

MAT 253

Discrete Structures

UNCG

Dan Yasaki

Key words and phrases. discrete structures

Contents

List of Figures	v
List of Tables	vii
List of Symbols	ix
Preface	xi
Chapter 1. Logic and Proofs	1
1.1. Propositional logic	1
1.2. Propositional equivalence	16
1.3. Predicates and quantifiers	22
1.4. Introduction to proofs	30
Chapter 2. Basic Structures	41
2.1. Sets	41
2.2. Set operations	47
2.3. Functions	56
2.4. Sequences and summations	70
Chapter 3. Number Theory and Applications	81
3.1. Divisibility and modular arithmetic	81
3.2. Integer representations and applications	89
3.3. Primes and greatest common divisors	98
3.4. Solving congruences	113
3.5. Cryptography	120
Chapter 4. Induction	129

4.1. Mathematical induction	129
4.2. Strong induction	136
Chapter 5. Counting	141
5.1. Basics of counting	141
5.2. Pigeonhole Principle	151
5.3. Permutations and combinations	155
Chapter 6. Relations	161
6.1. Relations and their properties	161
6.2. Equivalence relations	169
Appendix A. Programming assignments	177
Bibliography	195
Index	197

List of Figures

1.1.1.	xkcd: Communicating.	2
1.1.2.	xkcd: Formal Logic.	8
1.1.3.	xkcd: Correlation.	11
1.1.4.	xkcd: Protocol.	13
1.2.1.	xkcd: Honor Societies.	17
1.3.1.	xkcd: Existence Proof.	25
1.4.1.	xkcd: Principle of Explosion.	34
1.4.2.	xkcd: Proofs.	35
1.4.3.	xkcd: Python.	37
2.2.1.	Venn Diagrams for Union and Intersection.	48
2.2.2.	Venn Diagrams for Set Differences.	49
2.2.3.	Venn Diagram for Symmetric Difference.	49
2.2.4.	Venn Diagram for Complement.	50
2.3.1.	Function from A to B	56
2.3.2.	xkcd: Number Line.	60
2.3.3.	Inverse of a function from A to B	66
2.3.4.	Composition of two functions.	68
2.4.1.	xkcd: Tabletop Roleplaying.	74
3.2.1.	xkcd: 1 to 10.	90
3.3.1.	xkcd: Factoring the Time.	99
3.3.2.	Sieve of Eratosthenes.	101

3.3.3.	xkcd: Haiku Proof.	103
3.3.4.	xkcd: Goldbach Conjectures	104
3.5.1.	xkcd: Code Talkers.	120
3.5.2.	Rivest, Shamir, and Adleman.	122
3.5.3.	xkcd: Alice and Bob.	125
3.5.4.	xkcd: Security.	126
4.1.1.	xkcd: Win By Induction.	131
4.2.1.	xkcd: Set Theory.	139
5.1.1.	The positive rationals are countable.	144
6.1.1.	xkcd: Approximations.	162
6.1.2.	Composition of two relations.	167
6.2.1.	xkcd: Soda Sugar Comparisons.	170

List of Tables

1.1.1.	Truth Tables for Negation and Conjunction.	5
1.1.2.	Truth Tables for Disjunction and Exclusive Disjunction.	7
1.1.3.	Truth Tables for Conditional and Biconditional.	8
1.1.4.	Truth Tables for Converse, Contrapositive, and Inverse.	10
1.1.5.	Precedence Order for Logical Operators.	10
1.3.1.	Quantifiers.	25
3.3.1.	Number of primes up to bound.	102
3.5.1.	Caesar cipher lookup table.	121

List of Symbols

$\neg p$	Negation: Not p , (see page 4)
$p \wedge q$	Conjunction: p and q , (see page 4)
$p \vee q$	Disjunction: p or q , (see page 5)
$p \oplus q$	Exclusive or: p or q but not both, (see page 6)
$p \rightarrow q$	Conditional: if p then q , (see page 6)
$p \leftrightarrow q$	Biconditional: p if and only if q , (see page 7)
$\forall xP(x)$	Universal Quantification: for all x , $P(x)$, (see page 23)
\forall	Universal quantifier, (see page 23)
$\exists xP(x)$	Existential quantification: there exists an x such that $P(x)$, (see page 24)
\exists	Existential quantifier, (see page 24)
$S \equiv T$	S is equivalent to T , (see page 26)
$a \in A$	In: a is an element of A , (see page 41)
\mathbb{Z}	Integers, (see page 41)
\mathbb{Q}	Rational numbers, (see page 42)
\mathbb{R}	Real numbers, (see page 42)
\emptyset	Empty set, (see page 42)
$A \subseteq B$	Subset: A is a subset of B , (see page 42)
$A \subset B$	Proper subset: A is a proper subset of B , (see page 42)
$ A $	Cardinality: cardinality of A , (see page 44)
$\mathcal{P}(A)$	Power set: the power set of A , (see page 44)
$A \times B$	Cartesian product: product of A and B , (see page 45)

$A \cup B$	Union: A union B , (see page 47)
$A \cap B$	Intersection: A intersect B , (see page 48)
$A - B$	Difference: A minus B , (see page 48)
$A \setminus B$	Difference: A minus B , (see page 48)
\overline{A}	Complement: complement of A , (see page 50)
$f: A \rightarrow B$	Function: f from A to B , (see page 56)
$\text{im}(f)$	Image or Range: image of f , (see page 57)
$f(S)$	Image: image of S under f , (see page 57)
$\lfloor x \rfloor$	Floor: floor of x , (see page 59)
$\lceil x \rceil$	Ceiling: ceiling of x , (see page 60)
f^{-1}	Inverse: f inverse, (see page 66)
$g \circ f$	Composition: g of f , (see page 67)
$\{a_n\}$	Sequence, (see page 70)
$\sum_{i=1}^n a_i$	Summation, (see page 74)
$a \mid b$	Divides: a divides b , (see page 81)
$a \nmid b$	Does not divide: a does not divide b , (see page 81)
$a \text{ div } d$	Quotient: $a \text{ div } d$, (see page 83)
$a \text{ mod } d$	Remainder: $a \text{ mod } d$, (see page 83)
$a \equiv b \pmod{m}$	Congruent: a is congruent to b modulo m , (see page 83)
$[a]$	Congruence class: congruence class of a , (see page 84)
\mathbb{Z}_m	Integers mod m , (see page 85)
$(a_k a_{k-1} \dots a_1 a_0)_b$	Base b expansion, (see page 89)
$\text{gcd}(a, b)$	Greatest common divisor of a and b , (see page 103)
$\phi(n)$	Euler phi function, (see page 104)
$\text{lcm}(a, b)$	Least common multiple of a and b , (see page 104)
$P(n, r)$	Number of r -permutations of n elements, (see page 155)
$C(n, r)$	Number of r -combinations of n elements, (see page 157)
aRb	Related: a is related to b , (see page 163)
$[a]_R$	Equivalence class: equivalence class of a , (see page 172)
$[a]$	Equivalence class: equivalence class of a , (see page 172)

Preface

This document grew from lecture notes following the seventh edition of *Discrete Mathematics and its Applications* by Rosen [5]. I used various versions of the notes in conjunction with the book over the years whenever I taught MAT 253 *Discrete Structures*. I will continue to develop this document incorporating feedback from readers. This version was last modified: January 9, 2023. The most current version is available on my webpage.

<https://mathstats.uncg.edu/yasaki/teaching/>

MAT 253 core course in the mathematics curriculum designed for mathematics majors as an early introduction to discrete mathematical structures, rigorous proof techniques, and mathematical programming.

Catalogue description: A rigorous introduction to discrete mathematical structures, proof techniques, and programming. Topics include sets, functions, sequences, relations, induction, propositional and predicate logic, modular arithmetic, and mathematical programming.

Student learning outcomes: Upon successful completion of this course, students will be able to:

- define the fundamental discrete mathematical structures.
- identify and describe various types of relations.
- explain how RSA encryption allows for secure message transcription.
- translate pseudocode algorithms into Python scripts.
- compute the number of solutions to several arrangement problems.
- analyze simple algorithms and identify values of variables at various stages of completion.

- combine definitions and results produced in class to create rigorous proofs of basic statements about discrete mathematical structures.
- evaluate an argument for logical validity.

The choice of programming language is *Python 3.X*. You can use Python on the central Linux server of UNCG. It is also installed on the computers in all ITS computer labs. You can download it for free at

<http://www.python.org/download/>

In a typical semester, we cover most of the sections 1–5 of the *Python Tutorial* available at

<http://docs.python.org/py3k/tutorial/>

In writing this text, I wanted to produce an streamlined, yet inviting introduction to discrete structures. The text is interspersed with fun, yet (mostly) relevant xkcd comics. The prerequisites for the book are minimal; pre-calculus and an open mind. Foundations are built from definitions.



The Bourbaki *dangerous bend* symbol is used to highlight subtle ideas that may be missed on a first reading.

It is best to pair these notes with additional resources. Some free resources are identified below.

A major goal of this course is to teach you to communicate mathematics clearly. This involves learning precise statements of definitions, and learning to write clear, concise proofs. Here are two books that may help.

- *Book of Proof* by Richard Hammock: This book is an introduction to the standard methods of proving mathematical theorems.

<http://www.people.vcu.edu/~rhammack/BookOfProof/>

It has been approved by the American Institute of Mathematics' Open Textbook Initiative.

- *The Art of Proof* by Matthias Beck and Ross Geoghegan: This book is an excellent introduction to writing good proofs. This book is not open source, but our library has a Springer subscription that includes this book. Go to the UNCG Library Catalog to find this book. You can download a free copy by entering your UNCG iSpartan credentials.

Another component of the course is mathematical programming. The language chosen for this course is Python 3. There are several additional resources you may find helpful. Choose based on your programming background and desired difficulty level.

- Python 3 Tutorial: This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and

will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in The Python Standard Library. This will be the main source of information for the programming assignments in the Appendix.

<https://docs.python.org/3/tutorial/>

- Non-Programmer's Tutorial for Python 3: The Non-Programmers' Tutorial For Python 3 is a tutorial designed to be an introduction to the Python programming language. This guide is for someone with no programming experience.

https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- Python for Non-Programmers: If you've never programmed before, the tutorials on this page are recommended for you; they don't assume that you have previous experience.

<https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>

- Python for Programmers: The tutorials on this page are aimed at people who have previous experience with other programming languages (C, Perl, Lisp, Visual Basic, etc.).

<https://wiki.python.org/moin/BeginnersGuide/Programmers>

- The Python Wiki: This Wiki is a community place to gather and organize all things about Python. Feel free to exercise your editorial skills and expertise to make it a useful knowledge base and up-to-date reference on all Python-related topics.

<https://wiki.python.org/moin/FrontPage>

- Learn Python the Hard Way: This book instructs you in Python by slowly building and establishing skills through techniques like practice and memorization, then applying them to increasingly difficult problems. By the end of the book you will have the tools needed to begin learning more complex programming topics.

<https://learnpythonthehardway.org/book/>

I thank Office of the Provost and the University Libraries for the Open Educational Resources Mini-Grant in summer 2018 that allowed me the extra time to adjust the course syllabus to accommodate this text.

Thanks to Cliff Smyth and Sebastian Pauli for piloting the use of these notes in their courses. Thanks to others that found typos and mistakes, including H. Parlaman.

Please submit errata and suggestions for improvement:

<https://goo.gl/forms/1KyKylptFg3K6SX62>

Dan Yasaki
January 9, 2023

Logic and Proofs

Contrariwise, if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic.

Lewis Carroll (1832–1898)

In this chapter, we discuss logic and proofs. The rules of logic specify the meaning of mathematical statements. These rules help us understand and reason with these statements to construct arguments to justify the truth of certain statements. Once we prove a mathematical statement true, we call it a *theorem*. The argument of justification is called a *proof*.

Clear reasoning and communication of ideas is important in all disciplines. We restrict ourselves to a small, but important corner of mathematics where we can completely describe the theory with minimal prerequisites. The skills you gain are applicable to many other situations.

1.1. Propositional logic

<p>Goals. To introduce the basic terminology of propositional logic, including logical operators; to show how to construct truth tables.</p>

While we may wish to understand all statements, this is too daunting of a task. Human language is just too subtle and nuanced. Instead, we develop a theory for understanding certain types of simpler statements known as *propositions*. There is an overlap between these simple statements and statements that are used outside of this context. For the most part the meanings agree, but there are instances where the intended meanings

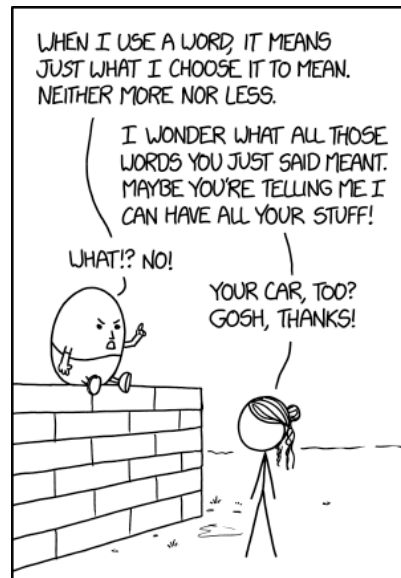


Figure 1.1.1. xkcd: Communicating. (<https://xkcd.com/1860/>)
 You're saying that the responsibility for avoiding miscommunication lies entirely with the listener, not the speaker, which explains why you haven't been able to convince anyone to help you down from that wall.

are different. Because of this, we will be precise about what we mean mathematically.

In this section, we develop a framework to study propositions. The rules of logic give precise meaning to propositions and help us understand the validity of arguments.

1.1.1. Propositions.

Definition 1.1.1. A *proposition* is a declarative statement (a statement that declares a fact) that is either true or false but not both. The *truth value* of a proposition is often denoted T for true and F for false.

Let's first look at some statements that are not propositions. These fall outside the scope of our study of propositional logic.

Example 1.1.2 (Question). Consider the statement "Where is the bookstore?" This is an interrogative, not declarative, statement. Questions are not propositions.

Example 1.1.3 (Command). Consider the statement "Tie your shoe." This is an imperative, not declarative, statement. Commands are not propositions.

Example 1.1.4 (Paradox). Consider the statement "This statement is false." This is a declarative statement, but it is not a proposition. This is more

subtle. Suppose the statement is true. The statement is asserting it is false. The only way for that to be true would be for the statement to be false. In that case, the assertion that it is false would be true. . . . This sort of self-referential statement that does not allow for a single truth value is not a proposition.

The last example is a bit more subtle. Here, the statement is not a proposition because there are variables in the statement that are not quantified. We will see how to deal with such things in detail in §1.3.

Example 1.1.5 (Unquantified). Consider the statement “ $x + 1 = 3$.” This is a declarative statement, but it is not a proposition. Why? The truthfulness of the sentence depends on the value of the variable x , so it does not have a well-defined truth value. For example, it is true when $x = 2$, but it is false when $x = 3$. This sort of ambiguous statement is not a proposition.

Now, let’s look at some statements that are propositions.

Example 1.1.6. Consider the statement “Washington DC is the capital of the United States.” This is a proposition since it is a declarative statement that is true.

Note that propositions are allowed to be false.

Example 1.1.7. The statement “Charlotte is the capital of NC,” is a proposition since it is a declarative statement that is that is false.

Example 1.1.8. Consider the statement “ $3 \cdot 5 = 8$.” This is a proposition since it is a declarative statement that is false.

We also allow statements that declare facts for which we may not know the truth value.

Example 1.1.9. Consider the statement “Bob is taller than Alice.” This is a proposition since it is a declarative statement that is either true or false, but not both. It does not bother us that we don’t know if it is true or false. The statement is a proposition because it declares a fact that is either true or false.

We will often use *propositional variables* to represent propositions. For example, let $p = “2 + 3 = 5,”$ and let $q = “My name is Dan.”$ In this case, the proposition p is true. We also have that the proposition q is true, provided we agree on the convention that first person pronouns refer to me, the author.

1.1.2. Compound propositions. Using logical operators, we will be able to use existing propositions to create *compound propositions*.

Definition 1.1.10. Let p be a proposition. The *negation* of p , denoted $\neg p$, is the proposition

$$\neg p = \text{“It is not the case that } p\text{.”}$$

It is read as “Not p ” and has the opposite truth value from p .

Table 1.1.1 gives the truth table for negation. If p is true, then $\neg p$ is false. If p is false, then $\neg p$ is true.

Example 1.1.11. Consider the proposition $p = “2 + 3 = 5.”$ The negation $\neg p$ is the proposition

$$\neg p = \text{“It is not the case that } 2 + 3 = 5\text{.”}$$

More directly, $\neg p$ can be expressed as “ $2 + 3 \neq 5$.” The original proposition is true, so the negation is false.

Example 1.1.12. Consider the proposition $q = “2^3 = 5.”$ The negation $\neg q$ is the proposition

$$\neg q = \text{“It is not the case that } 2^3 = 5\text{.”}$$

More directly, $\neg q$ can be expressed as “ $2^3 \neq 5$.” The original proposition is false, so the negation is true.

Definition 1.1.13. Let p and q be propositions. The *conjunction* of p and q , denoted $p \wedge q$, is the proposition “ p and q .” It is true when p and q are both true and false otherwise.

The conjunction is often referred to as *and*. Table 1.1.1 gives the truth table for conjunction. If p is true and q is true, then $p \wedge q$ is true. If p or q are false (including the case where both p and q are false), then the conjunction $p \wedge q$ is false.

Example 1.1.14. Let p be the proposition “ $1 + 2 = 3$,” and let q be the proposition “ $3 + 4 = 7$.” Since p and q are both true, the conjunction $p \wedge q$ is true. The conjunction can be expressed as

$$p \wedge q = \text{“} 1 + 2 = 3 \text{ and } 3 + 4 = 7\text{.”}$$

Example 1.1.15. Let p be the proposition “ $5 < 3$,” and let q be the proposition “ $4 = 7$.” Both p and q are false. The conjunction $p \wedge q$ is false. The conjunction can be expressed as

$$p \wedge q = \text{“} 5 < 3 \text{ and } 4 = 7\text{.”}$$

Example 1.1.16. Let p be the proposition “ $5 < 5$,” and let q be the proposition “ $10 = 5 \cdot 2$.” The conjunction $p \wedge q$ is false because p is false. The fact that q is true is not enough to make the conjunction true. The conjunction can be expressed as

$$p \wedge q = \text{“} 5 < 5 \text{ and } 10 = 5 \cdot 2\text{.”}$$

Table 1.1.1. Truth Tables for Negation and Conjunction.

p	$\neg p$
T	F
F	T

Negation
(not p)

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Conjunction
(p and q)

Example 1.1.17. Consider the propositions $a =$ “Alice is tall,” and $b =$ “Bob is tall”. The conjunction $a \wedge b$ of a and b is the proposition

$$a \wedge b = \text{“Alice is tall, and Bob is tall.”}$$

More directly, $a \wedge b$ can be expressed as “Alice and Bob are tall.” The conjunction $a \wedge b$ is true if Alice and Bob are both tall; it is false if either one is not tall.

Definition 1.1.18. Let p and q be propositions. The *disjunction* of p and q , denoted $p \vee q$, is the proposition “ p or q .” It is false when p and q are both false and true otherwise.

The disjunction is often referred to as *or*. Table 1.1.2 gives the truth table for conjunction. If at least one of p or q is true, then the disjunction $p \vee q$ is true. If p and q are both false, then $p \vee q$ is false.

A disjunction is true if at least one of the propositions is true.

Example 1.1.19. Consider the propositions $s =$ “Squares have four sides,” and $t =$ “Triangles have five sides.” The disjunction $s \vee t$ of s and t is the proposition

$$s \vee t = \text{“Squares have four sides, or triangles have five sides.”}$$

The disjunction is true because squares have four sides. It doesn’t matter that triangles don’t have five sides. Disjunctions are only false when both propositions being combined are false.

The disjunction of two true propositions is true.

Example 1.1.20. Let g be the proposition “I live in Greensboro,” and let j be the proposition “I was born in Japan.” Then g and j are both true. The disjunction $g \vee j$ is true and can be expressed as

$$g \vee j = \text{“I live in Greensboro, or I was born in Japan.”}$$

The only way the disjunction of two propositions is false is if both propositions are false.

Example 1.1.21. Let s be the proposition “UNCG’s mascot is Rudy the Rodeo Clown,” and let m be the the proposition “The UNCG motto is *Sleep*.” The disjunction can be expressed as

$$s \vee m = \text{“UNCG’s mascot is Rudy the Rodeo Clown and motto is } \textit{Sleep}.\text{”}$$

Since the mascot is Spiro the Spartan and the motto is *Service*, both s and m are false. Thus the disjunction $s \vee m$ is also false.

Definition 1.1.22. Let p and q be propositions. The *exclusive disjunction* of p and q , denoted $p \oplus q$, is the proposition “ p or q but not both.” It is true when exactly one of p or q is true and false otherwise.

The exclusive disjunction is often referred to as *exclusive or* or *xor*. Table 1.1.2 gives the truth table for conjunction.

The exclusive disjunction of one true proposition and one false proposition is true.

