonian circuit?

9.6.4: Further Reading

[1] Wilson, R., *Four Colors Suffice - How the Map Problem Was Solved*Princeton, NJ: Princeton U. Press, 2013.

This page titled 9.6: Planarity and Colorings is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

10: Trees

In this chapter we will study the class of graphs called trees. Trees are frequently used in both mathematics and the sciences. Our solution of Example 2.1.1 is one simple instance. Since they are often used to illustrate or prove other concepts, a poor understanding of trees can be a serious handicap. For this reason, our ultimate goals are to: (1) define the various common types of trees, (2) identify some basic properties of trees, and (3) discuss some of the common applications of trees.

10.1: What is a Tree?

10.2: Spanning Trees

10.3: Rooted Trees

10.4: Binary Trees

This page titled 10: Trees is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

10.1: What is a Tree?

10.1.1: Definition

What distinguishes trees from other types of graphs is the absence of certain paths called cycles. Recall that a path is a sequence of consecutive edges in a graph, and a circuit is a path that begins and ends at the same vertex.

Definition 10.1.1: Cycle

A cycle is a circuit whose edge list contains no duplicates. It is customary to use C_n to denote a cycle with n edges.

The simplest example of a cycle in an undirected graph is a pair of vertices with two edges connecting them. Since trees are cycle-free, we can rule out all multigraphs having at least one pair of vertices connected with two or more edges from consideration as trees.

Trees can either be undirected or directed graphs. We will concentrate on the undirected variety in this chapter.

Definition 10.1.2: Tree

An undirected graph is a tree if it is connected and contains no cycles or self-loops.

Example 10.1.1: Some Trees and Non-Trees

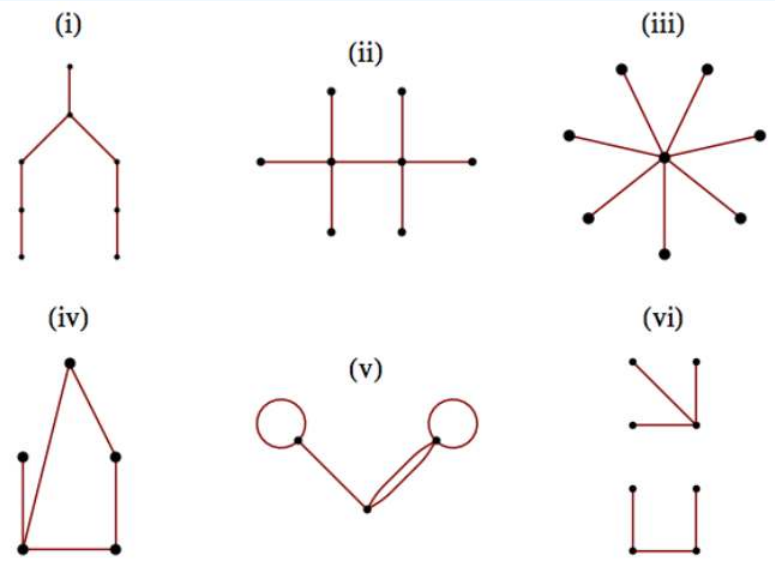


Figure 10.1.1: Some trees and some non-trees

a. Graphs i, ii and iii in Figure 10.1.1 are all trees, while graphs iv, v, and vi are not trees.
b. A K_2 is a tree. However, if $n \geq 3$, a K_n is not a tree.
c. In a loose sense, a botanical tree is a mathematical tree. There are usually no cycles in the branch structure of a botanical tree.
d. The structures of some chemical compounds are modeled by a tree. For example, butane Figure 10.1.2 consists of four carbon atoms and ten hydrogen atoms, where an edge between two atoms represents a bond between them. A bond is a force that keeps two atoms together. The same set of atoms can be linked together in a different tree structure to give us the compound isobutane Figure 10.1.3. There are some compounds whose graphs are not trees. One example is benzene Figure 10.1.4

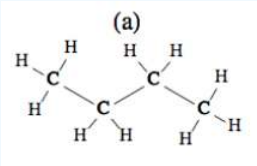


Figure 10.1.2: Butane

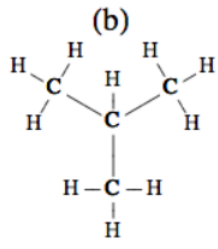


Figure 10.1.3: Isobutane

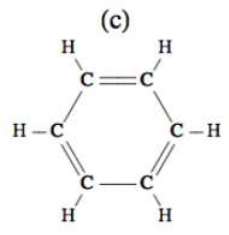


Figure 10.1.4: Benzene

One type of graph that is not a tree, but is closely related, is a forest.

Definition 10.1.3: Forest

A forest is an undirected graph whose components are all trees.

Example 10.1.2: A Forest

The top half of Figure 10.1.1 can be viewed as a forest of three trees. Graph (vi) in this figure is also a forest.

10.1.2: Conditions for a graph to be a tree

We will now examine several conditions that are equivalent to the one that defines a tree. The following theorem will be used as a tool in proving that the conditions are equivalent.

Lemma 10.1.1

Let $G = (V, E)$ be an undirected graph with no self-loops, and let $v_a, v_b \in V$. If two different simple paths exist between v_a and v_b, then there exists a cycle in G.

Proof

Let $p_1 = (e_1, e_2, \ldots, e_m)$ and $p_2 = (f_1, f_2, \ldots, f_n)$ be two different simple paths from v_a to v_b. The first step we will take is to delete from p_1 and p_2 the initial edges that are identical. That is, if $e_1 = f_1$, $e_2 = f_2, \ldots$, $e_j = f_j$, and $e_{j+1} \neq f_{j+1}$ delete the first j edges of both paths. Once this is done, both paths start at the same vertex, call it v_c, and both still end at v_b. Now we construct a cycle by starting at v_c and following what is left of p_1 until we first meet what is left of p_2. If this first meeting occurs at vertex v_d, then the remainder of the cycle is completed by following the portion of the reverse of p_2 that starts at v_d and ends at v_c.

Theorem 10.1.1: Equivalent Conditions for a Graph to be a Tree

Let $G=(V,E)$ be an undirected graph with no self-loops and $|V|=n$. The following are all equivalent:

1. G is a tree.
2. For each pair of distinct vertices in V, there exists a unique simple path between them.
3. G is connected, and if $e\in E$, then $(V,E-\{e\})$ is disconnected.
4. G contains no cycles, but by adding one edge, you create a cycle.
5. G is connected and $|E|=n-1$.

Proof

Proof Strategy. Most of this theorem can be proven by proving the following chain of implications: $(1)\Rightarrow(2)$, $(2)\Rightarrow(3)$, $(3)\Rightarrow(4)$, and $(4)\Rightarrow(1)$. Once these implications have been demonstrated, the transitive closure of $\Rightarrow$ on $1,2,3,4$ establishes the equivalence of the first four conditions. The proof that Statement 5 is equivalent to the first four can be done by induction, which we will leave to the reader.

$(1)\Rightarrow(2)$ (Indirect). Assume that G is a tree and that there exists a pair of vertices between which there is either no path or there are at least two distinct paths. Both of these possibilities contradict the premise that G is a tree. If no path exists, G is disconnected, and if two paths exist, a cycle can be obtained by Theorem 10.1.1

$(2)\Rightarrow(3)$. We now use Statement 2 as a premise. Since each pair of vertices in V are connected by exactly one path, G is connected. Now if we select any edge e in E, it connects two vertices, v_1 and v_2. By (2), there is no simple path connecting v_1 to v_2 other than e. Therefore, no path at all can exist between v_1 and v_2 in $(V,E-\{e\})$. Hence $(V,E-\{e\})$ is disconnected.

$(3)\Rightarrow(4)$. Now we will assume that Statement 3 is true. We must show that G has no cycles and that adding an edge to G creates a cycle. We will use an indirect proof for this part. Since (4) is a conjunction, by DeMorgan's Law its negation is a disjunction and we must consider two cases. First, suppose that G has a cycle. Then the deletion of any edge in the cycle keeps the graph connected, which contradicts (3). The second case is that the addition of an edge to G does not create a cycle. Then there are two distinct paths between the vertices that the new edge connects. By Lemma 10.1.1, a cycle can then be created, which is a contradiction.

$(4)\Rightarrow(1)$ Assume that G contains no cycles and that the addition of an edge creates a cycle. All that we need to prove to verify that G is a tree is that G is connected. If it is not connected, then select any two vertices that are not connected. If we add an edge to connect them, the fact that a cycle is created implies that a second path between the two vertices can be found which is in the original graph, which is a contradiction.

The usual definition of a directed tree is based on whether the associated undirected graph, which is created by "erasing" its directional arrows, is a tree. In Section 10.3 we will introduce the rooted tree, which is a special type of directed tree.

10.1.3: Exercises

Exercise 10.1.1

Given the following vertex sets, draw all possible undirected trees that connect them.

a. $V_a=\{\text{right},\text{left}\}$
b. $V_b=\{+,-,0\}$
c. $V_c=\{\text{north},\text{south},\text{east},\text{west}\}$.

Answer

The number of trees are: (a) 1, (b) 3, and (c) 16. The trees that connect V_c are:

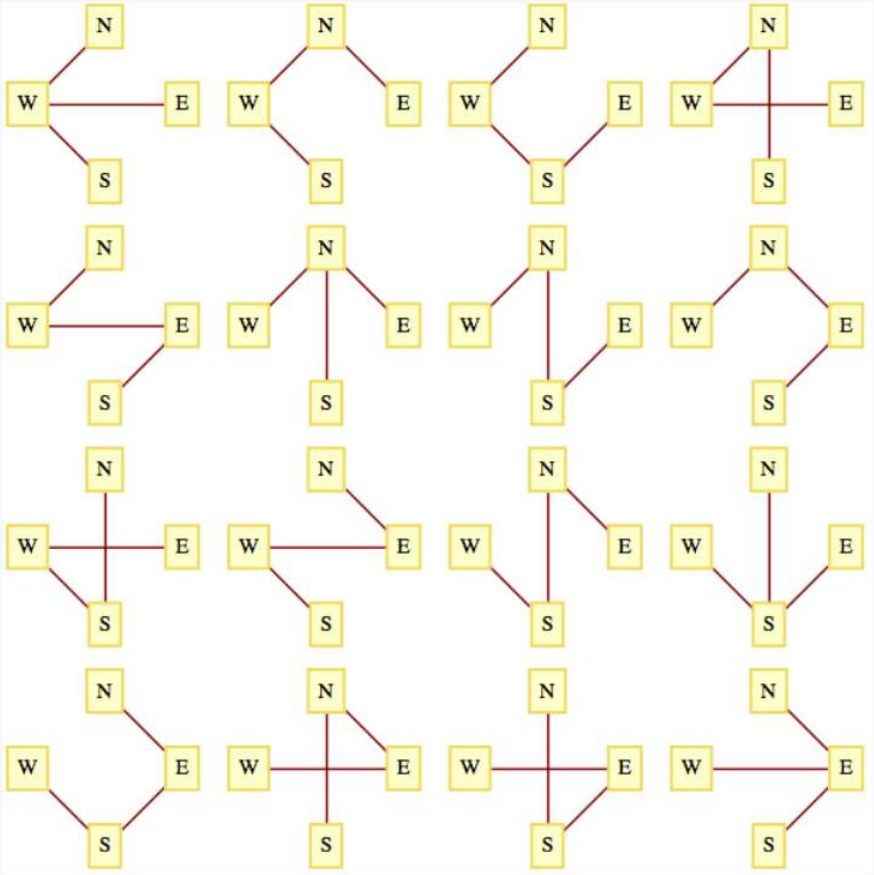


Figure 10.1.5

Exercise 10.1.2

Are all trees planar? If they are, can you explain why? If they are not, you should be able to find a nonplanar tree.

Exercise 10.1.3

Prove that if G is a simple undirected graph with no self-loops, then G is a tree if and only if G is connected and $|E| = |V| - 1$.

Hint

Use induction on $|E|$.

Exercise 10.1.4

a. Prove that if $G = (V, E)$ is a tree and $e \in E$, then $(V, E - \{e\})$ is a forest of two trees.
b. Prove that if (V_1, E_1) and (V_2, E_2) are disjoint trees and e is an edge that connects a vertex in V_1 to a vertex in V_2, then $(V_1 \cup V_2, E_1 \cup E_2 \cup \{e\})$ is a tree.

Exercise 10.1.5

a. Prove that any tree with at least two vertices has at least two vertices of degree 1.
b. Prove that if a tree has n vertices, $n \geq 4$, and is not a path graph, P_n, then it has at least three vertices of degree 1.

Answer

a. Assume that (V, E) is a tree with $|V| \geq 2$, and all but possibly one vertex in V has degree two or more.

$$\begin{aligned} 2|E| = \sum_{v \in V} \deg(v) \geq 2|V| - 1 \;\; &\Rightarrow |E| \geq |V| - \frac{1}{2} \\ &\Rightarrow |E| \geq |V| \\ &\Rightarrow (V, E) \text{ is not a tree.} \end{aligned}$$

b. The proof of this part is similar to part a in that we can infer $2|E| \geq 2|V| - 1$, using the fact that a non-chain tree has at least one vertex of degree three or more.

This page titled 10.1: What is a Tree? is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

10.2: Spanning Trees

10.2.1: Motivation

The topic of spanning trees is motivated by a graph-optimization problem.

A graph of Atlantis University (Figure 10.2.1) shows that there are four campuses in the system. A new secure communications system is being installed and the objective is to allow for communication between any two campuses; to achieve this objective, the university must buy direct lines between certain pairs of campuses. Let G be the graph with a vertex for each campus and an edge for each direct line. Total communication is equivalent to G being a connected graph. This is due to the fact that two campuses can communicate over any number of lines. To minimize costs, the university wants to buy a minimum number of lines.

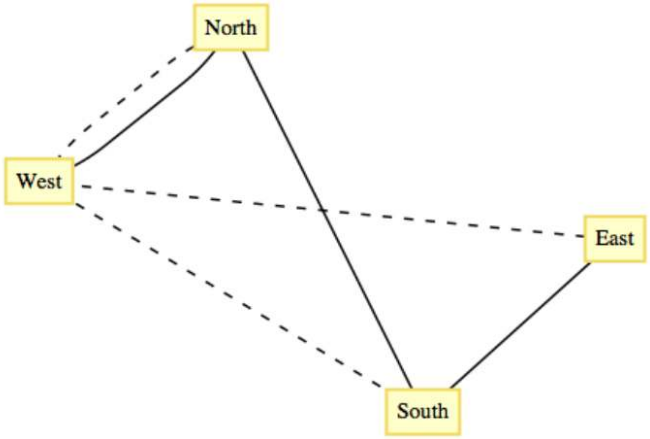


Figure 10.2.1: Atlantis University Graph

The solutions to this problem are all trees. Any graph that satisfies the requirements of the university must be connected, and if a cycle does exist, any line in the cycle can be deleted, reducing the cost. Each of the sixteen trees that can be drawn to connect the vertices North, South, East, and West (see Exercise 10.1.1) solves the problem as it is stated. Note that in each case, three direct lines must be purchased. There are two considerations that can help reduce the number of solutions that would be considered.

- Objective 1: Given that the cost of each line depends on certain factors, such as the distance between the campuses, select a tree whose cost is as low as possible.
- Objective 2: Suppose that communication over multiple lines is noisier as the number of lines increases. Select a tree with the property that the maximum number of lines that any pair of campuses must use to communicate with is as small as possible.

Typically, these objectives are not compatible; that is, you cannot always simultaneously achieve these objectives. In the case of the Atlantis university system, the solution with respect to Objective 1 is indicated with solid lines in Figure 10.2.1. There are four solutions to the problem with respect to Objective 2: any tree in which one campus is directly connected to the other three. One solution with respect to Objective 2 is indicated with dotted lines in Figure 10.2.1. After satisfying the conditions of Objective 2, it would seem reasonable to select the cheapest of the four trees.

10.2.2: Definition

Definition 10.2.1: Spanning Tree

Let $G = (V, E)$ be a connected undirected graph. A spanning tree for G is a spanning subgraph, Definition 9.1.5, of G that is a tree.

Note 10.2.1

a. If (V, E') is a spanning tree, $|E'| = |V| - 1$.
b. The significance of a spanning tree is that it is a minimal spanning set. A smaller set would not span the graph, while a larger set would have a cycle, which has an edge that is superfluous.

For the remainder of this section, we will discuss two of the many topics that relate to spanning trees. The first is the problem of finding Minimal Spanning Trees, which addresses Objective 1 above. The second is the problem of finding Minimum Diameter Spanning Trees, which addresses Objective 2.

Definition 10.2.2: Minimal Spanning Tree

Given a weighted connected undirected graph $G = (V, E, w)$, a minimal spanning tree is a spanning tree (V, E') for which $\sum_{e \in E'} w(e)$ is as small as possible.

10.2.3: Prim's Algorithm

Unlike many of the graph-optimization problems that we've examined, a solution to this problem can be obtained efficiently. It is a situation in which a greedy algorithm works.

Definition 10.2.3: Bridge

Let $G = (V, E)$ be an undirected graph and let $\{L, R\}$ be a partition of V. A bridge between L and R is an edge in E that connects a vertex in L to a vertex in R.

Theorem 10.2.1

Let $G = (V, E, w)$ be a weighted connected undirected graph. Let V be partitioned into two sets L and R. If e^* is a bridge of least weight between L and R, then there exists a minimal spanning tree for G that includes e^*.

Proof

ASuppose that no minimal spanning tree including e^* exists. Let $T = (V, E')$ be a minimal spanning tree. If we add e^* to T, a cycle is created, and this cycle must contain another bridge, e, between L and R. Since $w(e^*) \leq w(e)$, we can delete e and the new tree, which includes e^* must also be a minimal spanning tree.

Example 10.2.1: Some Bridges

The bridges between the vertex sets $\{a, b, c\}$ and $\{d, e\}$ in Figure 10.2.2 are the edges $\{b, d\}$ and $\{c, e\}$. According to the theorem above, a minimal spanning tree that includes $\{b, d\}$ exists. By examination, you should be able to see that this is true. Is it true that only the bridges of minimal weight can be part of a minimal spanning tree?

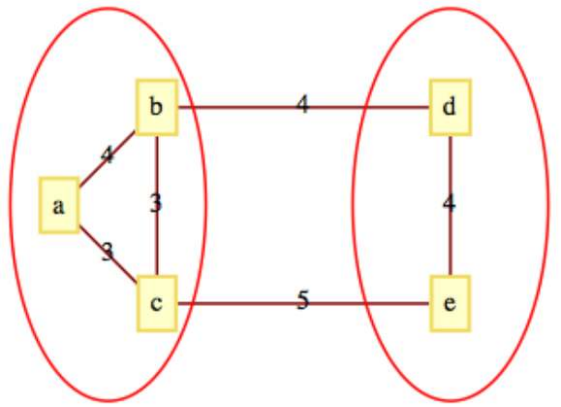


Figure 10.2.2: Bridges between two sets

Theorem 10.2.1 essentially tells us that a minimal spanning tree can be constructed recursively by continually adding minimally weighted bridges to a set of edges.

Algorithm 10.2.1: Prim's Algorithm

Let $G = (V, E, w)$ be a connected, weighted, undirected graph, and let v_0 be an arbitrary vertex in V. The following steps lead to a minimal spanning tree for G. L and R will be sets of vertices and E' is a set of edges.

1. (Initialize) $L = V - \{v_0\}$; $R = \{v_0\}$; $E' = \emptyset$.
2. (Build the tree) While $L \neq \emptyset$:
    1. Find $e^* = \{v_L, v_R\}$, a bridge of minimum weight between L and R.
    2. $R = R \cup \{v_L\}$; $L = L - \{v_L\}$; $E' = E' \cup \{e^*\}$
3. Terminate with a minimal spanning tree (V, E').

Note 10.2.2

a. If more than one minimal spanning tree exists, then the one that is obtained depends on v_0 and the means by which e^* is selected in Step 2.
b. Warning: If two minimally weighted bridges exist between L and R, do not try to speed up the algorithm by adding both of them to E'.
c. That Algorithm 10.2.1 yields a minimal spanning tree can be proven by induction with the use of Theorem 10.2.1.
d. If it is not known whether G is connected, Algorithm 10.2.1 can be revised to handle this possibility. The key change (in Step 2.1) would be to determine whether any bridge at all exists between L and R. The condition of the while loop in Step 2 must also be changed somewhat.

Example 10.2.2: A Small Example

Consider the graph in Figure 10.2.3. If we apply Prim's Algorithm, Algorithm 10.2.1, starting at a, we obtain the following edge list in the order given: $\{a, f\}, \{f, e\}, \{e, c\}, \{c, d\}, \{f, b\}, \{b, g\}$ The total of the weights of these edges is 20. The method that we have used (in Step 2.1) to select a bridge when more than one minimally weighted bridge exists is to order all bridges alphabetically by the vertex in L and then, if further ties exist, by the vertex in R. The first vertex in that order is selected in Step 2.1 of the algorithm.

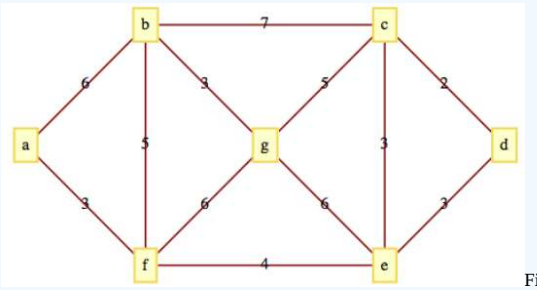


Figure 10.2.3: A small weighted graph

Definition 10.2.4: Minimum Diameter Spanning Tree

Given a connected undirected graph $G = (V, E)$, find a spanning tree $T = (V, E')$ of G such that the longest path in T is as short as possible.

Example 10.2.3: The Case for Complete Graphs

The Minimum Diameter Spanning Tree Problem is trivial to solve in a K_n. Select any vertex v_0 and construct the spanning tree whose edge set is the set of edges that connect v_0 to the other vertices in the K_n. Figure 10.2.4 illustrates a solution for $n = 5$.

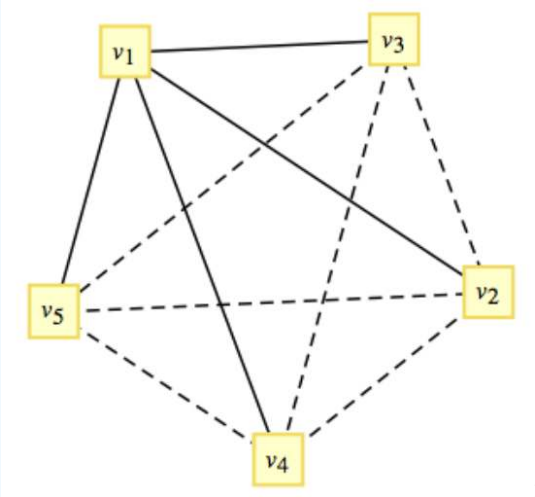


Figure 10.2.4: Minimum diameter spanning tree for K_5

For incomplete graphs, a two-stage algorithm is needed. In short, the first step is to locate a "center" of the graph. The maximum distance from a center to any other vertex is as small as possible. Once a center is located, a breadth-first search of the graph is used to construct the spanning tree.

10.2.4: Exercises

Exercise 10.2.1

Suppose that after Atlantis University's phone system is in place, a fifth campus is established and that a transmission line can be bought to connect the new campus to any old campus. Is this larger system the most economical one possible with respect to Objective 1? Can you always satisfy Objective 2?

Answer

It might not be most economical with respect to Objective 1. You should be able to find an example to illustrate this claim. The new system can always be made most economical with respect to Objective 2 if the old system were designed with that objective in mind.

Exercise 10.2.2

Construct a minimal spanning tree for the capital cities in New England (see Table 9.5.1).

Exercise 10.2.3

Show that the answer to the question posed in Example 10.2.1 is "no."

Answer

In the figure below, $\{1, 2\}$ is not a minimal bridge between $L = \{1, 4\}$ and $R = \{2, 3\}$, but it is part of the minimal spanning tree for this graph.

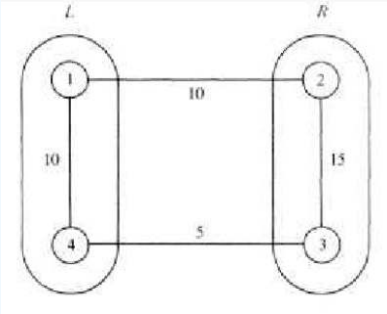


Figure 10.2.5

Exercise 10.2.4

Find a minimal spanning tree for the following graphs.

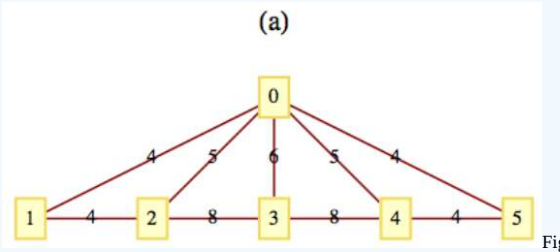


Figure 10.2.6

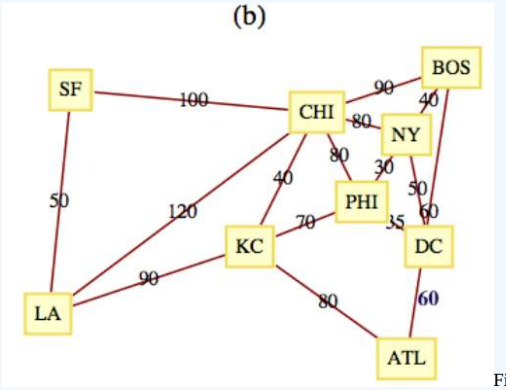


Figure 10.2.7

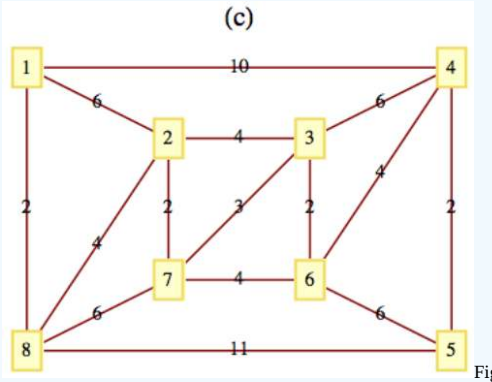


Figure 10.2.8

Exercise 10.2.5

Find a minimum diameter spanning tree for the following graphs.

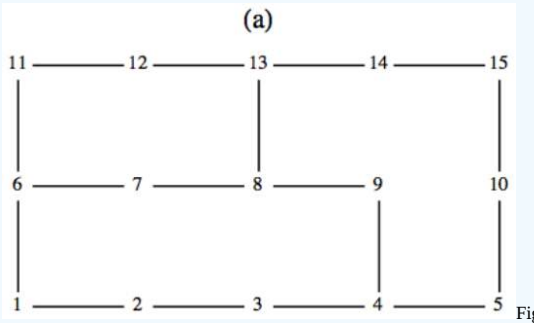


Figure 10.2.9

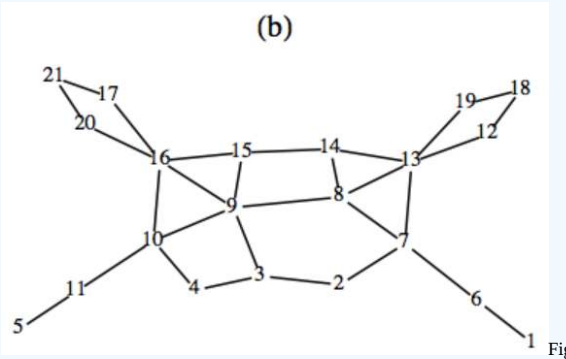


Figure 10.2.10

Answer

a. Edges in one solution are: $\{8,7\},\{8,9\},\{8,13\},\{7,6\},\{9,4\},\{13,12\},\{13,14\},\{6,11\},\{6,1\},\{1,2\},\{4,3\},\{4,5\},\{14,15\},$ and $\{5,10\}$

b. Vertices 8 and 9 are centers of the graph. Starting from vertex 8, a minimum diameter spanning tree is
$\{\{8,3\},\{8,7\},\{8,13\},\{8,14\},\{8,9\},\{3,2\},\{3,4\},\{7,6\},\{13,12\},\{13,19\},\{14,15\},\{9,16\},\{9,10\},\{6,1\},\{12,18\},\{16,20\},\{16,17\},\{10,11\},\{20,21\},\{11,5\}\}$.
The diameter of the tree is 7.

Exercise 10.2.6

In each of the following parts justify your answer with either a proof or a counterexample.

a. Suppose a weighted undirected graph had distinct edge weights. Is it possible that no minimal spanning tree includes the edge of minimal weight?

b. Suppose a weighted undirected graph had distinct edge weights. Is it possible that every minimal spanning tree includes the edge of maximal weight? If true, under what conditions would it happen?

This page titled 10.2: Spanning Trees is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

10.3: Rooted Trees

In the next two sections, we will discuss rooted trees. Our primary foci will be on general rooted trees and on a special case, ordered binary trees.

10.3.1: Definition and Terminology

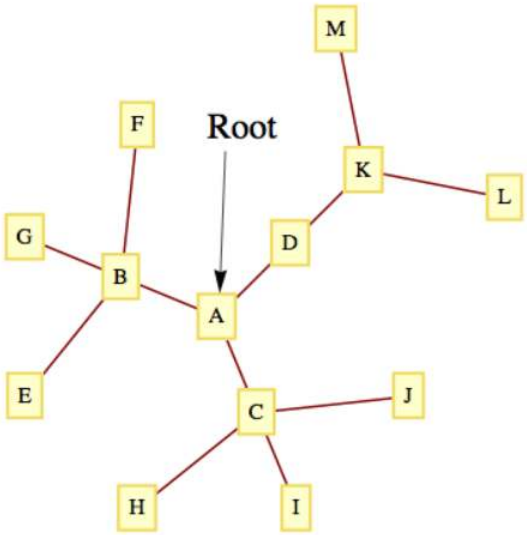


Figure 10.3.1: A Rooted Tree

List 10.3.1: Informal Definition and Terminology

What differentiates rooted trees from undirected trees is that a rooted tree contains a distinguished vertex, called the root. Consider the tree in Figure 10.3.1. Vertex A has been designated the root of the tree. If we choose any other vertex in the tree, such as M, we know that there is a unique path from A to M. The vertices on this path, (A, D, K, M), are described in genealogical terms:

- M is a child of K (so is L)
- K is M's parent.
- A, D, and K are M's ancestors.
- D, K, and M are descendants of A.

These genealogical relationships are often easier to visualize if the tree is rewritten so that children are positioned below their parents, as in Figure 10.3.2

With this format, it is easy to see that each vertex in the tree can be thought of as the root of a tree that contains, in addition to itself, all of its descendants. For example, D is the root of a tree that contains D, K, L, and M. Furthermore, K is the root of a tree that contains K, L, and M. Finally, L and M are roots of trees that contain only themselves. From this observation, we can give a formal definition of a rooted tree.

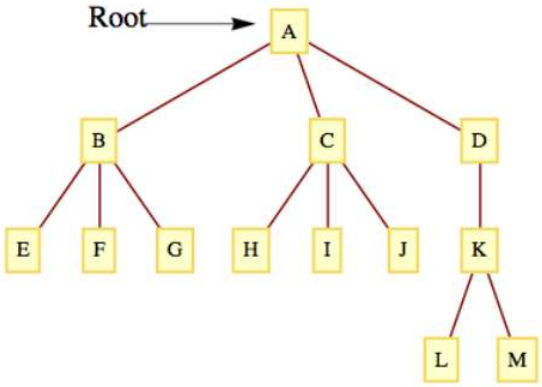


Figure 10.3.2: A Rooted Tree, redrawn

Definition 10.3.1: Rooted Tree

a. Basis: A tree with no vertices is a rooted tree (the empty tree).
b. A single vertex with no children is a rooted tree.
c. Recursion: Let $T_1, T_2, \ldots, T_r, r \geq 1$, be disjoint rooted trees with roots $v_1, v_2, \ldots, v_r$, respectively, and let v_0 be a vertex that does not belong to any of these trees. Then a rooted tree, rooted at v_0, is obtained by making v_0 the parent of the vertices $v_1, v_2, \ldots,$ and v_r. We call $T_1, T_2, \ldots, T_r$ subtrees of the larger tree.

The ***level of a vertex*** of a rooted tree is the number of edges that separate the vertex from the root. The level of the root is zero. The depth of a tree is the maximum level of the vertices in the tree. The depth of a tree in Figure $10.3.2$ is three, which is the level of the vertices L and M. The vertices E, F, G, H, I, J, and K have level two. B, C, and D are at level one and A has level zero.

Example 10.3.1: A Decision Tree

Figure 2.1.1 is a rooted tree with `Start` as the root. It is an example of what is called a decision tree.

Example 10.3.2: Tree Structure of Data

One of the keys to working with large amounts of information is to organize it in a consistent, logical way. A ***data structure*** is a scheme for organizing data. A simple example of a data structure might be the information a college admissions department might keep on their applicants. Items might look something like this:

$$\begin{aligned} ApplicantItem = (&FirstName, MiddleInitial, LastName, StreetAddress,\\ &City, State, Zip, HomePhone, CellPhone, EmailAddress,\\ &HighSchool, Major, ApplicationPaid, MathSAT, VerbalSAT,\\ &Recommendation1, Recommendation2, Recommendation3) \end{aligned}$$

This structure is called a “flat file”.

A spreadsheet can be used to arrange data in this way. Although a “flat file” structure is often adequate, there are advantages to clustering some the information. For example the applicant information might be broken into four parts: name, contact

information, high school, and application data:

$$\begin{aligned} ApplicantItem = ((&FirstName, MiddleInitial, LastName),\\ &((StreetAddress, City, State, Zip),\\ &(HomePhone, CellPhone), EmailAddress),\\ &HighSchool,\\ &(Major, ApplicationPaid, (MathSAT, VerbalSAT),\\ &(Recommendation1, Recommendation2, Recommendation3)) \end{aligned}$$

The first item in each ApplicantItem is a list $(FirstName, MiddleInitial, LastName)$, with each item in that list being a single field of the original flat file. The third item is simply the single high school item from the flat file. The application data is a list and one of its items, is itself a list with the recommendation data for each recommendation the applicant has.

The organization of this data can be visualized with a rooted tree such as the one in Figure 10.3.3

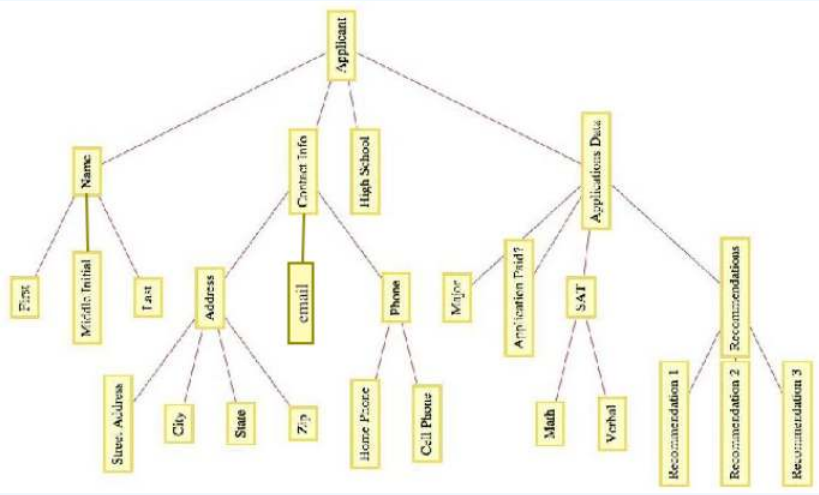


Figure 10.3.3: Applicant Data in a Rooted Tree

In general, you can represent a data item, T, as a rooted tree with T as the root and a subtree for each field. Those fields that are more than just one item are roots of further subtrees, while individual items have no further children in the tree.

10.3.2: Kruskal's Algorithm

An alternate algorithm for constructing a minimal spanning tree uses a forest of rooted trees. First we will describe the algorithm in its simplest terms. Afterward, we will describe how rooted trees are used to implement the algorithm. Finally, we will demonstrate the SageMath implementation of the algorithm. In all versions of this algorithm, assume that $G = (V, E, w)$ is a weighted undirected graph with $|V| = m$ and $|E| = n$.

Algorithm 10.3.1: Kruskal's Algorithm - Informal Version

1. Sort the edges of G in ascending order according to weight. That is,
$$i \leq j \Leftrightarrow w\left(e_i\right) \leq w\left(e_j\right).$$
2. Go down the list from Step 1 and add edges to a set (initially empty) of edges so that the set does not form a cycle. When an edge that would create a cycle is encountered, ignore it. Continue examining edges until either $m-1$ edges have been selected or you have come to the end of the edge list. If $m-1$ edges are selected, these edges make up a minimal spanning tree for G. If fewer than $m-1$ edges are selected, G is not connected.

Step 1 can be accomplished using one of any number of standard sorting routines. Using the most efficient sorting routine, the time required to perform this step is proportional to $n \log n$. The second step of the algorithm, also of $n \log n$ time complexity, is the one that uses a forest of rooted trees to test for whether an edge should be added to the spanning set.

Algorithm 10.3.2: Kruskal's Algorithm

1. Sort the edges of G in ascending order according to weight. That is,

$$i \leq j \Leftrightarrow w\left(e_j\right) \leq w\left(e_j\right).$$

2.
    1. Initialize each vertex in V to be the root of its own rooted tree.
    2. Go down the list of edges until either a spanning tree is completed or the edge list has been exhausted. For each edge $e = \{v_1, v_2\}$, we can determine whether e can be added to the spanning set without forming a cycle by determining whether the root of $v_1{}'s$ tree is equal to the root of $v_2{}'s$ tree. If the two roots are equal, then ignore e. If the roots are different, then we can add e to the spanning set. In addition, we merge the trees that v_1 and v_2 belong to. This is accomplished by either making $v_1{}'s$ root the parent of $v_2{}'s$ root or vice versa.

Note 10.3.1

a. Since we start the Kruskal's algorithm with m trees and each addition of an edge decreases the number of trees by one, we end the algorithm with one rooted tree, provided a spanning tree exists.
b. The rooted tree that we develop in the algorithm is not the spanning tree itself.

10.3.3: SageMath Note - Implementation of Kruskal's Algorithm

Kruskal's algorithm has been implemented in Sage. We illustrate how the spanning tree for a weighted graph in can be generated. First, we create such a graph

We will create a graph using a list of triples of the form $(\text{vertex}, \text{vertex}, \text{label})$. The $weighted$ method tells Sage to consider the labels as weights.

```
edges=[(1, 2, 4), (2, 8, 4), (3, 8, 4), (4, 7, 5), (6, 8, 5), (1, 3, 6), (1,
7, 6), (4, 5, 6), (5, 10, 9), (2, 10, 7), (4, 6, 7), (2, 4, 8), (1,8, 9), (1,
9, 9), (5, 6, 9), (1, 10, 10), (2, 9, 10), (4, 9, 10), (5, 9, 10), (6, 9, 10)]
G=Graph(edges)
G.weighted(True)
G.graphplot(edge_labels=True,save_pos=True).show()
```

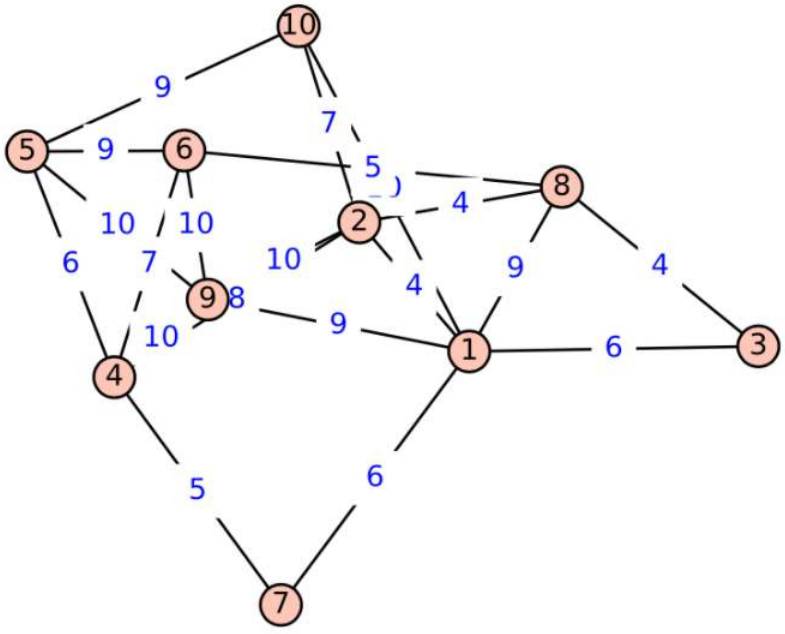


Figure 10.3.4: Weighted graph, SageMath output

Next, we load the kruskal function and use it to generate the list of edges in a spanning tree of G.

```
from sage.graphs.spanning_tree import kruskal
E = kruskal(G, check=True);E
```

To see the resulting tree with the same embedding as G, we generate a graph from the spanning tree edges. Next, we set the positions of the vertices to be the same as in the graph. Finally, we plot the tree.

```
T=Graph(E)
T.set_pos(G.get_pos())
T.graphplot(edge_labels=True).show()
```

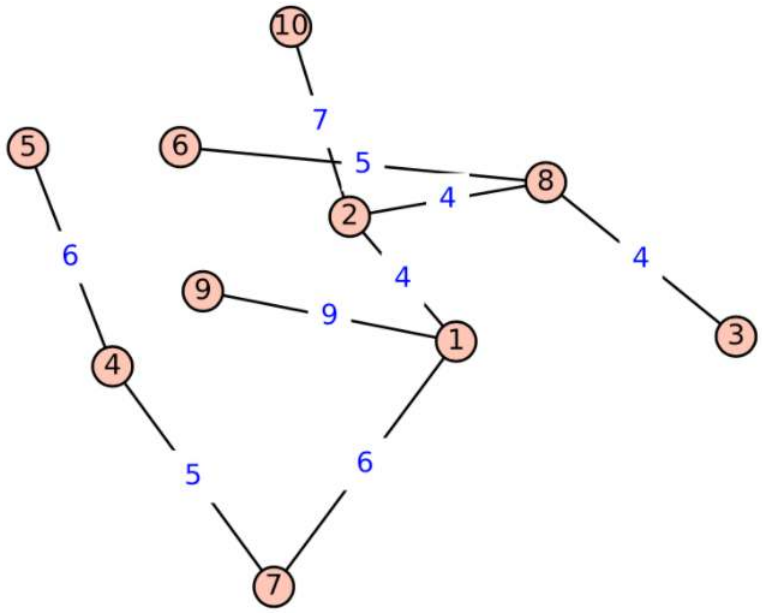


Figure 10.3.5: Spanning tree, SageMath output

10.3.4: Exercises

Exercise 10.3.1

Suppose that an undirected tree has diameter d and that you would like to select a vertex of the tree as a root so that the resulting rooted tree has the smallest depth possible. How would such a root be selected and what would be the depth of the tree (in terms of d)?

Answer

Locate any simple path of length d and locate the vertex in position $\lceil d/2 \rceil$ on the path. The tree rooted at that vertex will have a depth of $\lceil d/2 \rceil$, which is minimal.

Exercise 10.3.2

Use Kruskal's algorithm to find a minimal spanning tree for the following graphs. In addition to the spanning tree, find the final rooted tree in the algorithm. When you merge two trees in the algorithm, make the root with the lower number the root of the new tree.

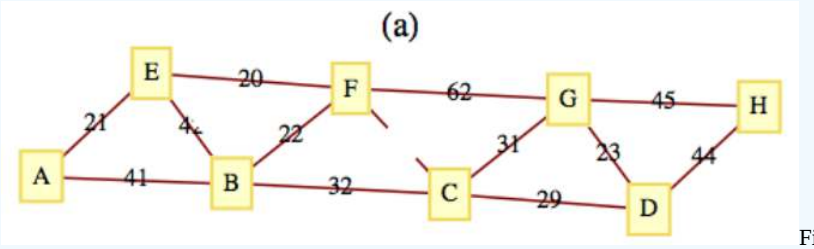


Figure 10.3.6

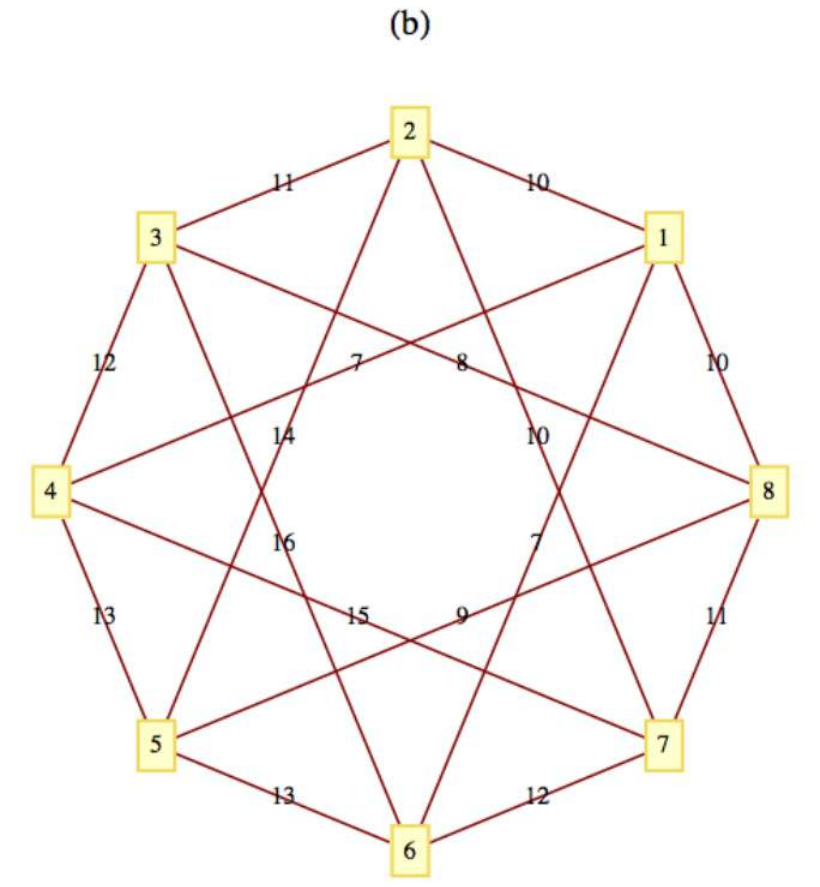


Figure 10.3.7

Exercise 10.3.3

Suppose that information on buildings is arranged in records with five fields: the name of the building, its location, its owner, its height, and its floor space. The location and owner fields are records that include all of the information that you would expect, such as street, city, and state, together with the owner's name (first, middle, last) in the owner field. Draw a rooted tree to describe this type of record

Answer

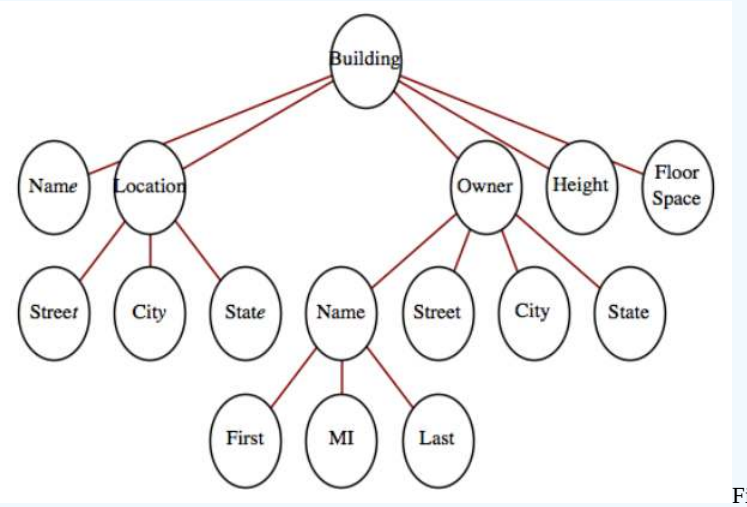


Figure 10.3.8

Exercise 10.3.4

Step through Kruskal's Algorithm by hand to verify that the example of a minimal spanning tree using Sage in Subsection 10.3.3 is correct.

This page titled 10.3: Rooted Trees is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

10.4: Binary Trees

10.4.1: Definition of a Binary Tree

An ***ordered rooted tree*** is a rooted tree whose subtrees are put into a definite order and are, themselves, ordered rooted trees. An empty tree and a single vertex with no descendants (no subtrees) are ordered rooted trees.

Example 10.4.1: Distinct Ordered Rooted Trees

The trees in Figure 10.4.1 are identical rooted trees, with root 1, but as ordered trees, they are different.

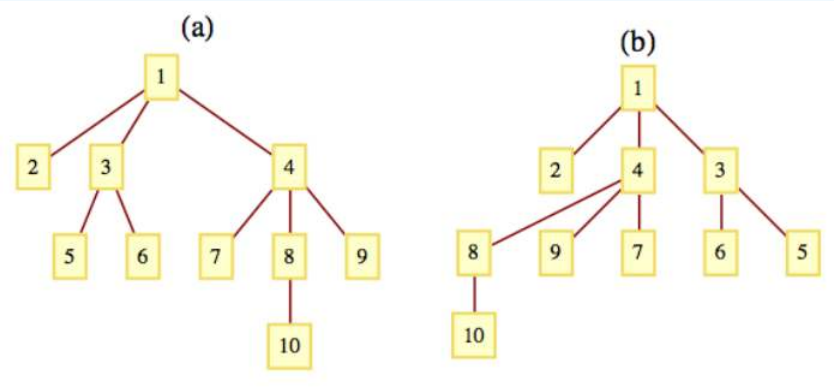


Figure 10.4.1: Two different ordered rooted trees

If a tree rooted at v has p subtrees, we would refer to them as the first, second,..., p^{th} subtrees. There is a subtle difference between certain ordered trees and binary trees, which we define next.

Definition 10.4.1: Binary Tree

1. A tree consisting of no vertices (the empty tree) is a binary tree
2. A vertex together with two subtrees that are both binary trees is a binary tree. The subtrees are called the left and right subtrees of the binary tree.

The difference between binary trees and ordered trees is that every vertex of a binary tree has exactly two subtrees (one or both of which may be empty), while a vertex of an ordered tree may have any number of subtrees. But there is another significant difference between the two types of structures. The two trees in Figure 10.4.2 would be considered identical as ordered trees. However, they are different binary trees. Tree (a) has an empty right subtree and Tree (b) has an empty left subtree.

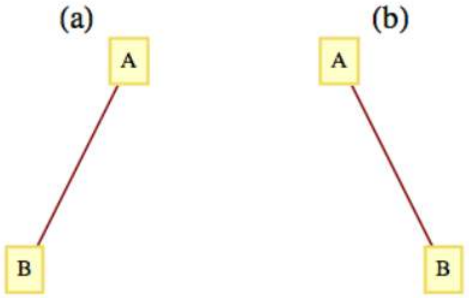


Figure 10.4.2: Two different binary trees

List 10.4.1: Terminology and General Facts about Binary Trees

a. A vertex of a binary tree with two empty subtrees is called a ***leaf***. All other vertices are called *internal vertices*.
b. The number of leaves in a binary tree can vary from one up to roughly half the number of vertices in the tree (see Exercise 10.4.4 of this section).
c. The maximum number of vertices at level k of a binary tree is 2^k , $k \geq 0$ (see Exercise 10.4.6 of this section).

d. A ***full binary tree*** is a tree for which each vertex has either zero or two empty subtrees. In other words, each vertex has either two or zero children. See Exercise 10.4.7 of this section for a general fact about full binary trees.

10.4.2: Traversals of Binary Trees

The traversal of a binary tree consists of visiting each vertex of the tree in some prescribed order. Unlike graph traversals, the consecutive vertices that are visited are not always connected with an edge. The most common binary tree traversals are differentiated by the order in which the root and its subtrees are visited. The three traversals are best described recursively and are:

Definition 10.4.2: Preorder Traversal

1. Visit the root of the tree.
2. Preorder traverse the left subtree.
3. Preorder traverse the right subtree.

Definition 10.4.3: Inorder Traversal

1. Inorder traverse the left subtree.
2. Visit the root of the tree.
3. Inorder traverse the right subtree.

Definition 10.4.4: Postorder Traversal

1. Postorder traverse the left subtree.
2. Postorder traverse the right subtree.
3. Visit the root of the tree.

Any traversal of an empty tree consists of doing nothing.

Example 10.4.2: Traversal Examples

For the tree in Figure 10.4.3, the orders in which the vertices are visited are:

- A-B-D-E-C-F-G, for the preorder traversal.
- D-B-E-A-F-C-G, for the inorder traversal.
- D-E-B-F-G-C-A, for the postorder traversal.

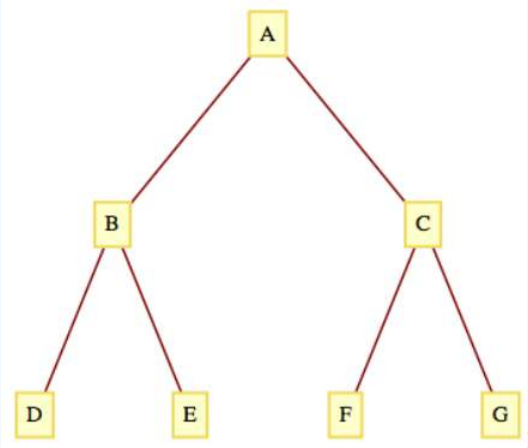


Figure 10.4.3: A Complete Binary Tree to Level 2

Binary Tree Sort. Given a collection of integers (or other objects than can be ordered), one technique for sorting is a binary tree sort. If the integers are $a_1, a_2, \ldots, a_n$, $n \geq 1$, we first execute the following algorithm that creates a binary tree:

Algorithm 10.4.1: Binary Sort Tree Creation

1. Insert a_1 into the root of the tree.
2. For k := 2 to n // insert a_k into the tree
   a. r = a_1
   b. inserted = false
   c. while not(inserted):
      if $a_k < r$:
         if r has a left child:
            r = left child of r
         else:
            make a_k the left child of r
            inserted = true
      else:
         if r has a right child:
            r = right child of r
         else:
            make a_k the right child of r
            inserted = true

If the integers to be sorted are 25, 17, 9, 20, 33, 13, and 30, then the tree that is created is the one in Figure 10.4.4 The inorder traversal of this tree is 9, 13, 17, 20, 25, 30, 33, the integers in ascending order. In general, the inorder traversal of the tree that is constructed in the algorithm above will produce a sorted list. The preorder and postorder traversals of the tree have no meaning here.

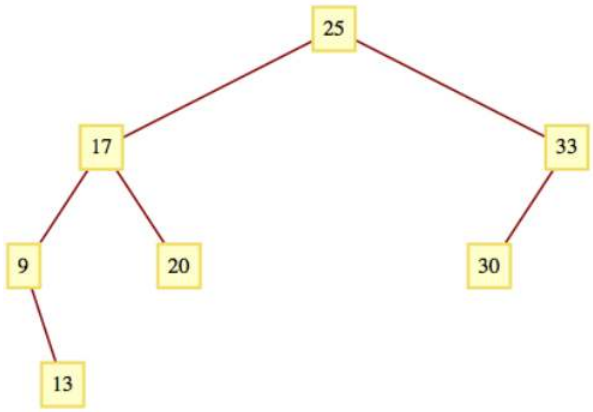


Figure 10.4.4: A Binary Sorting Tree

10.4.3: Expression Trees

A convenient way to visualize an algebraic expression is by its expression tree. Consider the expression

$$X = a * b - c/d + e.$$

Since it is customary to put a precedence on multiplication/divisions, X is evaluated as $((a * b) - (c/d)) + e$. Consecutive multiplication/divisions or addition/subtractions are evaluated from left to right. We can analyze X further by noting that it is the sum of two simpler expressions $(a * b) - (c/d)$ and e. The first of these expressions can be broken down further into the difference of the expressions $a * b$ and c/d. When we decompose any expression into $(\text{left expression})\text{operation}(\text{right expression})$, the expression tree of that expression is the binary tree whose root contains the operation and whose left and right subtrees are the trees of the left and right expressions, respectively. Additionally, a simple variable or a number has an expression tree that is a single vertex containing the variable or number. The evolution of the expression tree for expression X appears in Figure 10.4.5

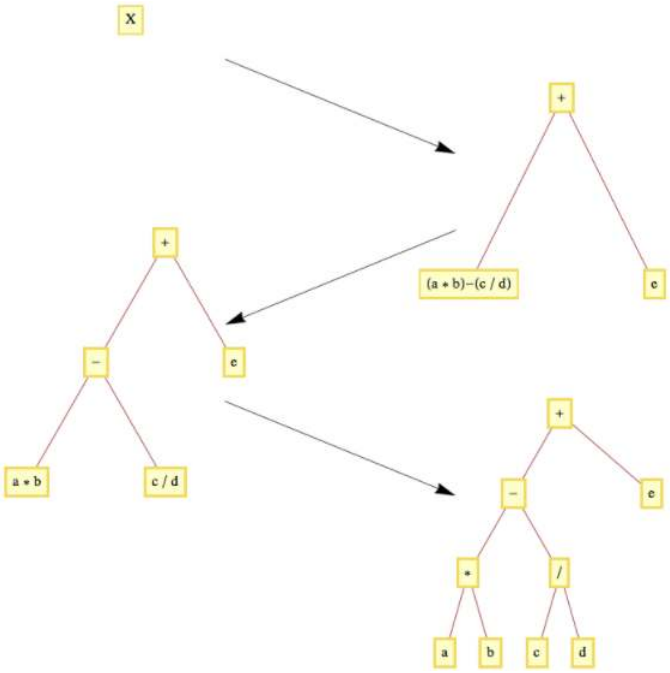


Figure 10.4.5: Building an Expression Tree

Example 10.4.3: Some Expression Trees

a. If we intend to apply the addition and subtraction operations in X first, we would parenthesize the expression to $a*(b-c)/(d+e)$. Its expression tree appears in Figure 10.4.6(a).
b. The expression trees for a^2-b^2 and for $(a+b)*(a-b)$ appear in Figure 10.4.6(b) and Figure 10.4.6(c).

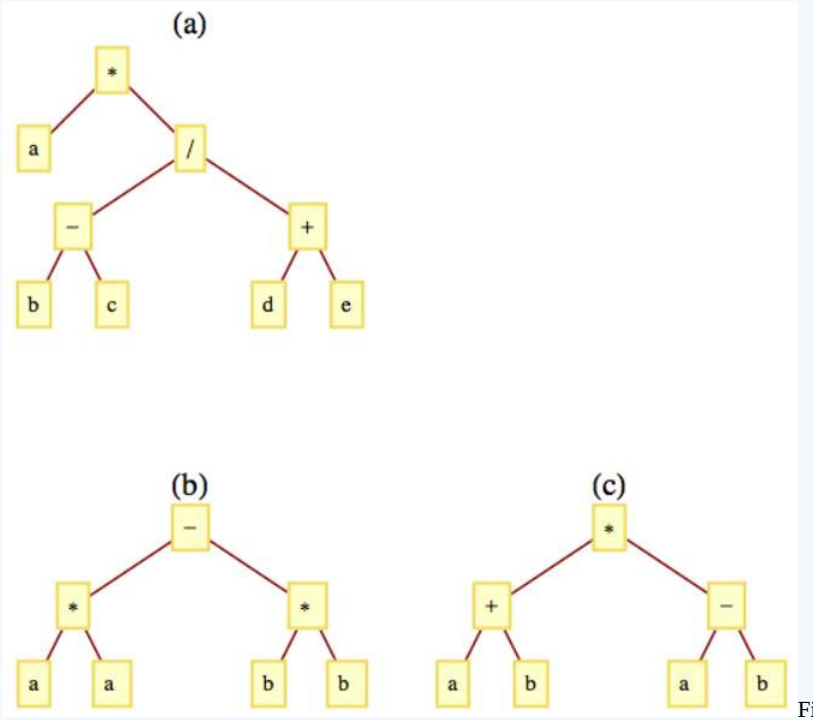


Figure 10.4.6: Expression Tree Examples

The three traversals of an operation tree are all significant. A binary operation applied to a pair of numbers can be written in three ways. One is the familiar infix form, such as $a+b$ for the sum of a and b. Another form is prefix, in which the same sum is written $+ab$. The final form is postfix, in which the sum is written $ab+$. Algebraic expressions involving the four standard arithmetic operations $(+, -, *, \text{and} /)$ in prefix and postfix form are defined as follows:

List 10.4.2: Prefix and Postfix Forms of an Algebraic Expression

Prefix

a. A variable or number is a prefix expression
b. Any operation followed by a pair of prefix expressions is a prefix expression.

Postfix

a. A variable or number is a postfix expression
b. Any pair of postfix expressions followed by an operation is a postfix expression.

The connection between traversals of an expression tree and these forms is simple:

a. The preorder traversal of an expression tree will result in the prefix form of the expression.
b. The postorder traversal of an expression tree will result in the postfix form of the expression.
c. The inorder traversal of an operation tree will not, in general, yield the proper infix form of the expression. If an expression requires parentheses in infix form, an inorder traversal of its expression tree has the effect of removing the parentheses.

Example 10.4.4: Traversing an Expression Tree

The preorder traversal of the tree in Figure 10.4.5 is $+-*ab/cde$, which is the prefix version of expression X. The postorder traversal is $ab*cd/-e+$. Note that since the original form of X needed no parentheses, the inorder traversal, $a*b-c/d+e$, is the correct infix version.

10.4.4: Counting Binary Trees

We close this section with a formula for the number of different binary trees with n vertices. The formula is derived using generating functions. Although the complete details are beyond the scope of this text, we will supply an overview of the derivation in order to illustrate how generating functions are used in advanced combinatorics.

Let $B(n)$ be the number of different binary trees of size n (n vertices), $n \geq 0$. By our definition of a binary tree, $B(0) = 1$. Now consider any positive integer $n+1$, $n \geq 0$. A binary tree of size $n+1$ has two subtrees, the sizes of which add up to n. The possibilities can be broken down into $n+1$ cases:

> *Case 0: Left subtree has size 0; right subtree has size n.*
>
> *Case 1: Left subtree has size 1; right subtree has size $n-1$.*
>
> $\vdots$
>
> *Case k: Left subtree has size k; right subtree has size $n-k$.*
>
> $\vdots$
>
> *Case n: Left subtree has size n; right subtree has size 0.*

In the general Case k, we can count the number of possibilities by multiplying the number of ways that the left subtree can be filled, $B(k)$, by the number of ways that the right subtree can be filled. $B(n-k)$. Since the sum of these products equals $B(n+1)$, we obtain the recurrence relation for $n \geq 0$:

$$\begin{aligned} B(n+1) &= B(0)B(n) + B(1)B(n-1) + \cdots + B(n)B(0) \\ &= \sum_{k=0}^{n} B(k)B(n-k) \end{aligned}$$

Now take the generating function of both sides of this recurrence relation:

$$\sum_{n=0}^{\infty} B(n+1)z^n = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} B(k)B(n-k) \right) z^n \tag{10.4.1}$$

or

$$G(B\uparrow; z) = G(B * B; z) = G(B; z)^2 \tag{10.4.2}$$

Recall that $G(B\uparrow; z) = \frac{G(B;z)-B(0)}{z} = \frac{G(B;z)-1}{z}$ If we abbreviate $G(B; z)$ to G, we get

$$\frac{G-1}{z} = G^2 \Rightarrow zG^2 - G + 1 = 0$$

Using the quadratic equation we find two solutions:

$$G_1 = \frac{1+\sqrt{1-4z}}{2z} \text{ and} \tag{10.4.3}$$

$$G_2 = \frac{1-\sqrt{1-4z}}{2z} \tag{10.4.4}$$

The gap in our derivation occurs here since we don't presume a knowledge of calculus. If we expand G_1 as an extended power series, we find

$$G_1 = \frac{1+\sqrt{1-4z}}{2z} = \frac{1}{z} - 1 - z - 2z^2 - 5z^3 - 14z^4 - 42z^5 + \cdots \tag{10.4.5}$$

The coefficients after the first one are all negative and there is a singularity at 0 because of the $\frac{1}{z}$ term. However if we do the same with G_2 we get

$$G_2 = \frac{1-\sqrt{1-4z}}{2z} = 1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + \cdots \tag{10.4.6}$$

Further analysis leads to a closed form expression for $B(n)$, which is

$$B(n) = \frac{1}{n+1} \binom{2n}{n}$$

This sequence of numbers is often called the ***Catalan numbers***. For more information on the Catalan numbers, see the entry A000108 in The On-Line Encyclopedia of Integer Sequences.

10.4.5: SageMath Note - Power Series

It may be of interest to note how the extended power series expansions of G_1 and G_2 are determined using Sage. In Sage, one has the capability of being very specific about how algebraic expressions should be interpreted by specifying the underlying ring. This can make working with various algebraic expressions a bit more confusing to the beginner. Here is how to get a Laurent expansion for G_1 above.

```
R.<z>=PowerSeriesRing(ZZ,'z')
G1=(1+sqrt(1-4*z))/(2*z)
G1
```

The first Sage expression above declares a structure called a ***ring*** that contains power series. We are not using that whole structure, just a specific element, `G1`. So the important thing about this first input is that it establishes `z` as being a variable associated with power series over the integers. When the second expression defines the value of `G1` in terms of `z`, it is automatically converted to a power series.

The expansion of G_2 uses identical code, and its coefficients are the values of $B(n)$.

In Chapter 16 we will introduce rings and will be able to take further advantage of Sage's capabilities in this area.

10.4.6: Exercises

Exercise 10.4.1

Draw the expression trees for the following expressions:

a. $a(b+c)$
b. $ab+c$
c. $ab+ac$
d. $bb-4ac$
e. $((a_3x+a_2)x+a_1)x+a_0$

Answer

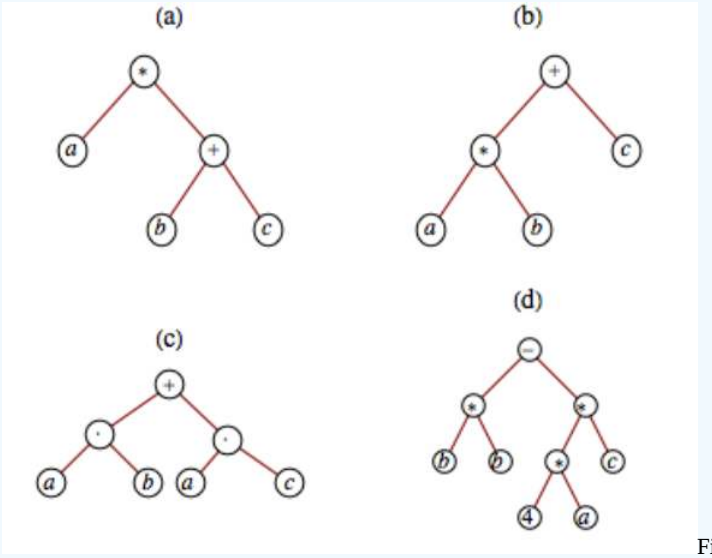


Figure 10.4.7

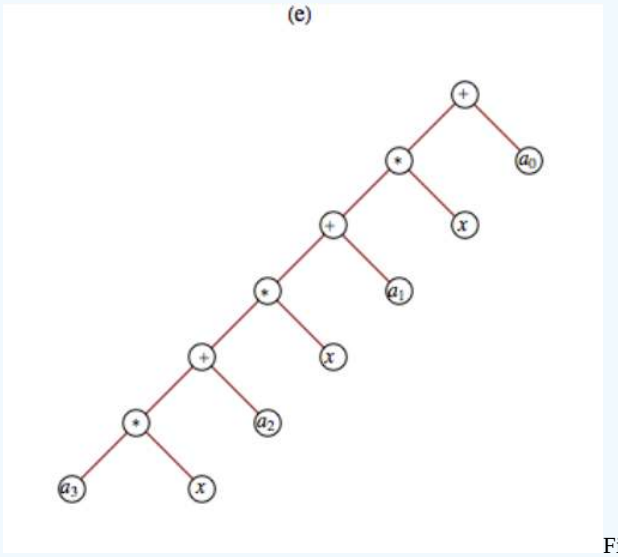


Figure 10.4.8

Exercise 10.4.2

Draw the expression trees for

a. $\dfrac{x^2-1}{x-1}$
b. $xy+xz+yz$

Exercise 10.4.3

Write out the preorder, inorder, and postorder traversals of the trees in Exercise 10.4.1 above.

Answer

	Preorder	Inorder	Postorder
(a)	$\cdot a + bc$	$a \cdot b + c$	$abc + \cdot$
(b)	$+ \cdot abc$	$a \cdot b + c$	$ab \cdot c +$
(c)	$+ \cdot ab \cdot ac$	$a \cdot b + a \cdot c$	$ab \cdot ac \cdot +$

Exercise 10.4.4

Verify the formula for $B(n)$, $0 \leq n \leq 3$ by drawing all binary trees with three or fewer vertices.

Exercise 10.4.5

a. Draw a binary tree with seven vertices and only one leaf.
b. Draw a binary tree with seven vertices and as many leaves as possible.

Answer

Add texts here. Do not delete this text first.

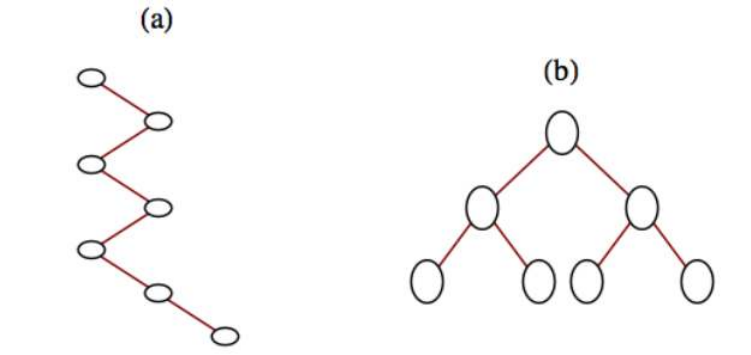


Figure 10.4.9

Exercise 10.4.6

Prove that the maximum number of vertices at level k of a binary tree is 2^k and that a tree with that many vertices at level k must have $2^{k+1} - 1$ vertices.

Exercise 10.4.7

Prove that if T is a full binary tree, then the number of leaves of T is one more than the number of internal vertices (non-leaves).

Answer

Solution 1:

Basis: A binary tree consisting of a single vertex, which is a leaf, satisfies the equation leaves $=$ internal vertices $+1$

Induction: Assume that for some $k \geq 1$, all full binary trees with k or fewer vertices have one more leaf than internal vertices. Now consider any full binary tree with $k+1$ vertices. Let T_A and T_B be the left and right subtrees of the tree which, by the definition of a full binary tree, must both be full. If i_A and i_B are the numbers of internal vertices in T_A and T_B, and j_A and j_B are the numbers of leaves, then $j_A = i_A + 1$ and $j_B = i_B + 1$. Therefore, in the whole tree,

$$\begin{aligned}\text{the number of leaves} &= j_A + j_B \\ &= (i_A + 1) + (i_B + 1) \\ &= (i_A + i_B + 1) + 1 \\ &= (\text{number of internal vertices}) + 1\end{aligned}$$

Solution 2:

Imagine building a full binary tree starting with a single vertex. By continuing to add leaves in pairs so that the tree stays full, we can build any full binary tree. Our starting tree satisfies the condition that the number of leaves is one more than the number of internal vertices . By adding a pair of leaves to a full binary tree, an old leaf becomes an internal vertex, increasing the number of internal vertices by one. Although we lose a leaf, the two added leaves create a net increase of one leaf. Therefore, the desired equality is maintained.

This page titled 10.4: Binary Trees is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

11: Algebraic Structures

> ***Abelian Group***
>
> *In **Abelian groups**, when computing,*
> *With operands there's no refuting:*
> *The expression bc*
> *Is the same as cb.*
> *Not en route to your job, yet commuting.*
>
> *Howard Spindel, The Omnificent English Dictionary in Limerick Form*

The primary goal of this chapter is to make the reader aware of what an algebraic system is and how algebraic systems can be studied at different levels of abstraction. After describing the concrete, axiomatic, and universal levels, we will introduce one of the most important algebraic systems at the axiomatic level, the group. In this chapter, group theory will be a vehicle for introducing the universal concepts of isomorphism, direct product, subsystem, and generating set. These concepts can be applied to all algebraic systems. The simplicity of group theory will help the reader obtain a good intuitive understanding of these concepts. In Chapter 15, we will introduce some additional concepts and applications of group theory. We will close the chapter with a discussion of how some computer hardware and software systems use the concept of an algebraic system.

11.1: Operations
11.2: Algebraic Systems
11.3: Some General Properties of Groups
11.4: Greatest Common Divisors and the Integers Modulo n
11.5: Subsystems
11.6: Direct Products
11.7: Isomorphisms

This page titled 11: Algebraic Structures is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.1: Operations

One of the first mathematical skills that we all learn is how to add a pair of positive integers. A young child soon recognizes that something is wrong if a sum has two values, particularly if his or her sum is different from the teacher's. In addition, it is unlikely that a child would consider assigning a non-positive value to the sum of two positive integers. In other words, at an early age we probably know that the sum of two positive integers is unique and belongs to the set of positive integers. This is what characterizes all binary operations on a set.

11.1.1: Operation?

Definition 11.1.1: Binary Operation

Let S be a nonempty set. A binary operation on S is a rule that assigns to each ordered pair of elements of S a unique element of S. In other words, a binary operation is a function from $S \times S$ into S.

Example 11.1.1: Some Common Binary Operations

Union and intersection are both binary operations on the power set of any universe. Addition and multiplication are binary operators on the natural numbers. Addition and multiplication are binary operations on the set of 2 by 2 real matrices, $M_{2\times 2}(\mathbb{R})$. Division is a binary operation on some sets of numbers, such as the positive reals. But on the integers $(1/2 \notin \mathbb{Z})$ and even on the real numbers $(1/0$ is not defined), division is not a binary operation.

Note 11.1.1

a. We stress that the image of each ordered pair must be in S. This requirement disqualifies subtraction on the natural numbers from consideration as a binary operation, since $1-2$ is not a natural number. Subtraction is a binary operation on the integers.
b. On Notation. Despite the fact that a binary operation is a function, symbols, not letters, are used to name them. The most commonly used symbol for a binary operation is an asterisk, $*$. We will also use a diamond, $\diamond$, when a second symbol is needed.

If $*$ is a binary operation on S and $a, b \in S,$ there are three common ways of denoting the image of the pair (a, b). They are:

$*ab$	$a*b$	$ab*$
Prefix Form	Infix Form	Postfix Form

We are all familiar with infix form. For example, $2+3$ is how everyone is taught to write the sum of 2 and 3. But notice how $2+3$ was just described in the previous sentence! The word sum preceded 2 and 3. Orally, prefix form is quite natural to us. The prefix and postfix forms are superior to infix form in some respects. In Chapter 10, we saw that algebraic expressions with more than one operation didn't need parentheses if they were in prefix or postfix form. However, due to our familiarity with infix form, we will use it throughout most of the remainder of this book.

Some operations, such as negation of numbers and complementation of sets, are not binary, but unary operators.

Definition 11.1.2: Unary Operation

Let S be a nonempty set. A unary operator on S is a rule that assigns to each element of S a unique element of S. In other words, a unary operator is a function from S into S.

11.1.2: Properties of Operations

Whenever an operation on a set is encountered, there are several properties that should immediately come to mind. To effectively make use of an operation, you should know which of these properties it has. By now, you should be familiar with most of these properties. We will list the most common ones here to refresh your memory and define them for the first time in a general setting.

First we list properties of a single binary operation.

Definition 11.1.3: Commutative Property

Let $*$ be a binary operation on a set S. We say that * is ***commutative*** if and only if $a * b = b * a$ for all $a, b \in S$.

Definition 11.1.4: Associative Property

Let $*$ be a binary operation on a set S. We say that $*$ is ***associative*** if and only if $(a * b) * c = a * (b * c)$ for all $a, b, c \in S$.

Definition 11.1.5: Identity Property

Let $*$ be a binary operation on a set S. We say that $*$ ***has an identity*** if and only if there exists an element, e, in S such that $a * e = e * a = a$ for all $a \in S$.

The next property presumes that $*$ has the identity property.

Definition 11.1.6: Inverse Property

Let $*$ be a binary operation on a set S. We say that $*$ has the ***inverse property*** if and only if for each $a \in S$, there exists $b \in S$ such that $a * b = b * a = e$. We call b an inverse of a.

Definition 11.1.7: Idempotent Property

Let $*$ be a binary operation on a set S. We say that $*$ is ***idempotent*** if and only if $a * a = a$ for all $a \in S$.

Now we list properties that apply to two binary operations.

Definition 11.1.8: Left Distributive Property

Let $*$ and $\diamond$ be binary operations on a set S. We say that $\diamond$ is left distributive over * if and only if $a \diamond (b * c) = (a \diamond b) * (a \diamond c)$ for all $a, b, c \in S$.

Definition 11.1.9: Right Distributive Property

Let $*$ and $\diamond$ be binary operations on a set S. We say that $\diamond$ is right distributive over * if and only if $(b * c) \diamond a = (b \diamond a) * (c \diamond a)$ for all $a, b, c \in S$.

Definition 11.1.10: Distributive Property

Let $*$ and $\diamond$ be binary operations on a set S. We say that $\diamond$ is distributive over $*$ if and only if $\diamond$ is both left and right distributive over $*$.

There is one significant property of unary operations.

Definition 11.1.11: Involution Property

Let $-$ be a unary operation on S. We say that $-$ has the involution property if $-(-a) = a$ for all $a \in S$.

Finally, a property of sets, as they relate to operations.

Definition 11.1.12: Closure Property

Let T be a subset of S and let $*$ be a binary operation on S. We say that T is closed under $*$ if $a, b \in T$ implies that $a * b \in T$.

In other words, T is closed under $*$ if by operating on elements of T with $*$, you can't get new elements that are outside of T.

Example 11.1.2: Some Examples of Closure and Non-Closure

a. The odd integers are closed under multiplication, but not under addition.
b. Let p be a proposition over U and let A be the set of propositions over U that imply p. That is; $q \in A$ if $q \Rightarrow p$. Then A is closed under both conjunction and disjunction.
c. The set of positive integers that are multiples of 5 is closed under both addition and multiplication.

It is important to realize that the properties listed above depend on both the set and the operation(s). Statements such as "Multiplication is commutative." or "The positive integers are closed." are meaningless on their own. Naturally, if we have established a context in which the missing set or operation is clearly implied, then they would have meaning.

11.1.3: Operation Tables

If the set on which a binary operation is defined is small, a table is often a good way of describing the operation. For example, we might want to define $\oplus$ on $\{0,1,2\}$ by $a \oplus b = \begin{cases} a+b & \text{if } a+b<3 \\ a+b-3 & \text{if } a+b \geq 3 \end{cases}$ The table for $\oplus$ is

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

The top row and left column of an operation table are the column and row headings, respectively. To determine $a \oplus b,$ find the entry in the row labeled a and the column labeled b. The following operation table serves to define * on $\{i,j,k\}$.

$*$	i	j	k
i	i	i	i
j	j	j	j
k	k	k	k

Note that $j*k=j,$ yet $k*j=k.$ Thus, $*$ is not commutative. Commutativity is easy to identify in a table: the table must be symmetric with respect to the diagonal going from the top left to lower right.

11.1.4: Exercises

Exercise 11.1.1

Determine the properties that the following operations have on the positive integers.

a. addition
b. multiplication
c. M defined by $aMb=$ larger of a and b
d. m defined by $amb=$ smaller of a and b
e. @ defined by $a@b=a^b$

Answer

a. Commutative, and associative. Notice that zero is the identity for addition, but it is not a positive integer.
b. Commutative, associative, and has an identity (1)
c. Commutative, associative, has an identity (1), and is idempotent
d. Commutative, associative, and idempotent
e. None. Notice that $2@(3@3)=134217728,$ while $(2@3)@3=512;$ and $a@1=a,$ while $1@a=1.$

Exercise 11.1.2

Which pairs of operations in Exercise 11.1.1 are distributive over one another?

Exercise 11.1.3

Let $*$ be an operation on a set S and $A, B \subseteq S$. Prove that if A and B are both closed under $*$, then $A \cap B$ is also closed under $*$, but $A \cup B$ need not be.

Answer

$$\begin{aligned} a, b \in A \cap B \ &\Rightarrow a, b \in A \text{ by the definition of intersection} \\ &\Rightarrow a * b \in A \text{ by the closure of } A \text{ with respect to } * \end{aligned}$$

Similarly, $a, b \in A \cap B \Rightarrow a * b \in B$. Therefore, $a * b \in A \cap B$. The set of positive integers is closed under addition, and so is the set of negative integers, but $1 + -1 = 0$. Therefore, their union, the nonzero integers, is not closed under addition.

Exercise 11.1.4

How can you pick out the identity of an operation from its table?

Exercise 11.1.5

Define $a * b$ by $|a - b|$, the absolute value of $a - b$. Which properties does $*$ have on the set of natural numbers, $\mathbb{N}$?

Answer

a. $*$ is commutative since $|a - b| = |b - a|$ for all $a,\ b \in N$
b. $*$ is not associative. Take $a = 1$, $b = 2$, and $c = 3$, then $(a * b) * c = ||1 - 2| - 3| = 2$, and $a * (b * c) = |1 - |2 - 3|| = 0$.
c. Zero is the identity for $*$ on $\mathbb{N}$, since $a * 0 = |a - 0| = a = |0 - a| = 0 * a$.
d. Each element of $\mathbb{N}$ inverts itself since $a * a = |a - a| = 0$.
e. $*$ is not idempotent, since, for $a \neq 0$, $a * a = 0 \neq a$.

This page titled 11.1: Operations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.2: Algebraic Systems

An algebraic system is a mathematical system consisting of a set called the domain and one or more operations on the domain. If V is the domain and $*_1, *_2, \ldots, *_n$ are the operations, $[V; *_1, *_2, \ldots, *_n]$ denotes the mathematical system. If the context is clear, this notation is abbreviated to V.

11.2.1: Monoids at Two Levels

Consider the following two examples of algebraic systems.

a. Let B^* be the set of all finite strings of 0's and 1's including the null (or empty) string, λ. An algebraic system is obtained by adding the operation of concatenation. The concatenation of two strings is simply the linking of the two strings together in the order indicated. The concatenation of strings a with b is denoted $a + b$. For example, $01101 + 101 = 01101101$and $\lambda + 100 = 100$. Note that concatenation is an associative operation and that λ is the identity for concatenation.
A note on notation: There isn't a standard symbol for concatenation. We have chosen $+$ to be consistent with the notation used in Python and Sage for the concatenation.
b. Let M be any nonempty set and let * be any operation on M that is associative and has an identity in M. Any such system is called a ***monoid***. We introduce monoids briefly here, but will discuss them further in Chapter 14

Our second example might seem strange, but we include it to illustrate a point. The algebraic system $[B^*; +]$ is a special case of $[M; *]$. Most of us are much more comfortable with B^* than with M. No doubt, the reason is that the elements in B^* are more concrete. We know what they look like and exactly how they are combined. The description of M is so vague that we don't even know what the elements are, much less how they are combined. Why would anyone want to study M? The reason is related to this question: What theorems are of interest in an algebraic system? Answering this question is one of our main objectives in this chapter. Certain properties of algebraic systems are called algebraic properties, and any theorem that says something about the algebraic properties of a system would be of interest. The ability to identify what is algebraic and what isn't is one of the skills that you should learn from this chapter.

Now, back to the question of why we study M. Our answer is to illustrate the usefulness of M with a theorem about M.

Theorem 11.2.1: A Monoid Theorem

If a, b are elements of M and $a * b = b * a$, then $(a * b) * (a * b) = (a * a) * (b * b)$.

Proof

$$\begin{aligned}
(a*b)*(a*b) &= a*(b*(a*b)) && \text{Why?}\\
&= a*((b*a)*b) && \text{Why?}\\
&= a*((a*b)*b) && \text{Why?}\\
&= a*(a*(b*b)) && \text{Why?}\\
&= (a*a)*(b*b) && \text{Why?}
\end{aligned}$$

The power of this theorem is that it can be applied to any algebraic system that M describes. Since B^* is one such system, we can apply Theorem 11.2.1 to any two strings that commute. For example, 01 and 0101. Although a special case of this theorem could have been proven for B^*, it would not have been any easier to prove, and it would not have given us any insight into other special cases of M.

Example 11.2.1: Another Monoid

Consider the set of 2×2 real matrices, $M_{2\times 2}(\mathbb{R})$, with the operation of matrix multiplication. In this context, Theorem 11.2.1 can be interpreted as saying that if $AB = BA$, then $(AB)^2 = A^2B^2$. One pair of matrices that this theorem applies to is $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ and $\begin{pmatrix} 3 & -4 \\ -4 & 3 \end{pmatrix}$.

11.2.2: Levels of Abstraction

One of the fundamental tools in mathematics is abstraction. There are three levels of abstraction that we will identify for algebraic systems: concrete, axiomatic, and universal.

11.2.2.1: Concrete Level

Almost all of the mathematics that you have done in the past was at the concrete level. As a rule, if you can give examples of a few typical elements of the domain and describe how the operations act on them, you are describing a concrete algebraic system. Two examples of concrete systems are B^* and $M_{2\times 2}(\mathbb{R})$. A few others are:

a. The integers with addition. Of course, addition isn't the only standard operation that we could include. Technically, if we were to add multiplication, we would have a different system.
b. The subsets of the natural numbers, with union, intersection, and complementation.
c. The complex numbers with addition and multiplication.

11.2.2.2: Axiomatic Level

The next level of abstraction is the axiomatic level. At this level, the elements of the domain are not specified, but certain axioms are stated about the number of operations and their properties. The system that we called M is an axiomatic system. Some combinations of axioms are so common that a name is given to any algebraic system to which they apply. Any system with the properties of M is called a monoid. The study of M would be called monoid theory. The assumptions that we made about M, associativity and the existence of an identity, are called the monoid axioms. One of your few brushes with the axiomatic level may have been in your elementary algebra course. Many algebra texts identify the properties of the real numbers with addition and multiplication as the field axioms. As we will see in Chapter 16, “Rings and Fields,” the real numbers share these axioms with other concrete systems, all of which are called fields.

11.2.2.3: Universal Level

The final level of abstraction is the universal level. There are certain concepts, called universal algebra concepts, that can be applied to the study of all algebraic systems. Although a purely universal approach to algebra would be much too abstract for our purposes, defining concepts at this level should make it easier to organize the various algebraic theories in your own mind. In this chapter, we will consider the concepts of isomorphism, subsystem, and direct product.

11.2.3: Groups

To illustrate the axiomatic level and the universal concepts, we will consider yet another kind of axiomatic system, the group. In Chapter 5 we noted that the simplest equation in matrix algebra that we are often called upon to solve is $AX = B,$ where A and B are known square matrices and X is an unknown matrix. To solve this equation, we need the associative, identity, and inverse laws. We call the systems that have these properties groups.

Definition 11.2.1: Group

A group consists of a nonempty set G and a binary operation $*$ on G satisfying the properties

a. $*$ is associative on G: $(a*b)*c = a*(b*c)$ for all $a, b, c \in G$.
b. There exists an identity element, $e \in G,$ such that $a*e = e*a = a$ for all $a \in G$.
c. For all $a \in G,$ there exists an inverse; that is, there exists $b \in G$ such that $a*b = b*a = e$.

A group is usually denoted by its set's name, G, or occasionally by $[G; *]$ to emphasize the operation. At the concrete level, most sets have a standard operation associated with them that will form a group. As we will see below, the integers with addition is a group. Therefore, in group theory $\mathbb{Z}$ always stands for $[\mathbb{Z}; +]$.

Note 11.2.1: Generic Symbols

At the axiomatic and universal levels, there are often symbols that have a special meaning attached to them. In group theory, the letter e is used to denote the identity element of whatever group is being discussed. A little later, we will prove that the inverse of a group element, a, is unique and its inverse is usually denoted a^{-1} and is read “a inverse.” When a concrete group is discussed, these symbols are dropped in favor of concrete symbols. These concrete symbols may or may not be similar to the

generic symbols. For example, the identity element of the group of integers is 0, and the inverse of n is denoted by $-n$, the additive inverse of n.

The asterisk could also be considered a generic symbol since it is used to denote operations on the axiomatic level.