Example 1.1.23. Let o be the statement “An ostrich is a bird,” and let b be the statement “A beaver is a bird.” The exclusive disjunction is

$$o \oplus b = \text{“An ostrich is a bird, or a beaver is a bird but not both.”}$$

The exclusive disjunction is true, because an ostrich is a bird and a beaver is not a bird.

The exclusive disjunction of two false propositions is false.

Example 1.1.24. The exclusive disjunction of “Dogs are reptiles,” and “Snakes are mammals,” is “Dogs are reptiles or snakes are mammals, but not both.” The exclusive disjunction is false, because dogs are not reptiles, and snakes are not mammals. Exclusive disjunctions are true when exactly one of the propositions being combined is true.

The exclusive disjunction of two true propositions is false.

Example 1.1.25. The exclusive disjunction of “Dogs are mammals,” and “Snakes are reptiles,” is “Dogs are mammals or snakes are reptiles, but not both.” The exclusive disjunction is false, because dogs are mammals and snakes are reptiles. Exclusive disjunctions are true when exactly one of the propositions being combined is true.

Definition 1.1.26. Let p and q be propositions. The *conditional* of p and q , denoted $p \rightarrow q$, is the proposition “if p then q .” It is false when p is true and q is false and true otherwise. In the conditional $p \rightarrow q$, p is called the *hypothesis* and q is called the *conclusion*.

The conditional is often referred to as *if then*. Table 1.1.3 gives the truth table for the conditional.

Table 1.1.2. Truth Tables for Disjunction and Exclusive Disjunction.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Disjunction
(p or q)

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Exclusive Disjunction
(p xor q)

Example 1.1.27. The conditional of “ $5^2 = 10$ ” and “ $3 \cdot 4 = 12$ ” is “if $5^2 = 10$, then $3 \cdot 4 = 12$.” The conditional is true because the hypothesis is false. Similarly, the conditional “if $5^2 = 10$, then $3 \cdot 4 = 13$,” is also true.

Example 1.1.28. The conditional “if $3 \cdot 4 = 12$, then $5^2 = 10$,” is false because the hypothesis is true, but the conclusion is false.

Definition 1.1.29. Let p and q be propositions. The *biconditional* of p and q , denoted $p \leftrightarrow q$, is the proposition “ p if and only if q .” In other words, $p \leftrightarrow q$ is the proposition $(p \rightarrow q) \wedge (q \rightarrow p)$. It is true when p and q have the same truth values and false otherwise.

The biconditional is often referred to as *if and only if* or *iff*. Table 1.1.3 gives the truth table for the conditional.

The biconditional $p \leftrightarrow q$ is true when p and q are both true.

Example 1.1.30. The biconditional “ $2 + 4 = 6$ if and only if $2 \cdot 4 = 8$.” is true because both component propositions are true. Biconditionals are true exactly when the component propositions have the same truth value.

The biconditional $p \leftrightarrow q$ is also true when p and q are both false.

Example 1.1.31. The biconditional “ $2 + 4 = 7$ if and only if $2 \cdot 4 = 9$ ” is true because both component propositions are false. Biconditionals are true exactly when the component propositions have the same truth value.

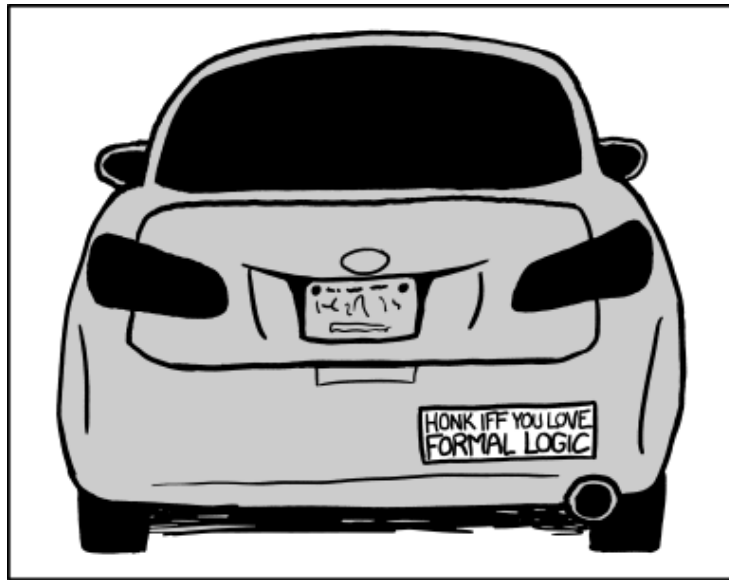
The biconditional $p \leftrightarrow q$ is false when p is true and q is false, or when p is false and q is true.

Example 1.1.32. The biconditional “ $1 = 2$ if and only if $3 = 3$ ” is false, because “ $1 = 2$ ” is false and “ $3 = 3$ ” is true.

Table 1.1.3. Truth Tables for Conditional and Biconditional.

p	q	$p \rightarrow q$	p	q	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	F	T	F	F	T

Conditional
(if p then q)
Biconditional
(p if and only if q)

**Figure 1.1.2.** xkcd: Formal Logic. (<https://xkcd.com/1033/>) Note that this implies you should NOT honk solely because I stopped for a pedestrian and you're behind me.

Example 1.1.33. Suppose p be the proposition “I’m in a store,” and let q be the proposition “I’m singing.” Then we have the following.

- $\neg p$ = “I’m not in a store.”
- $p \wedge q$ = “I’m in a store, and I’m singing.”
- $p \vee q$ = “I’m in a store, or I’m singing.”
- $p \oplus q$ = “I’m in a store, or I’m singing but not both.”
- $p \rightarrow q$ = “If I’m in a store, then I’m singing.”
- $p \leftrightarrow q$ = “I’m in a store if and only if I’m singing.”

Example 1.1.34. Let p be the proposition “Swimming is allowed,” and let q be the proposition “Sharks have been spotted.”

- “Swimming is not allowed,” can be written symbolically as $\neg p$.
- “Sharks have been spotted, but swimming is allowed,” can be written as $q \wedge p$. Note that “but” has been converted logically to “and” in this context. We will see more trickiness of the English language in §1.1.6.
- “If sharks have not been spotted, then swimming is allowed,” can be written as $\neg q \rightarrow p$.

1.1.3. Truth tables. The *truth table* for a compound proposition gives the possible truth values of a compound proposition in terms of the truth values of the original propositions.

Remark 1.1.35. In general, if a compound proposition involves k propositional variables, the truth table will have 2^k rows.

The truth tables of the standard compound propositions are given in Tables 1.1.1, 1.1.2, and 1.1.3.

Example 1.1.36. Let’s construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

There are two propositional variables, so the table will have $2^2 = 4$ rows.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

1.1.4. Converse, contrapositive, and inverse.

Definition 1.1.37. Let p and q be propositions, and consider the conditional $p \rightarrow q$. The *converse* of $p \rightarrow q$ is $q \rightarrow p$. The *contrapositive* of $p \rightarrow q$ is $\neg q \rightarrow \neg p$. The *inverse* of $p \rightarrow q$ is $\neg p \rightarrow \neg q$.

The truth tables for the converse, contrapositive, and inverse of a conditional statement are given in Table 1.1.4.

Example 1.1.38. Consider the conditional “If I studied, then I passed the course.”

- The converse is the conditional “If I passed the course, then I studied.”
- The contrapositive is the conditional “If I did not pass the course, then I did not study.”
- The inverse is the conditional “If I did not study, then I did not pass the course.”

Table 1.1.4. Truth Tables for Converse, Contrapositive, and Inverse.

				Conditional	Converse	Contrapositive	Inverse
p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$q \rightarrow p$	$\neg q \rightarrow \neg p$	$\neg p \rightarrow \neg q$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	T
F	T	T	F	T	F	T	F
F	F	T	T	T	T	T	T

Table 1.1.5. Precedence Order for Logical Operators.

Operator	Precedence
\neg	1
\wedge	2
\vee	3
\rightarrow	4
\leftrightarrow	5

Remark 1.1.39. Table 1.1.4 shows that a conditional and its contrapositive have the same truth values. It also shows that the inverse of a conditional has the same truth values as the converse.

1.1.5. Logical operators. Negation can be viewed as an operator that takes a proposition and returns a proposition. Similarly, conjunction, disjunction, conditional, and biconditional can be viewed as binary operators that take two propositions and returns a proposition. Like the usual arithmetic operators we are used to, there is a prescribed order of precedence for the operators. It is given in Table 1.1.5.

For example, the precedence order means that $\neg p \wedge q$ means the same thing as $(\neg p) \wedge q$. Omitting the parentheses in this case is standard and acceptable. We will allow this.

The precedence also means that $p \wedge q \vee r$ means $(p \wedge q) \vee r$. The omission of parentheses in this case is confusing. We will not allow this in this course.

Another example that will not be allowed is the following. The precedence gives that $p \wedge q \rightarrow r$ means $(p \wedge q) \rightarrow r$. In this case also, parentheses should be added for clarity.

1.1.6. English is hard. The English language is tricky and subtle. There are many ways to say the same thing, and there are many preconceptions that people have that must be let go when studying logic. In this section, we give lots of examples and highlight some of the common misconceptions.

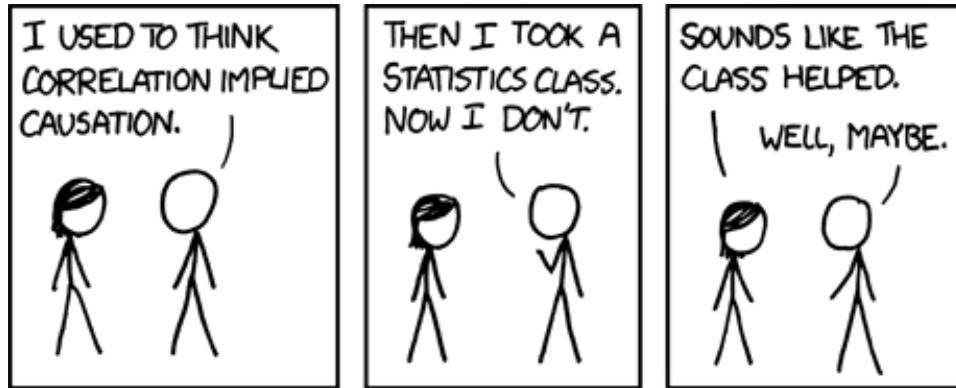


Figure 1.1.3. xkcd: Correlation. (<https://xkcd.com/552/>) Correlation doesn't imply causation, but it does waggle its eyebrows suggestively and gesture furtively while mouthing 'look over there'.

Example 1.1.40. Let p be the proposition "It is raining," and let q be the proposition "I am happy." Then the following are some of the correct ways to express $p \wedge q$.

- It is raining, and I am happy.
- It is raining, but I am happy.
- It is raining, yet I am happy.
- Although it is raining, I am happy.

For propositions p and q , the following are correct ways to express $p \rightarrow q$.

- | | |
|--|---|
| • If p , then q . | • q whenever p . |
| • If p , q . | • q if p . |
| • p implies q . | • q follows from p . |
| • p only if q . | • q unless $\neg p$. |
| • p is sufficient for q . | • A sufficient condition for q is p . |
| • A necessary condition for p is q . | • q is necessary for p . |

Note that " p only if q " says that p cannot be true when q is not true. Similarly, " q unless $\neg p$ " says that if $\neg p$ is false, then q must be true. Looking at the truth tables, we see that they are restatements of if p , then q .

⚠ Conditionals are independent of cause and effect. The hypothesis and conclusion may not be related. There is no implied cause and effect or any other type of relationship between the statements in a true conditional proposition.

Example 1.1.41. “If the sun comes up in the morning, then $2 + 2 = 4$.” This conditional is true since the hypothesis and conclusion are true. The conditional does not say anything about causation.

Let’s look at some English conditionals. We will rewrite them in the “if p then q ” form to make them easier to analyze. Recall the truth table for $p \rightarrow q$.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Example 1.1.42. Consider the proposition “You will pass this course only if you put in the effort.” Recall the conditional $p \rightarrow q$ can be phrased as “ p only if q .” We can rewrite this as “If you pass this course, then you must have put in the effort.” Note: It is possible that you fail this course, even if you put in the effort. This conditional says that if you pass this course, then we can deduce that you put in the effort.

Example 1.1.43. Consider the proposition “You will pass this course unless you don’t put in the effort.” Recall the conditional $p \rightarrow q$ can be phrased “ q unless not p .” Thus we can rewrite the given proposition as “If you put in the effort, then you will pass this course.” Note: It is possible to pass the course even if you do not put in the effort, but if you put in the effort, we can deduce that you will pass this course.

Example 1.1.44. Consider the proposition “It is necessary to wash the boss’s car to get promoted.” Recall that the conditional $p \rightarrow q$ can be phrased as “ q is necessary for p ”. Namely, in order for p to be true, q must be true. The given proposition is that washing the car is necessary for the promotion. We can rephrase that as “If you get promoted, then you must have washed the boss’s car.” A useful way to think of these things is in terms of an obligation or contract. Note: This proposition makes no claims about what will happen if you wash the boss’s car. Specifically, if you wash the boss’s car, you should not expect to be promoted because of that.

Example 1.1.45. Express the following sentence symbolically: “You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.”

Let r , f , and s be the propositions

- r = “You can ride the roller coaster,”
 f = “You are under 4 feet tall,”
 s = “You are older than 16 years old.”



Figure 1.1.4. xkcd: Protocol. (<https://xkcd.com/1323/>) Changing the names would be easier, but if you're not comfortable lying, try only making friends with people named Alice, Bob, Carol, etc.

Then we the sentence can be expressed symbolically as

$$\neg s \rightarrow (f \rightarrow \neg r).$$

Can you convince yourself that the statement can also be expressed as

$$(f \wedge \neg s) \rightarrow \neg r?$$

We will see how to reconcile this in §1.2.

Example 1.1.46. An island has two tribes, truth-tellers and liars. You encounter two people, Alice and Bob, on the island. Alice says “Bob is a truth-teller,” and Bob says, “We are from different tribes.” Let’s figure out who is from which tribe.

Let a and b be the propositions

a = “Alice is a truth-teller,”

b = “Bob is a truth-teller.”

Then Alice’s statement is b . Bob’s statement is that a and b have opposite truth values, so his statement is $a \leftrightarrow \neg b$. Note that Alice’s statement must have the same truth value as a because her statement is true if and only if she is a truth-teller. That means $a \leftrightarrow b$ must be true. Analogously, Bob’s statement has the same truth value as b , so $(a \leftrightarrow \neg b) \leftrightarrow b$ must be true.

Let's work out the truth table. We are looking for a row where $a \leftrightarrow b$ and $(a \leftrightarrow \neg b) \leftrightarrow b$ are both true.

a	b	$\neg b$	$a \leftrightarrow \neg b$	$a \leftrightarrow b$	$(a \leftrightarrow \neg b) \leftrightarrow b$
T	T	F	F	T	F
T	F	T	T	F	F
F	T	F	T	F	T
F	F	T	F	T	T

From the last two columns, we see that $b \leftrightarrow a$ and $(a \leftrightarrow \neg b) \leftrightarrow b$ are both true in the row where a and b are false. In other words, we have that Alice and Bob are both liars.

Exercises

- Give the definition for these terms. Be sure to set up any notation that is required.
 - proposition
 - negation of a proposition
 - conjunction of two propositions
 - disjunction of two propositions
 - exclusive disjunction of two propositions
 - conditional of two propositions
 - biconditional of two propositions
 - converse of a conditional
 - contrapositive of a conditional
 - inverse of a conditional
- Complete the following truth table.

p	q	$p \vee q$	$p \oplus q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T					
T	F					
F	T					
F	F					

- Identify each of the following as a proposition or not. For each proposition, give the truth value.
 - Greensboro is the capital of North Carolina.
 - Let's meet at the dining hall at 7pm.
 - Squares have three sides.
 - $2 \cdot 5 = 11$.
 - $a^2 + b^2 = c^2$.
- Identify each of the following as a proposition or not. For each proposition, give the truth value.
 - $1 + 1 = 2$

- (b) Buy a bag of chips.
 - (c) If $2 + 3 = 7$, then $11 - 4 = 7$.
 - (d) If $11 - 4 = 7$, then $2 + 3 = 7$.
 - (e) 3 is odd or 253 is odd.
 - (f) If the sun came up today, then squares have four sides.
5. Identify each of the following as a proposition or not. For each proposition, give the truth value.
- (a) Cats are reptiles.
 - (b) Dogs are mammals.
 - (c) If cats are reptiles, then dogs are mammals.
 - (d) If dogs are mammals, then cats are reptiles.
 - (e) Cats are reptiles if and only if dogs are mammals.
6. What is the negation of each of these propositions?
- (a) There are 60 minutes in an hour.
 - (b) 169 is a perfect square.
 - (c) Alice is less than 5 feet tall.
 - (d) Bob has more than 20 books.
 - (e) $5 \cdot 6 = 30$
7. Let r and w be the following propositions.

$r =$ "It is raining."

$w =$ "The ground is wet."

Express each of the propositions as an English sentence.

- (a) $\neg r$
 - (b) $r \wedge w$
 - (c) $r \rightarrow w$
 - (d) $\neg r \rightarrow \neg w$
 - (e) $\neg r \wedge \neg w$
8. Let p and q be the following propositions.

$p =$ "There are 8 pints in a gallon."

$q =$ "There are 4 quarts in a gallon."

Write each of these propositions using p and q and logical operators (including negations).

- (a) There are 8 pints in a gallon, or there are 4 quarts in a gallon.
- (b) If there are 8 pints in a gallon, then there are not 4 quarts in a gallon.
- (c) There are not 8 pints in a gallon, or there are 8 pints in a gallon and 4 quarts in a gallon.
- (d) There are 8 pints in a gallon if and only if there are 4 quarts in a gallon.
- (e) There are not 8 pints in a gallon, but there are 4 quarts in a gallon.

9. Let p , q , and r be the propositions

p = “You have the flu.”

q = “You miss the final exam.”

r = “You pass the course.”

Express each proposition as an English sentence.

- (a) $p \rightarrow (q \vee r)$
 - (b) $(\neg q \wedge r) \vee r$
 - (c) $p \rightarrow (q \wedge r)$
 - (d) $(\neg p \vee \neg q) \rightarrow r$
 - (e) $(p \wedge q) \vee (\neg q \wedge r)$
10. Write each of these statements in the form “if p then q .”
- (a) It is necessary to study every day to pass MAT 253.
 - (b) Alice gets caught whenever she cheats.
 - (c) To pass MAT 253, it is sufficient to attend class every day.
 - (d) The warranty is good only if you bought the computer less than 1 year ago.
 - (e) Bob will get a good job unless he does not learn discrete mathematics.
11. State the converse, contrapositive, and inverse of each of these conditional statements.
- (a) If it is snowing tonight, I will stay home.
 - (b) The home team wins whenever it rains.
 - (c) Alice eats a desert only if she eats her vegetables.
 - (d) When Bob stays up late, it is necessary for him to sleep until noon.
 - (e) A positive integer is prime only if it has no positive divisors other than 1 and itself.
12. Construct a truth table for each of these compound propositions.
- (a) $(p \wedge q) \rightarrow r$
 - (b) $(p \vee q) \rightarrow \neg r$
 - (c) $p \rightarrow (q \wedge r)$
 - (d) $\neg q \rightarrow (r \vee \neg p)$
 - (e) $p \wedge (q \vee \neg r)$

1.2. Propositional equivalence

Goals. To show how propositional equivalences are established and to introduce the most important such equivalences.

1.2.1. Terminology. Some compound propositions are true for a silly reason. Basically, the compound proposition is true, regardless of the truth values of the component propositions. For example, “I’ll get to it when I get

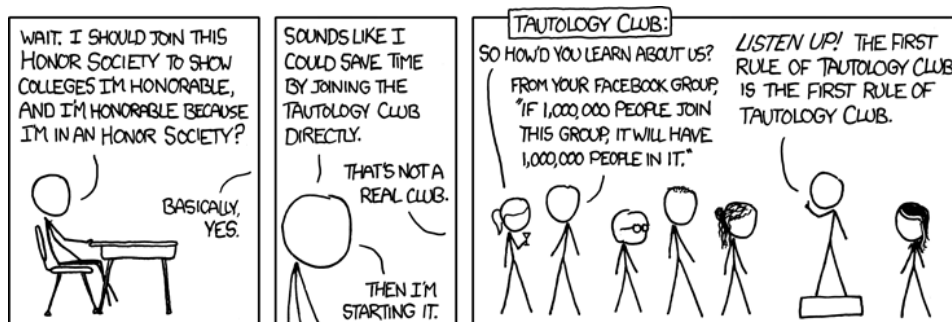


Figure 1.2.1. xkcd: Honor Societies. (<https://xkcd.com/703/>) Hey, why do YOU get to be the president of Tautology Clu— wait, I can guess.

to it.” If g is the proposition “I get to it,”, then this compound proposition is the conditional $g \rightarrow g$. Whether g is true or false, the conditional is true. This is an example of a tautology.

Definition 1.2.1. A *tautology* is a compound proposition that is true no matter what the truth values of the propositional variables that occur in it.