Example 11.2.2: Some Concrete Groups

a. The integers with addition is a group. We know that addition is associative. Zero is the identity for addition: $0+n=n+0=n$ for all integers n. The additive inverse of any integer is obtained by negating it. Thus the inverse of n is $-n$.
b. The integers with multiplication is not a group. Although multiplication is associative and 1 is the identity for multiplication, not all integers have a multiplicative inverse in $\mathbb{Z}$. For example, the multiplicative inverse of 10 is $\frac{1}{10}$, but $\frac{1}{10}$ is not an integer.
c. The power set of any set U with the operation of symmetric difference, $\oplus$, is a group. If A and B are sets, then $A \oplus B = (A \cup B) - (A \cap B)$. We will leave it to the reader to prove that $\oplus$ is associative over $\mathcal{P}(U)$. The identity of the group is the empty set: $A \oplus \emptyset = A$. Every set is its own inverse since $A \oplus A = \emptyset$. Note that $\mathcal{P}(U)$ is not a group with union or intersection.

Definition 11.2.2: Abelian Group

A group is abelian if its operation is commutative.

Note 11.2.2: Abel

Most of the groups that we will discuss in this book will be abelian. The term abelian is used to honor the Norwegian mathematician N. Abel (1802-29), who helped develop group theory.



Figure 11.2.1: Norwegian Stamp honoring Abel

11.2.4: Exercises

Exercise 11.2.1

Discuss the analogy between the terms generic and concrete for algebraic systems and the terms generic and trade for prescription drugs.

Answer

The terms “generic” and “trade” for prescription drugs are analogous to “generic” and “concrete” algebraic systems. Generic aspirin, for example, has no name, whereas Bayer, Tylenol, Bufferin, and Anacin are all trade or specific types of aspirins. The same can be said of a generic group $[G; *]$ where G is a nonempty set and $*$ is a binary operation on G, When examples of typical domain elements can be given along with descriptions of how operations act on them, such as $\mathbb{Q}^*$ or $M_{2\times 2}(\mathbb{R})$, then the system is concrete (has a specific name, as with the aspirin). Generic is a way to describe a general algebraic system, whereas a concrete system has a name or symbols making it distinguishable from other systems.

Exercise 11.2.2

Discuss the connection between groups and monoids. Is every monoid a group? Is every group a monoid?

Exercise 11.2.3

Which of the following are groups?

a. B^* with concatenation (see Subsection 11.2.1).
b. $M_{2\times 3}(\mathbb{R})$ with matrix addition.
c. $M_{2\times 3}(\mathbb{R})$ with matrix multiplication.
d. The positive real numbers, $\mathbb{R}^+$, with multiplication.
e. The nonzero real numbers, $\mathbb{R}^*$, with multiplication.
f. $\{1,-1\}$ with multiplication.
g. The positive integers with the operation M defined by $aMb =$ the larger of a and b.

Answer

The systems in parts b, d, e, and f are groups.

Exercise 11.2.4

Prove that, $\oplus$, defined by $A \oplus B = (A \cup B) - (A \cap B)$ is an associative operation on $\mathcal{P}(U)$.

Exercise 11.2.5

The following problem supplies an example of a non-abelian group. A rook matrix is a matrix that has only 0's and 1's as entries such that each row has exactly one 1 and each column has exactly one 1. The term rook matrix is derived from the fact that each rook matrix represents the placement of n rooks on an $n \times n$ chessboard such that none of the rooks can attack one another. A rook in chess can move only vertically or horizontally, but not diagonally. Let R_n be the set of $n \times n$ rook matrices. There are six 3×3 rook matrices:

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad R_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad R_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$F_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad F_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad F_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

a. List the 2×2 rook matrices. They form a group, R_2, under matrix multiplication. Write out the multiplication table. Is the group abelian?
b. Write out the multiplication table for R_3 . This is another group. Is it abelian?
c. How many 4×4 rook matrices are there? How many $n \times n$ rook matrices are there?

Answer

a. Elements are $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, and $T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, the group is abelian. Operation table is

$\cdot$	I	T
I	I	T
T	T	I

b.

	I	R_1	R_2	F_1	F_2	F_3
I	I	R_1	R_2	F_1	F_2	F_3
R_1	R_1	R_2	I	F_2	F_3	F_1
R_2	R_2	I	R_1	F_3	F_1	F_2
F_1	F_1	F	F_2	I	R_2	R_1
F_2	F_2	F_1	F_3	R_1	I	R_2
F_3	F_3	F_2	F_1	R_2	R_1	I

This group is non-abelian since, for example, $F_1F_2 = R_2$ and $F_2F_1 = R_2$.

c. 4! = 24, $n!$.

Exercise 11.2.6

For each of the following sets, identify the standard operation that results in a group. What is the identity of each group?

a. The set of all 2×2 matrices with real entries and nonzero determinants.
b. The set of 2×3 matrices with rational entries.

Exercise 11.2.7

Let $V = \{e, a, b, c\}$. Let $*$ be defined (partially) by $x * x = e$ for all $x \in V$. Write a complete table for $*$ so that $[V; *]$ is a group.

Answer

The identity is e. $a * b = c$, $a * c = b$, $b * c = a$, and $[V; *]$ is abelian. (This group is commonly called the Klein-4 group.)

Exercise 11.2.8

Consider the following set of six algebraic expressions, each defining a function on the set of real numbers excluding the numbers 0 and 1.

$$\mathcal{H} = \left\{x, 1-x, \frac{1}{1-x}, \frac{1}{x}, \frac{x-1}{x}, \frac{x}{x-1}\right\} = \{y_1, y_2, y_3, y_4, y_5, y_6\}$$

We can operate on any two of these expressions using function composition. For example,

$$(y_3 \circ y_4)(x) = y_3(y_4(x)) = y_3\left(\frac{1}{x}\right) = \frac{1}{1-\frac{1}{x}} = \frac{x}{x-1} = y_6(x)$$

Therefore, $y_3 \circ y_4 = y_6$. Complete the following operation table for function composition on $\mathcal{H}$.

$\circ$	y_1	y_2	y_3	y_4	y_5	y_6
y_1	y_1	y_2	y_3	y_4	y_5	y_6
y_2	y_2	y_1	y_6	y_5	y_4	
y_3	y_3	y_4		y_6		
y_4		y_3	y_2			
y_5						
y_6						

Figure 11.2.2: Partially completed operation table for $\mathcal{H}$

Is $[\mathcal{H}, \circ]$ a monoid? Is it a group?

Answer

Yes, this is a group. You might see some similarities with the group of three by three rook matrices.

This page titled 11.2: Algebraic Systems is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.3: Some General Properties of Groups

In this section, we will present some of the most basic theorems of group theory. Keep in mind that each of these theorems tells us something about every group. We will illustrate this point with concrete examples at the close of the section.

11.3.1: First Theorems

Theorem 11.3.1: Identities are Unique

The identity of a group is unique.

One difficulty that students often encounter is how to get started in proving a theorem like this. The difficulty is certainly not in the theorem's complexity. It's too terse! Before actually starting the proof, we rephrase the theorem so that the implication it states is clear.

Theorem 11.3.2: Identities are Unique - Rephrased

If $G = [G; *]$ is a group and e is an identity of G, then no other element of G is an identity of G.

Proof

(Indirect): Suppose that $f \in G$, $f \neq e$, and f is an identity of G. We will show that $f = e$, which is a contradiction, completing the proof.

$$\begin{aligned} f &= f * e \quad \text{Since } e \text{ is an identity} \\ &= e \quad \text{Since } f \text{ is an identity} \end{aligned}$$

Next we justify the phrase "... *the* inverse of an element of a group."

Theorem 11.3.3: Inverses are Unique

The inverse of any element of a group is unique.

The same problem is encountered here as in the previous theorem. We will leave it to the reader to rephrase this theorem. The proof is also left to the reader to write out in detail. Here is a hint: If b and c are both inverses of a, then you can prove that $b = c$. If you have difficulty with this proof, note that we have already proven it in a concrete setting in Chapter 5.

As mentioned above, the significance of Theorem 11.3.3 is that we can refer to *the* inverse of an element without ambiguity. The notation for the inverse of a is usually a^{-1} (note the exception below).

Example 11.3.1: Some Inverses

a. In any group, e^{-1} is the inverse of the identity e, which always is e.
b. $\left(a^{-1}\right)^{-1}$ is the inverse of a^{-1} , which is always equal to a (see Theorem 11.3.4 below).
c. $(x * y * z)^{-1}$ is the inverse of $x * y * z$.
d. In a concrete group with an operation that is based on addition, the inverse of a is usually written $-a$. For example, the inverse of $k - 3$ in the group $[\mathbb{Z}; +]$ is written $-(k - 3) = 3 - k$. In the group of 2×2 matrices over the real numbers under matrix addition, the inverse of $\begin{pmatrix} 4 & 1 \\ 1 & -3 \end{pmatrix}$ is written $-\begin{pmatrix} 4 & 1 \\ 1 & -3 \end{pmatrix}$, which equals $\begin{pmatrix} -4 & -1 \\ -1 & 3 \end{pmatrix}$.

Theorem 11.3.4: Inverse of Inverse Theorem

If a is an element of group G, then $\left(a^{-1}\right)^{-1} = a$.

Again, we rephrase the theorem to make it clear how to proceed.

Theorem 11.3.5: Inverse of Inverse Theorem (Rephrased)

If a has inverse b and b has inverse c, then $a = c$.

Proof

$$\begin{aligned} a &= a * e && e \text{ is the identity of } G \\ &= a * (b * c) && \text{because } c \text{ is the inverse of } b \\ &= (a * b) * c && \text{why?} \\ &= e * c && \text{why?} \\ &= c && \text{by the identity property} \end{aligned}$$

The next theorem gives us a formula for the inverse of $a * b$. This formula should be familiar. In Chapter 5 we saw that if A and B are invertible matrices, then $(AB)^{-1} = B^{-1}A^{-1}$.

Theorem 11.3.6: Inverse of a Product

If a and b are elements of group G, then $(a * b)^{-1} = b^{-1} * a^{-1}$.

Proof

Let $x = b^{-1} * a^{-1}$. We will prove that x inverts $a * b$. Since we know that the inverse is unique, we will have proved the theorem.

$$\begin{aligned} (a * b) * x &= (a * b) * \left(b^{-1} * a^{-1}\right) \\ &= a * \left(b * \left(b^{-1} * a^{-1}\right)\right) \\ &= a * \left(\left(b * b^{-1}\right) * a^{-1}\right) \\ &= a * \left(e * a^{-1}\right) \\ &= a * a^{-1} \\ &= e \end{aligned}$$

Similarly, $x * (a * b) = e$; therefore, $(a * b)^{-1} = x = b^{-1} * a^{-1}$

Theorem 11.3.7: Cancellation Laws

If a, b, and c are elements of group G, then

$$\begin{aligned} &\text{left cancellation:} && (a * b = a * c) \Rightarrow b = c \\ &\text{right cancellation:} && (b * a = c * a) \Rightarrow b = c \end{aligned}$$

Proof

We will prove the left cancellation law. The right law can be proved in exactly the same way. Starting with $a * b = a * c$, we can operate on both $a * b$ and $a * c$ on the left with a^{-1}:

$$a^{-1} * (a * b) = a^{-1} * (a * c)$$

Applying the associative property to both sides we get

$$\begin{aligned} (a^{-1} * a) * b = (a^{-1} * a) * c \quad &\Rightarrow e * b = e * c \\ &\Rightarrow b = c \end{aligned}$$

Theorem 11.3.8: Linear Equations in a Group

If G is a group and $a, b \in G$, the equation $a * x = b$ has a unique solution, $x = a^{-1} * b$. In addition, the equation $x * a = b$ has a unique solution, $x = b * a^{-1}$.

Proof

We prove the theorem only for $a*x=b,$ since the second statement is proven identically.

$$\begin{aligned} a*x=b &= e*b \\ &= (a*a^{-1})*b \\ &= a*(a^{-1}*b) \end{aligned}$$

By the cancellation law, we can conclude that $x=a^{-1}*b$.

If c and d are two solutions of the equation $a*x=b,$ then $a*c=b=a*d$ and, by the cancellation law, $c=d$. This verifies that $a^{-1}*b$ is the only solution of $a*x=b$.

Note 11.3.1

Our proof of Theorem 11.3.8 was analogous to solving the concrete equation $4x=9$ in the following way:

$$4x=9=\left(4\cdot\frac{1}{4}\right)9=4\left(\frac{1}{4}9\right)$$

Therefore, by cancelling 4,

$$x=\frac{1}{4}\cdot 9=\frac{9}{4}$$

11.3.2: Exponents

If a is an element of a group G, then we establish the notation that

$$a*a=a^2 \qquad a*a*a=a^3 \qquad \text{etc.}$$

In addition, we allow negative exponent and define, for example,

$$a^{-2}=\left(a^2\right)^{-1}$$

Although this should be clear, proving exponentiation properties requires a more precise recursive definition.

Definition 11.3.1: Exponentiation in Groups

For $n\geq 0,$ define a^n recursively by $a^0=e$ and if $n>0, a^n=a^{n-1}*a.$ Also, if $n>1,$ $a^{-n}=(a^n)^{-1}$.

Example 11.3.2: Some Concrete Exponentiations

a. In the group of positive real numbers with multiplication,

$$5^3=5^2\cdot 5=\left(5^1\cdot 5\right)\cdot 5=\left(\left(5^0\cdot 5\right)\cdot 5\right)\cdot 5=((1\cdot 5)\cdot 5)\cdot 5=5\cdot 5\cdot 5=125$$

and

$$5^{-3}=(125)^{-1}=\frac{1}{125}$$

b. In a group with addition, we use a different form of notation, reflecting the fact that in addition repeated terms are multiples, not powers. For example, in $[\mathbb{Z};+], a+a$ is written as $2a, a+a+a$ is written as $3a$, etc. The inverse of a multiple of a such as $-(a+a+a+a+a)=-(5a)$ is written as $(-5)a$.

Although we define, for example, $a^5=a^4*a,$ we need to be able to extract the single factor on the left. The following lemma justifies doing precisely that.

Lemma 11.3.1

Let G be a group. If $b \in G$ and $n \geq 0$, then $b^{n+1} = b * b^n,$ and hence $b * b^n = b^n * b$.

Proof

(By induction): If $n = 0$,

$$\begin{aligned} b^1 &= b^0 * b \text{ by the definition of exponentiation} \\ &= e * b \text{ by the basis for exponentiation} \\ &= b * e \text{ by the identity property} \\ &= b * b^0 \text{ by the basis for exponentiation} \end{aligned}$$

Now assume the formula of the lemma is true for some $n \geq 0$.

$$\begin{aligned} b^{(n+1)+1} &= b^{(n+1)} * b \text{ by the definition of exponentiation} \\ &= (b * b^n) * b \text{ by the induction hypothesis} \\ &= b * (b^n * b) \text{ associativity} \\ &= b * \left(b^{n+1}\right) \text{ definition of exponentiation} \end{aligned}$$

Based on the definitions for exponentiation above, there are several properties that can be proven. They are all identical to the exponentiation properties from elementary algebra.

Theorem 11.3.9: Properties of Exponentiation

If a is an element of a group G, and m and n are integers,

1. $a^{-n} = \left(a^{-1}\right)^n$ and hence $\left(a^n\right)^{-1} = \left(a^{-1}\right)^n$
2. $a^{n+m} = a^n * a^m$
3. $\left(a^n\right)^m = a^{nm}$

Proof

We will leave the proofs of these properties to the reader. All three parts can be done by induction. For example the proof of the second part would start by defining the proposition $p(m)$, $m \geq 0$, to be $a^{n+m} = a^n * a^m$ for all n. The basis is $p(0) : a^{n+0} = a^n * a^0$.

Our final theorem is the only one that contains a hypothesis about the group in question. The theorem only applies to finite groups.

Theorem 11.3.10

If G is a finite group, $|G| = n,$ and a is an element of G, then there exists a positive integer m such that $a^m = e$ and $m \leq n$.

Proof

Consider the list $a, a^2, \ldots, a^{n+1}$. Since there are $n+1$ elements of G in this list, there must be some duplication. Suppose that $a^p = a^q,$ with $p < q$. Let $m = q - p$. Then

$$\begin{aligned} a^m &= a^{q-p} \\ &= a^q * a^{-p} \\ &= a^q * \left(a^p\right)^{-1} \\ &= a^q * \left(a^q\right)^{-1} \\ &= e \end{aligned}$$

Furthermore, since $1 \leq p < q \leq n+1, \quad m = q - p \leq n.$

Consider the concrete group $[\mathbb{Z}; +]$. All of the theorems that we have stated in this section except for the last one say something about $\mathbb{Z}$. Among the facts that we conclude from the theorems about $\mathbb{Z}$ are:

- Since the inverse of 5 is -5, the inverse of -5 is 5.
- The inverse of $-6+71$ is $-(71)+-(-6)=-71+6$.
- The solution of $12+x=22$ is $x=-12+22$.
- $-4(6)+2(6)=(-4+2)(6)=-2(6)=-(2)(6)$.
- $7(4(3))=(7\cdot 4)(3)=28(3)$ (twenty-eight 3s).

11.3.3: Exercises

Exercise 11.3.1

Let $[G;*]$ be a group and a be an element of G. Define $f:G\to G$ by $f(x)=a*x$.

a. Prove that f is a bijection.
b. On the basis of part a, describe a set of bijections on the set of integers.

Answer

a. f is injective:

$$\begin{aligned} f(x)=f(y) &\Rightarrow a*x=a*y \\ &\Rightarrow x=y \text{ by left cancellation} \end{aligned}.$$

f is surjective: For all $b\in G$, $f(x)=b$ has the solution $a^{-1}*b$.
b. Functions of the form $f(x)=a+x$, where a is any integer, are bijections

Exercise 11.3.2

Rephrase Theorem 11.3.3 and write out a clear proof.

Exercise 11.3.3

Prove by induction on n that if $a_1,\ a_2,\ldots,a_n$ are elements of a group G, $n\geq 2$, then $(a_1*a_2*\cdots*a_n)^{-1}=a_n^{-1}*\cdots*a_2^{-1}*a_1^{-1}$. Interpret this result in terms of $[\mathbb{Z};+]$ and $[\mathbb{R}^*;\cdot]$.

Answer

Basis: $(n=2)\ (a_1*a_2)^{-1}=a_2^{-1}*a_1^{-1}$ by Theorem 11.3.6

Induction: Assume that for some $n\geq 2$,

$$(a_1*a_2*\cdots*a_n)^{-1}=a_n^{-1}*\cdots*a_2^{-1}*a_1^{-1}$$

We must show that

$$(a_1*a_2*\cdots*a_n*a_{n+1})^{-1}=a_{n+1}^{-1}*a_n^{-1}*\cdots*a_2^{-1}*a_1^{-1}$$

This can be accomplished as follows:

$$\begin{aligned} (a_1*a_2*\cdots*a_n*a_{n+1})^{-1} &= ((a_1*a_2*\cdots*a_n)*a_{n+1})^{-1} \text{ by the associative law} \\ &= a_{n+1}^{-1}*(a_1*a_2*\cdots*a_n)^{-1} \text{ by the basis} \\ &= a_{n+1}^{-1}*\left(a_n^{-1}*\cdots*a_2^{-1}*a_1^{-1}\right) \text{ by the induction hypothesis} \\ &= a_{n+1}^{-1}*a_n^{-1}*\cdots*a_2^{-1}*a_1^{-1} \text{ by the associative law} \end{aligned}$$

Exercise 11.3.4

True or false? If a, b, c are elements of a group G, and $a * b = c * a$, then $b = c$. Explain your answer.

Exercise 11.3.5

Prove Theorem 11.3.9

Answer

In this answer, we will refer to Lemma 11.3.1simply as "the lemma."

a. Let $p(n)$ be $a^{-n} = \left(a^{-1}\right)^n$, where a is any element of group $[G; *]$. First we will prove that $p(n)$ is true for all $n \geq 0$.
Basis: If $n = 0$, Using the definition of the zero exponent, $\left(a^0\right)^{-1} = e^{-1} = e$, while $\left(a^{-1}\right)^0 = e$. Therefore, $p(0)$ is true.
Induction: Assume that for some $n \geq 0$, $p(n)$ is true.

$$\begin{aligned}\left(a^{n+1}\right)^{-1} &= (a^n * a)^{-1} \text{ by the definition of exponentiation}\\ &= a^{-1} * (a^n)^{-1} \text{ by the lemma}\\ &= a^{-1} * \left(a^{-1}\right)^n \text{ by the induction hypothesis}\\ &= \left(a^{-1}\right)^{n+1} \text{ by the lemma}\end{aligned}$$

If n is negative, then $-n$ is positive and

$$\begin{aligned}a^{-n} &= \left(\left(\left(a^{-1}\right)^{-1}\right)^{-n}\right)\\ &= \left(a^{-1}\right)^{-(-n)} \text{ since the property is true for positive numbers}\\ &= \left(a^{-1}\right)^n\end{aligned}$$

b. For $m > 1$, let $p(m)$ be $a^{n+m} = a^n * a^m$ for all $n \geq 1$. The basis for this proof follows directly from the basis for the definition of exponentiation.
Induction: Assume that for some $m > 1$, $p(m)$ is true. Then

$$\begin{aligned}a^{n+(m+1)} &= a^{(n+m)+1} \text{ by the associativity of integer addition}\\ &= a^{n+m} * a^1 \text{ by the definition of exponentiation}\\ &= (a^n * a^m) * a^1 \text{ by the induction hypothesis}\\ &= a^n * \left(a^m * a^1\right) \text{ by associativity}\\ &= a^n * a^{m+1} \text{ by the definition of exponentiation}\end{aligned}$$

To complete the proof, you need to consider the cases where m and/or n are negative.

c. Let $p(m)$be $(a^n)^m = a^{nm}$ for all integers n.
Basis: $(a^m)^0 = e$ and $a^{m \cdot 0} = a^0 = e$ therefore, $p(0)$ is true.
Induction; Assume that $p(m)$ is true for some $m > 0$,

$$\begin{aligned}(a^n)^{m+1} &= (a^n)^m * a^n \text{ by the definition of exponentiation}\\ &= a^{nm} * a^n \text{ by the induction hypothesis}\\ &= a^{nm+n} \text{ by part (b) of this proof}\\ &= a^{n(m+1)}\end{aligned}$$

Finally, if m is negative, we can verify that $(a^n)^m = a^{nm}$ using many of the same steps as the positive case.

Exercise 11.3.6

Each of the following facts can be derived by identifying a certain group and then applying one of the theorems of this section to it. For each fact, list the group and the theorem that are used.

a. $\left(\frac{1}{3}\right) 5$ is the only solution of $3x = 5$.
b. $-(-(-18)) = -18$.
c. If A, B, C are 3×3 matrices over the real numbers, with $A + B = A + C$, then $B = C$.
d. There is only one subset of the natural numbers for which $K \oplus A = A$ for every $A \subseteq N$.

This page titled 11.3: Some General Properties of Groups is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.4: Greatest Common Divisors and the Integers Modulo n

In this section introduce the greatest common divisor operation, and introduce an important family of concrete groups, the integers modulo n.

11.4.1: Greatest Common Divisors

We start with a theorem about integer division that is intuitively clear. We leave the proof as an exercise.

Theorem 11.4.1: The Division Property for Integers

If $m, n \in \mathbb{Z}$, $n > 0$, then there exist two unique integers, q (the quotient) and r (the remainder), such that $m = nq + r$ and $0 \leq r < n$.

Note 11.4.1

The division property says that if m is divided by n, you will obtain a quotient and a remainder, where the remainder is less than n. This is a fact that most elementary school students learn when they are introduced to long division. In doing the division problem $1986 \div 97$, you obtain a quotient of 20 and a remainder of 46. This result could either be written $\frac{1986}{97} = 20 + \frac{46}{97}$ or $1986 = 97 \cdot 20 + 46$. The latter form is how the division property is normally expressed in higher mathematics.

List 11.4.1

We now remind the reader of some interchangeable terminology that is used when $r = 0$, i. e., $a = bq$. All of the following say the same thing, just from slightly different points of view.

divides

b divides a

multiple

a is a multiple of b

factor

b is a factor of a

divisor

b is a divisor of a

We use the notation $b \mid a$ if b divides a.

For example $2 \mid 18$ and $9 \mid 18$, but $4 \nmid 18$.

Caution: Don't confuse the “divides” symbol with the “divided by” symbol. The former is vertical while the latter is slanted. Notice however that the statement $2 \mid 18$ is related to the fact that $18/2$ is a whole number.

Definition 11.4.1: Greatest Common Divisor

Given two integers, a and b, not both zero, the greatest common divisor of a and b is the positive integer $g = \gcd(a, b)$ such that $g \mid a$, $g \mid b$, and

$$c \mid a \text{ and } c \mid b \Rightarrow c \mid g$$

A little simpler way to think of $\gcd(a, b)$ is as the largest positive integer that is a divisor of both a and b. However, our definition is easier to apply in proving properties of greatest common divisors.

For small numbers, a simple way to determine the greatest common divisor is to use factorization. For example if we want the greatest common divisor of 660 and 350, you can factor the two integers: $660 = 2^2 \cdot 3 \cdot 5 \cdot 11$ and $350 = 2 \cdot 5^2 \cdot 7$. Single factors

of 2 and 5 are the only ones that appear in both factorizations, so the greatest common divisor is $2 \cdot 5 = 10$.

Some pairs of integers have no common divisors other than 1. Such pairs are called *relatively prime pairs*.

Definition 11.4.2: Relatively Prime

A pair of integers, a and b, are relatively prime if $\gcd(a, b) = 1$

For example, $128 = 2^7$ and $135 = 3^3 \cdot 5$ are relatively prime. Notice that neither 128 nor 135 are primes. In general, a and b need not be prime in order to be relatively prime. However, if you start with a prime, like 23, for example, it will be relatively prime to everything but its multiples. This theorem, which we prove later generalizes this observation.

Theorem 11.4.2

If p is a prime and a is any integer such that $p \nmid a$ then $\gcd(a, p) = 1$

11.4.2: Euclidean Algorithm

As early as Euclid's time it was known that factorization wasn't the best way to compute greatest common divisors.

The Euclidean Algorithm is based on the following properties of the greatest common divisor.

$$\gcd(a, 0) = a \text{ for } a \neq 0 \tag{11.4.1}$$

$$\gcd(a, b) = \gcd(b, r) \text{ if } b \neq 0 \text{ and } a = bq + r \tag{11.4.2}$$

To compute $\gcd(a, b)$, we divide b into a and get a remainder r such that $0 \leq r < |b|$. By the property above, $\gcd(a, b) = \gcd(b, r)$. We repeat the process until we get zero for a remainder. The last nonzero number that is the second entry in our pairs is the greatest common divisor. This is inevitable because the second number in each pair is smaller than the previous one. Table 11.4.1shows an example of how this calculation can be systematically performed.

Table 11.4.1: A Table to Compute $\gcd(99, 53)$

q	a	b
-	99	53
1	53	46
1	46	7
6	7	4
1	4	3
1	3	1
3	1	0

Here is a Sage computation to verify that $\gcd(99, 53) = 1$. At each line, the value of a is divided by the value of b. The quotient is placed on the next line along with the new value of a, which is the previous b; and the remainder, which is the new value of b. Recall that in Sage, `a%b` is the remainder when dividing `b` into `a` .

```
a=99
b=53
while b>0:
    print('computing gcd of '+str(a)+' and '+str(b))
    [a,b]=[b,a%b]
print('result is '+str(a))
```

Investigation 11.4.1

If you were allowed to pick two numbers less than 100, which would you pick in order to force Euclid to work hardest? Here's a hint: The size of the quotient at each step determines how quickly the numbers decrease.

Solution

If quotient in division is 1, then we get the slowest possible completion. If $a = b + r,$ then working backwards, each remainder would be the sum of the two previous remainders. This described a sequence like the Fibonacci sequence and indeed, the greatest common divisor of two consecutive Fibonacci numbers will take the most steps to reach a final value of 1.

For fixed values of a and b, consider integers of the form $ax + by$ where x and y can be any two integers. For example if a = 36 and b = 27, some of these results are tabulated below with x values along the left column and the y values on top.

x \ y	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
−6	−378	−351	−324	−297	−270	−243	−216	−189	−162	−135	−108	−81	−54
−5	−342	−315	−288	−261	−234	−207	−180	−153	−126	−99	−72	−45	−18
−4	−306	−279	−252	−225	−198	−171	−144	−117	−90	−63	−36	−9	18
−3	−270	−243	−216	−189	−162	−135	−108	−81	−54	−27	0	27	54
−2	−234	−207	−180	−153	−126	−99	−72	−45	−18	9	36	63	90
−1	−198	−171	−144	−117	−90	−63	−36	−9	18	45	72	99	126
0	−162	−135	−108	−81	−54	−27	0	27	54	81	108	135	162
1	−126	−99	−72	−45	−18	9	36	63	90	117	144	171	198
2	−90	−63	−36	−9	18	45	72	99	126	153	180	207	234
3	−54	−27	0	27	54	81	108	135	162	189	216	243	270
4	−18	9	36	63	90	117	144	171	198	225	252	279	306
5	18	45	72	99	126	153	180	207	234	261	288	315	342
6	54	81	108	135	162	189	216	243	270	297	324	351	378

Figure 11.4.1: Linear combinations of 36 and 27

Do you notice any patterns? What is the smallest positive value that you see in this table? How is it connected to 36 and 27?

Theorem 11.4.3

If a and b are positive integers, the smallest positive value of $ax + by$ is the greatest common divisor of a and b, $\gcd(a, b)$.

Proof

If $g = \gcd(a, b),$ since $g \mid a$ and $g \mid b$, we know that $g \mid (ax + by)$ for any integers x and y, so $ax + by$ can't be less than g. To show that g is exactly the least positive value, we show that g can be attained by extending the Euclidean Algorithm. Performing the extended algorithm involves building a table of numbers. The way in which it is built maintains an invariant, and by The Invariant Relation Theorem, Theorem 17.1.2.1. we can be sure that the desired values of x and y are produced.

To illustrate the algorithm, Table $11.4.2$ displays how to compute $\gcd(152, 53)$. In the r column, you will find 152 and 53, and then the successive remainders from division. So each number in r after the first two is the remainder after dividing the number immediately above it into the next number up. To the left of each remainder is the quotient from the division. In this case the third row of the table tells us that $152 = 53 \cdot 2 + 46$. The last nonzero value in r is the greatest common divisor.

Table $11.4.2$ The extended Euclidean algorithm to compute $\gcd(152, 53)$

q	r	s	t
--	152	1	0
--	53	0	1
2	46	1	-2
1	7	-1	3

q	r	s	t
6	4	7	-20
1	3	-8	23
1	1	15	-43
3	0	-53	152

The "s" and "t" columns are new. The values of s and t in each row are maintained so that $152s+53t$ is equal to the number in the r column. Notice that

Table 11.4.3: Invariant in computing $\gcd(152, 53)$

$152 = 152 \cdot 1 + 53 \cdot 0$
$53 = 152 \cdot 0 + 53 \cdot 1$
$46 = 152 \cdot 1 + 53 \cdot (-2)$
$\vdots$
$1 = 152 \cdot 15 + 53 \cdot (-43)$
$0 = 152 \cdot (-53) + 53 \cdot 152$

The next-to-last equation is what we're looking for in the end! The main problem is to identify how to determine these values after the first two rows. The first two rows in these columns will always be the same. Let's look at the general case of computing $\gcd(a, b)$. If the s and t values in rows $i-1$ and $i-2$ are correct, we have

$$(A) \begin{cases} as_{i-2} + bt_{i-2} = r_{i-2} \\ as_{i-1} + bt_{i-1} = r_{i-1} \end{cases}$$

In addition, we know that

$$r_{i-2} = r_{i-1}q_i + r_i \;\Rightarrow\; r_i = r_{i-2} - r_{i-1}q_i$$

If you substitute the expressions for r_{i-1} and r_{i-2} from (A) into this last equation and then collect the a and b terms separately you get

$$r_i = a\left(s_{i-2} - q_i s_{i-1}\right) + b\left(t_{i-2} - q_i t_{i-1}\right)$$

or

$$s_i = s_{i-2} - q_i s_{i-1} \text{ and } t_i = t_{i-2} - q_i t_{i-1}$$

Look closely at the equations for $r_i, s_i,$ and t_i. Their forms are all the same. With a little bit of practice you should be able to compute s and t values quickly.

11.4.3: Modular Arithmetic

We remind you of the relation on the integers that we call Congruence Modulo n, Definition 6.3.7. If two numbers, a and b, differ by a multiple of n, we say that they are congruent modulo n, denoted $a \equiv b \pmod{n}$. For example, $13 \equiv 38 \pmod{5}$ because $13 - 38 = -25$, which is a multiple of 5.

Definition 11.4.3: Modular Addition

If n is a positive integer, we define addition modulo n $(+_n)$ as follows. If $a, b \in \mathbb{Z}$,

$$a +_n b = \text{ the remainder after } a + b \text{ is divided by } n$$

Definition 11.4.4: Modular Multiplication

If n is a positive integer, we define multiplication modulo n ($\times_n$) as follows. If $a, b \in \mathbb{Z}$,

$$a \times_n b = \text{ the remainder after } a \cdot b \text{ is divided by } n$$

Note 11.4.2

a. The result of doing arithmetic modulo n is always an integer between 0 and $n-1$, by the Division Property. This observation implies that $\{0, 1, \ldots, n-1\}$ is closed under modulo n arithmetic.
b. It is always true that $a +_n b \equiv (a+b) \pmod{n}$ and $a \times_n b \equiv (a \cdot b) \pmod{n}$. For example, $4 +_7 5 = 2 \equiv 9 \pmod{7}$ and $4 \times_7 5 = 6 \equiv 20 \pmod{7}$.
c. We will use the notation $\mathbb{Z}_n$ to denote the set $\{0, 1, 2, \ldots, n-1\}$.

Example 11.4.1: Some Examples

a. We are all somewhat familiar with $\mathbb{Z}_{12}$ since the hours of the day are counted using this group, except for the fact that 12 is used in place of 0. Military time uses the mod 24 system and does begin at 0. If someone started a four-hour trip at hour 21, the time at which she would arrive is $21 +_{24} 4 = 1$. If a satellite orbits the earth every four hours and starts its first orbit at hour 5, it would end its first orbit at time $5 +_{24} 4 = 9$. Its tenth orbit would end at $5 +_{24} 10 \times_{24} 4 = 21$ hours on the clock
b. Virtually all computers represent unsigned integers in binary form with a fixed number of digits. A very small computer might reserve seven bits to store the value of an integer. There are only 2^7 different values that can be stored in seven bits. Since the smallest value is 0, represented as 0000000, the maximum value will be $2^7 - 1 = 127$, represented as 1111111. When a command is given to add two integer values, and the two values have a sum of 128 or more, overflow occurs. For example, if we try to add 56 and 95, the sum is an eight-digit binary integer 10010111. One common procedure is to retain the seven lowest-ordered digits. The result of adding 56 and 95 would be $0010111_{\text{two}} = 23 \equiv 56 + 95 \pmod{128}$. Integer arithmetic with this computer would actually be modulo 128 arithmetic.

11.4.4: Properties of Modular Arithmetic

Theorem 11.4.4

If $a \in \mathbb{Z}_n$, $a \neq 0$, then the additive inverse of a is $n - a$.

Proof

$a + (n-a) = n \equiv 0 (\bmod n)$, since $n = n \cdot 1 + 0$. Therefore, $a +_n (n-a) = 0$.

Addition modulo n is always commutative and associative; 0 is the identity for $+_n$ and every element of $\mathbb{Z}_n$ has an additive inverse. These properties can be summarized by noting that for each $n \geq 1$, $[\mathbb{Z}_n; +_n]$ is a group.

Definition 11.4.5: The Additive Group of Integers Modulo n

The Additive Group of Integers Modulo n is the group with domain $\{0, 1, 2, \ldots, n-1\}$ and with the operation of mod n addition. It is denoted as $\mathbb{Z}_n$.

Multiplication modulo n is always commutative and associative, and 1 is the identity for $\times_n$.

Notice that the algebraic properties of $+_n$ and $\times_n$ on $\mathbb{Z}_n$ are identical to the properties of addition and multiplication on $\mathbb{Z}$.

Notice that a group cannot be formed from the whole set $\{0, 1, 2, \ldots, n-1\}$ with mod n multiplication since zero never has a multiplicative inverse. Depending on the value of n there may be other restrictions. The following group will be explored in Exercise 11.4.9.

Definition 11.4.6: The Multiplicative Group of Integers Modulo n

The Multiplicative Group of Integers Modulo n is the group with domain $\{k \in \mathbb{Z} | 1 \leq k \leq n-1 \text{ and } \gcd(n,k) = 1\}$ and with the operation of mod n multiplication. It is denoted as $\mathbb{U}_n$.

Example 11.4.2: Some Operation Tables

Here are examples of operation tables for modular groups. Notice that although 8 is greater than 5, the two groups $\mathbb{U}_5$ and $\mathbb{U}_8$ both have order 4. In the case of $\mathbb{U}_5$, since 5 is prime all of the nonzero elements of $\mathbb{Z}_5$ are included. Since 8 isn't prime we don't include integers that share a common factor with 8, the even integers in this case.

Table 11.4.4: Operation Table for the group $\mathbb{Z}_5$

$+_5$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	\(3

Table 11.4.5: Operation table for the group $\mathbb{U}_5$

$\times_5$	**1**	**2**	**3**	**4**
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Table 11.4.6: Operation table for the group $\mathbb{U}_8$

$\times_8$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	\91\)	7	5
5	5	7	1	3
7	7	5	3	1

11.4.5: SageMath Note - Modular Arithmetic

Sage inherits the basic integer division functions from Python that compute a quotient and remainder in integer division. For example, here is how to divide 561 into 2017 and get the quotient and remainder.

```
a=2017
b=561
[q,r]=[a//b,a%b]
[q,r]
```

In Sage, *gcd* is the greatest common divisor function. It can be used in two ways. For the gcd of 2343 and 4319 we can evaluate the expression $gcd(2343, 4319)$. If we are working with a fixed modulus m that has a value established in your Sage session, the

expression $m.gcd(k)$ to compute the greatest common divisor of m and any integer value k. The extended Euclidean algorithm can also be called upon with $xgcd$:

```
a=2017
b=561
print(gcd(a,b))
print(xgcd(a,b))
```

Sage has some extremely powerful tool for working with groups. The integers modulo n are represented by the expression $Integers(n)$ and the addition and multiplications tables can be generated as follows.

```
R = Integers(6)
print(R.addition_table('elements'))
print(R.multiplication_table('elements'))
```

Once we have assigned R a value of $Integers(6)$, we can do calculations by wrapping $R()$ around the integers 0 through 5. Here is a list containing the mod 6 sum and product, respectively, of 5 and 4:

```
[R(5)+R(4), R(5)*R(4)]
```

Generating the multiplication table for the family of groups $\mathbb{U}_n$ takes a bit more code. Here we restrict the allowed inputs to be integers from 2 to 64.

```
def U_table(n):
    if n.parent()!=2.parent() or n < 2 or n > 64:
        return "input error/out of range"
    R=Integers(n)
    els=[]
    for k in filter(lambda k:gcd(n,k)==1,range(n)):
        els=els+[str(k)]
    return R.multiplication_table(elements=els,names="elements")

U_table(18)
```

11.4.6: Exercises

Exercise 11.4.1

Determine the greatest common divisors of the following pairs of integers without using any computational assistance.

a. $2^3 \cdot 3^2 \cdot 5$ and $2^2 \cdot 3 \cdot 5^2 \cdot 7$
b. $7!$ and $3 \cdot 5 \cdot 7 \cdot 9 \cdot 11 \cdot 13$
c. 19^4 and 19^5
d. 12112 and 0

Answer

a. $2^2 \cdot 3 \cdot 5$
b. $3^2 \cdot 5 \cdot 7$
c. 19^4
d. 12112

Exercise 11.4.2

Find all possible values of the following, assuming that m is a positive integer.

a. $\gcd(m+1, m)$
b. $\gcd(m+2, m)$
c. $\gcd(m+4, m)$

Exercise 11.4.3

Calculate:

a. $7 +_8 3$
b. $7 \times_8 3$
c. $4 \times_8 4$
d. $10 +_{12} 2$
e. $6 \times_8 2 +_8 6 \times_8 5$
f. $6 \times_8 (2 +_8 5)$
g. $3 \times_5 3 \times_5 3 \times_5 3 \equiv 3^4 (\mod 5)$
h. $2 \times_{11} 7$
i. $2 \times_{14} 7$

Answer

a. 2
b. 5
c. 0
d. 0
e. 2
f. 2
g. 1
h. 3
i. 0

Exercise 11.4.4

List the additive inverses of the following elements:

a. 4, 6, 9 in $\mathbb{Z}_{10}$
b. 16, 25, 40 in $\mathbb{Z}_{50}$

Exercise 11.4.5

In the group $\mathbb{Z}_{11}$, what are:

a. 3(4)?
b. 36(4)?
c. How could you efficiently compute $m(4), m \in \mathbb{Z}$?

Answer

a. 1
b. 1
c. $m(4) = r(4),$ where $m = 11q + r,\ 0 \leq r < 11$

Exercise 11.4.6

Prove that $\{1,2,3,4\}$is a group under the operation $\times_5$.

Exercise 11.4.7

A student is asked to solve the following equations under the requirement that all arithmetic should be done in $\mathbb{Z}_2$. List all solutions.

a. $x^2+1=0$.
b. $x^2+x+1=0$.

Answer

Since the solutions, if they exist, must come from $\mathbb{Z}_2$, substitution is the easiest approach.

a. 1 is the only solution, since $1^2+_2 1=0$ and $0^2+_2 1=1$
b. No solutions, since $0^2+_2 0+_2 1=1$, and $1^2+_2 1+_2 1=1$

Exercise 11.4.8

Determine the solutions of the same equations as in Exercise $11.4.5$in $\mathbb{Z}_5$.

Exercise 11.4.9

a. Write out the operation table for $\times_8$ on $\{1,3,5,7\}$,and convince your self that this is a group.
b. Let $\mathbb{U}_n$ be the elements of $\mathbb{Z}_n$ that have inverses with respect to $\times_n$. Convince yourself that $\mathbb{U}_n$ is a group under $\times_n$.
c. Prove that the elements of $\mathbb{U}_n$ are those elements $a\in\mathbb{Z}_n$ such that $\gcd(n,a)=1$. You may use Theorem $11.4.3$in this proof.

Exercise 11.4.10

Prove the division property, Theorem $11.4.1$.

Hint

Prove by induction on m that you can divide any positive integer into m. That is, let $p(m)$ be "For all n greater than zero, there exist unique integers q and r such that . . . ." In the induction step, divide n into $m-n$.

Exercise 11.4.11

Suppose $f:\mathbb{Z}_{17}\to\mathbb{Z}_{17}$ such $f(i)=a\times_{17} i+_{17} b$ where a and b are integer constants. Furthermore, assume that $f(1)=11$ and $f(2)=4$. Find a formula for $f(i)$and also find a formula for the inverse of f.

Answer

The given conditions can be converted to a system of linear equations:

$$f(1)=11\Rightarrow a+_{17} b=11$$
$$f(2)=4\Rightarrow 2\times_{17} a+_{17} b=4$$

If we subtract the first equation from the second, we get $a=4+_{17}(-11)=4+_{17}6=10$. This implies that $b=1$, and $f(i)=10\times+17i+1$. To get a formula for the inverse of f we solve $f(j)=i$ for j, using the fact that the multiplicative inverse of 10 (mod 17) is 12.

$$\begin{aligned} f(j)=i &\Rightarrow 10\times+17j+1=i\\ &\Rightarrow 10\times+17j=i+_{17}16\\ &\Rightarrow j=12\times_{17}(i+_{17}16)\end{aligned}$$

Therefore $f^{-1}(i) = 12 \times_{17} (i +_{17} 16) = 12 \times_{17} i +_{17} 5$.

Exercise 11.4.12

Write out the operation table for mod 10 multiplication on $T = \{0, 2, 4, 6, 8\}$. Is $[T; \times_{10}]$ a monoid? Is it a group?

Answer

This system is a monoid with identity 6 (surprise!). However it is not a group since 0 has no inverse.

This page titled 11.4: Greatest Common Divisors and the Integers Modulo n is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.5: Subsystems

11.5.1: Definition

The subsystem is a fundamental concept of algebra at the universal level.

Definition 11.5.1: Subsystem

If $[V; *_1, \ldots, *_n]$ is an algebraic system of a certain kind and W is a subset of V, then W is a subsystem of V if $[W; *_1, \ldots, *_n]$ is an algebraic system of the same kind as V. The usual notation for "W is a subsystem of V" is $W \leq V$.

Since the definition of a subsystem is at the universal level, we can cite examples of the concept of subsystems at both the axiomatic and concrete level.

Example 11.5.1: Examples of Subsystems

a. (Axiomatic) If $[G; *]$ is a group, and H is a subset of G, then H is a subgroup of G if $[H; *]$ is a group.
b. (Concrete) $U = \{-1, 1\}$ is a subgroup of $[\mathbb{R}^*; \cdot]$. Take the time now to write out the multiplication table of U and convince yourself that $[U; \cdot]$ is a group.
c. (Concrete) The even integers, $2\mathbb{Z} = \{2k : k \text{ is an integer}\}$ is a subgroup of $[\mathbb{Z}; +]$. Convince yourself of this fact.
d. (Concrete) The set of nonnegative integers is not a subgroup of $[\mathbb{Z}; +]$. All of the group axioms are true for this subset except one: no positive integer has a positive additive inverse. Therefore, the inverse property is not true. Note that every group axiom must be true for a subset to be a subgroup.
e. (Axiomatic) If M is a monoid and P is a subset of M, then P is a submonoid of M if P is a monoid.
f. (Concrete) If B^* is the set of strings of 0's and 1's of length zero or more with the operation of concatenation, then two examples of submonoids of B^* are: (i) the set of strings of even length, and (ii) the set of strings that contain no 0's. The set of strings of length less than 50 is not a submonoid because it isn't closed under concatenation. Why isn't the set of strings of length 50 or more a submonoid of B^*?

11.5.2: Subgroups

For the remainder of this section, we will concentrate on the properties of subgroups. The first order of business is to establish a systematic way of determining whether a subset of a group is a subgroup.

Theorem 11.5.1: Subgroup Conditions

To determine whether H, a subset of group $[G; *]$, is a subgroup, it is sufficient to prove:

a. H is closed under $*$; that is, $a, b \in H \Rightarrow a * b \in H$;
b. H contains the identity element for $*$; and
c. H contains the inverse of each of its elements; that is, $a \in H \Rightarrow a^{-1} \in H$.

Proof

Our proof consists of verifying that if the three properties above are true, then all the axioms of a group are true for $[H; *]$. By Condition (a), $*$ can be considered an operation on H. The associative, identity, and inverse properties are the axioms that are needed. The identity and inverse properties are true by conditions (b) and (c), respectively, leaving only the associative property. Since, $[G; *]$ is a group, $a * (b * c) = (a * b) * c$ for all $a, b, c \in G$. Certainly, if this equation is true for all choices of three elements from G, it will be true for all choices of three elements from H, since H is a subset of G.

For every group with at least two elements, there are at least two subgroups: they are the whole group and $\{e\}$. Since these two are automatic, they are not considered very interesting and are called the improper subgroups of the group; $\{e\}$ is sometimes referred to as the trivial subgroup. All other subgroups, if there are any, are called proper subgroups.

We can apply Theorem 11.5.1 at both the concrete and axiomatic levels.

Example 11.5.2: Applying Conditions for a Subgroup

a. (Concrete) We can verify that $2\mathbb{Z} \leq \mathbb{Z}$, as stated in Example 11.5.1. Whenever you want to discuss a subset, you must find some convenient way of describing its elements. An element of $2\mathbb{Z}$ can be described as 2 times an integer; that is, $a \in 2\mathbb{Z}$ is equivalent to $(\exists k)_{\mathbb{Z}}(a = 2k)$. Now we can verify that the three conditions of Theorem 11.5.1 are true for $2\mathbb{Z}$. First, if $a, b \in 2\mathbb{Z}$, then there exist $j, k \in \mathbb{Z}$ such that $a = 2j$ and $b = 2k$. A common error is to write something like $a = 2j$ and $b = 2j$. This would mean that $a = b$, which is not necessarily true. That is why two different variables are needed to describe a and b. Returning to our proof, we can add a and b: $a + b = 2j + 2k = 2(j + k)$. Since $j + k$ is an integer, $a + b$ is an element of $2\mathbb{Z}$. Second, the identity, 0, belongs to $2\mathbb{Z}$ $(0 = 2(0))$. Finally, if $a \in 2\mathbb{Z}$ and $a = 2k, -a = -(2k) = 2(-k)$, and $-k \in \mathbb{Z}$, therefore, $-a \in 2\mathbb{Z}$. By Theorem 11.5.1, $2\mathbb{Z} \leq \mathbb{Z}$. How would this argument change if you were asked to prove that $3\mathbb{Z} \leq \mathbb{Z}$? or $n\mathbb{Z} \leq \mathbb{Z}, n \geq 2$?

b. (Concrete) We can prove that $H = \{0, 3, 6, 9\}$ is a subgroup of $\mathbb{Z}_{12}$. First, for each ordered pair $(a, b) \in H \times H$, $a +_{12} b$ is in H. This can be checked without too much trouble since $|H \times H| = 16$. Thus we can conclude that H is closed under $+_{12}$. Second, $0 \in H$. Third, $-0 = 0$, $-3 = 9$, $-6 = 6$, and $-9 = 3$. Therefore, the inverse of each element of H is in H.

c. (Axiomatic) If H and K are both subgroups of a group G, then $H \cap K$ is a subgroup of G. To justify this statement, we have no concrete information to work with, only the facts that $H \leq G$ and $K \leq G$. Our proof that $H \cap K \leq G$ reflects this and is an exercise in applying the definitions of intersection and subgroup, (i) If a and b are elements of $H \cap K$, then a and b both belong to H, and since $H \leq G$, $a * b$ must be an element of H. Similarly, $a * b \in K$; therefore, $a * b \in H \cap K$. (ii) The identity of G must belong to both H and K; hence it belongs to $H \cap K$. (iii) If $a \in H \cap K$, then $a \in H$, and since $H \leq G$, $a^{-1} \in H$. Similarly, $a^{-1} \in K$. Hence, by the theorem, $H \cap K \leq G$. Now that this fact has been established, we can apply it to any pair of subgroups of any group. For example, since $2\mathbb{Z}$ and $3\mathbb{Z}$ are both subgroups of $[\mathbb{Z}; +]$, $2\mathbb{Z} \cap 3\mathbb{Z}$ is also a subgroup of $\mathbb{Z}$. Note that if $a \in 2\mathbb{Z} \cap 3\mathbb{Z}$, a must have a factor of 3; that is, there exists $k \in \mathbb{Z}$ such that $a = 3k$. In addition, a must be even, therefore k must be even. There exists $j \in \mathbb{Z}$ such that $k = 2j$, therefore $a = 3(2j) = 6j$. This shows that $2\mathbb{Z} \cap 3\mathbb{Z} \subseteq 6\mathbb{Z}$. The opposite containment can easily be established; therefore, $2\mathbb{Z} \cap 3\mathbb{Z} = 6\mathbb{Z}$.

Given a finite group, we can apply Theorem 11.3.10 to obtain a simpler condition for a subset to be a subgroup.

Theorem 11.5.2: Condition for a Subgroup of Finite Group

Given that $[G; *]$ is a finite group and H is a nonempty subset of G, if H is closed under $*$, then H is a subgroup of G.

Proof

In this proof, we demonstrate that Conditions (b) and (c) of Theorem 11.5.1 follow from the closure of H under $*$, which is condition (a) of the theorem. First, select any element of H; call it β. The powers of β : $\beta^1, \beta^2, \beta^3, \ldots$ are all in H by the closure property. By Theorem 11.3.10, there exists $m, m \leq |G|$, such that $\beta^m = e$; hence $e \in H$. To prove that (c) is true, we let a be any element of H. If $a = e$, then a^{-1} is in H since $e^{-1} = e$. If $a \neq e$, $a^q = e$ for some q between 2 and $|G|$ and

$$e = a^q = a^{q-1} * a$$

Therefore, $a^{-1} = a^{q-1}$, which belongs to H since $q - 1 \geq 1$.

11.5.3: Note - Applying the condition for a subgroup of a finite group

To determine whether $H_1 = \{0, 5, 10\}$ and $H_2 = \{0, 4, 8, 12\}$ are subgroups of $\mathbb{Z}_{15}$, we need only write out the addition tables (modulo 15) for these sets. This is easy to do with a bit of Sage code that we include below and then for any modulus and subset, we can generate the body of an addition table. The code is set up for H_1 but can be easily adjusted for H_2.

```
def addition_table(n,H):
    for a in H:
        line=[]
        for b in H:
            line+=[(a+b)%n]
        print(line)
```

```
addition_table(15,Set([0,5,10]))
```

Note that H_1 is a subgroup of $\mathbb{Z}_{15}$. Since the interior of the addition table for H_2 contains elements that are outside of H_2, H_2 is not a subgroup of $\mathbb{Z}_{15}$.

11.5.4: Cyclic Subgroups

One kind of subgroup that merits special mention due to its simplicity is the cyclic subgroup.

Definition 11.5.2: Cyclic Subgroup

If G is a group and $a \in G$, the cyclic subgroup generated by a, $\langle a \rangle$, is the set of all powers of a:

$$\langle a \rangle = \{a^n : n \in \mathbb{Z}\}.$$

We refer to a as a generator of subgroup $\langle a \rangle$.

A subgroup H of a group G is cyclic if there exists $a \in H$ such that $H = \langle a \rangle$.

Note 11.5.1

If the operation on G is additive, then $\langle a \rangle = \{(n)a : n \in \mathbb{Z}\}$.

Definition 11.5.3: Order of a Group Element

The order of an element a of group G is the number of elements in the cyclic subgroup of G generated by a. The order of a is denoted $ord(a)$.

Example 11.5.3

a. In $[\mathbb{R}^*; \cdot], \langle 2 \rangle = \{2^n : n \in \mathbb{Z}\} = \left\{\ldots, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 16, \ldots\right\}$.
b. In $\mathbb{Z}_{15}$, $\langle 6 \rangle = \{0, 3, 6, 9, 12\}$.If G is finite, you need list only the positive powers (or multiples) of a up to the first occurrence of the identity to obtain all of $\langle a \rangle$. In $\mathbb{Z}_{15}$, the multiples of 6 are 6, $(2)6 = 12$, $(3)6 = 3$, $(4)6 = 9$, and $(5)6 = 0$. Note that $\{0, 3, 6, 9, 12\}$is also $\langle 3 \rangle$,$\langle 9 \rangle$, and $\langle 12 \rangle$. This shows that a cyclic subgroup can have different generators.

If you want to list the cyclic subgroups of a group, the following theorem can save you some time.

Theorem 11.5.3

If a is an element of group G, then $\langle a \rangle = \langle a^{-1} \rangle$.

This is an easy way of seeing, for example, that $\langle 9 \rangle$ in $\mathbb{Z}_{15}$ equals $\langle 6 \rangle$, since $-6 = 9$.

11.5.5: Exercises

Exercise 11.5.1

Which of the following subsets of the real numbers is a subgroup of $[\mathbb{R}; +]$?

a. the rational numbers
b. the positive real numbers
c. $\{k/2 \mid k \text{ is an integer}\}$
d. $\{2^k \mid k \text{ is an integer}\}$
e. $\{x \mid -100 \leq x \leq 100\}$

Answer

Only a and c are subgroups.

Exercise 11.5.2

Describe in simpler terms the following subgroups of $\mathbb{Z}$:

a. $5\mathbb{Z} \cap 4\mathbb{Z}$
b. $4\mathbb{Z} \cap 6\mathbb{Z}$ (be careful)
c. the only finite subgroup of $\mathbb{Z}$

Exercise 11.5.3

Find at least two proper subgroups of R_3, the set of 3×3 rook matrices (see Exercise 11.2.5).

Answer

$\{I, R_1, R_2\}, \{I, F_1\}, \{I, F_2\}$, and $\{I, F_3\}$ are all the proper subgroups of R_3.

Exercise 11.5.4

Where should you place the following in Figure 11.5.1?

a. e
b. a^{-1}
c. $x * y$

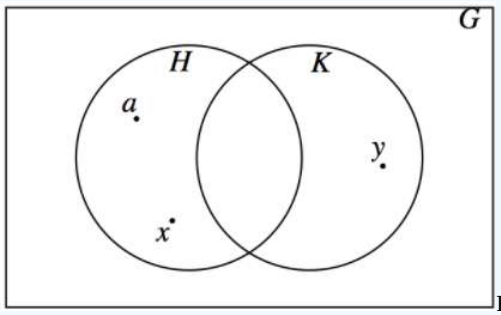


Figure 11.5.1: Figure for Exercise 11.5.4

Exercise 11.5.5

a. List the cyclic subgroups of $\mathbb{Z}_6$ and draw an ordering diagram for the relation "is a subset of" on these subgroups.
b. Do the same for $\mathbb{Z}_{12}$.
c. Do the same for $\mathbb{Z}_8$.
d. On the basis of your results in parts a, b, and c, what would you expect if you did the same with $\mathbb{Z}_{24}$?

Answer

a. $\langle 1\rangle = \langle 5\rangle = \mathbb{Z}_6, \quad \langle 2\rangle = \langle 4\rangle = \{2, 4, 0\}, \langle 3\rangle = \{3, 0\}, \langle 0\rangle = \{0\}$
b. $\langle 1\rangle = \langle 5\rangle = \langle 7\rangle = \langle 11\rangle = \mathbb{Z}_{12}, \quad \langle 2\rangle = \langle 10\rangle = \{2, 4, 6, 8, 10, 0\}, \langle 3\rangle = \langle 9\rangle = \{3, 6, 9, 0\}, \langle 4\rangle = \langle 8\rangle = \{4, 8, 0\}, \langle 6\rangle = \{6, 0\}, \langle 0\rangle = \{0\}$
c. $\langle 1\rangle = \langle 3\rangle = \langle 5\rangle = \langle 7\rangle = \mathbb{Z}_8, \quad \langle 2\rangle = \langle 6\rangle = \{2, 4, 6, 0\}, \langle 4\rangle = \{4, 0\}, \langle 0\rangle = \{0\}$
d. Based on the ordering diagrams for parts a through c in Figure 11.5.2 we would expect to see an ordering diagram similar to the one for divides on $\{1, 2, 3, 4, 6, 8, 12, 24\}$ (the divisors of 24) if we were to examine the subgroups of $\mathbb{Z}_{24}$. This is indeed the case.

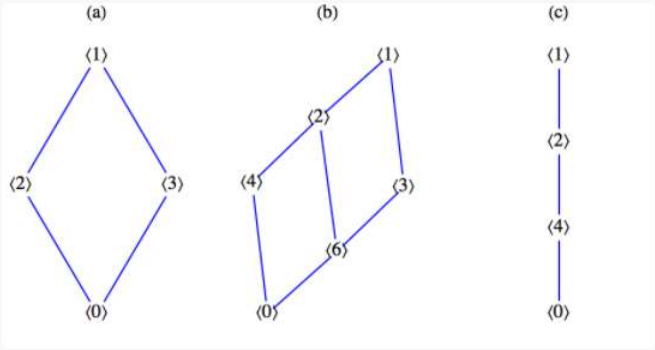


Figure 11.5.2: Figure for Exercise 11.5.5

Exercise 11.5.6: Subgroups Generated by Subsets of a Group

The concept of a cyclic subgroup is a special case of the concept that we will discuss here. Let $[G;*]$ be a group and S a nonempty subset of G. Define the set $\langle S\rangle$ recursively by:

- If $a \in S$, then $a \in \langle S\rangle$.
- If $a, b \in \langle S\rangle$, then $a * b \in \langle S\rangle$, and
- If $a \in \langle S\rangle$, then $a^{-1} \in \langle S\rangle$.

a. By its definition, $\langle S\rangle$ has all of the properties needed to be a subgroup of G. The only thing that isn't obvious is that the identity of G is in $\langle S\rangle$. Prove that the identity of G is in $\langle S\rangle$.
b. What is $\langle\{9, 15\}\rangle$ in $[\mathbb{Z}; +]$?
c. Prove that if $H \leq G$ and $S \subseteq H$, then $\langle S\rangle \leq H$. This proves that $\langle S\rangle$ is contained in every subgroup of G that contains S; that is, $\langle S\rangle = \bigcap_{S\subseteq H, H\leq G} H$.
d. Describe $\langle\{0.5, 3\}\rangle$ in $\left[\mathbb{R}^+; \cdot\right]$ and in $[\mathbb{R}; +]$.
e. If $j, k \in \mathbb{Z}$, $\langle\{j, k\}\rangle$ is a cyclic subgroup of $\mathbb{Z}$. In terms of j and k, what is a generator of $\langle\{j, k\}\rangle$?

Exercise 11.5.7

Prove that if $H, K \leq G$, and $H \cup K = G$, then $H = G$ or $K = G$.

Hint

Use an indirect argument.

Answer

Assume that H and K are subgroups of group G, and that, as in Figure 11.5.1, there are elements $x \in H - K$ and $y \in K - H$. Consider the product $x * y$. Where could it be placed in the Venn diagram? If we can prove that it must lie in the outer region, $H^c \cap K^c = (H \cup K)^c$, then we have proven that $H \cup K$ is not closed under $*$ and cannot be a subgroup of G, Assume that $x * y \in H$. Since x is in H, x^{-1} is in H and so by closure $x^{-1} * (x * y) = y \in H$ which is a contradiction. Similarly, $x * y \notin K$.

One way to interpret this theorem is that no group is the union of two groups.

Exercise 11.5.8

Prove that the order of an element, a of a group is the least positive integer, k, such that a^k is the identity of the group.

This page titled 11.5: Subsystems is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.6: Direct Products

11.6.1: Definition

Our second universal algebraic concept lets us look in the opposite direction from subsystems. Direct products allow us to create larger systems. In the following definition, we avoid complicating the notation by not specifying how many operations the systems have.

Definition 11.6.1: Direct Product

If $[V_i; *_i, \diamond_i, \ldots]$, $i = 1, 2, \ldots, n$ are algebraic systems of the same kind, then the direct product of these systems is $V = V_1 \times V_2 \times \cdots \times V_n$, with operations defined below. The elements of V are n-tuples of the form $(a_1, a_2, \ldots, a_n)$, where $a_k \in V_k$, $k = 1, \ldots, n$. The systems $V_1, V_2, \ldots, V_n$ are called the factors of V. There are as many operations on V as there are in the factors. Each of these operations is defined componentwise:

If $(a_1, a_2, \ldots, a_n), (b_1, b_2, \ldots, b_n) \in V$,

$$(a_1, a_2, \ldots, a_n) * (b_1, b_2, \ldots, b_n) = (a_1 *_1 b_1, a_2 *_2 b_2, \ldots, a_n *_n b_n)$$
$$(a_1, a_2, \ldots, a_n) \diamond (b_1, b_2, \ldots, b_n) = (a_1 \diamond_1 b_1, a_2 \diamond_2 b_2, \ldots, a_n \diamond_n b_n)$$
$$\text{etc.}$$

Example 11.6.1: A Direct Product of Monoids

Consider the monoids $\mathbb{N}$ (the set of natural numbers with addition) and B^* (the set of finite strings of 0's and 1's with concatenation). The direct product of $\mathbb{N}$ with B^* is a monoid. We illustrate its operation, which we will denote by $*$, with examples:

$$(4, 001) * (3, 11) = (4 + 3, 001 + 11) = (7, 00111)$$
$$(0, 11010) * (3, 01) = (3, 1101001)$$
$$(0, \lambda) * (129, 00011) = (0 + 129, \lambda + 00011) = (129, 00011)$$
$$(2, 01) * (8, 10) = (10, 0110), \text{ and}$$
$$(8, 10) * (2, 01) = (10, 1001)$$

Note that our new monoid is not commutative. What is the identity for $*$?

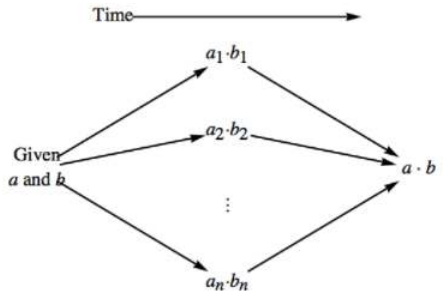


Figure 11.6.1: Concurrent calculation in a direct product

Note 11.6.1

a. On notation. If two or more consecutive factors in a direct product are identical, it is common to combine them using exponential notation. For example, $\mathbb{Z} \times \mathbb{Z} \times \mathbb{R}$ can be written $\mathbb{Z}^2 \times \mathbb{R}$, and $\mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ can be written $\mathbb{R}^4$. This is purely a notational convenience; no exponentiation is really taking place.

b. We call the operations in a direct product ***componentwise operations***, and they are indeed operations on V. If two n-tuples, a and b, are selected from V, the first components of a and b, a_1 and b_1, are operated on with $*_1$ to obtain $a_1 *_1 b_1$, the first component of $a * b$. Note that since $*_1$ is an operation on V_1, $a_1 *_1 b_1$ is an element of V_1. Similarly, all other components of $a * b$, as they are defined, belong to their proper sets.

c. One significant fact about componentwise operations is that the components of the result can all be computed at the same time (concurrently). The time required to compute in a direct product can be reduced to a length of time that is not much longer than the maximum amount of time needed to compute in the factors.
d. A direct product of algebraic systems is not always an algebraic system of the same type as its factors. This is due to the fact that certain axioms that are true for the factors may not be true for the set of n-tuples. This situation does not occur with groups however. You will find that whenever a new type of algebraic system is introduced, call it type T, one of the first theorems that is usually proven, if possible, is that the direct product of two or more systems of type T is a system of type T.

11.6.2: Direct Products of Groups

Theorem 11.6.1: The Direct Product of Groups is a Group

The direct product of two or more groups is a group; that is, the algebraic properties of a system obtained by taking the direct product of two or more groups includes the group axioms.

Proof

We will only present the proof of this theorem for the direct product of two groups. Some slight revisions can be made to produce a proof for any number of factors.

Stating that the direct product of two groups is a group is a short way of saying that if $[G_1; *_1]$ and $[G_2; *_2]$ are groups, then $[G_1 \times G_2; *]$ is also a group, where $*$ is the componentwise operation on $G_1 \times G_2$. Associativity of $*$: If $a, b, c \in G_1 \times G_2$,

$$\begin{aligned} a*(b*c) &= (a_1, a_2)*((b_1, b_2)*(c_1, c_2)) \\ &= (a_1, a_2)*(b_1 *_1 c_1, b_2 *_2 c_2) \\ &= (a_1 *_1 (b_1 *_1 c_1), a_2 *_2 (b_2 *_2 c_2)) \\ &= ((a_1 *_1 b_1) *_1 c_1, (a_2 *_2 b_2) *_2 c_2) \\ &= (a_1 *_1 b_1, a_2 *_2 b_2)*(c_1, c_2) \\ &= ((a_1, a_2)*(b_1, b_2))*(c_1, c_2) \\ &= (a*b)*c \end{aligned}$$

Notice how the associativity property hinges on the associativity in each factor. An identity for $*$: As you might expect, if e_1 and e_2 are identities for G_1 and G_2, respectively, then $e = (e_1, e_2)$ is the identity for $G_1 \times G_2$. If $a \in G_1 \times G_2$,

$$\begin{aligned} a*e &= (a_1, a_2)*(e_1, e_2) \\ &= (a_1 *_1 e_1, a_2 *_2 e_2) \\ &= (a_1, a_2) = a \end{aligned}$$

Similarly, $e*a = a$.

Inverses in $G_1 \times G_2$: The inverse of an element is determined componentwise $a^{-1} = (a_1, a_2)^{-1} = \left({a_1}^{-1}, {a_2}^{-1}\right)$. To verify, we compute $a*a^{-1}$:

$$\begin{aligned} a*a^{-1} &= (a_1, a_2)*\left({a_1}^{-1}, {a_2}^{-1}\right) \\ &= \left(a_1 *_1 {a_1}^{-1}, a_2 *_2 {a_2}^{-1}\right) \\ &= (e_1, e_2) = e \end{aligned}$$

Similarly, $a^{-1}*a = e$.

Example 11.6.2: Some New Groups

a. If $n \geq 2$, ${\mathbb{Z}_2}^n$, the direct product of n factors of $\mathbb{Z}_2$, is a group with 2^n elements. We will take a closer look at ${\mathbb{Z}_2}^3 = \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$. The elements of this group are triples of zeros and ones. Since the operation on $\mathbb{Z}_2$ is $+_2$, we will use the symbol + for the operation on ${\mathbb{Z}_2}^3$. Two of the eight triples in the group are $a = (1, 0, 1)$ and $b = (0, 0, 1)$. Their "sum" is $a + b = (1 +_2 0, 0 +_2 0, 1 +_2 1) = (1, 0, 0)$. One interesting fact about this group is that each element is its own

inverse. For example $a+a=(1,0,1)+(1,0,1)=(0,0,0)$; therefore $-a=a$. We use the additive notation for the inverse of a because we are using a form of addition. Note that $\{(0,0,0),(1,0,1)\}$is a subgroup of ${\mathbb{Z}_2}^3$. Write out the "addition" table for this set and apply Theorem 11.5.2. The same can be said for any set consisting of (0, 0, 0) and another element of ${\mathbb{Z}_2}^3$.

b. The direct product of the positive real numbers with the integers modulo 4, $\mathbb{R}^+ \times \mathbb{Z}_4$ is an infinite group since one of its factors is infinite. The operations on the factors are multiplication and modular addition, so we will select the neutral symbol $\diamond$ for the operation on $\mathbb{R}^+ \times \mathbb{Z}_4$. If $a=(4,3)$ and $b=(0.5,2)$, then

$$a \diamond b=(4,3)\diamond(0.5,2)=(4\cdot 0.5, 3+_4 2)=(2,1)$$
$$b^2=b\diamond b=(0.5,2)\diamond(0.5,2)=(0.25,0)$$
$$a^{-1}=\left(4^{-1},-3\right)=(0.25,1)$$
$$b^{-1}=\left(0.5^{-1},-2\right)=(2,2)$$

It would be incorrect to say that $\mathbb{Z}_4$ is a subgroup of $\mathbb{R}^+ \times \mathbb{Z}_4$, but there is a subgroup of the direct product that closely resembles $\mathbb{Z}_4$. It is $\{(1,0),(1,1),(1,2),(1,3)\}$.Its table is

$\diamond$	$(1,0)$	$(1,1)$	$(1,2)$	$(1,3)$
$(1,0)$	$(1,0)$	$(1,1)$	$(1,2)$	$(1,3)$
$(1,1)$	$(1,1)$	$(1,2)$	$(1,3)$	$(1,0)$
$(1,2)$	$(1,2)$	$(1,3)$	$(1,0)$	$(1,1)$
$(1,3)$	$(1,3)$	$(1,0)$	$(1,1)$	$(1,2)$

Imagine erasing $(1,)$ throughout the table and writing $+_4$ in place of $\diamond$. What would you get? We will explore this phenomenon in detail in the next section.
The whole direct product could be visualized as four parallel half-lines labeled 0, 1, 2, and 3 as in Figure 11.6.2 On the kth line, the point that lies x units to the right of the zero mark would be (x,k). The set $\{(2^n,(n)1) \mid n \in \mathbb{Z}\}$, which is depicted on the figure is a subgroup of $\mathbb{R}^+ \times \mathbb{Z}_4$. What cyclic subgroup is it?
The answer: $\langle(2,1)\rangle$ or $\langle(1/2,3)\rangle$.There are two different generators.

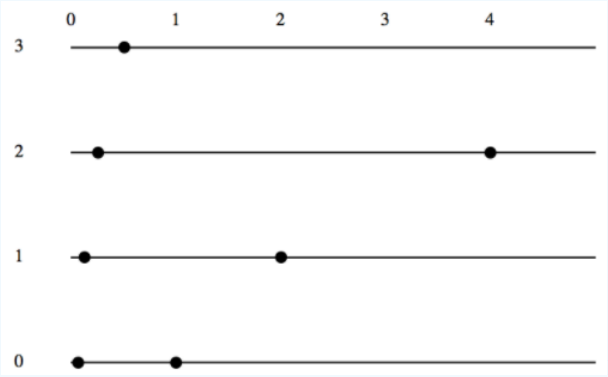


Figure 11.6.2: Visualization of the group $\mathbb{R}^+ \times \mathbb{Z}_4$

A more conventional direct product is $\mathbb{R}^2$, the direct product of two factors of $[\mathbb{R};+]$. The operation on $\mathbb{R}^2$ is componentwise addition; hence we will use + as the operation symbol for this group. You should be familiar with this operation, since it is identical to addition of 2×1 matrices. The Cartesian coordinate system can be used to visualize $\mathbb{R}^2$ geometrically. We plot the pair (s,t) on the plane in the usual way: s units along the x axis and t units along the y axis. There is a variety of different subgroups of $\mathbb{R}^2$, a few of which are:

a. $\{(x,0) \mid x \in \mathbb{R}\}$,all of the points on the x axis;
b. $\{(x,y) \mid 2x-y=0\}$, all of the points that are on the line 2x - y = 0;
c. If $a,b \in \mathbb{R}$, $\{(x,y) \mid ax+by=0\}$. The first two subgroups are special cases of this one, which represents any line that passes through the origin.
d. $\{(x,y) \mid 2x-y=k, k \in \mathbb{Z}\}$, a union of a set of lines that are parallel to $2x-y=0$.

e. $\{(n, 3n) \mid n \in \mathbb{Z}\}$, which is the only countable subgroup that we have listed.

We will leave it to the reader to verify that these sets are subgroups. We will only point out how the fourth example, call it H, is closed under "addition." If $a = (p, q)$ and $b = (s, t)$ and both belong to H, then $2p - q = j$ and $2s - t = k$, where both j and k are integers. $a + b = (p, q) + (s, t) = (p + s, q + t)$ We can determine whether $a + b$ belongs to H by deciding whether or not $2(p + s) - (q + t)$ is an integer:

$$\begin{aligned} 2(p+s)-(q+t) &= 2p+2s-q-t \\ &= (2p-q)+(2s-t) \\ &= j+k \end{aligned}$$

Since j and k are integers, so is $j + k$. This completes a proof that H is closed under the operation of $\mathbb{R}^2$.

Several useful facts can be stated in regards to the direct product of two or more groups. We will combine them into one theorem, which we will present with no proof. Parts a and c were derived for $n = 2$ in the proof of Theorem 11.6.1.

Theorem 11.6.2: Properties of Direct Products of Groups

If $G = G_1 \times G_2 \times \cdots \times G_n$ is a direct product of n groups and $(a_1, a_2, \ldots, a_n) \in G$, then:

a. The identity of G is $(e_1, e_2, \ldots, e_n)$, where e_k is the identity of G_k.
b. $(a_1, a_2, \ldots, a_n)^{-1} = \left({a_1}^{-1}, {a_2}^{-1}, \ldots, {a_n}^{-1}\right)$.
c. $(a_1, a_2, \ldots, a_n)^m = ({a_1}^m, {a_2}^m, \ldots, {a_n}^m)$ for all $m \in \mathbb{Z}$.
d. G is abelian if and only if each of the factors $G_1, G_2, \ldots, G_n$ is abelian.
e. If $H_1, H_2, \ldots, H_n$ are subgroups of the corresponding factors, then $H_1 \times H_2 \times \cdots \times H_n$ is a subgroup of G.

Not all subgroups of a direct product can be created using part e of Theorem 11.6.2 For example, $\{(n, n) \mid n \in \mathbb{Z}\}$ is a subgroup of $\mathbb{Z}^2$, but is not a direct product of two subgroups of $\mathbb{Z}$.

Example 11.6.3: Linked Lists using a Direct Product

Using the identity $(x + y) + x = y$, in $\mathbb{Z}_2$, we can devise a scheme for representing a symmetrically linked list using only one link field. A symmetrically linked list is a list in which each node contains a pointer to its immediate successor and its immediate predecessor (see Figure 11.6.3). If the pointers are n-digit binary addresses, then each pointer can be taken as an element of ${\mathbb{Z}_2}^n$. Lists of this type can be accomplished using cells with only one link. In place of a left and a right pointer, the only "link" is the value of the sum (left link) + (right link). All standard list operations (merge, insert, delete, traverse, and so on) are possible with this structure, provided that you know the value of the nil pointer and the address, f, of the first (i. e., leftmost) cell. Since first f. left is nil, we can recover f. right by adding the value of nil: $f + \text{ nil} = (\text{ nil} + f.\text{right}) + \text{nil} = f.\text{right}$, which is the address of the second item. Now if we temporarily retain the address, s, of the second cell, we can recover the address of the third item. The link field of the second item contains the sum $s.\text{ left} + s.\text{ right} = \text{ first} + \text{ third}$. Therefore

$$\begin{aligned} (\text{first}+\text{third})+\text{first} &= s+s.\text{left} \\ &= (s.\text{left}+s.\text{right})+s.\text{left} \\ &= s.\text{right} \\ &= \text{third} \end{aligned}$$

We no longer need the address of the first cell, only the second and third, to recover the fourth address, and so forth.

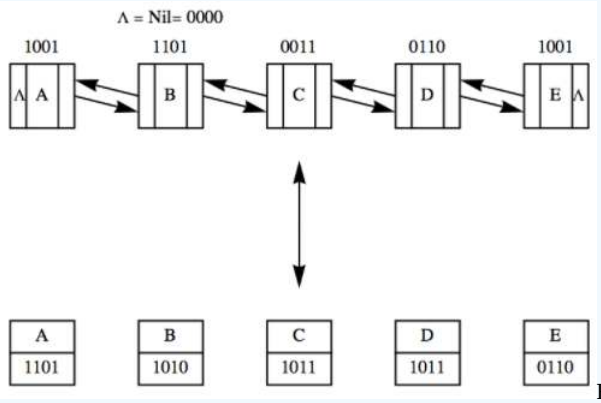


Figure 11.6.3: Symmetric Linked Lists

The following more formal algorithm uses names that reflects the timing of the visits.

Given a symmetric list, a traversal of the list is accomplished as follows, where *first* is the address of the first cell. We presume that each item has some information that is represented by item. info and a field called item.link that is the sum of the left and right links.

Table 11.6.1

(1) yesterday =nil
(2) today =first
(3) while today $\neq$ nil:
(3.1)Write(today.info)
(3.2)tomorrow = today.link + yesterday
(3.3)yesterday = today
(3.4)today = tomorrow.

At any point in this algorithm it would be quite easy to insert a cell between today and tomorrow. Can you describe how this would be accomplished?

11.6.3: Exercises

Exercise 11.6.1

Write out the group table of $\mathbb{Z}_2 \times \mathbb{Z}_3$ and find the two proper subgroups of this group.

Answer

Table of $\mathbb{Z}_2 \times \mathbb{Z}_3$:

$+$	$(0,0)$	$(0,1)$	$(0,2)$	$(1,0)$	$(1,1)$	$(1,2)$
$(0,0)$	$(0,0)$	$(0,1)$	$(0,2)$	$(1,0)$	$(1,1)$	$(1,2)$
$(0,1)$	$(0,1)$	$(0,2)$	$(0,0)$	$(1,1)$	$(1,2)$	$(1,0)$
$(0,2)$	$(0,2)$	$(0,0)$	$(0,1)$	$(1,2)$	$(1,0)$	$(1,1)$
$(1,0)$	$(1,0)$	$(1,1)$	$(1,2)$	$(0,0)$	$(0,1)$	$(0,2)$
$(1,1)$	$(1,1)$	$(1,2)$	$(1,0)$	$(0,1)$	$(0,2)$	$(0,0)$
$(1,2)$	$(1,2)$	$(1,0)$	$(1,1)$	$(0,2)$	$(0,0)$	$(0,1)$

The only two proper subgroups are $\{(0,0),(1,0)\}$and $\{(0,0),(0,1),(0,2)\}$

Exercise 11.6.2

List more examples of proper subgroups of $\mathbb{R}^2$ that are different from the ones listed in this section.

Exercise 11.6.3: Algebraic Properties of the n-Cube

a. The four elements of ${\mathbb{Z}_2}^2$ can be visualized geometrically as the four corners of the 2-cube. Algebraically describe the statements:
   i. Corners a and b are adjacent.
   ii. Corners a and b are diagonally opposite one another.

b. The eight elements of ${\mathbb{Z}_2}^3$ can be visualized as the eight corners of the 3-cube. One face contains $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \{0\}$ and the opposite face contains the remaining four elements so that $(a,b,1)$ is behind $(a,b,0)$. As in part a, describe statements i and ii algebraically.

c. If you could imagine a geometric figure similar to the square or cube in n dimensions, and its corners were labeled by elements of ${\mathbb{Z}_2}^n$ as in parts a and b, how would statements i and ii be expressed algebraically?

Answer

a. (i) $a+b$ could be $(1,0)$ or $(0,1)$. (ii) $a+b=(1,1)$.
b. (i) $a+b$ could be$(1,0,0),(0,1,0),$ or $(0,0,1)$.(ii) $a+b=(1,1,1)$.
c. (i) $a+b$ has exactly one 1. (ii) $a+b$ has all $1's$.

Exercise 11.6.4

a. Suppose that you were to be given a group $[G;*]$ and asked to solve the equation $x*x=e$. Without knowing the group, can you anticipate how many solutions there will be?

b. Answer the same question as part a for the equation $x*x=x$.

Exercise 11.6.5

Which of the following sets are subgroups of $\mathbb{Z}\times\mathbb{Z}$? Give a reason for any negative answers.

a. $\{0\}$
b. $\{(2j,2k)\mid j,k\in\mathbb{Z}\}$
c. $\{(2j+1,2k)\mid j,k\in\mathbb{Z}\}$
d. $\{(n,n^2)\mid n\in\mathbb{Z}\}$
e. $\{(j,k)\mid j+k \text{ is even}\}$

Answer

a. No, 0 is not an element of $\mathbb{Z}\times\mathbb{Z}$.
b. Yes.
c. No, $(0,0)$ is not an element of this set.
d. No, the set is not closed: $(1,1)+(2,4)=(3,5)$ and $(3,5)$ is not in the set.

e. Yes.

Exercise 11.6.6

Determine the following values in the group $\mathbb{Z}_3 \times \mathbb{R}^*$:

a. $(2,1) * (1,2)$
b. the identity element
c. $(1,1/2)^{-1}$

This page titled 11.6: Direct Products is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

11.7: Isomorphisms

The following informal definition of isomorphic systems should be memorized. No matter how technical a discussion about isomorphic systems becomes, keep in mind that this is the essence of the concept.

Definition 11.7.1: Isomorphic Systems/Isomorphism - Informal

Two algebraic systems are isomorphic if there exists a translation rule between them so that any true statement in one system can be translated to a true statement in the other.

Example 11.7.1: How to Do Greek Arithmetic

Imagine that you are a six-year-old child who has been reared in an English-speaking family, has moved to Greece, and has been enrolled in a Greek school. Suppose that your new teacher asks the class to do the following addition problem that has been written out in Greek.

$$\tau\rho\acute{\iota}\alpha \quad \sigma\upsilon\nu \quad \tau\acute{\varepsilon}\sigma\sigma\varepsilon\rho\alpha \quad \iota\sigma o\acute{\upsilon}\tau\alpha\iota \quad \_\_\_\_$$

The natural thing for you to do is to take out your Greek-English/English-Greek dictionary and translate the Greek words to English, as outlined in Figure 11.7.1 After you've solved the problem, you can consult the same dictionary to find the proper Greek word that the teacher wants. Although this is not the recommended method of learning a foreign language, it will surely yield the correct answer to the problem. Mathematically, we may say that the system of Greek integers with addition ($\sigma\upsilon\nu$) is isomorphic to English integers with addition (plus). The problem of translation between natural languages is more difficult than this though, because two complete natural languages are not isomorphic, or at least the isomorphism between them is not contained in a simple dictionary.

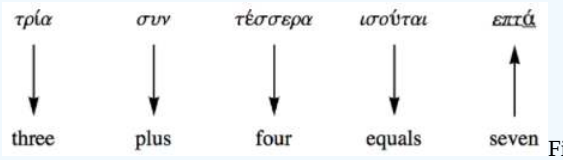


Figure 11.7.1: Solution of a Greek arithmetic problem

Example 11.7.2: Software Implementation of Sets

In this example, we will describe how set variables can be implemented on a computer. We will describe the two systems first and then describe the isomorphism between them.

System 1: The power set of $\{1, 2, 3, 4, 5\}$with the operation union, $\cup$. For simplicity, we will only discuss union. However, the other operations are implemented in a similar way.

System 2: Strings of five bits of computer memory with an OR gate. Individual bit values are either zero or one, so the elements of this system can be visualized as sequences of five 0's and 1's. An OR gate, Figure 11.7.2, is a small piece of computer hardware that accepts two bit values at any one time and outputs either a zero or one, depending on the inputs. The output of an OR gate is one, except when the two bit values that it accepts are both zero, in which case the output is zero. The operation on this system actually consists of sequentially inputting the values of two bit strings into the OR gate. The result will be a new string of five 0's and 1's. An alternate method of operating in this system is to use five OR gates and to input corresponding pairs of bits from the input strings into the gates concurrently.

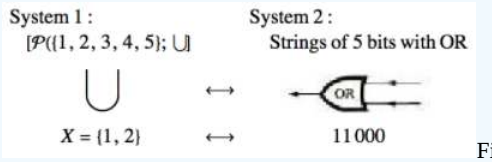


Figure 11.7.2: Translation between sets and strings of bits

The Isomorphism: Since each system has only one operation, it is clear that union and the OR gate translate into one another. The translation between sets and bit strings is easiest to describe by showing how to construct a set from a bit string. If $a_1a_2a_3a_4a_5$, is a bit string in System 2, the set that it translates to contains the number k if and only if a_k equals 1. For

example, 10001 is translated to the set $\{1,5\}$, while the set $\{1,2\}$ is translated to 11000. Now imagine that your computer is like the child who knows English and must do a Greek problem. To execute a program that has code that includes the set expression $\{1,2\} \cup \{1,5\}$, it will follow the same procedure as the child to obtain the result, as shown in Figure 11.7.3

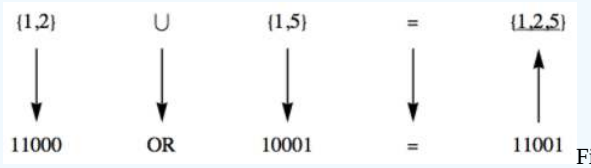


Figure 11.7.3: Translation of a problem in set theory

11.7.1: Group Isomorphisms

Example 11.7.3: Multiplying Without Doing Multiplication

This isomorphism is between $[\mathbb{R}^+;\cdot]$ and $[\mathbb{R};+]$. Until the 1970s, when the price of calculators dropped, multiplication and exponentiation were performed with an isomorphism between these systems. The isomorphism ($\mathbb{R}^+$ to $\mathbb{R}$) between the two groups is that $\cdot$ is translated into $+$ and any positive real number a is translated to the logarithm of a. To translate back from $\mathbb{R}$ to $\mathbb{R}^+$, you invert the logarithm function. If base ten logarithms are used, an element of $\mathbb{R}$, b, will be translated to 10^b. In pre-calculator days, the translation was done with a table of logarithms or with a slide rule. An example of how the isomorphism is used appears in Figure 11.7.4

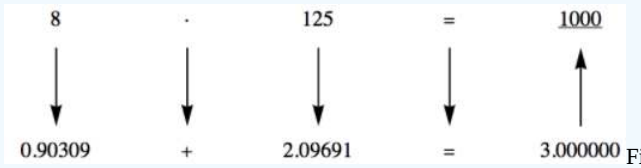


Figure 11.7.4: Multiplication using logarithms

The following definition of an isomorphism between two groups is a more formal one that appears in most abstract algebra texts. At first glance, it appears different, it is really a slight variation on the informal definition. It is the common definition because it is easy to apply; that is, given a function, this definition tells you what to do to determine whether that function is an isomorphism.

Definition 11.7.2: Group Isomorphism

If $[G_1;*_1]$ and $[G_2;*_2]$ are groups, $f: G_1 \to G_2$ is an isomorphism from G_1 into G_2 if:

1. f is a bijection, and
2. $f(a *_1 b) = f(a) *_2 f(b)$ for all $a, b \in G_1$

If such a function exists, then we say G_1 is isomorphic to G_2, denoted $G_1 \cong G_2$.

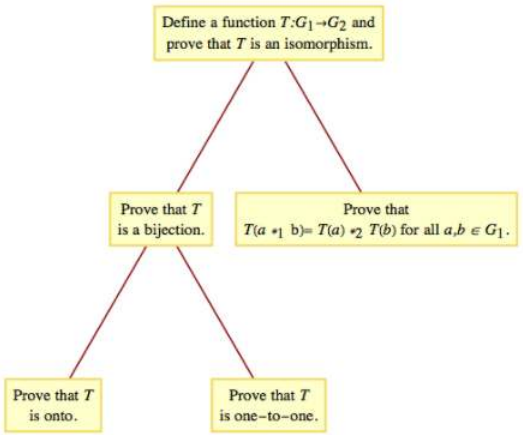


Figure 11.7.5: Steps in proving that G_1 and G_2 are isomorphic

Note 11.7.1

a. There could be several different isomorphisms between the same pair of groups. Thus, if you are asked to demonstrate that two groups are isomorphic, your answer need not be unique.

b. Any application of this definition requires a procedure outlined in Figure 11.7.5. The first condition, that an isomorphism be a bijection, reflects the fact that every true statement in the first group should have exactly one corresponding true statement in the second group. This is exactly why we run into difficulty in translating between two natural languages. To see how Condition (b) of the formal definition is consistent with the informal definition, consider the function $L : \mathbb{R}^+ \to \mathbb{R}$ defined by $L(x) = \log_{10} x$. The translation diagram between $\mathbb{R}^+$ and $\mathbb{R}$ for the multiplication problem $a \cdot b$ appears in Figure 11.7.6. We arrive at the same result by computing $L^{-1}(L(a) + L(b))$ as we do by computing $a \cdot b$. If we apply the function L to the two results, we get the same image:

$$L(a \cdot b) = L(L^{-1}(L(a) + L(b))) = L(a) + L(b) \tag{11.7.1}$$

since $L\left(L^{-1}(x)\right) = x$. Note that (11.7.1) is exactly Condition b of the formal definition applied to the two groups $\mathbb{R}^+$ and $\mathbb{R}$.

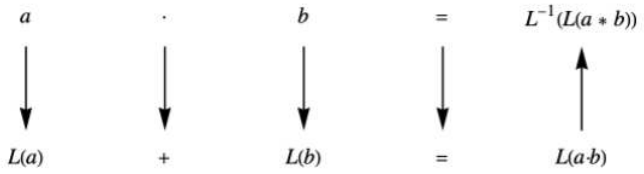


Figure 11.7.6: General Multiplication using Logarithms

Example 11.7.4

Consider $G = \left\{ \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \middle| a \in \mathbb{R} \right\}$ with matrix multiplication. The group $[\mathbb{R}; +]$ is isomorphic to G. Our translation rule is the function $f : \mathbb{R} \to G$ defined by $f(a) = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$. Since groups have only one operation, there is no need to state explicitly that addition is translated to matrix multiplication. That f is a bijection is clear from its definition.

If a and b are any real numbers,

$$\begin{aligned} f(a)f(b) &= \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & a+b \\ 0 & 1 \end{pmatrix} \\ &= f(a+b) \end{aligned}$$

We can apply this translation rule to determine the inverse of a matrix in G. We know that $a + (-a) = 0$ is a true statement in $\mathbb{R}$. Using f to translate this statement, we get

$$f(a)f(-a) = f(0)$$

or

$$\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

therefore,

$$\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}$$

The next theorem summarizes some of the general facts about group isomorphisms that are used most often in applications. We leave the proof to the reader.

Theorem 11.7.1: Properties of Isomorphisms

If $[G;*]$ and $[H;\diamond]$ are groups with identities e and e', respectively, and $T: G \to H$ is an isomorphism from G into H, then:

a. $T(e) = e'$
b. $T(a)^{-1} = T\left(a^{-1}\right)$ for all $a \in G$, and
c. If K is a subgroup of G, then $T(K) = \{T(a) : a \in K\}$ is a subgroup of H and is isomorphic to K.

"Is isomorphic to" is an equivalence relation on the set of all groups. Therefore, the set of all groups is partitioned into equivalence classes, each equivalence class containing groups that are isomorphic to one another.

11.7.2: Conditions for groups to not be isomorphic

How do you decide that two groups are not isomorphic to one another? The negation of "G and H are isomorphic" is that no translation rule between G and H exists. If G and H have different cardinalities, then no bijection from G into H can exist. Hence they are not isomorphic. Given that $|G| = |H|$, it is usually impractical to list all bijections from G into H and show that none of them satisfy Condition b of the formal definition. The best way to prove that two groups are not isomorphic is to find a true statement about one group that is not true about the other group. We illustrate this method in the following checklist that you can apply to most pairs of non-isomorphic groups in this book.

Assume that $[G;*]$ and $[H;\diamond]$ are groups. The following are reasons for G and H to be not isomorphic.

a. G and H do not have the same cardinality. For example, $\mathbb{Z}_{12} \times \mathbb{Z}_5$ can't be isomorphic to $\mathbb{Z}_{50}$ and $[\mathbb{R};+]$ can't be isomorphic to $\left[\mathbb{Q}^+;\cdot\right]$.
b. G is abelian and H is not abelian since $a * b = b * a$ is always true in G, but $T(a) \diamond T(b) = T(b) \diamond T(a)$ would not always be true. We have seen two groups with six elements that apply here. They are $\mathbb{Z}_6$ and the group of 3×3 rook matrices (see Exercise 11.2.5). The second group is non-abelian, therefore it can't be isomorphic to $\mathbb{Z}_6$.
c. G has a certain kind of subgroup that H doesn't have. Part (c) of Theorem 11.7.1 states that this cannot happen if G is isomorphic to H. $[\mathbb{R}^*;\cdot]$ and $\left[\mathbb{R}^+;\cdot\right]$ are not isomorphic since $\mathbb{R}^*$ has a subgroup with two elements, $\{-1,1\}$, while the proper subgroups of $\mathbb{R}^+$ are all infinite (convince yourself of this fact!).
d. The number of solutions of $x * x = e$ in G is not equal to the number of solutions of $y \diamond y = e'$ in H. $\mathbb{Z}_8$ is not isomorphic to ${\mathbb{Z}_2}^3$ since $x +_8 x = 0$ has two solutions, 0 and 4, while $y + y = (0,0,0)$ is true for all $y \in {\mathbb{Z}_2}^3$. If the operation in G is defined by a table, then the number of solutions of $x * x = e$ will be the number of occurrences of e in the main diagonal of the table. The equations $x^3 = e,\ x^4 = e, \ldots$ can also be used in the same way to identify pairs of non-isomorphic groups.
e. One of the cyclic subgroups of G equals G (i. e., G is cyclic), while none of H's cyclic subgroups equals H (i. e., H is noncyclic). This is a special case of Condition c. $\mathbb{Z}$ and $\mathbb{Z} \times \mathbb{Z}$ are not isomorphic since $\mathbb{Z} = \langle 1 \rangle$ and $\mathbb{Z} \times \mathbb{Z}$ is not cyclic.

11.7.3: Exercises

Exercise 11.7.1

State whether each pair of groups below is isomorphic. For each pair that is, give an isomorphism; for those that are not, give your reason.

a. $\mathbb{Z} \times \mathbb{R}$ and $\mathbb{R} \times \mathbb{Z}$
b. $\mathbb{Z}_2 \times \mathbb{Z}$ and $\mathbb{Z} \times \mathbb{Z}$
c. $\mathbb{R}$ and $\mathbb{Q} \times \mathbb{Q}$
d. $\mathcal{P}(\{1,2\})$ with symmetric difference and ${\mathbb{Z}_2}^2$
e. ${\mathbb{Z}_2}^2$ and $\mathbb{Z}_4$
f. $\mathbb{R}^4$ and $M_{2\times 2}(\mathbb{R})$ with matrix addition
g. $\mathbb{R}^2$ and $\mathbb{R} \times \mathbb{R}^+$
h. $\mathbb{Z}_2$ and the 2×2 rook matrices
i. $\mathbb{Z}_6$ and $\mathbb{Z}_2 \times \mathbb{Z}_3$

Answer

a. Yes, $f(n,x) = (x,n)$ for $(n,x) \in \mathbb{Z} \times \mathbb{R}$ is an isomorphism.
b. No, $\mathbb{Z}_2 \times \mathbb{Z}$ has a two element subgroup while $\mathbb{Z} \times \mathbb{Z}$ does not.

c. No. $\mathbb{Q}\times\mathbb{Q}$ is countable and $\mathbb{R}$ is not. Therefore, no bijection can exist between them.
d. Yes.
e. No.
f. Yes, one isomorphism is defined by $f(a_1,a_2,a_3,a_4)=\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$.
g. Yes, one isomorphism is defined by $f(a_1,a_2)=(a_1,10^{a_2})$.
h. Yes.
i. Yes $f(k)=k(1,1)$.

Exercise 11.7.2

If you know two natural languages, show that they are not isomorphic.

Exercise 11.7.3

Prove that the relation "is isomorphic to" on groups is transitive.

Answer

Consider three groups G_1, G_2, and G_3 with operations $*,\diamond$, and $\star$, respectively. We want to show that if G_1 is isomorphic to G_2, and if G_2 is isomorphic to G_3 , then G_1 is isomorphic to G_3.

$$G_1 \text{ isomorphic to } G_2 \Rightarrow \text{ there exists an isomorphism } f: G_1 \to G_2$$

$$G_2 \text{ isomorphic to } G_3 \Rightarrow \text{ there exists an isomorphism } g: G_2 \to G_3$$

If we compose g with f, we get the function $g\circ f: G_1 \to G_3$, By Theorem 7.3.2 and Theorem 7.3.3, $g\circ f$ is a bijection, and if $a,b\in G_1$,

$$\begin{aligned}(g\circ f)(a*b) &= g(f(a*b)) \\ &= g(f(a)\diamond f(b)) \quad \text{since } f \text{ is an isomorphism} \\ &= g(f(a))\star g(f(b)) \quad \text{since } g \text{ is an isomorphism} \\ &= (g\circ f)(a)\star(g\circ f)(b)\end{aligned}$$

Therefore, $g\circ f$ is an isomorphism from G_1 into G_3, proving that "is isomorphic to" is transitive.

Exercise 11.7.4

a. Write out the operation table for $G=[\{1,-1,i,-i\};\cdot]$where i is the complex number for which $i^2=-1$. Show that G is isomorphic to $[\mathbb{Z}_4;+_4]$.
b. Solve $x^2=-1$ in G by first translating the equation to $\mathbb{Z}_4$, solving the equation in $\mathbb{Z}_4$, and then translating back to G.

Exercise 11.7.5

The two groups $[\mathbb{Z}_4;+_4]$ and $[U_5;\times_5]$ are isomorphic. One isomorphism $T:\mathbb{Z}_4\to U_5$ is partially defined by $T(1)=3$. Determine the values of $T(0)$, $T(2)$, and $T(3)$.

Answer

By Theorem 11.7.1(a), $T(0)$ must be 1. $T(2)=T(1+_4 1)=T(1)\times_5 T(1)=3\times_5 3=4$. Since T is a bijection, $T(3)=2$.

Exercise 11.7.6

Prove Theorem 11.7.1.

Exercise 11.7.7

Prove that all infinite cyclic groups are isomorphic to $\mathbb{Z}$.

Answer

Let G be an infinite cyclic group generated by a. Then, using multiplicative notation, $G = \{a^n | \, n \in \mathbb{Z}\}$. The map $T : G \to \mathbb{Z}$ defined by $T(a^n) = n$ is an isomorphism. This is indeed a function, since $a^n = a^m$ implies $n = m$. Otherwise, a would have a finite order and would not generate G.

a. T is one-to-one, since $T(a^n) = T(a^m)$ implies $n = m$, so $a^n = a^m$.
b. T is onto, since for any $n \in \mathbb{Z}$, $T(a^n) = n$.
c. $T(a^n * a^m) = T\left(a^{n+m}\right) = n + m = T(a^n) + T(a^m)$

Exercise 11.7.8

a. Prove that $\mathbb{R}^*$ is isomorphic to $\mathbb{Z}_2 \times \mathbb{R}$.
b. Describe how multiplication of nonzero real numbers can be accomplished doing only additions and translations.

Exercise 11.7.9

Prove that if G is any group and g is some fixed element of G, then the function ϕ_g defined by $\phi_g(x) = g * x * g^{-1}$ is an isomorphism from G into itself. An isomorphism of this type is called an inner automorphism.

Exercise 11.7.10

It can be shown that there are five non-isomorphic groups of order eight. You should be able to describe at least three of them. Do so without use of tables. Be sure to explain why they are not isomorphic.

Answer

$\mathbb{Z}_8$, $\mathbb{Z}_2 \times \mathbb{Z}_4$, and $\mathbb{Z}_2^3$. One other is the fourth dihedral group, introduced in Section 15.3.

This page titled 11.7: Isomorphisms is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

12: More Matrix Algebra

augmented matrix

There's a Gaussian technique whose intent
Is to solve the constraints you present
As a matrix equation—
Once you've had the occasion
*To write down your constants (**augment**)*

Steve Ngai, The Omnificent English Dictionary in Limerick Form

In Chapter 5 we studied matrix operations and the algebra of sets and logic. We also made note of the strong resemblance of matrix algebra to elementary algebra. The reader should briefly review this material. In this chapter we shall look at a powerful matrix tool in the applied sciences, namely a technique for solving systems of linear equations. We will then use this process for determining the inverse of $n \times n$ matrices, $n \geq 2$, when they exist. We proceed with a development of the diagonalization process, with a discussion of several of its applications. Finally, we discuss the solution of linear equations over the integers modulo 2.

12.1: Systems of Linear Equations
12.2: Matrix Inversion
12.3: An Introduction to Vector Spaces
12.4: The Diagonalization Process
12.5: Some Applications
12.6: Linear Equations over the Integers Mod 2

This page titled 12: More Matrix Algebra is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

12.1: Systems of Linear Equations

12.1.1: Solutions

The method of solving systems of equations by matrices that we will look at is based on procedures involving equations that we are familiar with from previous mathematics courses. The main idea is to reduce a given system of equations to another simpler system that has the same solutions.

Definition 12.1.1: Solution Set

Given a system of equations involving real variables $x_1, x_2, \ldots, x_n$, the solution set of the system is the set of n-tuples in $\mathbb{R}^n$, $(a_1, a_2, \ldots, a_n)$ such that the substitutions $x_1 = a_1,\ x_2 = a_2, \ldots,\ x_n = a_n$ make all the equations true.

In terms of logic, a solution set is a truth set of a system of equations, which is a proposition over n-tuples of real numbers.

In general, if the variables are from a set S, then the solution set will be a subset of S^n. For example, in number theory mathematicians study Diophantine equations, where the variables can only take on integer values instead of real values.

Definition 12.1.2: Equivalent Systems of Equations

Two systems of linear equations are called equivalent if they have the same set of solutions.

Example 12.1.1: Two Equivalent Systems

The previous definition tells us that if we know that the system

$$\begin{aligned} 4x_1 + 2x_2 + x_3 &= 1 \\ 2x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 + x_3 &= 3 \end{aligned}$$

is equivalent to the system

$$\begin{aligned} x_1 + 0x_2 + 0x_3 &= -1 \\ 0x_1 + x_2 \ + 0x_3 &= -1 \\ 0x_1 + 0x_2 \ + x_3 &= 7 \end{aligned}$$

then both systems have the solution set $\{(-1, -1, 7)\}$. In other words, the simultaneous values $x_1 = -1,\ x_2 = -1,$ and $x_3 = 7$ are the only values of the variables that make all three equations in either system true.

12.1.2: Elementary Operations on Equations

Theorem 12.1.1: Elementary Operations on Equations

If any sequence of the following operations is performed on a system of equations, the resulting system is equivalent to the original system:

a. Interchange any two equations in the system.
b. Multiply both sides of any equation by a nonzero constant.
c. Multiply both sides of any equation by a nonzero constant and add the result to a second equation in the system, with the sum replacing the latter equation.

Let us now use the above theorem to work out the details of Example 12.1.1 and see how we can arrive at the simpler system.

The original system:

$$\begin{aligned} 4x_1 + 2x_2 + x_3 &= 1 \\ 2x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 + x_3 &= 3 \end{aligned} \tag{12.1.1}$$

Step 1. We will first change the coefficient of x_1 in the first equation to one and then use it as a pivot to obtain 0's for the coefficients of x_1 in Equations 2 and 3.

- Multiply Equation 1 by $\frac{1}{4}$ to obtain

$$\begin{aligned} x_1 + \frac{x_2}{2} + \frac{x_3}{4} &= \frac{1}{4} \\ 2x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 + x_3 &= 3 \end{aligned} \tag{12.1.2}$$

- Multiply Equation 1 by -2 and add the result to Equation 2 to obtain

$$\begin{aligned} x_1 + \frac{x_2}{2} + \frac{x_3}{4} &= \frac{1}{4} \\ 0x_1 + 0x_2 + \frac{x_3}{2} &= \frac{7}{2} \\ 2x_1 + 2x_2 + x_3 &= 3 \end{aligned} \tag{12.1.3}$$

- Multiply Equation 1 by -2 and add the result to Equation 3 to obtain

$$\begin{aligned} x_1 + \frac{x_2}{2} + \frac{x_3}{4} &= \frac{1}{4} \\ 0x_1 + 0x_2 + \frac{x_3}{2} &= \frac{7}{2} \\ 0x_1 + x_2 + \frac{x_3}{2} &= \frac{5}{2} \end{aligned} \tag{12.1.4}$$

We've explicitly written terms with zero coefficients such as $0x_1$ to make a point that all variables can be thought of as being involved in all equations. After this example is complete, we will discontinue this practice in favor of the normal practice of making these terms “disappear.”

Step 2. We would now like to proceed in a fashion analogous to Step 1; namely, multiply the coefficient of x_2 in the second equation by a suitable number so that the result is 1. Then use it as a pivot to obtain 0's as coefficients for x_2 in the first and third equations. This is clearly impossible (Why?), so we will first interchange Equations 2 and 3 and proceed as outlined above.

- Exchange Equations 2 and 3 to obtain

$$\begin{aligned} x_1 + \frac{x_2}{2} + \frac{x_3}{4} &= \frac{1}{4} \\ 0x_1 + x_2 + \frac{x_3}{2} &= \frac{5}{2} \\ 0x_1 + 0x_2 + \frac{x_3}{2} &= \frac{7}{2} \end{aligned} \tag{12.1.5}$$

- Multiply Equation 2 by $\frac{1}{2}$ and subtract the result from Equation 1 to obtain

$$\begin{aligned} x_1 + 0x_2 + 0x_3 &= -1 \\ 0x_1 + x_2 + \frac{x_3}{2} &= \frac{5}{2} \\ 0x_1 + 0x_2 + \frac{x_3}{2} &= \frac{7}{2} \end{aligned} \tag{12.1.6}$$

Step 3. Next, we will change the coefficient of x_3 in the third equation to one and then use it as a pivot to obtain 0's for the coefficients of x_3 in Equations 1 and 2. Notice that the coefficient of x_3 is already zero in Equation 1, so we have been saved some work!

- Multiply Equation 3 by 2 to obtain

$$\begin{aligned} x_1 + 0x_2 + 0x_3 &= -1 \\ 0x_1 + x_2 + \frac{x_3}{2} &= \frac{5}{2} \\ 0x_1 + 0x_2 + x_3 &= 7 \end{aligned}$$

- Multiply Equation 3 by $-1/2$ and add the result to Equation 2 to obtain

$$\begin{aligned} x_1 + 0x_2 + 0x_3 &= -1 \\ 0x_1 + x_2 + 0x_3 &= -1 \\ 0x_1 + 0x_2 + x_3 &= 7 \end{aligned} \tag{12.1.7}$$

From the system of equations at the end of Step 3, we see that the solution to the original system is $x_1 = -1$, $x_2 = -1$, and $x_3 = 7$.

12.1.3: Transition to Matrices

In the above sequence of steps, we note that the variables serve the sole purpose of keeping the coefficients in the appropriate location. This we can effect by using matrices. The matrix of the original system in our example is

$$\left(\begin{array}{ccc|c} 4 & 2 & 1 & 1 \\ 2 & 1 & 1 & 4 \\ 2 & 2 & 1 & 3 \end{array}\right)$$

where the matrix of the first three columns is called the coefficient matrix and the complete matrix is referred to as the augmented matrix. Since we are now using matrices to solve the system, we will translate Theorem 12.1.1 into matrix language.

12.1.4: Elementary Row Operations

Theorem 12.1.2: Elementary Row Operations

If any sequence of the following operations is performed on the augmented matrix of a system of equations, the resulting matrix is a system that is equivalent to the original system. The following operations on a matrix are called elementary row operations:

1. Exchange any two rows of the matrix.
2. Multiply any row of the matrix by a nonzero constant.
3. Multiply any row of the matrix by a nonzero constant and add the result to a second row, with the sum replacing that second row.

Definition 12.1.3: Row Equivalent Matrices

Two matrices, A and B, are said to be row-equivalent if one can be obtained from the other by any sequence of zero or more elementary row operations.

If we use the notation R_i to stand for Row i of a matrix and $\longrightarrow$ to stand for row equivalence, then

$$A \overset{cR_i+R_j}{\longrightarrow} B$$

means that the matrix B is obtained from the matrix A by multiplying the Row i of A by c and adding the result to Row j. The operation of multiplying row i by c is indicated by

$$A \overset{cR_i}{\longrightarrow} B$$

while exchanging rows i and j is denoted by

$$A \overset{R_i \leftrightarrow R_j}{\longrightarrow} B.$$

The matrix notation for the system given in our first example, with the subsequent steps, is:

$$\left(\begin{array}{ccc|c} 4 & 2 & 1 & 1 \\ 2 & 1 & 1 & 4 \\ 2 & 2 & 1 & 3 \end{array}\right) \xrightarrow{\frac{1}{4}R_1} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 2 & 1 & 1 & 4 \\ 2 & 2 & 1 & 3 \end{array}\right) \xrightarrow{-2R_1+R_2} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{7}{2} \\ 2 & 2 & 1 & 3 \end{array}\right)$$

$$\xrightarrow{-2R_1+R_3} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{7}{2} \\ 0 & 1 & \frac{1}{2} & \frac{5}{2} \end{array}\right) \xrightarrow{R_2 \leftrightarrow R_3} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & \frac{1}{2} & \frac{7}{2} \end{array}\right)$$

$$\xrightarrow{-\frac{1}{2}R_2+R_1} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & \frac{1}{2} & \frac{7}{2} \end{array}\right) \xrightarrow{2R_3} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 1 & 7 \end{array}\right)$$

$$\xrightarrow{-\frac{1}{2}R_3+R_2} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 7 \end{array}\right)$$

This again gives us the solution. This procedure is called the ***Gauss-Jordan elimination method***.

It is important to remember when solving any system of equations via this or any similar approach that at any step in the procedure we can rewrite the matrix in "equation format" to help us to interpret the meaning of the augmented matrix.

In our first example we found a unique solution, only one triple, namely $(-1,-1,7)$, which satisfies all three equations. For a system involving three unknowns, are there any other possible results? To answer this question, let's review some basic facts from analytic geometry.

The graph of a linear equation in three-dimensional space is a plane. So geometrically we can visualize the three linear equations as three planes in three-space. Certainly the three planes can intersect in a unique point, as in the first example, or two of the planes could be parallel. If two planes are parallel, there are no common points of intersection; that is, there are no triple of real numbers that will satisfy all three equations. Another possibility is that the three planes could intersect along a common axis or line. In this case, there would be an infinite number of real number triples in $\mathbb{R}^3$. Yet another possibility would be if the first two planes intersect in a line, but the line is parallel to, but not on, the third plane, giving us no solution. Finally if all three equations describe the same plane, the solution set would be that plane.

We can generalize these observations. In a system of n linear equations, n unknowns, there can be

1. a unique solution,
2. no solution, or
3. an infinite number of solutions.

To illustrate these points, consider the following examples:

Example 12.1.2: A System with No Solutions

Find all solutions to the system

$$\begin{aligned} x_1 + 3x_2 \ + x_3 &= 2 \\ x_1 \ + x_2 + 5x_3 &= 4 \\ 2x_1 + 2x_2 + 10x_3 &= 6 \end{aligned}$$

The reader can verify that the augmented matrix of this system, $\left(\begin{array}{ccc|c} 1 & 3 & 1 & 2 \\ 1 & 1 & 5 & 4 \\ 2 & 2 & 10 & 6 \end{array}\right)$, reduces to $\left(\begin{array}{ccc|c} 1 & 3 & 1 & 2 \\ 1 & 1 & 5 & 4 \\ 0 & 0 & 0 & -2 \end{array}\right)$.

We can attempt to row-reduce this matrix further if we wish. However, any further row-reduction will not substantially change the last row, which, in equation form, is $0x_1 + 0x_2 + 0x_3 = -2$, or simply $0 = -2$. It is clear that we cannot find real numbers x_1, x_2, and x_3 that will satisfy this equation. Hence we cannot find real numbers that will satisfy all three original equations simultaneously. When this occurs, we say that the system has no solution, or the solution set is empty.

Example 12.1.3: A System with an Infinite Number of Solutions

Next, let's attempt to find all of the solutions to:

$$\begin{aligned} x_1 + 6x_2 + 2x_3 &= 1 \\ 2x_1 + x_2 + 3x_3 &= 2 \\ 4x_1 + 2x_2 + 6x_3 &= 4 \end{aligned}$$

The augmented matrix for the system is

$$\left(\begin{array}{ccc|c} 1 & 6 & 2 & 1 \\ 2 & 1 & 3 & 2 \\ 4 & 2 & 6 & 4 \end{array}\right) \tag{12.1.8}$$

which reduces to

$$\left(\begin{array}{ccc|c} 1 & 0 & \frac{16}{11} & 1 \\ 0 & 1 & \frac{1}{11} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) \tag{12.1.9}$$

If we apply additional elementary row operations to this matrix, it will only become more complicated. In particular, we cannot get a one in the third row, third column. Since the matrix is in simplest form, we will express it in equation format to help us determine the solution set.

$$\begin{aligned} x_1 \quad + \frac{16}{11}x_3 &= 1 \\ x_2 + \frac{1}{11}x_3 &= 0 \\ 0 &= 0 \end{aligned} \tag{12.1.10}$$

Any real numbers will satisfy the last equation. However, the first equation can be rewritten as $x_1 = 1 - \frac{16}{11}x_3,$ which describes the coordinate x_1 in terms of x_3 . Similarly, the second equation gives x_2 in terms of x_3 . A convenient way of listing the solutions of this system is to use set notation. If we call the solution set of the system S, then

$$S = \left\{ \left(1 - \frac{16}{11}x_3, -\frac{1}{11}x_3, x_3\right) \mid x_3 \in \mathbb{R} \right\}.$$

What this means is that if we wanted to list all solutions, we would replace x_3 by all possible numbers. Clearly, there is an infinite number of solutions, two of which are $(1, 0, 0)$ and $(-15, -1, 11),$ when x_3 takes on the values 0 and 11, respectively.

A Word Of Caution: Frequently we may can get “different-looking” answers to the same problem when a system has an infinite number of solutions. Assume the solutions set in this example is reported to be $A = \{(1 + 16x_2, x_2, -11x_3) \mid x_3 \in \mathbb{R}\}$. Certainly the result described by S looks different from that described by A. To see whether they indeed describe the same set, we wish to determine whether every solution produced in S can be generated in A. For example, the solution generated by S when $x_3 = 11$ is $(-15, -1, 11)$. The same triple can be produced by A by taking $x_2 = -1$. We must prove that every solution described in S is described in A and, conversely, that every solution described in A is described in S. (See Exercise 12.1.6 of this section.)

To summarize the procedure in the Gauss-Jordan technique for solving systems of equations, we attempt to obtain 1's along the main diagonal of the coefficient matrix with 0's above and below the diagonal. We may find in attempting this that this objective cannot be completed, as in the last two examples we have seen. Depending on the way we interpret the results in equation form, we either recognize that no solution exists, or we identity “free variables” on which an infinite number of solutions are based. The final matrix forms that we have produced in our examples are referred to as ***echelon forms***.

In practice, larger systems of linear equations are solved using computers. Generally, the Gauss-Jordan algorithm is the most useful; however, slight variations of this algorithm are also used. The different approaches share many of the same advantages and disadvantages. The two major concerns of all methods are:

1. minimizing inaccuracies due to round-off errors, and
2. minimizing computer time.

12.1.5: Gauss-Jordan Algorithm

The accuracy of the Gauss-Jordan method can be improved by always choosing the element with the largest absolute value as the pivot element, as in the following algorithm.

Algorithm 12.1.1: The Gauss-Jordan Algorithm

Given a matrix equation $Ax = b$, where A is $n \times m$, let C be the augmented matrix $[A|b]$. The process of row-reducing to echelon form involves performing the following algorithm where $C[i]$ is the ith row of C.

1. `i = 1`
2. `j = 1`
3. while `i <= n and j <= m)` :
   a. `maxi=i`
   b. for k = i+1 to n:
      if `abs(C[k,j])>abs(C[maxi,j])` : then `maxi=k`
   c. if `C[maxi,j] != 0` then:
      i. exchange rows `i` and `maxi`
      ii. divide each entry in row `i` by `C[i,j]`
      iii. for `u = i+1 to n` :
         subtract `C[u,j]*C[i]` from `C[u]`
      iv. `i = i+1`
   d. `j=j+1`

Note 12.1.1

At the end of this algorithm, with the final form of C you can revert back to the equation form of the system and a solution should be clear. In general,

- If any row of C is all zeros, it can be ignored.
- If any row of C has all zero entries except for the entry in the $(m+1)^{\text{st}}$ position, the system has no solution. Otherwise, if a column has no pivot, the variable corresponding to it is a free variable. Variables corresponding to pivots are basic variables and can be expressed in terms of the free variables.

Example 12.1.4

If we apply The Gauss-Jordan Algorithm, Algorithm 12.1.1 to the system

$$\begin{aligned} 5x_1 + x_2 + 2x_3 + x_4 &= 2 \\ 3x_1 + x_2 - 2x_3 \quad\;\; &= 5 \\ x_1 + x_2 + 3x_3 - x_4 &= -1 \end{aligned}$$

the augmented matrix is

$$\left(\begin{array}{cccc|c} 5 & 1 & 2 & 1 & 2 \\ 3 & 1 & -2 & 0 & 5 \\ 1 & 1 & 3 & -1 & -1 \end{array}\right)$$

is reduced to

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 & -\frac{3}{2} & \frac{3}{2} \\ 0 & 0 & 1 & 0 & -1 \end{array}\right)$$

Therefore, x_4 is a free variable in the solution and general solution of the system is

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} - \frac{1}{2}x_4 \\ \frac{3}{2} + \frac{3}{2}x_4 \\ -1 \\ x_4 \end{pmatrix}$$

This conclusion is easy to see if you revert back to the equations that the final value the reduced matrix represents.

12.1.6: SageMath Note - Matrix Reduction

Given an augmented matrix, C, there is a matrix method called `echelon_form` that can be used to row reduce C. Here is the result for the system in Example 12.1.4 In the assignment of a matrix value to C, notice that the first argument is QQ, which indicates that the entries should be rational numbers. As long as all the entries are rational, which is the case here since integers are rational, the row-reduced matrix will be all rational.

```
C = Matrix(QQ,[[5,1,2,1,2],[3,1,-2,0,5],[1,1,3,-1,-1]])
C.echelon_form()
```

If we don't specify the set from which entries are taken, it would assumed to be the integers and we do not get a fully row-reduced matrix. This is because the next step in working with the next output would involve multiplying row 2 by $\frac{1}{2}$ and row 3 by $\frac{1}{9}$, but these multipliers are not integers.

```
C2 = Matrix([[5,1,2,1,2],[3,1,-2,0,5],[1,1,3,-1,-1]])
C2.echelon_form()
```

If we specifying real entries, the result isn't as nice and clean as the rational output.

```
C3 = Matrix(RR,[[5,1,2,1,2],[3,1,-2,0,5],[1,1,3,-1,-1]])
C3.echelon_form()
```

The default number of decimal places may vary from what you see here, but it can be controlled. The single small number in row three column four isn't exactly zero because of round-off but we could just set it to zero.

12.1.7: Exercises

Exercise 12.1.1

Solve the following systems by describing the solution sets completely:

a. $2x_1 + x_2 = 3$
$x_1 - x_2 = 1$
$2x_1 + x_2 + 3x_3 = 5$

b. $4x_1 + x_2 + 2x_3 = -1$
$8x_1 + 2x_2 + 4x_3 = -2$
$x_1 + x_2 + 2x_3 = 1$

c. $x_1 + 2x_2 - x_3 = -1$
$x_1 + 3x_2 + x_3 = 5$

d. $x_1 - x_2 + 3x_3 = 7$
$x_1 + 3x_2 + x_3 = 4$

Answer

a. $\{(4/3, 1/3)\}$

b. $\{(\frac{1}{2}x_3 - 3, -4x_3 + 11, x_3) \mid x_3 \in \mathbb{R}\}$

c. $\{(-5, 14/5, 8/5)\}$
d. $\{(6.25 - 2.5x_3, -0.75 + 0.5x_3, x_3) \mid x_3 \in \mathbb{R}\}$

Exercise 12.1.2

Solve the following systems by describing the solution sets completely:

a.
$$\begin{aligned} 2x_1 + 2x_2 + 4x_3 &= 2 \\ 2x_1 + x_2 + 4x_3 &= 0 \\ 3x_1 + 5x_2 + x_3 &= 0 \\ 2x_1 + x_2 + 3x_3 &= 2 \end{aligned}$$

b.
$$\begin{aligned} 4x_1 + x_2 + 2x_3 &= -1 \\ 8x_1 + 2x_2 + 4x_3 &= 4 \\ x_1 + x_2 + 2x_3 + x_4 &= 3 \end{aligned}$$

c.
$$\begin{aligned} x_1 - x_2 + 3x_3 - x_4 &= -2 \\ 3x_1 + 3x_2 + 6x_3 + 3x_4 &= 9 \\ 6x_1 + 7x_2 + 2x_3 &= 3 \end{aligned}$$

d.
$$\begin{aligned} 4x_1 + 2x_2 + x_3 &= -2 \\ 6x_1 + x_2 + x_3 &= 1 \\ x_1 + x_2 - x_3 + 2x_4 &= 1 \end{aligned}$$

e.
$$\begin{aligned} x_1 + 2x_2 + 3x_3 + x_4 &= 5 \\ x_1 + 3x_2 + 2x_3 - x_4 &= -1 \end{aligned}$$

Exercise 12.1.3

Given the final augmented matrices below from the Gauss-Jordan Algorithm, identify the solutions sets. Identify the basic and free variables, and describe the solution set of the original system.

a. $\left(\begin{array}{cccc|c} 1 & 0 & -5 & 0 & 1.2 \\ 0 & 1 & 4 & 0 & 2.6 \\ 0 & 0 & 0 & 1 & 4.5 \end{array}\right)$

b. $\left(\begin{array}{ccc|c} 1 & 0 & 9 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{array}\right)$

c. $\left(\begin{array}{ccc|c} 1 & 0 & 6 & 5 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right)$

d. $\left(\begin{array}{cccc|c} 1 & 0 & 0 & -3 & 1 \\ 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & -1 & 1 \end{array}\right)$

Answer

a. Basic variables: x_1, x_2 and x_4. Free variable: x_3. Solution set: $\{(1.2 + 5x_3, 2.6 - 4x_3, 4.5) \mid x_3 \in \mathbb{R}\}$
b. Basic variables: x_1 and x_2. Free variable: x_3. The solution set is empty because the last row of the matrix converts to the inconsistent equation $0 = 1$.
c. Basic variables: x_1 and x_2. Free variable: x_3. Solution set: $\{(-6x_3 + 5, 2x_3 + 1, x_3) \mid x_3 \in \mathbb{R}\}$
d. Basic variables: x_1, x_2 and x_3. Free variable: x_4. Solution set: $\{(3x_4 + 1, -2x_4 + 2, x_4 + 1, x_4) \mid x_4 \in \mathbb{R}\}$

Exercise 12.1.4

a. Write out the details of Example 12.1.2
b. Write out the details of Example 12.1.3
c. Write out the details of Example 12.1.4

Exercise 12.1.5

Solve the following systems using only mod 5 arithmetic. Your solutions should be $n-\text{tuples}$ from $\mathbb{Z}_5$.

a. $\begin{aligned} 2x_1+x_2&=3\\ x_1+4x_2&=1 \end{aligned}$ (compare your solution to the system in 5(a))

b. $\begin{aligned} x_1+x_2+2x_3&=1\\ x_1+2x_2+4x_3&=4\\ x_1+3x_2+3x_3&=0 \end{aligned}$

Answer

a. $\{(3,0)\}$
b. $\{(3,0,4)\}$

Exercise 12.1.6

a. Use the solution set S of Example 12.1.3to list three different solutions to the given system. Then show that each of these solutions can be described by the set A in the same example.
b. Prove that $S=A$.

Exercise 12.1.7

Given a system of n linear equations in n unknowns in matrix form $AX=b,$ prove that if b is a matrix of all zeros, then the solution set of $AX=b$ is a subgroup of $\mathbb{R}^n$.

Answer

Proof: Since b is the $n\times 1$ matrix of $0\{'\}$s, let's call it **0**. Let S be the set of solutions to $AX=\mathbf{0}$. If X_1 and X_2 be in S. Then

$$A(X_1+X_2)=AX_1+AX_2=0+0=0$$

so $X_1+X_2\in S$; or in other words, S is closed under addition in $\mathbb{R}^n$.

The identity of $\mathbb{R}^n$ is **0**, which is in S. Finally, let X be in S. Then

$$A(-X)=-(AX)=-0=0$$

and so $-X$ is also in S.

This page titled 12.1: Systems of Linear Equations is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

12.2: Matrix Inversion

12.2.1: Developing the Process

In Chapter 5 we defined the inverse of an $n \times n$ matrix. We noted that not all matrices have inverses, but when the inverse of a matrix exists, it is unique. This enables us to define the inverse of an $n \times n$ matrix A as the unique matrix B such that $AB = BA = I$, where I is the $n \times n$ identity matrix. In order to get some practical experience, we developed a formula that allowed us to determine the inverse of invertible 2×2 matrices. We will now use the Gauss-Jordan procedure for solving systems of linear equations to compute the inverses, when they exist, of $n \times n$ matrices, $n \geq 2$. The following procedure for a 3×3 matrix can be generalized for $n \times n$ matrices, $n \geq 2$.

Given the matrix $A = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 4 \\ 3 & 5 & 1 \end{pmatrix}$, we want to find its inverse, the matrix $B = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$, if it exists, such that $AB = I$ and $BA = I$. We will concentrate on finding a matrix that satisfies the first equation and then verify that B also satisfies the second equation.

The equation

$$\begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 4 \\ 3 & 5 & 1 \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is equivalent to

$$\begin{pmatrix} x_{11}+x_{21}+2x_{31} & x_{12}+x_{22}+2x_{32} & x_{13}+x_{23}+2x_{33} \\ 2x_{11}+x_{21}+4x_{31} & 2x_{12}+x_{22}+4x_{32} & 2x_{13}+x_{23}+4x_{33} \\ 3x_{11}+5x_{21}+x_{31} & 3x_{12}+5x_{22}+x_{32} & 3x_{13}+5x_{23}+x_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

By definition of equality of matrices, this gives us three systems of equations to solve. The augmented matrix of one of the systems, the one equating the first columns of the two matrices is:

$$\left(\begin{array}{ccc|c} 1 & 1 & 2 & 1 \\ 2 & 1 & 4 & 0 \\ 3 & 5 & 1 & 0 \end{array}\right) \tag{12.2.1}$$

Using the Gauss-Jordan algorithm, we have:

$$\left(\begin{array}{ccc|c} 1 & 1 & 2 & 1 \\ 2 & 1 & 4 & 0 \\ 3 & 5 & 1 & 0 \end{array}\right) \xrightarrow{-2R_1+R_2} \left(\begin{array}{ccc|c} 1 & 1 & 2 & 1 \\ 0 & -1 & 0 & -2 \\ 3 & 5 & 1 & 0 \end{array}\right) \xrightarrow{-3R_1+R_3} \left(\begin{array}{ccc|c} 1 & 1 & 2 & 1 \\ 0 & -1 & 0 & -2 \\ 0 & 2 & -5 & -3 \end{array}\right)$$

$$\xrightarrow{-1R_2} \left(\begin{array}{ccc|c} 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 2 & -5 & -3 \end{array}\right)$$

$$\xrightarrow{-R_2+R_1 \text{ and} -2R_2+R_3} \left(\begin{array}{ccc|c} 1 & 0 & 2 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & -5 & -7 \end{array}\right)$$

$$\xrightarrow{-\frac{1}{5}R_3} \left(\begin{array}{ccc|c} 1 & 0 & 2 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 7/5 \end{array}\right) \xrightarrow{-2R_3+R_1} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -\frac{19}{5} \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & \frac{7}{5} \end{array}\right)$$

So $x_{11} = -19/5, x_{21} = 2$ and $x_{31} = 7/5$, which gives us the first column of B.

The matrix form of the system to obtain x_{12}, x_{22}, and x_{32}, the second column of B, is:

$$\left(\begin{array}{ccc|c} 1 & 1 & 2 & 0 \\ 2 & 1 & 4 & 1 \\ 3 & 5 & 1 & 0 \end{array}\right) \tag{12.2.2}$$

which reduces to

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & \frac{9}{5} \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -\frac{2}{5} \end{array}\right) \tag{12.2.3}$$

The critical thing to note here is that the coefficient matrix in (12.2.2) is the same as the matrix in (12.2.1), hence the sequence of row operations that we used in row reduction are the same in both cases.

To determine the third column of B, we reduce

$$\left(\begin{array}{ccc|c} 1 & 1 & 2 & 0 \\ 2 & 1 & 4 & 0 \\ 3 & 5 & 1 & 1 \end{array}\right)$$

to obtain $x_{13} = 2/5, x_{23} = 0$ and $x_{33} = -1/5$. Here again it is important to note that the sequence of row operations used to solve this system is exactly the same as those we used in the first system. Why not save ourselves a considerable amount of time and effort and solve all three systems simultaneously? This we can do this by augmenting the coefficient matrix by the identity matrix I. We then have, by applying the same sequence of row operations as above,

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 2 & 1 & 0 & 0 \\ 2 & 1 & 4 & 0 & 1 & 0 \\ 3 & 5 & 1 & 0 & 0 & 1 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -\frac{19}{5} & \frac{9}{5} & \frac{2}{5} \\ 0 & 1 & 0 & 2 & -1 & 0 \\ 0 & 0 & 1 & \frac{7}{5} & -\frac{2}{5} & -\frac{1}{5} \end{array}\right)$$

So that

$$B = \begin{pmatrix} -\frac{19}{5} & \frac{9}{5} & \frac{2}{5} \\ 2 & -1 & 0 \\ \frac{7}{5} & -\frac{2}{5} & -\frac{1}{5} \end{pmatrix}$$

The reader should verify that $BA = I$ so that $A^{-1} = B$.

12.2.2: General Method for Computing Inverses

As the following theorem indicates, the verification that $BA = I$ is not necessary. The proof of the theorem is beyond the scope of this text. The interested reader can find it in most linear algebra texts.

Theorem 12.2.1

Let A be an $n \times n$ matrix. If a matrix B can be found such that $AB = I$, then $BA = I$, so that $B = A^{-1}$. In fact, to find A^{-1}, we need only find a matrix B that satisfies one of the two conditions $AB = I$ or $BA = I$.

It is clear from Chapter 5 and our discussions in this chapter that not all $n \times n$ matrices have inverses. How do we determine whether a matrix has an inverse using this method? The answer is quite simple: the technique we developed to compute inverses is a matrix approach to solving several systems of equations simultaneously.

Example 12.2.1: Recognition of a Non-Invertible Matrix

The reader can verify that if $A = \begin{pmatrix} 1 & 2 & 1 \\ -1 & -2 & -1 \\ 0 & 5 & 8 \end{pmatrix}$ then the augmented matrix $\left(\begin{array}{ccc|ccc} 1 & 2 & 1 & 1 & 0 & 0 \\ -1 & -2 & -2 & 0 & 1 & 0 \\ 0 & 5 & 8 & 0 & 0 & 1 \end{array}\right)$ reduces to

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 5 & 8 & 0 & 0 & 1 \end{array}\right) \tag{12.2.4}$$

Although this matrix can be row-reduced further, it is not necessary to do so since, in equation form, we have:

Table 12.2.1

$x_{11}+2x_{21}+x_{31}=1$ $0=1$ $5x_{21}+8x_{31}=0$	$x_{12}+2x_{22}+x_{32}=0$ $0=1$ $5x_{22}+8x_{32}=0$	$x_{13}+2x_{23}+x_{33}=0$ $0=0$ $5x_{23}+8x_{33}=1$

Clearly, there are no solutions to the first two systems, therefore A^{-1} does not exist. From this discussion it should be obvious to the reader that the zero row of the coefficient matrix together with the nonzero entry in the fourth column of that row in matrix (12.2.4) tells us that A^{-1} does not exist.

12.2.3: Exercises

Exercise 12.2.1

In order to develop an understanding of the technique of this section, work out all the details of Example 12.2.1.

Exercise 12.2.2

Use the method of this section to find the inverses of the following matrices whenever possible. If an inverse does not exist, explain why.

a. $\begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}$

b. $\begin{pmatrix} 0 & 3 & 2 & 5 \\ 1 & -1 & 4 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 3 & -1 \end{pmatrix}$

c. $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$

d. $\begin{pmatrix} 1 & 2 & -1 \\ -2 & -3 & 1 \\ 1 & 4 & -3 \end{pmatrix}$

e. $\begin{pmatrix} 6 & 7 & 2 \\ 4 & 2 & 1 \\ 6 & 1 & 1 \end{pmatrix}$

f. $\begin{pmatrix} 2 & 1 & 3 \\ 4 & 2 & 1 \\ 8 & 2 & 4 \end{pmatrix}$

Exercise 12.2.3

Use the method of this section to find the inverses of the following matrices whenever possible. If an inverse does not exist, explain why.

a. $\begin{pmatrix} \frac{1}{3} & 2 \\ \frac{1}{5} & -1 \end{pmatrix}$

b. $\begin{pmatrix} 1 & 0 & 0 & 3 \\ 2 & -1 & 0 & 6 \\ 0 & 2 & 1 & 0 \\ 0 & -1 & 3 & 2 \end{pmatrix}$

c. $\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$

d. $\begin{pmatrix} 1 & 0 & 0 \\ 2 & 2 & -1 \\ 1 & -1 & 1 \end{pmatrix}$

e. $\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{pmatrix}$

f. $\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$

Answer

a. $\begin{pmatrix} \frac{15}{11} & \frac{30}{11} \\ \frac{3}{11} & -\frac{5}{11} \end{pmatrix}$

b. $\left(\begin{array}{cccc|c} -20 & \frac{21}{2} & \frac{9}{2} & -\frac{3}{2} \\ 2 & -1 & 0 & 0 \\ -4 & 2 & 1 & 0 \\ 7 & -\frac{7}{2} & -\frac{3}{2} & \frac{1}{2} \end{array}\right)$

c. The inverse does not exist. When the augmented matrix is row-reduced (see below), the last row of the first half cannot be manipulated to match the identity matrix.

d. $\begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 1 \\ -4 & 1 & 2 \end{pmatrix}$

e. The inverse does not exist.

f. $\begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix}$

Exercise 12.2.4

a. Find the inverses of the following matrices.

i. $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix}$

ii. $\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & \frac{5}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{7} & 0 \\ 0 & 0 & 0 & \frac{3}{4} \end{pmatrix}$

b. If D is a diagonal matrix whose diagonal entries are nonzero, what is D^{-1}?

Exercise 12.2.5

Express each system of equations in Exercise 12.1.1 in the form $Ax = B$. When possible, solve each system by first finding the inverse of the matrix of coefficients.

Answer

The solutions are in the solution section of Section 12.1, Exercise 12.1.1, We illustrate with the outline of the solution to part (c). The matrix version of the system is

$$\begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & -1 \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 5 \end{pmatrix}$$

We compute the inverse of the matrix of coefficients and get

$$A^{-1} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & -1 \\ 1 & 3 & 1 \end{pmatrix}^{-1} = \frac{1}{5} \begin{pmatrix} 5 & 5 & -5 \\ -2 & -1 & 3 \\ 1 & -2 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = A^{-1} \begin{pmatrix} 1 \\ -1 \\ 5 \end{pmatrix} = \begin{pmatrix} -5 \\ \frac{14}{5} \\ \frac{8}{5} \end{pmatrix}$$

This page titled 12.2: Matrix Inversion is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

12.3: An Introduction to Vector Spaces

12.3.1: Motivation for the Study of Vector Spaces

When we encountered various types of matrices in Chapter 5, it became apparent that a particular kind of matrix, the diagonal matrix, was much easier to use in computations. For example, if $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$, then A^5 can be found, but its computation is tedious. If $D = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$ then

$$D^5 = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}^5 = \begin{pmatrix} 1^5 & 0 \\ 0 & 4^5 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1024 \end{pmatrix}$$

Even when presented with a non-diagonal matrix, we will see that it is sometimes possible to do a bit of work to be able to work with a diagonal matrix. This process is called ***diagonalization***.

In a variety of applications it is beneficial to be able to diagonalize a matrix. In this section we will investigate what this means and consider a few applications. In order to understand when the diagonalization process can be performed, it is necessary to develop several of the underlying concepts of linear algebra.

12.3.2: Vector Spaces

By now, you realize that mathematicians tend to generalize. Once we have found a "good thing," something that is useful, we apply it to as many different concepts as possible. In doing so, we frequently find that the "different concepts" are not really different but only look different. Four sentences in four different languages might look dissimilar, but when they are translated into a common language, they might very well express the exact same idea.

Early in the development of mathematics, the concept of a vector led to a variety of applications in physics and engineering. We can certainly picture vectors, or "arrows," in the $xy-$ plane and even in the three-dimensional space. Does it make sense to talk about vectors in four-dimensional space, in ten-dimensional space, or in any other mathematical situation? If so, what is the essence of a vector? Is it its shape or the rules it follows? The shape in two- or three-space is just a picture, or geometric interpretation, of a vector. The essence is the rules, or properties, we wish vectors to follow so we can manipulate them algebraically. What follows is a definition of what is called a vector space. It is a list of all the essential properties of vectors, and it is the basic definition of the branch of mathematics called linear algebra.

Definition 12.3.1: Vector Space

Let V be any nonempty set of objects. Define on V an operation, called addition, for any two elements $\vec{x}, \vec{y} \in V$, and denote this operation by $\vec{x} + \vec{y}$. Let scalar multiplication be defined for a real number $a \in \mathbb{R}$ and any element $\vec{x} \in V$ and denote this operation by $a\vec{x}$. The set V together with operations of addition and scalar multiplication is called a vector space over $\mathbb{R}$ if the following hold for all $\vec{x}, \vec{y}, \vec{z} \in V$, and $a, b \in \mathbb{R}$:

- $\vec{x} + \vec{y} = \vec{y} + \vec{x}$
- $(\vec{x} + \vec{y}) + \vec{z} = \vec{x} + (\vec{y} + \vec{z})$
- There exists a vector $\vec{0} \in V$, such that $\vec{x} + \vec{0} = \vec{x}$ for all $x \in V$.
- For each vector $\vec{x} \in V$, there exists a unique vector $-\vec{x} \in V$, such that $-\vec{x} + \vec{x} = \vec{0}$.

These are the main properties associated with the operation of addition. They can be summarized by saying that $[V; +]$ is an abelian group.

The next four properties are associated with the operation of scalar multiplication and how it relates to vector addition.

- $a(\vec{x} + \vec{y}) = a\vec{x} + a\vec{y}$
- $(a + b)\vec{x} = a\vec{x} + b\vec{x}$
- $a(b\vec{x}) = (ab)\vec{x}$
- $1\vec{x} = \vec{x}$.

In a vector space it is common to call the elements of V vectors and those from $\mathbb{R}$ scalars. Vector spaces over the real numbers are also called real vector spaces.

Example 12.3.1: A Vector Space of Matrices

Let $V = M_{2\times 3}(\mathbb{R})$ and let the operations of addition and scalar multiplication be the usual operations of addition and scalar multiplication on matrices. Then V together with these operations is a real vector space. The reader is strongly encouraged to verify the definition for this example before proceeding further (see Exercise $12.3.3$ of this section). Note we can call the elements of $M_{2\times 3}(\mathbb{R})$ vectors even though they are not arrows.

Example 12.3.2: The Vector Space $\mathbb{R}^2$

Let $\mathbb{R}^2 = \{(a_1, a_2) \mid a_1, a_2 \in \mathbb{R}\}$. If we define addition and scalar multiplication the natural way, that is, as we would on 1×2 matrices, then $\mathbb{R}^2$ is a vector space over $\mathbb{R}$. See Exercise $12.3.4$ of this section.

In this example, we have the "bonus" that we can illustrate the algebraic concept geometrically. In mathematics, a "geometric bonus" does not always occur and is not necessary for the development or application of the concept. However, geometric illustrations are quite useful in helping us understand concepts and should be utilized whenever available.

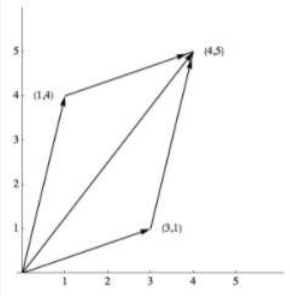


Figure 12.3.1: Sum of two vectors in $\mathbb{R}^2$

Let's consider some illustrations of the vector space $\mathbb{R}^2$. Let $\vec{x} = (1, 4)$ and $\vec{y} = (3, 1)$. We illustrate the vector (a_1, a_2) as a directed line segment, or "arrow," from the point $(0, 0)$ to the point (a_1, a_2). The vectors $\vec{x}$ and $\vec{y}$ are as shown in Figure $12.3.1$ together with $\vec{x} + \vec{y} = (1, 4) + (3, 1) = (4, 5)$. The vector $2\vec{x} = 2(1, 4) = (2, 8)$ is a vector in the same direction as $\vec{x}$, but with twice its length.

Note 12.3.1

1. The common convention is to use that boldface letters toward the end of the alphabet for vectors, while letters early in the alphabet are scalars.
2. A common alternate notation for vectors is to place an arrow about a variable to indicate that it is a vector such as this: $\overrightarrow{x}$.
3. The vector $(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$ is referred to as an n-tuple.
4. For those familiar with vector calculus, we are expressing the vector $x = a_1\hat{\mathbf{i}} + a_2\hat{\mathbf{j}} + a_3\hat{\mathbf{k}} \in \mathbb{R}^3$ as (a_1, a_2, a_3). This allows us to discuss vectors in $\mathbb{R}^n$ in much simpler notation.

In many situations a vector space V is given and we would like to describe the whole vector space by the smallest number of essential reference vectors. An example of this is the description of $\mathbb{R}^2$, the xy-plane, via the x and y axes. Again our concepts must be algebraic in nature so we are not restricted solely to geometric considerations.

Definition 12.3.2: Linear Combination

A vector $\boldsymbol{y}$ in vector space V (over $\mathbb{R}$) is a linear combination of the vectors $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ if there exist scalars $a_1, a_2, \ldots, a_n$ in $\mathbb{R}$ such that $\vec{y} = a_1\vec{x}_1 + a_2\vec{x}_2 + \ldots + a_n\vec{x}_n$

Example 12.3.3: A Basic Example

The vector $(2,3)$ in $\mathbb{R}^2$ is a linear combination of the vectors $(1,0)$ and $(0,1)$ since $(2,3)=2(1,0)+3(0,1)$.

Example 12.3.4: A Little Less Obvious Example

Prove that the vector $(4,5)$ is a linear combination of the vectors (3, 1) and (1, 4).

By the definition we must show that there exist scalars a_1 and a_2 such that:

$$\begin{aligned}(4,5) &= a_1(3,1)+a_2(1,4) \\ &= (3a_1+a_2, a_1+4a_2)\end{aligned} \quad \Rightarrow \quad \begin{aligned}3a_1+a_2&=4\\ a_1+4a_2&=5\end{aligned}$$

This system has the solution $a_1=1,\ a_2=1$.

Hence, if we replace a_1 and a_2 both by 1, then the two vectors (3, 1) and (1, 4) produce, or generate, the vector (4,5). Of course, if we replace a_1 and a_2 by different scalars, we can generate more vectors from $\mathbb{R}^2$. If, for example, $a_1=3$ and $a_2=-2$, then

$$a_1(3,1)+a_2(1,4)=3(3,1)+(-2)(1,4)=(9,3)+(-2,-8)=(7,-5)$$

Will the vectors $(3,1)$ and $(1,4)$ generate any vector we choose in $\mathbb{R}^2$? To see if this is so, we let (b_1,b_2) be an arbitrary vector in $\mathbb{R}^2$ and see if we can always find scalars a_1 and a_2 such that $a_1(3,1)+a_2(1,4)=(b_1,b_2)$. This is equivalent to solving the following system of equations:

$$\begin{aligned}3a_1+a_2&=b_1\\ a_1+4a_2&=b_2\end{aligned}$$

which always has solutions for a_1 and a_2 , regardless of the values of the real numbers b_1 and b_2. Why? We formalize this situation in a definition:

Definition 12.3.3: Generation of a Vector Space

Let $\{\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_n\}$ be a set of vectors in a vector space V over $\mathbb{R}$. This set is said to ***generate***, or span, V if, for any given vector $\vec{y}\in V$, we can always find scalars $a_1,\ a_2,\ldots,\ a_n$ such that $\vec{y}=a_1\vec{x}_1+a_2\vec{x}_2+\ldots+a_n\vec{x}_n$. A set that generates a vector space is called a ***generating set***.

We now give a geometric interpretation of the previous examples.

We know that the standard coordinate system, x axis and y axis, were introduced in basic algebra in order to describe all points in the xy-plane algebraically. It is also quite clear that to describe any point in the plane we need exactly two axes.

We can set up a new coordinate system in the following way. Draw the vector $(3,1)$ and an axis from the origin through (3, 1) and label it the x' axis. Also draw the vector $(1,4)$ and an axis from the origin through $(1,4)$ to be labeled the y' axis. Draw the coordinate grid for the axis, that is, lines parallel, and let the unit lengths of this "new" plane be the lengths of the respective vectors, $(3,1)$ and $(1,4)$, so that we obtain Figure 12.3.2

From Example 12.3.4 and Figure 12.3.2, we see that any vector on the plane can be described using the standard xy-axes or our new $x'y'$-axes. Hence the position which had the name $(3,1)$ in reference to the standard axes has the name $(1,0)$ with respect to the $x'y'$ axes, or, in the phraseology of linear algebra, the coordinates of the point $(1,4)$ with respect to the $x'y'$ axes are $(1,0)$.

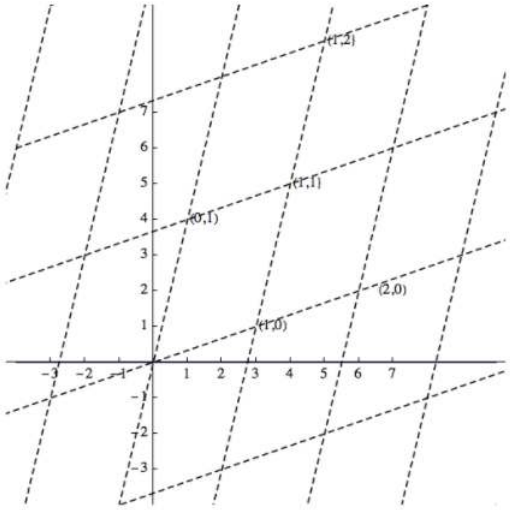


Figure 12.3.2: Two sets of axes for the plane

Example 12.3.5: One Point, Two Position Descriptions

From Example 12.3.4 we found that if we choose $a_1 = 1$ and $a_2 = 1$, then the two vectors $(3, 1)$ and $(1, 4)$ generate the vector $(4, 5)$. Another geometric interpretation of this problem is that the coordinates of the position $(4, 5)$ with respect to the $x'y'$ axes of Figure 12.3.2is $(1, 1)$. In other words, a position in the plane has the name $(4, 5)$ in reference to the xy-axes and the same position has the name $(1, 1)$ in reference to the $x'y'$ axes.

From the above, it is clear that we can use different axes to describe points or vectors in the plane. No matter what choice we use, we want to be able to describe each position in a unique manner. This is not the case in Figure 12.3.3 Any point in the plane could be described via the $x'y'$ axes, the $x'z'$ axes or the $y'z'$ axes. Therefore, in this case, a single point would have three different names, a very confusing situation.

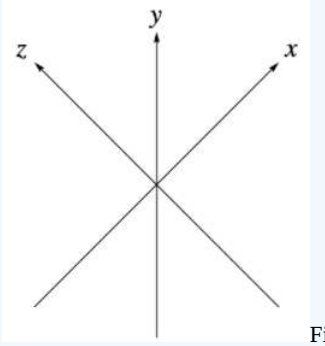


Figure 12.3.3: Three axes on a plane

We formalize the our observations in the previous examples in two definitions and a theorem.

Definition 12.3.4: Linear Independence/Linear Dependence

A set of vectors $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ from a real vector space V is ***linearly independent*** if the only solution to the equation $a_1\vec{x}_1 + a_2\vec{x}_2 + \ldots + a_n\vec{x}_n = \vec{0}$ is $a_1 = a_2 = \ldots = a_n = 0$. Otherwise the set is called a ***linearly dependent*** set.

Definition 12.3.5: Basis

A set of vectors $B = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ is a basis for a vector space V if:

1. B generates V, and
2. B is linearly independent.

Theorem 12.3.1: The Fundamental Property of a Basis

If $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ is a basis for a vector space V over $\mathbb{R}$, then any vector $y \in V$ can be uniquely expressed as a linear combination of the $\vec{x}_i$'s.

Proof

Assume that $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ is a basis for V over $\mathbb{R}$. We must prove two facts:

1. each vector $y \in V$ can be expressed as a linear combination of the $\vec{x}_i$'s, and
2. each such expression is unique.

Part 1 is trivial since a basis, by its definition, must generate all of V.

The proof of part 2 is a bit more difficult. We follow the standard approach for any uniqueness facts. Let y be any vector in V and assume that there are two different ways of expressing y, namely

$$y = a_1\vec{x}_1 + a_2\vec{x}_2 + \ldots + a_n\vec{x}_n$$

and

$$y = b_1\vec{x}_1 + b_2\vec{x}_2 + \ldots + b_n\vec{x}_n$$

where at least one a_i is different from the corresponding b_i. Then equating these two linear combinations we get

$$a_1\vec{x}_1 + a_2\vec{x}_2 + \ldots + a_n\vec{x}_n = b_1\vec{x}_1 + b_2\vec{x}_2 + \ldots + b_n\vec{x}_n$$

so that

$$(a_1 - b_1)\,\vec{x}_1 + (a_2 - b_2)\,\vec{x}_2 + \ldots + (a_n - b_n)\,\vec{x}_n = \vec{0}$$

Now a crucial observation: since the $\vec{x}_i's$ form a linearly independent set, the only solution to the previous equation is that each of the coefficients must equal zero, so $a_i - b_i = 0$ for $i = 1, 2, \ldots, n$. Hence $a_i = b_i,$ for all i. This contradicts our assumption that at least one a_i is different from the corresponding b_i, so each vector $\vec{y} \in V$ can be expressed in one and only one way.

This theorem, together with the previous examples, gives us a clear insight into the significance of linear independence, namely uniqueness in representing any vector.

Example 12.3.6: Another Basis for $\mathbb{R}^2$

Prove that $\{(1,1), (-1,1)\}$ is a basis for $\mathbb{R}^2$ over $\mathbb{R}$ and explain what this means geometrically.

First we show that the vectors $(1,1)$ and $(-1,1)$ generate all of $\mathbb{R}^2$. We can do this by imitating Example 12.3.4 and leave it to the reader (see Exercise 12.3.10 of this section). Secondly, we must prove that the set is linearly independent.

Let a_1 and a_2 be scalars such that $a_1(1,1) + a_2(-1,1) = (0,0)$. We must prove that the only solution to the equation is that a_1 and a_2 must both equal zero. The above equation becomes $(a_1 - a_2, a_1 + a_2) = (0,0)$ which gives us the system

$$a_1 - a_2 = 0$$
$$a_1 + a_2 = 0$$

The augmented matrix of this system reduces in such way that the only solution is the trivial one of all zeros:

$$\left(\begin{array}{cc|c} 1 & -1 & 0 \\ 1 & 1 & 0 \end{array}\right) \longrightarrow \left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right) \Rightarrow a_1 = a_2 = 0$$

Therefore, the set is linearly independent.

To explain the results geometrically, note through Exercise 12, part a, that the coordinates of each vector $\vec{y} \in \mathbb{R}^2$ can be determined uniquely using the vectors (1,1) and (-1, 1). The concept of dimension is quite obvious for those vector spaces that have an immediate geometric interpretation. For example, the dimension of $\mathbb{R}^2$ is two and that of $\mathbb{R}^3$ is three. How can we define the concept of dimension algebraically so that the resulting definition correlates with that of $\mathbb{R}^2$ and $\mathbb{R}^3$? First we need a theorem, which we will state without proof.

Theorem 12.3.2: Basis Size is Constant

If V is a vector space with a basis containing n elements, then all bases of V contain n elements.

Definition 12.3.6: Dimension of a Vector Space

Let V be a vector space over $\mathbb{R}$ with basis $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$. Then the dimension of V is n. We use the notation $\dim V = n$ to indicate that V is n-dimensional.

12.3.3: Exercises

Exercise 12.3.1

If $a = 2,\ b = -3,\ A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 3 & 4 \end{pmatrix},\ B = \begin{pmatrix} 2 & -2 & 3 \\ 4 & 5 & 8 \end{pmatrix}$, and $C = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 2 & -2 \end{pmatrix}$ verify that all properties of the definition of a vector space are true for $M_{2\times 3}(\mathbb{R})$ with these values.

Exercise 12.3.2

Let $a = 3,\ b = 4,\ \vec{x}$=$(-1, 3)$, $\vec{y} = (2, 3)$,and $\vec{z} = (1, 0)$. Verify that all properties of the definition of a vector space are true for $\mathbb{R}^2$ for these values.

Exercise 12.3.3

a. Verify that $M_{2\times 3}(\mathbb{R})$ is a vector space over $\mathbb{R}$. What is its dimension?
b. Is $M_{m\times n}(\mathbb{R})$ a vector space over $\mathbb{R}$? If so, what is its dimension?

Answer

The dimension of $M_{2\times 3}(\mathbb{R})$ is 6 and yes, $M_{m\times n}(\mathbb{R})$ is also a vector space of dimension $m \cdot n$. One basis for $M_{m\times n}(\mathbb{R})$ is $\{A_{ij} \mid 1 \le i \le m, 1 \le j \le n\}$ where A_{ij} is the $m \times n$ matrix with entries all equal to zero except for in row i, column j where the entry is 1.

Exercise 12.3.4

a. Verify that $\mathbb{R}^2$ is a vector space over $\mathbb{R}$.
b. Is $\mathbb{R}^n$ a vector space over $\mathbb{R}$ for every positive integer n?

Exercise 12.3.5

Let $P^3 = \{a_0 + a_1x + a_2x^2 + a_3x^3 \mid a_0, a_1, a_2, a_3 \in \mathbb{R}\}$; that is, P^3 is the set of all polynomials in x having real coefficients with degree less than or equal to three. Verify that P^3 is a vector space over $\mathbb{R}$. What is its dimension?

Exercise 12.3.6

For each of the following, express the vector $\boldsymbol{y}$ as a linear combination of the vectors x_1 and x_2.

a. $\vec{y} = (5, 6)$, $\vec{x}_1 = (1, 0)$, and $\vec{x}_2 = (0, 1)$
b. $\vec{y} = (2, 1)$, $\vec{x}_1 = (2, 1)$, and $\vec{x}_2 = (1, 1)$
c. $\vec{y} = (3, 4)$, $\vec{x}_1 = (1, 1)$, and $\vec{x}_2 = (-1, 1)$

Exercise 12.3.7

Express the vector $\begin{pmatrix} 1 & 2 \\ -3 & 3 \end{pmatrix} \in M_{2\times 2}(\mathbb{R})$, as a linear combination of $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 5 \\ 2 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

Answer

If the matrices are named B, A_1, A_2 , A_3, and A_4 , then

$$B = \frac{8}{3}A_1 + \frac{5}{3}A_2 + \frac{-5}{3}A_3 + \frac{23}{3}A_4$$

Exercise 12.3.8

Express the vector $x^3 - 4x^2 + 3 \in P^3$ as a linear combination of the vectors 1, x, x^2 , and x^3.

Exercise 12.3.9

a. Show that the set $\{\vec{x}_1, \vec{x}_2\}$ generates $\mathbb{R}^2$ for each of the parts in Exercise 12.3.6 of this section.
b. Show that $\{\vec{x}_1, \vec{x}_2, \vec{x}_3\}$ generates $\mathbb{R}^2$ where $\vec{x}_1 = (1, 1)$, $\vec{x}_2 = (3, 4)$, and $\vec{x}_3 = (-1, 5)$.
c. Create a set of four or more vectors that generates $\mathbb{R}^2$.
d. What is the smallest number of vectors needed to generate $\mathbb{R}^2$? $\mathbb{R}^n$?
e. Show that the set

$$\{A_1, A_2, A_3, A_4\} = \{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\}$$

generates $M_{2\times 2}(\mathbb{R})$
f. Show that $\{1, x, x^2, x^3\}$ generates P^3.

Answer

a. If $x_1 = (1, 0)$, $x_2 = (0, 1)$, and $y = (b_1, b_2)$, then $y = b_1x_1 + b_2x_2$. If $x_1 = (3, 2)$, $x_2 = (2, 1)$, and $y = (b_1, b_2)$, then $y = (-b_1 + 2b_2)\, x_1 + (2b_1 - 3b_2)\, x_2$.
b. If $y = (b_1, b_2)$ is any vector in $\mathbb{R}^2$, then $y = (-3b_1 + 4b_2)\, x_1 + (-b_1 + b_2)\, x_2 + (0)x_3$
c. One solution is to add any vector(s) to x_1, x_2, and x_3 of part b.
d. 2, n
e. $\begin{pmatrix} x & y \\ z & w \end{pmatrix} = xA_1 + yA_2 + zA_3 + wA_4$
f. $a_0 + a_1x + a_2x^2 + a_3x^3 = a_0(1) + a_1(x) + a_2\left(x^2\right) + a_3\left(x^3\right)$.

Exercise 12.3.10

Complete Example 12.3.6 by showing that $\{(1,1),(-1,1)\}$ generates $\mathbb{R}^2$.

Exercise 12.3.11

a. Prove that $\{(4,1),(1,3)\}$ is a basis for $\mathbb{R}^2$ over $\mathbb{R}$.
b. Prove that $\{(1,0),(3,4)\}$ is a basis for $\mathbb{R}^2$ over $\mathbb{R}$.
c. Prove that $\{(1,0,-1),(2,1,1),(1,-3,-1)\}$ is a basis for $\mathbb{R}^3$ over $\mathbb{R}$.
d. Prove that the sets in Exercise 12.3.9, parts e and f, form bases of the respective vector spaces.

Answer

a. The set is linearly independent: let a and b be scalars such that $a(4,1)+b(1,3)=(0,0)$, then $4a+b=0$ and $a+3b=0$ which has $a=b=0$ as its only solutions. The set generates all of $\mathbb{R}^2$: let (a,b) be an arbitrary vector in $\mathbb{R}^2$. We want to show that we can always find scalars β_1 and β_2 such that $\beta_1(4,1)+\beta_2(1,3)=(a,b)$. This is equivalent to finding scalars such that $4\beta_1+\beta_2=a$ and $\beta_1+3\beta_2=b$. This system has a unique solution $\beta_1=\frac{3a-b}{11}$, and $\beta_2=\frac{4b-a}{11}$. Therefore, the set generates $\mathbb{R}^2$.

Exercise 12.3.12

a. Determine the coordinates of the points or vectors $(3,4)$, $(-1,1)$, and $(1,1)$ with respect to the basis $\{(1,1),(-1,1)\}$ of $\mathbb{R}^2$. Interpret your results geometrically.
b. Determine the coordinates of the points or vector $(3,5,6)$ with respect to the basis $\{(1,0,0),(0,1,0),(0,0,1)\}$. Explain why this basis is called the standard basis for $\mathbb{R}^3$.

Exercise 12.3.13

a. Let $\vec{y}_1=(1,3,5,9)$, $\vec{y}_2=(5,7,6,3)$, and $c=2$. Find $\vec{y}_1+\vec{y}_2$ and $c\vec{y}_1$.
b. Let $f_1(x)=1+3x+5x^2+9x^3$, $f_2(x)=5+7x+6x^2+3x^3$ and $c=2$. Find $f_1(x)+f_2(x)$ and $cf_1(x)$.
c. Let $A=\begin{pmatrix}1&3\\5&9\end{pmatrix}$, $B=\begin{pmatrix}5&7\\6&3\end{pmatrix}$, and $c=2$. Find $A+B$ and cA.
d. Are the vector spaces $\mathbb{R}^4$, P^3 and $M_{2\times 2}(\mathbb{R})$ isomorphic to each other? Discuss with reference to previous parts of this exercise.

Answer

The answer to the last part is that the three vector spaces are all isomorphic to one another. Once you have completed part (a) of this exercise, the following translation rules will give you the answer to parts (b) and (c),

$$(a,b,c,d)\leftrightarrow\begin{pmatrix}a&b\\c&d\end{pmatrix}\leftrightarrow a+bx+cx^2+dx^2$$

This page titled 12.3: An Introduction to Vector Spaces is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

12.4: The Diagonalization Process

12.4.1: Eigenvalues and Eigenvectors

We now have the background to understand the main ideas behind the diagonalization process.

Definition 12.4.1: Eigenvalue, Eigenvector

Let A be an $n \times n$ matrix over $\mathbb{R}$. λ is an eigenvalue of A if for some nonzero column vector $\vec{x} \in \mathbb{R}^n$ we have $A\vec{x} = \lambda\vec{x}$. $\vec{x}$ is called an ***eigenvector*** corresponding to the ***eigenvalue*** λ.

Example 12.4.1: Examples of Eigenvalues and Eigenvectors

Find the eigenvalues and corresponding eigenvectors of the matrix $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$.

We want to find nonzero vectors $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and real numbers λ such that

$$\begin{aligned} AX = \lambda X \quad &\Leftrightarrow \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &\Leftrightarrow \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &\Leftrightarrow \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &\Leftrightarrow \left(\begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &\Leftrightarrow \begin{pmatrix} 2-\lambda & 1 \\ 2 & 3-\lambda \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned} \tag{12.4.1}$$

The last matrix equation will have nonzero solutions if and only if

$$\det \begin{pmatrix} 2-\lambda & 1 \\ 2 & 3-\lambda \end{pmatrix} = 0$$

or $(2-\lambda)(3-\lambda) - 2 = 0$, which simplifies to $\lambda^2 - 5\lambda + 4 = 0$. Therefore, the solutions to this quadratic equation, $\lambda_1 = 1$ and $\lambda_2 = 4$, are the eigenvalues of A. We now have to find eigenvectors associated with each eigenvalue.

Case 1. For $\lambda_1 = 1$, (12.4.1) becomes:

$$\begin{pmatrix} 2-1 & 1 \\ 2 & 3-1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

which reduces to the single equation, $x_1 + x_2 = 0$. From this, $x_1 = -x_2$. This means the solution set of this equation is (in column notation)

$$E_1 = \left\{ \begin{pmatrix} -c \\ c \end{pmatrix} \middle| \, c \in \mathbb{R} \right\}$$

So any column vector of the form $\begin{pmatrix} -c \\ c \end{pmatrix}$ where c is any nonzero real number is an eigenvector associated with $\lambda_1 = 1$. The reader should verify that, for example,

$$\begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

so that $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ is an eigenvector associated with eigenvalue 1.

Case 2. For $\lambda_2 = 4$ (12.4.1) becomes:

$$\begin{pmatrix} 2-4 & 1 \\ 2 & 3-4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$\begin{pmatrix} -2 & 1 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

which reduces to the single equation $-2x_1 + x_2 = 0$, so that $x_2 = 2x_1$. The solution set of the equation is

$$E_2 = \left\{ \begin{pmatrix} c \\ 2c \end{pmatrix} \middle| \, c \in \mathbb{R} \right\}$$

Therefore, all eigenvectors of A associated with the eigenvalue $\lambda_2 = 4$ are of the form $\begin{pmatrix} c \\ 2c \end{pmatrix}$, where c can be any nonzero number.

The following theorems summarize the most important aspects of the previous example.

Theorem 12.4.1: Characterization of Eigenvalues of a Square Matrix

Let A be any $n \times n$ matrix over $\mathbb{R}$. Then $\lambda \in \mathbb{R}$ is an eigenvalue of A if and only if $\det(A - \lambda I) = 0$.

The equation $\det(A - \lambda I) = 0$ is called the ***characteristic equation***, and the left side of this equation is called the ***characteristic polynomial*** of A.

Theorem 12.4.2: Linear Independence of Eigenvectors

Nonzero eigenvectors corresponding to distinct eigenvalues are linearly independent.

The solution space of $(A - \lambda I)\vec{x} = \vec{0}$ is called the eigenspace of A corresponding to λ. This terminology is justified by Exercise 2 of this section.

12.4.2: Diagonalization

We now consider the main aim of this section. Given an $n \times n$ (square) matrix A, we would like to transform A into a diagonal matrix D, perform our tasks with the simpler matrix D, and then describe the results in terms of the given matrix A.

Definition 12.4.2: Diagonalizable Matrix

An $n \times n$ matrix A is called diagonalizable if there exists an invertible $n \times n$ matrix P such that $P^{-1}AP$ is a diagonal matrix D. The matrix P is said to diagonalize the matrix A.

Example 12.4.2: Diagonalization of a Matrix

We will now diagonalize the matrix A of Example 12.4.1. We form the matrix P as follows: Let $P^{(1)}$ be the first column of P. Choose for $P^{(1)}$ any eigenvector from E_1. We may as well choose a simple vector in E_1 so $P^{(1)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ is our candidate.

Similarly, let $P^{(2)}$ be the second column of P, and choose for $P^{(2)}$ any eigenvector from E_2. The vector $P^{(2)} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ is a reasonable choice, thus

$$P = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} \text{ and } P^{-1} = \frac{1}{3}\begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

so that

$$P^{-1}AP = \frac{1}{3}\begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

Notice that the elements on the main diagonal of D are the eigenvalues of A, where D_{ii} is the eigenvalue corresponding to the eigenvector $P^{(i)}$.

Note 12.4.1

1. The first step in the diagonalization process is the determination of the eigenvalues. The ordering of the eigenvalues is purely arbitrary. If we designate $\lambda_1 = 4$ and $\lambda_2 = 1$, the columns of P would be interchanged and D would be $\begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$ (see Exercise 3b of this section). Nonetheless, the final outcome of the application to which we are applying the diagonalization process would be the same.
2. If A is an $n \times n$ matrix with distinct eigenvalues, then P is also an $n \times n$ matrix whose columns $P^{(1)}, P^{(2)}, \ldots, P^{(n)}$ are n linearly independent vectors.

Example 12.4.3: Diagonalization of a 3 by 3 Matrix

Diagonalize the matrix

$$A = \begin{pmatrix} 1 & 12 & -18 \\ 0 & -11 & 18 \\ 0 & -6 & 10 \end{pmatrix}.$$

First, we find the eigenvalues of A.

$$\begin{aligned} \det(A - \lambda I) &= \det\begin{pmatrix} 1-\lambda & 12 & -18 \\ 0 & -\lambda - 11 & 18 \\ 0 & -6 & 10-\lambda \end{pmatrix} \\ &= (1-\lambda)\det\begin{pmatrix} -\lambda - 11 & 18 \\ -6 & 10-\lambda \end{pmatrix} \\ &= (1-\lambda)((-\lambda-11)(10-\lambda)+108) = (1-\lambda)\left(\lambda^2+\lambda-2\right) \end{aligned}$$

Hence, the equation $\det(A - \lambda I)$ becomes

$$(1-\lambda)\left(\lambda^2+\lambda-2\right) = -(\lambda-1)^2(\lambda+2)$$

Therefore, our eigenvalues for A are $\lambda_1 = -2$ and $\lambda_2 = 1$. We note that we do not have three distinct eigenvalues, but we proceed as in the previous example.

Case 1. For $\lambda_1 = -2$ the equation $(A - \lambda I)\vec{x} = \vec{0}$ becomes

$$\begin{pmatrix} 3 & 12 & -18 \\ 0 & -9 & 18 \\ 0 & -6 & 12 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

We can row reduce the matrix of coefficients to $\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{pmatrix}$.

The matrix equation is then equivalent to the equations $x_1 = -2x_3$ and $x_2 = 2x_3$. Therefore, the solution set, or eigenspace, corresponding to $\lambda_1 = -2$ consists of vectors of the form

$$\begin{pmatrix} -2x_3 \\ 2x_3 \\ x_3 \end{pmatrix} = x_3\begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}$$

Therefore $\begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}$ is an eigenvector corresponding to the eigenvalue $\lambda_1 = -2$, and can be used for our first column of P:

$$P = \begin{pmatrix} -2 & ? & ? \\ 2 & ? & ? \\ 1 & ? & ? \end{pmatrix}$$

Before we continue we make the observation: E_1 is a subspace of $\mathbb{R}^3$ with basis $\left\{P^{(1)}\right\}$ and $\dim E_1 = 1$.

Case 2. If $\lambda_2 = 1$, then the equation $(A - \lambda I)\vec{x} = \vec{0}$ becomes

$$\begin{pmatrix} 0 & 12 & -18 \\ 0 & -12 & 18 \\ 0 & -6 & 9 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Without the aid of any computer technology, it should be clear that all three equations that correspond to this matrix equation are equivalent to $2x_2 - 3x_3 = 0$, or $x_2 = \frac{3}{2}x_3$. Notice that x_1 can take on any value, so any vector of the form

$$\begin{pmatrix} x_1 \\ \frac{3}{2}x_3 \\ x_3 \end{pmatrix} = x_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_3 \begin{pmatrix} 0 \\ \frac{3}{2} \\ 1 \end{pmatrix}$$

will solve the matrix equation.

We note that the solution set contains two independent variables, x_1 and x_3. Further, note that we cannot express the eigenspace E_2 as a linear combination of a single vector as in Case 1. However, it can be written as

$$E_2 = \left\{ x_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_3 \begin{pmatrix} 0 \\ \frac{3}{2} \\ 1 \end{pmatrix} \mid x_1, x_3 \in \mathbb{R} \right\}.$$

We can replace any vector in a basis is with a nonzero multiple of that vector. Simply for aesthetic reasons, we will multiply the second vector that generates E_2 by 2. Therefore, the eigenspace E_2 is a subspace of $\mathbb{R}^3$ with basis $\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \\ 2 \end{pmatrix} \right\}$ and so $\dim E_2 = 2$.

What this means with respect to the diagonalization process is that $\lambda_2 = 1$ gives us both Column 2 and Column 3 the diagonalizing matrix. The order is not important so we have

$$P = \begin{pmatrix} -2 & 1 & 0 \\ 2 & 0 & 3 \\ 1 & 0 & 2 \end{pmatrix}$$

The reader can verify (see Exercise 12.4.5 of this section) that $P^{-1} = \begin{pmatrix} 0 & 2 & -3 \\ 1 & 4 & -6 \\ 0 & -1 & 2 \end{pmatrix}$ and $P^{-1}AP = \begin{pmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

In doing Example 12.4.3, the given 3×3 matrix A produced only two, not three, distinct eigenvalues, yet we were still able to diagonalize A. The reason we were able to do so was because we were able to find three linearly independent eigenvectors. Again, the main idea is to produce a matrix P that does the diagonalizing. If A is an $n \times n$ matrix, P will be an $n \times n$ matrix, and its n columns must be linearly independent eigenvectors. The main question in the study of diagonalizability is "When can it be done?" This is summarized in the following theorem.

Theorem 12.4.3: A Condition for Diagonalizability

Let A be an $n \times n$ matrix. Then A is diagonalizable if and only if A has n linearly independent eigenvectors.

Proof

Outline of a proof: ($\Longleftarrow$) Assume that A has linearly independent eigenvectors, $P^{(1)}, P^{(2)}, \ldots, P^{(n)}$, with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. We want to prove that A is diagonalizable. Column i of the $n \times n$ matrix AP is $AP^{(i)}$ (see Exercise 12.4.7of this section). Then, since the $P^{(i)}$ is an eigenvector of A associated with the eigenvalue λ_i we have $AP^{(i)} = \lambda_i P^{(i)}$ for $i = 1, 2, \ldots, n$. But this means that $AP = PD$, where D is the diagonal matrix with diagonal entries $\lambda_1, \lambda_2, \ldots, \lambda_n$. If we multiply both sides of the equation by P^{-1} we get the desired $P^{-1}AP = D$.

($\Longrightarrow$) The proof in this direction involves a concept that is not covered in this text (rank of a matrix); so we refer the interested reader to virtually any linear algebra text for a proof.

We now give an example of a matrix that is not diagonalizable.

Example 12.4.4: A Matrix that is Not Diagonalizable

Let us attempt to diagonalize the matrix $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 1 & -1 & 4 \end{pmatrix}$

First, we determine the eigenvalues.

$$\begin{aligned} \det(A - \lambda I) &= \det \begin{pmatrix} 1-\lambda & 0 & 0 \\ 0 & 2-\lambda & 1 \\ 1 & -1 & 4-\lambda \end{pmatrix} \\ &= (1-\lambda) \det \begin{pmatrix} 2-\lambda & 1 \\ -1 & 4-\lambda \end{pmatrix} \\ &= (1-\lambda)((2-\lambda)(4-\lambda)+1) \\ &= (1-\lambda)\left(\lambda^2 - 6\lambda + 9\right) \\ &= (1-\lambda)(\lambda-3)^2 \end{aligned}$$

Therefore there are two eigenvalues, $\lambda_1 = 1$ and $\lambda_2 = 3$. Since λ_1 is an eigenvalue of degree one, it will have an eigenspace of dimension 1. Since λ_2 is a double root of the characteristic equation, the dimension of its eigenspace must be 2 in order to be able to diagonalize.

Case 1. For $\lambda_1 = 1$, the equation $(A - \lambda I)\vec{x} = \vec{0}$ becomes

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Row reduction of this system reveals one free variable and eigenspace

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -4x_3 \\ -x_3 \\ x_3 \end{pmatrix} = x_3 \begin{pmatrix} -4 \\ -1 \\ 1 \end{pmatrix}$$

Hence, $\left\{ \begin{pmatrix} -4 \\ -1 \\ 1 \end{pmatrix} \right\}$ is a basis for the eigenspace of $\lambda_1 = 1$.

Case 2. For $\lambda_2 = 3$, the equation $(A - \lambda I)\vec{x} = \vec{0}$ becomes

$$\begin{pmatrix} -2 & 0 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Once again there is only one free variable in the row reduction and so the dimension of the eigenspace will be one:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ x_3 \\ x_3 \end{pmatrix} = x_3 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Hence, $\left\{ \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ is a basis for the eigenspace of $\lambda_2 = 3$. This means that $\lambda_2 = 3$ produces only one column for P. Since we began with only two eigenvalues, we had hoped that $\lambda_2 = 3$ would produce a vector space of dimension two, or, in matrix terms, two linearly independent columns for P. Since A does not have three linearly independent eigenvectors A cannot be diagonalized.

12.4.3: SageMath Note - Diagonalization

We demonstrate how diagonalization can be done in Sage. We start by defining the matrix to be diagonalized, and also declare D and P to be variables.

```
var (' D, P')
A = Matrix (QQ, [[4, 1, 0], [1, 5, 1], [0, 1, 4]]);A
```

We have been working with “right eigenvectors” since the $\vec{x}$ in $A\vec{x} = \lambda\vec{x}$ is a column vector. It's not so common but still desirable in some situations to consider “left eigenvectors,” so SageMath allows either one. The `right_eigenmatrix` method returns a pair of matrices. The diagonal matrix, D, with eigenvalues and the diagonalizing matrix, P, which is made up of columns that are eigenvectors corresponding to the eigenvectors of D.

```
(D,P)=A.right_eigenmatrix();(D,P)
```

We should note here that P is not unique because even if an eigenspace has dimension one, any nonzero vector in that space will serve as an eigenvector. For that reason, the P generated by Sage isn't necessarily the same as the one computed by any other computer algebra system such as Mathematica. Here we verify the result for our Sage calculation. Recall that an asterisk is used for matrix multiplication in Sage.

```
P.inverse()*A*P
```

Here is a second matrix to diagonalize.

```
A2=Matrix(QQ,[[8,1,0],[1,5,1],[0,1,7]]);A2
```

Here we've already specified that the underlying system is the rational numbers. Since the eigenvalues are not rational, Sage will revert to approximate number by default. We'll just pull out the matrix of eigenvectors this time and display rounded entries.

```
P=A2.right_eigenmatrix()[1]
P.numerical_approx(digits=3)
print('------------------')
D=(P.inverse()*A2*P);D.numerical_approx(digits=3)
```

Finally, we examine how Sage reacts to the matrix from Example $12.4.4$ that couldn't be diagonalized. Notice that the last column is a zero column, indicating the absence of one needed eigenvector.

```
A3=Matrix(QQ,[[1, 0, 0],[0,2,1],[1,-1,4]])
(D,P)=A3.right_eigenmatrix();(D,P)
```

12.4.4: Exercises

Exercise 12.4.1

a. List three different eigenvectors of $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$, the matrix of Example 12.4.1, associated with each of the two eigenvalues 1 and 4. Verify your results.
b. Choose one of the three eigenvectors corresponding to 1 and one of the three eigenvectors corresponding to 4, and show that the two chosen vectors are linearly independent.