Example 1.2.2. Let p be a proposition. Then $p \rightarrow p$ is a tautology. To see this, we compute the truth table for $p \rightarrow p$.

p	$p \rightarrow p$
T	T
F	T

Since $p \rightarrow p$ is always true, $p \rightarrow p$ is a tautology.

Definition 1.2.3. A *contradiction* is a compound proposition that is always false no matter what the truth values of the propositional variables that occur in it.

Example 1.2.4. Let p be a proposition. Then $\neg(p \rightarrow p)$ is a contradiction. To see this, we compute the truth table for $\neg(p \rightarrow p)$.

p	$p \rightarrow p$	$\neg(p \rightarrow p)$
T	T	F
F	T	F

Since $\neg(p \rightarrow p)$ is always false, $\neg(p \rightarrow p)$ is a contradiction.

Remark 1.2.5. In general, if P is a tautology, then $\neg P$ is a contradiction.

Definition 1.2.6. A *contingency* is a compound proposition that is neither a tautology nor a contradiction.

Example 1.2.7. Let p and q be propositions. The conditional $p \rightarrow q$ is a contingency. To see this, compute the truth table for $p \rightarrow q$.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Since $p \rightarrow q$ is true for some choices of p and q and false for some other choices, $p \rightarrow q$ is a contingency.

Example 1.2.8. Let p be a proposition. Then $p \wedge \neg p$ is a contradiction and $p \vee \neg p$ is a tautology as shown in the table below. Note that the column for $p \wedge \neg p$ is all false, and the column for $p \vee \neg p$ is all true.

p	$\neg p$	$p \wedge \neg p$	$p \vee \neg p$
T	F	F	T
F	T	F	T

Definition 1.2.9. Two propositions p and q are *logically equivalent*, denoted $p \equiv q$, if $p \leftrightarrow q$ is a tautology.

Example 1.2.10. By looking at the truth table in Definition 1.1.37, we see that a conditional is logically equivalent to its contrapositive. Similarly, the converse of a conditional is logically equivalent to its inverse.

1.2.2. Important propositional equivalences. The following theorem can be thought of as telling us how to distribute a “not” across an “and” or an “or”.

Theorem 1.2.11 (De Morgan’s laws for propositions). *Let p and q be propositions.*

$$(1) \neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$(2) \neg(p \vee q) \equiv \neg p \wedge \neg q$$

Proof. We prove $\neg(p \vee q) \equiv \neg p \wedge \neg q$ and leave the other as an exercise. First compute the truth table.

p	q	$\neg p$	$\neg q$	$p \vee q$	$\neg(p \vee q)$	$\neg p \wedge \neg q$
T	T	F	F	T	F	F
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	F	T	T

Since the last two columns are the same, we have $\neg(p \vee q) \equiv \neg p \wedge \neg q$ as desired. \square

Example 1.2.12. Let's negate the statement "Jake is wearing khakis and sounds hideous." Let

j = "Jake is wearing khakis,"

h = "Jake sounds hideous."

Then the original statement is $j \wedge h$. By De Morgan's law, the negation is

$$\neg(j \wedge h) = \neg j \vee \neg h.$$

This is, "Jake is not wearing khakis, or he does not sound hideous."

There are several other important propositional equivalencies that we should know.

Theorem 1.2.13 (Proposition Identities I). *Let p be a proposition.*

Identity laws: $p \wedge T \equiv p$; $p \vee F \equiv p$.

Domination laws: $p \vee T \equiv T$; $p \wedge F \equiv F$.

Idempotent laws: $p \vee p \equiv p$; $p \wedge p \equiv p$.

Double negation law: $\neg(\neg p) \equiv p$

Negation law: $p \vee \neg p \equiv T$; $p \wedge \neg p \equiv F$.

Proof. Exercise. Just compute the truth tables and verify the corresponding columns are the same. \square

Theorem 1.2.14 (Proposition Identities II). *Let p and q be propositions.*

Commutative laws: $p \vee q \equiv q \vee p$; $p \wedge q \equiv q \wedge p$.

Absorption laws: $p \vee (p \wedge q) \equiv p$; $p \wedge (p \vee q) \equiv p$.

Contrapositive law: $p \rightarrow q \equiv \neg q \rightarrow \neg p$

Proof. Exercise. Just compute the truth tables and verify the corresponding columns are the same. \square

Theorem 1.2.15 (Proposition Identities III). *Let p , q , and r be propositions.*

Associative laws:

$$p \vee (q \vee r) \equiv (p \vee q) \vee r; \quad p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r.$$

Distributive laws:

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r); \quad p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r).$$

Proof. Exercise. Just compute the truth tables and verify the corresponding columns are the same. \square

We can also use logical equivalences to turn conditionals into disjunctions.

Theorem 1.2.16. *Let p and q be propositions. Then*

$$p \rightarrow q \equiv \neg p \vee q.$$

Proof. We compute the relevant truth table.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Since the last two columns are equal, we have $p \rightarrow q \equiv \neg p \vee q$. □

Once we have proven all of these important logical equivalencies, we can use them to prove additional equivalencies and simplify compound propositions without resorting to truth tables.

Example 1.2.17. Recall Example 1.1.45, where we translated “You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.”

Let

r = “You can ride the roller coaster,”

f = “You are under 4 feet tall,”

s = “You are older than 16 years old.”

Then we had two seemingly correct translations

$$\neg s \rightarrow (f \rightarrow \neg r)$$

and

$$(f \wedge \neg s) \rightarrow \neg r.$$

Let’s see that these statements are logically equivalent by simplifying each one. We change each conditional to a disjunction and simplify.

$$\begin{aligned} \neg s \rightarrow (f \rightarrow \neg r) &\equiv s \vee (f \rightarrow \neg r) \\ &\equiv s \vee (\neg f \vee \neg r) \end{aligned}$$

$$\begin{aligned} (f \wedge \neg s) \rightarrow \neg r &\equiv \neg(f \wedge \neg s) \vee \neg r \\ &\equiv (\neg f \vee s) \vee \neg r \\ &\equiv (s \vee \neg f) \vee \neg r \\ &\equiv s \vee (\neg f \vee \neg r) \end{aligned}$$

Since both statements are equivalent to $s \vee (\neg f \vee \neg r)$, the statements are equivalent to each other.

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) tautology
 - (b) contradiction
 - (c) contingency
 - (d) logically equivalent propositions
2. State precisely De Morgan's laws for propositions. Be sure to set up any notation that is required.
3. Use De Morgan's laws to find the negation of these statements.
 - (a) Alice will go to graduate school or get a job in industry.
 - (b) Bob majored in math and computer science.
 - (c) Carl is tall and thin.
 - (d) Dan has a laptop and a desktop.
 - (e) Eve or Frank will pick you up at the airport.
4. Show that each of these conditional statements is a tautology by using truth tables.
 - (a) $p \rightarrow (p \vee q)$
 - (b) $(p \wedge q) \rightarrow p$
 - (c) $(p \wedge q) \rightarrow (p \rightarrow q)$
 - (d) $(\neg p \wedge (p \vee q)) \rightarrow q$
 - (e) $\neg p \rightarrow (p \rightarrow q)$
5. Complete the following truth table. Is $p \rightarrow q$ is logically equivalent to $\neg p \vee q$? Justify. Be sure to say what portion of the computation explains your response.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T			
T	F			
F	T			
F	F			

6. Verify each associative law using a truth table. Which columns show the logical equivalence?
 - (a) $(p \vee q) \vee r \equiv p \vee (q \vee r)$
 - (b) $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
7. Verify each absorption law using a truth table. Which columns show the logical equivalence?
 - (a) $p \vee (p \wedge q) \equiv p$

- (b) $p \wedge (p \vee q) \equiv p$
8. Show that $\neg(p \oplus q)$ and $p \leftrightarrow q$ are logically equivalent.
 9. Show that $(p \rightarrow q) \vee (p \rightarrow r)$ and $p \rightarrow (q \vee r)$ are logically equivalent.
 10. Show that $(p \wedge q) \rightarrow r$ and $(p \rightarrow r) \wedge (q \wedge r)$ are not logically equivalent.
 11. Show that $p \leftrightarrow q$ and $\neg p \leftrightarrow \neg q$ are logically equivalent.
 12. Prove the proposition identities in Theorem 1.2.13.
 13. Prove the proposition identities in Theorem 1.2.14.
 14. Prove the proposition identities in Theorem 1.2.15.
-

1.3. Predicates and quantifiers

Goals. To introduce predicate logic, especially existential and universal quantification. Moreover, to explain how to translate between English sentences (or mathematical statements) and logical expressions.

1.3.1. Predicates. Statements involving variables, such as “ x is greater than 3” are often found in mathematical assertions. Such statements are not propositions as they are neither true nor false when the value of the variable is not specified. The statement has two parts. The first part, the variable x , is the *subject*. The second part, the *predicate*, is the property that the subject can have. In this case, the predicate is “is greater than 3”. We denote such statements as $P(x)$, and view it as a *propositional function* P evaluated at x ; i.e., P is a function which outputs a proposition for each input.

Example 1.3.1. Let $P(x)$ be the statement “ $x > 3$.” Then $P(2)$ is the proposition “ $2 > 3$,” which is false. The proposition $P(4)$ is the statement “ $4 > 3$,” which is true.

Example 1.3.2. Propositional functions can have many variables, such as

$$P(x, y, z) = \text{“}x, y, \text{ and } z \text{ live in the same dorm.”}$$

Then $P(\text{Alice}, \text{Bob}, \text{Carl})$ is the proposition “Alice, Bob, and Carl live in the same dorm.”

1.3.2. Universal quantification. *Quantification* expresses the extent to which a propositional function is true; e.g., all, some, none, many, etc.

Definition 1.3.3. The *universal quantification* of P , denoted $\forall xP(x)$, is the proposition

“ $P(x)$ for all x in the domain.”

The symbol \forall is the *universal quantifier*. The *domain* of P specifies the possible values for x .

Definition 1.3.4. A *counterexample* of $\forall xP(x)$ is an element x_0 such that $P(x_0)$ is false.

Remark 1.3.5. The universal quantification of a propositional function is a proposition, since it is a declarative sentence that is either true or false, but not both.

Proof Technique 1.3.6 (To show $\forall xP(x)$ is false). Suppose P is a propositional function, and we want to prove the universal quantification $\forall xP(x)$ is false.

- (1) Find a counterexample. Specifically, find x_0 in the domain of P such that $P(x_0)$ is false.
- (2) Conclude $\forall xP(x)$ is false.

Remark 1.3.7. To prove a universal quantification is false, it suffices to exhibit a single counterexample.

Example 1.3.8. Let $P(x) = “x > 0”$ with domain \mathbb{R} . Then $\forall xP(x)$ is false. To show this, it is enough to provide a single counterexample. Consider the real number $x_0 = -2$. Note that $-2 \not> 0$, so $P(-2)$ is false. Thus $\forall xP(x)$ is false.

Example 1.3.9. Consider the statement “All dogs have brown fur.” This universally quantified statement is false. My dog Duey provides a counterexample.

Let’s examine this in more detail. Let $B(x) = “x$ has brown fur,” with domain the set of all dogs. The universally quantified statement “All dogs have brown fur,” can be written as $\forall xB(x)$. This is false, because we can produce a counterexample, namely my dog Duey. He is a dog, so he is in the domain of B . He does not have brown fur, so $B(\text{Duey})$ is false. Thus $\forall xB(x)$ is false. In other words, not all dogs have brown fur.

Proof Technique 1.3.10 (To show $\forall xP(x)$ is true). Suppose P is a propositional function, and we want to prove the universal quantification $\forall xP(x)$ is true.

- (1) Fix a generic element x in the domain of P .
- (2) Show $P(x)$ is true for this fixed generic element.
- (3) Conclude $\forall xP(x)$ is true.

Example 1.3.11. Let $P(x) = "x^2 \geq 0"$ with domain \mathbb{R} . Then $\forall xP(x)$ is true. To see this, fix a generic real number x . Then x^2 is non-negative since the square of any real number is non-negative. Thus $\forall xP(x)$ is true.

1.3.3. Existential quantification.

Definition 1.3.12. The *existential quantification* of P , denoted $\exists xP(x)$ is the proposition

“There exists an element x in the domain such that $P(x)$.”

The symbol \exists is the *existential quantifier*.

Definition 1.3.13. A *witness* of $\exists xP(x)$ is an element x_0 in the domain of P such that $P(x_0)$.

Remark 1.3.14. To prove an existential quantification is true, it suffices to exhibit a single witness.

Proof Technique 1.3.15 (To show $\exists xP(x)$ is true). Suppose P is a propositional function, and we want to prove the existential quantification $\exists xP(x)$ is true.

- (1) Find a witness. Specifically, find an element x_0 in the domain of P such that $P(x_0)$ is true.
- (2) Conclude $\exists xP(x)$ is true.

To show an existential quantification false, we need to show that a witness does not exist. Specifically, to show $\exists xP(x)$ is false, we need to show that $P(x)$ is false for every x in the domain of P .

Proof Technique 1.3.16 (To show $\exists xP(x)$ is false). Suppose P is a propositional function, and we want to prove the existential quantification $\exists xP(x)$ is false.

- (1) Fix a generic element x in the domain of P .
- (2) Show $P(x)$ is false.
- (3) Conclude $\exists xP(x)$ is false.

Table 1.3.1. Quantifiers.

Statement	When true?	When false?
$\forall xP(x)$	$P(x)$ is true for every x .	There is an x_0 (counterexample) for which $P(x_0)$ is false.
$\exists xP(x)$	There is an x_0 (witness) for which $P(x_0)$ is true.	$P(x)$ is false for every x .

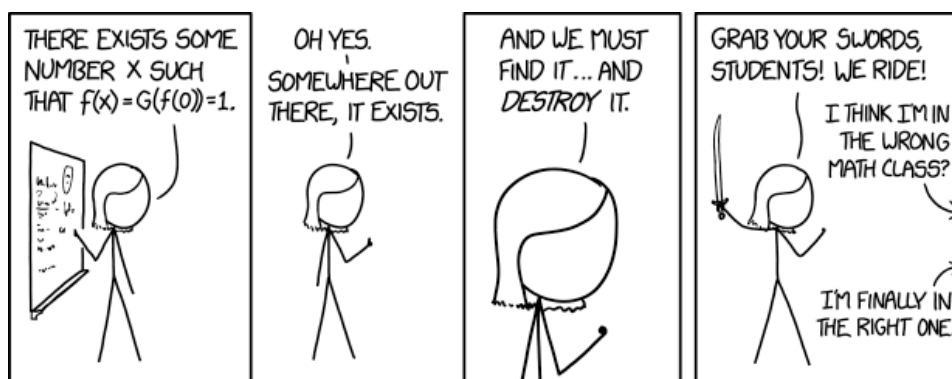


Figure 1.3.1. xkcd: Existence Proof. (<https://xkcd.com/1856/>) Real analysis is way realer than I expected.

Remark 1.3.17. To prove the existential quantification of P is true, it is enough to exhibit a single witness. To prove the existential quantification of P is false, we need to show that the universal quantification of $\neg P$ is true.

Example 1.3.18. Let $N(x) =$ “ x was born in NC,” with domain the MAT 253 students.

- (1) $\forall xN(x)$
- (2) $\exists xN(x)$

What are the truth values of the two propositions above? Do you need to discuss this with your classmates to determine the truth values?

- (1) $\forall xN(x)$ says “Every student in MAT 253 was born in NC.” To prove this statement false, it is enough to produce a counterexample. We need to just find one student in MAT 253 that was not born in NC. To prove this statement true, we need to check that every MAT 253 student was born in NC.
- (2) $\exists xN(x)$ says “There is a student in MAT 253 that was born in NC.” To prove this statement true, it is enough to produce a witness. We need to find just find one student in MAT 253 that was not born in NC.

Now suppose instead we take the domain of N to be all people in the US, but we want to have the same propositions about our MAT 253 class. Then we need to introduce a new propositional function $S(x) =$ “ x is a MAT 253 student.”

- (1) We can write “Every student in MAT 253 was born in NC,” as $\forall x(S(x) \rightarrow N(x))$.
- (2) We can write “There is a student in MAT 253 that was born in NC,” as $\exists x(S(x) \wedge N(x))$

Note that we have shown that $p \rightarrow q$ is equivalent to $\neg p \vee q$. That means that $\forall x(\neg S(x) \vee N(x))$ is a valid (though convoluted) way to write that every student in MAT 253 was born in NC.

1.3.4. Logical equivalence.

Definition 1.3.19. Two statements S and T involving predicates and quantifiers are *logically equivalent*, denoted $S \equiv T$, if they have the same truth value no matter which predicates are substituted into these statements and which domain is used.

Theorem 1.3.20 (De Morgan’s laws for quantifiers). *Let P be a propositional function. Then the following logical equivalencies hold.*

- (1) $\neg \forall x P(x) \equiv \exists x \neg P(x)$.
- (2) $\neg \exists x P(x) \equiv \forall x \neg P(x)$.

Proof. Let’s prove the first statement and leave the proof of the second as an exercise.

We want to show

$$\neg \forall x P(x) \equiv \exists x \neg P(x).$$

We just need to show that the left side and the right side have the same truth values, independent of what P or x actually is. The left side is the negation of $\forall x P(x)$. Thus the left side is true if $\forall x P(x)$ is false. Thus $\forall x P(x)$ must have a counterexample x_0 . Then $P(x_0)$ is false, which means $\neg P(x_0)$ is true. Thus x_0 provides the example showing $\exists x \neg P(x)$ is true. Therefore the right side is true as well.

Similarly, now suppose the left side is false. Then $\forall x P(x)$ is true. It follows that $\neg P(x)$ is never true, so the existential statement of the right side is also false. \square

Example 1.3.21. Consider the statement “No one is perfect.” Translate it into a logical expression using predicates and quantifiers. Then use De Morgan’s laws to rewrite it.

The statement is that there does not exist a person that is perfect. Let $P(x)$ = “ x is perfect,” with domain the set of all people. Then the statement is $\neg\exists xP(x)$. By De Morgan’s laws

$$\neg\exists xP(x) \equiv \forall x\neg P(x),$$

which says that everyone is imperfect.

Example 1.3.22 (Lewis Carroll).

- (1) All lions are fierce.
- (2) Some lions do not drink coffee.
- (3) Some fierce creatures do not drink coffee.

Let’s write the three statements into logical expression using predicates and quantifiers.

Let L , F , and C be propositional functions

$$\begin{aligned} L(x) &= \text{“}x \text{ is a lion,“} \\ F(x) &= \text{“}x \text{ is fierce,“} \\ C(x) &= \text{“}x \text{ drinks coffee,“} \end{aligned}$$

with domain all creatures. Then the sentences can be expressed follows.

- (1) $\forall x(L(x) \rightarrow F(x))$
- (2) $\exists x(L(x) \wedge \neg C(x))$
- (3) $\exists x(F(x) \wedge \neg C(x))$

Suppose we know that the first two statements are true. Does this allow us to deduce the third statement? We will see more examples in the following section, but let’s examine this particular example more closely first.

Suppose the first two statements are true. The second statement guarantees the existence of a creature, we’ll call him Bob, such that $L(\text{Bob})$ and $\neg C(\text{Bob})$ are both true. Since $L(\text{Bob})$ is true, the conditional in the first statement tells us that $F(\text{Bob})$ is true. Since $F(\text{Bob})$ and $\neg C(\text{Bob})$ are both true, Bob provides the example to show the third statement is true.

In other words, since some lions do not drink coffee, there must be a lion that does not drink coffee. Let’s call him Bob. Then since all lions are fierce, we have that Bob is fierce. Since Bob is fierce and does not drink coffee, we know that some fierce creatures do not drink coffee.

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) universal quantification of a propositional function

-
- (b) existential quantification of a proposition function
(c) logically equivalent statements involving predicates and quantifiers
2. State precisely De Morgan's laws for quantifiers. Be sure to set up any notation that is required.
3. Let $P(x)$ denote the propositional function " $x \leq 10$," with domain \mathbb{R} . What are these truth values?
- (a) $P(0)$
(b) $P(-2)$
(c) $P(253)$
(d) $\forall xP(x)$
(e) $\exists xP(x)$
4. Let $S(x)$ be the statement " x studies more than 30 hours per week," where the domain for x consists of all students in MAT 253. Express each of these quantifications in English.
- (a) $\exists xS(x)$
(b) $\exists x\neg S(x)$
(c) $\forall xS(x)$
(d) $\forall x\neg S(x)$
5. Let $B(x)$ be the statement " x has a bird," let $C(x)$ be the statement " x has a cat," and let $D(x)$ be the statement " x has a dog," where the domain for x is all MAT 253 students. Express each of these statements in terms of $B(x)$, $C(x)$, $D(x)$, quantifiers, and logical operators.
- (a) There is a student in MAT 253 that has a bird, a cat, and a dog.
(b) Every student in MAT 253 has a dog.
(c) No student in MAT 253 has a bird.
(d) Every student in MAT 253 that has a dog also has a bird.
(e) Some student in MAT 253 has a bird, a cat, or a dog.
6. Determine the truth value of each of these statements if the domain of all variables consists of all integers.
- (a) $\forall n(n^2 \geq 0)$
(b) $\forall n(n^2 > 0)$
(c) $\exists n(n^2 > 0)$
(d) $\forall n(n > 0 \vee n < 0)$
(e) $\exists n(n > 0 \wedge n < 0)$
7. Translate each of these statements into logical expressions using predicates, quantifiers, and logical operators.
- (a) Something is not right.
(b) Everything is fine.
(c) Every bird can fly.
(d) Some old dogs can learn new tricks.
(e) No one is perfect.