Answer

a. Any nonzero multiple of $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ is an eigenvector associated with $\lambda = 1$.
b. Any nonzero multiple of $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ is an eigenvector associated with $\lambda = 4$.
c. Let $x_1 = \begin{pmatrix} a \\ -a \end{pmatrix}$ and $x_2 = \begin{pmatrix} b \\ 2b \end{pmatrix}$. You can verify that $c_1 x_1 + c_2 x_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ if and only if $c_1 = c_2 = 0$. Therefore, $\{x_1, x_2\}$ is linearly independent.

Exercise 12.4.2

a. Verify that E_1 and E_2 in Example 12.4.1 are vector spaces over $\mathbb{R}$. Since they are also subsets of $\mathbb{R}^2$, they are called subvector-spaces, or subspaces for short, of $\mathbb{R}^2$. Since these are subspaces consisting of eigenvectors, they are called eigenspaces.
b. Use the definition of dimension in the previous section to find $\dim E_1$ and $\dim E_2$. Note that $\dim E_1 + \dim E_2 = \dim \mathbb{R}^2$. This is not a coincidence.

Exercise 12.4.3

a. Verify that $P^{-1}AP$ is indeed equal to $\begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$, as indicated in Example 12.4.2
b. Choose $P^{(1)} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $P^{(2)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and verify that the new value of P satisfies $P^{-1}AP = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$.
c. Take two different (from the previous part) linearly independent eigenvectors of the matrix A of Example 12.4.2 and verify that $P^{-1}AP$ is a diagonal matrix.

Answer

Part c: You should obtain $\begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$ or $\begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$, depending on how you order the eigenvalues.

Exercise 12.4.4

a. Let A be the matrix in Example 12.4.3 and $P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \end{pmatrix}$. Without doing any actual matrix multiplications, determine the value of $P^{-1}AP$
b. If you choose the columns of P in the reverse order, what is $P^{-1}AP$?

Exercise 12.4.5

Diagonalize the following, if possible:

a. $\begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}$

b. $\begin{pmatrix} -2 & 1 \\ -7 & 6 \end{pmatrix}$

c. $\begin{pmatrix} 3 & 0 \\ 0 & 4 \end{pmatrix}$

d. $\begin{pmatrix} 1 & -1 & 4 \\ 3 & 2 & -1 \\ 2 & 1 & -1 \end{pmatrix}$

e. $\begin{pmatrix} 6 & 0 & 0 \\ 0 & 7 & -4 \\ 9 & 1 & 3 \end{pmatrix}$

f. $\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$

Answer

a. If $P = \begin{pmatrix} 2 & 1 \\ 3 & -1 \end{pmatrix}$, then $P^{-1}AP = \begin{pmatrix} 4 & 0 \\ 0 & -1 \end{pmatrix}$.

b. If $P = \begin{pmatrix} 1 & 1 \\ 7 & 1 \end{pmatrix}$, then $P^{-1}AP = \begin{pmatrix} 5 & 0 \\ 0 & -1 \end{pmatrix}$.

c. If $P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, then $P^{-1}AP = \begin{pmatrix} 3 & 0 \\ 0 & 4 \end{pmatrix}$.

d. If $P = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 4 & 2 \\ -1 & 1 & 1 \end{pmatrix}$, then $P^{-1}AP = \begin{pmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

e. A is not diagonalizable. Five is a double root of the characteristic equation, but has an eigenspace with dimension only 1.

f. If $P = \begin{pmatrix} 1 & 1 & 1 \\ -2 & 0 & 1 \\ 1 & -1 & 1 \end{pmatrix}$, then $P^{-1}AP = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

Exercise 12.4.6

Diagonalize the following, if possible:

a. $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$

b. $\begin{pmatrix} 2 & 1 \\ 4 & 2 \end{pmatrix}$

c. $\begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix}$

d. $\begin{pmatrix} 1 & 3 & 6 \\ -3 & -5 & -6 \\ 3 & 3 & 6 \end{pmatrix}$

e. $\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$

f. $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$

Exercise 12.4.7

Let A and P be as in Example 12.4.3. Show that the columns of the matrix AP can be found by computing $AP^{(1)}$, $AP^{(2)}, \ldots, AP^{(n)}$.

Answer

This is a direct application of the definition of matrix multiplication. Let $A_{(i)}$ be the ith row of A, and let $P^{(j)}$ be the jth column of P. Then the jth column of the product AP is

$$\begin{pmatrix} A_{(1)}P^{(j)} \\ A_{(2)}P^{(j)} \\ \vdots \\ A_{(n)}P^{(j)} \end{pmatrix}$$

Hence, $(AP)^{(j)} = A\left(P^{(j)}\right)$ for $j = 1, 2, \ldots, n$. Thus, each column of AP depends on A and the jth column of P.

Exercise 12.4.8

Prove that if P is an $n \times n$ matrix and D is a diagonal matrix with diagonal entries $d_1, d_2, \ldots, d_n$, then PD is the matrix obtained from P, by multiplying column i of P by d_i, $i = 1, 2, \ldots, n$.

This page titled 12.4: The Diagonalization Process is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

12.5: Some Applications

A large and varied number of applications involve computations of powers of matrices. These applications can be found in science, the social sciences, economics, engineering, and, many other areas where mathematics is used. We will consider a few diverse examples the mathematics behind these applications here.

12.5.1: Diagonalization

We begin by developing a helpful technique for computing A^m, $m > 1$. If A can be diagonalized, then there is a matrix P such that $P^{-1}AP = D$, where D is a diagonal matrix and

$$A^m = PD^mP^{-1} \text{ for all } m \geq 1 \tag{12.5.1}$$

The proof of this identity was an exercise in Section 5.4. The condition that D be a diagonal matrix is not necessary but when it is, the calculation on the right side is particularly easy to perform. Although the formal proof is done by induction, the reason why it is true is easily seen by writing out an example such as $m = 3$:

$$\begin{aligned} A^3 &= \left(PDP^{-1}\right)^3 \\ &= \left(PDP^{-1}\right)\left(PDP^{-1}\right)\left(PDP^{-1}\right) \\ &= PD\left(P^{-1}P\right)D\left(P^{-1}P\right)DP^{-1} \quad \text{by associativity of matrix multiplication} \\ &= PDIDIDP^{-1} \\ &= PDDDP^{-1} \\ &= PD^3P^{-1} \end{aligned}$$

Example 12.5.1: Application to Recursion: Matrix Computation of the Fibonacci Sequence

Consider the computation of terms of the Fibonnaci Sequence. Recall that $F_0 = 1, F_1 = 1$ and $F_k = F_{k-1} + F_{k-2}$ for $k \geq 2$.

In order to formulate the calculation in matrix form, we introduced the "dummy equation" $F_{k-1} = F_{k-1}$ so that now we have two equations

$$\begin{aligned} F_k &= F_{k-1} + F_{k-2} \\ F_{k-1} &= F_{k-1} \end{aligned}$$

If $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, these two equations can be expressed in matrix form as

$$\begin{aligned} \begin{pmatrix} F_k \\ F_{k-1} \end{pmatrix} &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} F_{k-1} \\ F_{k-2} \end{pmatrix} \quad \text{if } k \geq 2 \\ &= A\begin{pmatrix} F_{k-1} \\ F_{k-2} \end{pmatrix} \\ &= A^2\begin{pmatrix} F_{k-2} \\ F_{k-3} \end{pmatrix} \quad \text{if } k \geq 3 \\ &\quad etc. \end{aligned}$$

We can use induction to prove that if $k \geq 2$,

$$\begin{pmatrix} F_k \\ F_{k-1} \end{pmatrix} = A^{k-1}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Next, by diagonalizing A and using the fact that $A^m = PD^mP^{-1}$. we can show that

$$F_k = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right)$$

Some comments on this example:

1. An equation of the form $F_k = aF_{k-1} + bF_{k-2}$, where a and b are given constants, is referred to as a linear homogeneous second-order difference equation. The conditions $F_0 = c_0$ and $F_1 = c_1$, where c_1 and c_2 are constants, are called initial conditions. Those of you who are familiar with differential equations may recognize that this language parallels what is used in differential equations. Difference (aka recurrence) equations move forward discretely; that is, in a finite number of positive steps. On the other hand, a differential equation moves continuously; that is, takes an infinite number of infinitesimal steps.
2. A recurrence relationship of the form $S_k = aS_{k-1} + b,$ where a and b are constants, is called a first-order difference equation. In order to write out the sequence, we need to know one initial condition. Equations of this type can be solved similarly to the method outlined in the example by introducing the superfluous equation $1 = 0 \cdot F_{k-1} + 1$ to obtain in matrix equation:

$$\begin{pmatrix} F_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} S_{k-1} \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} F_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}^k \begin{pmatrix} F_0 \\ 1 \end{pmatrix}$$

12.5.2: Counting

In the next example, we apply the following theorem, which can be proven by induction.

Theorem 12.5.1: Path Counting Theorem

If A is the adjacency matrix of a graph with vertices $\{v_1, v_2, \ldots, v_n\}$, then the entry $\left(A^k\right)_{ij}$ is the number of paths of length k from node v_i to node v_j.

Example 12.5.2: Counting Paths with Diagonalization

Consider the graph in Figure 12.5.1.

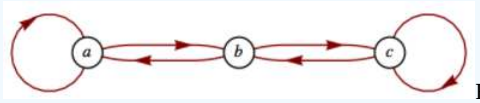


Figure 12.5.1: Counting Numbers of Paths

As we saw in Section 6.4, the adjacency matrix of this graph is $A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$.

Recall that A^k is the adjacency matrix of the relation r^k, where r is the relation $\{(a,a),(a,b),(b,a),(b,c),(c,b),(c,c)\}$ of the above graph. Also recall that in computing A^k, we used Boolean arithmetic. What happens if we use "regular" arithmetic?

If we square A we get $A^2 = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

How can we interpret this? We note that $A_{33} = 2$ and that there are two paths of length two from c (the third node) to c. Also, $A_{13} = 1,$ and there is one path of length 2 from a to c. The reader should verify these claims by examining the graph.

How do we compute A^k for possibly large values of k? From the discussion at the beginning of this section, we know that $A^k = PD^kP^{-1}$ if A is diagonalizable. We leave to the reader to show that $\lambda = 1, 2,$ and -1 are eigenvalues of A with eigenvectors

$$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

Then

$$A^k = P \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2^k & 0 \\ 0 & 0 & (-1)^k \end{pmatrix} P^{-1}$$

where $P = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -2 \\ -1 & 1 & 1 \end{pmatrix}$ and $P^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \end{pmatrix}$

See Exercise 12.5.5 of this section for the completion of this example.

12.5.3: Matrix Calculus

Example 12.5.3: Matrix Calculus - Exponentials

Those who have studied calculus recall that the Maclaurin series is a useful way of expressing many common functions. For example, $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$. Indeed, calculators and computers use these series for calculations. Given a polynomial $f(x)$, we defined the matrix-polynomial $f(A)$ for square matrices in Chapter 5. Hence, we are in a position to describe e^A for an $n \times n$ matrix A as a limit of polynomials, the partial sums of the series. Formally, we write

$$e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

Again we encounter the need to compute high powers of a matrix. Let A be an $n \times n$ diagonalizable matrix. Then there exists an invertible $n \times n$ matrix P such that $P^{-1}AP = D$, a diagonal matrix, so that

$$\begin{aligned} e^A &= e^{PDP^{-1}} \\ &= \sum_{k=0}^{\infty} \frac{\left(PDP^{-1}\right)^k}{k!} \\ &= P\left(\sum_{k=0}^{\infty} \frac{D^k}{k!}\right) P^{-1} \end{aligned}$$

The infinite sum in the middle of this final expression can be easily evaluated if D is diagonal. All entries of powers off the diagonal are zero and the ith entry of the diagonal is

$$\left(\sum_{k=0}^{\infty} \frac{D^k}{k!}\right)_{ii} = \sum_{k=0}^{\infty} \frac{{D_{ii}}^k}{k!} = e^{D_{ii}}$$

For example, if $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$, the first matrix we diagonalized in Section 12.3, we found that $P = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$ and $D = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$.

Therefore,

$$\begin{aligned} e^A &= \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} e & 0 \\ 0 & e^4 \end{pmatrix} \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{pmatrix} \\ &= \begin{pmatrix} \frac{2e}{3} + \frac{e^4}{3} & -\frac{e}{3} + \frac{e^4}{3} \\ -\frac{2e}{3} + \frac{2e^4}{3} & \frac{e}{3} + \frac{2e^4}{3} \end{pmatrix} \\ &\approx \begin{pmatrix} 20.0116 & 17.2933 \\ 34.5866 & 37.3049 \end{pmatrix} \end{aligned}$$

Remark 12.5.1

Many of the ideas of calculus can be developed using matrices. For example, if $A(t) = \begin{pmatrix} t^3 & 3t^2 + 8t \\ e^t & 2 \end{pmatrix}$ then $\frac{dA(t)}{dt} = \begin{pmatrix} 3t^2 & 6t + 8 \\ e^t & 0 \end{pmatrix}$

Many of the basic formulas in calculus are true in matrix calculus. For example,

$$\frac{d(A(t)+B(t))}{dt} = \frac{dA(t)}{dt} + \frac{dB(t)}{dt}$$

and if A is a constant matrix,

$$\frac{de^{At}}{dt} = Ae^{At}$$

Matrix calculus can be used to solve systems of differential equations in a similar manner to the procedure used in ordinary differential equations.

12.5.4: SageMath Note - Matrix Exponential

Sage's matrix exponential method is `exp` .

```
A=Matrix(QQ,[[2,1],[2,3]])
A.exp()
```

12.5.5: Exercises

Exercise 12.5.1

a. Write out all the details of Example 12.5.1 to show that the formula for F_k given in the text is correct.

b. Use induction to prove the assertion made in the example that $\begin{pmatrix} F_k \\ F_{k-1} \end{pmatrix} = A^{k-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Exercise 12.5.2

a. Do Example 8.3.7 using the method outlined in Example 12.5.1. Note that the terminology characteristic equation, characteristic polynomial, and so on, introduced in Chapter 8, comes from the language of matrix algebra,

b. What is the significance of Algorithm 8.3.1, part c, with respect to this section?

Exercise 12.5.3

Solve $S(k) = 5S(k-1) + 4$, with $S(0) = 0$, using the method of this section.

Exercise 12.5.4

How many paths are there of length 6 from vertex 1 to vertex 3 in Figure 12.5.2? How many paths from vertex 2 to vertex 2 of length 6 are there?

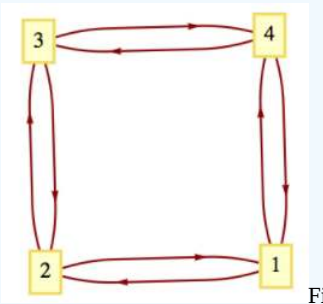


Figure 12.5.2: Graph for Exercise 12.5.4

Hint

The characteristic polynomial of the adjacency matrix is $\lambda^4 - 4\lambda^2$.

Exercise 12.5.5

Regarding Example 12.5.2

a. Use matrices to determine the number of paths of length 1 that exist from vertex a to each of the vertices in the given graph. Verify using the graph. Do the same for vertices b and c.
b. Verify all the details provided in the example.
c. Use matrices to determine the number of paths of length 4 there between each pair of nodes in the graph. Verify your results using the graph.

Answer

a. Since $A = A^1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, there are 0 paths of length 1 from: node c to node a, node b to node b, and node a to node c; and there is 1 path of length 1 for every other pair of nodes.

b. The characteristic polynomial is $|A - cI| = \begin{vmatrix} 1-c & 1 & 0 \\ 1 & -c & 1 \\ 0 & 1 & 1-c \end{vmatrix} = -c^3 + 2c^2 + c - 2$

Solving the characteristic equation $-c^3 + 2c^2 + c - 2 = 0$ we find solutions 1, 2, and -1.

If $c = 1$, we find the associated eigenvector by finding a nonzero solution to $\begin{pmatrix} 0 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ One of these, which will be the first column of P, is $\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$

If $c = 2$, the system $\begin{pmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ yields eigenvectors, including $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, which will be the second column of P.

If $c = -1$, then the system determining the eigenvectors is $\begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ and we can select $\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$, although any nonzero multiple of this vector could be the third column of P.

c. Assembling the results of part (b) we have $P = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -2 \\ -1 & 1 & 1 \end{pmatrix}$.

$$A^4 = P \begin{pmatrix} 1^4 & 0 & 0 \\ 0 & 2^4 & 0 \\ 0 & 0 & (-1)^4 \end{pmatrix} P^{-1} = P \begin{pmatrix} 1 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 1 \end{pmatrix} P^{-1}$$
$$= \begin{pmatrix} 1 & 16 & 1 \\ 0 & 16 & -2 \\ -1 & 16 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \end{pmatrix}$$
$$= \begin{pmatrix} 6 & 5 & 5 \\ 5 & 6 & 5 \\ 5 & 5 & 6 \end{pmatrix}$$

Hence there are five different paths of length 4 between distinct vertices, and six different paths that start and end at the same vertex. The reader can verify these facts from Figure 12.5.1

Exercise 12.5.6

Let $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$

a. Find e^A

b. Recall that $\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k x^k}{(2k+1)!}$ and compute $\sin A$.

c. Formulate a reasonable definition of the natural logarithm of a matrix and compute $\ln A$.

Exercise 12.5.7

We noted in Chapter 5 that since matrix algebra is not commutative under multiplication, certain difficulties arise. Let $A = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$.

a. Compute e^A, e^B, and e^{A+B}. Compare $e^A e^B$, $e^B e^A$ and e^{A+B} .

b. Show that if $\mathbf{0}$ is the 2×2 zero matrix, then $e^{\mathbf{0}} = I$.

c. Prove that if A and B are two matrices that do commute, then $e^{A+B} = e^A e^B$, thereby proving that e^A and e^B commute.

d. Prove that for any matrix A, $\left(e^A\right)^{-1} = e^{-A}$.

Answer

a. $e^A = \begin{pmatrix} e & e \\ 0 & 0 \end{pmatrix}$, $e^B = \begin{pmatrix} 0 & 0 \\ 0 & e^2 \end{pmatrix}$, and $e^{A+B} = \begin{pmatrix} e & e^2 - e \\ 0 & e^2 \end{pmatrix}$

b. Let $\mathbf{0}$ be the zero matrix, $e^{\mathbf{0}} = I + \mathbf{0} + \frac{\mathbf{0}^2}{2} + \frac{\mathbf{0}^3}{6} + \ldots = I$.

c. Assume that A and B commute. We will examine the first few terms in the product $e^A e^B$. The pattern that is established does continue in general. In what follows, it is important that $AB = BA$. For example, in the last step, $(A+B)^2$ expands to $A^2 + AB + BA + B^2$, not $A^2 + 2AB + B^2$, if we can't assume commutativity.

$$\begin{aligned} e^A e^B &= \left(\sum_{k=0}^{\infty} \frac{A^k}{k!}\right)\left(\sum_{k=0}^{\infty} \frac{B^k}{k!}\right) \\ &= \left(I + A + \frac{A^2}{2} + \frac{A^3}{6} + \cdots\right)\left(I + B + \frac{B^2}{2} + \frac{B^3}{6} + \cdots\right) \\ &= I + A + B + \frac{A^2}{2} + AB + \frac{B^2}{2} + \frac{A^3}{6} + \frac{A^2 B}{2} + \frac{AB^2}{2} + \frac{B^3}{6} + \cdots \\ &= I + (A+B) + \frac{1}{2}\left(A^2 + 2AB + B^2\right) + \frac{1}{6}\left(A^3 + 3A^2 B + 3AB^2 + B^3\right) + \cdots \\ &= I + (A+B) + \frac{1}{2}(A+B)^2 + \frac{1}{6}(A+B)^3 + \cdots \\ &= e^{A+B} \end{aligned}$$

d. Since A and $-A$ commute, we can apply part d;

$$e^A e^{-A} = e^{A+(-A)} = e^{\mathbf{0}} = I$$

Exercise 12.5.8

Another observation for adjacency matrices: For the matrix in Example 12.5.2 note that the sum of the elements in the row corresponding to the node a (that is, the first row) gives the outdegree of a. Similarly, the sum of the elements in any given column gives the indegree of the node corresponding to that column.

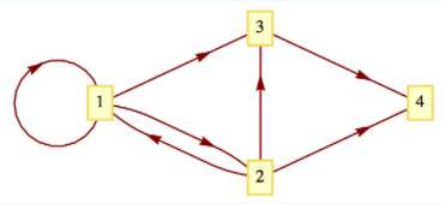


Figure 12.5.3: Graph for Exercise 12.5.8

a. Using the matrix A of Example 12.5.2, find the outdegree and the indegree of each node. Verify by the graph.
b. Repeat part (a) for the directed graphs in Figure 12.5.3.

This page titled 12.5: Some Applications is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

12.6: Linear Equations over the Integers Mod 2

12.6.1: reduction mod 2

The methods we have studied for solving systems of equations up to this point can be applied to systems in which all arithmetic is done over other algebraic systems, including the integers modulo 2. The mod 2 case will become particularly useful in our later study of coding theory.

When solving systems of equations with mod 2 arithmetic, the elementary row operations are still fundamental. However, since there is only one nonzero element, 1, you never need to multiply a row by a nonzero constant. One other big difference is that the number of possible solutions is always finite. If you have m linear equations in n unknowns, each unknown can only take on one of two values, 0 or 1. Therefore there are only 2^n possible n-tuples to from which to draw a solution set. Assuming $m \leq n$, you typically (but not always) will have m basic variables after row-reduction and $n-m$ free variable. If this is the case, and any solution exists, there will be 2^{n-m} different solutions.

Let's look at an example, which is coverted to matrix form immediately.

$$\begin{aligned} x_1 + x_2 + x_4 &= 1 \\ x_1 + x_3 + x_5 &= 0 \\ x_2 + x_3 + x_6 &= 0 \end{aligned}$$

The augmented matrix of the system is

$$\left(\begin{array}{cccccc|c} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right)$$

The steps in row-reducing this matrix follow. Entries on which we "pivot" are displayed in bold face to more easily identify the basic variables.

$$\left(\begin{array}{cccccc|c} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right) \xrightarrow{\text{add } R_1 \text{ to } R_2} \left(\begin{array}{cccccc|c} \mathbf{1} & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right)$$

$$\xrightarrow{\text{add } R_2 \text{ to } R_1} \left(\begin{array}{cccccc|c} \mathbf{1} & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right)$$

$$\xrightarrow{\text{add } R_2 \text{ to } R_3} \left(\begin{array}{cccccc|c} \mathbf{1} & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

Notice that at this point, we cannot pivot on the third row, third column since that entry is zero. Therefore we move over to the next column, making the x_4 basic.

$$\xrightarrow{\text{add } R_3 \text{ to } R_2} \left(\begin{array}{cccccc|c} \mathbf{1} & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 1 & 1 & 1 \end{array}\right)$$

This completes the row reduction and we can now identify the solution set. Keep in mind that since addition is subtraction, terms can be moved to either side of an equals sign without any change in sign. The basic variables are x_1, x_2, and x_4, while the other three variables are free. The general solution of the system is

$$
\begin{aligned}
x_1 &= x_3 + x_5 \\
x_2 &= x_3 + x_6 \\
x_3 &= x_3 \\
x_4 &= 1 + x_5 + x_6 \\
x_5 &= x_5 \\
x_6 &= x_6
\end{aligned}
$$

With three free variables, there are $2^3 = 8$ solutions to this system. For example, one of them is obtained by setting $x_3 = 1$, $x_5 = 1$, and $x_6 = 0$, which produces $(x_1, x_2, x_3, x_4, x_5, x_6) = (0, 1, 1, 0, 1, 0)$.

We can check our row reduction with SageMath:

```
H=Matrix(Integers(2),[[1,1,0,1,0,0,1],
   [1,0,1,0,1,0,0],[0,1,1,0,0,1,0]])
H.echelon_form()
```

12.6.2: Exercises

In all of the exercises that follow, the systems of equations are over $\mathbb{Z}_2$, and so mod 2 arithmetic should be used in solving them.

Exercise 12.6.1

Solve the following systems, describing the solution sets completely:

a.
$$
\begin{aligned}
x_1 + x_2 &= 0 \\
x_1 + x_3 &= 0 \\
x_1 + x_2 &= 0
\end{aligned}
$$

b.
$$
\begin{aligned}
x_2 + x_3 &= 0 \\
x_3 + x_4 &= 1 \\
x_1 + x_2 + x_3 &= 1
\end{aligned}
$$

Answer

a. $\{(0,0,0), (1,1,1)\}$
b. $\{(1,1,1,0)\}$

Exercise 12.6.2

This exercise provides an example in which the number of basic variables is less than the number of equations. The only difference between the two systems below is the right hand sides. You can start with an augmented matrix having two right side columns and do row reduction for both systems at the same time.

a.
$$
\begin{aligned}
x_1 + x_2 + x_4 &= 1 \\
x_1 + x_3 + x_4 &= 0 \\
x_2 + x_3 &= 1
\end{aligned}
$$

b.
$$
\begin{aligned}
x_1 + x_2 + x_4 &= 1 \\
x_1 + x_3 + x_4 &= 0 \\
x_2 + x_3 &= 0
\end{aligned}
$$

Answer

As suggested here is the augmented matrix with both right sides, and its row reduction:

$$
\left(\begin{array}{cccc|cc}
1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0
\end{array}\right) \longrightarrow \left(\begin{array}{cccc|cc}
\mathbf{1} & 0 & 1 & 1 & 0 & 0 \\
0 & \mathbf{1} & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array}\right)
$$

There are only two basic variables here because the left side of the last equation is the sum of the left sides of the first two equations.

a. Ignoring the last column of both matrices, we see that the last equation of the first system reduces to $0 = 0$, which is always true, and the first two equations yield two free variables, x_3 and x_4. The general solution is the set of quadruples $\{(x_3 +_2 x_4, x_3 +_2 1, x_3, x_4) \mid x_3, x_4 \in \mathbb{Z}_2\}$. The cardinality of the solution set is 4.
b. If we replace the fifth column with the sixth one, the last row indicates that $0 = 1$, which means that the solution set is empty.

Exercise 12.6.3

This exercise motivates the concept of a coset in Chapter 15.

a. Solve the following system and prove that the solution set is a linear combination of vectors in $\mathbb{Z}_2^5$ and also a subgroup of the group $\mathbb{Z}_2^5$ under coordinatewise mod 2 addition.

$$\begin{array}{rcrcrcrcl} x_1 & + & x_2 & & & & + & x_5 & = 0 \\ x_1 & & & + & x_3 & & + & x_5 & = 0 \\ x_1 & & & + & x_3 & + & x_4 & & = 0 \\ & & x_2 & + & x_3 & + & x_4 & & = 0 \end{array}$$

b. Describe the solution set to the following system as it relates to the solution set to the system in the previous part of this exercise.

$$\begin{array}{rcrcrcrcl} x_1 & + & x_2 & & & & + & x_5 & = 1 \\ x_1 & & & + & x_3 & & + & x_5 & = 0 \\ x_1 & & & + & x_3 & + & x_4 & & = 1 \\ & & x_2 & + & x_3 & + & x_4 & & = 0 \end{array}$$

Answer

a. Row reduction produces a solution with one free variable, x_3.

$$\begin{aligned} (x_1,\, x_2,\, x_3,\, x_4,\, x_5) &= (x_3,\, x_3,\, x_3,\, 0,\, 0) \\ &= x_3(1,\, 1,\, 1,\, 0,\, 0) \end{aligned}$$

The solution set has only two elements. It is $\{(0,\, 0,\, 0,\, 0,\, 0),\, (1,\, 1,\, 1,\, 0,\, 0)\}$. Since $\mathbb{Z}_2^5$ is a finite group, the solution set is a subgroup because it is closed with respect to coordinate-wise mod 2 addition.
b. The row-reduced augmented matrix of coefficients provides the solution

$$\begin{aligned} (x_1,\, x_2,\, x_3,\, x_4,\, x_5) &= (x_3,\, 1 + x_3,\, x_3,\, 1,\, 0) \\ &= (0,\, 1,\, 0,\, 1,\, 0) + x_3(1,\, 1,\, 1,\, 0,\, 0) \end{aligned}$$

Therefore, the solution to this system is a shift of the solution set to the homogeneous system by the vector $(0,\, 1,\, 0,\, 1,\, 0)$ which is $\{(0,\, 1,\, 0,\, 1,\, 0),\, (1,\, 0,\, 1,\, 1,\, 0)\}$

This page titled 12.6: Linear Equations over the Integers Mod 2 is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

13: Boolean Algebra



Figure 13.1: George Boole, 1815 - 1864

George Boole

George Boole *wasn't idle a lot.*
He churned out ideas on the spot,
Making marvellous use of
Inclusive/exclusive
Expressions like AND, OR, and NOT

Andrew Robinson, The Omnificent English Dictionary in Limerick Form

In this chapter we will develop a type of algebraic system, Boolean algebras, that is particularly important to computer scientists, as it is the mathematical foundation of computer design, or switching theory. The similarities of Boolean algebras and the algebra of sets and logic will be discussed, and we will discover properties of finite Boolean algebras.

In order to achieve these goals, we will recall the basic ideas of posets introduced in Chapter 6 and develop the concept of a lattice. The reader should view the development of the topics of this chapter as another example of an algebraic system. Hence, we expect to define first the elements in the system, next the operations on the elements, and then the common properties of the operations in the system.

13.1: Posets Revisited
13.2: Lattices
13.3: Boolean Algebras
13.4: Atoms of a Boolean Algebra
13.5: Finite Boolean Algebras as n-tuples of 0's and 1's
13.6: Boolean Expressions
13.7: A Brief Introduction to Switching Theory and Logic Design

This page titled 13: Boolean Algebra is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.1: Posets Revisited

We recall the definition a partially ordering:

Definition 13.1.1: Partial Ordering

Let $\preceq$ be a relation on a set L. We say that $\preceq$ is a partial ordering on L if it is reflexive, antisymmetric, and transitive. That is:

1. $\preceq$ is reflexive: $a \preceq a \quad \forall a \in L$
2. $\preceq$ is antisymmetric: $a \preceq b$ and $a \neq b \Rightarrow b \npreceq a \quad \forall a, b \in L$
3. $\preceq$ is transitive: $a \preceq b$ and $b \preceq c \Rightarrow a \preceq c \quad \forall a, b, c \in L$

The set together with the relation $(L, \preceq)$ is called a poset.

Example 13.1.1: Some Posets

We recall a few examples of posets:

a. $(\mathbb{R}, \leq)$ is a poset. Notice that our generic symbol for the partial ordering, $\preceq$, is selected to remind us that a partial ordering is similar to "less than or equal to."
b. Let $A = \{a, b\}$. Then $(\mathcal{P}(A), \subseteq)$ is a poset.
c. Let $L = \{1, 2, 3, 6\}$. Then $(L, |)$ is a poset.

The posets we will concentrate on in this chapter will be those which have upper and lower bounds in relation to any pair of elements. Next, we define this concept precisely.

Definition 13.1.2: Lower Bound, Upper Bound

Let $(L, \preceq)$ be a poset, and $a, b \in L$. Then $c \in L$ is a lower bound of a and b if $c \preceq a$ and $c \preceq b$. Also, $d \in L$ is an upper bound of a and b if $a \preceq d$ and $b \preceq d$.

In most of the posets that will interest us, every pair of elements have both upper and lower bounds, though there are posets for which this is not true.

Definition 13.1.3: Greatest Lower Bound

Let $(L, \preceq)$ be a poset. If $a, b \in L$, then $\ell \in L$ is a greatest lower bound of a and b if and only if

- $\ell \preceq a$
- $\ell \preceq b$
- If $\ell' \in L$ such that $\ell' \preceq a$ and $\ell' \preceq b$, then $\ell' \preceq \ell$.

The last condition in the definition of Greatest Lower Bound says that if ℓ' is also a lower bound, then ℓ is "greater" in relation to $\preceq$ than ℓ'. The definition of a least upper bound is a mirror image of a greatest lower bound:

Definition 13.1.4: Least Upper Bound

Let $(L, \preceq)$ be a poset. If $a, b \in L$, then $u \in L$ is a least upper bound of a and b if and only if

- $a \preceq u$
- $b \preceq u$
- If $u' \in L$ such that if $a \preceq u'$ and $b \preceq u'$, then $u \preceq u'$.

Notice that the two definitions above refer to "...a greatest lower bound" and "a least upper bound." Any time you define an object like these you need to have an open mind as to whether more than one such object can exist. In fact, we now can prove that there can't be two greatest lower bounds or two least upper bounds.

Theorem 13.1.1: Uniqueness of Least Upper and Greatest Lower Bounds

Let $(L, \preceq)$ be a poset, and $a, b \in L$. If a greatest lower bound of a and b exists, then it is unique. The same is true of a least upper bound, if it exists.

Proof

Let ℓ and ℓ' be greatest lower bounds of a and b. We will prove that $\ell = \ell'$.

1. ℓ a greatest lower bound of a and $b \Rightarrow \ell$ is a lower bound of a and b.
2. ℓ' a greatest lower bound of a and b and ℓ a lower bound of a and $b \Rightarrow \ell \preceq \ell'$, by the definition of greatest lower bound.
3. ℓ' a greatest lower bound of a and $b \Rightarrow \ell'$ is a lower bound of a and b.
4. ℓ a greatest lower bound of a and b and ℓ' a lower bound of a and $b. \Rightarrow \ell' \preceq \ell$ by the definition of greatest lower bound.
5. $\ell \preceq \ell'$ and $\ell' \preceq \ell \Rightarrow \ell = \ell'$ by the antisymmetry property of a partial ordering.

The proof of the second statement in the theorem is almost identical to the first and is left to the reader.

Definition 13.1.5: Greatest Element, Least Element

Let $(L, \preceq)$ be a poset. $M \in L$ is called the greatest (maximum) element of L if, for all $a \in L$, $a \preceq M$. In addition, $m \in L$ is called the least (minimum) element of L if for all $a \in L$, $m \preceq a$. The greatest and least elements, when they exist, are frequently denoted by **1** and **0** respectively.

Example 13.1.2: Bounds on the Divisors of 105

Consider the partial ordering "divides" on $L = \{1, 3, 5, 7, 15, 21, 35, 105\}$Then $(L, |)$ is a poset. To determine the least upper bound of 3 and 7, we look for all $u \in L$, such that $3|u$ and $7|u$. Certainly, both $u = 21$ and $u = 105$ satisfy these conditions and no other element of L does. Next, since $21|105$,21 is the least upper bound of 3 and 7. Similarly, the least upper bound of 3 and 5 is 15. The greatest element of L is 105 since $a|105$ for all $a \in L$. To find the greatest lower bound of 15 and 35, we first consider all elements g of L such that $g \mid 15$. They are 1, 3, 5, and 15. The elements for which $g \mid 35$ are 1, 5, 7, and 35. From these two lists, we see that $\ell = 5$ and $\ell = 1$ satisfy the required conditions. But since $1|5$, the greatest lower bound is 5. The least element of L is 1 since $1|a$ for all $a \in L$.

Definition 13.1.6: The Set of Divisors of an Integer

For any positive integer n, the divisors of n is the set of integers that divide evenly into n. We denote this set D_n.

For example, the set L of Example 13.1.2is D_{105}.

Example 13.1.3: The Power Set of a Three Element Set

Consider the poset $(\mathcal{P}(A), \subseteq)$, where $A = \{1, 2, 3\}$. The greatest lower bound of $\{1, 2\}$ and $\{1, 3\}$ is $\ell = \{1\}$. For any other element ℓ' which is a subset of $\{a, b\}$ and $\{a, c\}$ (there is only one; what is it?), $\ell' \subseteq \ell$. The least element of $\mathcal{P}(A)$ is $\emptyset$ and the greatest element is $A = \{a, b, c\}$. The Hasse diagram of this poset is shown in Figure 13.1.1.

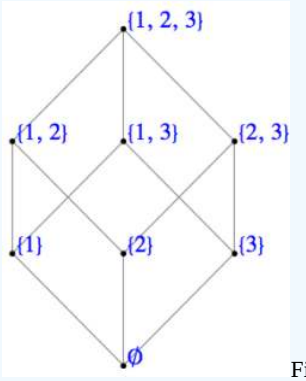


Figure 13.1.1: Power Set of $\{1,\ 2,\ 3\}$

The previous examples and definitions indicate that the least upper bound and greatest lower bound are defined in terms of the partial ordering of the given poset. It is not yet clear whether all posets have the property such every pair of elements always has both a least upper bound and greatest lower bound. Indeed, this is not the case (see Exercise 13.1.1).

13.1.1: Exercises

Exercise 13.1.1

Consider the poset $(D_{30}, |)$, where $D_{30} = \{1, 2, 3, 5, 6, 10, 15, 30\}$.

a. Find all lower bounds of 10 and 15.
b. Find the greatest lower bound of 10 and 15.
c. Find all upper bounds of 10 and 15.
d. Determine the least upper bound of 10 and 15.
e. Draw the Hasse diagram for D_{30} with respect to $|$. Compare this Hasse diagram with that of Example 13.1.3. Note that the two diagrams are structurally the same.

Answer

a. 1, 5
b. 5
c. 30
d. 30
e. See the Sage cell below with the default input displaying a Hasse diagram for D_{12}.

Exercise 13.1.2

List the elements of the sets D_8, D_{50}, and D_{1001}. For each set, draw the Hasse diagram for “divides.”

Exercise 13.1.3

Figure 13.1.2 contains Hasse diagrams of posets.

a. Determine the least upper bound and greatest lower bound of all pairs of elements when they exist. Indicate those pairs that do not have a least upper bound (or a greatest lower bound).
b. Find the least and greatest elements when they exist.

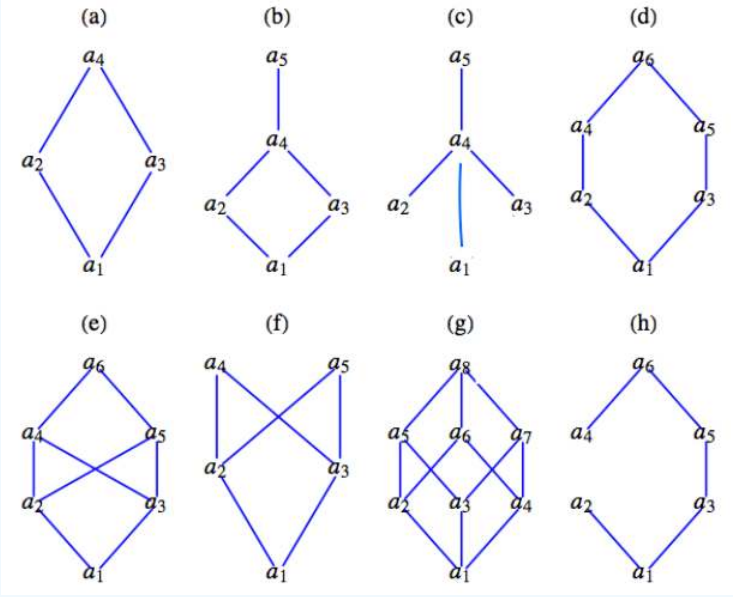


Figure 13.1.2: Figure for Exercise 13.1.3

Answer

- Solution for Hasse diagram (b):
  -

$\vee$	a_1	a_2	a_3	a_4	a_5
a_1	a_1	a_2	a_3	a_4	a_5
a_2	a_2	a_2	a_4	a_4	a_5
a_3	a_3	a_4	a_3	a_4	a_5
a_4	a_4	a_4	a_4	a_4	a_5
a_5	a_5	a_5	a_5	a_5	a_5

$\wedge$	a_1	a_2	a_3	a_4	a_5
a_1	a_1	a_1	a_1	a_1	a_1
a_2	a_1	a_2	a_1	a_2	a_2
a_3	a_1	a_1	a_3	a_3	a_3
a_4	a_1	a_2	a_3	a_4	a_4
a_5	a_1	a_2	a_3	a_4	a_5

    a_1 is the least element and a_5 is the greatest element.
- Partial solution for Hasse diagram (f):
  - $\text{lub}(a_2, a_3)$ and $\text{lub}(a_4, a_5)$ do not exist.
  - No greatest element exists, but a_1 is the least element.

Exercise 13.1.4

For the poset $(\mathbb{N}, \leq)$, what are the greatest lower bound and least upper bound of two elements a and b? Are there least and/or greatest elements?

Exercise 13.1.5

a. Prove the second part of Theorem 13.1.1, the least upper bound of two elements in a poset is unique, it one exists.
b. Prove that if a poset L has a least element, then that element is unique.

Answer

If 0 and $0'$ are distinct least elements, then

$$\left.\begin{array}{ll} 0 \leq 0' & \text{since } 0 \text{ is a least element} \\ 0' \leq 0 & \text{since } 0' \text{ is a least element} \end{array}\right\} \Rightarrow 0 = 0' \text{ by antisymmetry, a contradiction}$$

Exercise 13.1.6

We naturally order the numbers in $A_m = \{1, 2, \ldots, m\}$ with "less than or equal to," which is a partial ordering. We define an ordering, $\preceq$ on the elements of $A_m \times A_n$ by

$$(a, b) \preceq (a', b') \Leftrightarrow a \leq a' \text{ and } b \leq b'$$

a. Prove that $\preceq$ is a partial ordering on $A_m \times A_n$.
b. Draw the ordering diagrams for $\preceq$ on $A_2 \times A_2,\ A_2 \times A_3,$ and $A_3 \times A_3$.
c. In general, how does one determine the least upper bound and greatest lower bound of two elements of $A_m \times A_n,\ (a, b)$ and (a', b')?
d. Are there least and/or greatest elements in $A_m \times A_n$?

Exercise 13.1.7

Let $\mathcal{P}_0$ be the set of all subsets T of $S = \{1, 2, \ldots, 9\}$ such that the sum of the elements in T is even. (Note that the empty set $\emptyset$ will be included as an element of $\mathcal{P}_0$.) For instance, $\{2, 3, 6, 7\}$is in $\mathcal{P}_0$ because $2 + 3 + 6 + 7$ is even, but $\{1, 3, 5, 6\}$is not in $\mathcal{P}_0$ because $1 + 3 + 5 + 6$ is odd. Consider the poset $(\mathcal{P}_0, \subseteq)$. Let $A = \{1, 2, 3, 6\}$ and $B = \{2, 3, 6, 7\}$ be elements of $\mathcal{P}_0$.

a. Explain why $A \cap B$ is not element of the poset.
b. Use the definitions of the italicized terms and the given partial ordering to complete the following statements:
   i. $R \in \mathcal{P}_0$ is an *upper bound* of A and B if \rule{3cm}{0.01cm}
   ii. $R \in \mathcal{P}_0$ is the *least element* of $\mathcal{P}_0$ if \rule{3cm}{0.01cm}
c. Find three different upper bounds of A and B.
d. Find the least upper bound of A and B. If it doesn't exist, explain why not.

Answer

a. The sum of elements in $A \cap B = \{2,\ 3,\ 6\}$ is odd and disqualifies the set from being an element of the poset.
b. Use the definitions of the italicized terms and the given partial ordering to complete the following statements:
   i. $\ldots A \subseteq R$ and $B \subseteq R$
   ii. $\ldots$ for all $A \in \mathcal{P}_0$, $R \subseteq A$
c. Any set that contains the union of $A \cup B = \{1,\ 2,\ 3,\ 6,\ 7\}$but also contains 3 or 5, but not both will be an upper bound. You can create several by including on not including 4 or 8.
d. The least upper bound doesn't exist. Notice that the union of A and B isn't in $\mathcal{P}_0$. One of the two sets $\{1,\ 2,\ 3,\ 5,\ 6,\ 7\}$ and $\{1,\ 2,\ 3,\ 6,\ 7,\ 9\}$is contained within every upper bound of A and B but neither is contained within the other.

This page titled 13.1: Posets Revisited is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.2: Lattices

In this section, we restrict our discussion to lattices, those posets for which every pair of elements has both a greatest lower bound and least upper bound. We first introduce some notation.

Definition 13.2.1: Join, Meet

Let $(L, \preceq)$ be a poset, and $a, b \in L$. We define:

- $a \vee b$, read "a join b", as the least upper bound of a and b, if it exists. and
- $a \wedge b$, read "a meet b", as the greatest lower bound of a and b, if it exists.

Since the join and meet produce a unique result in all cases where they exist, by Theorem 13.1.1, we can consider them as binary operations on a set if they always exist. Thus the following definition:

Definition 13.2.2: Lattice

A lattice is a poset $(L, \preceq)$ for which every pair of elements has a greatest lower bound and least upper bound. Since a lattice L is an algebraic system with binary operations $\vee$ and $\wedge$, it is denoted by $[L; \vee, \wedge]$. If we want to make it clear what partial ordering the lattice is based on, we say it is a lattice under $\preceq$.

Example 13.2.1: The Power Set of a Three Element Set

Consider the poset $(\mathcal{P}(A), \subseteq)$ we examined in Example 13.1.3. It isn't too surprising that every pair of sets had a greatest lower bound and least upper bound. Thus, we have a lattice in this case; and $A \vee B = A \cup B$ and $A \wedge B = A \cap B$. The reader is encouraged to write out the operation tables $[\mathcal{P}(A); \cup, \cap]$.

Our first concrete lattice can be generalized to the case of any set A, producing the lattice $[\mathcal{P}(A); \vee, \wedge]$, where the join operation is the set operation of union and the meet operation is the operation intersection; that is, $\vee = \cup$ and $\wedge = \cap$.

It can be shown (see the exercises) that the commutative laws, associative laws, idempotent laws, and absorption laws are all true for any lattice. A concrete example of this is clearly $[\mathcal{P}(A); \cup, \cap]$, since these laws hold in the algebra of sets. This lattice also has distributive property in that join is distributive over meet and meet is distributive over join. However, this is not always the case for lattices in general.

Definition 13.2.3: Distributive Lattice

Let $\mathcal{L} = [L; \vee, \wedge]$ be a lattice under $\preceq$. $\mathcal{L}$ is called a distributive lattice if and only if the distributive laws hold; that is, for all $a, b, c \in L$ we have

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$
$$and$$
$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Example 13.2.2: A Nondistributive Lattice

We now give an example of a lattice where the distributive laws do not hold. Let $L = \{\mathbf{0}, a, b, c, \mathbf{1}\}$. We define the partial ordering $\preceq$ on L by the set

$$\{(\mathbf{0}, \mathbf{0}), (\mathbf{0}, a), (\mathbf{0}, b), (\mathbf{0}, c), (\mathbf{0}, \mathbf{1}), (a, a), (a, \mathbf{1}), (b, b), (b, \mathbf{1}), (c, c), (c, \mathbf{1}), (\mathbf{1}, \mathbf{1})\}$$

The operation tables for $\vee$ and $\wedge$ on L are:

$\vee$	$\mathbf{0}$	a	b	c	$\mathbf{1}$
$\mathbf{0}$	$\mathbf{0}$	a	b	c	$\mathbf{1}$
a	a	a	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$
b	b	$\mathbf{1}$	b	$\mathbf{1}$	$\mathbf{1}$
c	c	$\mathbf{1}$	$\mathbf{1}$	c	$\mathbf{1}$
$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$

$\wedge$	$\mathbf{0}$	a	b	c	$\mathbf{1}$
$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
a	$\mathbf{0}$	a	$\mathbf{0}$	$\mathbf{0}$	a
b	$\mathbf{0}$	$\mathbf{0}$	b	$\mathbf{0}$	b
c	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	c	c
$\mathbf{1}$	$\mathbf{0}$	a	b	c	$\mathbf{1}$

Since every pair of elements in L has both a join and a meet, $[L; \vee, \wedge]$ is a lattice (under divides). Is this lattice distributive? We note that: $a \vee (c \wedge b) = a \vee \mathbf{0} = a$ and $(a \vee c) \wedge (a \vee b) = \mathbf{1} \wedge \mathbf{1} = \mathbf{1}$. Therefore, $a \vee (b \wedge c) \neq (a \vee b) \wedge (a \vee c)$ for some values of $a, b, c \in L$. Thus, this lattice is not distributive.

Our next observation uses the term “sublattice”, which we have not defined at this point, but we would hope that you could anticipate a definition, and we will leave it as an exercises to do so.

It can be shown that a lattice is nondistributive if and only if it contains a sublattice isomorphic to one of the lattices in Figure 13.2.1. The ordering diagram on the right of this figure, produces the ***diamond lattice***, which is precisely the one that is defined in Example 13.2.2 The lattice based on the left hand poset is called the ***pentagon lattice***.

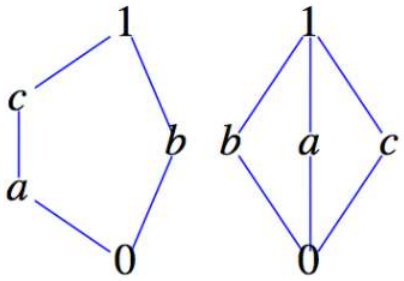


Figure 13.2.1: Nondistributive lattices, the pentagon and diamond lattices

13.2.1: 13.2.1: Exercises

Exercise 13.2.1

Let L be the set of all propositions generated by p and q. What are the meet and join operations in this lattice under implication? What are the maximum and minimum elements?

Exercise 13.2.2

Which of the posets in Exercise 13.1.3 are lattices? Which of the lattices are distributive?

Exercise 13.2.3

a. State the commutative laws, associative laws, idempotent laws, and absorption laws for lattices.
b. Prove laws you stated.

Exercise 13.2.4

Demonstrate that the pentagon lattice is nondistributive.

Exercise 13.2.5

What is a reasonable definition of the term ***sublattice***?

Answer

One reasonable definition would be this: Let $[L; \vee, \wedge]$ be a lattice and let K be a nonempty subset of L. Then K is a sublattice of L if and only if K is closed under both $\vee$ and $\wedge$

Exercise 13.2.6

Let $[L; \vee, \wedge]$ be a lattice based on a partial ordering $\preceq$. Prove that if $a, b, c \in L$,

a. $a \preceq a \vee b$.
b. $a \wedge b \preceq a$.
c. $b \preceq a$ and $c \preceq a \Rightarrow b \vee c \preceq a$.

This page titled 13.2: Lattices is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.3: Boolean Algebras

In order to define a Boolean algebra, we need the additional concept of complementation. A lattice must have both a greatest element and a least element in order for complementation to take place. The following definition will save us some words in the rest of this section.

Definition 13.3.1: Bounded Lattice

A bounded lattice is a lattice that contains both a least element and a greatest element.

We use the symbols **0** and **1** for the least and greatest elements of a bounded lattice in the remainder of this section.

Definition 13.3.2: The Complement of a Lattice Element

Let $[L; \vee, \wedge]$ be a bounded lattice. If $a \in L$, then a has a complement if there exists $b \in L$ such that

$$a \vee b = \mathbf{1}$$
and
$$a \wedge b = \mathbf{0}$$

Notice that by the commutative laws for lattices, if b complements a, then a complements b.

Definition 13.3.3: Complemented Lattice

Let $\mathcal{L} = [L; \vee, \wedge]$ be a bounded lattice. $\mathcal{L}$ is a complemented lattice if every element of L has a complement in L.

Example 13.3.1: Set Complement is a Complement

In Chapter 1, we defined the complement of a subset of any universe. This turns out to be a concrete example of the general concept we have just defined, but we will reason through why this is the case here. Let $L = \mathcal{P}(A)$, where $A = \{a, b, c\}$. Then $[L; \cup, \cap]$ is a bounded lattice with $0 = \emptyset$ and $1 = A$. To find the complement, if it exists, of $B = \{a, b\} \in L$, for example, we want D such that

$$\{a, b\} \cap D = \emptyset$$
$$and$$
$$\{a, b\} \cup D = A$$

It's not too difficult to see that $D = \{c\}$, since we need to include c to make the first condition true and can't include a or b if the second condition is to be true. Of course this is precisely how we defined A^c in Chapter 1. Since it can be shown that each element of L has a complement (see Exercise 1), $[L; \cup, \cap]$ is a complemented lattice. Note that if A is any set and $L = \mathcal{P}(A)$, then $[L; \cup, \cap]$ is a complemented lattice where the complement of $B \in L$ is $B^c = A - B$.

In Example 13.3.1, we observed that the complement of each element of L is unique. Is this always true in a complemented lattice? The answer is no. Consider the following.

Example 13.3.2: A Lattice for Which Complements are Not Unique

Let $L = \{1, 2, 3, 5, 30\}$ and consider the lattice $[L; \vee, \wedge]$ (under "divides"). The least element of L is 1 and the greatest element is 30. Let us compute the complement of the element $a = 2$. We want to determine $\bar{a}$ such that $2 \wedge \bar{a} = 1$ and $2 \vee \bar{a} = 30$. Certainly, $\bar{a} = 3$ works, but so does $\bar{a} = 5$, so the complement of $a = 2$ in this lattice is not unique. However, $[L; \vee, \wedge]$ is still a complemented lattice since each element does have at least one complement.

Definition 13.3.4: Complementation as an Operation

If a complemented lattice has the property that the complement of every element is unique, then we consider complementation to be a unary operation. The usual notation for the complement of a is $\bar{a}$.

The following theorem gives us an insight into when uniqueness of complements occurs.

Theorem 13.3.1: One Condition for Unique Complements

If $[L; \vee, \wedge]$ is a complemented, distributive lattice, then the complement of each element $a \in L$ is unique.

Proof

Let $a \in L$ and assume to the contrary that a has two complements, namely a_1 and a_2. Then by the definition of complement,

$$\begin{array}{c} a \wedge a_1 = 0 \text{ and } a \vee a_1 = 1, \\ \textit{and} \\ a \wedge a_2 = 0 \text{ and } a \vee a_2 = 1, \end{array} .$$

Then

$$\begin{aligned} a_1 &= a_1 \wedge \mathbf{1} = a_1 \wedge (a \vee a_2) \\ &= (a_1 \wedge a) \vee (a_1 \wedge a_2) \\ &= \mathbf{0} \vee (a_1 \wedge a_2) \\ &= a_1 \wedge a_2 \end{aligned}$$

On the other hand,

$$\begin{aligned} a_2 &= a_2 \wedge \mathbf{1} = a_2 \wedge (a \vee a_1) \\ &= (a_2 \wedge a) \vee (a_2 \wedge a_1) \\ &= \mathbf{0} \vee (a_2 \wedge a_1) \\ &= a_2 \wedge a_1 \\ &= a_1 \wedge a_2 \end{aligned}$$

Hence $a_1 = a_2$, which contradicts the assumption that a has two different complements.

Definition 13.3.5: Boolean Algebra

A Boolean algebra is a lattice that contains a least element and a greatest element and that is both complemented and distributive. The notation $[B; \vee, \wedge, \bar{\ }]$ is used to denote the boolean algebra with operations join, meet and complementation.

Since the complement of each element in a Boolean algebra is unique (by Theorem 13.3.1), complementation is a valid unary operation over the set under discussion, which is why we will list it together with the other two operations to emphasize that we are discussing a set together with three operations. Also, to help emphasize the distinction between lattices and lattices that are Boolean algebras, we will use the letter B as the generic symbol for the set of a Boolean algebra; that is, $[B; \vee, \wedge, \bar{\ }]$ will stand for a general Boolean algebra.

Example 13.3.3: Boolean Algebra of Sets

Let A be any set, and let $B = \mathcal{P}(A)$. Then $[B; \cup, \cap, ^c]$ is a Boolean algebra. Here, ^{c} stands for the complement of an element of B with respect to A, $A - B$.

This is a key example for us since all finite Boolean algebras and many infinite Boolean algebras look like this example for some A. In fact, a glance at the basic Boolean algebra laws in Table 13.3.1, in comparison with the set laws of Chapter 4 and the basic laws of logic of Chapter 3, indicates that all three systems behave the same; that is, they are isomorphic.

Example 13.3.4: Divisors of 30

A somewhat less standard example of a boolean algebra is derived from the lattice of divisors of 30 under the relation "divides". If you examine the ordering diagram for this lattice, you see that it is structurally the same as the boolean algebra of subsets of a three element set. Therefore, the join, meet and complementation operations act the same as union, intersection and set complementation. We might conjecture that the lattice of divisors of any integer will produce a boolean algebra, but it is only the case of certain integers. Try out a few integers to see if you can identify what is necessary to produce a boolean algebra.

Table 13.3.1: Basic Boolean Algebra Laws

Commutative Laws	$a \vee b = b \vee a$	$a \wedge b = b \wedge a$
Associative Laws	$a \vee (b \vee c) = (a \vee b) \vee c$	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$
Distributive Laws	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
Identity Laws	$a \vee 0 = 0 \vee a = a$	$a \wedge 1 = 1 \wedge a = a$
Complement Laws	$a \vee \bar{a} = 1$	$a \wedge \bar{a} = 0$
Idempotent Laws	$a \vee a = a$	$a \wedge a = a$
Null Laws	$a \vee 1 = 1$	$a \wedge 0 = 0$
Absorption Laws	$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$
DeMorgan's Laws	$\overline{a \vee b} = \bar{a} \wedge \bar{b}$	$\overline{a \wedge b} = \bar{a} \vee \bar{b}$
Involution Law	$\bar{\bar{a}} = a$	

The "pairings" of the boolean algebra laws reminds us of the principle of duality, which we state for a Boolean algebra.

Definition 13.3.6: Principle of Duality for Boolean Algebras

Let $\mathcal{B} = [B; \vee, \wedge, {}^c]$ be a Boolean algebra under $\preceq$, and let S be a true statement for $\mathcal{B}$. If S^* is obtained from S by replacing $\preceq$ with $\succeq$ (this is equivalent to turning the graph upside down), $\vee$ with $\wedge$, $\wedge$ with $\vee$, $\mathbf{0}$ with $\mathbf{1}$, and $\mathbf{1}$ with $\mathbf{0}$, then S^* is also a true statement in $\mathcal{B}$.

13.3.1: Exercises

Exercise 13.3.1

Determine the complement of each element $B \in L$ in Example 13.3.1. Is this lattice a Boolean algebra? Why?

Answer

B	Complement of B
$\emptyset$	A
$\{a\}$	$\{b, c\}$
$\{b\}$	$\{a, c\}$
$\{c\}$	$\{a, b\}$
$\{a, b\}$	$\{c\}$
$\{a, c\}$	$\{b\}$
$\{b, c\}$	$\{a\}$
A	$\emptyset$

This lattice is a Boolean algebra since it is a distributive complemented lattice.

Exercise 13.3.2

a. Determine the complement of each element of D_6 in $[D_6; \vee, \wedge]$.
b. Repeat part a using the lattice in Example 13.2.2.
c. Repeat part a using the lattice in Exercise 13.1.1.
d. Are the lattices in parts a, b, and c Boolean algebras? Why?

Exercise 13.3.3

Determine which of the lattices of Exercise 13.1.3 of Section 13.1 are Boolean algebras.

Answer

a and g.

Exercise 13.3.4

Let $A = \{a, b\}$ and $B = \mathcal{P}(A)$.

a. Prove that $[B; \cup, \cap, {}^c]$ is a Boolean algebra.
b. Write out the operation tables for the Boolean algebra.

Exercise 13.3.5

It can be shown that the following statement, S, holds for any Boolean algebra $[B; \vee, \wedge, -]$: $(a \wedge b) = a$ if and only if $a \leq b$.

a. Write the dual, S^*, of the statement S.
b. Write the statement S and its dual, S^*, in the language of sets.
c. Are the statements in part b true for all sets?
d. Write the statement S and its dual, S^*, in the language of logic.
e. Are the statements in part d true for all propositions?

Answer

a. $S^*: a \vee b = a$ if $a \geq b$
b. The dual of $S: A \cap B = A$ if $A \subseteq B$ is $S^*: A \cup B = A$ if $A \supseteq B$
c. Yes
d. The dual of $S: p \wedge q \Leftrightarrow p$ if $p \Rightarrow q$ is $S^*: p \vee q \Leftrightarrow p$ if $q \Rightarrow p$
e. Yes

Exercise 13.3.6

State the dual of:

a. $a \vee (b \wedge a) = a$.
b. $a \vee \left(\overline{(\bar{b} \vee a) \wedge b}\right) = 1$.
c. $\left(\overline{a \wedge \bar{b}}\right) \wedge b = a \vee b$.

Exercise 13.3.7

Formulate a definition for isomorphic Boolean algebras.

Answer

$[B; \wedge, \vee, -]$ is isomorphic to $[B'; \wedge, \vee, \tilde{}]$ if and only if there exists a function $T: B \to B'$ such that

a. T is a bijection;
b. $T(a \wedge b) = T(a) \wedge T(b)$ for all $a, b \in B$
c. $T(a \vee b) = T(a) \vee T(b)$ for all $a, b \in B$
d. $T(\bar{a}) = \tilde{T(a)}$ for all $a \in B$.

Exercise 13.3.8

For what positive integers, n, does the lattice $[D_n, |]$ produce a boolean algebra?

This page titled 13.3: Boolean Algebras is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.4: Atoms of a Boolean Algebra

In this section we will look more closely at something we've hinted at, which is that every finite Boolean algebra is isomorphic to an algebra of sets. We will show that every finite Boolean algebra has 2^n elements for some n with precisely n generators, called atoms.

Consider the Boolean algebra $[B; \vee, \wedge, \bar{\ }]$, whose ordering diagram is depicted in Figure 13.4.1

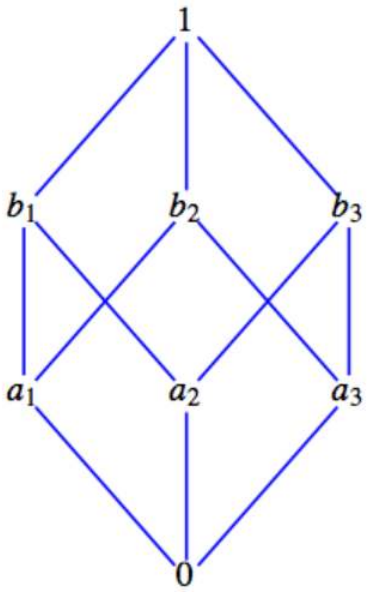


Figure 13.4.1: Illustration of the atom concept

We note that $1 = a_1 \vee a_2 \vee a_3$, $b_1 = a_1 \vee a_2$, $b_2 = a_1 \vee a_3$, and $b_3 = a_2 \vee a_3$; that is, each of the elements above level one can be described completely and uniquely in terms of the elements on level one. The a_i's have uniquely generated the non-least elements of B much like a basis in linear algebra generates the elements in a vector space. We also note that the a_i's are the immediate successors of the minimum element, 0. In any Boolean algebra, the immediate successors of the minimum element are called ***atoms***. For example, let A be any nonempty set. In the Boolean algebra $[\mathcal{P}(A); \cup, \cap, {}^c]$ (over $\subseteq$), the singleton sets are the generators, or atoms, of the algebraic structure since each element $\mathcal{P}(A)$ can be described completely and uniquely as the join, or union, of singleton sets.

Definition 13.4.1: Atom

A non-least element a in a Boolean algebra $[B; \vee, \wedge, \bar{\ }]$ is called an atom if for every $x \in B$, $x \wedge a = a$ or $x \wedge a = 0$.

The condition that $x \wedge a = a$ tells us that x is a successor of a; that is, $a \preceq x$, as depicted in Figure 13.4.2(a)

The condition $x \wedge a = 0$ is true only when x and a are "not connected." This occurs when x is another atom or if x is a successor of atoms different from a, as depicted in Figure 13.4.2(b).

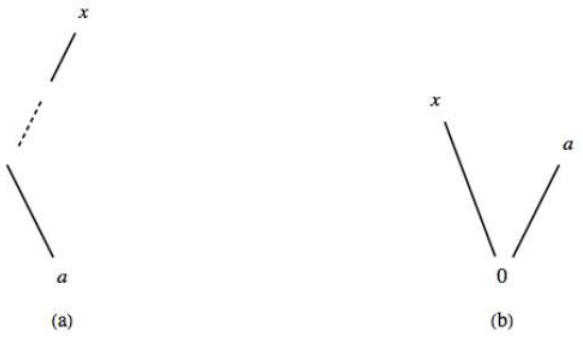


Figure 13.4.2: Conditions for an atom

An alternate definition of an atom is based on the concept of "covering."

Definition 13.4.2: The Covering Relation

Given a Boolean algebra $[B; \vee, \wedge, \bar{}\,]$, let $x, z \in B$. We say that z ***covers*** x iff $x \prec z$ and there does not exist $y \in B$ with $x \prec y \prec z$.

It can be proven that the atoms of Boolean algebra are precisely those elements that cover the zero element.

The set of atoms of the Boolean algebra $[D_{30}; \vee, \wedge, \bar{}\,]$ is $M = \{2, 3, 5\}$. To see that $a = 2$ is an atom, let x be any non-least element of D_{30} and note that one of the two conditions $x \wedge 2 = 2$ or $x \wedge 2 = 1$ holds. Of course, to apply the definition to this Boolean algebra, we must remind ourselves that in this case the 0-element is 1, the operation $\wedge$ is greatest common divisor, and the poset relation is "divides." So if $x = 10$, we have $10 \wedge 2 = 2$ (or $2 \mid 10$), so Condition 1 holds. If $x = 15$, the first condition is not true. (Why?) However, Condition 2, $15 \wedge 2 = 1$, is true. The reader is encouraged to show that 3 and 5 also satisfy the definition of an atom. Next, if we should compute the join (the least common multiple in this case) of all possible combinations of the atoms 2, 3, and 5 to generate all nonzero (non-1 in this case) elements of D_{30}. For example, $2 \vee 3 \vee 5 = 30$ and $2 \vee 5 = 10$. We state this concept formally in the following theorem, which we give without proof.

Theorem 13.4.1

Let $\mathcal{B} = [B; \vee, \wedge, \bar{}\,]$ be any finite Boolean algebra. Let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of all atoms of $\mathcal{B}$. Then every element in B can be expressed uniquely as the join of a subset of A.

The least element in relation to this theorem bears noting. If we consider the empty set of atoms, we would consider the join of elements in the empty set to be the least element. This makes the statement of the theorem above a bit more tidy since we don't need to qualify what elements can be generated from atoms.

We now ask ourselves if we can be more definitive about the structure of different Boolean algebras of a given order. Certainly, the Boolean algebras $[D_{30}; \vee, \wedge, \wedge\ \bar{}\,]$ and $[\mathcal{P}(A); \cup, \cap, {}^c]$ have the same graph (that of Figure 13.4.1), the same number of atoms, and, in all respects, look the same except for the names of the elements and the operations. In fact, when we apply corresponding operations to corresponding elements, we obtain corresponding results. We know from Chapter 11 that this means that the two structures are isomorphic as Boolean algebras. Furthermore, the graphs of these examples are exactly the same as that of Figure 13.4.1, which is an arbitrary Boolean algebra of order $8 = 2^3$.

In these examples of a Boolean algebra of order 8, we note that each had 3 atoms and $2^3 = 8$ number of elements, and all were isomorphic to $[\mathcal{P}(A); \cup, \cap, {}^c]$, where $A = \{a, b, c\}$. This leads us to the following questions:

- Are there any different (nonisomorphic) Boolean algebras of order 8?
- What is the relationship, if any, between finite Boolean algebras and their atoms?
- How many different (nonisomorphic) Boolean algebras are there of order 2? Order 3? Order 4? etc.

The answers to these questions are given in the following theorem and corollaries.

Theorem 13.4.2

Let $\mathcal{B} = [B; \vee, \wedge, -]$ be any finite Boolean algebra, and let A be the set of all atoms of $\mathcal{B}$. Then $[\mathcal{P}(A); \cup, \cap, {}^c]$ is isomorphic to $[B; \vee, \wedge, -]$

Proof

An isomorphism that serves to prove this theorem is $T : \mathcal{P}(A) \to B$ defined by $T(S) = \bigvee_{a \in S} a$, where $T(\emptyset)$ is interpreted as the zero of $\mathcal{B}$. We leave it to the reader to prove that this is indeed an isomorphism.

Corollary 13.4.1

Every finite Boolean algebra $\mathcal{B} = [B; \vee, \wedge, \bar{}\,]$ has 2^n elements for some positive integer n.

Proof

Let A be the set of all atoms of $\mathcal{B}$ and let $|A| = n$. Then there are exactly 2^n elements (subsets) in $\mathcal{P}(A)$, and by Theorem 13.4.2 $[B; \vee, \wedge, \bar{}\,]$ is isomorphic to $[\mathcal{P}(A); \cup, \cap\, {}^c]$ and must also have 2^n elements.

Corollary 13.4.2

All Boolean algebras of order 2^n are isomorphic to one another.

Proof

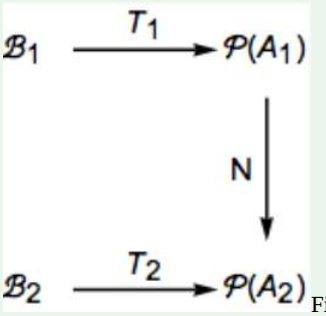


Figure 13.4.1: Isomorphisms to be combined

Every Boolean algebra of order 2^n is isomorphic to $[\mathcal{P}(A); \cup, \cap, {}^c]$ when $|A| = n$. Hence, if $\mathcal{B}_1$ and $\mathcal{B}_2$ each have 2^n elements, they each have n atoms. Suppose their sets of atoms are A_1 and A_2, respectively. We know there are isomorphisms T_1 and T_2, where $T_i : \mathcal{B}_i \to \mathcal{P}(A_i),\ i = 1, 2$. In addition we have an isomorphism, N from $\mathcal{P}(A_1)$ into $\mathcal{P}(A_2)$, which we ask you to prove in Exercise 13.4.9 We can combine these isomorphisms to produce the isomorphism $T_2^{-1} \circ N \circ T_1 : \mathcal{B}_1 \to \mathcal{B}_2,$ which proves the corollary.

The above theorem and corollaries tell us that we can only have finite Boolean algebras of orders $2^1, 2^2, 2^3, \ldots, 2^n$, and that all finite Boolean algebras of any given order are isomorphic. These are powerful tools in determining the structure of finite Boolean algebras. In the next section, we will discuss one of the easiest ways of describing a Boolean algebra of any given order.

13.4.1: Exercises

Exercise 13.4.1

a. Show that $a = 2$ is an atom of the Boolean algebra $[D_{30}; \vee, \wedge, -]$.
b. Repeat part a for the elements 3 and 5 of D_{30}.
c. Verify Theorem 13.4.1 for the Boolean algebra $[D_{30}; \vee, \wedge, -]$.

Answer

a. For $a = 3$ we must show that for each $x \in D_{30}$ one of the following is true: $x \wedge 3 = 3$ or $x \wedge 3 = 1$. We do this through the following table:

x	verification
1	$1 \wedge 3 = 1$
2	$2 \wedge 3 = 1$
3	$3 \wedge 3 = 3$
5	$5 \wedge 3 = 1$
6	$6 \wedge 3 = 3$
10	$20 \wedge 3 = 1$
15	$15 \wedge 3 = 3$
30	$30 \wedge 3 = 3$

For $a = 5$, a similar verification can be performed.
b. $6 = 2 \vee 3,\ 10 = 2 \vee 5,\ 15 = 3 \vee 5$, and $30 = 2 \vee 3 \vee 5$.

Exercise 13.4.2

Let $A = \{a, b, c\}$.

a. Rewrite the definition of atom for $[\mathcal{P}(A); \cup, \cap, c]$. What does $a \leq x$ mean in this example?
b. Find all atoms of $[\mathcal{P}(A); \cup, \cap, c]$.
c. Verify Theorem 13.4.1 for $[\mathcal{P}(A); c, \cup, \cap]$.

Exercise 13.4.3

Verify Theorem 13.4.2 and its corollaries for the Boolean algebras in Exercises 13.4.1 and 13.4.2 of this section.

Answer

If $B = D_{30}$ 30 then $A = \{2, 3, 5\}$ and D_{30} is isomorphic to $\mathcal{P}(A)$, where

$1 \leftrightarrow \emptyset$ $\quad$ $5 \leftrightarrow \{5\}$
$2 \leftrightarrow \{2\}$ $\quad$ $10 \leftrightarrow \{2, 5\}$
$3 \leftrightarrow \{3\}$ $\quad$ $15 \leftrightarrow \{3, 5\}$
$6 \leftrightarrow \{2, 3\}$ $\quad$ $30 \leftrightarrow \{2, 3, 5\}$

and

$$\text{Join} \leftrightarrow \text{Union}$$
$$\text{Meet} \leftrightarrow \text{Intersection}$$
$$\text{Complement} \leftrightarrow \text{Set Complement}$$

Exercise 13.4.4

Give an example of a Boolean algebra of order 16 whose elements are certain subsets of the set $\{1, 2, 3, 4, 5, 6, 7\}$

Exercise 13.4.5

Corollary 13.4.1 implies that there do not exist Boolean algebras of orders 3, 5, 6, 7, 9, etc. (orders different from 2^n). Without this corollary, directly show that we cannot have a Boolean algebra of order 3.

Hint

Assume that $[B; \vee, \wedge, -]$ is a Boolean algebra of order 3 where $B = \{0, x, 1\}$ and show that this cannot happen by investigating the possibilities for its operation tables.

Answer

Assume that $x \neq 0$ or 1 is the third element of a Boolean algebra. Then there is only one possible set of tables for join and meet, all following from required properties of the Boolean algebra.

$\vee$	0	x	1
0	0	x	1
x	x	x	1
1	1	1	1

$\wedge$	0	x	1
0	0	0	0
x	0	x	x
1	0	x	1

Next, to find the complement of x we want y such that $x \wedge y = 0$ and $x \vee y = 1$. No element satisfies both conditions; hence the lattice is not complemented and cannot be a Boolean algebra. The lack of a complement can also be seen from the ordering diagram from which $\wedge$ and $\vee$ must be derived.

Exercise 13.4.6

a. There are many different, yet isomorphic, Boolean algebras with two elements. Describe one such Boolean algebra that is derived from a power set, $\mathcal{P}(A)$, under $\subseteq$. Describe a second that is described from D_n, for some $n \in P$, under "divides."
b. Since the elements of a two-element Boolean algebra must be the greatest and least elements, 1 and 0, the tables for the operations on $\{0, 1\}$ are determined by the Boolean algebra laws. Write out the operation tables for $[\{0, 1\}; \vee, \wedge, -]$.

Exercise 13.4.7

Find a Boolean algebra with a countably infinite number of elements.

Answer

Let X be any countably infinite set, such as the integers. A subset of X is ***cofinite*** if it is finite or its complement is finite. The set of all cofinite subsets of X is:

a. Countably infinite - this might not be obvious, but here is a hint. Assume $X = \{x_0, x_1, x_2, \ldots\}$. For each finite subset A of X, map that set to the integer $\sum_{i=0}^{\infty} \chi_A(x_i)\, 2^i$ You can do a similar thing to sets that have a finite complement, but map them to negative integers. Only one minor adjustment needs to be made to accommodate both the empty set and X.
b. Closed under union
c. Closed under intersection, and
d. Closed under complementation.

Therefore, if $B = \{A \subseteq X : A \text{ is cofinite}\}$, then B is a countable Boolean algebra under the usual set operations.

Exercise 13.4.8

Prove that the direct product of two Boolean algebras is a Boolean algebra.

Hint

"Copy" the corresponding proof for groups in Section 11.6.

Exercise 13.4.9

Prove if two finite sets A_1 and A_2 both have n elements then $[\mathcal{P}(A_1); \cup, \cap, {}^c]$ is isomorphic to $[\mathcal{P}(A_2); \cup, \cap, {}^c]$

Exercise 13.4.10

Prove an element of a Boolean algebra is an atom if and only if it covers the zero element.

This page titled 13.4: Atoms of a Boolean Algebra is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.5: Finite Boolean Algebras as n-tuples of 0's and 1's

From the previous section we know that all finite Boolean algebras are of order 2^n, where n is the number of atoms in the algebra. We can therefore completely describe every finite Boolean algebra by the algebra of power sets. Is there a more convenient, or at least an alternate way, of defining finite Boolean algebras? In Chapter 11 we found that we could produce new groups by taking Cartesian products of previously known groups. We imitate this process for Boolean algebras.

The simplest nontrivial Boolean algebra is the Boolean algebra on the set $B_2 = \{0, 1\}$. The ordering on B_2 is the natural one, $0 \leq 0, 0 \leq 1, 1 \leq 1$. If we treat 0 and 1 as the truth values "false" and "true," respectively, we see that the Boolean operations $\vee(\text{join})$ and $\wedge(\text{meet})$ are nothing more than the logical operation with the same symbols. The Boolean operation, $-$, (complementation) is the logical $\neg$ (negation). In fact, this is why these symbols were chosen as the names of the Boolean operations. The operation tables for $[B_2; \vee, \wedge, -]$ are simply those of "or," "and," and "not," which we repeat here.

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

u	$\bar{u}$
0	1
1	0

By Theorem 13.4.2 and its corollaries, all Boolean algebras of order 2 are isomorphic to this one.

We know that if we form $B_2 \times B_2 = B_2^2$ we obtain the set $\{(0,0),(0,1),(1,0),(1,1)\}$,a set of order 4. We define operations on B_2^2 the natural way, namely componentwise, so that $(0,1) \vee (1,1) = (0 \vee 1, 1 \vee 1) = (1,1)$, $(0,1) \wedge (1,1) = (0 \wedge 1, 1 \wedge 1) = (0,1)$ and $\overline{(0,1)} = (\bar{0}, \bar{1}) = (1,0)$. We claim that B_2^2 is a Boolean algebra under the componentwise operations. Hence, $\left[B_2^2; \vee, \wedge, {}^- \right]$ is a Boolean algebra of order 4. Since all Boolean algebras of order 4 are isomorphic to one other, we have found a simple way of describing all Boolean algebras of order 4.

It is quite clear that we can describe any Boolean algebra of order 8 by considering $B_2 \times B_2 \times B_2 = B_2^3$ and, more generally, any Boolean algebra of order 2^n with $B_2^n = B_2 \times B_2 \times \cdots \times B_2$ (n factors).

13.5.1: Exercises

Exercise 13.5.1

a. Write out the operation tables for $\left[B_2^2; \vee, \wedge, -\right]$.
b. Draw the Hasse diagram for $\left[B_2^2; \vee, \wedge, -\right]$ and compare your results with Figure 6.3.1.
c. Find the atoms of this Boolean algebra.

Answer

a.

$\vee$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,1)$	$(1,1)$	$(1,1)$
$(1,0)$	$(1,0)$	$(1,1)$	$(1,0)$	$(1,1)$
$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$

$\wedge$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(0,1)$	$(0,0)$	$(0,1)$	$(0,0)$	$(0,1)$
$(1,0)$	$(0,0)$	$(0,0)$	$(1,0)$	$(1,0)$
$(1,1)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$

u	$\bar{u}$
$(0,0)$	$(1,1)$
$(0,1)$	$(1,0)$
$(1,0)$	$(0,1)$
$(1,1)$	$(0,0)$

b. The graphs are isomorphic.
c. (0, 1) and (1,0)

Exercise 13.5.2

a. Write out the operation tables for $\left[B_2^3; \vee, \wedge, -\right]$.
b. Draw the Hasse diagram for $\left[B_2^3; \vee, \wedge, -\right]$

Exercise 13.5.3

a. List all atoms of B_2^4.
b. Describe the atoms of $B_2^n, n \geq 1$.

Answer

a. $(1,0,0,0),(0,1,0,0),(0,0,1,0)$,and $(0,0,0,1)$are the atoms.
b. The n-tuples of bits with exactly one 1.

Exercise 13.5.4

Theorem 13.4.2 tells us we can think of any finite Boolean algebra in terms of sets. In Chapter 4, we defined minsets Definition 4.3.1 and minset normal form Definition 4.3.2. Rephrase these definitions in the language of Boolean algebra. The generalization of minsets are called ***minterms***.

This page titled 13.5: Finite Boolean Algebras as n-tuples of 0's and 1's is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.6: Boolean Expressions

In this section, we will use our background from the previous sections and set theory to develop a procedure for simplifying Boolean expressions. This procedure has considerable application to the simplification of circuits in switching theory or logical design.

Definition 13.6.1: Boolean Expression

Let $[B; \vee, \wedge, -]$ be any Boolean algebra, and let $x_1, x_2, \ldots, x_k$ be variables in B; that is, variables that can assume values from B. A Boolean expression generated by $x_1, x_2, \ldots, x_k$ is any valid combination of the x_i and the elements of B with the operations of meet, join, and complementation.

This definition is the analog of the definition of a proposition generated by a set of propositions, presented in Section 3.2.

Each Boolean expression generated by k variables, $e(x_1, \ldots, x_k)$, defines a function $f : B^k \to B$ where $f(a_1, \ldots, a_k) = e(a_1, \ldots, a_k)$. If B is a finite Boolean algebra, then there are a finite number of functions from B^k into B. Those functions that are defined in terms of Boolean expressions are called Boolean functions. As we will see, there is an infinite number of Boolean expressions that define each Boolean function. Naturally, the "shortest" of these expressions will be preferred. Since electronic circuits can be described as Boolean functions with $B = B_2$, this economization is quite useful.

In what follows, we make use of Exercise 7.1.5 in Section 7.1 for counting number of functions.

Example 13.6.1: Two Variables Over B_2

Consider any Boolean algebra of order 2, $[B; \vee, \wedge, -]$. How many functions $f : B^2 \to B$ are there? First, all Boolean algebras of order 2 are isomorphic to $[B_2; \vee, \wedge, -]$ so we want to determine the number of functions $f : B_2^2 \to B_2$. If we consider a Boolean function of two variables, x_1 and x_2, we note that each variable has two possible values 0 and 1, so there are 2^2 ways of assigning these two values to the $k = 2$ variables. Hence, the table below has $2^2 = 4$ rows. So far we have a table such as this one:

x_1	x_2	$f(x_1, x_2)$
0	0	?
0	1	?
1	0	?
1	1	?

How many possible different functions can there be? To list a few: $f_1(x_1, x_2) = x_1$, $f_2(x_1, x_2) = x_2$, $f_3(x_1, x_2) = x_1 \vee x_2$, $f_4(x_1, x_2) = (x_1 \wedge \overline{x_2}) \vee x_2$, $f_5(x_1, x_2) = x_1 \wedge x_2 \vee \overline{x_2}$, etc. Each of these will fill in the question marks in the table above. The tables for f_1 and f_3 are

x_1	x_2	$f_1(x_1, x_2)$
0	0	0
0	1	0
1	0	1
1	1	1

x_1	x_2	$f_3(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	1

Two functions are different if and only if their tables are different for at least one row. Of course by using the basic laws of Boolean algebra we can see that $f_3 = f_4$. Why? So if we simply list by brute force all "combinations" of x_1 and x_2 we will obtain unnecessary duplication. However, we note that for any combination of the variables x_1, and x_2 there are only two possible values for $f(x_1, x_2)$, namely 0 or 1. Thus, we could write $2^4 = 16$ different functions on 2 variables.

Now, let's count the number of different Boolean functions in a more general setting. We will consider two cases: first, when $B = B_2$, and second, when B is any finite Boolean algebra with 2^n elements.

Let $B = B_2$. Each function $f : B^k \to B$ is defined in terms of a table having 2^k rows. Therefore, since there are two possible images for each element of B^k, there are 2 raised to the 2^k, or 2^{2^k} different functions. We will show that every one of these

functions is a Boolean function.

Now suppose that $|B| = 2^n > 2$. A function from B^k into B can still be defined in terms of a table. There are $|B|^k$ rows to each table and $|B|$ possible images for each row. Therefore, there are 2^n raised to the power 2^{nk} different functions. We will show that if $n > 1$, not every one of these functions is a Boolean function.

Since all Boolean algebras are isomorphic to a Boolean algebra of sets, the analogues of statements in sets are useful in Boolean algebras.

Definition 13.6.2: Minterm

A Boolean expression generated by $x_1, x_2, \ldots, x_k$ that has the form

$$\bigwedge_{i=1}^{k} y_i,$$

where each y_i may be either x_i or $\overline{x_i}$ is called a minterm generated by $x_1, x_2, \ldots, x_k$. We use the notation $M_{\delta_1 \delta_2 \cdots \delta_k}$ for the minterm generated by $x_1, x_2, \ldots, x_k$, where $y_i = x_i$ if $\delta_i = 1$ and $y_i = \bar{x}_i$ if $\delta_i = 0$

An example of the notation is that $M_{110} = x_1 \wedge x_2 \wedge \bar{x_3}$.

By a direct application of the Rule of Products we see that there are 2^k different minterms generated by $x_1, \ldots, x_k$.

Definition 13.6.3: Minterm Normal Form

A Boolean expression generated by $x_1, \ldots, x_k$ is in minterm normal form if it is the join of expressions of the form $a \wedge m$, where $a \in B$ and m is a minterm generated by $x_1, \ldots, x_k$. That is, it is of the form

$$\bigvee_{j=1}^{p} (a_j \wedge m_j) \tag{13.6.1}$$

where $p = 2^k$, and $m_1, m_2, \ldots, m_p$ are the minterms generated by $x_1, \ldots, x_k$.

Note 13.6.1

- We seem to require every minterm generated by $x_1, \ldots, x_k$, in (13.6.1), and we really do. However, some of the values of a_j can be $\mathbf{0}$, which effectively makes the corresponding minterm disappear.
- If $B = B_2$, then each a_j in a minterm normal form is either 0 or 1. Therefore, $a_j \wedge m_j$ is either 0 or m_j.

Theorem 13.6.1: Uniqueness of Minterm Normal Form

Let $e(x_1, \ldots, x_k)$ be a Boolean expression over B. There exists a unique minterm normal form $M(x_1, \ldots, x_k)$ that is equivalent to $e(x_1, \ldots, x_k)$ in the sense that e and M define the same function from B^k into B.

The uniqueness in this theorem does not include the possible ordering of the minterms in M (commonly referred to as "uniqueness up to the order of minterms"). The proof of this theorem would be quite lengthy, and not very instructive, so we will leave it to the interested reader to attempt. The implications of the theorem are very interesting, however.

If $|B| = 2^n$, then there are 2^n raised to the 2^k different minterm normal forms. Since each different minterm normal form defines a different function, there are a like number of Boolean functions from B^k into B. If $B = B_2$, there are as many Boolean functions (2 raised to the 2^k) as there are functions from B^k into B, since there are 2 raised to the 2^n functions from B^k into B. The significance of this result is that any desired function can be realized using electronic circuits having 0 or 1 (off or on, positive or negative) values.

More complex, multivalued circuits corresponding to boolean algebras with more than two values would not have this flexibility because of the number of minterm normal forms, and hence the number of boolean functions, is strictly less than the number of functions.

We will close this section by examining minterm normal forms for expressions over B_2 , since they are a starting point for circuit economization.

Example 13.6.2

Consider the Boolean expression $f(x_1, x_2) = x_1 \vee \overline{x_2}$. One method of determining the minterm normal form of f is to think in terms of sets. Consider the diagram with the usual translation of notation in Figure 13.6.1. Then

$$\begin{aligned} f(x_1, x_2) &= (\overline{x_1} \wedge \overline{x_2}) \vee (x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_2) \\ &= M_{00} \vee M_{10} \vee M_{11} \end{aligned}$$

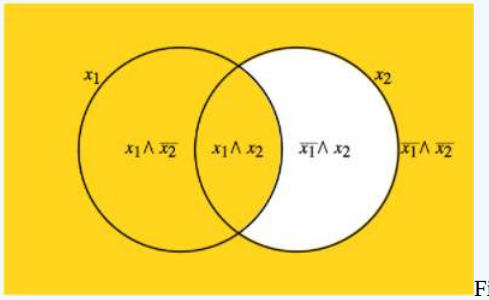


Figure 13.6.1: Visualization of minterms for $x_1 \vee \bar{x_2}$

Table 13.6.1: Definition of the boolean function g

x_1	x_2	x_3	$g(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Example 13.6.3

Consider the function $g : B_2^3 \to B_2$ defined by Table 13.6.1.