-
8. Prove each of these universally quantified statements false by providing a counterexample, where the domain for all the variables consists of all real numbers.
- $\forall x(x^2 > 0)$
 - $\forall x(x^2 \geq x)$
 - $\forall x((x + 3)^2 = x^2 + 3^2)$
 - $\forall x(\frac{x}{2} \text{ is rational})$
 - $\forall x(\sqrt{x^2} = x)$
9. Prove each of these existential statements true by providing a witness, where the domain for all the variables consists of all real numbers.
- $\exists x(3x \text{ is irrational})$
 - $\exists x(x^2 \leq 0)$
 - $\exists x(1 < x < 2)$
 - $\exists x((x + 3)^2 = x^2 + 3^2)$
 - $\exists x(\pi x^2 \text{ is an integer})$
10. Let $D(x) = "x \text{ is an odd duck}"$, and let $M(x) = "x \text{ is a mathematician}"$, with domain consisting of all people. Express the statement "Every mathematician is an odd duck" in terms of $D(x)$, $M(x)$, quantifiers, and logical operators.
11. Let $Q(x)$ be the statement " $2x < 0.001$ " with domain the set of all real numbers. What is the truth value of $\exists xQ(x)$? Explain.
12. Express each of these statements using quantifiers. Use De Morgan's laws for quantifiers to find the negation of each of these statements. Form the negation of the statement so that no negation is to the left of the quantifier. Finally, express the negation in simple English. (Do not simply use the phrase "It is not the case that ...")
- All dogs are nice.
 - Some students do not study every day.
 - All birds can fly.
 - Some dogs have fleas.
 - Some integers are rational numbers.
 - All real numbers are integers.
13. Consider the propositional functions $M(x) = "x \text{ is a mathematician}"$, $S(x) = "x \text{ is silly}"$, and $C(x) = "x \text{ drinks coffee}"$, with domain the set of all people.
- Express each of the following sentences in terms of $M(x)$, $S(x)$, $C(x)$, quantifiers, and logical operators.
- No mathematician is silly.
 - All mathematicians drink coffee.
 - Mathematicians that do not drink coffee are silly.
 - Some people that drink coffee are not silly.
 - Not all silly people are mathematicians.

14. Alice was overheard saying, “I will go out with Bob when pigs fly.” Rewrite this in the form “if p , then q .” Assuming that Alice speaks the truth, use propositional logic to explain what, if any, implications this has for Alice and Bob.
15. Suppose we have the following rules for Alice.
- Taylor’s rule:** Alice eats her veggies, or she can’t have dessert.
- Leslie’s rule:** If Alice eats her veggies, then she can have dessert.
- Alex’s rule:** Alice can have her dessert, when she eats her veggies.
- Cameron’s rule:** Alice can have her dessert, only if she eats her veggies.
- Alice’s rule:** Alice can have her dessert.
- If Alice’s parents agree on a rule, what are their names? (They do not have the same name. There may be more than one correct answer.)

1.4. Introduction to proofs

Goals. To introduce the notion of proof and basic methods of proof, including direct proof, proof by contraposition, and proof by contradiction. Furthermore, to learn how to distinguish between correct and incorrect arguments, and to understand and construct basic types of proofs.

1.4.1. Direct proof. Recall the conditional $p \rightarrow q$ has the following truth table.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

It follows that if we want to prove that $p \rightarrow q$ is true, we need to show that the second row ($p = \text{T}, q = \text{F}$) does not occur. One approach, called the *direct proof* is constructed when we start by assuming p is true. Then we show that q is true.

Proof Technique 1.4.1 (To show $p \rightarrow q$ with direct proof). Suppose p and q are propositions, and we want to prove the conditional “if p then q ” by direct proof.

- (1) Assume p is true.
- (2) Deduce q is true under the assumption.
- (3) Conclude $p \rightarrow q$.

Let's recall some definitions of certain types of integers so that we have some objects to play with.

Definition 1.4.2. An integer n is *even* if there exists an integer k such that $n = 2k$.

Proof Technique 1.4.3 (Show n is even). Suppose n is an integer, and we want to prove n is even.

- (1) Find k such that $n = 2k$.
- (2) Show k is an integer.
- (3) Conclude n is even, by definition.

Example 1.4.4. The integer 12 is even because $12 = 2 \cdot 6$, and 6 is an integer.

Example 1.4.5. The integer 0 is even because $0 = 2 \cdot 0$, and 0 is an integer.

Definition 1.4.6. An integer n is *odd* if there exists an integer k such that $n = 2k + 1$.

Proof Technique 1.4.7 (Show n is odd). Suppose n is an integer, and we want to prove n is odd.

- (1) Find k such that $n = 2k + 1$.
- (2) Show k is an integer.
- (3) Conclude n is odd, by definition.

Example 1.4.8. The integer 23 is odd because $23 = 2 \cdot 11 + 1$, and 11 is an integer.

Example 1.4.9. The integer -27 is odd because $-27 = 2 \cdot (-14) + 1$, and -14 is an integer.

Remark 1.4.10. Every integer is even or odd. No integer is both.

Example 1.4.11. Following the steps below, we prove that the sum of two odd integers is even.

- (1) First, we rewrite what we want to show in the form “if p , then q .” Additionally, we will give friendly names to objects.¹ Let's name our odd integers a and b . Then the statement we wish to prove is:

“If a and b are odd integers, then $a + b$ is even.”

¹Fear of a name only increases fear of the thing itself. –Dumbledore

- (2) When we want to prove a statement of the form “if p , then q ” directly, we assume p is true and try to show q . This is commonly where we set some notation as well. **Let a and b be odd integers.**
- (3) Next we need to recall what an *odd* integer is. An integer n is *odd* if there exists an integer k such that $n = 2k + 1$.
- (4) Now apply the definition to our situation. **Since a is odd, there exists an integer k such that $a = 2k + 1$.**
- (5) **Since b is odd, there exists an integer ℓ such that $b = 2\ell + 1$.**
- (6) Check back above to be sure that the two integers whose existence is guaranteed have different names. They need different names because they need not be the same integer.
- (7) Now look back to the goal that we set in 1. Since we want to say something about $a + b$, it makes sense to **Compute $a + b$ and simplify.**

$$\begin{aligned} a + b &= 2k + 1 + 2\ell + 1 \\ &= 2k + 2\ell + 2 \\ &= 2(k + \ell + 1). \end{aligned}$$

- (8) The line above should prove $a + b$ is even, provided the bit in the parentheses is an integer. Make some remark noting that, and we are done. **Since k and ℓ are integers, we have $k + \ell + 1$ is an integer, and so $a + b$ is even.**

Putting that all together yields the following.

Proof. Let a and b be odd integers. Since a is odd, there exists an integer k such that $a = 2k + 1$. Since b is odd, there exists an integer ℓ such that $b = 2\ell + 1$. Compute $a + b$ and simplify.

$$\begin{aligned} a + b &= 2k + 1 + 2\ell + 1 \\ &= 2k + 2\ell + 2 \\ &= 2(k + \ell + 1). \end{aligned}$$

Since k and ℓ are integers, we have $k + \ell + 1$ is an integer, and so $a + b$ is even. \square

Example 1.4.12. Prove that the product of two even integers is even.

Proof. Let m and n be even integers. Then there exist integers k and ℓ such that $m = 2k$ and $n = 2\ell$. Then

$$mn = 2k \cdot 2\ell = 4k\ell = 2(2k\ell).$$

Since k and ℓ are integers, $2k\ell$ is an integer. Thus $m \cdot n$ is even. \square

Example 1.4.13. Let m and n be real numbers. Prove that $m^2 = n^2$ if and only if $m = n$ or $m = -n$.

Proof. Suppose m and n are integers such that $m^2 = n^2$. Then

$$m^2 - n^2 = (m + n)(m - n) = 0.$$

It follows that $m + n = 0$ or $m - n = 0$. Thus $m = -n$ or $m = n$. \square

Example 1.4.14. Let $P(n)$ be the proposition “ $1 + 2 + \cdots + n = \frac{n(n+1)}{2}$.” Prove $P(1)$ and $P(2)$ are true.

Proof. $P(1)$ is the statement “ $1 = \frac{1(1+1)}{2}$.” This is true since the right side is

$$\frac{1(1+1)}{2} = \frac{2}{2} = 1,$$

which is equal to the left side.

Similarly, $P(2)$ is the statement “ $1 + 2 = \frac{2(2+1)}{2}$.” This is true since the left side is

$$1 + 2 = 3,$$

and the right side is

$$\frac{2(2+1)}{2} = \frac{6}{2} = 3.$$

\square

1.4.2. Proof by contraposition. Recall that the contrapositive of $p \rightarrow q$ is logically equivalent to $p \rightarrow q$. That means proving $p \rightarrow q$ is true is the same as proving $\neg q \rightarrow \neg p$ is true. This is called **proof by contraposition**.

Proof Technique 1.4.15 (To show $p \rightarrow q$ by contraposition). Suppose p and q are propositions, and we want to prove the conditional “if p then q ” by contraposition.

- (1) State that it is enough to prove the contrapositive.
- (2) Assume q is false, (or equivalently $\neg q$ is true).
- (3) Deduce p is false, (or equivalently $\neg p$ is true) under the assumption.
- (4) Conclude $p \rightarrow q$.

Example 1.4.16. Let n be an integer. Prove that if n^2 is even, then n is even.

Proof. This is equivalent to proving that if n is odd, then n^2 is odd. Suppose n is odd. Then there exists an integer k such that $n = 2k + 1$. Then

$$n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1.$$

Since k is an integer, we have $2k^2 + 2k$ is an integer, and so n^2 is odd. \square

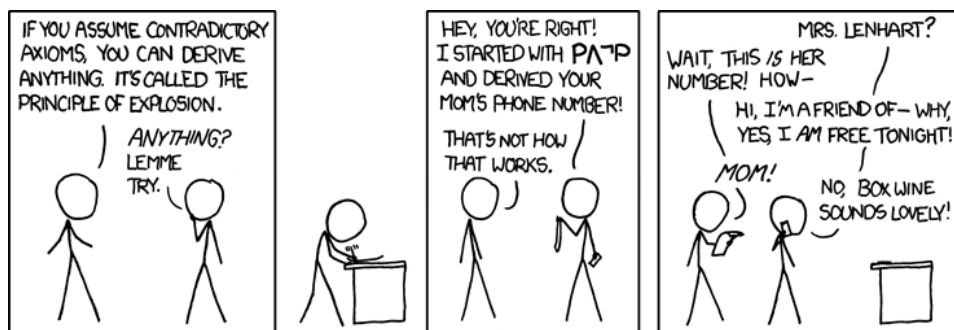


Figure 1.4.1. xkcd: Principle of Explosion. (<https://xkcd.com/704/>)
 You want me to pick up waffle cones? Oh, right, for the wine. One sec, let me just derive your son's credit card number and I'll be on my way.

1.4.3. Proof by contradiction. Let s and r be propositions. Recall that $r \wedge \neg r$ is a contradiction. In particular, $r \wedge \neg r$ is always false. If we can prove that $\neg s \rightarrow (r \wedge \neg r)$ is true, then $\neg s$ must be false. In this case, s must be true. Unwinding that, we get a technique known as *proof by contradiction*. Namely, we can prove s is true by producing a direct proof of $\neg s \rightarrow (r \wedge \neg r)$; i.e., to prove s is true, start by assuming $\neg s$ is true and try to reach a contradiction. This is called *proof by contradiction*.

Proof Technique 1.4.17 (To prove s is true by contradiction). Suppose s is a proposition, and we want to prove s by contradiction.

- (1) State that we will use proof by contradiction.
- (2) Assume s is false, (or equivalently $\neg s$ is true).
- (3) Deduce an auxiliary proposition r and its negation $\neg r$.
- (4) Announce the contradiction.
- (5) Conclude s is true.

Definition 1.4.18. A real number r is *rational* if there exists integers p and q with $q \neq 0$ such that $r = \frac{p}{q}$. A real number r is *irrational* if it is not rational. The set of rational numbers is denoted \mathbb{Q} .

Example 1.4.19. Prove that $\sqrt{2}$ is irrational.

Proof. We proceed by contradiction. Suppose $\sqrt{2}$ is rational. Then there exists integers p and q , with $q \neq 0$ such that $\sqrt{2} = \frac{p}{q}$. Note that we can arrange that the fraction is in lowest terms by canceling common factors, so without loss of generality assume that p and q have no common factors.

Then squaring both sides of $\sqrt{2} = \frac{p}{q}$ we get $2 = \frac{p^2}{q^2}$. Then $2q^2 = p^2$, so p^2 is even. By Example 1.4.16, this implies p is even. Then there exists an

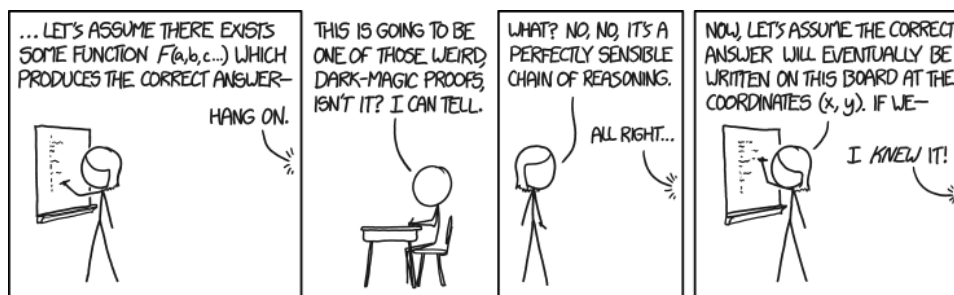


Figure 1.4.2. xkcd: Proofs. (<https://xkcd.com/1724/>) Next, let's assume the decision of whether to take the Axiom of Choice is made by a deterministic process . . .

integer k such that $p = 2k$. Then $2q^2 = (2k)^2 = 4k^2$. Dividing by 2, we get $q^2 = 2k^2$ so q^2 is even. As before, this implies q is even. \square

Example 1.4.20. Use a proof by contradiction to show that there is no rational number r for which $r^3 + r + 1 = 0$.

Proof. Suppose not. Suppose there is a rational number $r = a/b$ such that $r^3 + r + 1 = 0$, where we write a/b is lowest terms so that a and b have no common factors. Then

$$\left(\frac{a}{b}\right)^3 + \frac{a}{b} + 1 = 0.$$

Multiplying both sides by b^3 , we get

$$a^3 + ab^2 + b^3 = 0.$$

There are four cases to consider, depending on the parity (even/odd) of a and b .

a and b both even: Contradiction! We have a/b in lowest terms, but they have a common factor of 2.

a and b both odd: Then a^3 , ab^2 , and b^3 are all odd. The sum of three odd numbers is odd. Contradiction! 0 is even.

a is even b is odd: Then a^3 and ab^2 are even, and b^3 is odd. The sum of two even integers and an odd integer is odd. Contradiction! 0 is even.

a is odd b is even: Then a^3 is odd, and ab^2 and b^3 are even. The sum of an odd integer and two even integers is odd. Contradiction! 0 is even.

Thus there is not rational number r such that $r^3 + r + 1 = 0$. \square

1.4.4. Existence proofs. Many theorems assert the existence of an object with certain properties. They are of the form $\exists xP(x)$. One way to prove such a statement is to find a *witness* a such that $P(a)$ is true. This is called a *constructive existence proof*. See Proof Technique 1.3.15 for details.

Example 1.4.21. Prove there is an integer that can be written as the sum of cubes of positive integers in two different ways.

Proof. It suffices to provide a witness. We can find the following witness by Python code snippet 1.4.22, for example.

$$1729 = 10^3 + 9^3 = 12^3 + 1^3.$$

□

Python Code Snippet 1.4.22. Here is some Python code to find a witness for Example 1.4.21. We look in a box for a positive integer that can be expressed as the sum of cubes of positive integers in two different ways.

```
# empty dictionary to store possible witnesses
sum_dict = {}
# pick a bound for the box to search in
box_bound = 13
for i in range(box_bound):
    for j in range(i+1, box_bound):
        s = i**3 + j**3
        if s in sum_dict.keys():
            sum_dict[s].append([i, j])
        else:
            sum_dict[s] = [[i, j]]
for s, w in sum_dict.items():
    if len(w) > 1:
        print(s, w)
```

There are some proofs of existence that do not produce a witness. These are called *nonconstructive proofs*.

Example 1.4.23. Prove there exist irrational numbers a and b such that a^b is rational.

Proof. We have already seen that $\sqrt{2}$ is irrational in Example 1.4.19. If $\sqrt{2}^{\sqrt{2}}$ is rational, then $a = \sqrt{2}, b = \sqrt{2}$ is our witness, and we are done. Otherwise, $\sqrt{2}^{\sqrt{2}}$ is irrational. In this case,

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2,$$

which is rational, so $a = \sqrt{2}^{\sqrt{2}}, b = \sqrt{2}$ is our witness. □

Note that this proof does not tell us which case occurs, so it is nonconstructive.

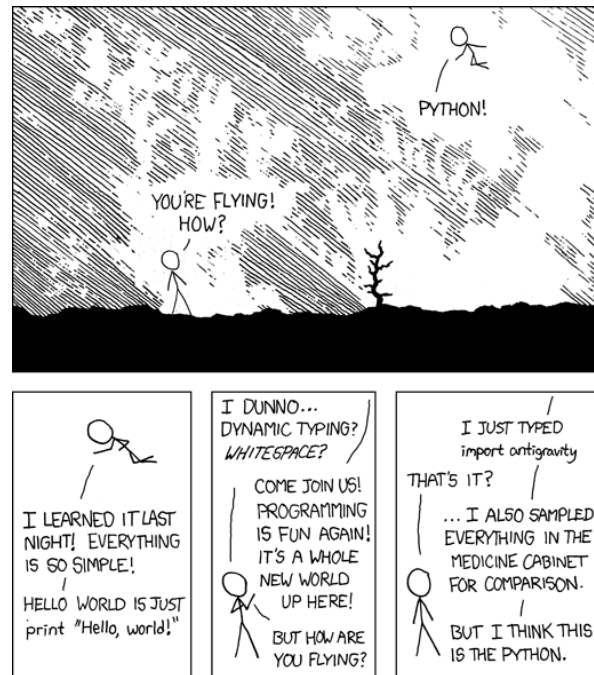


Figure 1.4.3. xkcd: Python. (<https://xkcd.com/353/>) I wrote 20 short programs in Python yesterday. It was wonderful. Perl, I'm leaving you.

1.4.5. Summary guidelines for proofs.

- (1) Use complete sentences.
- (2) Each sentence should set notation or be a true statement.
- (3) Each true statement should be a conclusion that can be drawn from the previous statements using a definition, computation, or result proved in class.
- (4) Do not assert the truth of statement to be proven at the beginning of a proof. Preface such statements with “We wish to prove” or something similar.
- (5) Oftentimes, a good first step is just unwinding the definitions.
- (6) To prove “if p , then q ” directly, start the proof by assuming p is true. Then deduce that q must be true. See Proof Technique 1.4.1.
- (7) To prove “if p , then q ” by contraposition, start the proof by assuming q is false. Then deduce that p must be false. See Proof Technique 1.4.15.
- (8) To prove p by contradiction, start the proof by assuming p is false. Then deduce a contradiction. See Proof Technique 1.4.17.

Here are some examples of what is meant by (2) above.

- ax
This is not a sentence.
- $ax = b$ has a solution.
This is a sentence, but it is not true or false. We need to know more about a and b .
- Let $a \in \mathbb{R}$, $a \neq 0$. Then $ax = b$ has a solution.
This is a bit better. The first sentence sets notation, but the second sentence is still neither true nor false since we have not specified the universe for b .
- Let $a \in \mathbb{R}$, $a \neq 0$. Then $ax = b$ has a solution for every $b \in \mathbb{R}$.
The first sentence sets notation. All of the notation is defined. The second sentence is true.

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) even integer
 - (b) odd integer
 - (c) rational number
2. Use a direct proof to show that the sum of two odd integers is even.
3. Use a direct proof to show that the sum of two even integers is even.
4. Use a direct proof to show that the product of two odd integers is odd.
5. Use a proof by contradiction to prove that the sum of an irrational number and a rational number is irrational.
6. Use a direct proof to show the product of two rational numbers is a rational number.
7. Prove each of these statements is false by providing a counterexample.
 - (a) The product of two irrational numbers is irrational.
 - (b) The sum of two irrational numbers is irrational.
 - (c) Every positive integer can be written as the sum of the squares of three integers.
 - (d) If n is an integer, then $n^2 > n$.
 - (e) If a and b are rational number, then a^b is also rational.
8. Let $P(n)$ be the proposition " $n^2 \geq n$." Prove $P(1)$ is true.
9. Use a proof by contradiction to prove that if n is an integer and $n^3 + 5$ is odd, then n is even.
10. Prove there is a right triangle with all three sides having rational lengths.
11. Prove there exists a pair of consecutive integers such that one of these integers is a perfect square and the other is a perfect cube.