The minterm normal form for g can be obtained by taking the join of minterms that correspond to rows that have an image value of 1. If $g(a_1, a_2, a_3) = 1$, then include the minterm $y_1 \wedge y_2 \wedge y_3$ where

$$y_j = \begin{cases} x_j & \text{if } a_j = 1 \\ \bar{x_j} & \text{if } a_j = 0 \end{cases}$$

Or, to use alternate notation, include $M_{a_1 a_2 a_3}$ in the expression if and only if $g(a_1, a_2, a_3) = 1$

Therefore,

$$g(x_1, x_2, x_3) = (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \overline{x_3}).$$

The minterm normal form is a first step in obtaining an economical way of expressing a given Boolean function. For functions of more than three variables, the above set theory approach tends to be awkward. Other procedures are used to write the normal form. The most convenient is the Karnaugh map, a discussion of which can be found in any logical design/switching theory text (see, for example, [18]), on Wikipedia.

13.6.1: Exercises

Exercise 13.6.1

a. Write the 16 possible functions of Example 13.6.1.
b. Write out the tables of several of the above Boolean functions to show that they are indeed different.
c. Determine the minterm normal forms of

i. $g_1(x_1, x_2) = x_1 \vee x_2$,
ii. $g_2(x_1, x_2) = \overline{x_1} \vee \overline{x_2}$
iii. $g_3(x_1, x_2) = \overline{x_1 \wedge x_2}$
iv. $g_4(x_1, x_2) = 1$

Answer

a.

$f_1(x_1, x_2) = 0$
$f_2(x_1, x_2) = (\overline{x_1} \wedge \overline{x_2})$
$f_3(x_1, x_2) = (\overline{x_1} \wedge x_2)$
$f_4(x_1, x_2) = (x_1 \wedge \overline{x_2})$
$f_5(x_1, x_2) = (x_1 \wedge x_2)$
$f_6(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)) = \overline{x_1}$
$f_7(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (x_1 \wedge \overline{x_2})) = \overline{x_2}$
$f_8(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (x_1 \wedge x_2)) = ((x_1 \wedge x_2) \vee (\overline{x_1} \wedge \overline{x_2}))$
$f_9(x_1, x_2) = ((\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})) = ((x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2))$
$f_{10}(x_1, x_2) = ((\overline{x_1} \wedge x_2) \vee (x_1 \wedge x_2)) = x_2$
$f_{11}(x_1, x_2) = ((x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_2)) = x_1$
$f_{12}(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})) = (\overline{x_1} \vee \overline{x_2})$
$f_{13}(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2) \vee (x_1 \wedge x_2)) = (\overline{x_1} \vee x_2)$
$f_{14}(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_2)) = (x_1 \vee \overline{x_2})$
$f_{15}(x_1, x_2) = ((\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_2)) = (x_1 \vee x_2)$
$f_{16}(x_1, x_2) = ((\overline{x_1} \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_2)) = 1$

b. The truth table for the functions in part (a) are

x_1	x_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
0	0	0	1	0	0	0	1	1	1
0	1	0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0	1	0
1	1	0	0	0	0	1	0	0	1

x_1	x_2	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	0	0	1	1	1	0	1
0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1	1	1

c.

i. $g_1(x_1, x_2) = f_{15}(x_1, x_2)$
ii. $g_2(x_1, x_2) = f_{12}(x_1, x_2)$
iii. $g_3(x_1, x_2) = f_{12}(x_1, x_2)$
iv. $g_4(x_1, x_2) = f_{16}(x_1, x_2)$

Exercise 13.6.2

Consider the Boolean expression $f(x_1, x_2, x_3) = (\overline{x_3} \wedge x_2) \vee (\overline{x_1} \wedge x_3) \vee (x_2 \wedge x_3)$ on $\left[B_2^3; \vee, \wedge, -\right]$.

a. Simplify this expression using basic Boolean algebra laws.
b. Write this expression in minterm normal form.
c. Write out the table for the given function defined by f and compare it to the tables of the functions in parts a and b.
d. How many possible different functions in three variables on $[B_2; \vee, \wedge, -]$ are there?

Exercise 13.6.3

Let $[B; \vee, \wedge, -]$ be a Boolean algebra of order 4, and let f be a Boolean function of two variables on B.

a. How many elements are there in the domain of f?
b. How many different Boolean functions are there of two, variables? Three variables?
c. Determine the minterm normal form of $f(x_1, x_2) = x_1 \vee x_2$.
d. If $B = \{0, a, b, 1\}$, define a function from B^2 into B that is not a Boolean function.

Answer

a. The number of elements in the domain of f is $16 = 4^2 = |B|^2$
b. With two variables, there are $4^3 = 256$ different Boolean functions. With three variables, there are $4^8 = 65536$ different Boolean functions.
c. $f(x_1,\ x_2) = (1 \wedge \overline{x_1} \wedge \overline{x_2}) \vee (1 \wedge \overline{x_1} \wedge \overline{x_2}) \vee (1 \wedge x1 \wedge \overline{x_2}) \vee (0 \wedge x1 \wedge x2)$
d. Consider $f : B^2 \to B$, defined by $f(0,0) = 0$, $f(0,1) = 1$, $f(1,0) = a$, $f(1,1) = a$, and $f(0,a) = b$, with the images of all other pairs in B^2 defined arbitrarily. This function is not a Boolean function. If we assume that it is Boolean function then f can be computed with a Boolean expression $M(x_1, x_2)$. This expression can be put into minterm normal form:

$$M(x_1, x_2) = (c_1 \wedge \overline{x_1} \wedge \overline{x_2}) \vee (c_2 \wedge \overline{x_1} \wedge x_2) \vee (c_3 \wedge x_1 \wedge \overline{x_2}) \vee (c_4 \wedge x_1 \wedge x_2)$$

$$\begin{aligned} f(0,0) &= 0 \Rightarrow M(0,0) = 0 \Rightarrow c_1 = 0 \\ f(0,1) &= 1 \Rightarrow M(0,0) = 1 \Rightarrow c_2 = 1 \\ f(1,0) &= a \Rightarrow M(0,0) = a \Rightarrow c_3 = a \\ f(1,1) &= a \Rightarrow M(0,0) = a \Rightarrow c_4 = a \end{aligned}$$

Therefore, $M(x_1, x_2) = (\overline{x_1} \wedge x_2) \vee (a \wedge x_1 \wedge \overline{x_2}) \vee (a \wedge x_1 \wedge x_2)$ and so, using this formula,
$M(0, a) = (\overline{0} \wedge a) \vee (a \wedge 0 \wedge \overline{a}) \vee (a \wedge 0 \wedge a) = a$ This contradicts $f(0, a) = b$, and so f is not a Boolean function.

This page titled 13.6: Boolean Expressions is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

13.7: A Brief Introduction to Switching Theory and Logic Design

Disclaimer

I'm still looking for a good application for drawing logic gates. The figures here are quite rough.

Early computers relied on many switches to perform the logical operations needed for computation. This was true as late as the 1970's when early personal computers such as the Altair (Figure 13.7.1) started to appear. Pioneering computer scientists such as Claude Shannon realized that the operation of these computers could be simplified by making use of an isomorphism between computer circuits and boolean algebra. The term ***Switching Theory*** was used at the time. Logical gates realized through increasingly smaller and smaller integrated circuits still perform the same functions as in early computers, but using purely electronic means. In this section, we give examples of some switching circuits. Soon afterward, we will transition to the more modern form of circuits that are studied in ***Logic Design***, where gates replace switches. Our main goal is to give you an overview of how boolean functions corresponds to any such circuit. We will introduce the common system notation used in logic design and show how it corresponds with the mathematical notation of Boolean algebras. Any computer scientist should be familiar with both systems.



Figure 13.7.1: The Altair Computer, an early PC, by Todd Dailey, Creative Commons

The simplest switching device is the on-off switch. If the switch is closed/ON, current will pass through it; if it is open/OFF, current will not pass through it. If we designate ON by 1, and OFF by 0, we can describe electrical circuits containing switches by Boolean expressions with the variables representing the variable states of switches or the variable bits passing through gates.

The electronics involved in these switches take into account whether we are negating a switch or not. For electromagnetic switches, a magnet is used to control whether the switch is open or closed. The magnets themselves may be controlled by simple ON/OFF switches. There are two types of electromagnetic switches. One is normally open (OFF) when the magnet is not activated, but activating the magnet will close the circuit and the switch is then ON. A separate type of switch corresponds with a negated switch. For that type, the switch is closed when the magnet is not activated, and when the magnet is activated, the switch opens. We won't be overly concerned with the details of these switches or the electronics corresponding to logical gates. We will simply assume they are available to plug into a circuit. For simplicity, we use the inversion symbol on a variable that labels a switch to indicate that it is a switch of the second type, as in Figure 13.7.3

Note 13.7.1

Standby power generators that many people have in their homes use a transfer switch to connect the generator to the home power system. This switch is open (OFF) if there is power coming from the normal municipal power supply. It stays OFF because a magnet is keeping it open. When power is lost, the magnet is no longer activated, and the switch closes and is ON. So the transfer switch is a normally ON switch.



Figure 13.7.2: Representation of a normally OFF switch controlled by variable x_1

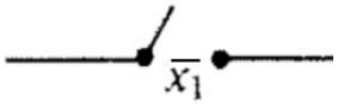


Figure 13.7.3: Representation of a normally ON switch controlled by variable x_1

The standard notation used for Boolean algebra operations in switching theory and logic design is $+$ for join, instead of $\vee$; and $\cdot$ for meet, instead of $\wedge$. Complementation is the same in both notational systems, denoted with an overline.

The expression $x_1 \cdot x_2$ represents the situation in which a series of two switches appears in sequence as in Figure 13.7.4 In order for current to flow through the circuit, both switches must be ON; that is, they must both have the value 1. Similarly, a pair of parallel switches, as in Figure 13.7.5, is described algebraically by $x_1 + x_2$. Here, current flows through this part of the circuit as long as at least on of the switches is ON.

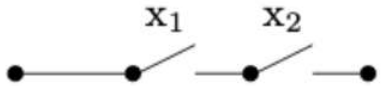


Figure 13.7.4: Two switches in AND configuration realizing $x_1 \cdot x_2$

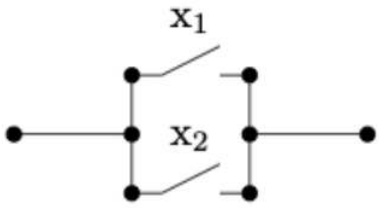


Figure 13.7.5: Two switches in OR configuration realizing $x_1 + x_2$

All laws and concepts developed previously for Boolean algebras hold. The only change is purely notational. We make the change in this section solely to introduce the reader to another frequently used system of notation.

Many of the laws of Boolean algebra can be visualized thought switching theory. For example, the distributive law of meet over join is expressed as

$$x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3.$$

The switching circuit analogue of the above statement is that the circuits in the two images below are equivalent. In circuit (b), the presence of two x_1's might represent two electromagnetic switches controlled by the same magnet.

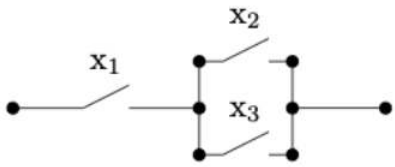


Figure 13.7.6: (a)

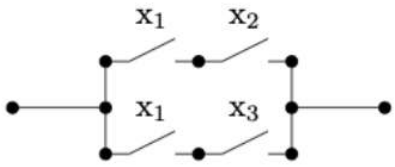


Figure 13.7.7: (b)

The circuits in a computer are now composed of large quantities of gates, which serve the same purpose as switches, but can be miniaturized to a great degree. For example, the OR gate, usually drawn as in Figure 13.7.8 implements the logical OR function. This happens electronically, but is equivalent to Figure 13.7.5. The AND gate, which is equivalent to two sequential switches is shown in Figure 13.7.8

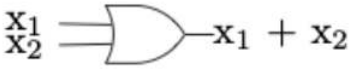


Figure 13.7.8: An OR gate



Figure 13.7.9: An AND gate

The complementation process is represented in a gate diagram by an inverter, as pictured in Figure 13.7.10

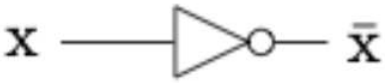


Figure 13.7.10: Inverter, or NOT gate

When drawing more complex circuits, multiple AND's or OR's are sometimes depicted using a more general gate drawing. For example if we want to depict an OR gate with three inputs that is ON as long as at least one input is ON, we would draw it as in Figure 13.7.11, although this would really be two binary gates, as in Figure 13.7.12 Both diagrams are realizing the boolean expression $x_1 + x_2 + x_3$. Strictly speaking, the gates in Figure 13.7.12represent $(x_1 + x_2) + x_3$, but the associative law for join tells us that the grouping doesn't matter.

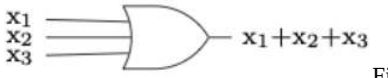


Figure 13.7.11: Simple version of a ternary OR gate



Figure 13.7.12: A ternary OR gate created with binary OR gates

In Figure 13.7.13 we show a few other commonly used gates, XOR, NAND, and NOR, which correspond to the boolean exressions $x_1 \oplus x_2$, $\overline{x_1 \cdot x_2}$, and $\overline{x_1 + x_2}$, respectively.

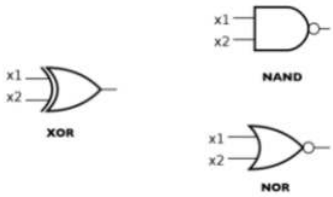


Figure 13.7.13: Other common gates

Let's start with a logic circuit and see how the laws of boolean algebra can help us simplify it.

Example 13.7.1: Simplification of a Circuit

Consider the circuit in Figure 13.7.14 As usual, we assume that three inputs enter on the left and the output exits on the right.

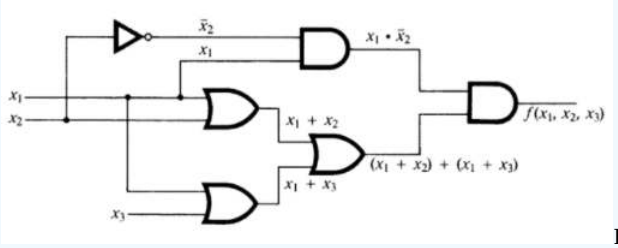


Figure 13.7.14: Initial gate diagram

If we trace the inputs through the gates we see that this circuit realizes the boolean function

$$f(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot ((x_1 + x_2) + (x_1 + x_3)).$$

We simplify the boolean expression that defines f, simplifying the circuit in so doing. You should be able to identify the laws of Boolean algebra that are used in each of the steps. See Exercise 13.7.1.

$$
\begin{aligned}
x_1 \cdot \overline{x_2} \cdot ((x_1 + x_2) + (x_1 + x_3)) &= x_1 \cdot \overline{x_2} \cdot (x_1 + x_2 + x_3) \\
&= x_1 \cdot \overline{x_2} \cdot x_1 + x_1 \cdot \overline{x_2} \cdot x_2 + x_1 \cdot \overline{x_2} \cdot x_3 \\
&= x_1 \cdot \overline{x_2} + 0 \cdot x_1 + x_3 \cdot x_1 \cdot \overline{x_2} \\
&= x_1 \cdot \overline{x_2} + x_3 \cdot x_1 \cdot \overline{x_2} \\
&= x_1 \cdot \overline{x_2} \cdot (1 + x_3) \\
&= x_1 \cdot \overline{x_2}
\end{aligned}
$$

Therefore, $f(x_1, x_2, x_3) = x_1 \cdot \overline{x_2}$, which can be realized with the much simpler circuit in Figure 13.7.15 without using the input x_3.

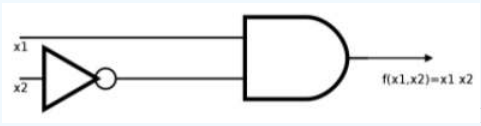


Figure 13.7.15: Simplified gate diagram

Next, we start with a table of desired outputs based on three bits of input and design an efficient circuit to realize this output.

Example 13.7.2

Consider the following table of desired outputs for the three input bits x_1, x_2, x_3.

Table 13.7.1: Desired output table

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

The first step is to write the Minterm Normal Form, Definition 13.6.3, of f. Since we are working with the two value Boolean algebra, B_2, the constants in each minterm are either 0 or 1, and we simply list the minterms that have a 1. These correspond with the rows of the table above that have an output of 1. We will then attempt to simplify the expression as much as possible.

$$
\begin{aligned}
f(x_1, x_2, x_3) &= (\overline{x_1} \cdot \overline{x_2} \cdot x_3) + (x_1 \cdot \overline{x_2} \cdot \overline{x_3}) + (x_1 \cdot \overline{x_2} \cdot x_3) \\
&= \overline{x_2} \cdot ((\overline{x_1} \cdot x_3) + (x_1 \cdot \overline{x_3}) + (x_1 \cdot x_3)) \\
&= \overline{x_2} \cdot ((\overline{x_1} \cdot x_3) + x_1 \cdot (\overline{x_3} + x_3)) \\
&= \overline{x_2} \cdot ((\overline{x_1} \cdot x_3) + x_1)
\end{aligned}
$$

Therefore we can realize our table with the boolean function $f(x_1, x_2, x_3) = \overline{x_2} \cdot ((\overline{x_1} \cdot x_3) + x_1)$. A circuit diagram for this function is Figure 13.7.16 But is this the simplest circuit that realizes the table? See Exercise 13.7.3

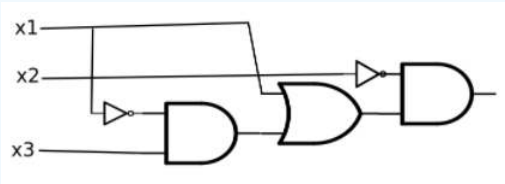


Figure 13.7.16: A realization of the table of desired outputs.

13.7.1: 13.7.1: Exercises

Exercise 13.7.1

List the laws of boolean algebra that justify the steps in the simplification of the boolean function $f(x_1, x_2, x_3)$ in Example 13.7.1. Some steps use more than one law.

Answer

1. Associative, commutative, and idempotent laws.
2. Distributive law.
3. Idempotent and complement laws.
4. Null and identity laws
5. Distributive law.
6. Null and identity laws.

Exercise 13.7.2

Write the following Boolean expression in the notation of logic design.

$$(x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_2) \vee (\overline{x_1} \wedge x_2)\,.$$

Answer

$$(x_1 \cdot \overline{x_2}) + (x_1 \cdot x_2) + (\overline{x_1} \cdot x_2).$$

Exercise 13.7.3

Find a further simplification of the boolean function in Example 13.7.2, and draw the corresponding gate diagram for the circuit that it realizes.

Answer

A simpler boolean expression for the function is $\overline{x_2} \cdot (x_1 + x_3)$.

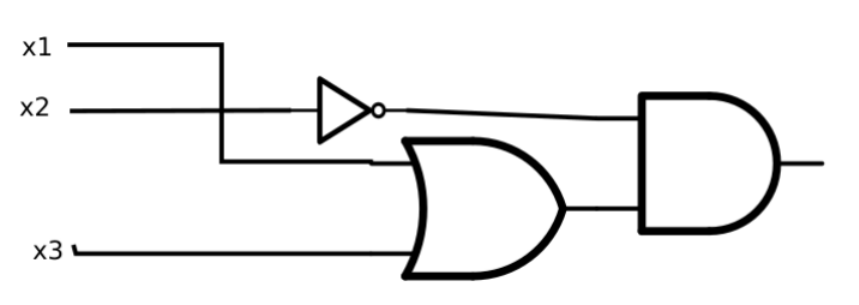


Figure 13.7.17: An even simpler circuit

Exercise 13.7.4

Consider the switching circuit in Figure 13.7.18

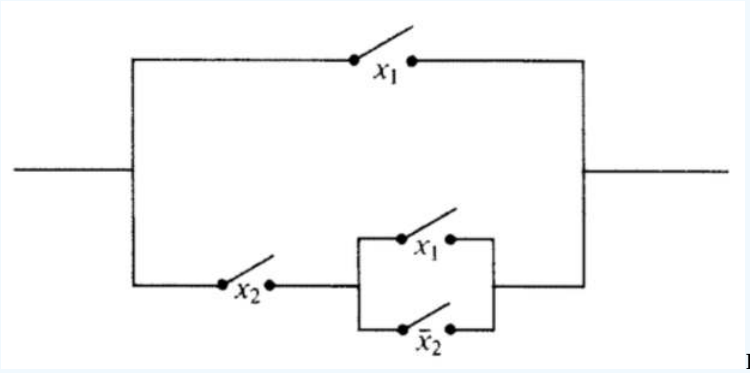


Figure 13.7.18: Can this circuit be simplified?

a. Draw the corresponding gate diagram for this circuit.
b. Construct a table of outputs for each of the eight inputs to this circuit.
c. Determine the minterm normal of the Boolean function based on the table.
d. Simplify the circuit as much as possible.

Exercise 13.7.5

Consider the circuit in Figure 13.7.19

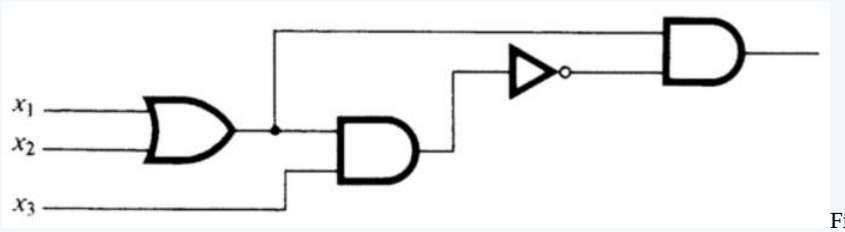


Figure 13.7.19: Can this circuit be simplified?

a. Trace the inputs though this circuit and determine the Boolean function that it realizes.
b. Construct a table of outputs for each of the eight inputs to this circuit.
c. Find the minterm normal form of f.
d. Draw the circuit based on the minterm normal form.
e. Simplify the circuit algebraically and draw the resulting circuit.

Exercise 13.7.6

Consider the Boolean function $f(x_1, x_2, x_3, x_4) = x_1 + (x_2 \cdot (\overline{x_1} + x_4) + x_3 \cdot (\overline{x_2} + \overline{x_4}))$.

a. Simplify f algebraically.
b. Draw the gate diagram based on the simplified version of f.

Exercise 13.7.7

Draw a logic circuit using only AND, OR and NOT gates that realizes an XOR gate.

Exercise 13.7.8

Draw a logic circuit using only AND, OR and NOT gates that realizes the Boolean function on three variables that returns 1 if the majority of inputs are 1 and 0 otherwise.

This page titled 13.7: A Brief Introduction to Switching Theory and Logic Design is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

14: Monoids and Automata

At first glance, the two topics that we will discuss in this chapter seem totally unrelated. The first is monoid theory, which we touched upon in Chapter 11. The second is automata theory, in which computers and other machines are described in abstract terms. After short independent discussions of these topics, we will describe how the two are related in the sense that each monoid can be viewed as a machine and each machine has a monoid associated with it.

14.1: Monoids
14.2: Free Monoids and Languages
14.3: Automata, Finite-State Machines
14.4: The Monoid of a Finite-State Machine
14.5: The Machine of a Monoid

This page titled 14: Monoids and Automata is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

14.1: Monoids

Recall that in Section 11.2 we introduced systems called monoids. Here is the formal definition.

Definition 14.1.1: Monoid

A monoid is a set M together with a binary operation $*$ with the properties

- $*$ is associative: $\forall a, b, c \in M,\ (a * b) * c = a * (b * c)$ and
- $*$ has an identity in M: $\exists e \in M$ such that $\forall a \in M,\ a * e = e * a = a$

Note 14.1.1

Since the requirements for a group contain the requirements for a monoid, every group is a monoid.

Example 14.1.1: Some Monoids

a. The power set of any set together with any one of the operations intersection, union, or symmetric difference is a monoid.
b. The set of integers, $\mathbb{Z}$, with multiplication, is a monoid. With addition, $\mathbb{Z}$ is also a monoid.
c. The set of $n \times n$ matrices over the integers, $M_n(\mathbb{Z})$, $n \geq 2$, with matrix multiplication, is a monoid. This follows from the fact that matrix multiplication is associative and has an identity, I_n. This is an example of a noncommutative monoid since there are matrices, A and B, for which $AB \neq BA$.
d. $[\mathbb{Z}_n; \times_n]$, $n \geqslant 2$, is a monoid with identity 1.
e. Let X be a nonempty set. The set of all functions from X into X, often denoted X^X, is a monoid over function composition. In Chapter 7, we saw that function composition is associative. The function $i : X \to X$ defined by $i(a) = a$ is the identity element for this system. If $|X|$ is greater than 1 then it is a noncommutative monoid. If X is finite, $|X^X| = |X|^{|X|}$. For example, if $B = \{0, 1\}$, $|B^B| = 4$. The functions $z, u, i,$ and t, defined by the graphs in Figure 14.1.1, are the elements of B^B. This monoid is not a group. Do you know why?
One reason why B^B is noncommutative is that $t \circ z \neq z \circ t$ because $(t \circ z)(0) = t(z(0)) = t(0) = 1$ while $(z \circ t)(0) = z(t(0)) = z(1) = 0$.

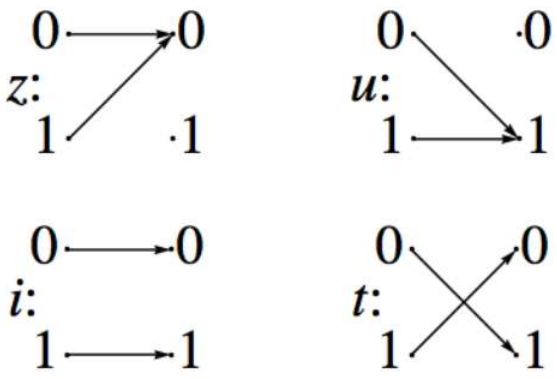


Figure 14.1.1: The functions on B_2

Virtually all of the group concepts that were discussed in Chapter 11 are applicable to monoids. When we introduced subsystems, we saw that a submonoid of monoid M is a subset of M; that is, it is a monoid with the operation of M. To prove that a subset is a submonoid, you can apply the following theorem.

Theorem 14.1.1: Submonoid Test

Assume $[M; *]$ is a monoid and K is a nonempty subset of M. Then K is a submonoid of M if and only if the following two conditions are met.

- If $a, b \in K$, then. $a * b \in K$; i. e., K is closed with under $*$.
- The identity of M belongs to K.

Often we will want to discuss the smallest submonoid that includes a certain subset S of a monoid M. This submonoid can be defined recursively by the following definition.

Definition 14.1.2: Submonoid Generated by a Set

If S is a subset of monoid $[M;*]$, the submonoid generated by S, $\langle S\rangle$, is defined by:.

a. (Basis) The identity of M belongs to $\langle S\rangle$; and $a\in S\Rightarrow a\in\langle S\rangle$.
b. (Recursion) $a,b\in\langle S\rangle\Rightarrow a*b\in\langle S\rangle$.

Note 14.1.2

If $S=\{a_1,a_2,\ldots,a_n\}$, we write $\langle a_1,a_2,\ldots,a_n\rangle$ in place of $\langle\{a_1,a_2,\ldots,a_n\}\rangle$.

Example 14.1.2: Some Submonoids

a. One example of a submonoid of $[\mathbb{Z};+]$ is $\langle 2\rangle=\{0,2,4,6,8,\ldots\}$.
b. The power set of $\mathbb{Z}$, $\mathcal{P}(\mathbb{Z})$, over union is a monoid with identity $\emptyset$. If $S=\{\{1\},\{2\},\{3\}\}$,then $\langle S\rangle$ is the power set of $\{1,2,3\}$.If $S=\{\{n\}:n\in\mathbb{Z}\}$, then $\langle S\rangle$ is the set of finite subsets of the integers.

As you might expect, two monoids are isomorphic if and only if there exists a translation rule between them so that any true proposition in one monoid is translated to a true proposition in the other.

Example 14.1.3

$M=[\mathcal{P}\{1,2,3\};\cap]$ is isomorphic to $M_2=\left[\mathbb{Z}_2^3;\cdot\right]$, where the operation in M_2 is componentwise mod 2 multiplication. A translation rule is that if $A\subseteq\{1,2,3\}$, then it is translated to (d_1,d_2,d_3) where

$$d_i=\begin{cases}1 & \text{if } i\in A\\ 0 & \text{if } i\notin A\end{cases}$$

Two cases of how this translation rule works are:

$$\begin{array}{ccc}\{1,2,3\} & \text{is the identity for } M_1 & \{1,2\}\cap\{2,3\}=\{2\}\\ \updownarrow & & \updownarrow\\ (1,1,1) & \text{is the identity for } M_2 & (1,1,0)\cdot(0,1,1)=(0,1,0)\end{array}.$$

A more precise definition of a monoid isomorphism is identical to the definition of a group isomorphism, Definition 11.7.2.

14.1.1: Exercises

Exercise 14.1.1

For each of the subsets of the indicated monoid, determine whether the subset is a submonoid.

a. $S_1=\{0,2,4,6\}$ and $S_2=\{1,3,5,7\}$ in $[\mathbb{Z}_8;\times_8]$.
b. $\{f\in\mathbb{N}^{\mathbb{N}}:f(n)\leqslant n,\forall n\in\mathbb{N}\}$ and $\left\{f\in\mathbb{N}^{\mathbb{N}}:f(1)=2\right\}$ in the monoid $[\mathbb{N}^{\mathbb{N}};\circ]$.
c. $\{A\subseteq\mathbb{Z}\mid A \text{ is finite}\}$ and $\{A\subseteq\mathbb{Z}\mid A^c \text{ is finite}\}$ in $[\mathcal{P}(\mathbb{Z});\cup]$.

Answer

1. S_1 is not a submonoid since the identity of $[\mathbb{Z}_8;\times_8]$,which is 1, is not in S_1. S_2 is a submonoid since $1\in S_2$ and S_2 is closed under multiplication; that is, for all $a,b\in S_2$, $a\times_8 b$ is in S_2.
2. The identity of $\mathbb{N}^{\mathbb{N}}$ is the identity function $i:\mathbb{N}\to\mathbb{N}$ defined by $i(a)=a,\ \forall a\in\mathbb{N}$. If $a\in\mathbb{N}$, $i(a)=a\leq a$, thus the identity of $\mathbb{N}^{\mathbb{N}}$ is in S_1. However, the image of 1 under any function in S_2 is 2, and thus the identity of $\mathbb{N}^{\mathbb{N}}$ is not in S_2, so S_2 is not a submonoid. The composition of any two functions in S_1, f and g, will be a function in S_1:

$$\begin{aligned}(f \circ g)(n) &= f(g(n)) \leq g(n) \text{ since } f \text{ is in } S_1 \\ &\leq n \text{ since } g \text{ is in } S_1 \Rightarrow f \circ g \in S_1\end{aligned}$$

and the two conditions of a submonoid are satisfied and S_1 is a submonoid of $\mathbb{N}^{\mathbb{N}}$.

3. The first set is a submonoid, but the second is not since the null set has a non-finite complement.

Exercise 14.1.2

For each subset, describe the submonoid that it generates.

a. $\{3\}$ in $[\mathbb{Z}_{12}; \times_{12}]$
b. $\{5\}$ in $[\mathbb{Z}_{25}; \times_{25}]$
c. the set of prime numbers in $[\mathbb{P}; \cdot]$
d. $\{3, 5\}$ in $[\mathbb{N}; +]$

Exercise 14.1.3

$n \times n$ matrix of real numbers is called ***stochastic*** if and only if each entry is nonnegative and the sum of entries in each column is 1. Prove that the set of stochastic matrices is a monoid over matrix multiplication.

Answer

The set of $n \times n$ real matrices is a monoid under matrix multiplication. This follows from the laws of matrix algebra in Chapter 5. To prove that the set of stochastic matrices is a monoid over matrix multiplication, we need only show that the identity matrix is stochastic (this is obvious) and that the set of stochastic matrices is closed under matrix multiplication. Let A and B be $n \times n$ stochastic matrices.

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

The sum of the jth column is

$$\begin{aligned}\sum_{j=1}^{n} (AB)_{ij} &= \sum_{k=1}^{n} a_{1k} b_{kj} + \sum_{k=1}^{n} a_{1k} b_{kj} + \cdots + \sum_{k=1}^{n} a_{nk} b_{kj} \\ &= \sum_{k=1}^{n} \left(a_{1k} b_{kj} + a_{1k} b_{kj} + \cdots + a_{nk} b_{kj}\right) \\ &= \sum_{k=1}^{n} b_{kj} \left(a_{1k} + a_{1k} + \cdots + a_{nk}\right) \\ &= \sum_{k=1}^{n} b_{kj} \quad \text{since } A \text{ is stochastic} \\ &= 1 \quad \text{since } B \text{ is stochastic}\end{aligned}$$

Exercise 14.1.4

A ***semigroup*** is an algebraic system $[S; *]$ with the only axiom that $*$ be associative on S. Prove that if S is a finite set, then there must exist an idempotent element, that is, an $a \in S$ such that $a * a = a$.

Exercise 14.1.5

Let B be a Boolean algebra and M the set of all Boolean functions on B. Let $*$ be defined on M by $(f * g)(a) = f(a) \wedge g(a)$. Prove that $[M; *]$ is a monoid. Construct the operation table of $[M; *]$ for the case of $B = B_2$.

Answer

Let $f,\ g,\ h \in M$, and $a \in B$.

$$\begin{aligned}((f*g)*h)(a) &= (f*g)(a) \wedge h(a) \\ &= (f(a) \wedge g(a)) \wedge h(a) \\ &= f(a) \wedge (g(a) \wedge h(a)) \\ &= f(a) \wedge (g*h)(a) \\ &= (f*(g*h))(a)\end{aligned}$$

Therefore $(f*g)*h = f*(g*h)$ and $*$ is associative.

The identity for $*$ is the function $u \in M$ where $u(a) = 1 =$ the "one" of B. If $a \in B$, $(f*u)(a) = f(a) \wedge u(a) = f(a) \wedge 1 = f(a)$. Therefore $f*u = f$. Similarly, $u*f = f$.

There are $2^2 = 4$ functions in M for $B = B_2$. These four functions are named in the text. See Figure 14.1.1 The table for $*$ is

$*$	z	i	t	u
z	z	z	z	z
i	z	i	z	i
t	z	z	t	t
u	z	i	t	u

This page titled 14.1: Monoids is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

14.2: Free Monoids and Languages

In this section, we will introduce the concept of a language. Languages are subsets of a certain type of monoid, the free monoid over an alphabet. After defining a free monoid, we will discuss languages and some of the basic problems relating to them. We will also discuss the common ways in which languages are defined.

Let A be a nonempty set, which we will call an alphabet. Our primary interest will be in the case where A is finite; however, A could be infinite for most of the situations that we will describe. The elements of A are called letters or symbols. Among the alphabets that we will use are $B = \{0, 1\}$, and the set of ASCII (American Standard Code for Information Interchange) characters, which we symbolize as $ASCII$.

Definition 14.2.1: Strings over an Alphabet

A string of length n, $n \geqslant 1$ over alphabet A is a sequence of n letters from A: $a_1 a_2 \ldots a_n$. The null string, λ, is defined as the string of length zero containing no letters. The set of strings of length n over A is denoted by A^n. The set of all strings over A is denoted A^*.

Note 14.2.1

a. If the length of string s is n, we write $|s| = n$.
b. The null string is not the same as the empty set, although they are similar in many ways. $A^0 = \{\lambda\}$.
c. $A^* = A^0 \cup A^1 \cup A^2 \cup A^3 \cup \cdots$ and if $i \neq j$, $A^i \cap A^j = \emptyset$; that is, $\{A^0, A^1, A^2, A^3, \ldots\}$ is a partition of A^*.
d. An element of A can appear any number of times in a string.

Theorem 14.2.1

If A is countable, then A^* is countable.

Proof

Case 1. Given the alphabet $B = \{0, 1\}$, we can define a bijection from the positive integers into B^*. Each positive integer has a binary expansion $d_k d_{k-1} \cdots d_1 d_0$, where each d_j is 0 or 1 and $d_k = 1$. If n has such a binary expansion, then $2^k \leq n \leq 2^{k+1}$. We define $f : \mathbb{P} \to B^*$ by $f(n) = f(d_k d_{k-1} \cdots d_1 d_0) = d_{k-1} \cdots d_1 d_0$, where $f(1) = \lambda$. Every one of the 2^k strings of length k are the images of exactly one of the integers between 2^k and $2^{k+1} - 1$. From its definition, f is clearly a bijection; therefore, B^* is countable.

Case 2: A is Finite. We will describe how this case is handled with an example first and then give the general proof. If $A = \{a, b, c, d, e\}$, then we can code the letters in A into strings from B^3. One of the coding schemes (there are many) is $a \leftrightarrow 000, b \leftrightarrow 001, c \leftrightarrow 010, d \leftrightarrow 011,$ and $e \leftrightarrow 100$.Now every string in A^* corresponds to a different string in B^*; for example, ace. would correspond with 000010100The cardinality of A^* is equal to the cardinality of the set of strings that can be obtained from this encoding system. The possible coded strings must be countable, since they are a subset of a countable set, B^*. Therefore, A^* is countable.

If $|A| = m$, then the letters in A can be coded using a set of fixed-length strings from B^*. If $2^{k-1} < m \leq 2^k$, then there are at least as many strings of length k in B^k as there are letters in A. Now we can associate each letter in A with with a different element of B^k. Then any string in A^*. corresponds to a string in B^*. By the same reasoning as in the example above, A^* is countable.

Case 3: A is Countably Infinite. We will leave this case as an exercise.

Definition 14.2.2: Concatenation

Let $a = a_1 a_2 \cdots a_m$ and $b = b_1 b_2 \cdots b_n$ be strings of length m and n, respectively. The concatenation of a with b, $a + b$, is the string $a_1 a_2 \cdots a_m b_1 b_2 \cdots b_n$ of length $m + n$.

There are several symbols that are used for concatenation. We chose to use the one that is also used in Python and SageMath.

```
1 'good'+'bye'
```

The set of strings over any alphabet is a monoid under concatenation.

Note 14.2.2

a. The null string is the identity element of $[A^*; +]$. Henceforth, we will denote the monoid of strings over A by A^*.
b. Concatenation is noncommutative, provided $|A| > 1$.
c. If $|A_1| = |A_2|$, then the monoids A_1^* and A_2^* are isomorphic. An isomorphism can be defined using any bijection $f : A_1 \to A_2$. If $a = a_1 a_2 \cdots a_n \in A_1^*$, $f^*(a) = (f(a_1)f(a_2) \cdots f(a_n))$ defines a bijection from A_1^* into A_2^*. We will leave it to the reader to prove that for all $a, b, \in A_1^*, f^*(a+b) = f^*(a) + f^*(b)$.

The languages of the world, English, German, Russian, Chinese, and so forth, are called natural languages. In order to communicate in writing in any one of them, you must first know the letters of the alphabet and then know how to combine the letters in meaningful ways. A formal language is an abstraction of this situation.

Definition 14.2.3: Formal Language

If A is an alphabet, a formal language over A is a subset of A^*.

Example 14.2.1: Some Formal Languages

a. English can be thought of as a language over of letters $A, B, \cdots Z$, both upper and lower case, and other special symbols, such as punctuation marks and the blank. Exactly what subset of the strings over this alphabet defines the English language is difficult to pin down exactly. This is a characteristic of natural languages that we try to avoid with formal languages.
b. The set of all ASCII stream files can be defined in terms of a language over ASCII. An ASCII stream file is a sequence of zero or more lines followed by an end-of-file symbol. A line is defined as a sequence of ASCII characters that ends with the a "new line" character. The end-of-file symbol is system-dependent.
c. The set of all syntactically correct expressions in any computer language is a language over the set of ASCII strings.
d. A few languages over B are
   - $L_1 = \{s \in B^* \mid s \text{ has exactly as many 1's as it has 0's}\}$
   - $L_2 = \{1 + s + 0 \mid s \in B^*\}$
   - $L_3 = \langle 0, 01 \rangle$ = the submonoid of B^* generated by $\{0, 01\}$.

Investigation 14.2.1: Two Fundamental Problems: Recognition and Generation

The generation and recognition problems are basic to computer programming. Given a language, L, the programmer must know how to write (or generate) a syntactically correct program that solves a problem. On the other hand, the compiler must be written to recognize whether a program contains any syntax errors.

Problem 14.2.1: The Recognition Problem

Given a formal language over alphabet A, the Recognition Problem is to design an algorithm that determines the truth of $s \in L$ in a finite number of steps for all $s \in A^*$. Any such algorithm is called a recognition algorithm.

Definition 14.2.4: Recursive Language

A language is recursive if there exists a recognition algorithm for it.

Example 14.2.2: Some Recursive Languages

a. The language of syntactically correct propositions over set of propositional variables expressions is recursive.

b. The three languages in 7(d) are all recursive. Recognition algorithms for L_1 and L_2 should be easy for you to imagine. The reason a recognition algorithm for L_3 might not be obvious is that the definition of L_3 is more cryptic. It doesn't tell us what belongs to L_3, just what can be used to create strings in L_3. This is how many languages are defined. With a second description of L_3, we can easily design a recognition algorithm. You can prove that

$$L_3 = \{s \in B^* \mid s = \lambda \text{ or } s \text{ starts with a 0 and has no consecutive 1's}\}.$$

Problem 14.2.2: The Generation Problem

Design an algorithm that generates or produces any string in L. Here we presume that A is either finite or countably infinite; hence, A^* is countable by Theorem 14.2.1, and $L \subseteq A^*$ must be countable. Therefore, the generation of L amounts to creating a list of strings in L. The list may be either finite or infinite, and you must be able to show that every string in L appears somewhere in the list.

Theorem 14.2.2: Recursive Implies Generating

a. If A is countable, then there exists a generating algorithm for A^*.
b. If L is a recursive language over a countable alphabet, then there exists a generating algorithm for L.

Proof

Part (a) follows from the fact that A^* is countable; therefore, there exists a complete list of strings in A^*.

To generate all strings of L, start with a list of all strings in A^* and an empty list, W, of strings in L. For each string s, use a recognition algorithm (one exists since L is recursive) to determine whether $s \in L$. If $s \in L$, add it to W; otherwise "throw it out." Then go to the next string in the list of A^*.

Example 14.2.3

Since all of the languages in 7(d) are recursive, they must have generating algorithms. The one given in the proof of Theorem 14.2.2 is not usually the most efficient. You could probably design more efficient generating algorithms for L_2 and L_3; however, a better generating algorithm for L_1 is not quite so obvious.

The recognition and generation problems can vary in difficulty depending on how a language is defined and what sort of algorithms we allow ourselves to use. This is not to say that the means by which a language is defined determines whether it is recursive. It just means that the truth of the statement "L is recursive" may be more difficult to determine with one definition than with another. We will close this section with a discussion of grammars, which are standard forms of definition for a language. When we restrict ourselves to only certain types of algorithms, we can affect our ability to determine whether $s \in L$ is true. In defining a recursive language, we do not restrict ourselves in any way in regard to the type of algorithm that will be used. In the next section, we will consider machines called finite automata, which can only perform simple algorithms.

One common way of defining a language is by means of a ***phrase structure grammar*** (or grammar, for short). The set of strings that can be produced using set of grammar rules is called a phrase structure language.

Example 14.2.4: Zeros Before Ones

We can define the set of all strings over B for which all 0's precede all 1's as follows. Define the starting symbol S and establish rules that S can be replaced with any of the following: λ, $0S$, or $S1$. These replacement rules are usually called production rules. They are usually written in the format $S \to \lambda$, $S \to 0S$, and $S \to S1$. Now define L to be the set of all strings that can be produced by starting with S and applying the production rules until S no longer appears. The strings in L are exactly the ones that are described above.

Definition 14.2.5: Phase Structure Grammar

A phrase structure grammar consists of four components:

1. A nonempty finite set of terminal characters, T. If the grammar is defining a language over A, T is a subset of A^*.
2. A finite set of nonterminal characters, N.
3. A starting symbol, $S \in N$.
4. A finite set of production rules, each of the form $X \to Y$, where X and Y are strings over $A \cup N$ such that $X \neq Y$ and X contains at least one nonterminal symbol.

If G is a phrase structure grammar, $L(G)$ is the set of strings that can be produced by starting with S and applying the production rules a finite number of times until no nonterminal characters remain. If a language can be defined by a phrase structure grammar, then it is called a phrase structure language.

Example 14.2.5: Alternating Bits Language

The language over B consisting of strings of alternating 0's and 1's is a phrase structure language. It can be defined by the following grammar:

1. Terminal characters: λ, 0, and 1
2. Nonterminal characters: S, T, and U
3. Starting symbol: S
4. Production rules:

$$\begin{array}{lll} S \to T & S \to U & S \to \lambda \\ S \to 0 & & S \to 1 \\ S \to 0T & & S \to 1U \\ T \to 10T & & T \to 10 \\ U \to 01U & & U \to 01 \end{array}$$

These rules can be visualized with a graph:

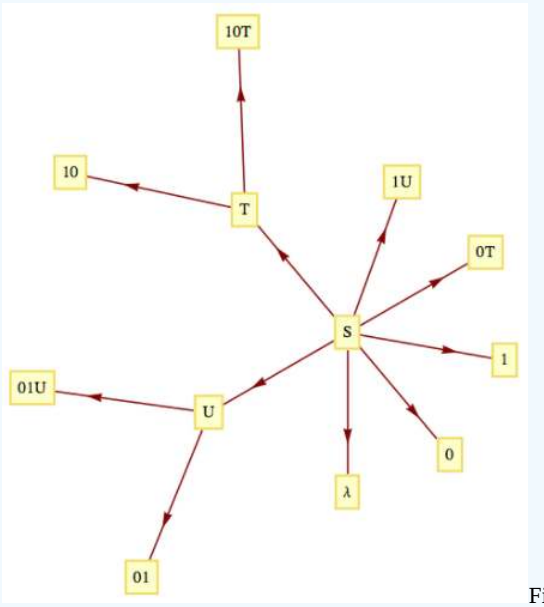


Figure 14.2.1: Production rules for the language of alternating 0's and 1's

We can verify that a string such as 10101 belongs to the language by starting with S and producing 10101 using the production rules a finite number of times: $S \to 1U \to 101U \to 10101$.

Example 14.2.6: Valid SageMath Variables

Let G be the grammar with components:

1. Terminal symbols = all letters of the alphabet (both upper and lower case), digits 0 through 9, and underscore
2. Nonterminal symbols: $\{I, X\}$,
3. Starting symbol: I
4. Production rules: $I \to \alpha$, where α is any letter, $I \to \alpha + X$ for any letter α, $X \to X + \beta$ for any letter, digit or underscore, β, and $X \to \beta$ for any letter, digit or underscore, β. There are a total of $52 + 52 + 63 + 63 = 230$ production rules for this grammar. The language $L(G)$ consists of all valid SageMath variable names.

Example 14.2.7: Backus-Naur Form

Backus-Naur form (BNF) is a popular alternate form of defining the production rules in a grammar. If the production rules $A \to B_1, A \to B_2, \ldots A \to B_n$ are part of a grammar, they would be written in BNF as $A ::= B_1 \mid B_2 \mid \cdots \mid B_n$. The symbol $\mid$ in BNF is read as "or" while the $::=$ is read as "is defined as." Additional notations of BNF are that $\{x\}$, represents zero or more repetitions of x and $[y]$ means that y is optional.

A BNF version of the production rules for a SageMath variable, I, is

$$\begin{aligned} letter &::= a \mid b \mid c \mid \cdots \mid z \mid A \mid B \mid \cdots \mid Z \\ digit &::= 0 \mid 1 \mid \cdots \mid 9 \\ I &::= letter\{letter \mid digit \mid \_\} \end{aligned}$$

Example 14.2.8: The Language of Simple Arithmetic Expressions

An arithmetic expression can be defined in BNF. For simplicity, we will consider only expressions obtained using addition and multiplication of integers. The terminal symbols are (,) , + , * , - , and the digits 0 through 9. The nonterminal symbols are E (for expression), T (term), F (factor), and N (number). The starting symbol is E. Production rules are

$$\begin{aligned} E &::= E + T \mid T \\ T &::= T * F \mid F \\ F &::= (E) \mid N \\ N &::= [-]\text{digit}\{\text{digit}\} \end{aligned} \quad .$$

One particularly simple type of phrase structure grammar is the regular grammar.

Definition 14.2.6: Regular Grammar

A regular (right-hand form) grammar is a grammar whose production rules are all of the form $A \to t$ and $A \to tB$, where A and B are nonterminal and t is terminal. A left-hand form grammar allows only $A \to t$ and $A \to Bt$. A language that has a regular phrase structure language is called a regular language.

Example 14.2.9

a. The set of Sage variable names is a regular language since the grammar by which we defined the set is a regular grammar.
b. The language of all strings for which all 0's precede all 1's (Example 14.2.4) is regular; however, the grammar by which we defined this set is not regular. Can you define these strings with a regular grammar?
c. The language of arithmetic expressions is not regular.

14.2.1: Exercises

Exercise 14.2.1

a. If a computer is being designed to operate with a character set of 350 symbols, how many bits must be reserved for each character? Assume each character will use the same number of bits.
b. Do the same for 3,500 symbols.

Answer

a. For a character set of 350 symbols, the number of bits needed for each character is the smallest n such that 2^n is greater than or equal to 350. Since $2^9 = 512 > 350 > 2^8$, 9 bits are needed,
b. $2^{12} = 4096 > 3500 > 2^{11}$; therefore, 12 bits are needed.

Exercise 14.2.2

It was pointed out in the text that the null string and the null set are different. The former is a string and the latter is a set, two different kinds of objects. Discuss how the two are similar.

Exercise 14.2.3

What sets of strings are defined by the following grammar?

a. Terminal symbols: λ , 0 and 1
b. Nonterminal symbols: S and E
c. Starting symbol: S
d. Production rules: $S \to 0S0, S \to 1S1, S \to E, E \to \lambda, E \to 0, E \to 1$

Answer

This grammar defines the set of all strings over B for which each string is a palindrome (same string if read forward or backward).

Exercise 14.2.4

What sets of strings are defined by the following grammar?

a. Terminal symbols: λ, a, b, and c
b. Nonterminal symbols: S, T, U and E
c. Starting symbol: S
d. Production rules:

$$\begin{array}{ccc} S \to aS & S \to T & T \to bT \\ T \to U & U \to cU & U \to E \\ & E \to \lambda & \end{array}$$

Exercise 14.2.5

Define the following languages over B with phrase structure grammars. Which of these languages are regular?

a. The strings with an odd number of characters.
b. The strings of length 4 or less.
c. The palindromes, strings that are the same backwards as forwards.

Answer

a. Terminal symbols: The null string, 0, and 1. Nonterminal symbols: S, E. Starting symbol: S. Production rules: $S \to 00S$, $S \to 01S$, $S \to 10S$, $S \to 11S$, $S \to E$, $E \to 0$, $E \to 1$ This is a regular grammar.

b. Terminal symbols: The null string, 0, and 1. Nonterminal symbols: S, A, B, C Starting symbol: S Production rules: $S \to 0A$, $S \to 1A$, $S \to \lambda$, $A \to 0B$, $A \to 1B$, $A \to \lambda$, $B \to 0C$, $B \to 1C$, $B \to A$, $C \to 0$, $C \to 1$, $C \to \lambda$ This is a regular grammar.
c. See Exercise 14.2.3 This language is not regular.

Exercise 14.2.6

Define the following languages over B with phrase structure grammars. Which of these languages are regular?

a. The strings with more 0's than 1's.
b. The strings with an even number of 1's.
c. The strings for which all 0's precede all 1's.

Exercise 14.2.7

Prove that if a language over A is recursive, then its complement is also recursive.

Answer

If s is in A^* and L is recursive, we can answer the question "Is s in L^c?" by negating the answer to "Is s in L?"

Exercise 14.2.8

Use BNF to define the grammars in Exercises 14.2.3and 14.2.4.

Exercise 14.2.9

a. Prove that if $X_1, X_2, \ldots$is a countable sequence of countable sets, the union of these sets, $\bigcup_{i=1}^{\infty} X_i$ is countable.
b. Using the fact that the countable union of countable sets is countable, prove that if A is countable, then A^* is countable.

Answer

a. List the elements of each set X_i in a sequence x_{i1}, x_{i2}, $x_{i3} \cdots$. Then draw arrows as shown below and list the elements of the union in order established by this pattern: x_{11}, x_{21}, x_{12}, x_{13}, x_{22}, x_{31}, x_{41}, x_{32}, x_{23}, x_{14}, $x_{15} \cdots$,
b. Each of the sets A^1, A^2, $A^3, \cdots$, are countable and A^* is the union of these sets; hence A^* is countable.

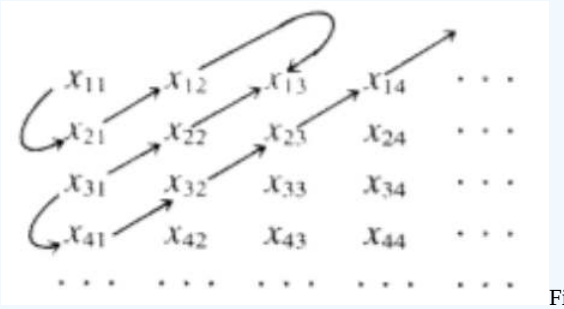


Figure 14.2.2: Exercise 14.2.9

This page titled 14.2: Free Monoids and Languages is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

14.3: Automata, Finite-State Machines

In this section, we will introduce the concept of an abstract machine. The machines we will examine will (in theory) be capable of performing many of the tasks associated with digital computers. One such task is solving the recognition problem for a language. We will concentrate on one class of machines, finite-state machines (finite automata). And we will see that they are precisely the machines that are capable of recognizing strings in a regular grammar.

Given an alphabet X, we will imagine a string in X^* to be encoded on a tape that we will call an input tape. When we refer to a tape, we might imagine a strip of material that is divided into segments, each of which can contain either a letter or a blank.

The typical abstract machine includes an input device, the read head, which is capable of reading the symbol from the segment of the input tape that is currently in the read head. Some more advanced machines have a read/write head that can also write symbols onto the tape. The movement of the input tape after reading a symbol depends on the machine. With a finite-state machine, the next segment of the input tape is always moved into the read head after a symbol has been read. Most machines (including finite-state machines) also have a separate output tape that is written on with a write head. The output symbols come from an output alphabet, Z, that may or may not be equal to the input alphabet. The most significant component of an abstract machine is its memory structure. This structure can range from a finite number of bits of memory (as in a finite-state machine) to an infinite amount of memory that can be stored in the form of a tape that can be read from and written on (as in a Turing machine).

Definition 14.3.1: Finite-State Machine

A finite-state machine is defined by a quintet (S, X, Z, w, t) where

1. $S = \{s_1, s_2, \ldots, s_r\}$ is the state set, a finite set that corresponds to the set of memory configurations that the machine can have at any time.
2. $X = \{x_1, x_2, \ldots, x_m\}$ is the input alphabet.
3. $Z = \{z_1, z_2, \ldots, z_n\}$ is the output alphabet.
4. $w : X \times S \to Z$ is the output function, which specifies which output symbol $w(x, s) \in Z$ is written onto the output tape when the machine is in state s and the input symbol x is read.
5. $t : X \times S \to S$ is the next-state (or transition) function, which specifies which state $t(x, s) \in S$ the machine should enter when it is in state s and it reads the symbol x.

Example 14.3.1: Vending Machine as a Finite-State Machine

Many mechanical devices, such as simple vending machines, can be thought of as finite-state machines. For simplicity, assume that a vending machine dispenses packets of gum, spearmint (S), peppermint (P), and bubble (B), for 25 cents each. We can define the input alphabet to be

$$\{\text{deposit 25 cents, press S, press P, press B}\}$$

and the state set to be $\{\text{Locked, Select}\}$, where the deposit of a quarter unlocks the release mechanism of the machine and allows you to select a flavor of gum. We will leave it to the reader to imagine what the output alphabet, output function, and next-state function would be. You are also invited to let your imagination run wild and include such features as a coin-return lever and change maker.

Example 14.3.2: A Parity Checking Machine

The following machine is called a parity checker. It recognizes whether or not a string in B^* contains an even number of 1s. The memory structure of this machine reflects the fact that in order to check the parity of a string, we need only keep track of whether an odd or even number of 1's has been detected.

The input alphabet is $B = \{0, 1\}$ and the output alphabet is also B. The state set is $\{even, odd\}$. The following table defines the output and next-state functions.

x	s	$w(x,s)$	$t(x,s)$
0	even	0	even
0	odd	1	odd
1	even	1	odd
1	odd	0	even

Note how the value of the most recent output at any time is an indication of the current state of the machine. Therefore, if we start in the even state and read any finite input tape, the last output corresponds to the final state of the parity checker and tells us the parity of the string on the input tape. For example, if the string 11001010 is read from left to right, the output tape, also from left to right, will be 10001100. Since the last character is a 0, we know that the input string has even parity.

An alternate method for defining a finite-state machine is with a transition diagram. A transition diagram is a directed graph that contains a node for each state and edges that indicate the transition and output functions. An edge (s_i, s_j) that is labeled x/z indicates that in state s_i the input x results in an output of z and the next state is s_j. That is, $w(x, s_i) = z$ and $t(x, s_i) = s_j$. The transition diagram for the parity checker appears in Figure 14.3.1. In later examples, we will see that if there are different inputs, x_i and x_j, while in the same state resulting in the same transitions and outputs, we label a single edge $x_i, x_j / z$ instead of drawing two edges with labels x_i / z and x_j / z.

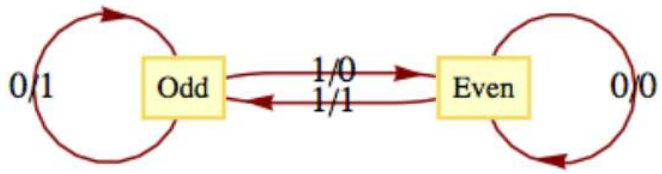


Figure 14.3.1: Transition Diagram for a Parity Checker

One of the most significant features of a finite-state machine is that it retains no information about its past states that can be accessed by the machine itself. For example, after we input a tape encoded with the symbols 01101010 into the parity checker, the current state will be even, but we have no indication within the machine whether or not it has always been in even state. Note how the output tape is not considered part of the machine's memory. In this case, the output tape does contain a "history" of the parity checker's past states. We assume that the finite-state machine has no way of recovering the output sequence for later use.

Example 14.3.3: A Baseball Machine

Consider the following simplified version of the game of baseball. To be precise, this machine describes one half-inning of a simplified baseball game. Suppose that in addition to home plate, there is only one base instead of the usual three bases. Also, assume that there are only two outs per inning instead of the usual three. Our input alphabet will consist of the types of hits that the batter could have: out (O), double play (DP), single (S), and home run (HR). The input DP is meant to represent a batted ball that would result in a double play (two outs), if possible. The input DP can then occur at any time. The output alphabet is the numbers 0, 1, and 2 for the number of runs that can be scored as a result of any input. The state set contains the current situation in the inning, the number of outs, and whether a base runner is currently on the base. The list of possible states is then 00 (for 0 outs and 0 runners), 01, 10, 11, and end (when the half-inning is over). The transition diagram for this machine appears in Figure 14.3.2

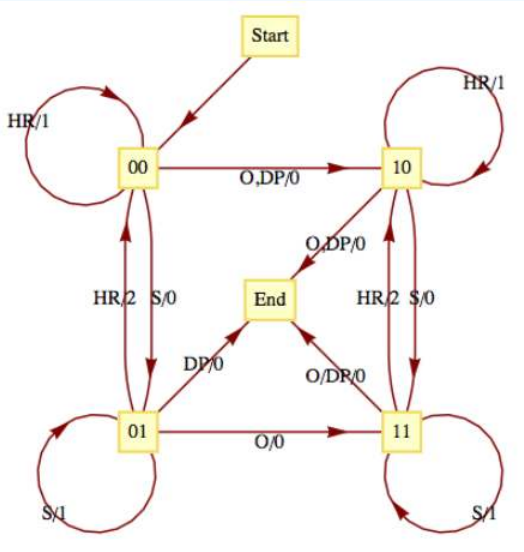


Figure 14.3.2: Transition Diagram for a Simplified Game of Baseball

Let's concentrate on one state. If the current state is 01, 0 outs and 1 runner on base, each input results in a different combination of output and next-state. If the batter hits the ball poorly (a double play) the output is zero runs and the inning is over (the limit of two outs has been made). A simple out also results in an output of 0 runs and the next state is 11, one out and one runner on base. If the batter hits a single, one run scores (output = 1) while the state remains 01. If a home run is hit, two runs are scored (output = 2) and the next state is 00. If we had allowed three outs per inning, this graph would only be marginally more complicated. The usual game with three bases would be quite a bit more complicated, however.

Example 14.3.4: Recognition in Regular Languages

As we mentioned at the outset of this section, finite-state machines can recognize strings in a regular language. Consider the language L over $\{a, b, c\}$ that contains the strings of positive length in which each a is followed by b and each b is followed by c. One such string is $bccabcbc$. This language is regular. A grammar for the language would be nonterminal symbols $\{A, B, C\}$ with starting symbol C and production rules $A \to bB$, $B \to cC$, $C \to aA$, $C \to bB$, $C \to cC$, $C \to c$. A finite-state machine (Figure 14.3.3) that recognizes this language can be constructed with one state for each nonterminal symbol and an additional state (Reject) that is entered if any invalid production takes place. At the end of an input tape that encodes a string in $\{a, b, c\}^*$, we will know when the string belongs to L based on the final output. If the final output is 1, the string belongs to L and if it is 0, the string does not belong to L. In addition, recognition can be accomplished by examining the final state of the machine. The input string belongs to the language if and only if the final state is C.

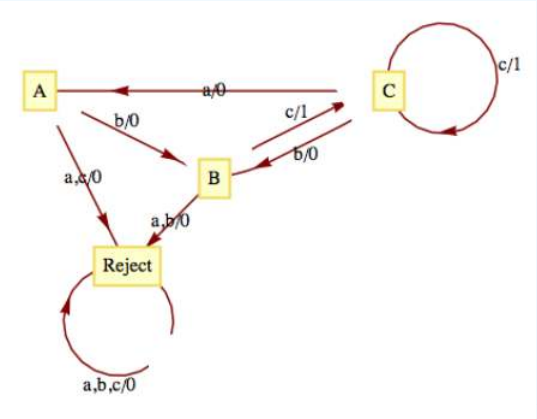


Figure 14.3.3

The construction of this machine is quite easy: note how each production rule translates into an edge between states other than Reject. For example, $C \to bB$ indicates that in State C, an input of b places the machine into State B. Not all sets of production rules can be as easily translated to a finite-state machine. Another set of production rules for L is $A \to aB$, $B \to bC$, $C \to cA$, $C \to cB$, $C \to cC$ and $C \to c$. Techniques for constructing finite-state machines from production rules is not our objective here. Hence we will only expect you to experiment with production rules until appropriate ones are found.

Example 14.3.5: A Binary Adder

A finite-state machine can be designed to add positive integers of any size. Given two integers in binary form, $a = a_n a_{n-1} \cdots a_1 a_0$ and $b = b_n b_{n-1} \cdots b_1 b_0$, the machine take as its input sequence the corresponding bits of a and b reading from right to left with a "parity bit" added

$$a_0 b_0 \left(a_0 +_2 b_0\right), a_1 b_1 \left(a_1 +_2 b_1\right) \ldots, a_n b_n \left(a_n +_2 b_n\right), 111$$

Notice the special input 111 at the end. All possible inputs except the last one must even parity (contain an even number of ones). The output sequence is the sum of a and b, starting with the units digit, and comes from the set $\{0, 1, \lambda\}$. The transition diagram for this machine appears in Figure 14.3.4

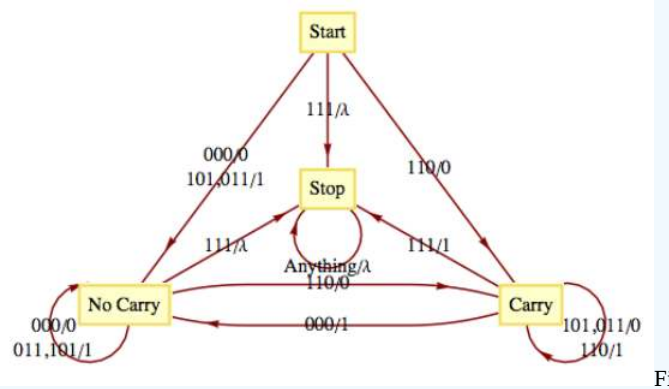


Figure 14.3.4: Transition Diagram for a binary adder

14.3.1: 14.3.1: Exercises

Exercise 14.3.1

Draw a transition diagram for the vending machine described in Example 14.3.1.

Answer

x	s	$Z(x,s)$	$t(x,s)$
Deposit25 ¢	Locked	Nothing	Select
Deposit25 ¢	Select	Return25 ¢	Select
Press$S$	Locked	Nothing	Locked
Press$S$	Select	Dispense$S$	Locked
Press$P$	Locked	Nothing	Locked
Press$P$	Select	Dispense$P$	Locked
Press$B$	Locked	Nothing	Locked
Press$B$	Select	Dispense$B$	Locked

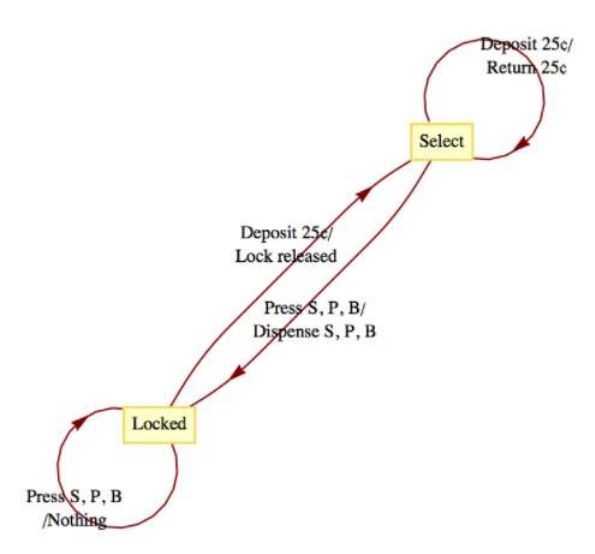


Figure 14.3.5: Vending Machine Transitions

Exercise 14.3.2

Construct finite-state machines that recognize the regular languages that you identified in Section 14.2.

Exercise 14.3.3

What is the input set for the binary adding machine in Example 14.3.5?

Answer

$\{000, 011, 101, 110, 111\}$

Exercise 14.3.4

What input sequence would be used to compute the sum of 1101 and 0111 (binary integers)? What would the output sequence be?

Exercise 14.3.5

The Gray Code Decoder. The finite-state machine defined by the following figure has an interesting connection with the Gray Code.

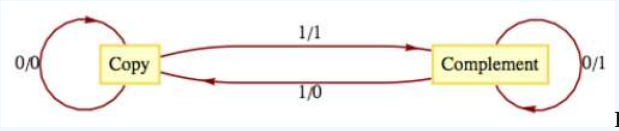


Figure 14.3.6: Gray Code Decoder

Given a string $x = x_1x_2 \cdots x_n \in B^n$, we may ask where x appears in G_n. Starting in Copy state, the input string x will result in an output string $z \in B^n$, which is the binary form of the position of x in G_n. Recall that positions are numbered from 0 to $2^n - 1$.

a. In what positions $(0 - 31)$ do 10110, 00100, and 11111 appear in G_5?
b. Prove that the Gray Code Decoder always works.

Answer

a.

- Input: 10110, Output: 11011 $\Rightarrow$ 10110 is in position 27
- Input: 00100, Output: 00111 $\Rightarrow$ 00100 is in position 7
- Input:11111, Output: 10101 $\Rightarrow$ 11111 is in position 21

b. Let $x = x_1x_2 \ldots x_n$ and recall that for $n \geq 1$, $G_{n+1} = \begin{pmatrix} 0G_n \\ 1G_n^r \end{pmatrix}$, where G_n^r is the reverse of G_n. To prove that the Gray Code Decoder always works, let $p(n)$ be the proposition "Starting in Copy state, x's output is the position of x in G_n; and starting in Complement state, x's output is the position of x in G_n^r." That p(1) is true is easy to verify for both possible values of x, 0 and 1. Now assume that for some $n \geq 1$, $p(n)$ is true and consider $x = x_1x_2 \ldots x_nx_{n+1}$.
If $x_1 = 0$, x's output is a zero followed by the output for $(x_2 \ldots x_nx_{n+1})$ starting in Copy state. By the induction hypothesis, this is zero followed by the position of $(x_2 \ldots x_nx_{n+1})$ in G_n, which is the position of x in G_{n+1}, by the definition of G.
If $x_1 = 1$, x's output is a one followed by the output for $(x_2 \ldots x_nx_{n+1})$ starting in Complement state. By the induction hypothesis, this is one followed by the position of $(x_2 \ldots x_nx_{n+1})$ in G_n^r, which is the position of x in G_{n+1}, by the definition of G. □

This page titled 14.3: Automata, Finite-State Machines is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

14.4: The Monoid of a Finite-State Machine

In this section, we will see how every finite-state machine has a monoid associated with it. For any finite-state machine, the elements of its associated monoid correspond to certain input sequences. Because only a finite number of combinations of states and inputs is possible for a finite-state machine there is only a finite number of input sequences that summarize the machine. This idea is illustrated best with a few examples.

Consider the parity checker. The following table summarizes the effect on the parity checker of strings in B^1 and B^2. The row labeled "Even" contains the final state and final output as a result of each input string in B^1 and B^2 when the machine starts in the even state. Similarly, the row labeled "Odd" contains the same information for input sequences when the machine starts in the odd state.

Input String	0	1	00	01	10	11
Even	$(\text{Even}, 0)$	$(\text{Odd}, 1)$	$(\text{Even}, 0)$	$(\text{Odd}, 1)$	$(\text{Odd}, 1)$	$(\text{Even}, 0)$
Odd	$(\text{Odd}, 1)$	$(\text{Even}, 1)$	$(\text{Odd}, 1)$	$(\text{Even}, 1)$	$(\text{Even}, 0)$	$(\text{Odd}, 1)$
Same Effect as			0	1	1	0

Note how, as indicated in the last row, the strings in B^2 have the same effect as certain strings in B^1. For this reason, we can summarize the machine in terms of how it is affected by strings of length 1. The actual monoid that we will now describe consists of a set of functions, and the operation on the functions will be based on the concatenation operation.

Let T_0 be the final effect (state and output) on the parity checker of the input 0. Similarly, T_1 is defined as the final effect on the parity checker of the input 1. More precisely,

$$T_0(\text{even}) = (\text{even}, 0) \quad \text{and} \quad T_0(\text{odd}) = (\text{odd}, 1),$$

while

$$T_1(\text{even}) = (\text{odd}, 1) \quad \text{and} \quad T_1(\text{odd}) = (\text{even}, 0).$$

In general, we define the operation on a set of such functions as follows: if s, t are input sequences and T_s and T_t, are functions as above, then $T_s * T_t = T_{st}$, that is, the result of the function that summarizes the effect on the machine by the concatenation of s with t. Since, for example, 01 has the same effect on the parity checker as 1, $T_0 * T_1 = T_{01} = T_1$. We don't stop our calculation at T_{01} because we want to use the shortest string of inputs to describe the final result. A complete table for the monoid of the parity checker is

$*$	T_0	T_1
T_0	T_0	T_1
T_1	T_1	T_0

What is the identity of this monoid? The monoid of the parity checker is isomorphic to the monoid $[\mathbb{Z}_2; +_2]$.

This operation may remind you of the composition operation on functions, but there are two principal differences. The domain of T_s is not the codomain of T_t and the functions are read from left to right unlike in composition, where they are normally read from right to left.

You may have noticed that the output of the parity checker echoes the state of the machine and that we could have looked only at the effect on the machine as the final state. The following example has the same property, hence we will only consider the final state.

Example 14.4.1

The transition diagram for the machine that recognizes strings in $B*$ that have no consecutive 1's appears in Figure 14.4.1. Note how it is similar to the graph in Figure 9.1.1. Only a "reject state" has been added, for the case when an input of 1 occurs while in State a. We construct a similar table to the one in the previous example to study the effect of certain strings on this machine. This time, we must include strings of length 3 before we recognize that no "new effects" can be found.

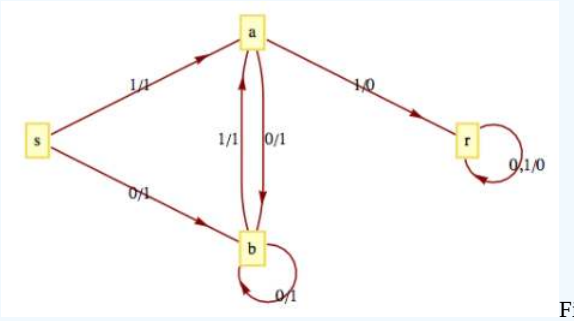


Figure 14.4.1: No Consecutive Ones Monoid

Inputs	0	1	00	01	10	11	000	001	010	011	100	101	110	111
s	b	a	b	a	b	r	b	a	b	r	b	a	r	r
a	b	r	b	a	r	r	b	a	b	r	r	r	r	r
b	b	a	b	a	b	r	b	a	b	r	b	a	r	r
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Same as			0				0	01	0	11	10	1	11	11

The following table summarizes how combinations of the strings $0, 1, 01, 10,$ and 11 affect this machine.

$*$	T_0	T_1	T_{01}	T_{10}	T_{11}
T_0	T_0	T_1	T_{01}	T_{10}	T_{11}
T_1	T_{10}	T_{11}	T_1	T_{11}	T_{11}
T_{01}	T_0	T_{11}	T_{01}	T_{11}	T_{11}
T_{10}	T_{10}	T_1	T_1	T_{10}	T_{11}
T_{11}	T_{11}	T_{11}	T_{11}	T_{11}	T_{11}

All the results in this table can be obtained using the previous table. For example,

$$T_{10} * T_{01} = T_{1001} = T_{100} * T_1 = T_{10} * T_1 = T_{101} = T_1$$

and

$$T_{01} * T_{01} = T_{0101} = T_{010} T_1 = T_0 T_1 = T_{01}$$

Note that none of the elements that we have listed in this table serves as the identity for our operation. This problem can always be remedied by including the function that corresponds to the input of the null string, T_λ. Since the null string is the identity for concatenation of strings, $T_s T_\lambda = T_\lambda T_s = T_s$ for all input strings s.

Example 14.4.2: The Unit-Time Delay Machine

A finite-state machine called the unit-time delay machine does not echo its current state, but prints its previous state. For this reason, when we find the monoid of the unit-time delay machine, we must consider both state and output. The transition diagram of this machine appears in Figure 14.4.2

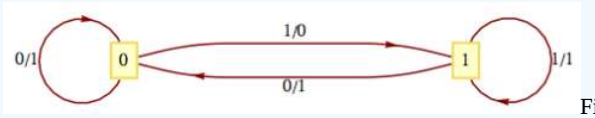


Figure 14.4.2

Input	0	1	00	01	10	11	100 or000	101 or001	110 or101	111 or011
0	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
1	$(0,1)$	$(1,1)$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
Same as							00	01	10	11

https://math.libretexts.org/@go/page/80567

Again, since no new outcomes were obtained from strings of length 3, only strings of length 2 or less contribute to the monoid of the machine. The table for the strings of positive length shows that we must add T_λ to obtain a monoid.

$*$	T_0	T_1	T_{00}	T_{01}	T_{10}	T_{11}
T_0	T_{00}	T_{01}	T_{00}	T_{01}	T_{10}	T_{11}
T_1	T_{10}	T_{11}	T_{00}	T_{01}	T_{10}	T_{11}
T_{00}	T_{00}	T_{01}	T_{00}	T_{01}	T_{10}	T_{11}
T_{01}	T_{10}	T_{11}	T_{00}	T_{01}	T_{10}	T_{11}
T_{10}	T_{00}	T_{01}	T_{00}	T_{01}	T_{10}	T_{11}
T_{11}	T_{10}	T_{11}	T_{00}	T_{01}	T_{10}	T_{11}

.

14.4.1: 14.4.1: Exercises

Exercise 14.4.1

For each of the transition diagrams in Figure 14.4.3, write out tables for their associated monoids. Identify the identity in terms of a string of positive length, if possible.

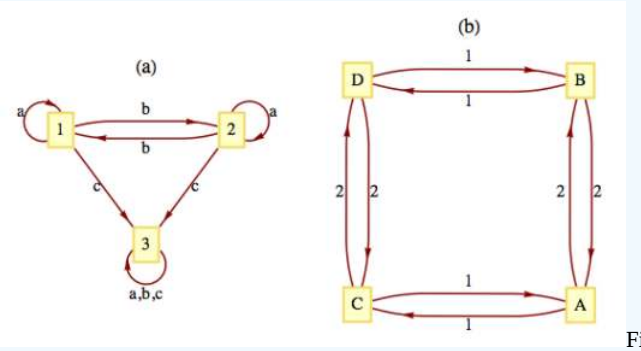


Figure 14.4.3: Exercise 14.4.3

Hint

Where the output echoes the current state, the output can be ignored.

Answer

a.

Input String	a	b	c	aa	ab	ac
1	$(a,1)$	$(a,2)$	$(c,3)$	$(a,1)$	$(a,2)$	$(c,3)$
2	$(a,2)$	$(a,1)$	$(c,3)$	$(a,2)$	$(a,1)$	$(c,3)$
3	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$
Input String	ba	bb	bc	ca	cb	cc
1	$(a,2)$	$(a,1)$	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$
2	$(a,1)$	$(a,2)$	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$
3	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$	$(c,3)$

We can see that $T_aT_a = T_{\text{aa}} = T_a$, $T_aT_b = T_{\text{ab}} = T_b$, etc. Therefore, we have the following monoid:

	T_a	T_b	T_b
T_a	T_a	T_b	T_c
T_b	T_b	T_a	T_c
T_c	T_c	T_c	T_c

Notice that T_a is the identity of this monoid.

b.

Input String	1	2	11	12	21	22
A	C	B	A	D	D	A
B	D	A	B	C	C	B
C	A	D	C	B	B	C
D	B	C	D	A	A	D

Input String	111	112	121	122	211	212	221	222
A	C	B	B	C	B	C	C	B
B	D	A	A	D	A	D	D	A
C	B	C	C	B	C	B	B	C
D	B	C	C	B	C	B	B	C

We have the following monoid:

	T_1	T_2	T_{11}	T_{12}
T_1	T_{11}	T_{12}	T_1	T_2
T_2	T_b	T_{11}	T_2	T_1
T_{11}	T_1	T_2	T_{11}	T_{12}
T_{12}	T_2	T_1	T_{12}	T_{11}

Notice that T_{11} is the identity of this monoid.

Exercise 14.4.2

What common monoids are isomorphic to the monoids obtained in the previous exercise?

Exercise 14.4.3

Can two finite-state machines with nonisomorphic transition diagrams have isomorphic monoids?

Answer

Yes, just consider the unit time delay machine of Figure 14.4.2 Its monoid is described by the table at the end of Section 14.4 where the T_λ row and T_λ column are omitted. Next consider the machine in Figure 14.5.3. The monoid of this machine is:

	T_λ	T_0	T_1	T_{00}	T_{01}	T_{10}	T_{11}
T_λ	T_λ	T_0	T_1	T_{00}	T_{01}	T_{10}	T_{11}
T_0	T_0	T_{00}	T_{01}	T_{00}	T_{01}	T_{10}	T_{11}
T_1	T_1	T_{10}	T_{11}	T_{00}	T_{01}	T_{10}	T_{11}
T_{00}	T_{00}	T_{00}	T_{01}	T_{00}	T_{01}	T_{10}	T_{11}
T_{01}	T_{01}	T_{10}	T_{11}	T_{00}	T_{01}	T_{10}	T_{11}
T_{10}	T_{10}	T_{00}	T_{01}	T_{00}	T_{01}	T_{10}	T_{11}
T_{11}	T_{11}	T_{10}	T_{11}	T_{00}	T_{01}	T_{10}	T_{11}

Hence both of these machines have the same monoid, however, their transition diagrams are nonisomorphic since the first has two vertices and the second has seven.

This page titled 14.4: The Monoid of a Finite-State Machine is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

14.5: The Machine of a Monoid

Any finite monoid $[M; *]$ can be represented in the form of a finite-state machine with input and state sets equal to M. The output of the machine will be ignored here, since it would echo the current state of the machine. Machines of this type are called ***state machines***. It can be shown that whatever can be done with a finite-state machine can be done with a state machine; however, there is a trade-off. Usually, state machines that perform a specific function are more complex than general finite-state machines.

Definition 14.5.1: Machine of a Monoid

If $[M; *]$ is a finite monoid, then the machine of M, denoted $m(M)$, is the state machine with state set M, input set M, and next-state function $t : M \times M \to M$ defined by $t(s, x) = s * x$.

Example 14.5.1

We will construct the machine of the monoid $[\mathbb{Z}_2; +_2]$. As mentioned above, the state set and the input set are both $\mathbb{Z}_2$. The next state function is defined by $t(s, x) = s +_2 x$. The transition diagram for $m(\mathbb{Z}_2)$ appears in Figure 14.5.1. Note how it is identical to the transition diagram of the parity checker, which has an associated monoid that was isomorphic to $[\mathbb{Z}_2; +_2]$.

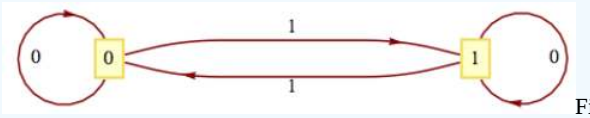


Figure 14.5.1: The machine of $[\mathbb{Z}_2; +_2]$

Example 14.5.2

The transition diagram of the monoids $[\mathbb{Z}_2; \times_2]$ and $[\mathbb{Z}_3; \times_3]$ appear in Figure 14.5.2

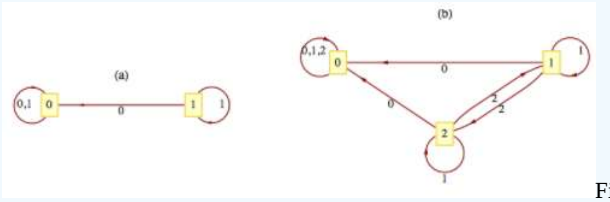


Figure 14.5.2: The machines of $[\mathbb{Z}_2; \times_2]$ and $\mathbb{Z}_3; \times_3]$

Example 14.5.3

Let U be the monoid that we obtained from the unit-time delay machine (Example 14.4.2). We have seen that the machine of the monoid of the parity checker is essentially the parity checker. Will we obtain a unit-time delay machine when we construct the machine of U? We can't expect to get exactly the same machine because the unit-time delay machine is not a state machine and the machine of a monoid is a state machine. However, we will see that our new machine is capable of telling us what input was received in the previous time period. The operation table for the monoid serves as a table to define the transition function for the machine. The row headings are the state values, while the column headings are the inputs. If we were to draw a transition diagram with all possible inputs, the diagram would be too difficult to read. Since U is generated by the two elements, T_0 and T_1, we will include only those inputs. Suppose that we wanted to read the transition function for the input T_{01}. Since $T_{01} = T_0 T_1$, in any state $s, t(s, T_{01}) = t(t(s, T_0), T_1)$. The transition diagram appears in Figure 14.5.3.

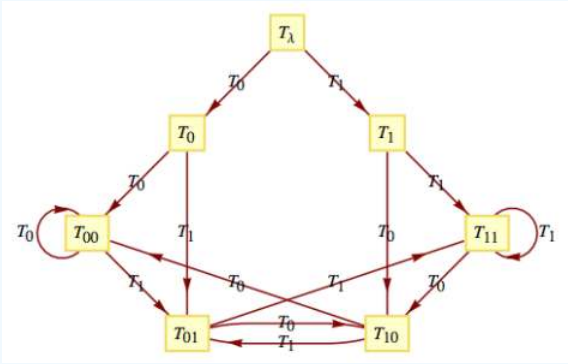


Figure 14.5.3: Copy and Paste Caption here. (Copyright; author via source)

If we start reading a string of 0's and 1's while in state T_λ and are in state T_{ab} at any one time, the input from the previous time period (not the input that sent us into T_{ab}, the one before that) is a. In states T_λ, T_0 and T_1, no previous input exists.

14.5.1: 14.5.1: Exercises

Exercise 14.5.1

Draw the transition diagrams for the machines of the following monoids:

a. $[\mathbb{Z}_4; +_4]$
b. The direct product of $[\mathbb{Z}_2; \times_2]$ with itself.

Answer

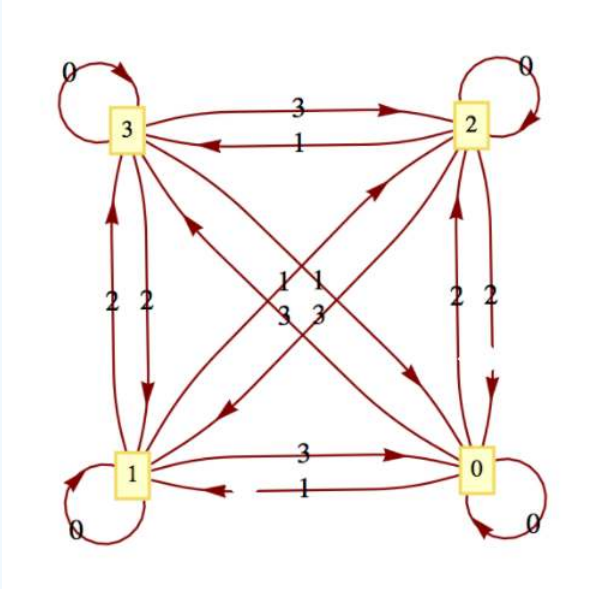


Figure 14.5.4: (a)

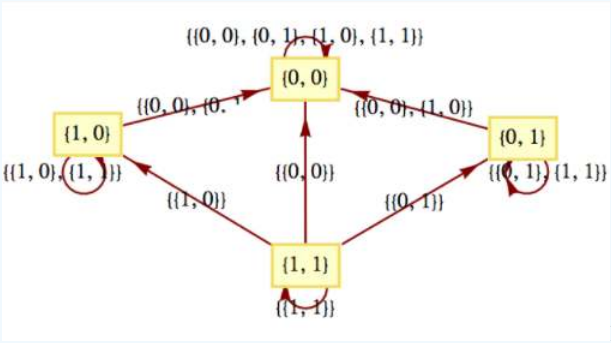


Figure 14.5.5: (b)

Exercise 14.5.2

Even though a monoid may be infinite, we can visualize it as an infinite-state machine provided that it is generated by a finite number of elements. For example, the monoid B^* is generated by 0 and 1. A section of its transition diagram can be obtained by allowing input only from the generating set. The monoid of integers under addition is generated by the set $\{-1, 1\}$. The transition diagram for this monoid can be visualized by drawing a small portion of it, as in Figure 14.5.6. The same is true for the additive monoid of integers, as seen in Figure 14.5.7.

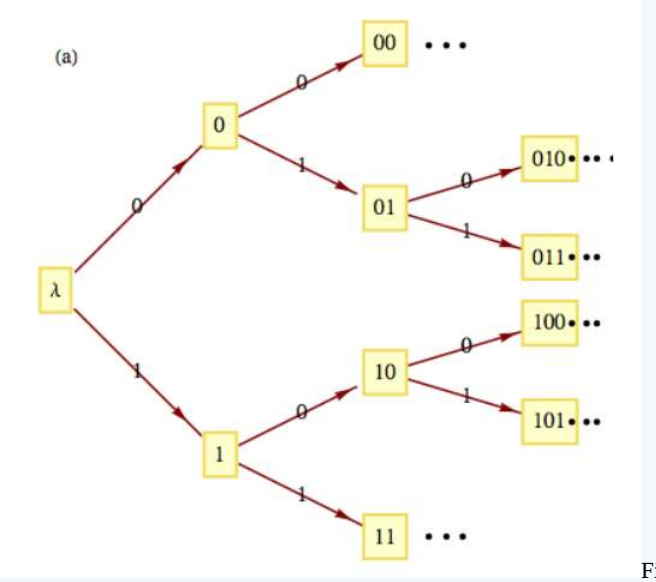


Figure 14.5.6: An infinite machine B^*

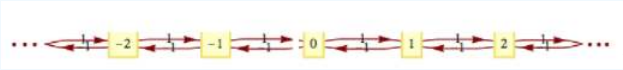


Figure 14.5.7: An infinite machine $[\mathbb{Z}; +]$

a. Draw a transition diagram for $\{a, b, c\}$
b. Draw a transition diagram for $[\mathbb{Z} \times \mathbb{Z}; \text{componentwise addition}]$.
c. Draw a transition diagram for $[\mathbb{Z}; +]$ with generating set $\{5, -2\}$.