12. Go to

<http://link.springer.com/book/10.1007%2F978-1-4419-7023-7>

while on campus and download *The Art of Proof* by Matthias Beck and Ross Geoghegan. Read carefully Chapters 1–7.

Basic Structures

In this chapter, we look at some fundamental structures in mathematics: sets, functions, and sequences.

2.1. Sets

Goals. To introduce the basic terminology of set theory.

2.1.1. Basic terminology.

Definition 2.1.1. A *set* is an unordered collection of objects, called *elements* or *members* of the set. We write $a \in A$ to denote that a is an element of the set A .

There are several ways to describe a set.

roster method: List the elements explicitly. e.g., $\{1, 2, 7\}$, $\{1, 2, \dots, 10\}$.

set builder notation: Characterize the elements of the set by stating properties they must have to be members. e.g.,

$$\{x \mid x \text{ is an odd positive integer less than } 10\}.$$

Example 2.1.2. Is $\{3\}$ the same thing as 3?

No. Sets are like bags. A bag with an apple in it is different from an apple. The set $\{3\}$ is a set with one element 3 in it. The number 3 is not the same as the set containing just 3.

There are some sets of numbers that we will often use.

- The set of *integers*, denoted \mathbb{Z} , is

$$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}.$$

- The set of *rational numbers*, denoted \mathbb{Q} , is

$$\mathbb{Q} = \left\{ \frac{p}{q} \mid p, q \in \mathbb{Z}, q \neq 0 \right\}.$$

- The set of *real numbers*, denoted \mathbb{R} , is actually fairly subtle to define. (Cauchy sequences of rational numbers is one way that you may see in MAT 395.) For our purposes, your intuitive idea of the real numbers is sufficient.

We also have special notation for intervals of real numbers.

$$[a, b] = \{x \mid x \in \mathbb{R}, a \leq x \leq b\}$$

$$(a, b] = \{x \mid x \in \mathbb{R}, a < x \leq b\}$$

$$[a, b) = \{x \mid x \in \mathbb{R}, a \leq x < b\}$$

$$(a, b) = \{x \mid x \in \mathbb{R}, a < x < b\}$$

Definition 2.1.3. Two sets A and B are *equal*, denoted $A = B$, if they have the same elements. i.e., $\forall x(x \in A \leftrightarrow x \in B)$.

Example 2.1.4. Let $A = \{1, 2, 2, 5\}$, and let $B = \{1, 2, 5\}$. Then $A = B$. Why? For every element x , the truth value of $x \in A$ is the same as the truth value of $x \in B$. Thus $A = B$ by definition.

To show two sets are equal, we use subsets as described later in Proof Technique 2.1.14. It is more straightforward to show two sets are not equal. Since set equality is defined as the universal quantification (“for all”) of a biconditional (“if and only if”), to show two sets are not equal, it is enough to find any counterexample to the biconditional. In particular, we just need to produce any element that they do not share.

Proof Technique 2.1.5 (To show $A \neq B$). Suppose A and B are sets, and we want to prove A is not equal to B .

- (1) Find a particular element a in A that is not a member of B ; or find a particular element b in B that is not a member of A .
- (2) Conclude $A \neq B$.

Definition 2.1.6. The *empty set*, denoted \emptyset , is the set with no elements.

Example 2.1.7. The empty set is $\{\}$.

2.1.2. Subsets.

Definition 2.1.8. A set A is a *subset* of a set B , denoted $A \subseteq B$, if every element of A is also an element of B . i.e., $\forall x(x \in A \rightarrow x \in B)$. If $A \subseteq B$, and $A \neq B$, we say A is a *proper subset* of B , denoted $A \subset B$.

Example 2.1.9. Let $A = \{a, b, c\}$, and let $B = \{a, b, c, d, e, f\}$. Then A is a subset of B . Why? For every x , whenever x is in A we have x is in B . Thus $A \subseteq B$, by definition.

Furthermore, we have A is a proper subset of B , since we additionally have that A is not equal to B . Thus $A \subset B$, by definition.

Remark 2.1.10. This notation is reminiscent of the notation for inequality and strict inequality.

Note that subsets are defined in terms of a universal quantification (“for all”). To prove universal statements, we use generic elements.

Proof Technique 2.1.11 (To show $A \subseteq B$). Suppose A and B are sets, and we want to prove A is a subset of B .

- (1) Let x be a generic element of A .
- (2) Show that x is in B .
- (3) Conclude $A \subseteq B$.

Theorem 2.1.12. For any set A ,

- (1) $\emptyset \subseteq A$, and
- (2) $A \subseteq A$.

Proof. We apply Proof Technique 2.1.11 to show $\emptyset \subseteq A$. Since \emptyset has no elements, there is nothing to check. In particular, we need to check that for each a in \emptyset , the statement $a \in A$ is true. Since there are no elements in \emptyset , there is nothing to check.

Next, we apply Proof Technique 2.1.11 to show $A \subseteq A$. Let x be a generic element in A . Then x is in A by construction. Thus $A \subseteq A$. \square

It seems like we just ran around in circles in the proof above. Read it over again, paying attention to what it is we need to show.

For a set A to be a proper subset of set B , we need A to be a subset of B that is not equal to B .

Proof Technique 2.1.13 (To show $A \subset B$). Suppose A and B are sets, and we want to prove A is a proper subset of B .

- (1) Use Proof Technique 2.1.5 to show $A \neq B$.
- (2) Use Proof Technique 2.1.11 to show $A \subseteq B$.
- (3) Conclude $A \subset B$.

By definition, two sets are equal if they have the same elements. This implies that each one is a subset of the other. We can turn this into a proof technique for showing two sets are equal.

Proof Technique 2.1.14 (To show $A = B$). Suppose A and B are sets, and we want to show A is equal to B .

- (1) Use Proof Technique 2.1.11 to show $A \subseteq B$.
- (2) Use Proof Technique 2.1.11 to show $B \subseteq A$.
- (3) Conclude $A = B$.

2.1.3. Size of a set.

Definition 2.1.15. Let A be a set. If there are exactly n distinct elements in A , where n is a non-negative integer, we say A is a *finite set* and that n is the *cardinality* of A , denoted $|A|$.

We discuss sets that are not finite and their cardinality in §5.1. The notion of infinity is subtle; there are different sizes of infinity.

Example 2.1.16. The empty set \emptyset has size 0

$$|\emptyset| = 0$$

since $\{\}$ has no elements.

Example 2.1.17. Let E be the even integers between -5 and 5 , not including -5 and 5 . Then we can compute the size of E by writing E using the roster method and counting.

$$E = \{-4, -2, 0, 2, 4\},$$

so $|E| = 5$.

Example 2.1.18. Let C be the set of consonants in the English alphabet. Instead of listing the elements of C in roster method, we note that there are 26 letters in the English alphabet. Of these, 5 are vowels and the rest are consonants. Thus

$$|C| = 26 - 5 = 21.$$

2.1.4. Power sets.

Definition 2.1.19. Let A be a set. The *power set* of A , denoted $\mathcal{P}(A)$, is the set of all subsets of A .

Example 2.1.20. Suppose $A = \{2, x, \pi\}$. The power set of A is the set of all subsets of A . Let's work out the subsets systematically, ordered by size.

Size 0: There is only 1 subset of size 0. It is the empty set: \emptyset .

Size 1: These singleton sets each contain one element of A . Since there are 3 elements in A , there are 3 subsets of size 1: $\{2\}$, $\{x\}$, and $\{\pi\}$.

Size 2: Subsets of A of size 2 can be constructed by omitting one element. Since there are 3 elements in A , there are 3 subsets of size 2: $\{2, x\}$, $\{2, \pi\}$, and $\{x, \pi\}$.

Size 3: There is only 1 subset of size 3. It is the set A itself: $\{2, x, \pi\}$.

Size > 3: Since $|A| = 3$, there are no subsets of A that have size larger than 3.

Thus the power set of A is

$$\mathcal{P}(A) = \{\emptyset, \{2\}, \{x\}, \{\pi\}, \{2, x\}, \{2, \pi\}, \{x, \pi\}, \{2, x, \pi\}\}.$$

Theorem 2.1.21. *If a set A has cardinality n , then the cardinality of the power set of A is $|\mathcal{P}(A)| = 2^n$.*

Proof. We prove this in §5.1 after we develop techniques on counting. \square

2.1.5. Cartesian products.

Definition 2.1.22. Let A and B be sets. The *Cartesian product* of A and B , denoted $A \times B$, is the set of ordered pairs (a, b) , where $a \in A$ and $b \in B$. That is,

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

Example 2.1.23. Let $A = \{0, 1\}$, and let $B = \{4, 5, 6\}$. Then the Cartesian product $A \times B$ is the set of all ordered pairs (a, b) , where a is an element from A and b is an element of B . Thus

$$A \times B = \{(0, 4), (0, 5), (0, 6), (1, 4), (1, 5), (1, 6)\}.$$

Analogously, the Cartesian product $B \times A$ is the set of all ordered pairs (b, a) , where b is an element of B and a is an element of A . Thus

$$B \times A = \{(4, 0), (4, 1), (5, 0), (5, 1), (6, 0), (6, 1)\}.$$

More generally, we can define the Cartesian product of more than two set.

Definition 2.1.24. The *Cartesian product* of sets A_1, A_2, \dots, A_n , denoted $A_1 \times A_2 \times \dots \times A_n$ is the set of ordered n -tuples (a_1, a_2, \dots, a_n) where $a_i \in A_i$ for $i = 1, 2, \dots, n$. That is,

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \dots, n\}.$$

Remark 2.1.25. We use the notation A^2 to denote $A \times A$. Similarly $A^3 = A \times A \times A$, and so on.

Example 2.1.26. \mathbb{R}^2 is the familiar Cartesian plane.

Definition 2.1.27. Let A and B be sets. A subset R of a $A \times B$ is called a *relation* from A to B . A relations from A to itself is called a relation on A .

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) set
 - (b) equal sets
 - (c) empty set
 - (d) subset
 - (e) proper subset
 - (f) cardinality of a set
 - (g) power set of a set
 - (h) Cartesian product of sets
 - (i) relation
2. Write each set in roster notation.
 - (a) $\{n \in \mathbb{Z} \mid n \text{ is odd and } n \leq 10\}$
 - (b) $\{n \in \mathbb{Z} \mid |n| \leq 3\}$
 - (c) the set of positive, even integers less than 6
 - (d) $\{x \in \mathbb{R} \mid x^2 - x - 1 = 0\}$
 - (e) $\{x \in \mathbb{R} \mid x \text{ is the square of an integer and } x < 100\}$
 - (f) $\{x \in \mathbb{R} \mid x^2 = -1\}$
3. Use set builder notation to give a description of each of these sets.
 - (a) $\{2, 3, 4, 5, 6, 7, 8\}$
 - (b) $\{0, 3, 6, 9, 12\}$
 - (c) $\{-2, -1, 0, 1, 2\}$
 - (d) $\{0, 4, 9, 16, 25, 36\}$
4. For each of these pairs of sets, determine whether the first is a subset of the second, the second is a subset of the first, or neither is a subset of the other.
 - (a) the set of dogs; the set of mammals
 - (b) the set of people who speak English; the set of people who speak Japanese
 - (c) the set of math majors; the set of computer science majors
 - (d) the set of people over 6 feet tall; the set of people over 5 feet tall
 - (e) the set of animals; the set of alligators
5. For each of these sets, determine whether 3 is an element of that set.
 - (a) $\{1, 3, 5, 7, 9\}$
 - (b) $\{\{3, 5\}, \{2, 4\}\}$

- (c) $\{(0, 3), (3, 0), (3, 3)\}$
 (d) $\{\{3\}, \{5\}\}$
 (e) the set of positive, odd integers less than 20
6. Find $A \times B$, where $A = \{a, b\}$ and $B = \{0, 4, 8\}$.
7. What is the cardinality of each of these sets?
 (a) $\{1, 3, 3, 5, 5, 6, 10, 10\}$
 (b) $\{\{1, 3, 5, 6\}, \{10, 12, 15\}\}$
 (c) \emptyset
 (d) $\{0, \emptyset\}$
 (e) $\{\emptyset, \{\}\}$
 (f) $\mathcal{P}(\{0, 1\})$
8. Suppose A , B , and C are sets such that $A \subseteq B$ and $B \subseteq C$. Prove that $A \subseteq C$.
9. Let A , A' , B , and B' be sets. Show that if $A \subseteq A'$ and $B \subseteq B'$, then $A \times B \subseteq A' \times B'$.
10. Let $A = \{a, b, c\}$, and let $B = \{x, y\}$. Find the following sets.
 (a) $A \times B$
 (b) $B \times A$
 (c) A^2
 (d) B^3 .
 (e) $\mathcal{P}(A)$
11. Determine whether these statements are true or false.
 (a) $x \in \{x\}$
 (b) $\{x\} \in \{x\}$
 (c) $x \subseteq \{x\}$
 (d) $\{x\} \subseteq \{x\}$
 (e) $\emptyset \subseteq \{x\}$

2.2. Set operations

Goals. To show how set identities are established and to introduce the most important such identities.

2.2.1. Basic operations.

Definition 2.2.1. Let A and B be sets. The *union* of sets A and B , denoted $A \cup B$, is the set that contains those elements that are in A or in B or in both. That is

$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$

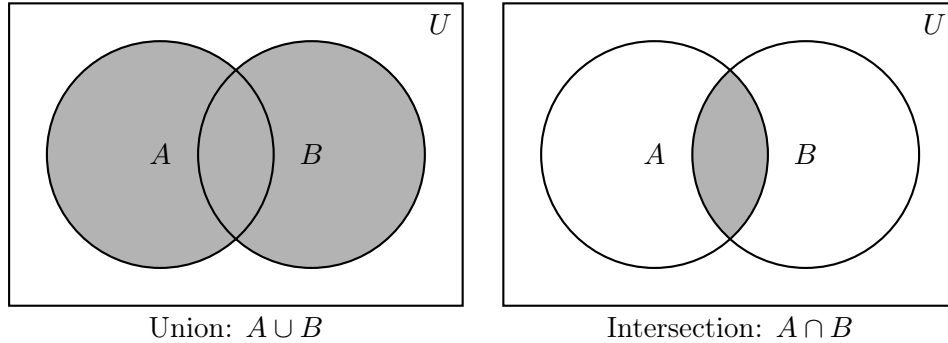


Figure 2.2.1. Venn Diagrams for Union and Intersection.

Example 2.2.2. Let $A = \{1, 2, 3, 5, 10\}$, and let $B = \{2, 3, 4, 5\}$. To compute the union $A \cup B$, we combine all of the elements of A together with all of the elements of B into one set. The union of A and B is

$$A \cup B = \{1, 2, 3, 4, 5, 10\}.$$

Definition 2.2.3. Let A and B be sets. The *intersection* of A and B , denoted $A \cap B$, is the set that contains those elements that are in A and also in B . That is

$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$

Example 2.2.4. Let $A = \{1, 2, 3, 5, 10\}$, and let $B = \{2, 3, 4, 5\}$. To compute the intersection $A \cap B$, we take the elements of A that are also elements of B . In other words, we only keep elements that A and B have in common. The intersection of A and B is

$$A \cap B = \{2, 3, 5\}.$$

Definition 2.2.5. The *difference* of A and B , denoted $A - B$ or $A \setminus B$, is the set containing elements that are in A but not in B . That is

$$A - B = \{x \mid x \in A \wedge x \notin B\}.$$

Analogously, *difference* of B and A , denoted $B - A$ or $B \setminus A$, is

$$B - A = \{x \mid x \in B \wedge x \notin A\}.$$

Example 2.2.6. Let $A = \{1, 2, 3, 5, 10\}$, and let $B = \{2, 3, 4, 5\}$. The difference of A and B consists of members of A that are not also members of B , so we take all the elements of A and throw out the ones that are in B . That gives

$$A - B = \{1, 10\}.$$

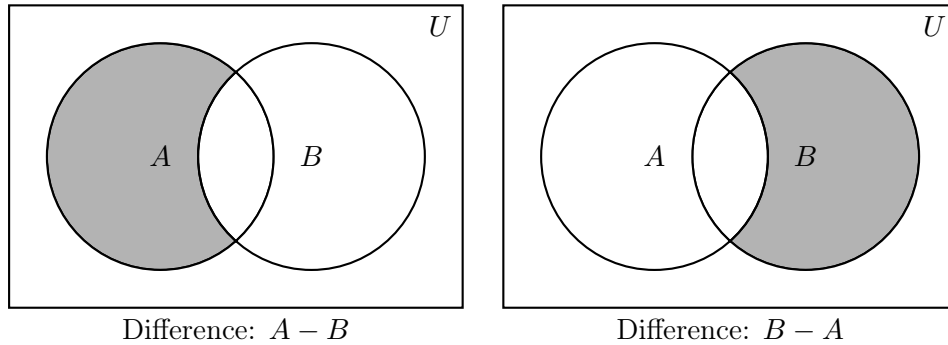


Figure 2.2.2. Venn Diagrams for Set Differences.

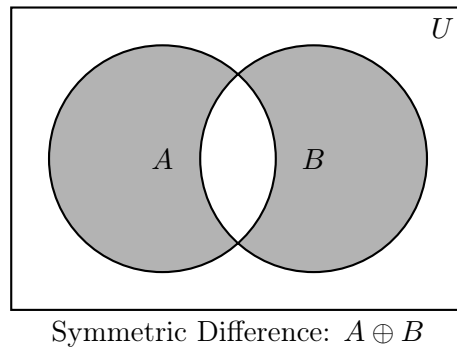


Figure 2.2.3. Venn Diagram for Symmetric Difference.

Analogously, the difference of B and A consists of members of B that are not also members of A , so

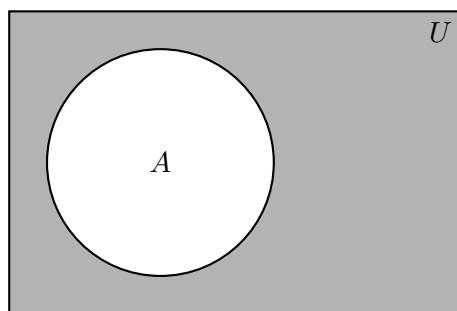
$$B - A = \{4\}.$$

Definition 2.2.7. The *symmetric difference* of A and B , denoted by $A \oplus B$, is the set containing those elements in either A or B , but not in both A and B . That is

$$A \oplus B = \{x \mid x \in A \oplus x \in B\}.$$

Example 2.2.8. Let $A = \{1, 2, 3, 5, 10\}$, and let $B = \{2, 3, 4, 5\}$. To compute the symmetric difference of A and B , we combine the elements of A and B , and then remove the elements that are members of both A and B . Thus

$$A \oplus B = \{1, 4\}.$$

Complement: \bar{A} **Figure 2.2.4.** Venn Diagram for Complement.

Definition 2.2.9. Let U be the universal set. The *complement* of A , denoted \bar{A} , is the difference of U and A . That is

$$\bar{A} = \{x \mid x \notin A\} = U - A.$$

Example 2.2.10. Let $A = \{1, 2, 3, 5, 10\}$, and let $B = \{2, 3, 4, 5\}$. Suppose $U = \{1, 2, \dots, 10\}$. Then the complement of A consists of elements in U that are not in A . We can compute it by taking all of the elements of U and throwing out the elements of A . That gives

$$\bar{A} = \{4, 6, 7, 8, 9\}.$$

Analogously, the complement of B consists of members of U that are not in B , so

$$\bar{B} = \{1, 6, 7, 8, 9, 10\}.$$

Definition 2.2.11. Two sets A and B are *disjoint* if $A \cap B = \emptyset$.

Example 2.2.12. Let $A = \{1, 3, 5, 7\}$, and let $B = \{2, 4, 6\}$. Then A and B have no elements in common, so $A \cap B = \emptyset$. Thus A and B are disjoint.

Example 2.2.13. Prove if $A \subseteq B$, then $A \cap B = A$.

Proof. We need to show

- (1) $A \cup B \subseteq B$, and
- (2) $B \subseteq A \cup B$.

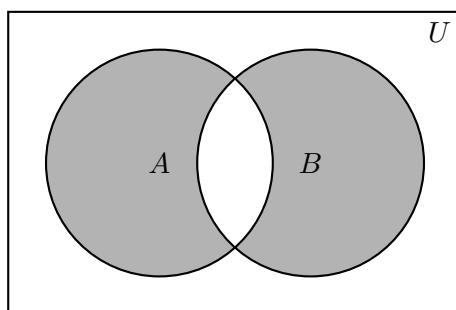
The second statement $B \subseteq A \cup B$ is clear from the definition. Namely, for any element b in B , we must have $b \in A \cup B$ by definition of union.

It remains to show $A \cup B \subseteq B$. Let x be a generic element in $A \cup B$. Then by definition, $x \in A$ or $x \in B$. Since $A \subseteq B$, in either case we have $x \in B$. Thus $A \cup B \subseteq B$.

Therefore $A \cup B = B$ as desired. \square

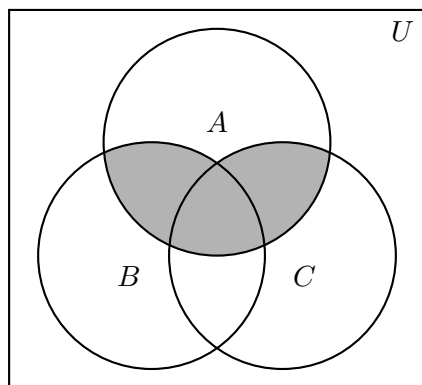
Example 2.2.14. Draw a Venn diagram representing $(A - B) \cup (B - A)$.

From the definition, $(A - B) \cup (B - A)$ consists of elements of A that are not in B together with elements of B that are not in A . We shade the part of A that is not in B and the part of B that is not in A .



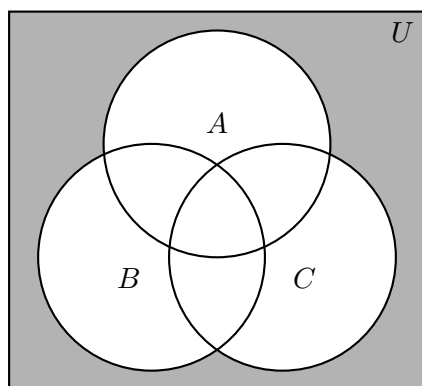
Example 2.2.15. Draw a Venn diagram representing $A \cap (B \cup C)$.

From the definition, we have that $A \cap (B \cup C)$ consists of elements that are in A and in $B \cup C$.



Example 2.2.16. Draw a Venn diagram representing $\bar{A} \cap \bar{B} \cap \bar{C}$.

From the definition, we have that $\bar{A} \cap \bar{B} \cap \bar{C}$ consists of elements that are not in A and not in B and not in C .



2.2.2. Set identities.

Theorem 2.2.17. *Let A and B be sets. Then*

$$A - B = A \cap \overline{B}.$$

One approach is to show $A - B \subseteq A \cap \overline{B}$ and $A \cap \overline{B} \subseteq A - B$. We can do this with generic elements.

Proof using generic elements. Let $x \in A - B$. Then by definition, $x \in A$ and $x \notin B$. Then $x \in \overline{B}$. Since $x \in A$ and $x \in \overline{B}$, we have $x \in A \cap \overline{B}$. Therefore $A - B \subseteq A \cap \overline{B}$.

Let $x \in A \cap \overline{B}$. Then by definition $x \in A$ and $x \in \overline{B}$. Then $x \notin B$. Since $x \in A$ and $x \notin B$, we have $x \in A - B$. Thus $A \cap \overline{B} \subseteq A - B$.

Therefore $A - B = A \cap \overline{B}$. \square

Another approach is to use set builder notation and logical equivalences.

Proof using set builder notation and logical equivalences.

$$\begin{aligned} A - B &= \{x \mid x \in A \wedge x \notin B\} \\ &= \{x \mid x \in A \wedge x \in \overline{B}\} \\ &= \{x \mid x \in A \cap \overline{B}\} \\ &= A \cap \overline{B}. \end{aligned}$$

\square

Another approach is to compute a *membership table*, a table displaying the membership of elements in sets. The concept is similar to using truth tables to display the truth value of propositions. Use 1 to indicate membership in the set and 0 to denote the element is not in the set.

Below, we use a membership table to prove Theorem 2.2.17.

Remark 2.2.18. When using a membership table to prove a set identity, remember to specify which columns give the desired result in the conclusion.

Proof using a membership table.

A	B	\overline{B}	$A \cap \overline{B}$	$A - B$
1	1	0	0	0
1	0	1	1	1
0	1	0	0	0
0	0	1	0	0

Since the last two columns are the same, we have $A \cap \overline{B} = A - B$. \square

Theorem 2.2.19 (Set identities I). *Let A be a set with universal set U .*

Identity laws: $A \cap U = A$; $A \cup \emptyset = A$

Domination laws: $A \cup U = U$; $A \cap \emptyset = \emptyset$

Idempotent laws: $A \cup A = A$; $A \cap A = A$

Complementation law: $\overline{\overline{A}} = A$

Complement laws: $A \cup \overline{A} = U$; $A \cap \overline{A} = \emptyset$

Proof. Exercise. Either compute membership tables and compare columns or use generic elements to show each set is a subset of the other. \square

Theorem 2.2.20 (Set identities II). *Let A and B be sets.*

Commutative laws: $A \cup B = B \cup A$; $A \cap B = B \cap A$.

Absorption laws: $A \cup (A \cap B) = A$; $A \cap (A \cup B) = A$.

Proof. Exercise. Either compute membership tables and compare columns or use generic elements to show each set is a subset of the other. \square

Theorem 2.2.21 (Set identities III). *Let A , B , and C be sets.*

Associative laws: $A \cup (B \cup C) = (A \cup B) \cup C$; $A \cap (B \cap C) = (A \cap B) \cap C$.

Distributive laws: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$; $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

We prove the associative law for intersection in several ways and leave the rest as an exercise. Let A and B be sets. We want to show

$$(A \cap B) \cap C = A \cap (B \cap C).$$

Proof using generic elements. We need to show

(1) $A \cap (B \cap C) \subseteq (A \cap B) \cap C$, and

(2) $(A \cap B) \cap C \subseteq A \cap (B \cap C)$.

Let $x \in A \cap (B \cap C)$. Then x must be in A and also in $B \cap C$. Hence x must be in A and also in B and in C . Since x is in both A and B , we conclude that $x \in A \cap B$. This, together with the fact that $x \in C$ tells us that $x \in (A \cap B) \cap C$, as desired.

Now we prove the other direction. (The argument is nearly identical.) Let $x \in (A \cap B) \cap C$. Then $x \in A \cap B$ and also in C . Hence x is in A and B and C . Since x is in B and in C , we have that $x \in B \cap C$. This, together with the fact that $x \in A$ gives $x \in A \cap (B \cap C)$, as desired. Therefore $A \cap (B \cap C) = (A \cap B) \cap C$ \square

An alternate approach is to use a membership table. Use 1 to indicate membership in the set and 0 to denote the element is not in the set. Remember to specify which columns give the desired result in the conclusion.

Proof using a membership table. We compute the membership table.

A	B	C	$A \cap B$	$(A \cap B) \cap C$	$B \cap C$	$A \cap (B \cap C)$
1	1	1	1	1	1	1
1	1	0	1	0	0	0
1	0	1	0	0	0	0
1	0	0	0	0	0	0
0	1	1	0	0	1	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0

Since the 5th and 7th columns are identical, we have

$$(A \cap B) \cap C = A \cap (B \cap C).$$

□

Theorem 2.2.22 (De Morgan's laws for sets). *Let A and B be sets.*

$$(1) \overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$(2) \overline{A \cap B} = \overline{A} \cup \overline{B}$$

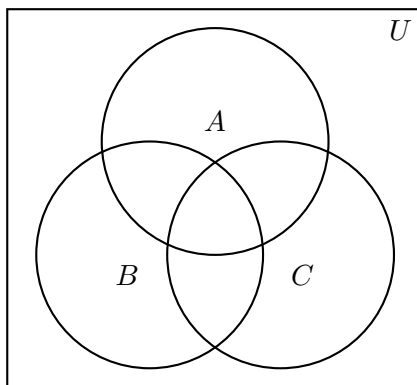
Proof. Exercise. Either compute membership tables and compare columns or use generic elements to show each set is a subset of the other. □

Exercises

- Give the definition for these terms. Be sure to set up any notation that is required.
 - union of sets
 - intersection of sets
 - difference of sets
 - complement of a set
 - disjoint sets
- State precisely De Morgan's laws for sets. Be sure to set up any notation that is required.
- Let $C = \{r, s, t, l, n, e\}$, and let $W = \{f, o, r, t, u, n, e\}$. Compute each of these sets.
 - $C \cap W$
 - $C \cup W$

- (c) $C \oplus W$
- (d) $C - W$
- (e) $W - C$

4. Let A, B, C be sets. Shade the portion of the Venn diagram corresponding to each of these sets.



- (a) $A \cup (B \cap C)$
 - (b) $A \cap (B \cap C)$
 - (c) $A \cap (B \cup C)$
 - (d) $(\overline{A} \cap \overline{B}) \cup \overline{C}$
 - (e) $(A - B) - C$
 - (f) $(A - B) \cup (A - C) \cup (B - C)$
5. Let $A = \{1, 3, 5, 7, 9\}$, and let $B = \{1, 2, 3, 4, 5\}$. Compute the following sets.
- (a) $A \cap B$
 - (b) $A - B$
 - (c) $A \cup B$
 - (d) $B - A$
 - (e) $B \oplus A$
6. Let A and B be sets. Prove that if $A \subseteq B$, then $A \cup B = B$.
7. Prove the set identities in Theorem 2.2.19.
8. Prove the set identities in Theorem 2.2.20.
9. Prove the set identities in Theorem 2.2.21.
10. Prove De Morgan's laws for sets.
11. Let A and B be sets. Prove $A \oplus B = (A - B) \cup (B - A)$.
12. Let A and B be sets.
- (a) Prove if $A \subseteq B$ then $\overline{B} \subseteq \overline{A}$.
 - (b) Prove if $\overline{B} \subseteq \overline{A}$ then $A \subseteq B$.

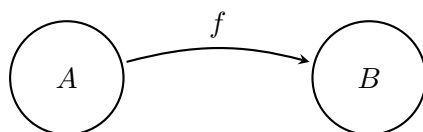


Figure 2.3.1. Function from A to B .

2.3. Functions

Goals. To introduce the concept of a function, the notion of injective functions, surjective functions, and the floor and ceiling functions.

2.3.1. Basic terminology. We have seen examples of functions in previous math classes. Most were likely given by formulas, such as $f(x) = x^2 + 2$. Functions can be given in other ways and are not always given by formulas. The key property of a function is that it accepts inputs and provides a corresponding output value for each possible input.

Definition 2.3.1. Let A and B be nonempty sets. A **function** or **mapping** or **transformation** from A to B , denoted $f: A \rightarrow B$, is an assignment of exactly one element of B to each element of A . In this case, A is called the **domain** and B is called the **codomain** of f .

Definition 2.3.2. Let $f: A \rightarrow B$ be a function, and let a be an element in A . The **image** of a is the unique element of B that is assigned by the function f to the element a . In this case, write $f(a) = b$, where b is the image of a . In this case, we say a is the **preimage** of b .

A function must output a value for *every* input from the domain. Not every element in the codomain needs to be an image, though.

Example 2.3.3. Consider $f(x) = x^2 + 3$. Then f defines a function from \mathbb{R} to \mathbb{R} . We write $f: \mathbb{R} \rightarrow \mathbb{R}$. The domain is \mathbb{R} . The codomain is \mathbb{R} .

On the other hand, $g(x) = \sqrt{x}$ is not a function from \mathbb{R} to \mathbb{R} . This is because, for example, -2 is an element of \mathbb{R} , but the formula does not define a value for $g(-2)$. We can view g as a function from $\mathbb{R}_{\geq 0}$ to \mathbb{R} .

Functions do not have to be given by formulas. They do not have to be from sets of numbers to sets of numbers.

Example 2.3.4. Each point on the surface of the earth has a particular temperature right now, and the temperature (in degrees Celsius) is a real number. Thus, temperature defines a function T from the surface of the earth to \mathbb{R} , where $T(x)$ is the temperature at the point x .

Definition 2.3.5. Let $f: A \rightarrow B$ be a function. The *range* or *image* of f , denoted $\text{im}(f)$ or $f(A)$, is the set of all images of elements of A ,

$$\text{im}(f) = \{f(a) \mid a \in A\}.$$

More generally, if S is a subset of A , the *image* of S under f , denoted $f(S)$, is the set

$$f(S) = \{f(s) \mid s \in S\}.$$

Functions are often given by a table of values.

Example 2.3.6. Consider the prices of some expensive gemstones.

Gemstone	\$ per carat
Blue Diamond	\$3,930,000
Pink Star Diamond	\$1,200,000
Musgravite	\$35,000
Jadeite	\$20,000
Alexandrite	\$12,000
Red Beryl	\$10,000
Padparadscha Sapphire	\$8000

This table describes a function p which gives the price per carat of these gemstones. The domain of p is {Blue Diamond, Pink Star Diamond, Musgravite, Jadeite, Alexandrite, Red Beryl, Padparadscha Sapphire}. We have that $p(\text{Jadeite}) = \$20,000$, but $p(\text{Opal})$ does not exist, since Opal is not in the domain of p .

For a function $f: A \rightarrow B$, we need each element a in A to be assigned to a unique element $f(a)$ in B . We do not require every element b in B to have a corresponding element in A . Specifically, an element of B may be the image of one, none, or several elements of A .

Example 2.3.7. Let P denote the set of all people (alive or dead). Let $F: P \rightarrow P$ be the function defined by the rule $F(x)$ is the father of x . For example, $F(\text{Ben Stiller}) = \text{Jerry Stiller}$, since Jerry is Ben's dad.

On the other hand, if we would run into trouble if we tried to define a function G for grandfathers. This is because people have two grandfathers (a paternal grandfather and a maternal grandfather), so $G(\text{Ben Stiller})$ is ambiguous. Functions are not allowed to have this ambiguity. Thus G does not define a function from P to P , though it *does* define a relation that we discuss in §6.1.

Example 2.3.8. Let $A = \{1, 2, 3\}$, and let $B = \{1, 2, 3, 4\}$. Let $f: A \rightarrow B$ be the function defined by $f(1) = 4$, $f(2) = 3$, and $f(3) = 1$. Let's see what some of the terms mean for this function.

- The *domain* is $\{1, 2, 3\}$.
- The *codomain* is $\{1, 2, 3, 4\}$.
- The *image* of 3 is 1, since $f(3) = 1$.
- A *preimage* of 3 is 2, since $f(2) = 3$.
- The element 2 has *no preimage*, since there is no element in the domain that f sends to 2.
- The *image* of $\{1, 2\}$ under f is $\{3, 4\}$, since

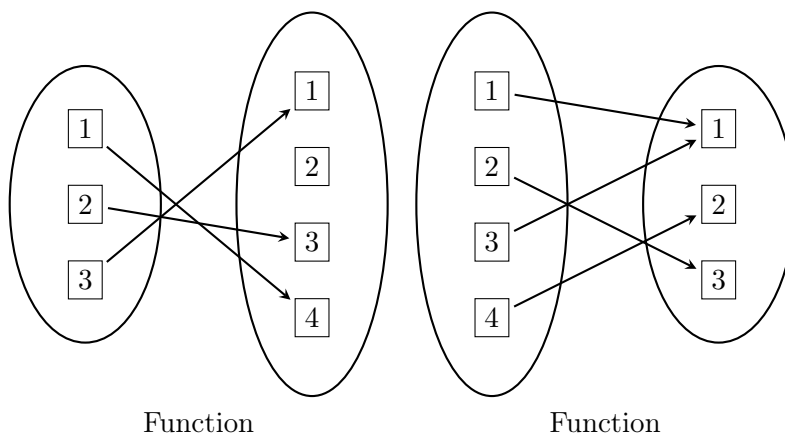
$$f(\{1, 2\}) = \{f(1), f(2)\} = \{4, 3\}.$$

- The *image* or *range* of the function f the same as the image of the domain A under f . This is $\{1, 3, 4\}$, since

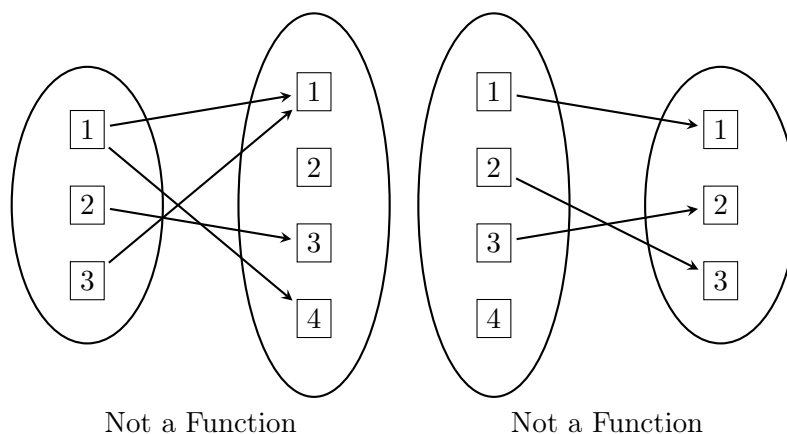
$$f(A) = f(\{1, 2, 3\}) = \{f(1), f(2), f(3)\} = \{4, 3, 1\}.$$

We can think about functions in terms of objects and arrows, where each arrow joins an element a in A to the corresponding element $f(a)$ in B as show in the next example.

Example 2.3.9. On the left, we represent the function Example 2.3.8 graphically using objects and arrows. On the right, we give the representation for a different function. Note that in each case, each element of A is assigned to a unique element of B . An element of B may be the image of 1, none, or several elements of A .



Example 2.3.10. These are graphical representations of relations that are not functions. On the left, the element 1 is assigned to 1 and 4. Functions need to assign exactly one element of the codomain to each element in the domain. On the right, the element 1 is not assigned to any element. Again, this is not a function since functions need to assign exactly one element of the codomain to each element in the domain.



Remark 2.3.11. A function is a special kind of relation from A to B . Specifically, each element a in A is related to the element $f(a)$ in B . We discuss relations in detail in §6.1.

Example 2.3.12. The volume of a box is a function of its length, width, and height. Let's break this example down and recognize the components described above.

What is the *domain*? Consider the set of triples of numbers representing the length, width, and height of a box. Then each number must be nonnegative (if we allow degenerate boxes). For concreteness, let's consider ordered triples (ℓ, w, h) for length, width, and height. The set of all such triples is the domain of the volume function.

The volume of a box is a real number, so we can take the *codomain* to be \mathbb{R} .

If V is the name of our volume function, then we have $V: \mathbb{R}_{\geq 0}^3 \rightarrow \mathbb{R}$. The *image* or *range* of V is the set of realizable outputs of the function V . Thus the image is the set of nonnegative real numbers,

$$\text{im}(V) = \mathbb{R}_{\geq 0}.$$

Definition 2.3.13. The *floor* of a real number x , denoted $\lfloor x \rfloor$, is the largest integer that is less than or equal to x .

Example 2.3.14. The floor $\lfloor 2.7 \rfloor = 2$, since 2 is the largest integer that is less than or equal to 2.7.

⚠ Computing the floor of a real number is different from just throwing away everything after the decimal. In the next example, we compute the floor of a negative number.

Example 2.3.15. The floor $\lfloor -2.7 \rfloor = -3$, since -3 is the largest integer that is less than or equal to -2.7 .

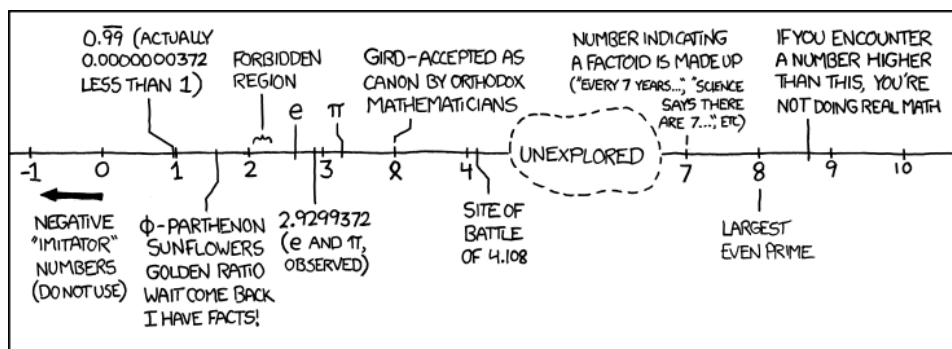


Figure 2.3.2. xkcd: Number Line. (<https://xkcd.com/899/>) The Wikipedia page List of Numbers opens with “This list is incomplete; you can help by expanding it.”

Example 2.3.16. The floor of an integer is the integer itself, so $\lfloor 3 \rfloor = 3$, $\lfloor -3 \rfloor = -3$, and $\lfloor 0 \rfloor = 0$.

Definition 2.3.17. The *ceiling* of a real number x , denoted $\lceil x \rceil$, is the smallest integer that is greater than or equal to x .

Example 2.3.18. The ceiling $\lceil 2.7 \rceil = 3$, since 3 is the smallest integer that is greater than or equal to 2.7.

⚠ Computing the ceiling of a real number is different from just throwing away everything after the decimal and adding 1. In the next example, we compute the ceiling of a negative number.

Example 2.3.19. The ceiling $\lceil -2.7 \rceil = -2$, since -2 is the smallest integer that is greater than or equal to -2.7 .

Example 2.3.20. The ceiling of an integer is the integer itself, so $\lceil 3 \rceil = 3$, $\lceil -3 \rceil = -3$, and $\lceil 0 \rceil = 0$.

Remark 2.3.21. If we think of x as a point on the usual real number line, then the floor of x is the integer that is directly to the left of x , unless x itself is an integer. The ceiling of x is the integer that is directly to the right of x , unless x itself is an integer.