This page titled 14.5: The Machine of a Monoid is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

15: Group Theory and Applications

> ***alternating group***
>
> *N objects are ordered, and you*
> *Switch consecutive pairs two by two.*
> *All reorders you get*
> *Will comprise a new set*
> *Called an **alternating group** when you're through.*
>
> *Chris Doyle, The Omnificent English Dictionary in Limerick Form*

In Chapter 11, we introduced groups as a typical algebraic system. The associated concepts of subgroup, group isomorphism, and direct products of groups were also introduced. Groups were chosen for that chapter because they are among the simplest types of algebraic systems. Despite this simplicity, group theory abounds with interesting applications. In this chapter we will introduce some more important concepts in elementary group theory, and some of their applications.

15.1: Cyclic Groups
15.2: Cosets and Factor Groups
15.3: Permutation Groups
15.4: Normal Subgroups and Group Homomorphisms
15.5: Coding Theory, Group Codes

This page titled 15: Group Theory and Applications is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

15.1: Cyclic Groups

Groups are classified according to their size and structure. A group's structure is revealed by a study of its subgroups and other properties (e.g., whether it is abelian) that might give an overview of it. Cyclic groups have the simplest structure of all groups.

Definition 15.1.1: Cyclic Group

Group G is cyclic if there exists $a \in G$ such that the cyclic subgroup generated by a, $\langle a \rangle$, equals all of G. That is, $G = \{na | n \in \mathbb{Z}\}$, in which case a is called a generator of G. The reader should note that additive notation is used for G.

Example 15.1.1: A Finite Cyclic Group

$\mathbb{Z}_{12} = [\mathbb{Z}_{12}; +_{12}]$, where $+_{12}$ is addition modulo 12, is a cyclic group. To verify this statement, all we need to do is demonstrate that some element of $\mathbb{Z}_{12}$ is a generator. One such element is 5; that is, $\langle 5 \rangle = \mathbb{Z}_{12}$. One more obvious generator is 1. In fact, 1 is a generator of every $[\mathbb{Z}_n; +_n]$. The reader is asked to prove that if an element is a generator, then its inverse is also a generator. Thus, $-5 = 7$ and $-1 = 11$ are the other generators of $\mathbb{Z}_{12}$. The remaining eight elements of the group are not generators.

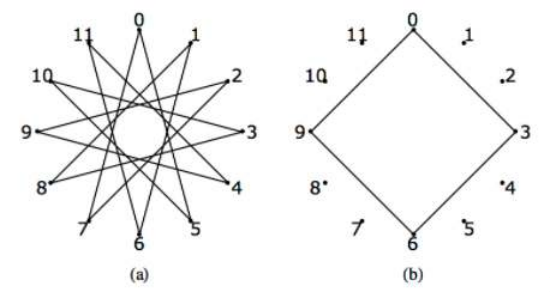


Figure 15.1.1: Copy and Paste Caption here. (Copyright; author via source)

Figure 15.1.1(a) is an example of “string art” that illustrates how 5 generates $\mathbb{Z}_{12}$. Twelve tacks are placed evenly around a circle and numbered 0 through 11. A string is tied to tack 0, and is then looped around every fifth tack. As a result, the numbers of the tacks that are reached are exactly the ordered multiples of 5 modulo 12: 5, 10, 3, ... , 7, 0. Note that if every seventh tack were used, the same artwork would be produced. If every third tack were connected, as in Figure 15.1.1(b), the resulting loop would only use four tacks; thus 3 does not generate $\mathbb{Z}_{12}$.

Example 15.1.2: The Group of Integers is Cyclic

The additive group of integers, $[\mathbb{Z}; +]$, is cyclic:

$$\mathbb{Z} = \langle 1 \rangle = \{n \cdot 1 | n \in \mathbb{Z}\}$$

This observation does not mean that every integer is the product of an integer times 1. It means that

$$\mathbb{Z} = \{0\} \cup \{\overbrace{1+1+\cdots+1}^{n \text{ terms}} \mid n \in \mathbb{P}\} \cup \{\overbrace{(-1)+(-1)+\cdots+(-1)}^{n \text{ terms}} \mid n \in \mathbb{P}\}$$

Theorem 15.1.1: Cyclic Implies Abelian

If $[G; *]$ is cyclic, then it is abelian.

Proof

Let a be any generator of G and let $b, c \in G$. By the definition of the generator of a group, there exist integers m and n such that $b = ma$ and $c = na$. Thus, using Theorem 11.3.9,

$$\begin{aligned} b*c &= (ma)*(na) \\ &= (m+n)a \\ &= (n+m)a \\ &= (na)*(ma) \\ &= c*b \end{aligned}$$

One of the first steps in proving a property of cyclic groups is to use the fact that there exists a generator. Then every element of the group can be expressed as some multiple of the generator. Take special note of how this is used in theorems of this section.

Up to now we have used only additive notation to discuss cyclic groups. Theorem 15.1.1 actually justifies this practice since it is customary to use additive notation when discussing abelian groups. Of course, some concrete groups for which we employ multiplicative notation are cyclic. If one of its elements, a, is a generator,

$$\langle a \rangle = \{a^n \mid n \in \mathbb{Z}\}$$

Example 15.1.3: A Cyclic Multiplicative Group

The group of positive integers modulo 11 with modulo 11 multiplication, $[\mathbb{Z}_{11}^*; \times_{11}]$, is cyclic. One of its generators is 6: $6^1 = 6$, $6^2 = 3$, $6^3 = 7, \ldots, 6^9 = 2$, and $6^{10} = 1$, the identity of the group.

Example 15.1.4: A Non-Cyclic Group

The real numbers with addition, $[\mathbb{R}; +]$ is a noncyclic group. The proof of this statement requires a bit more generality since we are saying that for all $r \in \mathbb{R}$, $\langle r \rangle$ is a proper subset of $\mathbb{R}$. If r is nonzero, the multiples of r are distributed over the real line, as in Figure 15.1.2 It is clear then that there are many real numbers, like $r/2$, that are not in $\langle r \rangle$.

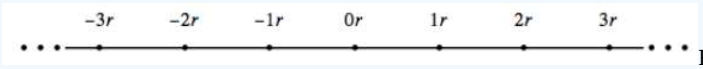


Figure 15.1.2: Elements of $\langle r \rangle, r > 0$

The next two proofs make use of the Theorem 11.4.1.

The following theorem shows that a cyclic group can never be very complicated.

Theorem 15.1.2: Possible Cyclic Group Structures

If G is a cyclic group, then G is either finite or countably infinite. If G is finite and $|G| = n$, it is isomorphic to $[\mathbb{Z}_n; +_n]$. If G is infinite, it is isomorphic to $[\mathbb{Z}; +]$.

Proof

Case 1: $|G| < \infty$. If a is a generator of G and $|G| = n$, define $\phi : \mathbb{Z}_n \to G$ by $\phi(k) = ka$ for all $k \in \mathbb{Z}_n$.

Since $\langle a \rangle$ is finite, we can use the fact that the elements of $\langle a \rangle$ are the first n nonnegative multiples of a. From this observation, we see that ϕ is a surjection. A surjection between finite sets of the same cardinality must be a bijection. Finally, if $p, q \in \mathbb{Z}_n$,

$$\begin{aligned} \phi(p) + \phi(q) &= pa + qa \\ &= (p+q)a \\ &= (p +_n q)a \quad \text{see exercise 15.1.10} \\ &= \phi(p +_n q) \end{aligned}$$

Therefore ϕ is an isomorphism.

Case 2: $|G| = \infty$. We will leave this case as an exercise.

Theorem 15.1.3: Subgroups of Cyclic Groups

Every subgroup of a cyclic group is cyclic.

Proof

Let G be cyclic with generator a and let $H \leq G$. If $H = \{e\}$, H has e as a generator. We may now assume that $|H| \geq 2$ and $a \neq e$. Let m be the least positive integer such that ma belongs to H. This is the key step. It lets us get our hands on a generator of H. We will now show that $c = ma$ generates H. Certainly, $\langle c \rangle \subseteq H$, but suppose that $\langle c \rangle \neq H$. Then there exists $b \in H$ such that $b \notin \langle c \rangle$. Now, since b is in G, there exists $n \in \mathbb{Z}$ such that $b = na$. We now apply the division property and divide n by m. $b = na = (qm + r)a = (qm)a + ra, \quad$ where $0 \leq r < m$. We note that r cannot be zero for otherwise we would have $b = na = q(ma) = qc \in \langle c \rangle$. Therefore, $ra = na - (qm)a \in H$. This contradicts our choice of m because $0 < r < m$.

Example 15.1.5: All Subgroups of $\mathbb{Z}_{10}$

The only proper subgroups of $\mathbb{Z}_{10}$ are $H_1 = \{0, 5\}$ and $H_2 = \{0, 2, 4, 6, 8\}$. They are both cyclic: $H_1 = \langle 5 \rangle$, while $H_2 = \langle 2 \rangle = \langle 4 \rangle = \langle 6 \rangle = \langle 8 \rangle$. The generators of $\mathbb{Z}_{10}$ are 1, 3, 7, and 9.

Example 15.1.6: All Subgroups of $\mathbb{Z}$

With the exception of $\{0\}$, all subgroups of $\mathbb{Z}$ are isomorphic to $\mathbb{Z}$. If $H \leq \mathbb{Z}$, then H is the cyclic subgroup generated by the least positive element of H. It is infinite and so by Theorem 15.1.3it is isomorphic to $\mathbb{Z}$.

We now cite a useful theorem for computing the order of cyclic subgroups of a cyclic group:

Theorem 15.1.4: The Order of Elements of a Finite Cyclic Group

If G is a cyclic group of order n and a is a generator of G, the order of ka is n/d, where d is the greatest common divisor of n and k.

Proof

The proof of this theorem is left to the reader.

Example 15.1.7: Computation of an Order in a Cyclic Group

To compute the order of $\langle 18 \rangle$ in $\mathbb{Z}_{30}$, we first observe that 1 is a generator of $\mathbb{Z}_{30}$ and $18 = 18(1)$. The greatest common divisor of 18 and 30 is 6. Hence, the order of $\langle 18 \rangle$ is 30/6, or 5.

At this point, we will introduce the idea of a fast adder, a relatively modern application (Winograd, 1965) of an ancient theorem, the Chinese Remainder Theorem. We will present only an overview of the theory and rely primarily on examples.

Out of necessity, integer addition with a computer is addition modulo n, for n some larger number. Consider the case where n is small, like 64. Then addition involves the addition of six-digit binary numbers. Consider the process of adding 31 and 1. Assume the computer's adder takes as input two bit strings $a = \{a_0, a_1, a_2, a_3, a_4, a_5\}$ and $b = \{b_0, b_1, b_2, b_3, b_4, b_5\}$ and outputs $s = \{s_0, s_1, s_2, s_3, s_4, s_5\}$, the sum of a and b. Then, if $a = 31 = (1, 1, 1, 1, 1, 0)$ and $b = 1 = (1, 0, 0, 0, 0, 0)$, s will be (0, 0, 0, 0, 0, 1), or 32. The output $s = 1$ cannot be determined until all other outputs have been determined. If addition is done with a finite-state machine, as in Example 14.3.5, the time required to get s will be six time units, where one time unit is the time it takes to get one output from the machine. In general, the time required to obtain s will be proportional to the number of bits. Theoretically, this time can be decreased, but the explanation would require a long digression and our relative results would not change that much. We will use the rule that the number of time units needed to perform addition modulo n is proportional to $\lceil \log_2 n \rceil$.

Now we will introduce a hypothetical problem that we will use to illustrate the idea of a fast adder. Suppose that we had to add 1,000 numbers modulo $27720 = 8 \cdot 9 \cdot 5 \cdot 7 \cdot 11$. By the rule above, since $2^{14} < 27720 < 2^{15}$, each addition would take 15 time

units. If the sum is initialized to zero, 1,000 additions would be needed; thus, 15,000 time units would be needed to do the additions. We can improve this time dramatically by applying the Chinese Remainder Theorem.

Theorem 15.1.5: Chinese Remainder Theorem (CRT)

Let $n_1, n_2, \ldots, n_p$ be integers that have no common factor greater than one between any pair of them; i. e., they are relatively prime. Let $n = n_1 n_2 \cdots n_p$. Define

$$\theta : \mathbb{Z}_n \to \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_p}$$

by

$$\theta(k) = (k_1, k_2, \ldots, k_p)$$

where for $1 \leq i \leq p$, $0 \leq k_i < n_i$ and $k \equiv k_i \pmod{n_i}$. Then θ is an isomorphism from $\mathbb{Z}_n$ into $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_p}$.

The Chinese Remainder Theorem can be stated in several different forms, and its proof can be found in many abstract algebra texts.

As we saw in Chapter 11, $\mathbb{Z}_6$ is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_3$. This is the smallest case to which the CRT can be applied. An isomorphism between $\mathbb{Z}_6$ and $\mathbb{Z}_2 \times \mathbb{Z}_3$ is

$$\begin{aligned} \theta(0) &= (0,0) \quad \theta(3) = (1,0) \\ \theta(1) &= (1,1) \quad \theta(4) = (0,1) \\ \theta(2) &= (0,2) \quad \theta(5) = (1,2) \end{aligned}$$

Let's consider a somewhat larger case. We start by selecting a modulus that can be factored into a product of relatively prime integers: $n = 21,600 = 2^5 3^3 5^2$. In this case the factors are $2^5 = 32$, $3^3 = 27$, and $5^2 = 25$. They need not be powers of primes, but it is easy to break the factors into this form to assure relatively prime numbers. To add in $\mathbb{Z}_n$, we need $\lceil \log_2 n \rceil = 15$ time units. Let $G = \mathbb{Z}_{32} \times \mathbb{Z}_{27} \times \mathbb{Z}_{25}$. The CRT gives us an isomorphism between $\mathbb{Z}_{21600}$ and G. The basic idea behind the fast adder, illustrated in Figure 15.1.3, is to make use of this isomorphism. The notation `x += a` is interpreted as the instruction to add the value of `a` to the variable `x`.

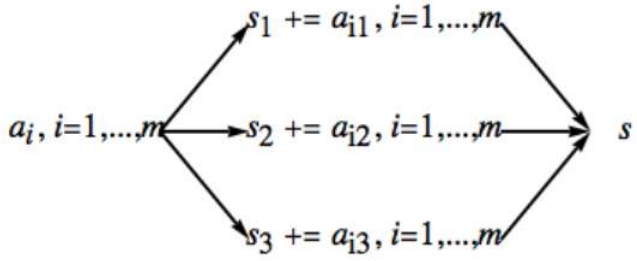


Figure 15.1.3: Fast Adder Scheme

Assume we have several integers $a_1, \ldots, a_m$ to be added. Here, we assume $m = 20$. We compute the sum `s` to compare our result with this true sum.

```
a=[1878,1384,84,2021,784,1509,1740,1201,2363,1774,
   1865,33,1477,894,690,520,198,1349,1278,650]
s =0
for t in a:
    s+=t
s
```

Although our sum is an integer calculation, we will put our calculation in the context of the integers modulo 21600. The isomophism from $\mathbb{Z}_{21600}$ into $G = \mathbb{Z}_{32} \times \mathbb{Z}_{27} \times \mathbb{Z}_{25}$ is defined in Sage as `theta`. In addition we demonstrate that the operations in these groups are preserved by `theta`.

```
G=cartesian_product([Integers(32),Integers(27),Integers(25)])
def theta(x):
    return G((x%32,x%27,x%25))
```

```
4 [theta(1878)+theta(1384),theta(1878+1384)]
```

We initialize the sums in each factor of the range of `theta` to zero and decompose each summand t into a triple $\theta(t) = (t_1, t_2, t_3) \in G$.

```
1 sum=G((0,0,0))
2 for t in a:
3     sum+=theta(t)
4 sum
```

Addition in G can be done in parallel so that each new subtotal in the form of the triple (s_1, s_2, s_3) takes only as long to compute as it takes to add in the largest modulus, $\log_2 32 = 5$ time units, if calculations are done in parallel. By the time rule that we have established, the addition of 20 numbers can be done in $20 \cdot 5 = 100$ time units, as opposed to $20 \cdot 15 = 300$ time units if we do the calculations in $\mathbb{Z}_{21600}$. However the result is a triple in G. The function that performs the inverse of `theta` is built into most mathematics programs, including Sage. In Sage the function is `crt` . We use this function to compute the inverse of our triple, which is an element of $\mathbb{Z}_{21600}$. The result isn't the true sum because the modulus 21600 is not large enough. However, we verify that our result is congruent to the true sum modulo 21600.

```
1 isum=crt([12,13,17],[32,27,25])
2 [isum,(s-isum)%(21600)]
```

In order to get the true sum from our scheme, the modulus would need to be increased by moving from 21600 to, for example, $21600 * 23 = 496800$. Mapping into the new group, $G = \mathbb{Z}_{32} \times \mathbb{Z}_{27} \times \mathbb{Z}_{25} \times \mathbb{Z}_{23}$ will take slightly longer, as will the inversion process with `crt` , but adding the summands that are in the form of quadruples can be done with no additional time.

The computation of $\theta^{-1}(s_1, s_2, s_3)$ that is done by the Sage function `crt` can be accomplished in a variety of ways. All of them ultimately are simplified by the fact that θ^{-1} is also an isomorphism. One approach is to use the isomorphism property to realize that the value of $\theta^{-1}(s_1, s_2, s_3)$ is $s_1\theta^{-1}(1,0,0) + s_2\theta^{-1}(0,1,0) + s_3\theta^{-1}(0,0,1)$. The arithmetic in this expression is in the domain of θ and is more time consuming, but it need only be done once. This is why the fast adder is only practical in situations where many additions must be performed to get a single sum.

The inverse images of the "unit vectors" can be computed ahead of time.

```
1 u=[crt([1,0,0],[32,27,25]),
2    crt([0,1,0],[32,27,25]),crt([0,0,1],[32,27,25])]
3 u
```

The result we computed earlier can be computed directly by in the larger modulus.

```
1 (7425*12 + 6400*13+ 7776* 17)%21600
```

To further illustrate the potential of fast adders, consider increasing the modulus to $n = 2^5 3^3 5^2 7^2 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \approx 3.1 \times 10^{21}$. Each addition using the usual modulo n addition with full adders would take 72 time units. By decomposing each summand into 15-tuples according to the CRT, the time is reduced to $\lceil \log_2 49 \rceil = 6$ time units per addition.

15.1.1: Exercises

Exercise 15.1.1

What generators besides 1 does $[\mathbb{Z}; +]$ have?

Answer

The only other generator is -1.

Exercise 15.1.2

Suppose $[G;*]$ is a cyclic group with generator g. If you build a graph of with vertices from the elements of G and edge set $E=\{(a,g*a)\mid a\in G\}$, what would the graph look like? If G is a group of even order, what would a graph with edge set $E'=\{(a,g^2*a)\mid a\in G\}$ look like?

Exercise 15.1.3

Prove that if $|G|>2$ and G is cyclic, G has at least two generators.

Answer

If $|G|=m$, $m>2$, and $G=\langle a\rangle$, then a, $a^2,\ldots,a^{m-1}$, $a^m=e$ are distinct elements of G. Furthermore, $a^{-1}=a^{m-1}\neq a$, If $1\leq k\leq m$, a^{-1} generates a^k:

$$\begin{aligned}\left(a^{-1}\right)^{m-k}&=\left(a^{m-1}\right)^{m-k}\\&=a^{m^2-m-mk+k}\\&=\left(a^m\right)^{m-k-1}*a^k\\&=e*a^k=a^k\end{aligned}$$

Similarly, if G is infinite and $G=\langle a\rangle$, then a^{-1} generates G.

Exercise 15.1.4

If you wanted to list the generators of $\mathbb{Z}_n$ you would only have to test the first $n/2$ positive integers. Why?

Exercise 15.1.5

Which of the following groups are cyclic? Explain.

a. $[\mathbb{Q};+]$
b. $[\mathbb{R}^+;\cdot]$
c. $[6\mathbb{Z};+]$ where $6\mathbb{Z}=\{6n|n\in\mathbb{Z}\}$
d. $\mathbb{Z}\times\mathbb{Z}$
e. $\mathbb{Z}_2\times\mathbb{Z}_3\times\mathbb{Z}_{25}$

Answer

a. No. Assume that $q\in\mathbb{Q}$ generates $\mathbb{Q}$. Then $\langle q\rangle=\{nq:n\in\mathbb{Z}\}$. But this gives us at most integer multiples of q, not every element in $\mathbb{Q}$.
b. No. Similar reasoning to part a.
c. Yes. 6 is a generator of $6\mathbb{Z}$.
d. No.
e. Yes, $(1,1,1)$is a generator of the group.

Exercise 15.1.6

For each group and element, determine the order of the cyclic subgroup generated by the element:

a. $\mathbb{Z}_{25}$, 15
b. $\mathbb{Z}_4\times\mathbb{Z}_9$, $(2,6)$ (apply Exercise 15.1.8)
c. $\mathbb{Z}_{64}$, 2

Exercise 15.1.7

How can Theorem 15.1.4be applied to list the generators of $\mathbb{Z}_n$? What are the generators of $\mathbb{Z}_{25}$? Of $\mathbb{Z}_{256}$?

Answer

Theorem 15.1.4implies that a generates $\mathbb{Z}_n$ if and only if the greatest common divisor of n and a is 1. Therefore the list of generators of $\mathbb{Z}_n$ are the integers in $\mathbb{Z}_n$ that are relatively prime to n. The generators of $\mathbb{Z}_{25}$ are all of the nonzero elements except 5, 10, 15, and 20. The generators of $\mathbb{Z}_{256}$ are the odd integers in $\mathbb{Z}_{256}$ since 256 is 2^8.

Exercise 15.1.8

Prove that if the greatest common divisor of n and m is 1, then (1, 1) is a generator of $\mathbb{Z}_n \times \mathbb{Z}_m$, and hence, $\mathbb{Z}_n \times \mathbb{Z}_m$ is isomorphic to $\mathbb{Z}_{nm}$.

Exercise 15.1.9

a. Illustrate how the fast adder can be used to add the numbers 21, 5, 7, and 15 using the isomorphism between $\mathbb{Z}_{77}$ and $\mathbb{Z}_7 \times \mathbb{Z}_{11}$.

b. If the same isomorphism is used to add the numbers 25, 26, and 40, what would the result be, why would it be incorrect, and how would the answer differ from the answer in part a?

Answer

a. $\theta : \mathbb{Z}_{77} \to \mathbb{Z}_7 \times \mathbb{Z}_{11}$ maps the given integers as follows:

$$\begin{array}{rcl} 21 & \to & (0,10) \\ 5 & \to & (5,5) \\ 7 & \to & (0,7) \\ 15 & \to & \underline{(1,4)} \\ \text{sum} = 48 & \leftarrow & (6,4) = \text{ sum} \end{array}$$

The final sum, 48, is obtained by using the facts that $\theta^{-1}(1,0) = 22$ and $\theta^{-1}(0,1) = 56$

$$\begin{aligned} \theta^{-1}(6,4) = 6 \times_{77} \theta^{-1}(1,0) + 4 \times_{77} \theta^{-1}(0,1) & \\ = 6 \times_{77} 22 +_{77} 4 \times_{77} 56 & \\ = 55 +_{77} 70 & \\ = 48 & \end{aligned}.$$

b. Using the same isomorphism:

$$\begin{array}{rcl} 25 & \to & (4,3) \\ 26 & \to & (5,4) \\ 40 & \to & (5,7) \\ & & \text{sum} = (0,3) \end{array}$$

$$\begin{aligned} \theta^{-1}(0,3) &= 3 \times_{77} \theta^{-1}(0,1) \\ &= 3 \times_{77} 56 \\ &= 14 \end{aligned}.$$

The actual sum is 91. Our result is incorrect, since 91 is not in $\mathbb{Z}_{77}$. Notice that 91 and 14 differ by 77. Any error that we get using this technique will be a multiple of 77.

Exercise 15.1.10

Prove that if G is a cyclic group of order n with generator a, and $p, q \in \{0, 1, \ldots, n-1\}$, then $(p+q)a = (p +_n q)\, a$.

This page titled 15.1: Cyclic Groups is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

15.2: Cosets and Factor Groups

Consider the group $[\mathbb{Z}_{12}; +_{12}]$. As we saw in the previous section, we can picture its cyclic properties with the string art of Figure 15.1.1. Here we will be interested in the non-generators, like 3. The solid lines in Figure 15.2.1 show that only one-third of the tacks have been reached by starting at zero and jumping to every third tack. The numbers of these tacks correspond to $\langle 3 \rangle = \{0, 3, 6, 9\}$.

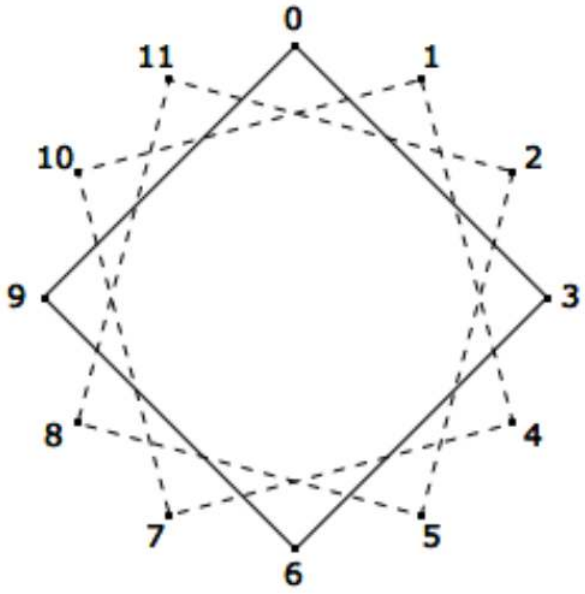


Figure 15.2.1: "String art" cosets

What happens if you start at one of the unused tacks and again jump to every third tack? The two broken paths on Figure 15.2.1 show that identical squares are produced. The tacks are thus partitioned into very similar subsets. The subsets of $\mathbb{Z}_{12}$ that they correspond to are $\{0, 3, 6, 9\}, \{1, 4, 7, 10\}$,and $\{2, 5, 8, 11\}$.These subsets are called cosets. In particular, they are called cosets of the subgroup $\{0, 3, 6, 9\}$.We will see that under certain conditions, cosets of a subgroup can form a group of their own. Before pursuing this example any further we will examine the general situation.

Definition 15.2.1: Coset

If $[G; *]$ is a group, $H \leq G$ and $a \in G$, the left coset of H generated by a is

$$a * H = \{a * h | h \in H\}$$

and the right coset of H generated by a is

$$H * a = \{h * a | h \in H\}.$$

Note 15.2.1

a. H itself is both a left and right coset since $e * H = H * e = H$.
b. If G is abelian, $a * H = H * a$ and the left-right distinction for cosets can be dropped. We will normally use left coset notation in that situation.

Definition 15.2.2: Cost Representative

Any element of a coset is called a representative of that coset.

One might wonder whether a is in any way a special representative of $a * H$ since it seems to define the coset. It is not, as we shall see.

Remark 15.2.1: A Duality Principle

A duality principle can be formulated concerning cosets because left and right cosets are defined in such similar ways. Any theorem about left and right cosets will yield a second theorem when "left" and "right" are exchanged for "right" and "left."

Theorem 15.2.1

If $b \in a * H$, then $a * H = b * H$, and if $b \in H * a$, then $H * a = H * b$.

Proof

In light of the remark above, we need only prove the first part of this theorem. Suppose that $x \in a * H$. We need only find a way of expressing x as "b times an element of H." Then we will have proven that $a * H \subseteq b * H$. By the definition of $a * H$, since b and x are in $a * H$, there exist h_1 and h_2 in H such that $b = a * h_1$ and $x = a * h_2$. Given these two equations, $a = bh_1^{-1}$ and

$$x = a * h_2 = (b * h_1^{-1}) * h_2 = b * (h_1^{-1} * h_2)$$

Since $h_1, h_2 \in H$, $h_1^{-1} * h_2 \in H$, and we are done with this part of the proof. In order to show that $b * H \subseteq a * H$, one can follow essentially the same steps, which we will let the reader fill in.

Example 15.2.1

In Figure 15.2.1, you can start at either 1 or 7 and obtain the same path by taking jumps of three tacks in each step. Thus,

$$1 +_{12} \{0, 3, 6, 9\} = 7 +_{12} \{0, 3, 6, 9\} = \{1, 4, 7, 10\}.$$

The set of left (or right) cosets of a subgroup partition a group in a special way:

Theorem 15.2.2: Cosets Partition a Group

If $[G; *]$ is a group and $H \leq G$, the set of left cosets of H is a partition of G. In addition, all of the left cosets of H have the same cardinality. The same is true for right cosets.

Proof

That every element of G belongs to a left coset is clear because $a \in a * H$ for all $a \in G$. If $a * H$ and $b * H$ are left cosets, we will prove that they are either equal or disjoint. If $a * H$ and $b * H$ are not disjoint, $a * H \cap b * H$ is nonempty and some element $c \in G$ belongs to the intersection. Then by Theorem 15.2.1, $c \in a * H \Rightarrow a * H = c * H$ and $c \in b * H \Rightarrow b * H = c * H$. Hence $a * H = b * H$.

We complete the proof by showing that each left coset has the same cardinality as H. To do this, we simply observe that if $a \in G$, $\rho : H \to a * H$ defined by $\rho(h) = a * h$ is a bijection and hence $|H| = |a * H|$. We will leave the proof of this statement to the reader.

The function ρ has a nice interpretation in terms of our opening example. If $a \in \mathbb{Z}_{12}$, the graph of $\{0, 3, 6, 9\}$is rotated $(30a)^\circ$ to coincide with one of the three cosets of $\{0, 3, 6, 9\}$.

Corollary 15.2.1: A Coset Counting Formula

If $|G| < \infty$ and $H \leq G$, the number of distinct left cosets of H equals $\frac{|G|}{|H|}$. For this reason we use G/H to denote the set of left cosets of H in G

Proof

This follows from the partitioning of G into equal sized sets, one of which is H.

Example 15.2.2

The set of integer multiples of four, $4\mathbb{Z}$, is a subgroup of $[\mathbb{Z};+]$. Four distinct cosets of $4\mathbb{Z}$ partition the integers. They are $4\mathbb{Z}$, $1+4\mathbb{Z}$, $2+4\mathbb{Z}$, and $3+4\mathbb{Z}$, where, for example, $1+4\mathbb{Z}=\{1+4k|k\in\mathbb{Z}\}$. $4\mathbb{Z}$ can also be written $0+4\mathbb{Z}$.

Convention 15.2.1: Distinguished Representatives

Although we have seen that any representative can describe a coset, it is often convenient to select a distinguished representative from each coset. The advantage to doing this is that there is a unique name for each coset in terms of its distinguished representative. In numeric examples such as the one above, the distinguished representative is usually the smallest nonnegative representative. Remember, this is purely a convenience and there is absolutely nothing wrong in writing $-203+4\mathbb{Z}$, $5+4\mathbb{Z}$, or $621+4\mathbb{Z}$ in place of $1+4\mathbb{Z}$ because $-203,5,621\in 1+4\mathbb{Z}$.

Before completing the main thrust of this section, we will make note of a significant implication of Theorem 15.2.2 Since a finite group is divided into cosets of a common size by any subgroup, we can conclude:

Theorem 15.2.3: Lagrange's Theorem

The order of a subgroup of a finite group must divide the order of the group.

One immediate implication of Lagrange's Theorem is that if p is prime, $\mathbb{Z}_p$ has no proper subgroups.

We will now describe the operation on cosets which will, under certain circumstances, result in a group. For most of this section, we will assume that G is an abelian group. This is one sufficient (but not necessary) condition that guarantees that the set of left cosets will form a group.

Definition 15.2.3: Operation on Cosets

Let C and D be left cosets of H, a subgroup of G with representatives c and d, respectively. Then

$$C\otimes D=(c*H)\otimes(d*H)=(c*d)*H$$

The operation $\otimes$ is called the operation induced on left cosets by $*$.

In Theorem 15.2.4, later in this section, we will prove that if G is an abelian group, $\otimes$ is indeed an operation. In practice, if the group G is an additive group, the symbol $\otimes$ is replaced by $+$, as in the following example.

Example 15.2.3: Computing with Cosets of $4\mathbb{Z}$

Consider the cosets described in Example 15.2.2 For brevity, we rename $0+4\mathbb{Z}$, $1+4\mathbb{Z}$, $2+4\mathbb{Z}$, and $3+4\mathbb{Z}$ with the symbols $\bar{0}$, $\bar{1}$, $\bar{2}$, and $\bar{3}$. Let's do a typical calculation, $\bar{1}+\bar{3}$. We will see that the result is always going to be $\bar{0}$, no matter what representatives we select. For example, $9\in\bar{1}$, $7\in\bar{3}$, and $9+7=16\in\bar{0}$. Our choice of the representatives $\bar{1}$ and $\bar{3}$ were completely arbitrary.

In general, $C\otimes D$ can be computed in many ways, and so it is necessary to show that the choice of representatives does not affect the result. When the result we get for $C\otimes D$ is always independent of our choice of representatives, we say that "$\otimes$ is well defined." Addition of cosets is a well-defined operation on the left cosets of $4\mathbb{Z}$ and is summarized in the following table. Do you notice anything familiar?

$\otimes$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$
$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$
$\bar{1}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{0}$
$\bar{2}$	$\bar{2}$	$\bar{3}$	$\bar{0}$	$\bar{1}$
$\bar{3}$	$\bar{3}$	$\bar{0}$	$\bar{1}$	$\bar{2}$

Example 15.2.4: Cosets of the Integers in the Group of Real Numbers

Consider the group of real numbers, $[\mathbb{R};+]$, and its subgroup of integers, $\mathbb{Z}$. Every element of $\mathbb{R}/\mathbb{Z}$ has the same cardinality as $\mathbb{Z}$. Let $s,t\in\mathbb{R}$. $s\in t+\mathbb{Z}$ if s can be written $t+n$ for some $n\in\mathbb{Z}$. Hence s and t belong to the same coset if they differ by an integer. (See Exercise 15.2.6 for a generalization of this fact.)

Now consider the coset $0.25+\mathbb{Z}$. Real numbers that differ by an integer from 0.25 are $1.25, 2.25, 3.25, \ldots$ and $-0.75, -1.75, -2.75, \ldots$ If any real number is selected, there exists a representative of its coset that is greater than or equal to 0 and less than 1. We will call that representative the distinguished representative of the coset. For example, 43.125 belongs to the coset represented by 0.125; $-6.382+\mathbb{Z}$ has 0.618 as its distinguished representative. The operation on $\mathbb{R}/\mathbb{Z}$ is commonly called addition modulo 1. A few typical calculations in $\mathbb{R}/\mathbb{Z}$ are

$$\begin{gathered}(0.1+\mathbb{Z})+(0.48+\mathbb{Z})=0.58+\mathbb{Z}\\(0.7+\mathbb{Z})+(0.31+\mathbb{Z})=0.01+\mathbb{Z}\\-(0.41+\mathbb{Z})=-0.41+\mathbb{Z}=0.59+\mathbb{Z}\\\text{and in general, } -(a+\mathbb{Z})=(1-a)+\mathbb{Z}\end{gathered}.$$

Example 15.2.5: Cosets in a Direct Product

Consider $F=(\mathbb{Z}_4\times\mathbb{Z}_2)/H$, where $H=\{(0,0),(0,1)\}$. Since $\mathbb{Z}_4\times\mathbb{Z}_2$ is of order 8, each element of F is a coset containing two ordered pairs. We will leave it to the reader to verify that the four distinct cosets are $(0,0)+H$, $(1,0)+H$, $(2,0)+H$ and $(3,0)+H$. The reader can also verify that F is isomorphic to $\mathbb{Z}_4$, since F is cyclic. An educated guess should give you a generator.

Example 15.2.6

Consider the group ${\mathbb{Z}_2}^4=\mathbb{Z}_2\times\mathbb{Z}_2\times\mathbb{Z}_2\times\mathbb{Z}_2$. Let H be $\langle(1,0,1,0)\rangle$, the cyclic subgroup of ${\mathbb{Z}_2}^4$ generate by (1,0,1,0). Since

$$(1,0,1,0)+(1,0,1,0)=(1+_2 1, 0+_2 0, 1+_2 1, 0+_2 0)=(0,0,0,0)$$

the order of H is 2 and , ${\mathbb{Z}_2}^4/H$ has $|\mathbb{Z}_2^4/H|=\frac{|\mathbb{Z}_2^4|}{|H|}=\frac{16}{2}=8$ elements. A typical coset is

$$C=(0,1,1,1)+H=\{(0,1,1,1),(1,1,0,1)\}$$

Note that since $2(0,1,1,1)=(0,0,0,0), 2C=C\otimes C=H,$ the identity for the operation on ${\mathbb{Z}_2}^4/H$. The orders of non-identity elements of this factor group are all 2, and it can be shown that the factor group is isomorphic to ${\mathbb{Z}_2}^3$.

Theorem 15.2.4: Coset Operation is Well-Defined (Abelian Case)

If G is an abelian group, and $H\leq G,$ the operation induced on cosets of H by the operation of G is well defined.

Proof

Suppose that a, b, and a', b'. are two choices for representatives of cosets C and D. That is to say that $a,a'\in C$, $b,b'\in D$. We will show that $a*b$ and $a'*b'$ are representatives of the same coset. Theorem 15.2.1 implies that $C=a*H$ and $D=b*H,$ thus we have $a'\in a*H$ and $b'\in b*H$. Then there exists $h_1,h_2\in H$ such that $a'=a*h_1$ and $b'=b*h_2$ and so

$$a'*b'=(a*h_1)*(b*h_2)=(a*b)*(h_1*h_2)$$

by various group properties and the assumption that G is abelian, which lets us reverse the order in which b and h_1 appear in the chain of equalities. This last expression for $a'*b'$ implies that $a'*b'\in(a*b)*H$ since $h_1*h_2\in H$ because H is a subgroup of G. Thus, we get the same coset for both pairs of representatives.

Theorem 15.2.5

Let G be a group and $H \leq G$. If the operation induced on left cosets of H by the operation of G is well defined, then the set of left cosets forms a group under that operation.

Proof

Let C_1, C_2, and C_3 be the left cosets with representatives r_1, r_2, and r_3, respectively. The values of $C_1 \otimes (C_2 \otimes C_3)$ and $(C_1 \otimes C_2) \otimes C_3$ are determined by $r_1 * (r_2 * r_3)$ and $(r_1 * r_2) * r_3$, respectively. By the associativity of $*$ in G, these two group elements are equal and so the two coset expressions must be equal. Therefore, the induced operation is associative. As for the identity and inverse properties, there is no surprise. The identity coset is H, or $e * H$, the coset that contains G's identity. If C is a coset with representative a; that is, if $C = a * H$, then C^{-1} is $a^{-1} * H$.

$$(a * H) \otimes \left(a^{-1} * H\right) = \left(a * a^{-1}\right) * H = e * H = \text{ identity coset.}$$

Definition 15.2.4: Factor Group

Let G be a group and $H \leq G$. If the set of left cosets of H forms a group, then that group is called the factor group of "G modulo H." It is denoted G/H.

Note 15.2.2

If G is abelian, then every subgroup of G yields a factor group. We will delay further consideration of the non-abelian case to Section 15.4.

Remark 15.2.2: On Notation

It is customary to use the same symbol for the operation of G/H as for the operation on G. The reason we used distinct symbols in this section was to make the distinction clear between the two operations.

15.2.1: Exercises

Exercise 15.2.1

Consider $\mathbb{Z}_{10}$ and the subsets of $\mathbb{Z}_{10}$, $\{0, 1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$. Why is the operation induced on these subsets by modulo 10 addition not well defined?

Answer

An example of a valid correct answer: Call the subsets A and B respectively. If we choose $0 \in A$ and $5 \in B$ we get $0 +_{10} 5 = 5 \in B$. On the other hand, if we choose $3 \in A$ and $8 \in B$, we get $3 +_{10} 8 = 1 \in A$. Therefore, the induced operation is not well defined on $\{A, B\}$.

Exercise 15.2.2

Can you think of a group G, with a subgroup H such that $|H| = 6$ and $|G/H| = 6$? Is your answer unique?

Exercise 15.2.3

For each group and subgroup, what is G/H isomorphic to?

a. $G = \mathbb{Z}_4 \times \mathbb{Z}_2$ and $H = \langle (2, 0) \rangle$. Compare to Example 15.2.5
b. $G = [\mathbb{C}; +]$ and $H = \mathbb{R}$.
c. $G = \mathbb{Z}_{20}$ and $H = \langle 8 \rangle$.

Answer

a. The four distinct cosets in G/H are $H=\{(0,0),(2,0)\}, (1,0)+H=\{(1,0),(3,0)\}, (0,1)+H=\{(0,1),(2,1)\}$, and $(1,1)+H=\{(1,1),(3,1)\}$. None of these cosets generates G/H; therefore G/H is not cyclic. Hence G/H must be isomorphic to $\mathbb{Z}_2\times\mathbb{Z}_2$.
b. The factor group is isomorphic to $[\mathbb{R};+]$. Each coset of $\mathbb{R}$ is a line in the complex plane that is parallel to the x-axis: $\tau:\mathbb{C}/\mathbb{R}\to\mathbb{R},$ where $T(\{a+bi\mid a\in\mathbb{R}\})=b$ is an isomorphism.
c. $\langle 8\rangle=\{0,4,8,12,16\}\Rightarrow|\mathbb{Z}_{20}/\langle 8\rangle|=4$. The four cosets are: $\bar{0}$, $\bar{1}$, $\bar{2}$, and $\bar{3}$. 1 generates all four cosets. The factor group is isomorphic to $[\mathbb{Z}_4;+_4]$ because $\bar{1}$ is a generator.

Exercise 15.2.4

For each group and subgroup, what is G/H isomorphic to?

a. $G=\mathbb{Z}\times\mathbb{Z}$ and $H=\{(a,a)|a\in\mathbb{Z}\}$.
b. $G=[\mathbb{R}^*;\cdot]$ and $H=\{1,-1\}$.
c. $G={\mathbb{Z}_2}^5$ and $H=\langle(1,1,1,1,1)\rangle$.

Exercise 15.2.5

Assume that G is a group, $H\leq G,$ and $a,b\in G$. Prove that $a*H=b*H$ if and only if $b^{-1}*a\in H$.

Answer

$$\begin{aligned} a*H=b*H &\Leftrightarrow a\in bH \\ &\Leftrightarrow a=b*h \text{ for some } h\in H \\ &\Leftrightarrow b^{-1}*a=h \text{ for some } h\in H \\ &\Leftrightarrow b^{-1}*a\in H \end{aligned}$$

Exercise 15.2.6

a. Real addition modulo r, $r>0,$ can be described as the operation induced on cosets of $\langle r\rangle$ by ordinary addition. Describe a system of distinguished representatives for the elements of $\mathbb{R}/\langle r\rangle$.
b. Consider the trigonometric function sine. Given that $\sin(x+2\pi k)=\sin x$ for all $x\in\mathbb{R}$ and $k\in\mathbb{Z},$ show how the distinguished representatives of $\mathbb{R}/\langle 2\pi\rangle$ can be useful in developing an algorithm for calculating the sine of a number.

Exercise 15.2.7

Complete the proof of Theorem 15.2.2by proving that if $a\in G,$ $\rho:H\to a*H$ defined by $\rho(h)=a*h$ is a bijection.

This page titled 15.2: Cosets and Factor Groups is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

15.3: Permutation Groups

15.3.1: Symmetric Groups

At the risk of boggling the reader's mind, we will now examine groups whose elements are functions. Recall that a permutation on a set A is a bijection from A into A. Suppose that $A = \{1, 2, 3\}$. There are $3! = 6$ different permutations on A. We will call the set of all 6 permutations S_3. They are listed in the following table. The matrix form for describing a function on a finite set is to list the domain across the top row and the image of each element directly below it. For example $r_1(1) = 2$.

Table 15.3.1: Elements of S_3

$i = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	$r_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$	$r_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$
$f_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$	$f_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$	$f_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$

The operation that will give $\{i, r_1, r_2, f_1, f_2, f_3\}$ a group structure is function composition. Consider the "product" $r_1 \circ f_3$:

$$\begin{aligned} r_1 \circ f_3(1) &= r_1\left(f_3(1)\right) = r_1(2) = 3 \\ r_1 \circ f_3(2) &= r_1\left(f_3(2)\right) = r_1(1) = 2 \\ r_1 \circ f_3(3) &= r_1\left(f_3(3)\right) = r_1(3) = 1 \end{aligned} .$$

The images of 1, 2, and 3 under $r_1 \circ f_3$ and f_2 are identical. Thus, by the definition of equality for functions, we can say $r_1 \circ f_3 = f_2$. The complete table for the operation of function composition is given in Table 15.3.2

Table 15.3.2: Operation Table for S_3

$\circ$	i	r_1	r_2	f_1	f_2	f_3
i	i	r_1	r_2	f_1	f_2	f_3
r_1	r_1	r_2	i	f_3	f_1	f_2
r_2	r_2	i	r_1	f_2	f_3	f_1
f_1	f_1	f_2	f_3	i	r_1	r_2
f_2	f_2	f_3	f_1	r_2	i	r_1
f_3	f_3	f_1	f_2	r_1	r_2	i

List 15.3.1

We don't even need the table to verify that we have a group. Based on the following observations, the set of all permutations on any finite set will be a group.

1. Function composition is always associative.
2. The identity for the group is i. If g is any one of the permutations on A and $x \in A$,

$$(g \circ i)(x) = g(i(x)) = g(x) \qquad (i \circ g)(x) = i(g(x)) = g(x)$$

   Therefore $g \circ i = i \circ g = g$.
3. A permutation, by definition, is a bijection. In Chapter 7 we proved that this implies that it must have an inverse and the inverse itself is a bijection and hence a permutation. Hence all elements of S_3 have an inverse in S_3. If a permutation is displayed in matrix form, its inverse can be obtained by exchanging the two rows and rearranging the columns so that the top row is in order. The first step is actually sufficient to obtain the inverse, but the sorting of the top row makes it easier to recognize the inverse.

   For example, let's consider a typical permutation on $\{1, 2, 3, 4, 5\}$, $f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 2 & 1 & 4 \end{pmatrix}$.

$$f^{-1} = \begin{pmatrix} 5 & 3 & 2 & 1 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 2 & 5 & 1 \end{pmatrix}.$$

Note 15.3.1

From Table 15.3.2, we can see that S_3 is non-abelian. Remember, non-abelian is the negation of abelian. The existence of two elements that don't commute is sufficient to make a group non-abelian. In this group, r_1 and f_3 is one such pair: $r_1 \circ f_3 = f_2$ while $f_3 \circ r_1 = f_1$, so $r_1 \circ f_3 \neq f_3 \circ r_1$. Caution: Don't take this to mean that every pair of elements has to have this property. There are several pairs of elements in S_3 that do commute. In fact, the identity, i, must commute with everything. Also every element must commute with its inverse.

Definition 15.3.1: Symmetric Group

Let A be a nonempty set. The set of all permutations on A with the operation of function composition is called the symmetric group on A, denoted S_A.

The cardinality of a finite set A is more significant than the elements, and we will denote by S_n the symmetric group on any set of cardinality n, $n \geq 1$.

Example 15.3.1: The Significance of S_3

Our opening example, S_3, is the smallest non-abelian group. For that reason, all of its proper subgroups are abelian: in fact, they are all cyclic. Figure 15.3.1 shows the Hasse diagram for the subgroups of S_3.

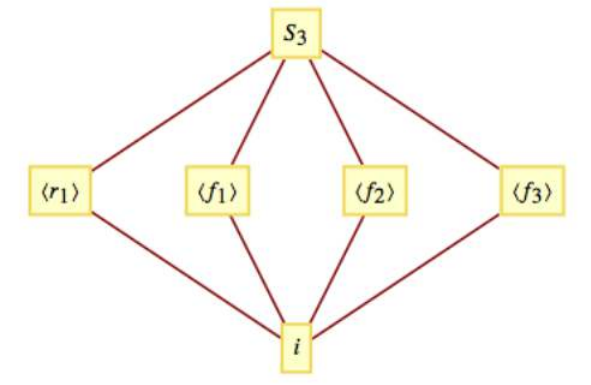


Figure 15.3.1: Lattice diagram of subgroups of S_3

Example 15.3.2: Smallest Symmetric Groups

The only abelian symmetric groups are S_1 and S_2 , with 1 and 2 elements, respectively. The elements of S_2 are $i = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$ and $\alpha = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$. S_2 is isomorphic to $\mathbb{Z}_2$.

Theorem 15.3.1

For $n \geq 1$, $|S_n| = n!$ and for $n \geq 3$, S_n is non-abelian.

Proof

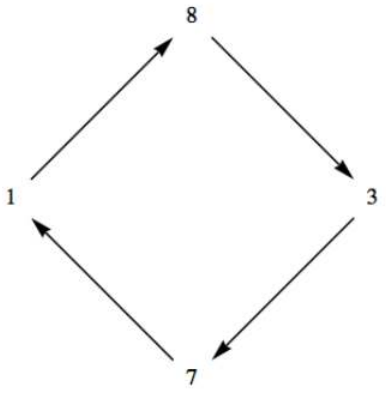
The first part of the theorem follows from the extended rule of products (see Chapter 2). We leave the details of proof of the second part to the reader after the following hint. Consider f in S_n where $f(1) = 2$, $f(2) = 3$, $f(3) = 1$, and $f(j) = j$ for $3 < j \leq n$. Therefore the cycle representation of f is $(1, 2, 3)$. Now define g in a similar manner so that when you compare $f(g(1))$ and $g(f(1))$ you get different results.

15.3.2: Cycle Notation

A second way of describing a permutation is by means of cycles, which we will introduce first with an example. Consider $f \in S_8$ defined using the now-familiar matrix notation:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 2 & 7 & 6 & 5 & 4 & 1 & 3 \end{pmatrix}$$

Consider the images of 1 when f is applied repeatedly. The images $f(1)$, $f(f(1))$, $f(f(f(1)))$, ... are $8, 3, 7, 1, 8, 3, 7, \ldots$.In Figure 15.3.2(a), this situation is represented by a graph with vertices 1, 8, 3, and 7 and shows that the values that you get by repeatedly applying f cycle through those values. This is why we refer to this part of f as a cycle of length 4. Of course starting at 8, 3, or 7 also produces the same cycle with only the starting value changing.



(a)

(8,3,7,1)
(3,7,1,8)
(7,1,8,3)
or
(1,8,3,7)

(b)

Figure 15.3.2: Representations of a cycle of length 4

Figure 15.3.2(a) illustrates how the cycle can be represented in a visual manner, but it is a bit awkward to write. Part (b) of the figure presents a more universally recognized way to write a cycle. In (b), a cycle is represented by a list where the image of any number in the list is its successor. In addition, the last number in the list has as its image the first number.

The other elements of the domain of f are never reached if you start in the cycle $(1, 8, 3, 7)$,and so looking at the images of these other numbers will produce numbers that are disjoint from the set $\{1, 8, 3, 7\}$.The other disjoint cycles of f are (2), (4, 6), and (5). We can express f as a product of disjoint cycles: $f = (1, 8, 3, 7)(2)(4, 6)(5)$or $f = (1, 8, 3, 7)(4, 6)$,where the absence of 2 and 5 implies that $f(2) = 2$ and $f(5) = 5$.

Note 15.3.2: Disjoint Cycles

We say that two cycles are disjoint if no number appears in both cycles, as is the case in our expressions for f above. Disjoint cycles can be written in any order. Thus, we could also say that $f = (4, 6)(1, 8, 3, 7)$.

Note 15.3.3: Composition of Permutations

We will now consider the composition of permutations written in cyclic form by an example. Suppose that $f = (1, 8, 3, 7)(4, 6)$and $g = (1, 5, 6)(8, 3, 7, 4)$are elements of S_8. To calculate $f \circ g$, we start with simple concatenation:

$$f \circ g = (1, 8, 3, 7)(4, 6)(1, 5, 6)(8, 3, 7, 4) \tag{15.3.1}$$

Although this is a valid expression for $f \circ g$, our goal is to express the composition as a product of disjoint cycles as f and g were individually written. We will start by determining the cycle that contains 1. When combining any number of cycles, they are always read from right to left, as with all functions. The first cycle in (15.3.1) does not contain 1; thus we move on to the second. The image of 1 under that cycle is 5. Now we move on to the next cycle, looking for 5, which doesn't appear. The fourth cycle does not contain a 5 either; so $f \circ g(1) = 5$.

At this point, we would have written "$f \circ g = (1, 5$" on paper. We repeat the steps to determine $f \circ g(5)$. This time the second cycle of (15.3.1) moves 5 to 6 and then the third cycle moves 6 to 4. Therefore, $f \circ g(5) = 4$. We continue until the cycle (1, 5, 4, 3) is completed by determining that $f \circ g(3) = 1$. The process is then repeated starting with any number that does not appear in the cycle(s) that have already been completed.

The final result for our example is $f \circ g = (1,5,4,3)(6,8,7)$. Since $f(2) = 2$ and $g(2) = 2$, $f \circ g(2) = 2$ and we need not include the one-cycle (2) in the final result, although it can be included.

Example 15.3.3: Some Compositions

a. $(1,2,3,4)(1,2,3,4) = (1,3)(2,4)$
b. $(1,4)(1,3)(1,2) = (1,2,3,4)$.

Notice that cyclic notation does not indicate the set which is being permuted. The examples above could be in S_5, where the image of 5 is 5. This ambiguity is usually overcome by making the context clear at the start of a discussion.

Definition 15.3.2: Transposition

A transposition is a cycle of length 2.

Observation 15.3.1: About Transpositions

$f = (1,4)$ and $g = (4,5)$ are transpositions in S_5. However, $f \circ g = (1,4,5)$ and $g \circ f = (1,5,4)$ are not transpositions; thus, the set of transpositions is not closed under composition. Since $f^2 = f \circ f$ and $g^2 = g \circ g$ are both equal to the identity permutation, f and g are their own inverses. In fact, every transposition is its own inverse.

Theorem 15.3.2: Decomposition into Cycles

Every cycle of length greater than 2 can be expressed as a product of transpositions.

Proof

We need only indicate how the product of transpositions can be obtained. It is easy to verify that a cycle of length k, $(a_1, a_2, a_3, \ldots, a_k)$, is equal to the following product of $k-1$ transpositions:

$$(a_1, a_k) \cdots (a_1, a_3)(a_1, a_2)$$

Of course, a product of cycles can be written as a product of transpositions just as easily by applying the rule above to each cycle. For example,

$$(1,3,5,7)(2,4,6) = (1,7)(1,5)(1,3)(2,6)(2,4)$$

Unlike the situation with disjoint cycles, we are not free to change the order of these transpositions.

15.3.3: Parity of Permutations and the Alternating Group

A decomposition of permutations into transpositions makes it possible to classify then and identify an important family of groups.

The proofs of the following theorem appears in many abstract algebra texts.

Theorem 15.3.3

Every permutation on a finite set can be expressed as the product of an even number of transpositions or an odd number of transpositions, but not both.

Theorem 15.3.3 suggests that S_n can be partitioned into its "even" and "odd" elements. For example, the even permutations of S_3 are i, $r_1 = (1,2,3) = (1,3)(1,2)$ and $r_2 = (1,3,2) = (1,2)(1,3)$. They form a subgroup, $\{i, r_1, r_2\}$ of S_3.

In general:

Definition 15.3.3: The Alternating Group

Let $n \geq 2$. The set of even permutations in S_n is a proper subgroup of S_n called the alternating group on $\{1, 2, \ldots, n\}$, denoted A_n.

We justify our statement that A_n is a group:

Theorem 15.3.4

Let $n \geq 2$. The alternating group is indeed a group and has order $\frac{n!}{2}$.

Proof

In this proof, the symbols s_i and t_i stand for transpositions and p, q are even nonnegative integers. If $f, g \in A_n$, we can write the two permutations as products of even numbers of transpositions, $f = s_1 s_2 \cdots s_p$ and $g = t_1 t_2 \cdots t_q$. Then

$$f \circ g = s_1 s_2 \cdots s_p t_1 t_2 \cdots t_q$$

Since $p + q$ is even, $f \circ g \in A_n$, and A_n is closed with respect to function composition. With this, we have proven that A_n is a subgroup of S_n by Theorem 11.5.2.

To prove the final assertion, let B_n be the set of odd permutations and let $\tau = (1, 2)$. Define $\theta : A_n \to B_n$ by $\theta(f) = f \circ \tau$. Suppose that $\theta(f) = \theta(g)$. Then $f \circ \tau = g \circ \tau$ and by the right cancellation law, $f = g$. Hence, θ is an injection. Next we show that θ is also a surjection. If $h \in B_n$, h is the image of an element of A_n. Specifically, h is the image of $h \circ \tau$.

$$\begin{aligned} \theta(h \circ \tau) &= (h \circ \tau) \circ \tau \\ &= h \circ (\tau \circ \tau) \quad \text{Why?} \\ &= h \circ i \quad \text{Why?} \\ &= h \end{aligned}$$

Since θ is a bijection, $|A_n| = |B_n| = \frac{n!}{2}$.

Example 15.3.4: The Sliding Tile Puzzle

Consider the sliding-tile puzzles pictured in Figure 15.3.3 Each numbered square is a tile and the dark square is a gap. Any tile that is adjacent to the gap can slide into the gap. In most versions of this puzzle, the tiles are locked into a frame so that they can be moved only in the manner described above. The object of the puzzle is to arrange the tiles as they appear in Configuration (a). Configurations (b) and (c) are typical starting points. We propose to show why the puzzle can be solved starting with (b), but not with (c).

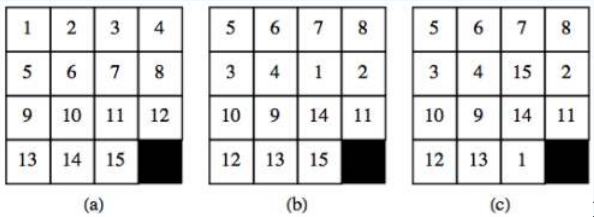


Figure 15.3.3: Configurations of the sliding tile puzzle

We will associate a change in the configuration of the puzzle with an element of S_{16}. Imagine that a tile numbered 16 fills in the gap. For any configuration of the puzzle, the identity i, is the function that leave the configurate "as is." In general, if $f \in S_{16}$, and $1 \leq k \leq 16$, $f(k)$ is the position to which the tile in position k is moved by f that appears in the position of k in configuration (a). If we call the functions that, starting with configuration (a), result in configurations (b) and (c) by the names f_1 and f_2, respectively,

$$f_1 = (1, 5, 3, 7)(2, 6, 4, 8)(9, 10)(11, 14, 13, 12)(15)(16)$$

and

$$f_2 = (1,5,3,7,15)(2,6,4,8)(9,10)(11,14,13,12)(16).$$

How can we interpret the movement of one tile as a permutation? Consider what happens when the 12 tile of i slides into the gap. The result is a configuration that we would interpret as $(12,16)$, a single transposition. Now if we slide the 8 tile into the 12 position, the result is or $(8,16,12)$. Hence, by "exchanging" the tiles 8 and 16, we have implemented the function $(8,12)(12,16) = (8,12,16)$.

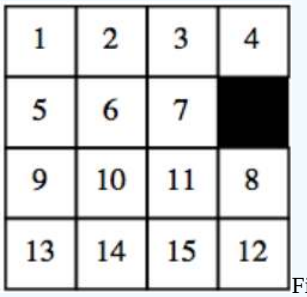


Figure 15.3.4: The configuration $(8,12,16)$

Every time you slide a tile into the gap, the new permutation is a transposition composed with the old permutation. Now observe that to start with initial configuration and terminate after a finite number of moves with the gap in its original position, you must make an even number of moves. Thus, configuration corresponding any permutation that leaves 16 fixed cannot be solved if the permutation is odd. Note that f_2 is an odd permutation; thus, Puzzle (c) can't be solved. The proof that all even permutations, such as f_1, can be solved is left to the interested reader to pursue.

15.3.4: Dihedral Groups

Observation 15.3.2: Realization of Groups

By now we've seen several instances where a group can appear through an isomorphic copy of itself in various settings. The simplest such example is the cyclic group of order 2. When this group is mentioned, we might naturally think of the group $[\mathbb{Z}_2; +_2]$, but the groups $[\{-1,1\}; \cdot]$ and $[S_2; \circ]$ are isomorphic to it. None of these groups are necessarily more natural or important than the others. Which one you use depends on the situation you are in and all are referred to as ***realizations*** of the cyclic group of order 2. The next family of groups we will study, the dihedral groups, has two natural realizations, first as permutations and second as geometric symmetries.

The family of dihedral groups is indexed by the positive integers greater than or equal to 3. For $k \geq 3$, $\mathcal{D}_k$ will have $2k$ elements. We first describe the elements and the operation on them using geometry.

We can describe $\mathcal{D}_n$ in terms of symmetries of a regular n-gon ($n = 3$: equilateral triangle, $n = 4$: square, $n = 5$: regular pentagon,. . .). Here we will only concentrate on the case of $\mathcal{D}_4$. If a square is fixed in space, there are several motions of the square that will, at the end of the motion, not change the apparent position of the square. The actual changes in position can be seen if the corners of the square are labeled. In Figure 15.3.5, the initial labeling scheme is shown, along with the four axes of symmetry of the square.

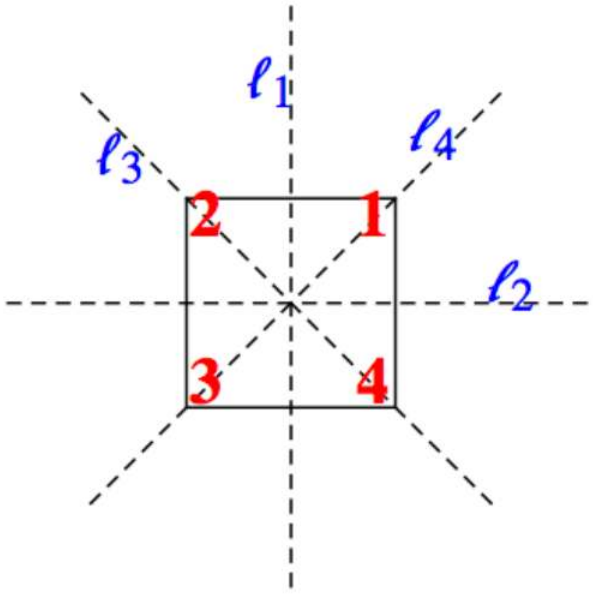


Figure 15.3.5: Axes of symmetry of the square

It might be worthwhile making a square like this with a sheet of paper. Be careful to label the back so that the numbers match the front. Two motions of the square will be considered equivalent if the square is in the same position after performing either motion. There are eight distinct motions. The first four are 0°, 90°, 180°, and 270° clockwise rotations of the square, and the other four are the 180° flips along the axes l_1, l_2, l_3, and l_4. We will call the rotations i, r_1, r_2, and r_3, respectively, and the flips f_1, f_2, f_3, and f_4, respectively. Figure 15.3.6 illustrates r_1 and f_1. For future reference, we also include the permutations to which they correspond.

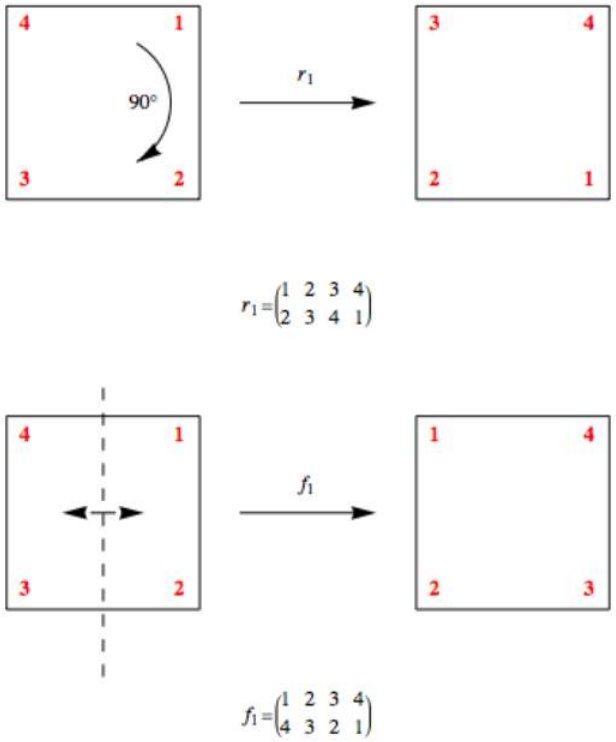


Figure 15.3.6: Two elements of $\mathcal{D}_4$

What is the operation on this set of symmetries? We will call the operation "followed by" and use the symbol $*$ to represent it. The operation will be to combine motions, applying motions from right to left, as with functions. We will illustrate how $*$ is computed by finding $r_1 * f_1$. Starting with the initial configuration, if you perform the f_1 motion, and then immediately perform r_1 on the result, we get the same configuration as if we just performed f_4, which is to flip the square along the line l_4. Therefore, $r_1 * f_1 = f_4$. An important observation is that $f_1 * r_1 \neq f_4$, meaning that this group is nonabelian. The reader is encouraged to verify this on their own.

We can also realize the dihedral groups as permutations. For any symmetric motion of the square we can associate with it a permutation. In the case of $\mathcal{D}_4$, the images of each of the numbers 1 through 4 are the positions on the square that each of the corners 1 through 4 are moved to. For example, since corner 4 moves to position 1 when you perform r_1, the corresponding function will map 4 to 1. In addition, 1 gets mapped to 2, 2 to 3 and 3 to 4. Therefore, r_1 is the cycle $(1, 2, 3, 4)$. The flip f_1 transposes two pairs of corners and corresponds to $(1, 4)(2, 3)$. If we want to combine these two permutations, using the same names as with motions, we get

$$r_1 \circ f_1 = (1, 2, 3, 4) \circ (1, 4)(2, 3) = (1)(2, 4)(3) = (2, 4)$$

Notice that this permutation corresponds with the flip f_4.

Although $\mathcal{D}_4$ isn't cyclic (since it isn't abelian), it can be generated from the two elements r_1 and f_1:

$$\mathcal{D}_4 = \langle r_1, f_1 \rangle = \left\{ i, r_1, {r_1}^2, {r_1}^3, f_1, r_1 \circ f_1, {r_1}^2 \circ f_1, {r_1}^3 \circ f_1 \right\}$$

It is quite easy to describe any of the dihedral groups in a similar fashion. Here is the formal definition

Definition 15.3.4: Dihedral Group

Let n be a positive integer greater than or equal to 3. If $r = (1, 2, \ldots, n)$, an n-cycle, and $f = (1, n)(2, n-1)\ldots$ Then

$$\mathcal{D}_n = \langle r, f \rangle = \left\{ i, r, r^2, \ldots, r^{n-1}, f, r \circ f, r^2 \circ f, \ldots, r^{n-1} \circ f \right\}$$

is the nth dihedral group.

Note 15.3.4: Caution

You might notice that we use a script D, $\mathcal{D}$, for the dihedral groups. Occasionally you might see an ordinary D in other sources for the dihedral groups. Don't confuse it with the set of divisors of n, which we denote by D_n. Normally the context of the discussion should make the meaning of D_n clear.

Example 15.3.5: A Letter-Facing Machine

An application of $\mathcal{D}_4$ is in the design of a letter-facing machine. Imagine letters entering a conveyor belt to be postmarked. They are placed on the conveyor belt at random so that two sides are parallel to the belt. Suppose that a postmarker can recognize a stamp in the top right corner of the envelope, on the side facing up. In Figure 15.3.7, a sequence of machines is shown that will recognize a stamp on any letter, no matter what position in which the letter starts. The letter P stands for a postmarker. The letters R and F stand for rotating and flipping machines that perform the motions of r_1 and f_1.

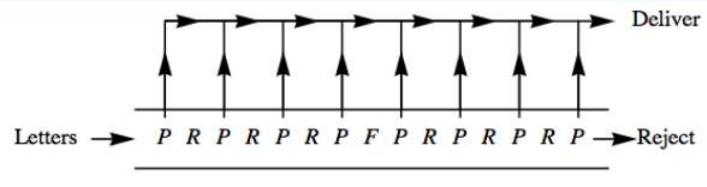


Figure 15.3.7: A letter facer

The arrows pointing up indicate that if a letter is postmarked, it is taken off the conveyor belt for delivery. If a letter reaches the end, it must not have a stamp. Letter-facing machines like this have been designed (see [16]). One economic consideration is that R-machines tend to cost more than F-machines. R-machines also tend to damage more letters. Taking these facts into consideration, the reader is invited to design a better letter-facing machine. Assume that R-machines cost \$800 and F-machines cost \$500. Be sure that all corners of incoming letters will be examined as they go down the conveyor belt.

15.3.5: Exercises

Exercise 15.3.1

Given $f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$, $g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$, and $h = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}$, compute

a. $f \circ g$
b. $g \circ h$
c. $(f \circ g) \circ h$
d. $f \circ (g \circ h)$
e. h^{-1}
f. $h^{-1} \circ g \circ h$
g. f^{-1}

Answer

a. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$
b. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix}$

c. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$
d. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$
e. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$
f. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$
g. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$

Exercise 15.3.2

Write f, g, and h from Exercise 15.3.1 as products of disjoint cycles and determine whether each is odd or even.

Exercise 15.3.3

Do the left cosets of $A_3 = \{i, r_1, r_2\}$ over S_3 form a group under the induced operation on left cosets of A_3? What about the left cosets of $\langle f_1 \rangle$?

Answer

S_3/A_3 is a group of order two. The operation on left cosets of $H = \langle f_1 \rangle$ is not well defined and so a group cannot be formed from left cosets of H.

Exercise 15.3.4

In its realization as permutations, the dihedral group $\mathcal{D}_3$ is equal to S_3. Can you give a geometric explanation why? Why isn't $\mathcal{D}_4$ equal to S_4?

Exercise 15.3.5

a. Complete the list of elements of $\mathcal{D}_4$ and write out a table for the group in its realization as symmetries.
b. List the subgroups of $\mathcal{D}_4$ in a lattice diagram. Are they all cyclic? To what simpler groups are the subgroups of $\mathcal{D}_4$ isomorphic?

Answer

$\mathcal{D}_4 = \{i, r, r^2, r^3, f_1, f_2, f_3, f_4\}$ Where i is the identity function, $r = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$, and

$$f_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix} \quad f_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$
$$f_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix} \quad f_4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$$

The operation table for the group is

$\circ$	i	r	r^2	r^3	f_1	f_2	f_3	f_4
i	i	r	r^2	r^3	f_1	f_2	f_3	f_4
r	r	r^2	r^3	i	f_4	f_3	f_1	f_2
r^2	r^2	r^3	i	r	f_2	f_1	f_4	f_3
r^3	r^3	i	r	r^2	f_3	f_4	f_2	f_1
f_1	f_1	f_3	f_2	f_4	i	r^2	r	r^3
f_2	f_2	f_4	f_1	f_3	r^2	i	r^3	r
f_3	f_3	f_2	f_4	f_1	r^3	r	i	r^2
f_4	f_4	f_1	f_3	f_2	r	r^3	r^2	i

A lattice diagram of its subgroups is

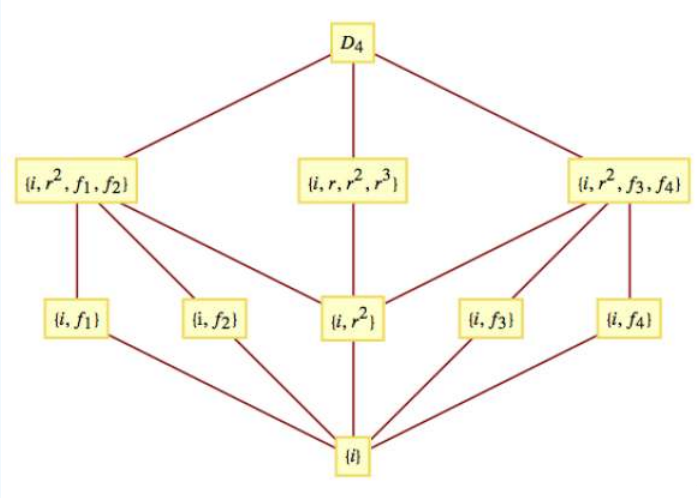


Figure 15.3.8: Subgroups of $\mathcal{D}_4$

All proper subgroups are cyclic except $\{i, r^2, f_1, f_2\}$ and $\{i, r^2, f_3, f_4\}$. Each 2-element subgroup is isomorphic to $\mathbb{Z}_2$; $\{i, r, r^2, r^3\}$ is isomorphic to $\mathbb{Z}_4$; and $\{i, r^2, f_1, f_2\}$ and $\{i, r^2, f_3, f_4\}$ are isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

Exercise 15.3.6

Design a better letter-facing machine (see Example 15.3.5). How can you verify that a letter-facing machine does indeed check every corner of a letter? Can it be done on paper without actually sending letters through it?

Exercise 15.3.7

Prove by induction that if $r \geq 1$ and each t_i, is a transposition, then $(t_1 \circ t_2 \circ \cdots \circ t_r)^{-1} = t_r \circ \cdots \circ t_2 \circ t_1$

Answer

One solution is to cite Exercise 11.3.3 at the end of Section 11.3. It can be directly applied to this problem. An induction proof of the problem at hand would be almost identical to the proof of the more general statement.

$(t_1 t_2 \cdots t_r)^{-1} = t_r^{-1} \cdots t_2^{-1} t_1^{-1}$ by Exercise 11.3.3 of Section 11.3

$= t_r \cdots t_2 t_1$ since each transposition inverts itself. ■

Exercise 15.3.8

How many elements are there in $\mathcal{D}_5$? Describe them geometrically.

Exercise 15.3.9

Complete the proof of Theorem 15.3.1.

Answer

Part I: That $|S_k| = k!$ follows from the Rule of Products.

Part II: Let f be the function defined on $\{1, 2, \ldots, n\}$by $f(1) = 2$, $f(2) = 3$, $f(3) = 1$, and $f(j) = j$ for $4 \leq j \leq n$; and let g be defined by $g(1) = 1$, $g(2) = 3$, $g(3) = 2$, and $g(j) = j$ for $4 \leq j \leq n$. Note that f and g are elements of S_n. Next, $(f \circ g)(1) = f(g(1)) = f(1) = 2$, while $(g \circ f)(1) = g(f(1)) = g(2) = 3$, hence $f \circ g \neq g \circ f$ and S_n is non-abelian for any $n \geq 3$.

Exercise 15.3.10

How many left cosets does A_n, $n \geq 2$ have?

Exercise 15.3.11

Prove that $f \circ r = r^{n-1} \circ f$ in $\mathcal{D}_n$.

Exercise 15.3.12

a. Prove that the tile puzzles corresponding to $A_{16} \cap \{f \in S_{16} | \, f(16) = 16\}$ are solvable.
b. If $f(16) \neq 16$, how can you determine whether f's puzzle is solvable?

Exercise 15.3.13

a. Prove that S_3 is isomorphic to R_3, the group of 3×3 rook matrices (see Section 11.2 exercises).
b. Prove that for each $n \geq 2$, R_n is isomorphic to S_n.

Answer

a. Both groups are non-abelian and of order 6; so they must be isomorphic, since only one such group exists up to isomorphism. The function $\theta : S_3 \to R_3$ defined by

$$\begin{array}{ll} \theta(i) = I & \theta(f_1) = F_1 \\ \theta(r_1) = R_1 & \theta(f_2) = F_2 \\ \theta(r_2) = R_2 & \theta(f_3) = F_3 \end{array}$$

is an isomorphism,

b. Recall that since every function is a relation, it is natural to translate functions to Boolean matrices. Suppose that $f \in S_n$. We will define its image, $\theta(f)$, by

$$\theta(f)_{kj} = 1 \Leftrightarrow f(j) = k$$

That θ is a bijection follows from the existence of θ^{-1}. If A is a rook matrix,

$$\begin{aligned} \theta^{-1}(A)(j) = k &\Leftrightarrow \text{The 1 in column } j \text{ of } A \text{ appears in row } k \\ &\Leftrightarrow A_{kj} = 1 \end{aligned}$$

For $f, g \in S_n$,

$$\begin{aligned} \theta(f \circ g)_{kj} = 1 &\Leftrightarrow (f \circ g)(j) = k \\ &\Leftrightarrow \exists l \text{ such that } g(j) = l \text{ and } f(l) = k \\ &\Leftrightarrow \exists l \text{ such that } \theta(g)_{lj} = 1 \text{ and } \theta(f)_{kl} = 1 \\ &\Leftrightarrow (\theta(f)\theta(g))_{kj} = 1 \end{aligned}$$

Therefore, θ is an isomorphism.

This page titled 15.3: Permutation Groups is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

15.4: Normal Subgroups and Group Homomorphisms

Our goal in this section is to answer an open question from earlier in this chapter and introduce a related concept. The question is: When are left cosets of a subgroup a group under the induced operation? This question is open for non-abelian groups. Now that we have some examples to work with, we can try a few experiments.

15.4.1: Normal Subgroups

Example 15.4.1: Cosets of A_3

We have seen that $A_3 = \{i, r_1, r_2\}$ is a subgroup of S_3, and its left cosets are A_3 itself and $B_3 = \{f_1, f_2, f_3\}$. Whether $\{A_3, B_3\}$ is a group boils down to determining whether the induced operation is well defined. Consider the operation table for S_3 in Figure 15.4.1.

$\circ$	i	r_1	r_2	f_1	f_2	f_3
i	i	r_1	r_2	f_1	f_2	f_3
r_1	r_1	r_2	i	f_3	f_1	f_2
r_2	r_2	i	r_1	f_2	f_3	f_1
f_1	f_1	f_2	f_3	i	r_1	r_2
f_2	f_2	f_3	f_1	r_2	i	r_1
f_3	f_3	f_1	f_2	r_1	r_2	i

Figure 15.4.1: Operation table for S_3

We have shaded in all occurrences of the elements of B_3 in gray. We will call these elements the gray elements and the elements of A_3 the white ones.

Now consider the process of computing the coset product $A_3 \circ B_3$. The "product" is obtained by selecting one white element and one gray element. Note that white "times" gray is always gray. Thus, $A_3 \circ B_3$ is well defined. Similarly, the other three possible products are well defined. The table for the factor group S_3/A_3 is

$\circ$	A_3	B_3
A_3	A_3	B_3
B_3	B_3	A_3

Clearly, S_3/A_3 is isomorphic to $\mathbb{Z}_2$. Notice that A_3 and B_3 are also the right cosets of A_3. This is significant.

Example 15.4.2: Cosets of Another Subgroup of S_3

Now let's try the left cosets of $\langle f_1 \rangle$ in S_3. There are three of them. Will we get a complicated version of $\mathbb{Z}_3$? The left cosets are $C_0 = \langle f_1 \rangle$, $C_1 = r_1 \langle f_1 \rangle = \{r_1, f_3\}$, and $C_2 = r_2 \langle f_1 \rangle = \{r_2, f_2\}$.

The reader might be expecting something to go wrong eventually, and here it is. To determine $C_1 \circ C_2$ we can choose from four pairs of representatives:

$$r_1 \in C_1, r_2 \in C_2 \longrightarrow r_1 \circ r_2 = i \in C_0$$
$$r_1 \in C_1, f_2 \in C_2 \longrightarrow r_1 \circ f_2 = f \in C_0$$
$$f_3 \in C_1, r_2 \in C_2 \longrightarrow f_3 \circ r_2 = f_2 \in C_2$$
$$f_3 \in C_1, f_2 \in C_2 \longrightarrow f_3 \circ f_2 = r_2 \in C_2$$

This time, we don't get the same coset for each pair of representatives. Therefore, the induced operation is not well defined and no factor group is produced.

Observation 15.4.1

This last example changes our course of action. If we had gotten a factor group from $\{C_0, C_1, C_2\}$, we might have hoped to prove that every collection of left cosets forms a group. Now our question is: How can we determine whether we will get a factor group? Of course, this question is equivalent to: When is the induced operation well defined? There was only one step in the proof of Theorem 15.2.4, where we used the fact that G was abelian. We repeat the equations here:

$$a' * b' = (a * h_1) * (b * h_2) = (a * b) * (h_1 * h_2)$$

since G was abelian.

The last step was made possible by the fact that $h_1 * b = b * h_1$. As the proof continued, we used the fact that $h_1 * h_2$ was in H and so $a' * b'$ is $(a * b) * h$ for some h in H. All that we really needed in the "abelian step" was that $h_1 * b = b * (\text{something in } H) = b * h_3$. Then, since H is closed under G's operation, $h_3 * h_2$ is an element of H. The consequence of this observation is that we define a certain kind of subgroup that guarantees that the inducted operation is well-defined.

Definition 15.4.1: Normal Subgroup

If G is a group, $H \leq G$, then H is a normal subgroup of G, denoted $H \triangleleft G$, if and only if every left coset of H is a right coset of H; i. e. $a * H = H * a \quad \forall a \in G$

Theorem 15.4.1

If $H \leq G$, then the operation induced on left cosets of H by the operation of G is well defined if and only if any one of the following conditions is true:

a. H is a normal subgroup of G.
b. If $h \in H$, $a \in G$, then there exists $h' \in H$ such that $h * a = a * h'$.
c. If $h \in H$, $a \in G$, then $a^{-1} * h * a \in H$.

Proof

We leave the proof of this theorem to the reader.

Be careful, the following corollary is not an "...if and only if..." statement.

Corollary 15.4.1

If $H \leq G$, then the operation induced on left cosets of H by the operation of G is well defined if either of the following two conditions is true.

a. G is abelian.
b. $|H| = \frac{|G|}{2}$.

Example 15.4.3: A Non-Normal Subgroup

The right cosets of $\langle f_1 \rangle \leq S_3$ are $\{i, f_1\}$, $\{r_1 f_2\}$, and $\{r_2, f_3\}$. These are not the same as the left cosets of $\langle f_1 \rangle$. In addition, $f_2{}^{-1} f_1 f_2 = f_2 f_1 f_2 = f_3 \notin \langle f_1 \rangle$. Thus, $\langle f_1 \rangle$ is not normal.

The improper subgroups $\{e\}$ and G of any group G are normal subgroups. $G/\{e\}$ is isomorphic to G. All other normal subgroups of a group, if they exist, are called ***proper normal subgroups***.

Example 15.4.4

By Condition b of Corollary 15.4.1, A_n is a normal subgroup of S_n and S_n/A_n is isomorphic to $\mathbb{Z}_2$.

Example 15.4.5: Subgroups of A_5

A_5, a group in its own right with 60 elements, has many proper subgroups, but none are normal. Although this could be done by brute force, the number of elements in the group would make the process tedious. A far more elegant way to approach the verification of this statement is to use the following fact about the cycle structure of permutations. If $f \in S_n$ is a permutation with a certain cycle structure, $\sigma_1 \sigma_2 \cdots \sigma_k$, where the length of σ_i is ℓ_i, then for any $g \in S_n$, $g^{-1} \circ f \circ g$, which is the conjugate of f by g, will have a cycle structure with exactly the same cycle lengths. For example if we take $f = (1,2,3,4)(5,6)(7,8,9) \in S_9$ and conjugate by $g = (1,3,5,7,9)$,

$$\begin{aligned} g^{-1} \circ f \circ g &= (1,9,7,5,3) \circ (1,2,3,4)(5,6)(7,8,9) \circ (1,3,5,7,9) \\ &= (1,4,9,2)(3,6)(5,8,7) \end{aligned}$$

Notice that the condition for normality of a subgroup H of G is that the conjugate of any element of H by an element of G must be remain in H.

To verify that A_5 has no proper normal subgroups, you can start by cataloging the different cycle structures that occur in A_5 and how many elements have those structures. Then consider what happens when you conjugate these different cycle structures with elements of A_5. An outline of the process is in the exercises.

Example 15.4.6

Let G be the set of two by two invertible matrices of real numbers. That is,

$$G = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a, b, c, d \in \mathbb{R}, ad - bc \neq 0 \right\}$$

We saw in Chapter 11 that G is a group with matrix multiplication.

This group has many subgroups, but consider just two: $H_1 = \left\{ \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} \middle| a \neq 0 \right\}$ and $H_2 = \left\{ \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix} \middle| ad \neq 0 \right\}$. It is fairly simple to apply one of the conditions we have observed for normallity that H_1 a normal subgroup of G, while H_2 is not normal in G.

15.4.2: Homomorphisms

Think of the word isomorphism. Chances are, one of the first images that comes to mind is an equation something like

$$\theta(x * y) = \theta(x) \diamond \theta(y)$$

An isomorphism must be a bijection, but the equation above is the algebraic property of an isomorphism. Here we will examine functions that satisfy equations of this type.

Definition 15.4.2: Homomorphism

Let $[G; *]$ and $[G'; \diamond]$ be groups. $\theta : G \to G'$ is a homomorphism if $\theta(x * y) = \theta(x) \diamond \theta(y)$ for all $x, y \in G$.

Many homomorphisms are useful since they point out similarities between the two groups (or, on the universal level, two algebraic systems) involved.

Example 15.4.7: Decreasing Modularity

Define $\alpha : \mathbb{Z}_6 \to \mathbb{Z}_3$ by $\alpha(n) = n \bmod 3$. Therefore, $\alpha(0) = 0$, $\alpha(1) = 1$, $\alpha(2) = 2$, $\alpha(3) = 1+1+1 = 0$, $\alpha(4) = 1$, and $\alpha(5) = 2$. If $n, m \in \mathbb{Z}_6$. We could actually show that α is a homomorphism by checking all $6^2 = 36$ different cases for the formula

$$\alpha(n +_6 m) = \alpha(n) +_3 \alpha(m) \tag{15.4.1}$$

but we will use a line of reasoning that generalizes. We have already encountered the Chinese Remainder Theorem, which implies that the function $\beta : \mathbb{Z}_6 \to \mathbb{Z}_3 \times \mathbb{Z}_2$ defined by $\beta(n) = (n \bmod 3, n \bmod 2)$. We need only observe that equating the first coordinates of both sides of the equation

$$\beta(n +_6 m) = \beta(n) + \beta(m) \tag{15.4.2}$$

gives us precisely the homomorphism property.

Theorem 15.4.2: Group Homomorphism Properties

If $\theta : G \to G'$ is a homomorphism, then:

a. $\theta(e) = \theta(\text{the identity of } G) = \text{the identity of } G' = e'$.
b. $\theta\left(a^{-1}\right) = \theta(a)^{-1}$ for all $a \in G$.
c. If $H \leq G$, then $\theta(H) = \{\theta(h) | h \in H\} \leq G'$.

Proof

a. Let a be any element of G. Then $\theta(a) \in G'$.

$$\begin{aligned} \theta(a) \diamond e' &= \theta(a) && \text{by the definition of } e' \\ &= \theta(a * e) && \text{by the definition of } e \\ &= \theta(a) \diamond \theta(e) && \text{by the fact that } \theta \text{ is a homomorphism} \end{aligned}$$

By cancellation, $e' = \theta(e)$.

b. Again, let $a \in G$. $e' = \theta(e) = \theta\left(a * a^{-1}\right) = \theta(a) \diamond \theta\left(a^{-1}\right)$. Hence, by the uniqueness of inverses, $\theta(a)^{-1} = \theta\left(a^{-1}\right)$.

c. Let $b_1, b_2 \in \theta(H)$. Then there exists $a_1, a_2 \in H$ such that $\theta(a_1) = b_1$, $\theta(a_2) = b_2$. Recall that a compact necessary and sufficient condition for $H \leq G$ is that $x * y^{-1} \in H$ for all $x, y \in H$. Now we apply the same condition in G':

$$\begin{aligned} b_1 \diamond {b_2}^{-1} &= \theta(a_1) \diamond \theta(a_2)^{-1} \\ &= \theta(a_1) \diamond \theta\left({a_2}^{-1}\right) \\ &= \theta\left(a_1 * {a_2}^{-1}\right) \in \theta(H) \end{aligned}$$

since $a_1 * {a_2}^{-1} \in H$, and so we can conclude that $\theta(H) \leq G'$.