2.3.2. Injective functions. Some functions “lose information” in the sense that the output does not uniquely determine the input. For example, consider the father function F given in Example 2.3.7. Knowing that the father of x is George Foreman does uniquely determine x since George Foreman has twelve children (including five sons all named George). Other functions do not lose information in this way. This property is known as *injectivity*. Namely, *injective functions* are the functions that do not lose information.

Definition 2.3.22. Let $f: A \rightarrow B$ be a function. The function f is *injective* or *one-to-one* if for all a and \hat{a} in A , $f(a) = f(\hat{a})$ implies $a = \hat{a}$. An injective function is called an *injection*.

Example 2.3.23. The squaring function $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = x^2$ is not injective because $f(-2) = f(2)$, but $-2 \neq 2$.

Example 2.3.23 shows the general technique for showing a function is not injective.

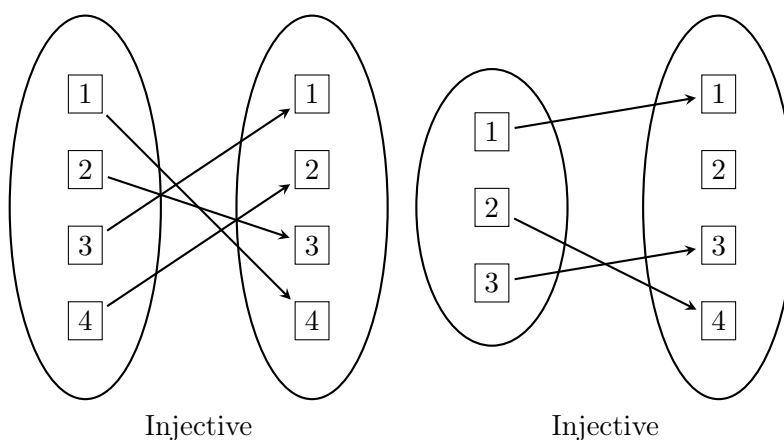
Proof Technique 2.3.24 (Show f is not injective). Suppose $f: A \rightarrow B$ is a function, and we want to show f is not injective.

- (1) Find two different elements $a \neq \hat{a}$ in the domain A such that $f(a) = f(\hat{a})$.
- (2) Conclude f is not injective.

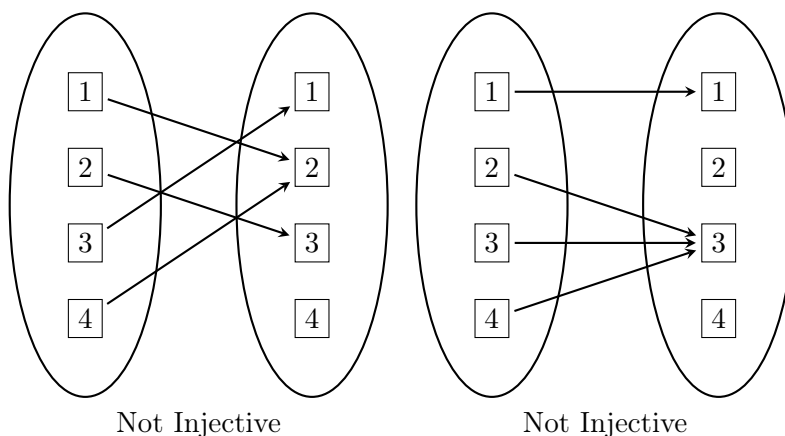
Example 2.3.25. The function U denote the set of UNCG students, and consider the function $\text{id}: U \rightarrow \mathbb{Z}$ that assigns to each student their university ID. In other words, $\text{id}(x)$ is the university ID of x . These ID numbers are unique by design so that so that no two students have the same university ID. Specifically, the only way to have $\text{id}(\text{student } a) = \text{id}(\text{student } \hat{a})$ is to have student a be the same student as \hat{a} . The id function is injective.

For an injective function, every element in the codomain can have *at most one* preimage.

Example 2.3.26. These are graphical representations of injective functions. In each instance, each element in the codomain has at most one preimage.



Example 2.3.27. These are graphical representations of functions that are not injective. In each instance, there is an element in the codomain has more than one preimage. On the left, both 1 and 4 map to 2, so 2 has two preimages. On the right, 2, 3, and 4 map to 3, so 3 has three preimages.



Example 2.3.28. Let $f: \mathbb{R} \rightarrow \mathbb{R}$ be the function $f(x) = 3x - 2$. Let's show f is injective using the definition.

- (1) First, we will rewrite what we want to show in the form “if p , then q .” This will involve giving names to objects, and recalling the definition of *injective*. We want to prove f is injective, which means we want to prove: **If $f(a_1) = f(a_2)$, then $a_1 = a_2$.**
- (2) As before, to prove a statement of the form “if p , then q ” directly, we assume p is true and try to show q . This is commonly where we set some notation as well. It is also a good place to remind the reader what we want to show. e.g., **Let a_1 and a_2 be real numbers such that $f(a_1) = f(a_2)$. We want to show that $a_1 = a_2$.**
- (3) Now look back at the goal. It should be some relationship between a_1 and a_2 . It makes sense to write out what we know and simplify to see if we get what we want.

$$\begin{aligned}
 f(a_1) &= f(a_2) \\
 3a_1 - 2 &= 3a_2 - 2 \\
 3a_1 &= 3a_2 \\
 a_1 &= a_2
 \end{aligned}$$

- (4) The part above should complete the proof. Since we chose generic a_1 and a_2 from the domain of f , the argument covers all a_1 and a_2 in the domain of f .

The technique in Example 2.3.28 can be generalized to prove a function is injective.

Proof Technique 2.3.29 (Show f is injective). Suppose $f: A \rightarrow B$ is a function, and we want to show f is injective.

- (1) Fix generic elements a and \hat{a} in the domain A such that $f(a) = f(\hat{a})$.
- (2) Deduce $a = \hat{a}$.
- (3) Conclude f is injective.

2.3.3. Surjective functions. Some functions have a codomain that is larger than it needs to be. Namely, they have a codomain that is larger than the image of the function. For example, consider the absolute value function $f: \mathbb{R} \rightarrow \mathbb{R}$ given by $f(x) = |x|$. Then the codomain of f is \mathbb{R} , but the image of f is just the set of nonnegative real numbers, $\text{im}(f) = \mathbb{R}_{\geq 0}$.

Negative numbers are in the codomain and have no preimage. Other functions have the property that the codomain is as small as possible. Namely, the codomain is exactly equal to the image. For example, we can modify f to define a new function $g: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ given by $g(x) = |x|$. Notice now that the codomain is equal to the image. Another way of saying that is every element of the codomain has at least one preimage. This property is known as *surjectivity*. Namely, *surjective functions* have codomains equal to their image.

Definition 2.3.30. Let $f: A \rightarrow B$ be a function. The function f is *surjective* or *onto* if for each b in the codomain B , there exists an element a in the domain A such that $f(a) = b$; equivalently, if $f(A) = B$. A surjective function is called a *surjection*.

Example 2.3.31. The squaring function $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = x^2$ is not surjective because $-2 \in \mathbb{R}$, but there is no $a \in \mathbb{R}$ such that $f(a) = -2$.

Example 2.3.31 shows the general technique for proving a function is not surjective.

Proof Technique 2.3.32 (Show f is not surjective). Suppose $f: A \rightarrow B$ is a function, and we want to prove f is not surjective.

- (1) Find particular b in the codomain B that is not in the image of f , i.e., b such that $f(a) \neq b$ for all a in the domain A .
- (2) Conclude f is not surjective.

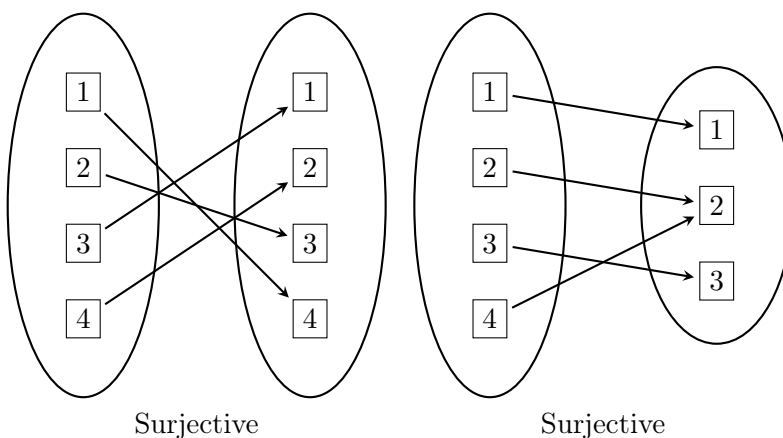
We can make a non-surjective function into a surjective one by shrinking the codomain.

Example 2.3.33. The squaring function is surjective onto $\mathbb{R}_{\geq 0}$. Specifically, the function $g: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ given by $g(x) = x^2$ is surjective.

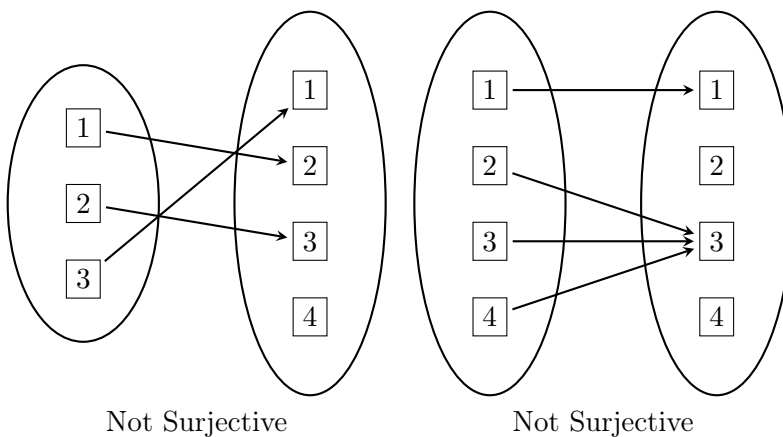
⚡ A function is a triple of data (f, A, B) . In particular, the domain and codomain are part of the function. Because of this, when we change the domain or codomain, we are really creating a *new* function. For example, the squaring function from \mathbb{R} to \mathbb{R} is *different* from the squaring function from \mathbb{R} to $\mathbb{R}_{\geq 0}$.

For a surjective function, every element in the codomain must have *at least one* preimage.

Example 2.3.34. These are graphical representations of surjective functions. In each instance, each element in the codomain has at least one preimage.



Example 2.3.35. These are graphical representations of functions that are not surjective. In each instance, there is an element in the codomain that has no preimage. On the left, there is no element that maps to 4, so 4 has no preimage. On the right, no elements map to 2 or 4, so both 2 and 4 have no preimages.



Example 2.3.36. Let $f(x) = 3x - 2$. We prove that f is surjective as follows.

- (1) First we recall the definition of surjective. At the same time, we give names to things. We want to prove: **For every b in \mathbb{R} (the codomain of f), there exists a in \mathbb{R} (the domain of f) such that $f(a) = b$.**
- (2) Now this still does not look like a “if p , then q ” type of statement. What helps is the same thing as the previous exercise. Namely, if we pick a generic b in \mathbb{R} , then the argument will work for every b in \mathbb{R} . That means we want to prove the statement: **If $b \in \mathbb{R}$, then there exists $a \in \mathbb{R}$ such that $f(a) = b$.**
- (3) As before, to prove a statement of the form “if p , then q ” directly, we assume p is true and try to show q . This is commonly where we set some notation as well. It is also a good place to remind the reader what we want to show. e.g., **Let $b \in \mathbb{R}$. We want to find $a \in \mathbb{R}$ such that $f(a) = b$.**
- (4) Since we want to find a , we should write out the conditions that a must satisfy, and see if we can solve for a .

$$\begin{aligned} f(a) &= b \\ 3a - 2 &= b \\ 3a &= b + 2 \\ a &= \frac{b + 2}{3}. \end{aligned}$$

- (5) We have found a as desired. Since b is generic, the argument holds for all b in \mathbb{R} . That completes the proof.

The technique in Example 2.3.36 can be generalized to prove a function is surjective.

Proof Technique 2.3.37 (Show f is surjective). Suppose $f: A \rightarrow B$ is a function, and we want to prove f is surjective.

- (1) Fix a generic element b in the codomain B .
- (2) Find a in the domain A such that $f(a) = b$.
- (3) Conclude f is surjective.

2.3.4. Bijective functions.

Definition 2.3.38. Let $f: A \rightarrow B$. The function f is **bijective** if f is injective and surjective, i.e., one-to-one and onto. A bijective function is called a **bijection** or **one-to-one correspondence**.

Example 2.3.39. The identity function $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = x$ is bijective.

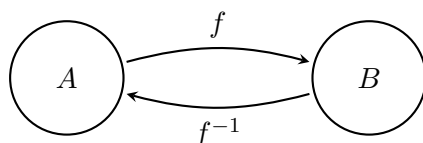


Figure 2.3.3. Inverse of a function from A to B .

Proof Technique 2.3.40 (Show f is bijective). Suppose $f: A \rightarrow B$ is a function, and we want to prove f is bijective.

- (1) Use Proof Technique 2.3.29 to show f is injective.
- (2) Use Proof Technique 2.3.37 to show f is surjective.
- (3) Conclude f is bijective.

Remark 2.3.41. If A and B are finite sets of the same size, a function $f: A \rightarrow B$ is surjective if and only if f is injective. Furthermore, if $|A| < |B|$ then f cannot be surjective. If $|B| < |A|$ then f cannot be injective.

Example 2.3.42. The area of a circle is a function of its radius. This function is injective, because if the area of a circle of radius r is equal to the area of a circle of radius r' , we must have $r = r'$. This function is not surjective, when viewed as a function from $\mathbb{R}_{>0}$ to \mathbb{R} , since there is no radius that would give rise to area -1 . When we restrict the codomain to $\mathbb{R}_{>0}$, this function is surjective since we can construct a circle for any given area.

Proof Technique 2.3.43 (Show f is not bijective). Suppose $f: A \rightarrow B$ is a function, and we want to prove f is not bijective.

- (1) Complete one of the following.
 - (a) Use Proof Technique 2.3.24 to show f is not injective.
 - (b) Use Proof Technique 2.3.32 to show f is not surjective.
- (2) Conclude f is not bijective.

Definition 2.3.44. Let $f: A \rightarrow B$ be a bijection. The *inverse* of f , denoted f^{-1} , is a function $f^{-1}: B \rightarrow A$ that assigns to an element $b \in B$, the unique element $a \in A$ such that $f(a) = b$.

From the definition, it immediately follows that f is an invertible function if and only if

$$(2.1) \quad f(f^{-1}(b)) = b \quad \text{for all } b \in B; \text{ and}$$

$$(2.2) \quad f^{-1}(f(a)) = a \quad \text{for all } a \in A.$$

Remark 2.3.45.

- (1) The definition of inverse makes sense since f is a bijection.
- (2) f^{-1} does not mean $\frac{1}{f}$.
- (3) Non-bijective functions do not have an inverse.

Example 2.3.46. Consider the function $f(x) = 3x + 2$. Let's compute the inverse of f .

Let $y = f(x)$. Then $f^{-1}(y) = x$. That means we can just take $y = 3x + 2$, and solve for x . Doing so, we see that $x = \frac{y-2}{3}$. That means $f^{-1}(y) = \frac{y-2}{3}$. Convention often dictates the input variable be named x , so we write it as

$$f^{-1}(x) = \frac{x-2}{3}.$$

Finding the inverse of a function is more difficult than checking if given functions are inverses of each other.

Example 2.3.47. Let f and g be functions from \mathbb{R} to \mathbb{R} given by $f(x) = x^3 - 4$ and $g(x) = \sqrt[3]{x+4}$. Let's check whether g is the inverse of f .

We just need to check that (2.1) and (2.2) both hold. We compute each one separately. For all $x \in \mathbb{R}$, we have

$$\begin{aligned} f(g(x)) &= f(\sqrt[3]{x+4}) \\ &= (\sqrt[3]{x+4})^3 - 4 \\ &= (x+4) - 4 \\ &= x; \end{aligned}$$

$$\begin{aligned} g(f(x)) &= g(x^3 - 4) \\ &= \sqrt[3]{(x^3 - 4) + 4} \\ &= \sqrt[3]{x^3} \\ &= x. \end{aligned}$$

Thus g is the inverse of f . (Note: This also shows that f is the inverse of g .)

Definition 2.3.48. Let $f: A \rightarrow B$, and let $g: B \rightarrow C$. The *composition*, denoted $g \circ f$, is the function $g \circ f: A \rightarrow C$ defined by

$$(g \circ f)(x) = g(f(x)).$$

Example 2.3.49. Let $f(x) = x^2 + 2x - 1$ and $g(x) = 3x - 5$ be functions from \mathbb{R} to \mathbb{R} .

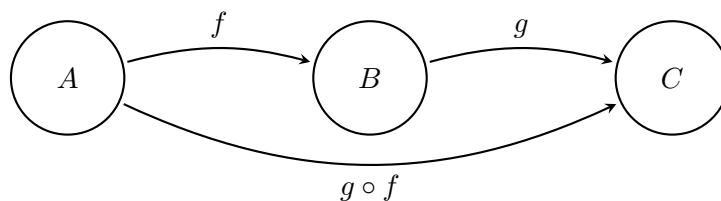


Figure 2.3.4. Composition of two functions.

Let's compute some compositions. First, consider $f \circ g$.

$$\begin{aligned}
 (f \circ g)(x) &= f(g(x)) \\
 &= f(3x - 5) \\
 &= (3x - 5)^2 + 2(3x - 5) - 1 \\
 &= (9x^2 - 30x + 25) + (6x - 10) - 1 \\
 &= 9x^2 - 24x + 14.
 \end{aligned}$$

Next, let's compose them in the opposite order.

$$\begin{aligned}
 (g \circ f)(x) &= g(f(x)) \\
 &= g(x^2 + 2x - 1) \\
 &= 3(x^2 + 2x - 1) - 5 \\
 &= (3x^2 + 6x - 3) - 5 \\
 &= 3x^2 + 6x - 8.
 \end{aligned}$$

Finally, let's compose g with itself.

$$\begin{aligned}
 (g \circ g)(x) &= g(g(x)) \\
 &= g(3x - 5) \\
 &= 3(3x - 5) - 5 \\
 &= (9x - 15) - 5 \\
 &= 9x - 20.
 \end{aligned}$$

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) function
 - (b) image of an element

- (c) image or range of a function
 - (d) image of a set under a function
 - (e) preimage of an element
 - (f) floor of a real number
 - (g) ceiling of a real number
 - (h) injective function
 - (i) surjective function
 - (j) bijective function
 - (k) inverse of a function
 - (l) composition of functions
2. Let $f(x) = x^2 + 1$ and $g(x) = x + 2$ be functions from \mathbb{R} to \mathbb{R} .
 - (a) Find the composition $h = f \circ g$. What is the image of h ?
 - (b) Find the composition $i = g \circ f$. What is the image of i ?
 3. Construct a function that is injective but not surjective.
 4. Construct a function that is surjective but not injective.
 5. Construct a function that is neither surjective nor injective.
 6. Construct a function that is both injective and surjective.
 7. Prove that $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = 3x + 1$ is surjective.
 8. Prove that $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = 3x + 1$ is injective.
 9. Let \mathcal{W} be the set of words in the Oxford English Dictionary, and let \mathcal{A} be the English alphabet. Let $f: \mathcal{W} \rightarrow \mathcal{A}$ be the function defined by
$$f(x) = \text{first letter of } x.$$
 - (a) What is the image of f ?
 - (b) Is f surjective? Justify.
 - (c) Is f injective? Justify.
 10. Give an explicit example of a function from \mathbb{Z} to \mathbb{Z} that is
 - (a) injective, but not surjective.
 - (b) surjective, but not injective.
 - (c) injective and surjective.
 - (d) neither injective nor surjective.
 11. Let $S = \{-2, -1, 0, 3, 4, 5\}$ Find $f(S)$ for each of the following. Be sure to express the answer as a set.
 - (a) $f(x) = 1$
 - (b) $f(x) = 2x + 1$
 - (c) $f(x) = x^2$
 - (d) $f(x) = \lceil \frac{x}{2} \rceil$
 - (e) $f(x) = \lfloor \frac{x^2+1}{3} \rfloor$
 12. Let A and B be sets, and let $f: A \rightarrow B$ be a function. Let S and T be subsets of A . Prove $f(S \cup T) = f(S) \cup f(T)$.

13. Let A and B be sets, and let $f: A \rightarrow B$ be a function. Let S and T be subsets of A .
- Prove $f(S \cap T) \subseteq f(S) \cap f(T)$.
 - Give an explicit example that shows the inclusion above can be proper.
14. Let A and B be sets, and let $f: A \rightarrow B$ be a function. Let S and T be subsets of A . Prove that if f is injective, then $f(S \cap T) = f(S) \cap f(T)$.

2.4. Sequences and summations

Goals. To introduce terminology used for sequences and summations. To introduce recurrence relations and some methods for solving them. To work with summations and establish several important summation techniques.