Corollary 15.4.2

Since a homomorphism need not be a surjection and part (c) of Theorem 15.4.2is true for the case of $H = G$, the range of θ, $\theta(G)$, is a subgroup of G'

Example 15.4.8

If we define $\pi : \mathbb{Z} \to \mathbb{Z}/4\mathbb{Z}$ by $\pi(n) = n + 4\mathbb{Z}$, then π is a homomorphism. The image of the subgroup $4\mathbb{Z}$ is the single coset $0 + 4\mathbb{Z}$, the identity of the factor group. Homomorphisms of this type are called natural homomorphisms. The following theorems will verify that π is a homomorphism and also show the connection between homomorphisms and normal subgroups. The reader can find more detail and proofs in most abstract algebra texts.

Theorem 15.4.3

If $H \triangleleft G$, then the function $\pi : G \to G/H$ defined by $\pi(a) = aH$ is a homomorphism.

Proof

We leave the proof of this theorem to the reader.

Definition 15.4.3: Natural Homomorphism

If $H \triangleleft G$, then the function $\pi : G \to G/H$ defined by $\pi(a) = aH$ is called the natural homomorphism.

Based on Theorem 15.4.3, every normal subgroup gives us a homomorphism. Next, we see that the converse is true.

Definition 15.4.4: Kernel of a Homomorphism

Let $\theta : G \to G'$ be a homomorphism, and let e and e' be the identities of G and G', respectively. The kernel of θ is the set $\ker\theta = \{a \in G \mid \theta(a) = e'\}$

Theorem 15.4.4

Let $\theta : G \to G'$ be a homomorphism from G into G'. The kernel of θ is a normal subgroup of G.

Proof

Let $K = \ker\theta$. We can see that K is a subgroup of G by letting $a, b \in K$ and verify that $a * b^{-1} \in K$ by computing $\theta(a * b^{-1}) = \theta(a) * \theta(b)^{-1} = e' * e'^{-1} = e'$. To prove normality, we let g be any element of G and $k \in K$. We compute $\theta(g * k * g^{-1})$ to verify that $g * k * g^{-1} \in K$.

$$\begin{aligned}\theta(g * k * g^{-1}) &= \theta(g) * \theta(k) * \theta(g^{-1}) \\ &= \theta(g) * \theta(k) * \theta(g)^{-1} \\ &= \theta(g) * e' * \theta(g)^{-1} \\ &= \theta(g) * \theta(g)^{-1} \\ &= e'\end{aligned}$$

Based on this most recent theorem, every homomorphism gives us a normal subgroup.

Theorem 15.4.5: Fundamental Theorem of Group Homomorphisms

Let $\theta : G \to G'$ be a homomorphism. Then $\theta(G)$ is isomorphic to $G/\ker\theta$.

Example 15.4.9

Define $\theta : \mathbb{Z} \to \mathbb{Z}_{10}$ by $\theta(n) = n \bmod 10$. The three previous theorems imply the following:

- $\pi : \mathbb{Z} \to \mathbb{Z}/10\mathbb{Z}$ defined by $\pi(n) = n + 10\mathbb{Z}$ is a homomorphism.
- $\{n \in \mathbb{Z} | \theta(n) = 0\} = \{10n \mid n \in \mathbb{Z}\} = 10\mathbb{Z} \triangleleft \mathbb{Z}$.
- $\mathbb{Z}/10\mathbb{Z}$ is isomorphic to $\mathbb{Z}_{10}$.

Example 15.4.10

Let G be the same group of two by two invertible real matrices as in Example 15.4.6. Define $\Phi : G \to G$ by $\Phi(A) = \frac{A}{\sqrt{|\det A|}}$. We will let the reader verify that Φ is a homomorphism. The theorems above imply the following.

- $\ker \Phi = \{A \in G | \Phi(A) = I\} = \left\{ \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} \mid a \in \mathbb{R}, a \neq 0 \right\} \triangleleft G$. This verifies our statement in Example 15.4.6. As in that example, let $\ker \Phi = H_1$.
- G/H_1 is isomorphic to $\{A \in G \mid \det A = 1\}$.
- $\pi : G \to G/H_1$ defined, naturally, by $\pi(A) = AH_1$ is a homomorphism.

For the remainder of this section, we will be examining certain kinds of homomorphisms that will play a part in our major application to homomorphisms, coding theory.

Example 15.4.11

Consider $\Phi : {\mathbb{Z}_2}^2 \to {\mathbb{Z}_2}^3$ defined by $\Phi(a,b) = (a, b, a +_2 b)$. If $(a_1, b_1), (a_2, b_2) \in {\mathbb{Z}_2}^2$,

$$\begin{aligned} \Phi\left((a_1, b_1) + (a_2, b_2)\right) &= \Phi\left(a_1 +_2 a_2, b_1 +_2 b_2\right) \\ &= \left(a_1 +_2 a_2, b_1 +_2 b_2, a_1 +_2 a_2 +_2 b_1 +_2 b_2\right) \\ &= (a_1, b_1, a_1 +_2 b_1) + (a_2, b_2, a_2 +_2 b_2) \\ &= \Phi\left(a_1, b_1\right) + \Phi\left(a_2, b_2\right) \end{aligned}$$

Since $\Phi(a,b)=(0,0,0)$ implies that $a = 0$ and $b = 0$, the kernel of Φ is $\{(0,0)\}$. By previous theorems, $\Phi\left({\mathbb{Z}_2}^2\right) = \{(0,0,0), (1,0,1), (0,1,1), (1,1,0)\}$ is isomorphic to ${\mathbb{Z}_2}^2$.

We can generalize the previous example as follows: If $n, m \geq 1$ and A is an $m \times n$ matrix of 0's and 1's (elements of $\mathbb{Z}_2$), then $\Phi : {\mathbb{Z}_2}^m \to {\mathbb{Z}_2}^n$ defined by

$$\Phi\left(a_1, a_2, \ldots, a_m\right) = \left(a_1, a_2, \ldots, a_m\right) A$$

is a homomorphism. This is true because matrix multiplication is distributive over addition. The only new idea here is that computation is done in $\mathbb{Z}_2$. If $a = (a_1, a_2, \ldots, a_m)$ and $b = (b_1, b_2, \ldots, b_m)$, $(a+b)A = aA + bA$ is true by basic matrix laws. Therefore, $\Phi(a+b) = \Phi(a) + \Phi(b)$.

15.4.3: Exercises

Exercise 15.4.1

Which of the following functions are homomorphisms? What are the kernels of those functions that are homomorphisms?

a. $\theta_1 : \mathbb{R}^* \to \mathbb{R}^+$ defined by $\theta_1(a) = |a|$.

b. $\theta_2 : \mathbb{Z}_5 \to \mathbb{Z}_2$ where $\theta_2(n) = \begin{cases} 0 & \text{if } n \text{ is even} \\ 1 & \text{if } n \text{ is odd} \end{cases}$.

c. $\theta_3 : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, where $\theta_3(a,b) = a + b$.

d. $\theta_4 : S_4 \to S_4$ defined by $\theta_4(f) = f \circ f = f^2$.

Answer

a. Yes, the kernel is $\{1, -1\}$

b. No, since $\theta_2\left(2 +_5 4\right) = \theta_2(1) = 1$, but $\theta_2(2) +_2 \theta_2(4) = 0 +_2 0 = 0$
A follow-up might be to ask what happens if 5 is replaced with some other positive integer in this part.

c. Yes, the kernel is $\{(a, -a) | a \in \mathbb{R}\}$

d. No. A counterexample, among many, would be to consider the two transpositions $t_1 = (1,3)$ and $t_2 = (1,2)$. Compare $\theta_4(t_1 \circ t_2)$ and $\theta_4(t_1) \circ \theta_4(t_2)$.

Exercise 15.4.2

Which of the following functions are homomorphisms? What are the kernels of those functions that are homomorphisms?

a. $\alpha_1 : M_{2\times 2}(\mathbb{R}) \to \mathbb{R}$, defined by $\alpha_1(A) = A_{11}A_{22} + A_{12}A_{21}$.
b. $\alpha_2 : (\mathbb{R}^*)^2 \to \mathbb{R}^*$ defined by $\alpha_2(a, b) = ab$.
c. $\alpha_3 : \{A \in M_{2\times 2}(\mathbb{R}) | \det A \neq 0\} \to \mathbb{R}^*$, where $\alpha_3(A) = \det A$.
d. $\alpha_4 : S_4 \to S_4$ defined by $\alpha_4(f) = f^{-1}$.

Exercise 15.4.3

Show that D_4 has one proper normal subgroup, but that $\langle (1, 4)(2, 3)\rangle$ is not normal.

Answer

$\langle r\rangle = \{i, r, r^2, r^3\}$ is a normal subgroup of D_4. To see you could use the table given in the solution of Exercise 15.3.5 of Section 15.3 and verify that $a^{-1}ha \in \langle r\rangle$ for all $a \in D_4$ and $h \in \langle r\rangle$. A more efficient approach is to prove the general theorem that if H is a subgroup G with exactly two distinct left cosets, than H is normal. $\langle f_1\rangle$ is not a normal subgroup of D_4. $\langle f_1\rangle = \{i, f_1\}$ and if we choose $a = r$ and $h = f_1$ then $a^{-1}ha = r^3 f_1 r = f_2 \notin \langle f_1\rangle$

Exercise 15.4.4

Prove that the function Φ in Example 15.4.10is a homomorphism.

Exercise 15.4.5

Define the two functions $\alpha : {\mathbb{Z}_2}^3 \to {\mathbb{Z}_2}^4$ and $\beta : {\mathbb{Z}_2}^4 \to \mathbb{Z}_2$ by $\alpha(a_1, a_2, a_3) = (a_1, a_2, a_3, a_1 +_2 a_2 +_2 a_3)$, and $\beta(b_1, b_2, b_3, b_4) = b_1 + b_2 + b_3 + b_4$ Describe the function $\beta \circ \alpha$. Is it a homomorphism?

Answer

$(\beta \circ \alpha)(a_1, a_2, a_3) = 0$ and so $\beta \circ \alpha$ is the trivial homomorphism, but a homomorphism nevertheless.

Exercise 15.4.6

Express Φ in Example 15.4.10in matrix form.

Exercise 15.4.7

Prove that if G is an abelian group, then $q(x) = x^2$ defines a homomorphism from G into G. Is q ever an isomorphism?

Answer

Let $x, y \in G$.

$$\begin{aligned} q(x * y) &= (x * y)^2 \\ &= x * y * x * y \\ &= x * x * y * y \quad \text{since } G \text{ is abelian} \\ &= x^2 * y^2 \\ &= q(x) * q(y) \end{aligned}$$

Hence, q is a homomorphism. In order for q to be an isomorphism, it must be the case that no element other than the identity is its own inverse.

$$\begin{aligned} x \in \text{Ker}(q) &\Leftrightarrow q(x) = e \\ &\Leftrightarrow x * x = e \\ &\Leftrightarrow x^{-1} = x \end{aligned}$$

Exercise 15.4.8

Prove that if $\theta : G \to G'$ is a homomorphism, and $H \triangleleft G$, then $\theta(H) \triangleleft \theta(G)$. Is it also true that $\theta(H) \triangleleft G'$?

Exercise 15.4.9

Prove that if $\theta : G \to G'$ is a homomorphism, and $H' \leq \theta(G)$, then $\theta^{-1}(H') = \{a \in G | \theta(a) \in H'\} \leq G$.

Answer

Proof: Recall that the inverse image of H' under θ is $\theta^{-1}(H') = \{g \in G | \theta(g) \in H'\}$.

Closure: Let $g_1, g_2 \in \theta^{-1}(H')$, then $\theta(g_1), \theta(g_2) \in H'$. Since H' is a subgroup of G',

$$\theta(g_1) \diamond \theta(g_2) = \theta(g_1 * g_2) \in H' \Rightarrow g_1 * g_2 \in \theta^{-1}(H')$$

Identity: By Theorem 15.4.2(a), $e \in \theta^{-1}(H')$.

Inverse: Let $a \in \theta^{-1}(H')$. Then $\theta(a) \in H'$ and by Theorem 15.4.2(b), $\theta(a)^{-1} = \theta\left(a^{-1}\right) \in H'$ and so $a^{-1} \in \theta^{-1}(H')$.

Exercise 15.4.10

Following up on Example 15.4.5, prove that A_5 is a simple group; i. e., it has no proper normal subgroups.

a. Make a list of the different cycle structures that occur in A_5 and how many elements have those structures.
b. Within each set of permutations with different cycle structures, identify which subsets are closed with respect to the conjugation operation. With this you will have a partition of A_5 into conjugate classes where for each class, C, $f, g \in C$ if and only if $\exists \phi \in A_5$ such that $\phi^{-1} \circ f \circ \phi = g$.
c. Use the fact that a normal subgroup of A_5 needs to be a union of conjugate classes and verify that no such union exists.

This page titled 15.4: Normal Subgroups and Group Homomorphisms is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

15.5: Coding Theory, Group Codes

15.5.1: Transmission Problem

In this section, we will introduce the basic ideas involved in coding theory and consider solutions of a coding problem by means of group codes.

Imagine a situation in which information is being transmitted between two points. The information takes the form of high and low pulses (for example, radio waves or electric currents), which we will label 1 and 0, respectively. As these pulses are sent and received, they are grouped together in blocks of fixed length. The length determines how much information can be contained in one block. If the length is r, there are 2^r different values that a block can have. If the information being sent takes the form of text, each block might be a character. In that case, the length of a block may be seven, so that $2^7 = 128$ block values can represent letters (both upper and lower case), digits, punctuation, and so on. During the transmission of data, noise can alter the signal so that what is received differs from what is sent. Figure $15.5.1$ illustrates the problem that can be encountered if information is transmitted between two points.

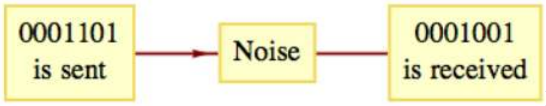


Figure 15.5.1: A noisy transmission

Noise is a fact of life for anyone who tries to transmit information. Fortunately, in most situations, we could expect a high percentage of the pulses that are sent to be received properly. However, when large numbers of pulses are transmitted, there are usually some errors due to noise. For the remainder of the discussion, we will make assumptions about the nature of the noise and the message that we want to send. Henceforth, we will refer to the pulses as bits.

We will assume that our information is being sent along a ***binary symmetric channel***. By this, we mean that any single bit that is transmitted will be received improperly with a certain fixed probability, p, independent of the bit value. The magnitude of p is usually quite small. To illustrate the process, we will assume that $p = 0.001$, which, in the real world, would be considered somewhat large. Since $1 - p = 0.999$, we can expect 99.9% of all bits to be properly received.

Suppose that our message consists of 3,000 bits of information, to be sent in blocks of three bits each. Two factors will be considered in evaluating a method of transmission. The first is the probability that the message is received with no errors. The second is the number of bits that will be transmitted in order to send the message. This quantity is called the rate of transmission:

$$\text{Rate} = \frac{\text{Message length}}{\text{Number of bits transmitted}}$$

As you might expect, as we devise methods to improve the probability of success, the rate will decrease.

Suppose that we ignore the noise and transmit the message without any coding. The probability of success is $0.999^{3000} = 0.0497124$. Therefore we only successfully receive the message in a totally correct form less than 5% of the time. The rate of $\frac{3000}{3000} = 1$ certainly doesn't offset this poor probability.

Our strategy for improving our chances of success will be to send an encoded message across the binary symmetric channel. The encoding will be done in such a way that small errors can be identified and corrected. This idea is illustrated in Figure $15.5.2$

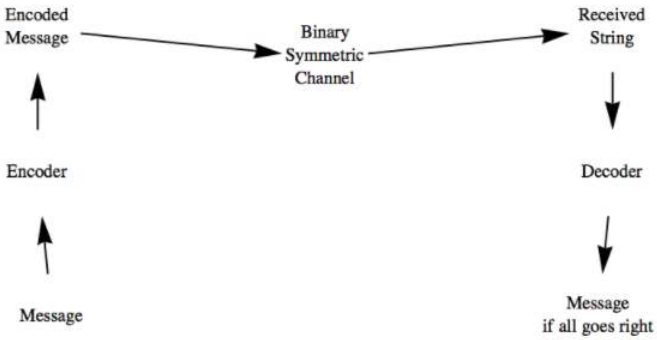


Figure 15.5.2: The Coding Process

In our examples, the functions that will correspond to our encoding and decoding devices will all be homomorphisms between Cartesian products of $\mathbb{Z}_2$.

15.5.2: Error Detection

Suppose that each block of three bits $a = (a_1, a_2, a_3)$ is encoded with the function $e : {\mathbb{Z}_2}^3 \to {\mathbb{Z}_2}^4$, where

$$e(a) = (a_1, a_2, a_3, a_1 +_2 a_2 +_2 a_3)$$

When the encoded block is received, the four bits will probably all be correct (they are correct approximately 99.6% of the time), but the added bit that is sent will make it possible to detect single errors in the block. Note that when $e(a)$ is transmitted, the sum of its components is $a_1 +_2 a_2 +_2 a_3 +_2 (a_1 +_2 a_2 +_2 a_3) = 0, \quad$ since $a_i + a_i = 0$ in $\mathbb{Z}_2$.

If any single bit is garbled by noise, the sum of the received bits will be 1. The last bit of $e(a)$ is called the parity bit. A parity error occurs if the sum of the received bits is 1. Since more than one error is unlikely when p is small, a high percentage of all errors can be detected.

At the receiving end, the decoding function acts on the four-bit block $b = (b_1, b_2, b_3, b_4)$ with the function $d : {\mathbb{Z}_2}^4 \to \mathbb{Z}_2^4$, where

$$d(b) = (b_1, b_2, b_3, b_1 +_2 b_2 +_2 b_3 +_2 b_4)$$

The fourth bit is called the parity-check bit. If no parity error occurs, the first three bits are recorded as part of the message. If a parity error occurs, we will assume that a retransmission of that block can be requested. This request can take the form of automatically having the parity-check bit of $d(b)$ sent back to the source. If 1 is received, the previous block is retransmitted; if 0 is received, the next block is sent. This assumption of two-way communication is significant, but it is desirable to make this coding system useful. It is reasonable to expect that the probability of a transmission error in the opposite direction is also 0.001. Without going into the details, we will report that the probability of success is approximately 0.990 and the rate is approximately 3/5. The rate includes the transmission of the parity-check bit to the source.

15.5.3: Error Correction

Next, we will consider a coding process that can correct errors at the receiving end so that only one-way communication is needed. Before we begin, recall that every element of ${\mathbb{Z}_2}^n$, $n \geq 1$, is its own inverse; that is, $-b = b$. Therefore, $a - b = a + b$.

Noisy three-bit message blocks are difficult to transmit because they are so similar to one another. If a and b are in ${\mathbb{Z}_2}^3$, their difference, $a +_2 b$, can be thought of as a measure of how close they are. If a and b differ in only one bit position, one error can change one into the other. The encoding that we will introduce takes a block $a = (a_1, a_2, a_3)$ and produces a block of length 6 called the ***code word*** of a. The code words are selected so that they are farther from one another than the messages are. In fact, each code word will differ from each other code word by at least three bits. As a result, any single error will not push a code word close enough to another code word to cause confusion. Now for the details.

Let $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$. We call G the generator matrix for the code, and let $a = (a_1, a_2, a_3)$ be our message.

Define $e : {\mathbb{Z}_2}^3 \to {\mathbb{Z}_2}^6$ by

$$e(a) = aG = (a_1, a_2, a_3, a_4, a_5, a_6)$$

where

$$\begin{aligned} a_4 &= a_1 +_2 a_2 \\ a_5 &= a_1 +_2 a_3 \\ a_6 &= a_2 +_2 a_3 \end{aligned}$$

Notice that e is a homomorphism. Also, if a and b are distinct elements of ${\mathbb{Z}_2}^3$, then $c = a + b$ has at least one coordinate equal to 1. Now consider the difference between $e(a)$ and $e(b)$:

$$\begin{aligned} e(a) + e(b) &= e(a + b) \\ &= e(c) \\ &= (d_1, d_2, d_3, d_4, d_5, d_6) \end{aligned}$$

Whether c has 1, 2, or 3 ones, $e(c)$ must have at least three ones. This can be seen by considering the three cases separately. For example, if c has a single one, two of the parity bits are also 1. Therefore, $e(a)$ and $e(b)$ differ in at least three bits.

Now consider the problem of decoding the code words. Imagine that a code word, $e(a)$, is transmitted, and $b=(b_1,b_2,b_3,b_4,b_5,b_6)$ is received. At the receiving end, we know the formula for $e(a)$, and if no error has occurred in transmission,

$$\begin{array}{l} b_1=a_1 \\ b_2=a_2 \\ b_3=a_3 \\ b_4=a_1+_2 a_2 \\ b_5=a_1+_2 a_3 \\ b_6=a_2+_2 a_3 \end{array} \Rightarrow \begin{array}{lllllllllll} b_1 & +_2 & b_2 & & & +_2 & b_4 & & & & =0 \\ b_1 & & & +_2 & b_3 & & & +_2 & b_5 & & =0 \\ & & b_2 & +_2 & b_3 & & & & & +_2 \; b_6 & =0 \end{array}$$

The three equations on the right are called parity-check equations. If any of them are not true, an error has occurred. This error checking can be described in matrix form.

Let

$$P=\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

P is called the parity-check matrix for this code. Now define $p:{\mathbb{Z}_2}^6 \to {\mathbb{Z}_2}^3$ by $p(b)=bP$. We call $p(b)$ the syndrome of the received block. For example, $p(0,1,0,1,0,1)=(0,0,0)$and $p(1,1,1,1,0,0)=(1,0,0)$

Note that p is also a homomorphism. If the syndrome of a block is $(0,0,0)$, we can be almost certain that the message block is (b_1,b_2,b_3).

Next we turn to the method of correcting errors. Despite the fact that there are only eight code words, one for each three-bit block value, the set of possible received blocks is ${\mathbb{Z}_2}^6$, with 64 elements. Suppose that b is not a code word, but that it differs from a code word by exactly one bit. In other words, it is the result of a single error in transmission. Suppose that w is the code word that b is closest to and that they differ in the first bit. Then $b+w=(1,0,0,0,0,0)$ and

$$\begin{aligned} p(b) &= p(b)+p(w) \quad \text{since } p(w)=(0,0,0) \\ &= p(b+w) \quad \text{since } p \text{ is a homomorphism} \\ &= p(1,0,0,0,0,0) \\ &= (1,1,0) \end{aligned}$$

Note that we haven't specified b or w, only that they differ in the first bit. Therefore, if b is received, there was probably an error in the first bit and $p(b)=(1,1,0)$, the transmitted code word was probably $b+(1,0,0,0,0,0)$ and the message block was $(b_1+_2 1,b_2,b_3)$. The same analysis can be done if b and w differ in any of the other five bits.

This process can be described in terms of cosets. Let W be the set of code words; that is, $W=e\left({\mathbb{Z}_2}^3\right)$. Since e is a homomorphism, W is a subgroup of ${\mathbb{Z}_2}^6$. Consider the factor group ${\mathbb{Z}_2}^6/W$:

$$|{\mathbb{Z}_2}^6/W|=\frac{|{\mathbb{Z}_2}^6|}{|W|}=\frac{64}{8}=8$$

Suppose that b_1 and b_2 are representatives of the same coset. Then $b_1=b_2+w$ for some w in W. Therefore,

$$\begin{aligned} p\left(b_1\right) &= p\left(b_1\right)+p(w) \quad \text{since } p(w)=(0,0,0) \\ &= p\left(b_1+w\right) \\ &= p\left(b_2\right) \end{aligned}$$

and so b_1 and b_2 have the same syndrome.

Finally, suppose that d_1 and d_2 are distinct and both have only a single coordinate equal to 1. Then d_1+d_2 has exactly two ones. Note that the identity of ${\mathbb{Z}_2}^6$, $(0,0,0,0,0,0)$,must be in W. Since d_1+d_2 differs from the identity by two bits, $d_1+d_2 \notin W$. Hence d_1 and d_2 belong to distinct cosets. The reasoning above serves as a proof of the following theorem.

Theorem 15.5.1

There is a system of distinguished representatives of ${\mathbb{Z}_2}^6 / W$ such that each of the six-bit blocks having a single 1 is a distinguished representative of its own coset.

Now we can describe the error-correcting process. First match each of the blocks with a single 1 with its syndrome. In addition, match the identity of W with the syndrome $(0,0,0)$ as in the table below. Since there are eight cosets of W, select any representative of the eighth coset to be distinguished. This is the coset with syndrome $(1,1,1)$.

Syndrome			Error Correction					
0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1
1	1	1	1	0	0	0	0	1

When block b is received, you need only compute the syndrome, $p(b)$, and add to b the error correction that matches $p(b)$.

We will conclude this example by computing the probability of success for our hypothetical situation. It is $\left(0.999^6+6\cdot 0.999^5\cdot 0.001\right)^{1000}=0.985151$. The rate for this code is $\frac{1}{2}$.

Example 15.5.1: Another Linear Code

Consider the linear code with generator matrix

$$G=\begin{pmatrix}1&1&0&1&0\\0&1&1&0&0\\0&0&0&1&1\end{pmatrix}.$$

Since G is 3×5, this code encodes three bits into five bits. The natural question to ask is what detection or correction does it afford? We can answer this question by constructing the parity check matrix. We observe that if $\vec{b}=(b_1,b_2,b_3)$ the encoding function is

$$e(\vec{b})=\vec{b}G=(b_1,b_1+b_2,b_2,b_1+b_3,b_3)$$

where addition is mod 2 addition. If we receive five bits (c_1,c_2,c_3,c_4,c_5) and no error has occurred, the following two equations would be true.

$$c_1+c_2+c_3=0 \tag{15.5.1}$$
$$c_1+c_4+c_5=0 \tag{15.5.2}$$

Notice that in general, the number of parity check equations is equal to the number of extra bits that are added by the encoding function. These equations are equivalent to the single matrix equation $(c_1,c_2,c_3,c_4,c_5)H=\vec{0}$, where

$$H=\begin{pmatrix}1&1\\1&0\\1&0\\0&1\\0&1\end{pmatrix}$$

At a glance, we can see that this code will not correct most single bit errors. Suppose an error $\vec{e} = (e_1, e_2, e_3, e_4, e_5)$ is added in the transmission of the five bits. Specifically, suppose that 1 is added (mod 2) in position j, where $1 \leq j \leq 5$ and the other coordinates of $\vec{e}$ are 0. Then when we compute the syndrome of our received transmission, we see that

$$\vec{c}H = (\vec{b}G + \vec{e})H = (\vec{b}G)H + \vec{e}H = \vec{e}H.$$

But $\vec{e}H$ is the jth row of H. If the syndrome is $(1, 1)$ we know that the error occurred in position 1 and we can correct it. However, if the error is in any other position we can't pinpoint its location. If the syndrome is $(1, 0)$, then the error could have occurred in either position 2 or position 3. This code does detect all single bit errors but only corrects one fifth of them.

15.5.4: Exercises

Exercise 15.5.1

If the error-detecting code is being used, how would you act on the following received blocks?

a. $(1, 0, 1, 1)$
b. $(1, 1, 1, 1)$
c. $(0, 0, 0, 0)$

Answer

a. Error detected, since an odd number of 1's was received; ask for retransmission.
b. No error detected; accept this block.
c. No error detected; accept this block.

Exercise 15.5.2

Express the encoding and decoding functions for the error-detecting code using matrices.

Exercise 15.5.3

If the error-correcting code from this section is being used, how would you decode the following blocks? Expect an error that cannot be fixed with one of these.

a. $(1, 0, 0, 0, 1, 1)$
b. $(1, 0, 1, 0, 1, 1)$
c. $(0, 1, 1, 1, 1, 0)$
d. $(0, 0, 0, 1, 1, 0)$
e. $(1, 0, 0, 0, 0, 1)$
f. $(1, 0, 0, 1, 0, 0)$

Answer

a. Syndrome = $(1, 0, 1)$. Corrected coded message is $(1, 1, 0, 0, 1, 1)$ and original message was $(1, 1, 0)$.
b. Syndrome = $(1, 1, 0)$. Corrected coded message is $(0, 0, 1, 0, 1, 1)$ and original message was $(0, 0, 1)$.
c. Syndrome = $(0, 0, 0)$. No error, coded message is $(0, 1, 1, 1, 1, 0)$ and original message was $(0, 1, 1)$.
d. Syndrome = $(1, 1, 0)$. Corrected coded message is $(1, 0, 0, 1, 1, 0)$ and original message was $(1, 0, 0)$.
e. Syndrome = $(1, 1, 1)$. This syndrome occurs only if two bits have been switched. No reliable correction is possible.
f. Syndrome = $(0, 1, 0)$. Corrected coded message is $(1, 0, 0, 1, 1, 0)$ and original message was $(1, 0, 0)$.

Exercise 15.5.5

Consider the linear code defined by the generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

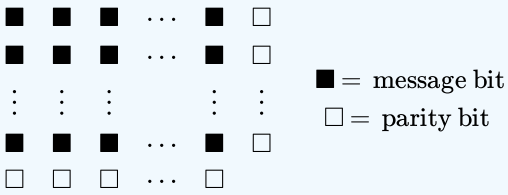
a. What size blocks does this code encode and what is the length of the code words?
b. What are the code words for this code?
c. With this code, can you detect single bit errors? Can you correct all, some, or no single bit errors?

Answer

Let G be the 9×10 matrix obtained by augmenting the 9×9 . The function $e : \mathbb{Z}_2{}^9 \to \mathbb{Z}_2{}^{10}$ defined by $e(a) = aG$ will allow us to detect single errors, since $e(a)$ will always have an even number of ones.

Exercise 15.5.6: Rectangular Codes

To build a rectangular code, you partition your message into blocks of length m and then factor m into $k_1 \cdot k_2$ and arrange the bits in a $k_1 \times k_2$ rectangular array as in the figure below. Then you add parity bits along the right side and bottom of the rows and columns. The code word is then read row by row.



For example, if m is 4, then our only choice is a 2 by 2 array. The message 1101 would be encoded as

1	1	0
0	1	1
1	0	

and the code word is the string 11001110.

a. Suppose that you were sent four bit messages using this code and you received the following strings. What were the messages, assuming no more than one error in the transmission of coded data?
   i. 11011000
   ii. 01110010
   iii. 10001111
b. If you encoded n^2 bits in this manner, what would be the rate of the code?
c. Rectangular codes are linear codes. For the 3 by 2 rectangular code, what are the generator and parity check matrices?

Exercise 15.5.7

Suppose that the code in Example 15.5.1is expanded to add the column

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

to the generator matrix G, can all single bit errors be corrected? Explain your answer.

Answer

Yes, you can correct all single bit errors because the parity check matrix for the expanded code is

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Since each possible syndrome of single bit errors is unique we can correct any error.

This page titled 15.5: Coding Theory, Group Codes is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

16: An Introduction to Rings and Fields

> ***field extension***
>
> ***Field extensions*** *are simple. Let's say*
> *That field* L *is a subfield of* K*,*
> *Then it goes without mention,*
> *Field* K*'s an extension*
> *Of* L *— like a shell, in a way.*
>
> *zqms, The Omnificent English Dictionary in Limerick Form*

In our early elementary school days we began the study of mathematics by learning addition and multiplication on the set of positive integers. We then extended this to operations on the set of all integers. Subtraction and division are defined in terms of addition and multiplication. Later we investigated the set of real numbers under the operations of addition and multiplication. Hence, it is quite natural to investigate those structures on which we can define these two fundamental operations, or operations similar to them. The structures similar to the set of integers are called rings, and those similar to the set of real numbers are called fields.

In coding theory, highly structured codes are needed for speed and accuracy. The theory of finite fields is essential in the development of many structured codes. We will discuss basic facts about finite fields and introduce the reader to polynomial algebra.

16.1: Rings, Basic Definitions and Concepts
16.2: Fields
16.3: Polynomial Rings
16.4: Field Extensions
16.5: Power Series

This page titled 16: An Introduction to Rings and Fields is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

16.1: Rings, Basic Definitions and Concepts

16.1.1: Basic Definitions

We would like to investigate algebraic systems whose structure imitates that of the integers.

Definition 16.1.1: Ring

A ring is a set R together with two binary operations, addition and multiplication, denoted by the symbols $+$ and $\cdot$ such that the following axioms are satisfied:

1. $[R; +]$ is an abelian group.
2. Multiplication is associative on R.
3. Multiplication is distributive over addition; that is, for all $a, b, c \in R$, the left distributive law, $a \cdot (b + c) = a \cdot b + a \cdot c$, and the right distributive law, $(b + c) \cdot a = b \cdot a + c \cdot a$.

Note 16.1.1

1. A ring is denoted $[R; +, \cdot]$ or as just plain R if the operations are understood.
2. The symbols $+$ and $\cdot$ stand for arbitrary operations, not just "regular" addition and multiplication. These symbols are referred to by the usual names. For simplicity, we may write ab instead of $a \cdot b$ if it is not ambiguous.
3. For the abelian group $[R; +]$, we use additive notation. In particular, the group identity is designated by 0 rather than by e and is customarily called the "zero" of the ring. The group inverse is also written in additive notation: $-a$ rather than a^{-1}.

We now look at some examples of rings. Certainly all the additive abelian groups of Chapter 11 are likely candidates for rings.

Example 16.1.1: The Ring of Integers

$[\mathbb{Z}; +, \cdot]$ is a ring, where $+$ and $\cdot$ stand for regular addition and multiplication on $\mathbb{Z}$. From Chapter 11, we already know that $[\mathbb{Z}; +]$ is an abelian group, so we need only check parts 2 and 3 of the definition of a ring. From elementary algebra, we know that the associative law under multiplication and the distributive laws are true for $\mathbb{Z}$. This is our main example of an infinite ring.

Example 16.1.2: The Ring of Integers Modulo n

$[\mathbb{Z}_n; +_n, \times_n]$ is a ring. The properties of modular arithmetic on $\mathbb{Z}_n$ were described in Section 11.4, and they give us the information we need to convince ourselves that $[\mathbb{Z}_n; +_n, \times_n]$ is a ring. This example is our main example of finite rings of different orders.

Definition 16.1.2: Commutative Ring

A ring in which multiplication is a commutative operation is called a commutative ring.

It is common practice to use the word "abelian" when referring to the commutative law under addition and the word "commutative" when referring to the commutative law under the operation of multiplication.

Definition 16.1.3: Unity of a Ring

A ring $[R; +, \cdot]$ that has a multiplicative identity is called a ring with unity. The multiplicative identity itself is called the unity of the ring. More formally, if there exists an element $1 \in R$, such that for all $x \in R$, $x \cdot 1 = 1 \cdot x = x$, then R is called a ***ring with unity***.

Example 16.1.3

The rings in our first two examples were commutative rings with unity, the unity in both cases being the number 1. The ring $[M_{2\times 2}(\mathbb{R}); +, \cdot]$ is a noncommutative ring with unity, the unity being the two by two identity matrix.

16.1.2: Direct Products of Rings

Products of rings are analogous to products of groups or products of Boolean algebras.

Let $[R_i; +_i, \cdot_i], i = 1, 2, \ldots, n$ be rings. Let $P = \overset{n}{\underset{i=1}{\times}} R_i$ and $a = (a_1, a_2, \ldots, a_n), b = (b_1, b_2, \ldots, b_n) \in P$.

From Chapter 11 we know that P is an abelian group under the operation of componentwise addition:

$$a + b = (a_1 +_1 b_1, a_2 +_2 b_2, \ldots, a_n +_n b_n)$$

We also define multiplication on P componentwise:

$$a \cdot b = (a_1 \cdot_1 b_1, a_2 \cdot_2 b_2, \ldots, a_n \cdot_n b_n)$$

To show that P is a ring under the above operations, we need only show that the (multiplicative) associative law and the distributive laws hold. This is indeed the case, and we leave it as an exercise. If each of the R_i is commutative, then P is commutative, and if each contains a unity, then P is a ring with unity, which is the n-tuple consisting of the unities of each of the R_i's.

Example 16.1.4

Since $[\mathbb{Z}_4; +_4, \times_4]$ and $[\mathbb{Z}_3; +_3, \times_3]$ are rings, then $\mathbb{Z}_4 \times \mathbb{Z}_3$ is a ring, where, for example,

$$(2, 1) + (2, 2) = (2 +_4 2, 1 +_3 2) = (0, 0)$$
$$and$$
$$(3, 2) \cdot (2, 2) = (3 \times_4 2, 2 \times_3 2) = (2, 1)$$

To determine the unity in the ring $\mathbb{Z}_4 \times \mathbb{Z}_3$, we look for the element (m, n) such that for all elements $(a, b) \in \mathbb{Z}_4 \times \mathbb{Z}_3$, $(a, b) = (a, b) \cdot (m, n) = (m, n) \cdot (a, b)$, or, equivalently,

$$(a \times_4 m, b \times_3 n) = (m \times_4 a, n \times_3 b) = (a, b)$$

So we want m such that $a \times_4 m = m \times_4 a = a$ in the ring $\mathbb{Z}_4$. The only element m in $\mathbb{Z}_4$ that satisfies this equation is $m = 1$. Similarly, we obtain value of 1 for n. So the unity of $\mathbb{Z}_4 \times \mathbb{Z}_3$, which is unique by Exercise 16.1.15of this section, is $(1, 1)$. We leave to the reader to verify that this ring is commutative.

16.1.3: Multiplicative Inverses in Rings

We now consider the extremely important concept of multiplicative inverses. Certainly many basic equations in elementary algebra (e.g., $2x = 3$) are solved with this concept.

Example 16.1.5

The equation $2x = 3$ has a solution in the ring $[\mathbb{Q}; +, \cdot]$ but does not have a solution in $[\mathbb{Z}; +, \cdot]$ since, to solve this equation, we multiply both sides of the equation $2x = 3$ by the multiplicative inverse of 2. This number, 2^{-1} exists in $\mathbb{Q}$ but does not exist in $\mathbb{Z}$. We formalize this important idea in a definition which by now should be quite familiar to you.

Definition 16.1.4: Multiplicative Inverses

Let $[R; +, \cdot]$ be a ring with unity, 1. If $u \in R$ and there exists an element $v \in R$ such that $u \cdot v = v \cdot u = 1$, then u is said to have a multiplicative inverse, v. A ring element that possesses a multiplicative inverse is a unit of the ring. The set of all units of a ring R is denoted by $U(R)$.

By Theorem 11.3.3, the multiplicative inverse of a ring element is unique, if it exists. For this reason, we can use the notation u^{-1} for the multiplicative inverse of u, if it exists.

Example 16.1.6

In the rings $[\mathbb{R}; +, \cdot]$ and $[\mathbb{Q}; +, \cdot]$ every nonzero element has a multiplicative inverse. The only elements in $[\mathbb{Z}; +, \cdot]$ that have multiplicative inverses are -1 and 1. That is, $U(\mathbb{R}) = \mathbb{R}^*$, $U(\mathbb{Q}) = \mathbb{Q}^*$, and $U(\mathbb{Z}) = \{-1, 1\}$.

Example 16.1.7

Let us find the multiplicative inverses, when they exist, of each element of the ring $[\mathbb{Z}_6; +_6, \times_6]$. If $u = 3$, we want an element v such that $u \times_6 v = 1$. We do not have to check whether $v \times_6 u = 1$ since $\mathbb{Z}_6$ is commutative. If we try each of the six elements, 0, 1, 2, 3, 4, and 5, of $\mathbb{Z}_6$, we find that none of them satisfies the above equation, so 3 does not have a multiplicative inverse in $\mathbb{Z}_6$. However, since $5 \times_6 5 = 1$, 5 does have a multiplicative inverse in $\mathbb{Z}_6$, namely itself: $5^{-1} = 5$. The following table summarizes all results for $\mathbb{Z}_6$.

u	u^{-1}
0	does not exist
1	1
2	does not exist
3	does not exist
4	does not exist
5	5

It shouldn't be a surprise that the zero of a ring is never going to have a multiplicative inverse.

16.1.4: Universal Concepts, Isomorphisms and Subrings

Isomorphism is a universal concept that is important in every algebraic structure. Two rings are isomorphic as rings if and only if they have the same cardinality and if they behave exactly the same under corresponding operations. They are essentially the same ring. For this to be true, they must behave the same as groups (under +) and they must behave the same under the operation of multiplication.

Definition 16.1.5: Ring Isomorphism

Let $[R; +, \cdot]$ and $[R'; +', \cdot']$ be rings. Then R is isomorphic to R' if and only if there exists a function, $f : R \to R'$, called a ring isomorphism, such that

1. f is a bijection
2. $f(a + b) = f(a) +' f(b)$ for all $a, b \in R$
3. $f(a \cdot b) = f(a) \cdot' f(b)$ for all $a, b \in R$.

Conditions 1 and 2 tell us that f is a group isomorphism.

This leads us to the problem of how to show that two rings are not isomorphic. This is a universal concept. It is true for any algebraic structure and was discussed in Chapter 11. To show that two rings are not isomorphic, we must demonstrate that they behave differently under one of the operations. We illustrate through several examples.

Example 16.1.8

Consider the rings $[\mathbb{Z}; +, \cdot]$ and $[2\mathbb{Z}; +, \cdot]$. In Chapter 11 we showed that as groups, the two sets $\mathbb{Z}$ and $2\mathbb{Z}$ with addition were isomorphic. The group isomorphism that proved this was the function $f : \mathbb{Z} \to 2\mathbb{Z}$, defined by $f(n) = 2n$. Is f a ring isomorphism? We need only check whether $f(m \cdot n) = f(m) \cdot f(n)$ for all $m, n \in \mathbb{Z}$. In fact, this condition is not satisfied:

$$f(m \cdot n) = 2 \cdot m \cdot n \quad \text{and} \quad f(m) \cdot f(n) = 2m \cdot 2n = 4 \cdot m \cdot n$$

Therefore, f is not a ring isomorphism. This does not necessarily mean that the two rings $\mathbb{Z}$ and $2\mathbb{Z}$ are not isomorphic, but simply that f doesn't satisfy the conditions. We could imagine that some other function does. We could try to find another function that is a ring isomorphism, or we could try to show that $\mathbb{Z}$ and $2\mathbb{Z}$ are not isomorphic as rings. To do the latter, we must find something different about the ring structure of $\mathbb{Z}$ and $2\mathbb{Z}$.

We already know that they behave identically under addition, so if they are different as rings, it must have something to do with how they behave under the operation of multiplication. Let's begin to develop a checklist of how the two rings could differ:

1. Do they have the same cardinality? Yes, they are both countable.
2. Are they both commutative? Yes.
3. Are they both rings with unity? No.

$\mathbb{Z}$ is a ring with unity, namely the number 1. $2\mathbb{Z}$ is not a ring with unity, $1 \notin 2\mathbb{Z}$. Hence, they are not isomorphic as rings.

Example 16.1.9

Next consider whether $[2\mathbb{Z}; +, \cdot]$ and $[3\mathbb{Z}; +, \cdot]$ are isomorphic. Because of the previous example, we might guess that they are not. However, checklist items 1 through 3 above do not help us. Why? We add another checklist item:

4. Find an equation that makes sense in both rings, which is solvable in one and not the other.

The equation $x + x = x \cdot x,$ or $2x = x^2$, makes sense in both rings. However, this equation has a nonzero solution, $x = 2,$ in $2\mathbb{Z}$, but does not have a nonzero solution in $3\mathbb{Z}$. Thus we have an equation solvable in one ring that cannot be solved in the other, so they cannot be isomorphic.

Another universal concept that applies to the theory of rings is that of a subsystem. A subring of a ring $[R; +, \cdot]$ is any nonempty subset S of R that is a ring under the operations of R. First, for S to be a subring of the ring R, S must be a subgroup of the group $[R; +]$. Also, S must be closed under $\cdot$, satisfy the associative law under $\cdot$, and satisfy the distributive laws. But since R is a ring, the associative and distributive laws are true for every element in R, and, in particular, for all elements in S, since $S \subseteq R$. We have just proven the following theorem:

Theorem 16.1.1

A nonempty subset S of a ring $[R; +, \cdot]$ is a subring of R if and only if:

1. $[S; +]$ is a subgroup of the group $[R; +]$
2. S is closed under multiplication: if $a, b \in S,$ then $a \cdot b \in S$.

Example 16.1.10

The set of even integers, $2\mathbb{Z}$, is a subring of the ring $[\mathbb{Z}; +, \cdot]$ since $[2\mathbb{Z}; +]$ is a subgroup of the group $[\mathbb{Z}; +]$ and since it is also closed with respect to multiplication:

$$2m, 2n \in 2\mathbb{Z} \Rightarrow (2m) \cdot (2n) = 2(2 \cdot m \cdot n) \in 2\mathbb{Z}$$

Several of the basic facts that we are familiar with are true for any ring. The following theorem lists a few of the elementary properties of rings.

Theorem 16.1.2: Some Basic Properties

Let $[R; +, \cdot]$ be a ring, with $a, b \in R$. Then

1. $a \cdot 0 = 0 \cdot a = 0$
2. $a \cdot (-b) = (-a) \cdot b = -(a \cdot b)$
3. $(-a) \cdot (-b) = a \cdot b$

Proof

1. $a \cdot 0 = a \cdot (0+0) = a \cdot 0 + a \cdot 0,$ the last equality valid by the left distributive axiom. Hence if we add $-(a \cdot 0)$ to both sides of the equality above, we obtain $a \cdot 0 = 0$. Similarly, we can prove that $0 \cdot a = 0$.
2. Before we begin the proof of this part, recall that the inverse of each element of the group $[R; +]$ is unique. Hence the inverse of the element $a \cdot b$ is unique and it is denoted $-(a \cdot b)$. Therefore, to prove that $a \cdot (-b) = -(a \cdot b),$ we need only show that $a \cdot (-b)$ inverts $a \cdot b$.

$$\begin{aligned} a \cdot (-b) + a \cdot b &= a \cdot (-b + b) && \text{by the left distributive axiom} \\ &= a \cdot 0 && \text{since } -b \text{ inverts } b \\ &= 0 && \text{by part 1 of this theorem} \end{aligned}$$

   Similarly, it can be shown that$(-a) \cdot b = -(a \cdot b)$.
3. We leave the proof of part 3 to the reader as an exercise.

Example 16.1.11

We will compute $2 \cdot (-2)$ in the ring $[\mathbb{Z}_6; +_6, \times_6]$. $2 \times_6 (-2) = -(2 \times_6 2) = -4 = 2,$ since the additive inverse of 4 (mod 6) is 2. Of course, we could have done the calculation directly as $2 \times_6 (-2) = 2 \times_6 4 = 2$

16.1.5: Integral Domains and Zero Divisors

As the example above illustrates, Theorem 16.1.2is a modest beginning in the study of which algebraic manipulations are possible when working with rings. A fact in elementary algebra that is used frequently in problem solving is the cancellation law. We know that the cancellation laws are true under addition for any ring, based on group theory. Are the cancellation laws true under multiplication, where the group axioms can't be counted on? More specifically, let $[R; +, \cdot]$ be a ring and let $a, b, c \in R$ with $a \neq 0$. When can we cancel the a's in the equation $a \cdot b = a \cdot c$? We can do so if a^{-1} exists, but we cannot assume that a has a multiplicative inverse. The answer to this question is found with the following definition and the theorem that follows.

Definition 16.1.6: Zero Divisor

Let $[R; +, \cdot]$ be a ring. If a and b are two nonzero elements of R such that $a \cdot b = 0,$ then a and b are called zero divisors.

Example 16.1.12

a. In the ring $[\mathbb{Z}_8; +_8, \times_8]$, the numbers 4 and 2 are zero divisors since $4 \times_8 2 = 0$. In addition, 6 is a zero divisor because $6 \times_8 4 = 0$.

b. In the ring $[M_{2\times 2}(\mathbb{R}); +, \cdot]$ the matrices $A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ are zero divisors since $AB = 0$.

c. $[\mathbb{Z}; +, \cdot]$ has no zero divisors.

Now, here is why zero divisors are related to cancellation.

Theorem 16.1.3: Multiplicative Cancellation

The multiplicative cancellation laws hold in a ring $[R; +, \cdot]$ if and only if R has no zero divisors.

Proof

We prove the theorem using the left cancellation axiom, namely that if $a \neq 0$ and $a \cdot b = a \cdot c,$ then $b = c$ for all $a, b, c \in R$. The proof using the right cancellation axiom is its mirror image.

($\Rightarrow$) Assume the left cancellation law holds in R and assume that a and b are two elements in R such that $a \cdot b = 0$. We must show that either $a = 0$ or $b = 0$. To do this, assume that $a \neq 0$ and show that b must be 0.

$$\begin{aligned} a \cdot b = 0 &\Rightarrow a \cdot b = a \cdot 0 \\ &\Rightarrow b = 0 \quad \text{by the left cancellation law} \end{aligned}$$

.

(⇐) Conversely, assume that R has no zero divisors and we will prove that the left cancellation law must hold. To do this, assume that $a, b, c \in R, a \neq 0,$ such that $a \cdot b = a \cdot c$ and show that $b = c$.

$$\begin{aligned} a \cdot b = a \cdot c \quad &\Rightarrow a \cdot b - a \cdot c = 0 \\ &\Rightarrow \ a \cdot (b - c) = 0 \\ &\Rightarrow b - c = 0 \quad \text{since there are no zero divisors} \\ &\Rightarrow b = c \end{aligned}$$

Hence, the only time that the cancellation laws hold in a ring is when there are no zero divisors. The commutative rings with unity in which the two conditions are true are given a special name.

Definition 16.1.7: Integral Domain

A commutative ring with unity containing no zero divisors is called an integral domain.

In this chapter, Integral domains will be denoted generically by the letter D. We state the following two useful facts without proof.

Theorem 16.1.4

If $m \in \mathbb{Z}_n$, $m \neq 0$, then m is a zero divisor if and only if m and n are not relatively prime; i.e., $gcd(m, n) > 1$.

Corollary 16.1.1

If p is a prime, then $\mathbb{Z}_p$ has no zero divisors.

Example 16.1.13

$[\mathbb{Z}; +, \cdot], [\mathbb{Z}_p; +_p, \times_p]$ with p a prime, $[\mathbb{Q}; +, \cdot], [\mathbb{R}; +, \cdot],$ and $[\mathbb{C}; +, \cdot]$ are all integral domains. The key example of an infinite integral domain is $[\mathbb{Z}; +, \cdot]$. In fact, it is from $\mathbb{Z}$ that the term integral domain is derived. Our main example of a finite integral domain is $[\mathbb{Z}_p; +_p, \times_p]$, when p is prime.

We close this section with the verification of an observation that was made in Chapter 11, namely that the product of two algebraic systems may not be an algebraic system of the same type.

Example 16.1.14

Both $[\mathbb{Z}_2; +_2, \times_2]$ and $[\mathbb{Z}_3; +_3, \times_3]$ are integral domains. Consider the direct product $\mathbb{Z}_2 \times \mathbb{Z}_3$. It's true that $\mathbb{Z}_2 \times \mathbb{Z}_3$ is a commutative ring with unity (see Exercise 16.1.13). However, $(1, 0) \cdot (0, 2) = (0, 0)$, so $\mathbb{Z}_2 \times \mathbb{Z}_3$ has zero divisors and is therefore not an integral domain.

16.1.6: Exercises

Exercise 16.1.1

Review the definition of rings to show that the following are rings. The operations involved are the usual operations defined on the sets. Which of these rings are commutative? Which are rings with unity? For the rings with unity, determine the unity and all units.

a. $[\mathbb{Z}; +, \cdot]$
b. $[\mathbb{C}; +, \cdot]$
c. $[\mathbb{Q}; +, \cdot]$
d. $[M_{2\times 2}(\mathbb{R}); +, \cdot]$
e. $[\mathbb{Z}_2; +_2, \times_2]$

Answer

All but ring d are commutative. All of the rings have a unity element. The number 1 is the unity for all of the rings except d. The unity for $M_{2\times 2}(\mathbb{R})$ is the two by two identity matrix. The units are as follows:

a. $\{1,-1\}$
b. $\mathbb{C}^*$
c. $\mathbb{Q}^*$
d. $\{A \mid A_{11}A_{22}-A_{12}A_{21}\neq 0\}$
e. $\{1\}$

Exercise 16.1.2

Follow the instructions for Exercise 16.1.1 and the following rings:

a. $[\mathbb{Z}_6;+_6,\times_6]$
b. $[\mathbb{Z}_5;+_5,\times_5]$
c. $\left[{\mathbb{Z}_2}^3;+,\cdot\right]$
d. $[\mathbb{Z}_8;+_8,\times_8]$
e. $[\mathbb{Z}\times\mathbb{Z};+,\cdot]$
f. $\left[\mathbb{R}^2;+,\cdot\right]$

Exercise 16.1.3

Show that the following pairs of rings are not isomorphic:

a. $[\mathbb{Z};+,\cdot]$ and $[M_{2\times 2}(\mathbb{Z});+,\cdot]$
b. $[3\mathbb{Z};+,\cdot]$ and $[4\mathbb{Z};+,\cdot]$.

Answer

a. Consider commutativity
b. Solve $x^2=3x$ in both rings.

Exercise 16.1.4

Show that the following pairs of rings are not isomorphic:

a. $[\mathbb{R};+,\cdot]$ and $[\mathbb{Q};+,\cdot]$.
b. $[\mathbb{Z}_2\times\mathbb{Z}_2;+,\cdot]$ and $[\mathbb{Z}_4;+,\cdot]$.

Exercise 16.1.5

a. Show that $3\mathbb{Z}$ is a subring of the ring $[\mathbb{Z};+,\cdot]$
b. Find all subrings of $\mathbb{Z}_8$.
c. Find all subrings of $\mathbb{Z}_2\times\mathbb{Z}_2$.

Answer

a. We already know that $3\mathbb{Z}$ is a subgroup of the group $\mathbb{Z}$. We need only show that $3\mathbb{Z}$ is closed with respect to multiplication. Let $3m,3n\in 3\mathbb{Z}$. $(3m)(3n)=3(3mn)\in 3\mathbb{Z}$, since $3mn\in\mathbb{Z}$.
b. The proper subrings are $\{0,2,4,6\}$ and $\{0,4\}$; while $\{0\}$ and $\mathbb{Z}_8$ are improper subrings.
c. The proper subrings are $\{00,01\}$, $\{00,10\}$, and $\{00,11\}$; while $\{00\}$ and $\mathbb{Z}_2\times\mathbb{Z}_2$ are improper subrings.

Exercise 16.1.6

Verify the validity of Theorem 16.1.3 by finding examples of elements a, b, and c, $a\neq 0$ in the following rings, where $a\cdot b=a\cdot c$ and yet $b\neq c$:

a. $\mathbb{Z}_8$
b. $M_{2\times 2}(\mathbb{R})$
c. ${\mathbb{Z}_2}^2$

Exercise 16.1.7

a. Determine all solutions of the equation $x^2-5x+6=0$ in $\mathbb{Z}$. Can there be any more than two solutions to this equation (or any quadratic equation) in $\mathbb{Z}$?
b. Find all solutions of the equation in part a in $\mathbb{Z}_{12}$. Why are there more than two solutions?

Answer

a. The left-hand side of the equation factors into the product $(x-2)(x-3)$. Since $\mathbb{Z}$ is an integral domain, $x=2$ and $x=3$ are the only possible solutions.
b. Over $\mathbb{Z}_{12}$, 2, 3, 6, and 11 are solutions. Although the equation factors into $(x-2)(x-3)$, this product can be zero without making x either 2 or 3. For example. If x = 6 we get $(6-2)\times_{12}(6-3)=4\times_{12}3=0$. Notice that 4 and 3 are zero divisors.

Exercise 16.1.8

Solve the equation $x^2+4x+4=0$ in the following rings. Interpret 4 as $1+1+1+1$, where 1 is the unity of the ring.

a. in $\mathbb{Z}_8$
b. in $M_{2\times 2}(\mathbb{R})$
c. in $\mathbb{Z}$
d. in $\mathbb{Z}_3$

Exercise 16.1.9

The relation “is isomorphic to” on rings is an equivalence relation. Explain the meaning of this statement.

Answer

Let R_1, R_2, and R_3 be any rings, then

a. R_1 is isomorphic to R_1 and so “is isomorphic to” is a reflexive relation on rings.
b. R_1 is isomorphic to $R_2 \Rightarrow R_2$ is isomorphic to R_1, and so “is isomorphic to” is a symmetric relation on rings,
c. R_1 is isomorphic to R_2, and R_2 is isomorphic to R_3 implies that R_1 is isomorphic to R_3, and so “is isomorphic to” is a transitive relation on rings.

We haven't proven these properties here, just stated them. The combination of these observations implies that “is isomorphic to” is an equivalence relation on rings.

Exercise 16.1.10

a. Let $R_1, R_2, \ldots, R_n$ be rings. Prove the multiplicative, associative, and distributive laws for the ring

$$R=\overset{n}{\underset{i=1}{\times}}R_i$$

b. If each of the R_i is commutative, is R commutative?
c. Under what conditions will R be a ring with unity?
d. What will the units of R be when it has a unity?

Exercise 16.1.11

a. Prove that the ring $\mathbb{Z}_2 \times \mathbb{Z}_3$ is commutative and has unity.
b. Determine all zero divisors for the ring $\mathbb{Z}_2 \times \mathbb{Z}_3$.
c. Give another example illustrating the fact that the product of two integral domains may not be an integral domain. Is there an example where the product is an integral domain?

Answer

a. Commutativity is clear from examination of a multiplication table for $\mathbb{Z}_2 \times \mathbb{Z}_3$. More generally, we could prove a theorem that the direct product of two or more commutative rings is commutative. $(1,1)$ is the unity of $\mathbb{Z}_2 \times \mathbb{Z}_3$.
b. $\{(m,n) | m = 0 \text{ or } n = 0, (m,n) \neq (0,0)\}$
c. Another example is $\mathbb{Z} \times \mathbb{Z}$. You never get an integral domain in this situation. By the definition an integral domain D must contain a "zero" so we always have $(1,0) \cdot (0,1) = (0,0)$ in $D \times D$.

Exercise 16.1.12: Boolean Rings

Let U be a nonempty set.

a. Verify that $[\mathcal{P}(U); \oplus, \cap]$ is a commutative ring with unity.
b. What are the units of this ring?

Exercise 16.1.13

a. For any ring $[R; +, \cdot]$, expand $(a+b)(c+d)$ for $a, b, c, d \in R$.
b. If R is commutative, prove that $(a+b)^2 = a^2 + 2ab + b^2$ for all $a, b \in R$.

Answer

a. $(a+b)(c+d) = (a+b)c + (a+b)d = ac + bc + ad + bd$

b.
$$\begin{aligned} (a+b)(a+b) &= aa + ba + ab + bb \quad \text{by part a} \\ &= aa + ab + ab + bb \quad \text{since } R \text{ is commutative} \\ &= a^2 + 2ab + b^2 \end{aligned}$$

Exercise 16.1.14

a. Let R be a commutative ring with unity. Prove by induction that for $n \geq 1$, $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$
b. Simplify $(a+b)^5$ in $\mathbb{Z}_5$.
c. Simplify $(a+b)^{10}$ in $\mathbb{Z}_{10}$.

Exercise 16.1.15

Prove part 3 of Theorem 16.1.2

Exercise 16.1.16

Let U be a finite set. Prove that the Boolean ring $[\mathcal{P}(U); \oplus, \cap]$ is isomorphic to the ring $[\mathbb{Z}_2{}^n; +, \cdot]$. where $n = |U|$.

This page titled 16.1: Rings, Basic Definitions and Concepts is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

16.2: Fields

Although the algebraic structures of rings and integral domains are widely used and play an important part in the applications of mathematics, we still cannot solve the simple equation $ax = b$, $a \neq 0$ in all rings or all integral domains, for that matter. Yet this is one of the first equations we learn to solve in elementary algebra and its solubility is basic to innumerable questions. If we wish to solve a wide range of problems in a system we need at least all of the laws true for rings and the cancellation laws together with the ability to solve the equation $ax = b$, $a \neq 0$. We summarize the above in a definition and list theorems that will place this concept in the context of the previous section.

Definition 16.2.1: Field

A field is a commutative ring with unity such that each nonzero element has a multiplicative inverse.

In this chapter, we denote a field generically by the letter F. The letters k, K and L are also conventionally used for fields.

Example 16.2.1: Some Common Fields

The most common infinite fields are $[\mathbb{Q}; +, \cdot]$, $[\mathbb{R}; +, \cdot]$, and $[\mathbb{C}; +, \cdot]$.

Remark 16.2.1

Since every field is a ring, all facts and concepts that are true for rings are true for any field.

Theorem 16.2.1: Field ⇒ Integral Domain

Every field is an integral domain.

Proof

The proof is fairly easy and a good exercise, so we provide a hint. Starting with the assumption that $a \cdot b = 0$ if we assume that $a \neq 0$ then the existence of a^{-1} makes it possible to infer that $b = 0$.

Of course the converse of Theorem 16.2.1 is not true. Consider $[\mathbb{Z}; +, \cdot]$. However, the next theorem proves the converse in finite fields.

Theorem 16.2.2: Finite Integral Domain ⇒ Field

Every finite integral domain is a field.

Proof

We leave the details to the reader, but observe that if D is a finite integral domain, we can list all elements as $a_1, a_2, \ldots, a_n$, where $a_1 = 1$. Now, to show that any a_i has a multiplicative inverse, consider the n products $a_i \cdot a_1, a_i \cdot a_2, \ldots, a_i \cdot a_n$. What can you say about these products?

If p is a prime, $p \mid (a \cdot b) \Rightarrow p \mid a$ or $p \mid b$. An immediate implication of this fact is the following corollary.

Corollary 16.2.1

If p is a prime, then $\mathbb{Z}_p$ is a field.

Example 16.2.2: A Field of Order 4

Corollary 16.2.1 gives us a large number of finite fields, but we must be cautious. This does not tell us that all finite fields are of the form $\mathbb{Z}_p$, p a prime. To see this, let's try to construct a field of order 4.

First the field must contain the additive and multiplicative identities, 0 and 1, so, without loss of generality, we can assume that the field we are looking for is of the form $F = \{0, 1, a, b\}$. Since there are only two nonisomorphic groups of order 4, we have only two choices for the group table for $[F; +]$. If the additive group is isomorphic to $\mathbb{Z}_4$ then two of the nonzero elements of F would not be their own additive inverse (as are 1 and 3 in $\mathbb{Z}_4$). Let's assume $\beta \in F$ is one of those elements and $\beta + \beta = \gamma \neq 0$. An isomorphism between the additive groups F and $\mathbb{Z}_4$ would require that γ in F correspond with 2 in $\mathbb{Z}_4$. We could continue our argument and infer that $\gamma \cdot \gamma = 0$, producing a zero divisor, which we need to avoid if F is to be a field. We leave the remainder of the argument to the reader. We can thus complete the addition table so that $[F; +]$ is isomorphic to ${\mathbb{Z}_2}^2$:

$+$	0	1	a	b
0	0	1	a	b
1	1	0	b	a
a	a	b	0	1
b	b	a	1	0

Next, since 1 is the unity of F, the partial multiplication table must look like:

$\cdot$	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	$-$	$-$
b	0	b	$-$	$-$

Hence, to complete the table, we have only four entries to find, and, since F must be commutative, this reduces our task to filling in three entries. Next, each nonzero element of F must have a unique multiplicative inverse. The inverse of a must be either a itself or b. If $a^{-1} = a$, then $b^{-1} = b$. (Why?) But $a^{-1} = a \Rightarrow a \cdot a = 1$. And if $a \cdot a = 1$, then $a \cdot b$ is equal to a or b. In either case, by the cancellation law, we obtain $a = 1$ or $b = 1$, which is impossible. Therefore we are forced to conclude that $a^{-1} = b$ and $b^{-1} = a$. To determine the final two products of the table, simply note that, $a \cdot a \neq a$ because the equation $x^2 = x$ has only two solutions, 0 and 1 in any field. We also know that $a \cdot a$ cannot be 1 because a doesn't invert itself and cannot be 0 because a can't be a zero divisor. This leaves us with one possible conclusion, that $a \cdot a = b$ and similarly $b \cdot b = a$. Hence, our multiplication table for F is:

$\cdot$	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	b	1
b	0	b	1	a

We leave it to the reader to verify that $[F; +, \cdot]$, as described above, is a field. Hence, we have produced a field of order 4. This construction would be difficult to repeat for larger fields. In section 16.4 we will introduce a different approach to constructing fields that will be far more efficient.

Even though not all finite fields are isomorphic to $\mathbb{Z}_p$ for some prime p, it can be shown that every field F must have either:

- a subfield isomorphic to $\mathbb{Z}_p$ for some prime p, or
- a subfield isomorphic to $\mathbb{Q}$.

One can think of all fields as being constructed from either $\mathbb{Z}_p$ or $\mathbb{Q}$.

Example 16.2.3

$[\mathbb{R}; +, \cdot]$ is a field, and it contains a subfield isomorphic to $[\mathbb{Q}; +, \cdot]$, namely $\mathbb{Q}$ itself.

Example 16.2.4

The field F that we constructed in Example 16.2.2has a subfield isomorphic to $\mathbb{Z}_p$ for some prime p. From the tables, we note that the subset $\{0,1\}$ of $\{0,1,a,b\}$ under the given operations of F behaves exactly like $[\mathbb{Z}_2;+_2,\times_2]$. Hence, F has a subfield isomorphic to $\mathbb{Z}_2$.

We close this section with a brief discussion of isomorphic fields. Again, since a field is a ring, the definition of isomorphism of fields is the same as that of rings. It can be shown that if f is a field isomorphism, then $f\left(a^{-1}\right)=f(a)^{-1}$; that is, inverses are mapped onto inverses under any field isomorphism. A major question to try to solve is: How many different non-isomorphic finite fields are there of any given order? If p is a prime, it seems clear from our discussions that all fields of order p are isomorphic to $\mathbb{Z}_p$. But how many nonisomorphic fields are there, if any, of order 4, 6, 8, 9, etc? The answer is given in the following theorem, whose proof is beyond the scope of this text.

Theorem 16.2.3

1. Any finite field F has order p^n for a prime p and a positive integer n.
2. For any prime p and any positive integer n there is a field of order p^n .
3. Any two fields of order p^n are isomorphic.

Note 16.2.1: Galois

The field of order p^n is frequently referred to as the Galois field of order p^n and it is denoted by $GF(p^n)$. Evariste Galois (1811-32) was a pioneer in the field of abstract algebra.



Figure 16.2.1: French stamp honoring Evariste Galois

This theorem tells us that there is a field of order $2^2=4$, and there is only one such field up to isomorphism. That is, all such fields of order 4 are isomorphic to F, which we constructed in the example above.

16.2.1: Exercises

Exercise 16.2.1

Write out the addition, multiplication, and “inverse” tables for each of the following fields'.

a. $[\mathbb{Z}_2;+_2,\times_2]$
b. $[\mathbb{Z}_3;+_3,\times_3]$
c. $[\mathbb{Z}_5;+_5,\times_5]$

Exercise 16.2.2

Show that the set of units of the fields in Exercise 16.2.1 form a group under the operation of the multiplication of the given field. Recall that a unit is an element which has a multiplicative inverse.

Exercise 16.2.3

Complete the proof of Theorem 16.2.2that every finite integral domain is a field.

Exercise 16.2.4

Write out the operation tables for ${\mathbb{Z}_2}^2$. Is ${\mathbb{Z}_2}^2$ a ring? An integral domain? A field? Explain.

Exercise 16.2.5

Determine all values x from the given field that satisfy the given equation:

a. $x+1=-1$ in $\mathbb{Z}_2$, $\mathbb{Z}_3$ and $\mathbb{Z}_5$
b. $2x+1=2$ in $\mathbb{Z}_3$ and $\mathbb{Z}_5$
c. $3x+1=2$ in $\mathbb{Z}_5$

Answer

a. 0 in $\mathbb{Z}_2$, 1 in $\mathbb{Z}_3$, 3 in $\mathbb{Z}_5$
b. 2 in $\mathbb{Z}_3$, 3 in $\mathbb{Z}_5$
c. 2 in $\mathbb{Z}_5$

Exercise 16.2.6

a. Prove that if p and q are prime, then $\mathbb{Z}_p \times \mathbb{Z}_q$ is never a field.
b. Can ${\mathbb{Z}_p}^n$ be a field for any prime p and any positive integer $n \geq 2$?

Exercise 16.2.7

Determine all solutions to the following equations over $\mathbb{Z}_2$. That is, find all elements of $\mathbb{Z}_2$ that satisfy the equations.

a. $x^2+x=0$
b. $x^2+1=0$
c. $x^3+x^2+x+1=0$
d. $x^3+x+1=0$

Answer

a. 0 and 1
b. 1
c. 1
d. none

Exercise 16.2.8

Determine the number of different fields, if any, of all orders 2 through 15. Wherever possible, describe these fields via a known field.

Exercise 16.2.9

Let $\mathbb{Q}\left(\sqrt{2}\right)=\left\{a+b\sqrt{2} \middle| a,b \in \mathbb{Q}\right\}$.

a. Prove that $\left[\mathbb{Q}\left(\sqrt{2}\right);+,\cdot\right]$ is a field.
b. Show that $\mathbb{Q}$ is a subfield of $\mathbb{Q}\left(\sqrt{2}\right)$. For this reason, $\mathbb{Q}\left(\sqrt{2}\right)$ is called an extension field of $\mathbb{Q}$.
c. Show that all the roots of the equation $x^2-4x+\frac{7}{2}=0$ lie in the extension field $\mathbb{Q}\left(\sqrt{2}\right)$.
d. Do the roots of the equation $x^2-4x+3=0$ lie in this field? Explain.

This page titled 16.2: Fields is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

16.3: Polynomial Rings

In the previous sections we examined the solutions of a few equations over different rings and fields. To solve the equation $x^2-2=0$ over the field of the real numbers means to find all solutions of this equation that are in this particular field $\mathbb{R}$. This statement can be replaced as follows: Determine all $a\in\mathbb{R}$ such that the polynomial $f(x)=x^2-2$ is equal to zero when evaluated at $x=a$. In this section, we will concentrate on the theory of polynomials. We will develop concepts using the general setting of polynomials over rings since results proven over rings are true for fields (and integral domains). The reader should keep in mind that in most cases we are just formalizing concepts that he or she learned in high school algebra over the field of reals.

Definition 16.3.1: Polynomial over a Ring

Let $[R;+,\cdot]$ be a ring. A polynomial, $f(x)$, over R is an expression of the form

$$f(x)=\sum_{i=0}^{n}a_ix^i=a_0+a_1x+a_2x^2+\cdots+a_nx^n$$

where $n\geq 0$, and $a_0,a_1,a_2,\ldots,a_n\in R$. If $a_n\neq 0$, then the degree of $f(x)$ is n. If $f(x)=0$, then the degree of $f(x)$ is undefined, but for convenience we say that $\deg 0=-\infty$. If the degree of $f(x)$ is n, we write $\deg f(x)=n$. The set of all polynomials in the indeterminate x with coefficients in R is denoted by $R[x]$.

Note 16.3.1

- The symbol x is an object called an ***indeterminate***, which is not an element of the ring R.
- Note that $R\subseteq R[x]$. The elements of R are called constant polynomials, with the nonzero elements of R being the polynomials of degree 0.
- R is called the ground, or base, ring for $R[x]$.
- In the definition above, we have written the terms in increasing degree starting with the constant. The ordering of terms can be reversed without changing the polynomial. For example, $1+2x-3x^4$ and $-3x^4+2x+1$ are the same polynomial.
- A term of the form x^k in a polynomial is understood to be $1x^k$.
- It is understood that if $\deg f(x)=n$, then coefficients of powers of x higher than n are equal to the zero of the base ring.

Definition 16.3.2: Polynomial Addition

Let $f(x)=a_0+a_1x+a_2x^2+\cdots+a_mx^m$ and $g(x)=b_0+b_1x+b_2x^2+\cdots+b_nx^n$ be elements in $R[x]$ so that $a_i\in R$ and $b_i\in R$ for all i. Let k be the maximum of m and n. Then $f(x)+g(x)=c_0+c_1x+c_2x^2+\cdots+c_kx^k$, where $c_i=a_i+b_i$ for $i=0,1,2,\ldots,k$.

Definition 16.3.3: Polynomial Multiplication

Let $f(x)=a_0+a_1x+a_2x^2+\cdots+a_mx^m$ and $g(x)=b_0+b_1x+b_2x^2+\cdots+b_nx^n$. Then

$$f(x)\cdot g(x)=d_0+d_1x+d_2x^2+\cdots+d_px^p \quad \text{where } p=m+n \text{ and}$$
$$d_s=\textstyle\sum_{i=0}^{s}a_ib_{s-i}=a_0b_s+a_1b_{s-1}+a_2b_{s-2}+\cdots+a_{s-1}b_1+a_sb_0$$
$$\text{for } 0\leq s\leq p$$

The important fact to keep in mind is that addition and multiplication in $R[x]$ depends on addition and multiplication in R. The powers of x merely serve the purpose of "place holders." All computations involving coefficients are done over the given ring. Powers of the indeterminate are computed formally applying the rule of adding exponents when multiplying powers.

Example 16.3.1

$f(x)=3$, $g(x)=2-4x+7x^2$, and $h(x)=2+x^4$ are all polynomials in $\mathbb{Z}[x]$. Their degrees are 0, 2, and 4, respectively.

Addition and multiplication of polynomials are performed as in high school algebra. However, we must do our computations in the ground ring of the polynomials.

Example 16.3.2

In $\mathbb{Z}_3[x]$, if $f(x) = 1 + x$ and $g(x) = 2 + x$, then

$$\begin{aligned} f(x) + g(x) &= (1 + x) + (2 + x) \\ &= (1 +_3 2) + (1 +_3 1)\, x \\ &= 0 + 2x \\ &= 2x \end{aligned}$$

and

$$\begin{aligned} f(x)g(x) &= (1 + x) \cdot (2 + x) \\ &= 1 \times_3 2 + (1 \times_3 1 +_3 1 \times_3 2)x + (1 \times_3 2)x^2 \\ &= 2 + 0x + x^2 \\ &= 2 + x^2 \end{aligned}$$

However, for the same polynomials as above, $f(x)$ and $g(x)$ in the more familiar setting of $\mathbb{Z}[x]$, we have

$$f(x) + g(x) = (1 + x) + (2 + x) = (1 + 2) + (1 + 1)x = 3 + 2x$$

and

$$\begin{aligned} f(x)g(x) = (1 + x) \cdot (2 + x) & \\ &= 1 \cdot 2 + (1 \cdot 1 + 1 \cdot 2)x + (1 \cdot 1)x^2 \\ &= 2 + 3x + x^2 \end{aligned}$$

Example 16.3.3

Let $f(x) = 2 + x^2$ and $g(x) = -1 + 4x + 3x^2$. We will compute $f(x) \cdot g(x)$ in $\mathbb{Z}[x]$. Of course this product can be obtained by the usual methods of high school algebra. We will, for illustrative purposes, use the above definition. Using the notation of the above definition, $a_0 = 2$, $a_1 = 0$, $a_2 = 1$, $b_0 = -1$, $b_1 = 4$, and $b_2 = 3$. We want to compute the coefficients d_0, d_1, d_2, d_3, and d_4 . We will compute d_3 , the coefficient of the x^3 term of the product, and leave the remainder to the reader (see Exercise 16.3.2of this section). Since the degrees of both factors is 2, $a_i = b_i = 0$ for $i \geq 3$. The coefficient of x^3 is

$$d_3 = a_0b_3 + a_1b_2 + a_2b_1 + a_3b_0 = 2 \cdot 0 + 0 \cdot 3 + 1 \cdot 4 + 0 \cdot (-1) = 4$$

The proofs of the following theorem are not difficult but rather long, so we omit them.

Theorem 16.3.1: Properties of Polynomial Rings

Let $[R; +, \cdot]$ be a ring. Then:

1. $R[x]$ is a ring under the operations of polynomial addition and multiplication.
2. If R is a commutative ring, then $R[x]$ is a commutative ring.
3. If R is a ring with unity, 1, then $R[x]$ is a ring with unity (the unity in $R[x]$ is $1 + 0x + 0x^2 + \cdots$).
4. If R is an integral domain, then $R[x]$ is an integral domain.
5. If F is a field, then $F[x]$ is not a field. However, $F[x]$ is an integral domain.

Next we turn to division of polynomials, which is not an operation since the result is a pair of polynomials, not a single one. From high school algebra we all learned the standard procedure for dividing a polynomial $f(x)$ by a second polynomial $g(x)$. This process of polynomial long division is referred to as the division property for polynomials. Under this scheme we continue to divide until the result is a quotient $q(x)$ and a remainder $r(x)$ whose degree is strictly less than that of the divisor $g(x)$. This property is valid over any field. Before giving a formal description, we consider some examples.

Example 16.3.4: Polynomial Division

Let $f(x)=1+x+x^3$ and $g(x)=1+x$ be two polynomials in $\mathbb{Z}_2[x]$. Let us divide $f(x)$ by $g(x)$. Keep in mind that we are in $\mathbb{Z}_2[x]$ and that, in particular, $-1=1$ in $\mathbb{Z}_2$. This is a case where reordering the terms in decreasing degree is preferred.

$$\begin{array}{r} x^2+x \\ x+1 \;\overline{)}\; x^3+0x^2+x+1 \\ \underline{x^3+x^2} \\ x^2+x+1 \\ \underline{x^2+x} \\ 1 \end{array}$$

Figure 16.3.1

Therefore,

$$\frac{x^3+x+1}{x+1}=x^2+x+\frac{1}{x+1}$$

or equivalently,

$$x^3+x+2=\left(x^2+x\right)\cdot(x+1)+1$$

That is, $f(x)=g(x)\cdot q(x)+r(x)$ where $q(x)=x^2+x$ and $r(x)=1$. Notice that $\deg(r(x))=0$, which is strictly less than $\deg(g(x))=1$.

Example 16.3.5

Let $f(x)=1+x^4$ and $g(x)=1+x$ be polynomials in $\mathbb{Z}_2[x]$. Let us divide $f(x)$ by $g(x)$:

$$\begin{array}{r} x^3+x^2+x+1 \\ x+1 \;\overline{)}\; x^4+0x^3+0x^2+0x+1 \\ \underline{x^4+x^3} \\ x^3+1 \\ \underline{x^3+x^2} \\ x^2+1 \\ \underline{x^2+x} \\ x+1 \\ \underline{x+1} \\ 0 \end{array}$$

Figure 16.3.2

Thus $x^4+1=\left(x^3+x^2+x+1\right)(x+1)$. Since we have 0 as a remainder, $x+1$ must be a factor of x^4+1. Also, since $x+1$ is a factor of x^4+1, 1 is a zero (or root) of x^4+1. Of course we could have determined that 1 is a root of $f(x)$ simply by computing $f(1)=1^4+_2 1=1+_2 1=0$.

Before we summarize the main results suggested by the previous examples, we should probably consider what could have happened if we had attempted to perform divisions of polynomials in the ring $\mathbb{Z}[x]$ rather than in the polynomials over the field $\mathbb{Z}_2$. For example, $f(x)=x^2-1$ and $g(x)=2x-1$ are both elements of the ring $\mathbb{Z}[x]$, yet $x^2-1=(\frac{1}{2}x+\frac{1}{2})(2x-1)-\frac{3}{4}$ The quotient and remainder are not a polynomials over $\mathbb{Z}$ but polynomials over the field of rational numbers. For this reason it would be wise to describe all results over a field F rather than over an arbitrary ring R so that we don't have to expand our possible set of coefficients.

Theorem 16.3.2: Division Property for Polynomials

Let $[F;+,\cdot]$ be a field and let $f(x)$ and $g(x)$ be two elements of $F[x]$ with $g(x)\neq 0$. Then there exist unique polynomials $q(x)$ and $r(x)$ in $F[x]$ such that $f(x)=g(x)q(x)+r(x)$, where $\deg r(x)<\deg g(x)$.

Proof

This theorem can be proven by induction on $\deg f(x)$.

Theorem 16.3.3: The Factor Theorem

Let $[F;+,\cdot]$ be a field. An element $a\in F$ is a zero of $f(x)\in F[x]$ if and only if $x-a$ is a factor of $f(x)$ in $F[x]$.

Proof

($\Rightarrow$) Assume that $a\in F$ is a zero of $f(x)\in F[x]$. We wish to show that $x-a$ is a factor of $f(x)$. To do so, apply the division property to $f(x)$ and $g(x)=x-a$. Hence, there exist unique polynomials $q(x)$ and $r(x)$ from $F[x]$ such that $f(x)=(x-a)\cdot q(x)+r(x)$ and the $\deg r(x)<\deg(x-a)=1$, so $r(x)=c\in F$, that is, $r(x)$ is a constant. Also, the fact that a is a zero of $f(x)$ means that $f(a)=0$. So $f(x)=(x-a)\cdot q(x)+c$ becomes $0=f(a)=(a-a)q(a)+c$. Hence $c=0$, so $f(x)=(x-a)\cdot q(x)$, and $x-a$ is a factor of $f(x)$. The reader should note that a critical point of the proof of this half of the theorem was the part of the division property that stated that $\deg r(x)<\deg g(x)$.

($\Leftarrow$) We leave this half to the reader as an exercise.

Theorem 16.3.4

A nonzero polynomial $f(x)\in F[x]$ of degree n can have at most n zeros.

Proof

Let $a\in F$ be a zero of $f(x)$. Then $f(x)=(x-a)\cdot q_1(x)$, $q_1(x)\in F[x]$, by the Factor Theorem. If $b\in F$ is a zero of $q_1(x)$, then again by Factor Theorem, $f(x)=(x-a)(x-b)q_2(x)$, $q_2(x)\in F[x]$. Continue this process, which must terminate in at most n steps since the degree of $q_k(x)$ would be $n-k$.

From The Factor Theorem, Theorem 16.3.3, we can get yet another insight into the problems associated with solving polynomial equations; that is, finding the zeros of a polynomial. The initial important idea here is that the zero a is from the ground field F. Second, a is a zero only if $(x-a)$ is a factor of $f(x)$ in $F[x]$; that is, only when $f(x)$ can be factored (or reduced) to the product of $(x-a)$ times some other polynomial in $F[x]$.

Example 16.3.6

Consider the polynomial $f(x)=x^2-2$ taken as being in $\mathbb{Q}$[x]. From high school algebra we know that $f(x)$ has two zeros (or roots), namely $\pm\sqrt{2}$, and x^2-2 can be factored as $\left(x-\sqrt{2}\right)\left(x+\sqrt{2}\right)$. However, we are working in $\mathbb{Q}[x]$, these two factors are not in the set of polynomials over the rational numbers, $\mathbb{Q}$ since $\sqrt{2}\notin\mathbb{Q}$. Therefore, x^2-2 does not have a zero in $\mathbb{Q}$ since it cannot be factored over $\mathbb{Q}$. When this happens, we say that the polynomial is irreducible over $\mathbb{Q}$.

The problem of factoring polynomials is tied hand-in-hand with that of the reducibility of polynomials. We give a precise definition of this concept.

Definition 16.3.4: Reducibility over a Field

Let $[F;+,\cdot]$ be a field and let $f(x)\in F[x]$ be a nonconstant polynomial. $f(x)$ is ***reducible*** over F if and only if it cannot be factored as a product of two nonconstant polynomials in $F[x]$. A polynomial is ***irreducible*** over F if it is not reducible over F.

Example 16.3.7

The polynomial $f(x) = x^4 + 1$ is reducible over $\mathbb{Z}_2$ since $x^4 + 1 = (x+1)\left(x^3 + x^2 + x - 1\right)$.

Example 16.3.8

Is the polynomial $f(x) = x^3 + x + 1$ reducible over $\mathbb{Z}_2$? Since a factorization of a cubic polynomial can only be as a product of linear and quadratic factors, or as a product of three linear factors, $f(x)$ is reducible if and only if it has at least one linear factor. From the Factor Theorem, $x - a$ is a factor of $x^3 + x + 1$ over $\mathbb{Z}_2$ if and only if $a \in \mathbb{Z}_2$ is a zero of $x^3 + x + 1$. So $x^3 + x + 1$ is reducible over $\mathbb{Z}_2$ if and only if it has a zero in $\mathbb{Z}_2$. Since $\mathbb{Z}_2$ has only two elements, 0 and 1, this is easy enough to check. $f(0) = 0^3 +_2 0 +_2 1 = 1$ and $f(1) = 1^3 +_2 1 +_2 1 = 1$, so neither 0 nor 1 is a zero of $f(x)$ over $\mathbb{Z}_2$. Hence, $x^3 + x + 1$ is irreducible over $\mathbb{Z}_2$.

From high school algebra we know that $x^3 + x + 1$ has three zeros from some field. Can we find this field? To be more precise, can we construct the field that contains $\mathbb{Z}_2$ and all zeros of $x^3 + x + 1$? We will consider this task in the next section.

We close this section with a final analogy. Prime numbers play an important role in mathematics. The concept of irreducible polynomials (over a field) is analogous to that of a prime number. Just think of the definition of a prime number. A useful fact concerning primes is: If p is a prime and if $p \mid ab$, then $p \mid a$ or $p \mid b$. We leave it to the reader to think about the veracity of the following: If $p(x)$ is an irreducible polynomial over F, $a(x), b(x) \in F[x]$ and $p(x) \mid a(x)b(x)$, then $p(x) \mid a(x)$ or $p(x) \mid b(x)$.

16.3.1: Exercises

Exercise 16.3.1

Let $f(x) = 1 + x$ and $g(x) = 1 + x + x^2$. Compute the following sums and products in the indicated rings.

a. $f(x) + g(x)$ and $f(x) \cdot g(x)$ in $\mathbb{Z}[x]$
b. $f(x) + g(x)$ and $f(x) \cdot g(x)$ in $\mathbb{Z}_2[x]$
c. $(f(x) \cdot g(x)) \cdot f(x)$ in $\mathbb{Q}[x]$
d. $(f(x) \cdot g(x)) \cdot f(x)$ in $\mathbb{Z}_2[x]$
e. $f(x) \cdot f(x) + f(x) \cdot g(x)$ in $\mathbb{Z}_2[x]$

Answer

a. $f(x) + g(x) = 2 + 2x + x^2$, $f(x) \cdot g(x) = 1 + 2x + 2x^2 + x^3$
b. $f(x) + g(x) = x^2$, $f(x) \cdot g(x) = 1 + x^3$
c. $1 + 3x + 4x^2 + 3x^3 + x^4$
d. $1 + x + x^3 + x^4$
e. $x^2 + x^3$

Exercise 16.3.2

Complete the calculations started in Example 16.3.3

Exercise 16.3.3

Prove that:

a. The ring $\mathbb{R}$ is a subring of the ring $\mathbb{R}[x]$.
b. The ring $\mathbb{Z}[x]$ is a subring of the $\mathbb{Q}[x]$.
c. The ring $\mathbb{Q}[x]$ is a subring of the ring $\mathbb{R}[x]$.

Answer

a. If $a, b \in \mathbb{R}$, $a - b$ and ab are in $\mathbb{R}$ since $\mathbb{R}$ is a ring in its own right. Therefore, $\mathbb{R}$ is a subring of $\mathbb{R}[x]$. The proofs of parts b and c are similar.

Exercise 16.3.4

a. Find all zeros of x^4+1 in $\mathbb{Z}_3$.
b. Find all zeros of x^5+1 in $\mathbb{Z}_5$.

Exercise 16.3.5

Determine which of the following are reducible over $\mathbb{Z}_2$. Explain.

a. $f(x)=x^3+1$
b. $g(x)=x^3+x^2+x$.
c. $h(x)=x^3+x^2+1$.
d. $k(x)=x^4+x^2+1$. (Be careful.)

Answer

a. Reducible, $(x+1)\left(x^2+x+1\right)$
b. Reducible, $x\left(x^2+x+1\right)$
c. Irreducible. If you could factor this polynomial, one factor would be either x or $x+1$, which would give you a root of 0 or 1, respectively. By substitution of 0 and 1 into this polynomial, it clearly has no roots.
d. Reducible, $(x+1)^4$

Exercise 16.3.6

Prove the second half of The Factor Theorem, Theorem 16.3.3

Exercise 16.3.7

Give an example of the contention made in the last paragraph of this section.

Answer

We illustrate this property of polynomials by showing that it is not true for a nonprime polynomial in $\mathbb{Z}_2[x]$. Suppose that $p(x)=x^2+1$, which can be reduced to $(x+1)^2$, $a(x)=x^2+x$, and $b(x)=x^3+x^2$. Since $a(x)b(x)=x^5+x^3=x^3\left(x^2+1\right)$, $p(x)|a(x)b(x)$. However, $p(x)$ is not a factor of either $a(x)$ or $b(x)$.

Exercise 16.3.8

Determine all zeros of x^4+3x^3+2x+4 in $\mathbb{Z}_5[x]$.

Exercise 16.3.9

Show that x^2-3 is irreducible over $\mathbb{Q}$ but reducible over the field of real numbers.

Answer

The only possible proper factors of x^2-3 are $\left(x-\sqrt{3}\right)$ and $\left(x+\sqrt{3}\right)$, which are not in $\mathbb{Q}[x]$ but are in $\mathbb{R}$[x].

Exercise 16.3.10

The definition of a vector space given in Chapter 13 holds over any field F, not just over the field of real numbers, where the elements of F are called scalars.

a. Show that $F[x]$ is a vector space over F.
b. Find a basis for $F[x]$ over F.
c. What is the dimension of $F[x]$ over F?

Exercise 16.3.11

Prove Theorem 16.3.2 the Division Property for Polynomials

Answer

For $n \geq 0$, let $S(n)$ be the proposition: For all $g(x) \neq 0$ and $f(x)$ with $\deg f(x) = n$, there exist unique polynomials $q(x)$ and $r(x)$ such that $f(x) = g(x)q(x) + r(x)$, and either $r(x) = 0$ or $\deg r(x) < \deg g(x)$.

Basis: $S(0)$ is true, for if $f(x)$ has degree 0, it is a nonzero constant, $f(x) = c \neq 0$, and so either $f(x) = g(x) \cdot 0 + c$ if $g(x)$ is not a constant, or $f(x) = g(x)g(x)^{-1} + 0$ if $g(x)$ is also a constant.

Induction: Assume that for some $n \geq 0$, $S(k)$ is true for all $k \leq n$, If $f(x)$ has degree $n+1$, then there are two cases to consider. If $\deg g(x) > n+1$, $f(x) = g(x) \cdot 0 + f(x)$, and we are done. Otherwise, if $\deg g(x) = m \leq n+1$, we perform long division as follows, where LDT's stand for terms of lower degree than $n+1$.

$$\begin{array}{rl} & f_{n+1} \cdot {g_m}^{-1} x^{n+1-m} \\ \hline g_m x^m + \text{ LDT}'s \;\;) & f_{n+1} x^{n+1} + \text{ LDT}'s \\ & \underline{f_{n+1} x^{n+1} + \text{ LDT}'s} \\ & h(x) \end{array}$$

Therefore,

$$h(x) = f(x) - \left(f_{n+1} \cdot {g_m}^{-1} x^{n+1-m}\right) g(x) \Rightarrow f(x) = \left(f_{n+1} \cdot {g_m}^{-1} x^{n+1-m}\right) g(x) + h(x)$$

Since $\deg h(x)$ is less than $n+1$, we can apply the induction hypothesis: $h(x) = g(x)q(x) + r(x)$ with $\deg r(x) < \deg g(x)$.

Therefore,

$$f(x) = g(x)\left(f_{n+1} \cdot {g_m}^{-1} x^{n+1-m} + q(x)\right) + r(x)$$

with $\deg r(x) < \deg g(x)$. This establishes the existence of a quotient and remainder. The uniqueness of $q(x)$ and $r(x)$ as stated in the theorem is proven as follows: if $f(x)$ is also equal to $g(x)\bar{q}(x) + \bar{r}(x)$ with $\deg \bar{r}(x) < \deg g(x)$, then

$$g(x)q(x) + r(x) = g(x)\bar{q}(x) + \bar{r}(x) \Rightarrow g(x)\left(\bar{q}(x) - q(x)\right) = r(x) - \bar{r}(x)$$

Since $\deg r(x) - \bar{r}(x) < \deg g(x)$, the degree of both sides of the last equation is less than $\deg g(x)$. Therefore, it must be that $\bar{q}(x) - q(x) = 0$, or $q(x) = \bar{q}(x)$ And so $r(x) = \bar{r}(x)$.

Exercise 16.3.12

a. Show that the field $\mathbb{R}$ of real numbers is a vector space over $\mathbb{R}$. Find a basis for this vector space. What is dim $\mathbb{R}$ over $\mathbb{R}$?
b. Repeat part a for an arbitrary field F.
c. Show that $\mathbb{R}$ is a vector space over $\mathbb{Q}$.

This page titled 16.3: Polynomial Rings is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

16.4: Field Extensions

From high school algebra we realize that to solve a polynomial equation means to find its roots (or, equivalently, to find the zeros of the polynomials). From Example 16.3.6 and Example 16.3.8 we know that the zeros may not lie in the given ground field. Hence, to solve a polynomial really involves two steps: first, find the zeros, and second, find the field in which the zeros lie. For economy's sake we would like this field to be the smallest field that contains all the zeros of the given polynomial. To illustrate this concept, let us reconsider the examples from the previous section.

Example 16.4.1: Extending the Rational Numbers

Let $f(x)=x^2-2\in\mathbb{Q}[x]$. It is important to remember that we are considering x^2-2 over $\mathbb{Q}$, no other field. We would like to find all zeros of $f(x)$ and the smallest field, call it S for now, that contains them. The zeros are $x=\pm\sqrt{2}$, neither of which is an element of $\mathbb{Q}$. The set S we are looking for must satisfy the conditions:

1. S must be a field.
2. S must contain $\mathbb{Q}$ as a subfield,
3. S must contain all zeros of $f(x)=x^2-2$

By the last condition $\sqrt{2}$ must be an element of S, and, if S is to be a field, the sum, product, difference, and quotient of elements in S must be in S. So operations involving this number, such as $\sqrt{2}$, $\left(\sqrt{2}\right)^2$, $\left(\sqrt{2}\right)^3$, $\sqrt{2}+\sqrt{2}$, $\sqrt{2}-\sqrt{2}$, and $\frac{1}{\sqrt{2}}$ must all be elements of S. Further, since S contains $\mathbb{Q}$ as a subset, any element of $\mathbb{Q}$ combined with $\sqrt{2}$ under any field operation must be an element of S. Hence, every element of the form $a+b\sqrt{2}$, where a and b can be any elements in $\mathbb{Q}$, is an element of S. We leave to the reader to show that $S=\{a+b\sqrt{2}\mid a,b\in\mathbb{Q}\}$ is a field (see Exercise 1 of this section). We note that the second zero of x^2-2, namely $-\sqrt{2}$, is an element of this set. To see this, simply take $a=0$ and $b=-1$. The field S is frequently denoted as $\mathbb{Q}\left(\sqrt{2}\right)$, and it is referred to as an extension field of $\mathbb{Q}$. Note that the polynomial $x^2-2=\left(x-\sqrt{2}\right)\left(x+\sqrt{2}\right)$ factors into linear factors, or ***splits***, in $\mathbb{Q}\left(\sqrt{2}\right)[x]$; that is, all coefficients of both factors are elements of the field $\mathbb{Q}\left(\sqrt{2}\right)$.

Example 16.4.2: Extending $\mathbb{Z}_2$

Consider the polynomial $g(x)=x^2+x+1\in\mathbb{Z}_2[x]$. Let's repeat the steps from the previous example to factor $g(x)$. First, $g(0)=1$ and $g(1)=1$, so none of the elements of $\mathbb{Z}_2$ are zeros of $g(x)$. Hence, the zeros of $g(x)$ must lie in an extension field of $\mathbb{Z}_2$. By Theorem 16.3.4, $g(x)=x^2+x+1$ can have at most two zeros. Let a be a zero of $g(x)$. Then the extension field S of $\mathbb{Z}_2$ must contain, besides a, $a\cdot a=a^2$, a^3, $a+a$, $a+1$, and so on. But, since $g(a)=0$, we have $a^2+a+1=0$, or equivalently, $a^2=-(a+1)=a+1$ (remember, we are working in an extension of $\mathbb{Z}_2$). We can use this recurrence relation to reduce powers of a. So far our extension field, S, of $\mathbb{Z}_2$ must contain the set $\{0,1,a,a+1\}$, and we claim that this the complete extension. For S to be a field, all possible sums, products, and differences of elements in S must be in S. Let's try a few: $a+a=a\left(1+_2 1\right)=a\cdot 0=0\in S$ Since $a+a=0$, $-a=a$, which is in S. Adding three a's together doesn't give us anything new: $a+a+a=a\in S$ In fact, na is in S for all possible positive integers n. Next,

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

Therefore, $a^{-1}=a+1=a^2$ and $(a+1)^{-1}=a$.

It is not difficult to see that a^n is in S for all positive n. Does S contain all zeros of x^2+x+1? Remember, $g(x)$ can have at most two distinct zeros and we called one of them a, so if there is a second, it must be $a+1$. To see if $a+1$ is indeed a zero of $g(x)$, simply compute $f(a+1)$:

$$\begin{aligned} f(a+1) &= (a+1)^2 + (a+1) + 1 \\ &= a^2 + 1 + a + 1 + 1 \\ &= a^2 + a + 1 \\ &= 0 \end{aligned}$$

Therefore, $a+1$ is also a zero of x^2+x+1. Hence, $S = \{0, 1, a, a+1\}$ is the smallest field that contains $\mathbb{Z}_2 = \{0, 1\}$ as a subfield and contains all zeros of x^2+x+1. This extension field is denoted by $\mathbb{Z}_2(a)$. Note that x^2+x+1 splits in $\mathbb{Z}_2(a)$; that is, it factors into linear factors in $\mathbb{Z}_2(a)$. We also observe that $\mathbb{Z}_2(a)$ is a field containing exactly four elements. By Theorem 16.2.3, we expected that $\mathbb{Z}_2(a)$ would be of order p^2 for some prime p and positive integer n. Also recall that all fields of order p^n are isomorphic. Hence, we have described all fields of order $2^2 = 4$ by finding the extension field of a polynomial that is irreducible over $\mathbb{Z}_2$.

The reader might feel somewhat uncomfortable with the results obtained in Example 16.4.2 In particular, what is a? Can we describe it through a known quantity? All we know about a is that it is a zero of $g(x)$ and that $a^2 = a+1$. We could also say that $a(a+1) = 1$, but we really expected more. However, should we expect more? In Example 16.4.1, $\sqrt{2}$ is a number we are more comfortable with, but all we really know about it is that $\alpha = \sqrt{2}$ is the number such that $\alpha^2 = 2$. Similarly, the zero that the reader will obtain in Exercise 16.4.2of this section is the imaginary number i. Here again, this is simply a symbol, and all we know about it is that $i^2 = -1$. Hence, the result obtained in Example 16.4.2is not really that strange.

The reader should be aware that we have just scratched the surface in the development of topics in polynomial rings. One area of significant applications is in coding theory.

Example 16.4.3: An Error Correcting Polynomial Code

An important observation regarding the previous example is that the nonzero elements of $GF(4)$ can be represented two ways. First as a linear combination of 1 and a. There are four such linear combinations, one of which is zero. Second, as powers of a. There are three distinct powers and the each match up with a nonzero linear combination:

$$\begin{aligned} a^0 &= 1 \cdot 1 + 0 \cdot a \\ a^1 &= 0 \cdot 1 + 1 \cdot a \\ a^2 &= 1 \cdot 1 + 1 \cdot a \end{aligned}$$

Next, we briefly describe the field $GF(8)$ and how an error correcting code can be build on a the same observation about that field.

First, we start with the irreducible polynomial $p(x) = x^3 + x + 1$ over $\mathbb{Z}_2$. There is another such cubic polynomial, but its choice produces essentially the same result. Just as we did in the previous example, we assume we have a zero of $p(x)$ and call it β. Since we have assumed that $p(\beta) = \beta^3 + \beta + 1 = 0$, we get the recurrence relation for powers $\beta^3 = \beta + 1$ that lets us reduce the seven powers β^k, $0 \le k \le 6$, to linear combinations of 1, β, and β^2. Higher powers will reduce to these seven, which make up the elements of a field with $2^3 = 8$ elements when we add zero to the set. We leave as an exercise for you to set up a table relating powers of β with the linear combinations.

With this information we are now in a position to take blocks of four bits and encode them with three parity bits to create an error correcting code. If the bits are $b_3b_4b_5b_6$, then we reduce the expression $B_m = b_3 \cdot \beta^3 + b_4 \cdot \beta^4 + b_5 \cdot \beta^5 + b_6 \cdot \beta^6$ using the recurrence relation to an expression $B_p = b_0 \cdot 1 + b_1 \cdot \beta + b_2 \cdot \beta^2$. Since we are equating equals within $GF(8)$, we have $B_p = B_m$, or $B_p + B_m = 0$. The encoded message is $b_0b_1b_2b_3b_4b_5b_6$, which is a representation of 0 in $GF(8)$. If the transmitted sequence of bits is received as $c_0c_1c_2c_3c_4c_5c_6$ we reduce $C = c_0 \cdot 1 + c_1 \cdot \beta + c_2 \cdot \beta^2 + c_3 \cdot \beta^3 + c_4 \cdot \beta^4 + c_5 \cdot \beta^5 + c_6 \cdot \beta^6$ using the recurrence. If there was no transmission error, the result is zero. If the reduced result is zero it is most likely that the original message was $c_3c_4c_5c_6$. If bit k is switched in the transmission, then

$$C = B_p + B_m + \beta^k = \beta^k$$

Therefore if we reduce C with the recurrence, we get the linear combination of 1, β, and β^2 that is equal to β^k and so we can identify the location of the error and correct it.

16.4.1: 16.4.1: Exercises

Exercise 16.4.1

a. Use the definition of a field to show that $\mathbb{Q}(\sqrt{2})$ is a field.
b. Use the definition of vector space to show that $\mathbb{Q}\left(\sqrt{2}\right)$ is a vector space over $\mathbb{Q}$.
c. Prove that $\left\{1, \sqrt{2}\right\}$ is a basis for the vector space $\mathbb{Q}\left(\sqrt{2}\right)$ over $\mathbb{Q}$, and, therefore, the dimension of $\mathbb{Q}(\sqrt{2})$ over $\mathbb{Q}$ is 2.

Answer

If $a_0 + a_1\sqrt{2} \in \mathbb{Q}\left[\sqrt{2}\right]$ is nonzero, then it has a multiplicative inverse:

$$\begin{aligned}\frac{1}{a_0 + a_1\sqrt{2}} &= \frac{1}{a_0 + a_1\sqrt{2}}\,\frac{a_0 - a_1\sqrt{2}}{a_0 - a_1\sqrt{2}} \\ &= \frac{a_0 - a_1\sqrt{2}}{{a_0}^2 - 2{a_1}^2} \\ &= \frac{a_0}{{a_0}^2 - 2{a_1}^2} - \frac{a_1}{{a_0}^2 - 2{a_1}^2}\sqrt{2}\end{aligned}$$

The denominator, ${a_0}^2 - 2{a_1}^2$, is nonzero since $\sqrt{2}$ is irrational. Since $\frac{a_0}{{a_0}^2 - 2{a_1}^2}$ and $\frac{-a_1}{{a_0}^2 - 2{a_1}^2}$ are both rational numbers, $a_0 + a_1\sqrt{2}$ is a unit of $\mathbb{Q}\left[\sqrt{2}\right]$. The field containing $\mathbb{Q}\left[\sqrt{2}\right]$ is denoted $\mathbb{Q}\left(\sqrt{2}\right)$ and so $\mathbb{Q}\left(\sqrt{2}\right) = \mathbb{Q}\left[\sqrt{2}\right]$

Exercise 16.4.2

a. Determine the splitting field of $f(x) = x^2 + 1$ over $\mathbb{R}$. This means consider the polynomial $f(x) = x^2 + 1 \in \mathbb{R}[x]$ and find the smallest field that contains $\mathbb{R}$ and all the zeros of $f(x)$. Denote this field by $\mathbb{R}(i)$.
b. $\mathbb{R}(i)$ is more commonly referred to by a different name. What is it?
c. Show that $\{1, i\}$ is a basis for the vector space $\mathbb{R}(i)$ over $\mathbb{R}$. What is the dimension of this vector space (over $\mathbb{R}$)?

Exercise 16.4.3

Determine the splitting field of $x^4 - 5x^2 + 6$ over $\mathbb{Q}$.

Answer

$x^4 - 5x^2 + 6 = (x^2 - 2)(x^2 - 3)$ has zeros $\pm\sqrt{2}$ and $\pm\sqrt{3}$.

$\mathbb{Q}(\sqrt{2}) = \{a + b\sqrt{2} \mid a, b \in \mathbb{Q}\}$ contains the zeros $\pm\sqrt{2}$ but does not contain $\pm\sqrt{3}$, since neither are expressible in the form $a + b\sqrt{2}$. If we consider the set $\{c + d\sqrt{3} \mid c, d \in \mathbb{Q}(\sqrt{2})\}$, then this field contains $\pm\sqrt{3}$ as well as $\pm\sqrt{2}$, and is denoted $\mathbb{Q}(\sqrt{2})(\sqrt{3}) = \mathbb{Q}(\sqrt{2}, \sqrt{3})$. Taking into account the form of c and d in the description above, we can expand to

$$\mathbb{Q}(\sqrt{2}, \sqrt{3}) = \{b_0 + b_1\sqrt{2} + b_2\sqrt{3} + b_3\sqrt{6} \mid b_i \in \mathbb{Q}\}$$

Exercise 16.4.4

a. Factor $x^2 + x + 1$ into linear factors in $\mathbb{Z}_2(a)$.
b. Write out the field tables for the field $\mathbb{Z}_2(a)$ and compare the results to the tables of Example 16.2.2.
c. Cite a theorem and use it to show why the results of part b were to be expected.

Exercise 16.4.5

a. Show that $x^3 + x + 1$ is irreducible over $\mathbb{Z}_2$.
b. Determine the splitting field of $x^3 + x + 1$ over $\mathbb{Z}_2$.
c. By Theorem 16.2.3, you have described all fields of order 2^3.

Answer

a. $f(x) = x^3 + x + 1$ is reducible if and only if it has a factor of the form $x - a$. By Theorem 16.3.3, $x - a$ is a factor if and only if a is a zero. Neither 0 nor 1 is a zero of $f(x)$ over $\mathbb{Z}_2$.

b. Since $f(x)$ is irreducible over $\mathbb{Z}_2$, all zeros of $f(x)$ must lie in an extension field of $\mathbb{Z}_2$. Let c be a zero of $f(x)$. $\mathbb{Z}_2(c)$ can be described several different ways. One way is to note that since $c \in \mathbb{Z}_2(c)$, $c^n \in \mathbb{Z}_2(c)$ for all n. Therefore, $\mathbb{Z}_2(c)$ includes 0, c, c^2, c^3, ... But $c^3 = c + 1$ since $f(c) = 0$. Furthermore, $c^4 = c^2 + c$, $c^5 = c^2 + c + 1$, $c^6 = c^2 + 1$, and $c^7 = 1$. Higher powers of c repeat preceding powers. Therefore,

$$\begin{aligned}\mathbb{Z}_2(c) &= \left\{0, 1, c, c^2, c+1, c^2+1, c^2+c+1, c^2+c\right\} \\ &= \left\{a_0 + a_1 c + a_2 c^2 \mid a_i \in \mathbb{Z}_2\right\}\end{aligned}$$

The three zeros of $f(x)$ are c, c^2 and $c^2 + c$.

$$f(x) = (x + c)\left(x + c^2\right)\left(x + c^2 + c\right)$$

c. Cite Theorem Theorem 16.2.3, part 3.

Exercise 16.4.6

a. List all polynomials of degree 1, 2, 3, and 4 over $\mathbb{Z}_2 = GF(2)$.
b. From your list in part a, identify all irreducible polynomials of degree 1, 2, 3, and 4.
c. Determine the splitting fields of each of the polynomials in part b.
d. What is the order of each of the splitting fields obtained in part c? Explain your results using Theorem 16.2.3.

Exercise 16.4.7

Is the polynomial code described in this section a linear code?

This page titled 16.4: Field Extensions is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

16.5: Power Series

16.5.1: Definition

Earlier in this chapter, we found that a polynomial of degree n over a ring R is an expression of the form

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

where $n \geq 0$, each of the a_i are elements of R and $a_n \neq 0$. In Section 8.5 we defined a generating function of a sequence s with terms $s_0, s_1, s_2, \ldots$ as the infinite sum

$$G(s, z) = \sum_{i=0}^{\infty} s_i z^i = s_0 + s_1 z + s_2 z^2 + \cdots$$

The main difference between these two expressions, disregarding notation, is that the latter is an infinite expression and the former is a finite expression. In this section we will extend the algebra of polynomials to the algebra of infinite expressions like $G(s, z)$, which are called power series.

Definition 16.5.1: Power Series

Let $[R; +, \cdot]$ be a ring. A power series over R is an expression of the form

$$f(x) = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots$$

where $a_1, a_2, a_3, \ldots \in R$. The set of all such expressions is denoted by R.

Our first observation in our comparison of $R[x]$ and R is that every polynomial is a power series and so $R[x] \subseteq R$. This is true because a polynomial $a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ of degree n in $R[x]$, can be thought of as an infinite expression where $a_i = 0$ for $i > n$. In addition, we will see that R is a ring with subring $R[x]$.

R is given a ring structure by defining addition and multiplication on power series as we did in $R[x]$, with the modification that, since we are dealing with infinite expressions, the sums and products will remain infinite expressions that we can determine term by term, as was done in with polynomials.

Definition 16.5.2: Power Series Addition

Given power series

$$f(x) = \textstyle\sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots \text{ and}$$
$$g(x) = \textstyle\sum_{i=0}^{\infty} b_i x^i = b_0 + b_1 x + b_2 x^2 + \cdots$$

their sum is

$$\begin{aligned} f(x) + g(x) &= \sum_{i=0}^{\infty} (a_i + b_i)\, x^i \\ &= (a_0 + b_0) + (a_1 + b_1)x + (a_2 + b_2)x^2 + (a_3 + b_3)x^3 + \cdots \end{aligned}$$

.

Definition 16.5.3: Power Series Multiplication

Given power series

$$f(x) = \textstyle\sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots \text{ and}$$
$$g(x) = \textstyle\sum_{i=0}^{\infty} b_i x^i = b_0 + b_1 x + b_2 x^2 + \cdots$$

their product is

$$f(x)\cdot g(x) = \sum_{i=0}^{\infty} d_i x^i \quad \text{where } d_i = \sum_{j=0}^{i} a_j b_{i-j}$$
$$= (a_0\cdot b_0) + (a_0\cdot b_1 + a_1\cdot b_0)x + (a_0\cdot b_2 + a_1\cdot b_1 + a_2\cdot b_0)x^2 + \cdots$$

Example 16.5.1: Some Power Series Calculations

Let

$$f(x) = \sum_{i=0}^{\infty} i x^i = 0 + 1x + 2x^2 + 3x^3 + \cdots \quad \text{and}$$
$$g(x) = \sum_{i=0}^{\infty} 2^i x^i = 1 + 2x + 4x^2 + 8x^3 + \cdots$$

be elements in $\mathbb{Z}$. Let us compute $f(x)+g(x)$ and $f(x)\cdot g(x)$. First the sum:

$$\begin{aligned} f(x)+g(x) &= \sum_{i=0}^{\infty} i x^i + \sum_{i=0}^{\infty} 2^i x^i \\ &= \sum_{i=0}^{\infty} \left(i + 2^i\right) x^i \\ &= 1 + 3x + 6x^2 + 11x^3 + \cdots \end{aligned}$$

The product is a bit more involved:

$$\begin{aligned} f(x)\cdot g(x) &= \left(\sum_{i=0}^{\infty} i x^i\right)\cdot\left(\sum_{i=0}^{\infty} 2^i x^i\right) \\ &= \left(0 + 1x + 2x^2 + 3x^3 + \cdots\right)\cdot\left(1 + 2x + 4x^2 + 8x^3 + \cdots\right) \\ &= 0\cdot 1 + (0\cdot 2 + 1\cdot 1)x + (0\cdot 4 + 1\cdot 2 + 2\cdot 1)x^2 + \cdots \\ &= x + 4x^2 + 11x^3 + \cdots \\ &= \sum_{i=0}^{\infty} d_i x^i \qquad \text{where } d_i = \sum_{j=0}^{i} j 2^{i-j} \end{aligned}$$

We can compute any value of d_i, with the amount of time/work required increasing as i increases.

```
def d(i):
    s=0
    for j in range(1,i+1):
        s+=j*2^(i-j)
    return s
d(20)
```

A closed-form expression for d_i would be desirable. Using techniques from Chapter 8, the formula is $d_i = 2^{i+1} - i - 2$, which we leave it to the reader to derive. Hence, $f(x)\cdot g(x) = \sum_{i=0}^{\infty}(2^{i+1} - i - 2)x^i$

16.5.2: Properties, Units

We have seen that addition and multiplication in R is virtually identical to that in $R[x]$. The following theorem parallels Theorem 16.3.1, establishing the ring properties of R.

Theorem 16.5.1: Properties of Power Series

Let $[R; +, \cdot]$ be a ring. Then:

1. R is a ring under the operations of power series addition and multiplication, which depend on the operations in R.
2. If R is a commutative ring, then R is a commutative ring.
3. If R is a ring with unity, 1, then R is a ring with unity (the unity in R[x] is $1 + 0x + 0x^2 + \cdots$).
4. If R is an integral domain, then R is an integral domain.

5. If F is a field, then F is not a field. However, F is an integral domain.

We are most interested in the situation when the set of coefficients is a field. The theorem above indicates that when F is a field, F is an integral domain. A reason that F is not a field is the same as one that we can cite for $F[x]$, namely that x does not have multiplicative inverse in F.

With all of these similarities, one might wonder if the rings of polynomials and power series over a field are isomorphic. It turns out that they are not. The difference between $F[x]$ and F becomes apparent when one studies which elements are units in each. First we prove that the only units in $F[x]$ are the nonzero constants; that is, the nonzero elements of F.

Theorem 16.5.2: Polynomial Units

Let $[F;+,\cdot]$ be a field. Polynomial $f(x)$ is a unit in $F[x]$ if and only if it is a nonzero constant polynomial.

Proof

($\Rightarrow$) Let $f(x)$ be a unit in $F[x]$. Then $f(x)$ has a multiplicative inverse, call it $g(x)$, such that $f(x)\cdot g(x)=1$. Hence, the $\deg(f(x)\cdot g(x))=\deg(1)=0$. But $\deg(f(x)\cdot g(x))=\deg f(x)+\deg g(x)$. So $\deg f(x)+\deg g(x)=0$, and since the degree of a polynomial is always nonnegative, this can only happen when the $\deg f(x)=\deg g(x)=0$. Hence, $f(x)$ is a constant, an element of F, which is a unit if and only if it is nonzero.

($\Leftarrow$) If $f(x)$ is a nonzero element of F, then it is a unit since F is a field. Thus it has an inverse, which is also in $F[x]$ and so $f(x)$ is a unit of $F[x]$.

Before we proceed to categorize the units in F, we remind the reader that two power series $a_0+a_1x+a_2x^2+\cdots$ and $b_0+b_1x+b_2x^2+\cdots$ are equal if and only if corresponding coefficients are equal, $a_i=b_i$ for all $i\geq 0$.

Theorem 16.5.3: Power Series Units

Let $[F;+,\cdot]$ be a field. Then $f(x)=\sum_{i=0}^{\infty}a_ix^i$ is a unit of F if and only if $a_0\neq 0$.

Proof

($\Rightarrow$) If $f(x)$ is a unit of F, then there exists $g(x)=\sum_{i=0}^{\infty}b_ix^i$ in F such that

$$\begin{aligned} f(x)\cdot g(x) &= \left(a_0+a_1x+a_2x^2+\cdots\right)\cdot\left(b_0+b_1x+b_2x^2+\cdots\right)\\ &=1\\ &=1+0x+0x^2+\cdots \end{aligned}.$$

Since corresponding coefficients in the equation above must be equal, $a_0\cdot b_0=1$, which implies that $a_0\neq 0$.

($\Leftarrow$) Assume that $a_0\neq 0$. To prove that $f(x)$ is a unit of F we need to find $g(x)=\sum_{i=0}^{\infty}b_ix^i$ in F such that $f(x)\cdot g(x)=\sum_{i=0}^{\infty}d_ix^i=1$. If we use the formula for the coefficients $f(x)\cdot g(x)$ and equate coefficients, we get

$$\begin{aligned} d_0=a_0\cdot b_0=1 &\quad\Rightarrow\quad b_0={a_0}^{-1}\\ d_1=a_0\cdot b_1+a_1\cdot b_0=0 &\quad\Rightarrow\quad b_1=-{a_0}^{-1}\cdot(a_1\cdot b_0)\\ d_2=a_0b_2+a_1b_1+a_2b_0=0 &\quad\Rightarrow\quad b_2=-{a_0}^{-1}\cdot(a_1\cdot b_1+a_2\cdot b_0)\\ \vdots \qquad\qquad\qquad\qquad &\quad\vdots\quad\vdots\\ d_s=a_0\cdot b_s+a_1\cdot b_{s-1}+\cdots+a_s\cdot b_0=0 &\quad\Rightarrow\quad b_s=-{a_0}^{-1}\cdot(a_1\cdot b_{s-1}+a_2\cdot b_{s-2}+\cdots+a_s\cdot b_0) \end{aligned}.$$

Therefore the powers series $\sum_{i=0}^{\infty}b_ix^i$ is an expression whose coefficients lie in F and that satisfies the statement $f(x)\cdot g(x)=1$. Hence, $g(x)$ is the multiplicative inverse of $f(x)$ and $f(x)$ is a unit.

Example 16.5.2

Let $f(x)=1+2x+3x^2+4x^3+\cdots=\sum_{i=0}^{\infty}(i+1)x^i$ be an element of $\mathbb{Q}$. Then, by Theorem 16.5.3, since $a_0=1\neq 0$, $f(x)$ is a unit and has an inverse, call it $g(x)$. To compute $g(x)$, we follow the procedure outlined in the above theorem. Using the formulas for the b_i's, we obtain

$$b_0 = 1$$
$$b_1 = -1(2 \cdot 1) = -2$$
$$b_2 = -1(2 \cdot (-2) + 3 \cdot 1) = 1$$
$$b_3 = -1(2 \cdot 1 + 3 \cdot (-2) + 4 \cdot 1) = 0$$
$$b_4 = -1(2 \cdot 0 + 3 \cdot 1 + 4 \cdot (-2) + 5 \cdot 1) = 0$$
$$b_5 = -1(2 \cdot 0 + 3 \cdot 0 + 4 \cdot (1) + 5 \cdot (-2) + 6 \cdot 1) = 0$$
$$\vdots$$

For $s \geq 3$, we have

$$b_s = -1(2 \cdot 0 + 3 \cdot 0 + \cdots (s-2) \cdot 0 + (s-1) \cdot 1 + s \cdot (-2) + (s+1) \cdot 1) = 0$$

Hence, $g(x) = 1 - 2x + x^2$ is the multiplicative inverse of $f(x)$.

Certainly F contains a wider variety of units than $F[x]$. Yet F is not a field, since $x \in F$ is not a unit. So concerning the algebraic structure of F, we know that it is an integral domain that contains $F[x]$. If we allow our power series to take on negative exponents; that is, consider expressions of the form $f(x) = \sum_{i=-\infty}^{\infty} a_i x^i$ where all but a finite number of terms with a negative index equal zero. These expressions are called extended power series. The set of all such expressions is a field, call it E. This set does contain, for example, the inverse of x, which is x^{-1}. It can be shown that each nonzero element of E is a unit.

16.5.3: Exercises

Exercise 16.5.1

Let $f(x) = \sum_{i=0}^{\infty} a_i x^i$ and $g(x) = \sum_{i=0}^{\infty} b_i x^i$ be elements of R. Let $f(x) \cdot g(x) = \sum_{i=0}^{\infty} d_i x^i = 1$. Apply basic algebra to $\left(a_0 + a_1 x + a_2 x^2 + \cdots\right) \cdot \left(b_0 + b_1 x + b_2 x^2 + \cdots\right)$ to derive the formula $d_s = \sum_{i=0}^{s} a_i b_{s-i}$ for the coefficients of $f(x) \cdot g(x)$. Hence, to show that $f(x) \cdot g(x) = \sum_{s=0}^{\infty} \left(\sum_{i=0}^{s} a_i b_{s-i}\right) x^s$

Exercise 16.5.2

a. Prove that for any integral domain D, the following can be proven: $f(x) = \sum_{i=0}^{\infty} a_i x^i$ is a unit of D if and only if a_0 is a unit in D.
b. Compare the statement in part a to that in Theorem 16.5.3
c. Give an example of the statement in part a in $\mathbb{Z}$.

Exercise 16.5.3

Use the formula for the product to verify that the expression $g(x)$ of Example 16.5.2is indeed the inverse of $f(x)$.

Exercise 16.5.4

a. Determine the inverse of $f(x) = 1 + x + x^2 + \cdots = \sum_{i=0}^{\infty} x^i$ in $\mathbb{Q}$.
b. Repeat part a with $f(x)$ taken in $\mathbb{Z}_2$.
c. Use the method outlined in Chapter 8 to show that the power series $f(x) = \sum_{i=0}^{\infty} x^i$ is the rational generating function $\frac{1}{1-x}$. What is the inverse of this function? Compare your results with those in part a.

Exercise 16.5.5

a. Determine the inverse of $h(x) = \sum_{i=0}^{\infty} 2^i x^i$ in $\mathbb{Q}$.
b. Use the procedures in Chapter 8 to find a rational generating function for $h(x)$ in part a. Find the multiplicative inverse of this function.

Answer

a.

$$b_0 = 1$$
$$b_1 = (-1)(2 \cdot 1) = -2$$
$$b_2 = (-1)(2 \cdot (-2) + 4 \cdot 1) = 0$$
$$b_3 = (-1)(2 \cdot 0 + 4 \cdot (-2) + 8 \cdot 1) = 0$$

All other terms are zero. Hence, $f(x)^{-1} = 1 - 2x$

b.
$$\begin{aligned} f(x) &= 1 + 2x + 2^2x^2 + 2^3x^3 + \cdots \\ &= (2x)^0 + (2x)^1 + (2x)^2 + (2x)^3 + \cdots \\ &= \frac{1}{1-2x} \end{aligned}$$

The last step follows from the formula for the sum of a geometric series.

Exercise 16.5.6

Let $a(x) = 1 + 3x + 9x^2 + 27x^3 + \cdots = \sum_{i=0}^{\infty} 3^i x^i$ and $b(x) = 1 + x + x^2 + x^3 + \cdots = \sum_{i=0}^{\infty} x^i$ both in $\mathbb{R}$.

a. What are the first four terms (counting the constant term as the 0th term) of $a(x) + b(x)$?
b. Find a closed form expression for $a(x)$.
c. What are the first four terms of $a(x)b(x)$?

Exercise 16.5.7

Write as an extended power series:

a. $\left(x^4 - x^5\right)^{-1}$
b. $\left(x^2 - 2x^3 + x^4\right)^{-1}$

Answer

a.
$$\begin{aligned} \left(x^4 - x^5\right)^{-1} &= (x^4(1-x))^{-1} \\ &= x^{-4}\frac{1}{1-x} \\ &= x^{-4}\left(\sum_{k=0}^{\infty} x^k\right) \cdot \\ &= \sum_{k=-4}^{\infty} x^k \end{aligned}$$

b.
$$\begin{aligned} \left(x^4 - 2x^3 + x^2\right)^{-1} &= \left(x^2\left(x^2 - 2x + 1\right)\right)^{-1} \\ &= x^{-2}\left(1 - 2x + x^2\right)^{-1} \\ &= x^{-2}\left(\sum_{k=0}^{\infty}(k+1)x^k\right) \cdot \\ &= \sum_{k=-2}^{\infty}(k+2)x^k \end{aligned}$$

Exercise 16.5.8

Derive the closed form expression $d_i = 2^{i+1} - i - 2$ for the coefficients of the product $f(x) \cdot g(x)$ in Example 16.5.1.

This page titled 16.5: Power Series is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

17: Appendix

17.1: A - Algorithms

17.1.1: An Introduction to Algorithms

17.1.2: The Invariant Relation Theorem

17.2: B - Python and SageMath

17.2.1: Python Iterators

17.2.2: Dictionaries

17.3: C - Determinants

17.3.1: Definition

17.3.2: Computation

17.4: D - Notation

17.5: E - References

This page titled 17: Appendix is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

17.1: A - Algorithms

> ***algorithm***
>
> *Using step-by-step math operations,*
> *It performs with exact calculations.*
> *An algorithm's job*
> *Is to work out a "prob"*
> *With repeated precise computations.*
>
> *Jesse Frankovich, The Omnificent English Dictionary in Limerick Form*

Computer programs, bicycle assembly instructions, knitting instructions, and recipes all have several things in common. They all tell us how to do something; and the usual format is as a list of steps or instructions. In addition, they are usually prefaced with a description of the raw materials that are needed (the input) to produce the end result (the output). We use the term algorithm to describe such lists of instructions. We assume that the reader may be unfamiliar with algorithms, so the first section of this appendix will introduce some of the components of the algorithms that appear in this book. Since we would like our algorithms to become computer programs in many cases, the notation will resemble a computer language such as Python or Sage; but our notation will be slightly less formal. In some cases we will also translate the pseudocode to Sage. Our goal will be to give mathematically correct descriptions of how to accomplish certain tasks. To this end, the second section of this appendix is an introduction to the Invariant Relation Theorem, which is a mechanism for algorithm verification that is related to Mathematical Induction

This page titled 17.1: A - Algorithms is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

17.1.1: An Introduction to Algorithms

Most of the algorithms in this book will contain a combination of three kinds of steps: the assignment step, the conditional step, and the loop.

17.1.1.1: Assignments

In order to assign a value to a variable, we use an assignment step, which takes the form:

$$\text{Variable} = \text{Expression to be computed}$$

The equals sign in most languages is used for assignment but some languages may use variations such as `:=` or a left pointing arrow. Logical equality, which produces a boolean result and would be used in conditional or looping steps, is most commonly expressed with a double-equals, `==` .

An example of an assignment is `k = n - 1` which tells us to subtract 1 from the value of `n` and assign that value to variable `k` . During the execution of an algorithm, a variable may take on only one value at a time. Another example of an assignment is `k = k - 1` . This is an instruction to subtract one from the value of `k` and then reassign that value to `k` .

17.1.1.2: Conditional steps

Frequently there are steps that must be performed in an algorithm if and only if a certain condition is met. The conditional or "if ... then" step is then employed. For example, suppose that in step 2 of an algorithm we want to assure that the values of variables x and y satisfy the condition `x <= y` . The following step would accomplish this objective.

```
2. If x > y:
    2.1 t = x
    2.2 x = y
    2.3 y = t
```

Listing 17.1.1.1

Steps 2.1 through 2.3 would be bypassed if the condition `x > y` were false before step 2.

One slight variation is the "if ... then ... else" step, which allows us to prescribe a step to be taken if the condition is false. For example, if you wanted to exercise today, you might look out the window and execute the following algorithm.

```
1. If it is cold or raining:
        exercise indoors
    else:
        go outside and run
2. Rest
```

Listing 17.1.1.2

17.1.1.3: 17 Loops

The conditional step tells us to do something once if a logical condition is true. A loop tells us to repeat one or more steps, called the body of the loop, while the logical condition is true. Before every execution of the body, the condition is tested. The following flow diagram serves to illustrate the steps in a While loop.

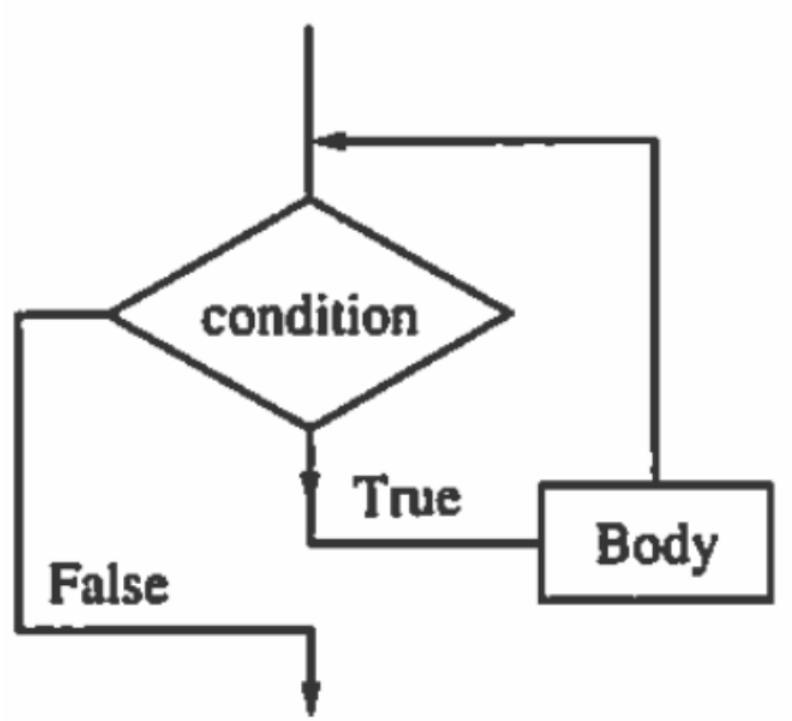


Figure 17.1.1.1: Flow diagram for a while loop

Suppose you wanted to solve the equation $f(x) = 0$. The following initial assignment and loop could be employed.

```
1. c = your first guess
2. While f(c) != 0:
        c = another guess
```

Listing 17.1.1.3

Note

One must always guard against the possibility that the condition of a While loop will never become false. Such "infinite loops" are the bane of beginning programmers. The loop above could very well be such a situation, particularly if the equation has no solution, or if the variable takes on real values

In cases where consecutive integer values are to be assigned to a variable, a different loop construction, a *For loop*, is often employed. For example, suppose we wanted to assign variable `k` each of the integer values from `m` to `n` and for each of these values perform some undefined steps. We could accomplish this with a While loop:

```
1. k := m
2. While k <= n:
    2.1 execute some steps
    2.2 k = k + 1
```

Listing 17.1.1.4

Alternatively, we can perform these steps with a For loop.

```
For k = m to n:
    execute some  steps
```

Listing 17.1.1.5

For loops such as this one have the advantage of being shorter than the equivalent While loop. The While loop construction has the advantage of being able to handle more different situations than the For loop.

17.1.1.4: Exercises

Exercise 17.1.1.1

What are the inputs and outputs of the algorithms listed in the first sentence of this section?

Exercise 17.1.1.2

What is wrong with this algorithm?

```
Input: a and b, integers
Output: the value of c will be a - b
(1) c = 0
(2) While a > b:
        (2.1) a := a - l
        (2.2) c := c + l
```

Listing 17.1.1.6

Answer

The algorithm only works when `a > b` .

Exercise 17.1.1.3

Describe, in words, what the following algorithm does:

```
Input: k, a positive integer
Output: s = ?
(1) s = 0
(2) While k > 0:
    (2.1) s = s + k
    (2.2) k = k - 1
```

Listing 17.1.1.7

Exercise 17.1.1.4

Write While loops to replace the For loops in the following partial algorithms:

a.

1. `S = 0`
2. `for k = 1 to 5: S = S + k^2`

b. The floor of a number is the greatest integer less than or equal to that number.

1. m = a positive integer greater than 1
2. B = floor(sqrt(m))
3. for i = 2 to B: if i divides evenly into m, jump to step 5
4. print "m is a prime" and exit
5. print "m is composite" and exit

Exercise 17.1.1.5

Describe in words what the following algorithm does:

```
Input: n, a positive integer
Output: k?
(1) f= 0
(2) k=n
(3) While k is even:
    (3.1) f = f+ 1
    (3.2) k = k div 2
```

Listing 17.1.1.8

Exercise 17.1.1.6

Fix the algorithm in Exercise 17.1.1.2

17.1.1: An Introduction to Algorithms is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

17.1.2: The Invariant Relation Theorem

17.1.2.1: Exponentiation Algorithms

Consider the following algorithm implemented in Sage to compute $a^m mod\, n$, given an arbitrary integer a, non-negative exponent m, and a modulus n, $n \geq 0$. The default sample evaluation computes $2^5\ mod\, 7 = 32\, mod\, 7 = 4$, but you can edit the final line for other inputs.

```
def slow_exp(a,m,n):
    b=1
    k=m
    while k>0:
        b=(b*a)%n  # % is integer remainder (mod) operation
        k-=1
    return b
slow_exp(2,5,7)
```

It should be fairly clear that this algorithm will successfully compute $a^m (mod\, n)$ since it mimics the basic definition of exponentiation. However, this algorithm is highly inefficient. The algorithm that is most commonly used for the task of exponentiation is the following one, also implemented in Sage.

```
def fast_exp(a,m,n):
    t=a
    b=1
    k=m
    while k>0:
        if k%2==1: b=(b*t)%n
        t=(t^2)%n
        k=k//2  # // is the integer quotient operation
    return b
fast_exp(2,5,7)
```

The only difficulty with the "fast algorithm" is that it might not be so obvious that it always works. When implemented, it can be verified by example, but an even more rigorous verification can be done using the Invariant Relation Theorem. Before stating the theorem, we define some terminology.

17.1.2.1.1: 17.1.2.2: Proving the Correctness of the Fast Algorithm

Definition 17.1.2.1: Pre and Post Values

Given a variable x, the pre value of x, denoted $\grave{x}$, is the value before an iteration of a loop. The post value, denoted $\acute{x}$, is the value after the iteration.

Example 17.1.2.1: Pre and Post Values in the Fast Exponentiation Algorithm

In the fast exponentiation algorithm, the relationships between the pre and post values of the three variables are as follows.

$$\acute{b} \equiv \grave{b}\grave{t}^{\grave{k} mod\, 2} (mod\, n)$$

$$\acute{t} \equiv \grave{t}^2 (mod\, n)$$

$$\acute{k} = \grave{k}//2$$

Definition 17.1.2.2: Invariant Relation

Given an algorithm's inputs and a set of variables that are used in the algorithm, an ***invariant relation*** is a set of one or more equations that are true prior to entering a loop and remain true in every iteration of the loop.

Example 17.1.2.2: Invariant Relation for Fast Exponentiation

We claim that the invariant relation in the fast algorithm is $bt^k = a^m (mod\, n)$. We will prove that this is indeed true below.

Theorem 17.1.2.1: The Invariant Relation Theorem

Given a loop within an algorithm, if R is a relation with the properties

a. R is true before entering the loop
b. the truth of R is maintained in any iteration of the loop
c. the condition for exiting the loop will always be reached in a finite number of iterations.

then R will be true upon exiting the loop.

Proof

The condition that the loop ends in a finite number of iterations lets us apply mathematical induction with the induction variable being the number of iterations. We leave the details to the reader.

We can verify the correctness of the fast exponentiation algorithm using the Invariant Relation Theorem. First we note that prior to entering the loop, $bt^k = 1a^m = a^m (mod\, n)$. Assuming the relation is true at the start of any iteration, that is $\grave{b}\grave{t}^{\grave{k}} = a^m (mod\, n)$, then

$$\begin{aligned} \acute{b}\acute{t}^{\acute{k}} &\equiv (\grave{b}\grave{t}^{\grave{k} \bmod 2})(\grave{t}^2)^{\grave{k}//2} (mod\, n) \\ &\equiv \grave{b}\grave{t}^{2(\grave{k}//2)+\grave{k} \bmod 2} (mod\, n) \\ &\equiv \grave{b}\grave{t}^{\grave{k}} (mod\, n) \\ &\equiv a^m (mod\, n) \end{aligned}$$

Finally, the value of k will decrease to zero in a finite number of steps because the number of binary digits of k decreases by one with each iteration. At the end of the loop,

$$b = bt^0 = bt^k \equiv a^m (mod\, n)$$

which verifies the correctness of the algorithm.

17.1.2.2: Exercises

Exercise 17.1.2.1

How are the pre and post values in the slow exponentiation algorithm related? What is the invariant relation between the variables in the slow algorithm?

Exercise 17.1.2.2

Verify the correctness of the following algorithm to compute the greatest common divisor of two integers that are not both zero.

```
1 def gcd(a,b):
2     r0=a
3     r1=b
4     while r1 !=0:
```

```
        t= r0 % r1
        r0=r1
        r1=t
    return r0
gcd(1001,154)  #test
```

Hint

The invariant of this algorithm is $gcd(r0, r1) = gcd(a, b)$.

Exercise 17.1.2.3

Verify the correctness of the Binary Conversion Algorithm, Algorithm 1.4.1, in Chapter 1.

17.1.2: The Invariant Relation Theorem is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

17.2: B - Python and SageMath

SageMath (originally Sage) is a computer algebra system that is built on top of Python, which is a popular general-purpose programming language. In this appendix we highlight a few features of Python through a series of SageMath cells. Pure Python code can generally be evaluated in these cells and most of what you see here is just Python. There are exceptions. For example, SageMath has enhanced capabilities to work with sets. In Python, the expression `set([0,1,2,3])` is a set of four integers, and certain basic set operations can be performed on these types of expressions. This is a valid expression in SageMath too, but a different SageMath expression, `Set([0,1,2,3])`, with a capital S, has enhanced properties. For example, we can create the power set of the SageMath expression, which we do in the discussion of iterators.

This page titled 17.2: B - Python and SageMath is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

17.2.1: Python Iterators

All programming languages allow for looping. A common form of loop is one in which a series of instructions are executed for each value of some index variable, commonly for values between two integers. Python allows a bit more generality by having structures called “iterators” over which looping can be done. An iterator can be as simple as a list, such as `[0,1,2,3]`, but also can be a power set of a finite set, as we see below, or the keys in a dictionary, which is described in the next section.

17.2.1.1: Counting Subsets

Suppose we want to count the number of subsets of $\{0,1,2,\ldots,9\}$ that contain no adjacent elements. First, we will define our universe and its power set. The plan will be to define a function that determines whether a subset is "valid" in the sense that it contains no adjacent elements. Then we will iterate over the subsets, counting the valid ones. We know that the number of all subsets will be 2 raised to the number of elements in U, which would be $2^{10}=1024$, but let's check.

```
U=Set(range(10))
power_set=U.subsets()
len(power_set)
```

The validity check in this case is very simple. For each element, k, of a set, B, we ask whether its successor, $k+1$, is also in the set. If we never get an answer of "True" then we consider the set valid. This function could be edited to define validity in other ways to answer different counting questions. It's always a good idea to test your functions, so we try two tests, one with a valid set and one with an invalid one.

```
def valid(B):
    v=true
    for k in B:
        if k+1 in B:
            v=false
            break
    return v
[valid(Set([1,3,5,9])),valid(Set([1,2,4,9]))]
```

Finally we do the counting over our power set, incrementing the count variable with each valid set.

```
count=0
for B in power_set:
    if valid(B):
        count+=1
count
```

17.2.1: Python Iterators is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

17.2.2: Dictionaries

17.2.2.1: Colors of Fruits

In Python and SageMath, a dictionary is a convenient data structure for establishing a relationship between sets of data. From the point of view of this text, we can think of a dictionary as a concrete realization of a relation between two sets or on a single set. A dictionary resembles a function in that there is a set of data values called the `keys`, and for each key, there is a `value`. The value associated with a key can be almost anything, but it is most commonly a list.

To illustrate the use of dictionaries, we will define a relationship between colors and fruits. The keys will be a set of colors and values associated with each color will be a list of fruits that can take on that color. We will demonstrate how to initialize the dictionary and how to add to it. The following series of assignments have no output, so we add a print statement to verify that this cell is completely evaluated.

```
fruit_color={}
fruit_color['Red']=['apple','pomegranate','blood orange']
fruit_color['Yellow']=['banana','apple','lemon']
fruit_color['Green']=['apple','pear','grape','lime']
fruit_color['Purple']=['plum','grape']
fruit_color['Orange']=['orange','pineapple']
print('done')
```

We distinguish a color from a fruit by capitalizing colors but not fruit. The keys of this dictionary are the colors. The `keys()` method returns an interator; so to get a list of keys we wrap the result with `list()`.

```
list(fruit_color.keys())
```

As an afterthought, we might add the information that a raspberry is red as follows. You have to be careful in that if 'Red' isn't already in the dictionary, it doesn't have a value. This is why we need an if statement.

```
if 'Red' in fruit_color:
    fruit_color['Red']=fruit_color['Red']+['raspberry']
else:
    fruit_color['Red']=['raspberry']
fruit_color['Red']
```

A dictionary is iterable, with an iterator taking on values that are the keys. Here we iterate over our dictionary to output lists consisting of a color followed by a list of fruits that come in that color.

```
for fruit in fruit_color:
    print([fruit,fruit_color[fruit]])
```

We can view a graph of this relation between colors and fruits, but the default view is a bit unconventional.

```
DiGraph(fruit_color).plot()
```

With a some additional coding we can line up the colors and fruits in their own column. First we set the positions of colors on the left with all x-coordinates equal to -5 using another dictionary called `vertex_pos`.

```
vertex_pos={}
k=0
for c in fruit_color.keys():
    vertex_pos[c]=(-5,k)
    k+=1
vertex_pos
```

Next, we place the fruit vertices in another column with x-coordinates all equal to 5. In order to do this, we first collect all the fruit values into one set we call `fruits` .

```
fruits=Set([ ])
for v in fruit_color.values():
     fruits=fruits.union(Set(v))
k=0
for f in fruits:
    vertex_pos[f]=(5,k)
    k+=1
vertex_pos
```

Now the graph looks like a conventional graph for a relation between two different sets. Notice that it's not a function.

```
DiGraph(fruit_color).plot(pos=vertex_pos,vertex_size=1)
```

17.2.2: Dictionaries is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

17.3: C - Determinants

In Chapter 5 we defined the determinant of a 2×2 matrix for the sole purpose of providing some hands-on experience in the computation of inverses of 2×2 matrices. In this appendix we will define the determinant of any square matrix, and summarize the main properties of determinants.

This page titled 17.3: C - Determinants is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

17.3.1: Definition

Associated with every square matrix is a number called its determinant. The most important information it provides us with is whether the matrix is invertible. A matrix has an inverse if and only if its determinant is nonzero. If A is a square matrix, then the determinant of A is commonly denoted either $\det(A)$ or $|A|$. Strictly speaking, we only need to define the determinant of a 1×1 matrix here and then define the higher ordered ones recursively, but for convenience we also recall the definition of the determinant of a 2×2 matrix.

Definition 17.3.1.1: Determinant of 1×1 and a 2×2 Matrices

- If A is a 1×1 matrix, then $|A| = A_{1,1}$
- If A is a 2×2 matrix, then $|A| = A_{1,1}A_{2,2} - A_{1,2}A_{2,1}$

We now proceed to define the determinant of an $n\times n$ matrix where $n>2$. This definition requires two preliminary definitions those of minors and cofactors.

Definition 17.3.1.2: Matrix Minor

Let A be an $n\times n$ matrix, $n\geq 2$. The determinant of the $(n-1)\times(n-1)$ matrix formed by removing the ith row and j^{th} column of A is the minor denoted by $M(A)_{i,j}$.

Example 17.3.1.1

Let $A=\begin{pmatrix}3&4&1\\1&3&4\\4&1&3\end{pmatrix}$ then A has nine minors, one of which is

$$M(A)_{1,1}=\begin{vmatrix}3&4\\1&3\end{vmatrix}=3\cdot 3-4\cdot 1=5$$

For our purposes in computing $|A|$, we only need minors corresponding to any one row or column. Completing the minors in the first row we have $M(A)_{1,2}=-13$ and $M(A)_{1,3}=-11$

Definition 17.3.1.3: Cofactor

Let A be an $n\times n$ matrix, $n\geq 2$. The ith row, j^{th} column cofactor of A, denoted $C(A)_{i,j}$, is defined by

$$C(A)_{i,j}=(-1)^{i+j}M(A)_{i,j}$$

Example 17.3.1.2

Using the values of minors computed in Example 17.3.1.1, we have $C(A)_{1,1}=(-1)^2M(A)_{1,1}=5$, $C(A)_{1,2}=(-1)^3M(A)_{1,2}=13$, and $C(A)_{1,3}=(-1)^4M(A)_{1,3}=-11$.

Finally, we will define the determinant of a square matrix. Our definition is practical in that you can apply it easily to any matrix. It isn't the most general, nor is it the best definiton for the purposes of proving properties of determinants. The more general definition is beyond our current scope, but can be easily stated with background in permutation groups.

Definition 17.3.1.4: Determinant of a Square Matrix

Let A be an $n\times n$ matrix, $n\geq 2$. The determinant of A is equal to

$$\sum_{j=1}^{n}A_{1,j}\cdot C(A)_{1,j}$$

Our definition of a determinant involves what is called expansion along the first row of the matrix A. It is certainly not obvious, but it is true, that the determinant of a matrix can be found by expanding along any row or any column.

Example 17.3.1.3

We have computed the cofactors for row 1 of $A = \begin{pmatrix} 3 & 4 & 1 \\ 1 & 3 & 4 \\ 4 & 1 & 3 \end{pmatrix}$ above and so the determinant is only a few operations away.

$$\begin{aligned} |A| &= A_{1,1} \cdot C(A)_{1,1} + A_{1,2} \cdot C(A)_{1,2} + A_{1,3} \cdot C(A)_{1,3} \\ &= 3 \cdot 5 + 4 \cdot 13 + 1 \cdot (-11) \\ &= 56 \end{aligned}$$

Example 17.3.1.4

Associated with any square matrix, A, is a characteristic polynomial which is defined to be the $|A - \lambda I|$. The roots of this polynomial are the eigenvalues of the matrix. Here, we compute the characteristic polynomial of $A = \begin{pmatrix} 3 & 4 & 1 \\ 1 & 3 & 4 \\ 4 & 1 & 3 \end{pmatrix}$.

To compute the determinant we expand along the first row.

$$\begin{aligned} \det(A - \lambda I) &= \begin{vmatrix} 3-\lambda & 4 & 1 \\ 1 & 3-\lambda & 4 \\ 4 & 1 & 3-\lambda \end{vmatrix} \\ &= (3-\lambda) \cdot \begin{vmatrix} 3-\lambda & 4 \\ 1 & 3-\lambda \end{vmatrix} + 4 \cdot (-1) \cdot \begin{vmatrix} 1 & 4 \\ 4 & 3-\lambda \end{vmatrix} + 1 \cdot \begin{vmatrix} 1 & 3-\lambda \\ 4 & 1 \end{vmatrix} \\ &= (3-\lambda)((3-\lambda)^2 - 4) - 4((3-\lambda) - 16) + (1 - 4(3-\lambda)) \\ &= -\lambda^3 + 9\lambda^2 - 15\lambda + 56 \end{aligned}$$

17.3.1: Definition is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

17.3.2: Computation

Our definition of determinant can be applied to estimate the worst case for the time to evaluate an $n \times n$ determinant. Let $M(n)$ be the number of multiplications to evaluate an $n \times n$ determinant. Then we have $M(2) = 2$. To determine the value of $M(3)$ we observe that this requires the computation of three minors, each a two by two matrix, and then a multiplication of each of them by the entries in row 1. Therefore, $M(3) = 3M(2) + 3 = 9$. Using the same logic in general, we have $M(n) = nM(n-1) + n$. The formula can be derived to be $M(n) = n! \sum_{k=1}^{n} \frac{1}{k!}$. For large n this is approximately $e \cdot n!$. Fortunately, there are ways to reduce the number of multiplications using properties of determinants, which we list here without proof.

Theorem 17.3.2.1: Properties of Determinants

Let A and B be $n \times n$ matrices, where $n \geq 2$.

1. $|A|$ can be found by expanding along any row or any column.
2. If two rows (or columns) of A are interchanged, $|A|$ changes sign.
3. The value of a determinant is unchanged if a multiple of one row (or column) of A is added to another row (or column) of A .
4. If one row (or column) of a matrix A is multiplied by a constant c, then the value of $|A|$ is multiplied by c.
5. $|AB| = |A| \cdot |B|$.
6. $|I| = 1$ where I is the $n \times n$ identity matrix.

Based on these properties, here are a few corollaries.

Corollary 17.3.2.1: Further Properties

Let A and B be $n \times n$ matrices, where $n \geq 2$.

1. If a row (or column) of A consists entirely of zeros, then $|A| = 0$.
2. If a matrix A has two equal rows (or columns) then $|A| = 0$.
3. If any row (or column) of A is a scalar multiple of any other row (or column) of A, then $|A| = 0$.
4. $|A^{-1}| = \frac{1}{|A|}$, if A^{-1} exists.

Example 17.3.2.1: Computation of a Determinant by Row Reduction

We will apply some of these properties, most notably the first and third of Theorem 17.3.2.1, to compute a four by four determinant without doing as many multiplications as expected. We will use SageMath to do the calculations for us. In SageMath, as in Python, numbering starts at zero, so we will describe the process using that numbering system. Let

$$A = \begin{pmatrix} 1 & 3 & 4 & 7 \\ 1 & 3 & 4 & 4 \\ 6 & 6 & 7 & 8 \\ 3 & 3 & 7 & 5 \end{pmatrix}$$

Our strategy will be to create a column that is mostly zero so that we can expand along that column and only need to compute one cofactor. That will be the 0th column. To do that we do the following row operations. We subtract row 0 from row 1, replacing row 1 with that result. Then we subtract six time row 0 from row 2, producing a new row 2. Finally, three times row 0 is subtracted from row 3 to produce a new row 3. The SageMath code below accomplishes this and produces a new matrix, B, which has the same determinant.

```
A=matrix([[1,3,4,7],[2,3,4,4],[5,6,7,4],[3,3,7,5]])
B=matrix([A[0],A[1]-2*A[0],A[2]-5*A[0],A[3]-3*A[0]]);B
```

Expanding this matrix along the column zero, we need only compute a single three by three cofactor. We will go one step further and do row operations to get a matrix with zeros in rows 2 and 3 of column 1. The SageMath code below tells what we are doing.

```
C=matrix([B[0],B[1],B[2]-3*B[1],B[3]-2*B[1]]);C
```

We are at a point where we can do the final calculation very easily.

$$|A| = |C| = 1 \cdot (-3 \cdot (-1 \cdot 4 - 3 \cdot (-1))) = 3$$

SageMath has a determinant function, `det` , that we can use to verify this calculation:

```
A=matrix([[1,3,4,7],[2,3,4,4],[5,6,7,4],[3,3,7,5]])
det(A)
```

17.3.2: Computation is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

17.4: D - Notation

The following table defines the notation used in this book. Page numbers or references refer to the first appearance of each symbol.

Symbol	Description	Location
$x \in A$	x is an element of A	Paragraph
$x \notin A$	x is not an element of A	Paragraph
$\mathbb{P}$	the positive integers	Item
$\mathbb{N}$	the natural numbers	Item
$\mathbb{Z}$	the integers	Item
$\mathbb{Q}$	the rational numbers	Item
$\mathbb{R}$	the real numbers	Item
$\mathbb{C}$	the complex numbers	Item
$\lvert A \rvert$	The number of elements in a finite set A.	Definition 1.1.2
$A \subseteq B$	A is a subset of B.	Definition 1.1.3
$\emptyset$	the empty set	Paragraph
$A \cap B$	The intersection of A and B.	Definition 1.2.1
$A \cup B$	The union of A and B.	Definition 1.2.3
$B - A$	The complement of A relative to B	Definition 1.2.5
A^c	The complement of set A relative to the universe.	Definition 1.2.5
$A \oplus B$	The symmetric difference of A and B.	Definition 1.2.6
$A \times B$	The cartesian product of A with B.	Definition 1.3.1
$\mathcal{P}(A)$	The power set of A, the set of all subsets of A.	Definition 1.3.2
$n!$	n factorial, the product of the first n positive integers	Definition 2.2.1
$\binom{n}{k}$	n choose k, the number of k element subsets of an n element set.	Definition 2.4.1
$p \wedge q$	the conjunction, p and q	Definition 3.1.2
$p \vee q$	the disjunction, p or q	Definition 3.1.3
$\neg p$	the negation of p, "not p"	Definition 3.1.4
$p \rightarrow q$	The conditional proposition If p then q.	Definition 3.1.5
$p \leftrightarrow q$	The biconditional proposition p if and only if q	Definition 3.1.8
1	symbol for a tautology	Definition 3.3.1
0	symbol for a contradiction	Definition 3.3.2
$r \iff s$	r is logically equivalent to s	Definition 3.3.3
$r \Rightarrow s$	r implies s	Definition 3.3.4

Symbol	Description	Location
$p\|q$	the Sheffer Stroke of p and q	Definition 3.3.5
T_p	the truth set of p	Definition 3.6.2
$(\exists n)_U(p(n))$	The statement that $p(n)$ is true for at least one value of n	Definition 3.8.1
$(\forall n)_U(p(n))$	The statement that $p(n)$ is always true.	Definition 3.8.2
$\mathbf{0}_{m\times n}$	the m by n zero matrix	Item
I_n	The $n \times n$ identity matrix	Definition 5.2.2
A^{-1}	A inverse, the multiplicative inverse of A	Definition 5.2.3
$\det\ A$ or $\|A\|$	The determinant of A, 2 by 2 case	Definition 5.2.4
$a\|b$	a divides b, or a divides evenly into b	Definition 6.1.3
xsy	x is related to y through the relation s	Paragraph
rs	the composition of relation r with relation s	Definition 6.1.4
$a \equiv_n b$	a is congruent to b modulo n	Definition 6.3.7
$a \equiv b (\mod n)$	a is congruent to b modulo n	Definition 6.3.7
$c(a)$	the equivalence class of a under r	Item
r^+	The transitive closure of r	Definition 6.5.1
$f : A \to B$	A function, f, from A into B	Definition 7.1.1
B^A	The set of all functions from A into B	Definition 7.1.2
$f(a)$	The image of a under f	Definition 7.1.3
$f(X)$	Range of function $f : X \to Y$	Definition 7.1.4
$\mathcal{X}s$	Characteristic function of the set S	Exercise 7.1.4
$\|A\| = n$	A has cardinality n	Definition 7.2.4
$(g \circ f)(x) = g(f(x))$	The composition of g with f	Definition 7.3.2
$f \circ f = f^2$	the "square" of a function.	Definition 7.3.3
i or i_A	The identity function (on a set A)	Definition 7.3.4
f^{-1}	The inverse of function f read "f inverse"	Definition 7.3.5
$log_b a$	Logarithm, base b of a	Definition 8.4.2
$S\uparrow$	S pop	Definition 8.5.2
$S\downarrow$	S push	Definition 8.5.2
$S * T$	Convolution of sequences S and T	Definition 8.5.2
$S\uparrow p$	Multiple pop operation on S	Definition 8.5.3
$S\downarrow p$	Multiple push operation on S	Definition 8.5.3
K_n	A complete undirected graph with n vertices	Definition 9.1.4

Symbol	Description	Location
$deg(v), indeg(v), outdeg(v)$	degree, indegree, and outdegree of vertex v	Definition 9.1.9
$e(v)$	The eccentricity of a vertex	Definition 9.3.1
$d(G)$	The diameter of graph G	Definition 9.3.2
$r(G)$	The radius of graph G	Definition 9.3.3
$C(G)$	The center of graph G	Definition 9.3.4
Q_n	the n-cube	Definition 9.4.3
$V(f)$	The value of flow f	Definition 9.5.3
P_n	a path graph of length n	Definition 9.6.2
$\chi(G)$	the chromatic number of G	Definition 9.6.3
C_n	A cycle with n edges.	Definition 10.1.1
$*$	generic symbol for a binary operation	Definition 11.1.1
$string1 + string2$	The concatenation of $string1$ and $string2$	Item
$[G; *]$	a group with elements G and binary operation $*$	Definition 11.2.1
$\gcd(a, b)$	the greatest common divisor of a and b	Definition 11.4.1
$a +_n b$	the mod n sum of a and b	Definition 11.4.3
$a \times_n b$	the mod n product of a and b	Definition 11.4.4
$\mathbb{Z}_n$	The Additive Group of Integer Modulo n	Definition 11.4.5
$\mathbb{U}_n$	The Multiplicative Group of Integer Modulo n	Definition 11.4.6
$W \leq V$	W is a subsystem of V	Definition 11.5.1
$\langle a \rangle$	the cyclic subgroup generated by a	Definition 11.5.2
$ord(a)$	Order of a	Definition 11.5.3
$V_1 \times V_2 \times \cdots \times V_n$	The direct product of algebraic structures $V_1, V_2, \ldots, V_n$	Definition 11.6.1
$G_1 \cong G_2$	G_1 is isomorphic to G_2	Definition 11.7.2
$dim(V)$	The dimension of vector space V	Definition 12.3.6
$\mathbf{0}$	least element in a poset	Definition 13.1.5
$\mathbf{1}$	greatest element in a poset	Definition 13.1.5
D_n	the set of divisors of integer n	Definition 13.1.6
$a \vee b$	the join, or least upper bound of a and b	Definition 13.2.1
$a \wedge b$	the meet, or greatest lower bound of a and b	Definition 13.2.1
$[L; \vee, \wedge]$	A lattice with domain having meet and join operations	Definition 13.2.2
$\bar{a}$	The complement of lattice element a	Definition 13.3.4

Symbol	Description	Location
$[B; \vee, \wedge, \bar{\ }\ ]$	a boolean algebra with operations join, meet and complementation	Definition 13.3.5
$M_{\delta_1 \delta_2 \cdots \delta_k}$	the minterm generated by $x_1, x_2, \ldots, x_k$, where $y_i = x_i$ if $\delta_i = 1$ and $y_i = \bar{x_i}$ if $\delta_i = 0$	Definition 13.6.2
A^*	The set of all strings over an alphabet A	Definition 14.2.1
A^n	The set of all strings of length n over an alphabet A	Definition 14.2.1
λ	The empty string	Definition 14.2.1
$s_1 + s_2$	The concatenation of strings s_1 and s_2	Definition 14.2.2
$L(G)$	Language created by phrase structure grammar G	Definition 14.2.5
(S, X, Z, w, t)	A finite-state machine with states S, input alphabet X, output alphabet X, and output function w and next-state function t	Definition 14.3.1
$m(M)$	The machine of monoid M	Definition 14.5.1
$a * H, H * a$	the left and right cosets generated by a	Definition 15.2.1
G/H	The factor group G mod H.	Definition 15.2.4
S_A	The group of permutations of the set A	Definition 15.3.1
S_n	The group of permutations on a set with n elements	Definition 15.3.1
A_n	The Alternating Group	Definition 15.3.3
$\mathcal{D}_n$	The nth dihedral group	Definition 15.3.4
$H \triangleleft G$	H is a normal subgroup of G	Definition 15.4.1
$ker\theta$	the kernel of homomorphism θ	Definition 15.4.4
$[R; +, \cdot]$	a ring with domain R and operations $+$ and $\cdot$.	Definition 16.1.1
$U(R)$	the set of units of a ring R	Definition 16.1.4
D	a generic integral domain	Definition 16.1.7
deg $f(x)$	the degree of polynomial $f(x)$	Definition 16.3.1
$R[x]$	the set of all polynomials in x over R	Definition 16.3.1
$R\,[[x]]$	the set of all power series in R	Definition 16.5.1
$\grave{x}, \acute{x}$	pre and post values of a variable x	Definition 17.1.2.1
$M(A)_{i,j}$	The i, j minor of A	Definition 17.3.1.2
$C(A)_{i,j}$	The i, j cofactor of A	Definition 17.3.1.3
$\det(A)$ or $\lvert A \rvert$	The determinant of A	Definition 17.3.1.4

This page titled 17.4: D - Notation is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

17.5: E - References

Many of the references listed here were used in preparing the original 1980's version of this book. In most cases, the mathematics that they contain is still worth reading for further background. Many can be found online, in university libraries or used bookstores. A few more current references have been added.

[1] Allenby, R.B.J.T, *Rings, Fields and Groups*, Edward Arnold, 1983.

[2] Appel, K., and W. Haken, *Every Planar Map Is 4-colorable*, Bull, Am. Math. Soc. no. 82 (1976): 711–12.

Note 17.5.1

This has historical significance in that it announced the first correct proof of the Four Color Theorem

[3] Austin, A. Keith, *An Elementary Approach to NP-Completeness* American Math. Monthly 90 (1983): 398-99.

[4] Beardwood, J., J. H. Halton, and J. M. Hammersley, *The Shortest Path Through Many Points* Proc. Cambridge Phil. Soc. no. 55 (1959): 299–327.

[5] Ben-Ari, M, *Principles of Concurrent Programming*, Englewood Cliffs, NJ: Prentice-Hall, 1982.

[6] Berge, C, *The Theory of Graphs and Its Applications*, New York: Wiley, 1962.

[7] Bogart, Kenneth P, *Combinatorics Through Guided Discovery*, 2005.

Note 17.5.2

This book may be freely downloaded and redistributed under the terms of the GNU Free Documentation License (FDL), as published by the Free Software Foundation.

[8] Busacker, Robert G., and Thomas L. Saaty, *Finite Graphs and Networks*, New York: McGraw-Hill, 1965.

[9] Connell, Ian, *Modern Algebra, A Constructive Introduction*, New York: North-Holland, 1982.

[10] Denning, Peter J., Jack B. Dennis, and Joseph L. Qualitz, *Machines, Languages, and Computation*, Englewood Cliffs, NJ: Prentice-Hall, 1978.

[11] Denning, Peter J, *Multigrids and Hypercubes*. American Scientist 75 (1987): 234-238.

[12] Dornhoff, L. L., and F. E. Hohn, *Applied Modern Algebra*, New York: Macmillan, 1978.

[13] Ford, L. R., Jr., and D. R. Fulkerson, *Flows in Networks*, Princeton, NJ: Princeton Univesity Press, 1962.

[14] Fraleigh, John B, *A First Course in Abstract Algebra*, 3rd ed. Reading, MA: Addison-Wesley, 1982.

[15] Gallian, Joseph A, *Contemporary Abstract Algebra*, D.C. Heath, 1986.

[16] Gallian, Joseph A, *Group Theory and the Design of a Letter-Facing Machine*, American Math. Monthly 84 (1977): 285-287.

[17] Hamming, R. W, *Coding and Information Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1980.

[18] Hill, F. J., and G. R. Peterson, *Switching Theory and Logical Design*, 2nd ed. New York: Wiley, 1974.

[19] Hofstadter, D. R, *Godel, Escher, Bach: An Eternal Golden Braid*, New York: Basic Books, 1979.

[20] Hohn, F. E, *Applied Boolean Algebra*, 2nd ed. New York: Macmillan, 1966.

[21] Hopcroft, J. E., and J. D. Ullman, *Formal Languages and Their Relation to Automata*, Reading, MA: Addison-Wesley, 1969.

[22] Hu, T. C, *Combinatorial Algorithms*, Reading, MA: Addison-Wesley, 1982.

[23] Knuth, D. E, *The Art of Computer Programming. Vol. 1, Fundamental Algorithms*, 2nd ed. Reading, MA: Addison-Wesley, 1973.

[24] Knuth, D. E, *The Art of Computer Programming. Vol. 2, Seminumerical Algorithms*, 2nd ed., Reading, MA: Addison-Wesley, 1981.

[25] Knuth, D. E, *The Art of Computer Programming. Vol. 3, Sorting and Searching*, Reading, MA: Addison-Wesley, 1973.

[26] Knuth, D. E, *The Art of Computer Programming. Vol. 4A, Combinatorial Algorithms, Part 1*, Upper Saddle River, New Jersey: Addison-Wesley, 2011.

Note 17.5.3

https://www-cs-faculty.stanford.edu/~knuth/taocp.html

[27] Kulisch, U. W., and Miranker, W. L, *Computer Arithmetic in Theory and Practice*, New York: Academic Press, 1981.

[28] Lipson, J. D, *Elements of Algebra and Algebraic Computing*, Reading, MA: Addison-Wesley, 1981.

[29] Liu, C. L, *Elements of Discrete Mathematics*, New York: McGraw-Hill, 1977.

[30] O'Donnell, *Analysis of Boolean Functions*.

Note 17.5.4

A book about Fourier analysis of boolean functions that is being developed online in a blog.

[31] *The Omnificent English Dictionary In Limerick Form* .

Note 17.5.5

The source of all limericks that appear at the beginning of most chapters. https://www.oedilf.com/

[32] Ore, O, *Graphs and Their Uses*, New York: Random House, 1963.

[33] Parry, R. T., and H. Pferrer, *The Infamous Traveling-Salesman Problem: A Practical Approach* Byte 6 (July 1981): 252-90.

[34] Pless, V, *Introduction to the Theory of Error-Correcting Codes*, New York: Wiley-Interscience, 1982.

[35] Purdom, P. W., and C. A. Brown, *The Analysis of Algorithms*, Holt, Rinehart, and Winston, 1985.

[36] Quine, W. V, *The Ways of Paradox and Other Essays*, New York: Random House, 1966.

[37] Ralston, A, *The First Course in Computer Science Needs a Mathematics Corequisite*, Communications of the ACM 27-10 (1984): 1002-1005.

[38] Solow, Daniel, *How to Read and Do Proofs*, New York: Wiley, 1982.

[39] Sopowit, K. J., E. M. Reingold, and D. A. Plaisted *The Traveling Salesman Problem and Minimum Matching in the Unit Square*.SIAM J. Computing, 1983,**12**, 144–56.

[40] Standish, T. A, *Data Structure Techniques*, Reading, MA: Addison-Wesley, 1980.

[41] Stoll, Robert R, Sets, *Logic and Axiomatic Theories*, San Francisco: W. H. Freeman, 1961.

[42] Strang, G, *Linear Algebra and Its Applications*, 2nd ed. New York: Academic Press, 1980.

[43] Tucker, Alan C, *Applied Combinatorics*, 2nd ed. New York: John Wiley and Sons, 1984.

[44] Wand, Mitchell, *Induction, Recursion, and Programming*, New York: North-Holland, 1980.

[45] Warshall, S, *A Theorem on Boolean Matrices* Journal of the Association of Computing Machinery, 1962, 11-12.

[46] Weisstein, Eric W. *Strassen Formulas*, MathWorld--A Wolfram Web Resource, http://mathworld.wolfram.com/StrassenFormulas.html.

[47] Wilf, Herbert S, *Some Examples of Combinatorial Averaging*, American Math. Monthly 92 (1985).

[48] Wilf, Herbert S. *generatingfunctionology*, A K Peters/CRC Press, 2005

Note 17.5.6

The 1990 edition of this book is available at https://www.math.upenn.edu/~wilf/DownldGF.html

[49] Winograd, S, *On the Time Required to Perform Addition*, J. Assoc. Comp. Mach. 12 (1965): 277-85.

[50] Wilson, R., *Four Colors Suffice - How the Map Problem Was Solved*Princeton, NJ: Princeton U. Press, 2013.

This page titled 17.5: E - References is shared under a CC BY-NC-SA 3.0 license and was authored, remixed, and/or curated by Al Doerr & Ken Levasseur via source content that was edited to the style and standards of the LibreTexts platform.

Index

1: M

monoids
14.1: Monoids

2: P

permutations
2.2: Permutations

3: S

semigroup
14.1: Monoids

set notation
1.1: Set Notation and Relations

stochastic matrices
14.1: Monoids

subset
1.1: Set Notation and Relations

Glossary

Sample Word 1 | Sample Definition 1

https://math.libretexts.org/@go/page/80586

Detailed Licensing

Overview

Title: Applied Discrete Structures (Doerr and Levasseur)

Webpages: 126

Applicable Restrictions: Noncommercial

All licenses found:

- CC BY-NC-SA 3.0: 92.9% (117 pages)
- Undeclared: 7.1% (9 pages)

By Page

- Applied Discrete Structures (Doerr and Levasseur) - *CC BY-NC-SA 3.0*
  - Front Matter - *CC BY-NC-SA 3.0*
    - TitlePage - *CC BY-NC-SA 3.0*
    - InfoPage - *CC BY-NC-SA 3.0*
    - Table of Contents - *Undeclared*
    - Colophon - *CC BY-NC-SA 3.0*
    - Licensing - *Undeclared*
    - Dedication - *CC BY-NC-SA 3.0*
    - Acknowledgements - *CC BY-NC-SA 3.0*
    - Preface - *CC BY-NC-SA 3.0*
  - 1: Set Theory - *CC BY-NC-SA 3.0*
    - 1.1: Set Notation and Relations - *CC BY-NC-SA 3.0*
    - 1.2: Basic Set Operations - *CC BY-NC-SA 3.0*
    - 1.3: Cartesian Products and Power Sets - *CC BY-NC-SA 3.0*
    - 1.4: Binary Representation of Positive Integers - *CC BY-NC-SA 3.0*
    - 1.5: Summation Notation and Generalizations - *CC BY-NC-SA 3.0*
  - 2: Combinatorics - *CC BY-NC-SA 3.0*
    - 2.1: Basic Counting Techniques - The Rule of Products - *CC BY-NC-SA 3.0*
    - 2.2: Permutations - *CC BY-NC-SA 3.0*
    - 2.3: Partitions of Sets and the Law of Addition - *CC BY-NC-SA 3.0*
    - 2.4: Combinations and the Binomial Theorem - *CC BY-NC-SA 3.0*
  - 3: Logic - *CC BY-NC-SA 3.0*
    - 3.1: Propositions and Logical Operators - *CC BY-NC-SA 3.0*
    - 3.2: Truth Tables and Propositions Generated by a Set - *CC BY-NC-SA 3.0*
    - 3.3: Equivalence and Implication - *CC BY-NC-SA 3.0*
    - 3.4: The Laws of Logic - *CC BY-NC-SA 3.0*
    - 3.5: Mathematical Systems and Proofs - *CC BY-NC-SA 3.0*
    - 3.6: Propositions over a Universe - *CC BY-NC-SA 3.0*
    - 3.7: Mathematical Induction - *CC BY-NC-SA 3.0*
    - 3.8: Quantifiers - *CC BY-NC-SA 3.0*
    - 3.9: A Review of Methods of Proof - *CC BY-NC-SA 3.0*
  - 4: More on Sets - *CC BY-NC-SA 3.0*
    - 4.1: Methods of Proof for Sets - *CC BY-NC-SA 3.0*
    - 4.2: Laws of Set Theory - *CC BY-NC-SA 3.0*
    - 4.3: Minsets - *CC BY-NC-SA 3.0*
    - 4.4: The Duality Principle - *CC BY-NC-SA 3.0*
  - 5: Introduction to Matrix Algebra - *CC BY-NC-SA 3.0*
    - 5.1: Basic Definitions and Operations - *CC BY-NC-SA 3.0*
    - 5.2: Special Types of Matrices - *CC BY-NC-SA 3.0*
    - 5.3: Laws of Matrix Algebra - *CC BY-NC-SA 3.0*
    - 5.4: Matrix Oddities - *CC BY-NC-SA 3.0*
  - 6: Relations - *CC BY-NC-SA 3.0*
    - 6.1: Basic Definitions - *CC BY-NC-SA 3.0*
    - 6.2: Graphs of Relations on a Set - *CC BY-NC-SA 3.0*
    - 6.3: Properties of Relations - *CC BY-NC-SA 3.0*
    - 6.4: Matrices of Relations - *CC BY-NC-SA 3.0*
    - 6.5: Closure Operations on Relations - *CC BY-NC-SA 3.0*
  - 7: Functions - *CC BY-NC-SA 3.0*
    - 7.1: Definition and Notation - *CC BY-NC-SA 3.0*
    - 7.2: Properties of Functions - *CC BY-NC-SA 3.0*
    - 7.3: Function Composition - *CC BY-NC-SA 3.0*
  - 8: Recursion and Recurrence Relations - *CC BY-NC-SA 3.0*
    - 8.1: The Many Faces of Recursion - *CC BY-NC-SA 3.0*
    - 8.2: Sequences - *CC BY-NC-SA 3.0*
    - 8.3: Recurrence Relations - *CC BY-NC-SA 3.0*
    - 8.4: Some Common Recurrence Relations - *CC BY-NC-SA 3.0*
    - 8.5: Generating Functions - *CC BY-NC-SA 3.0*

- 9: Graph Theory - *CC BY-NC-SA 3.0*
  - 9.1: Graphs - General Introduction - *CC BY-NC-SA 3.0*
  - 9.2: Data Structures for Graphs - *CC BY-NC-SA 3.0*
  - 9.3: Connectivity - *CC BY-NC-SA 3.0*
  - 9.4: Traversals- Eulerian and Hamiltonian Graphs - *CC BY-NC-SA 3.0*
  - 9.5: Graph Optimization - *CC BY-NC-SA 3.0*
  - 9.6: Planarity and Colorings - *CC BY-NC-SA 3.0*
- 10: Trees - *CC BY-NC-SA 3.0*
  - 10.1: What is a Tree? - *CC BY-NC-SA 3.0*
  - 10.2: Spanning Trees - *CC BY-NC-SA 3.0*
  - 10.3: Rooted Trees - *CC BY-NC-SA 3.0*
  - 10.4: Binary Trees - *CC BY-NC-SA 3.0*
- 11: Algebraic Structures - *CC BY-NC-SA 3.0*
  - 11.1: Operations - *CC BY-NC-SA 3.0*
  - 11.2: Algebraic Systems - *CC BY-NC-SA 3.0*
  - 11.3: Some General Properties of Groups - *CC BY-NC-SA 3.0*
  - 11.4: Greatest Common Divisors and the Integers Modulo n - *CC BY-NC-SA 3.0*
  - 11.5: Subsystems - *CC BY-NC-SA 3.0*
  - 11.6: Direct Products - *CC BY-NC-SA 3.0*
  - 11.7: Isomorphisms - *CC BY-NC-SA 3.0*
- 12: More Matrix Algebra - *CC BY-NC-SA 3.0*
  - 12.1: Systems of Linear Equations - *CC BY-NC-SA 3.0*
  - 12.2: Matrix Inversion - *CC BY-NC-SA 3.0*
  - 12.3: An Introduction to Vector Spaces - *CC BY-NC-SA 3.0*
  - 12.4: The Diagonalization Process - *CC BY-NC-SA 3.0*
  - 12.5: Some Applications - *CC BY-NC-SA 3.0*
  - 12.6: Linear Equations over the Integers Mod 2 - *CC BY-NC-SA 3.0*
- 13: Boolean Algebra - *CC BY-NC-SA 3.0*
  - 13.1: Posets Revisited - *CC BY-NC-SA 3.0*
  - 13.2: Lattices - *CC BY-NC-SA 3.0*
  - 13.3: Boolean Algebras - *CC BY-NC-SA 3.0*
  - 13.4: Atoms of a Boolean Algebra - *CC BY-NC-SA 3.0*
  - 13.5: Finite Boolean Algebras as n-tuples of 0's and 1's - *CC BY-NC-SA 3.0*
  - 13.6: Boolean Expressions - *CC BY-NC-SA 3.0*
  - 13.7: A Brief Introduction to Switching Theory and Logic Design - *CC BY-NC-SA 3.0*
- 14: Monoids and Automata - *CC BY-NC-SA 3.0*
  - 14.1: Monoids - *CC BY-NC-SA 3.0*
  - 14.2: Free Monoids and Languages - *CC BY-NC-SA 3.0*
  - 14.3: Automata, Finite-State Machines - *CC BY-NC-SA 3.0*
  - 14.4: The Monoid of a Finite-State Machine - *CC BY-NC-SA 3.0*
  - 14.5: The Machine of a Monoid - *CC BY-NC-SA 3.0*
- 15: Group Theory and Applications - *CC BY-NC-SA 3.0*
  - 15.1: Cyclic Groups - *CC BY-NC-SA 3.0*
  - 15.2: Cosets and Factor Groups - *CC BY-NC-SA 3.0*
  - 15.3: Permutation Groups - *CC BY-NC-SA 3.0*
  - 15.4: Normal Subgroups and Group Homomorphisms - *CC BY-NC-SA 3.0*
  - 15.5: Coding Theory, Group Codes - *CC BY-NC-SA 3.0*
- 16: An Introduction to Rings and Fields - *CC BY-NC-SA 3.0*
  - 16.1: Rings, Basic Definitions and Concepts - *CC BY-NC-SA 3.0*
  - 16.2: Fields - *CC BY-NC-SA 3.0*
  - 16.3: Polynomial Rings - *CC BY-NC-SA 3.0*
  - 16.4: Field Extensions - *CC BY-NC-SA 3.0*
  - 16.5: Power Series - *CC BY-NC-SA 3.0*
- 17: Appendix - *CC BY-NC-SA 3.0*
  - 17.1: A - Algorithms - *CC BY-NC-SA 3.0*
    - 17.1.1: An Introduction to Algorithms - *Undeclared*
    - 17.1.2: The Invariant Relation Theorem - *Undeclared*
  - 17.2: B - Python and SageMath - *CC BY-NC-SA 3.0*
    - 17.2.1: Python Iterators - *Undeclared*
    - 17.2.2: Dictionaries - *Undeclared*
  - 17.3: C - Determinants - *CC BY-NC-SA 3.0*
    - 17.3.1: Definition - *Undeclared*
    - 17.3.2: Computation - *Undeclared*
  - 17.4: D - Notation - *CC BY-NC-SA 3.0*
  - 17.5: E - References - *CC BY-NC-SA 3.0*
- Back Matter - *CC BY-NC-SA 3.0*
  - Index - *CC BY-NC-SA 3.0*
  - Glossary - *CC BY-NC-SA 3.0*
  - Detailed Licensing - *Undeclared*