2.4.1. Basic Terminology.

Definition 2.4.1. A *sequence* is a function from a subset of the integers (usually $\{0, 1, 2, \dots\}$ or $\{1, 2, 3, \dots\}$) to a set S . We use a_n to denote the image of n , and we call a_n a *term* of the sequence. We use the notation $\{a_n\}$ to describe the whole sequence, and a_n represents an individual term in the sequence.

Remark 2.4.2. Note the notation conflicts with our notation for sets. The context will make it clear which we are discussing.

Example 2.4.3. The sequence $\{a_n\}$, where $a_n = \frac{1}{n^2}$ for $n = 1, 2, 3, \dots$ has terms

$$1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \dots, \frac{1}{n^2}, \frac{1}{(n+1)^2}, \dots$$

2.4.2. Geometric and arithmetic sequences.

Definition 2.4.4. A *geometric progression* is a sequence of the form

$$a, ar, ar^2, ar^3, \dots, ar^n, ar^{n+1}, \dots,$$

where the *initial term* a and *common ratio* r are real numbers.

Suppose $\{a_n\}$ is a geometric progression. How do we compute its initial term and common ratio? The initial term is the first term in the sequence. The common ratio is the ratio of consecutive terms:

$$\frac{a_{n+1}}{a_n} = \frac{ar^{n+1}}{ar^n} = r,$$

assuming the sequence is not the zero sequence.

This is also how we can tell if a sequence is a geometric progression. A geometric progression must have the ratio of consecutive terms constant, no matter where we look in the sequence.

Example 2.4.5. The sequence $3, 6, 12, 24, \dots$ is a geometric progression with initial term $a = 3$. The common ratio is $r = 2$, since

$$\frac{6}{3} = \frac{12}{6} = \frac{24}{12} = \dots = 2.$$

Example 2.4.6. The sequence $\{a_n\} = 2 \cdot 5^n$ for $n = 0, 1, 2, \dots$ is a geometric progression with initial term $a = 2$, since $a_0 = 2 \cdot 5^0 = 2$. The common ratio $r = 5$, since

$$\frac{a_{n+1}}{a_n} = \frac{2 \cdot 5^{n+1}}{2 \cdot 5^n} = 5.$$

Example 2.4.7. The sequence $1, 3, 5, 7, 9, \dots$ is not a geometric progression since the ratio of consecutive terms is not constant. For example,

$$\frac{3}{1} \neq \frac{5}{3}.$$

Definition 2.4.8. An *arithmetic progression* is a sequence of the form

$$a, a + d, a + 2d, \dots, a + nd, \dots,$$

where the *initial term* a and *common difference* d are real numbers.

Suppose $\{a_n\}$ is an arithmetic progression. How do we compute its initial term and common difference? The initial term is the first term in the sequence. The common difference is the difference of consecutive terms:

$$a_{n+1} - a_n = (a + (n + 1)d) - (a + nd) = d.$$

This is also how we can tell if a sequence is an arithmetic progression. An arithmetic progression must have the difference of consecutive terms constant, no matter where we look in the sequence.

Example 2.4.9. The sequence $2, 5, 8, 11, 14, \dots$ is an arithmetic progression with initial term 2 . The common difference is 3 , since

$$5 - 2 = 8 - 5 = 11 - 8 = 14 - 11 = \dots = 3.$$

Example 2.4.10. The sequence $\{a_n\}$ defined by $a_n = 3 + 5n$ for $n = 0, 1, 2, \dots$ is an arithmetic progression with initial term $a = 3$, since $a_0 = 3 + 5 \cdot 0 = 3$. The common difference is $d = 5$, since

$$a_{n+1} - a_n = (3 + 5(n + 1)) - (3 + 5n) = 5.$$

Example 2.4.11. The sequence $1, 2, 4, 8, 16, 32, \dots$ is not an arithmetic progression, since the difference of consecutive terms is not constant. For example,

$$2 - 1 \neq 4 - 2.$$

2.4.3. Recurrence relations.

Definition 2.4.12. A *recurrence relation* for a sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence.

Example 2.4.13 (Compound interest). Suppose we deposit P_0 dollars in a savings account that yields 3% interest per year, compounded annually. Find a formula for the amount in the account after n years.

Let $\{P_n\}$ be the sequence where the n th term is the amount in the account after n years. Then

$$\begin{aligned} P_0 &= P_0 \\ P_1 &= P_0 + 0.03P_0 = 1.03P_0 \\ P_2 &= P_1 + 0.03P_1 = 1.03P_1 = (1.03)^2P_0 \\ P_3 &= P_2 + 0.03P_2 = 1.03P_2 = (1.03)^3P_0 \\ &\vdots \\ P_n &= 1.03P_{n-1} = (1.03)^nP_0. \end{aligned}$$

Example 2.4.14. Suppose $\{a_n\}$ is a sequence that satisfies the recurrence relation $a_n = -a_{n-1} + 1$ for $n = 1, 2, 3, \dots$, and suppose $a_0 = 2$.

Then

$$\begin{aligned} a_0 &= 2 \\ a_1 &= -a_0 + 1 = -2 + 1 = -1 \\ a_2 &= -a_1 + 1 = -(-1) + 1 = 2 \\ a_3 &= -a_2 + 1 = -2 + 1 = -1 \\ a_4 &= -a_3 + 1 = -(-1) + 1 = 2 \\ &\vdots \\ a_n &= \begin{cases} 2 & \text{if } n \text{ is even,} \\ -1 & \text{if } n \text{ is odd.} \end{cases} \end{aligned}$$

Definition 2.4.15. A *solution* of a recurrence relation is a sequence whose terms satisfy the recurrence relation.

Example 2.4.16. Is $\{a_n\}$, where $a_n = 3n$ for $n = 0, 1, 2, \dots$ a solution of the recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ for $n = 2, 3, \dots$?

We just need to check if the terms of the sequence satisfy the recurrence relation. We compute for $n \geq 2$,

$$2a_{n-1} - a_{n-2} = 2(3(n-1)) - (3(n-2)) = (6n-6) - (3n-6) = 3n = a_n.$$

Thus $a_n = 3n$ is a solution of the recurrence relation.

Note: It is easy to check that the sequence $\{b_n\}$, defined by $b_n = 5$ for $n = 0, 1, 2, \dots$ is also a solution of the recurrence relation. In particular, a recurrence relation can have more than one solution.

Example 2.4.17. The sequence $\{a_n\}$ defined by $a_n = n!$ for $n = 1, 2, 3, \dots$ is a solution to the recurrence relation $a_n = na_{n-1}$, since $n! = n((n-1)!)$.

Next, we define a famous recursively defined sequence named for a 12th century Italian mathematician.

Definition 2.4.18. The *Fibonacci sequence* f_0, f_1, f_2, \dots is defined by the initial conditions $f_0 = 0$, $f_1 = 1$, and the recurrence relation

$$f_n = f_{n-1} + f_{n-2},$$

for $n = 2, 3, \dots$

Example 2.4.19. It is straightforward to compute more terms of the Fibonacci sequence.

$$f_0 = 0$$

$$f_1 = 1$$

$$f_2 = f_1 + f_0 = 1 + 0 = 1$$

$$f_3 = f_2 + f_1 = 1 + 1 = 2$$

$$f_4 = f_3 + f_2 = 2 + 1 = 3$$

$$f_5 = f_4 + f_3 = 3 + 2 = 5$$

Using linear algebra, we can also find an explicit formula, called a *closed formula* for the terms of this sequence.

2.4.4. Summations. We now consider the addition of terms of a sequence.



Figure 2.4.1. xkcd: Tabletop Roleplaying. (<https://xkcd.com/244/>)
I may have also tossed one of a pair of teleportation rings into the ocean, with interesting results.

Definition 2.4.20. Let $\{a_n\}$ be a sequence of numbers. For a positive integer n , the **sum** from 1 to n of a_n , denoted $\sum_{i=1}^n a_i$, is

$$\sum_{i=1}^n a_i = a_1 + a_2 + \cdots + a_n.$$

More generally, for $m \leq n$, we have

$$\sum_{i=m}^n a_i = a_m + a_{m+1} + \cdots + a_n.$$

Even more generally, for a finite set I , we $\sum_{i \in I} a_i$ be the sum of the a_i with i in I .

Remark 2.4.21. We are using the fact that addition is an associative operator. In particular,

$$\sum_{i=1}^n a_i = (((\dots((a_1 + a_2) + a_3) + a_4) + \dots) + a_n)$$

For summing infinitely many things, one must define the sum to be the limit of partial sums as described above. This will be covered in more detail in Calculus, since the notion of limit is required.

Python Code Snippet 2.4.22. Given a sequence $\{a_i\}$ as a function in Python, the following function computes the sum from m to n of $\{a_i\}$.

```
def summation(a,m,n):
    '''
    Return summation of a_i from m to n.
    '''
    partial_sum = 0 # initialize total
    for i in range(m,n + 1):
        partial_sum += a(i)
    return total
```

Example 2.4.23.

$$\sum_{i=1}^3 i^2 = 1^2 + 2^2 + 3^2 = 14.$$

Theorem 2.4.24 (Properties of summation). *Let $\{a_n\}$ and $\{b_n\}$ be defined on a finite set S , and let c be a real number. Suppose $\sum_{i \in S} a_i$ and $\sum_{i \in S} b_i$ exists. Then*

$$\begin{aligned} \sum_{i \in S} a_i + \sum_{i \in S} b_i &= \sum_{i \in S} (a_i + b_i), \\ c \sum_{i \in S} a_i &= \sum_{i \in S} ca_i. \end{aligned}$$

We can use these properties to break down complicated sums into simpler ones.

Example 2.4.25. Break down the summation

$$\sum_{k=0}^{100} (3k^2 - 5k + 2)$$

to a sum of simpler summations.

We use the properties in Theorem 2.4.24

$$\begin{aligned} \sum_{k=0}^{100} (3k^2 - 5k + 2) &= \sum_{k=0}^{100} 3k^2 - \sum_{k=0}^{100} 5k + \sum_{k=0}^{100} 2 \\ &= 3 \sum_{k=0}^{100} k^2 - 5 \sum_{k=0}^{100} k + 2 \sum_{k=0}^{100} 1. \end{aligned}$$

It remains to work out formulas these simpler summations:

$$\sum_{k=0}^{100} k^2, \quad \sum_{k=0}^{100} k, \quad \text{and} \quad \sum_{k=0}^{100} 1.$$

That is done in Example 2.4.29.

Theorem 2.4.26. *If a and r are real numbers with $r \neq 0$, then*

$$\sum_{i=0}^n ar^i = \begin{cases} \frac{ar^{n+1} - a}{r-1} & \text{if } r \neq 1, \\ (n+1)a & \text{if } r = 1. \end{cases}$$

Proof. If $r = 1$, then

$$\sum_{i=0}^n ar^i = \sum_{i=0}^n a = \underbrace{a + a + \cdots + a}_{n+1 \text{ times}} = (n+1)a.$$

Now suppose $r \neq 1$. Let $s = \sum_{i=0}^n ar^i$. Then

$$rs = \sum_{i=0}^n ar^{i+1} = \sum_{i=1}^{n+1} ar^i.$$

Then

$$s(r-1) = rs - s = \sum_{i=1}^{n+1} ar^i - \sum_{i=0}^n ar^i = ar^{n+1} - a.$$

Dividing by $r-1$ gives the result. \square

Example 2.4.27. Let's compute the sum

$$1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^{100}}.$$

We rewrite the sum as

$$1 + \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right)^2 + \cdots + \left(\frac{1}{2}\right)^{100} = \sum_{i=0}^{100} \left(\frac{1}{2}\right)^i.$$

This is the sum of a geometric progression with initial term $a = 1$ and common ratio $r = \frac{1}{2}$. Using Theorem 2.4.26 with $a = 1$, $r = \frac{1}{2}$, and $n = 100$, we have that the sum is equal to

$$\frac{ar^{n+1} - a}{r-1} = \frac{\left(\frac{1}{2}\right)^{101} - 1}{\frac{1}{2} - 1} = 2 \left(1 - \frac{1}{2^{101}}\right),$$

which is just a bit less than 2.

Theorem 2.4.28 (Summation formulae).

$$\begin{aligned}\sum_{k=1}^n 1 &= n \\ \sum_{k=1}^n k &= \frac{n(n+1)}{2} \\ \sum_{k=1}^n k^2 &= \frac{n(n+1)(2n+1)}{6} \\ \sum_{k=1}^n k^3 &= \frac{n^2(n+1)^2}{4}.\end{aligned}$$

Proof. We prove the first two formulae here. We reprove the second and prove the latter two later in the section on mathematical induction §4.1.

We have

$$\sum_{k=1}^n 1 = \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}} = n.$$

Suppose n is even. Then

$$\sum_{k=1}^n k = \underbrace{(1+n) + (2+(n-1)) + \cdots + \left(\frac{n}{2} + \left(\frac{n}{2} + 1\right)\right)}_{\frac{n}{2} \text{ pairs that sum to } n+1} = \frac{n(n+1)}{2}.$$

Now, suppose n is odd. Then $n+1$ is even. From above, we have

$$\sum_{k=1}^{n+1} k = \frac{(n+1)(n+2)}{2}.$$

Then

$$\begin{aligned}\sum_{k=1}^n k &= \left(\sum_{k=1}^{n+1} k\right) - (n+1) \\ &= \frac{(n+1)(n+2)}{2} - (n+1) \\ &= \frac{(n+1)(n+2-2)}{2} \\ &= \frac{n(n+1)}{2}.\end{aligned}$$

□

Example 2.4.29. Let's use Theorem 2.4.28 to compute the sums from Example 2.4.25.

First, let's compute $\sum_{k=0}^{100} k^2$. We break off piece that starts at $k = 1$ so we can use Theorem 2.4.28,

$$\sum_{k=0}^{100} k^2 = 0^2 + \sum_{k=1}^{100} k^2.$$

Since $0^2 = 0$, this is just $\sum_{k=1}^{100} k^2$, which matches the formula in the theorem using $n = 100$. Thus

$$\sum_{k=0}^{100} k^2 = 0 + \frac{100(100+1)(2(100)+1)}{6} = 338,350.$$

Similarly, we write

$$\sum_{k=0}^{100} k = 0 + \sum_{k=1}^{100} k.$$

We use Theorem 2.4.28 with $n = 100$ to get

$$\sum_{k=0}^{100} k = 0 + \frac{100(100+1)}{2} = 5050.$$

Finally,

$$\sum_{k=0}^{100} 1 = 1 + \sum_{k=1}^{100} 1,$$

so by Theorem 2.4.28,

$$\sum_{k=0}^{100} 1 = 1 + 100 = 101.$$

Example 2.4.30.

$$\begin{aligned} \sum_{k=1}^{500} (7k+3) &= 7 \sum_{k=1}^{500} k + \sum_{k=1}^{500} 3 \\ &= 7 \cdot \frac{500(501)}{2} + 3(500) \\ &= 878,250. \end{aligned}$$

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) sequence
 - (b) geometric progression, common ratio, initial term
 - (c) arithmetic progression, common difference, initial term

- (d) recurrence relation
- (e) solution of a recurrence relation
- (f) Fibonacci sequence
- (g) sum

2. Compute the initial term and common ratio of the following geometric progression.

$$16, 8, 4, 2, 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$$

3. Compute $\sum_{i=1}^5 i$. Simplify.

4. Compute $\sum_{i=1}^{100} (3i + 7)$. Simplify.

5. Compute $\sum_{k=50}^{100} k^2$. Simplify.

6. Compute $\sum_{k=0}^{10} 3 \cdot 2^k$. Simplify.

7. What are the terms a_1, a_2, a_3 , and a_4 of each of these sequences $\{a_n\}$?

- (a) $a_n = 1$
- (b) $a_n = \lfloor \frac{n}{2} \rfloor$
- (c) $a_n = (-1)^n$
- (d) $a_n = 2(-3)^{n+1} + 4$
- (e) $a_n = (-\frac{1}{2})^n$

8. What are the terms a_1, a_2, a_3 , and a_4 of each of these sequences $\{a_n\}$?

- (a) $a_n = 1 + (-1)^n$
- (b) $a_n = -(-3)^n$
- (c) $a_n = 3n + 4$
- (d) $a_n = 3^n - 2^n$
- (e) $a_n = \sqrt{\lfloor n \rfloor}$

9. Show that each of these sequences $\{a_n\}$ is a solution of the recurrence relation


$$a_n = -3a_{n-1} + 4a_{n-2}.$$

- (a) $a_n = 0$
- (b) $a_n = 1$
- (c) $a_n = (-4)^n$
- (d) $a_n = 2(-4)^n + 3$

10. Suppose you are hired at company in this year with a starting salary of \$50,000. Every year, you receive a 5% raise plus an additional \$1000.

- (a) Set up a recurrence relation for your salary after n years.
- (b) What is your salary in 10 years?

(c) Find an explicit formula for your salary in n years.



Number Theory and Applications

Mathematics is the queen of the sciences and number theory is the queen of mathematics.

Carl Friedrich Gauss (1777–1855)

At its core, *number theory* is the study of integers. In this chapter, we introduce modular arithmetic to explore divisibility. We discuss modern applications of number theory including ASCII encoding and encryption.

3.1. Divisibility and modular arithmetic

Goals. To introduce some fundamental concepts from number theory, including the division algorithm, congruences, and the rules of modular arithmetic.

3.1.1. Divisibility.

Definition 3.1.1. Let a and b be integers, with $a \neq 0$. We say a *divides* b , denoted $a \mid b$, if there exists an integer k such that $b = ak$. In this case, we call a a *factor* of b , and b is a *multiple* of a . We write $a \nmid b$ when a does not divide b .

Example 3.1.2.

3 divides 12: $3 \mid 12$ since $12 = 3 \cdot 4$.

3 does not divide 7: $3 \nmid 7$ since there is no integer k such that $7 = 3k$.

In other words, $\frac{7}{3}$ is not an integer so $3 \nmid 7$.

Theorem 3.1.3. *Let a , b , and c be integers with $a \neq 0$.*

- (1) *If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$.*
- (2) *If $a \mid b$, then $a \mid bc$.*
- (3) *Suppose in addition $b \neq 0$. If $a \mid b$ and $b \mid c$, then $a \mid c$.*

Proof.

- (1) Suppose $a \mid b$ and $a \mid c$. Then there exists integers m and n such that $b = am$ and $c = an$. We want to show $a \mid (b + c)$. We compute

$$b + c = am + an = a(m + n).$$

Since m and n are integers, $m + n$ is an integer. Thus $a \mid (b + c)$.

- (2) Suppose $a \mid b$. Then there exists an integer n such that $b = an$. We want to show $a \mid bc$. We compute

$$bc = (an)c = a(nc).$$

Since n and c are integers, nc is an integer. Thus $a \mid bc$.

- (3) Suppose $a \mid b$ and $b \mid c$. Then there exist integers m and n such that $b = am$ and $c = bn$. We want to show $a \mid c$. We compute

$$c = bn = (am)n = a(mn).$$

Since m and n are integers, mn is an integer. Thus $a \mid mn$.

□

Corollary 3.1.4. *Let $a, b, c \in \mathbb{Z}$ with $a \neq 0$. If $a \mid b$ and $a \mid c$, then $a \mid (sb + tc)$ for all $s, t \in \mathbb{Z}$.*

Theorem 3.1.5 (Division algorithm). *Let a and d be integers with $d > 0$. There exists unique integers q and r with $0 \leq r < d$ such that $a = dq + r$.*

Remark 3.1.6. Given a and d , we can compute q as the floor $\lfloor \frac{a}{d} \rfloor$. After we have q , we can solve for r in the equation $a = dq + r$.

Example 3.1.7. Let's see what the division algorithm says in a few examples.

First consider $a = 253$ and $d = 17$. By long division, we see that $253/17 \approx 14.88$. Thus $q = \lfloor \frac{253}{17} \rfloor = 14$. Then $253 = 17 \cdot 14 + r$. Solving for r gives $r = 15$.

Next consider $a = -253$ and $d = 17$. By long division, we see $-253/17 \approx -14.88$. Thus $q = \lfloor \frac{-253}{17} \rfloor = -15$. Then $-253 = 17(-15) + r$. Solving for r gives $r = 2$.

Definition 3.1.8. Let a and d be integers, with $d > 0$. Let q and r be integers such that $a = dq + r$, with $0 \leq r < d$. We call d the *divisor*, and we call a the *dividend*. We call q the *quotient*, and denote it $a \operatorname{div} d$. We call r the *remainder*, and denote it $a \operatorname{mod} d$.

Remark 3.1.9. The definition of the quotient and remainder of a divided by d is well-defined because of the Division algorithm, which implies that the a and r are uniquely determined by those conditions.

We denote the quotient and remainder by

$$q = a \operatorname{div} d \quad \text{and} \quad r = a \operatorname{mod} d.$$

Python Code Snippet 3.1.10. In Python, the quotient and remainder can be computed as follows. Note: In Python, there is a function that returns both the quotient and remainder at once.

```
q = a // d
r = a % d
# or compute them both at the same time
q, r = divmod(a,d)
```

Example 3.1.11. $17 \operatorname{div} 3 = 5$ and $17 \operatorname{mod} 3 = 2$ since $17 = 5 \cdot 3 + 2$.

Example 3.1.12. $-17 \operatorname{div} 3 = -6$ and $-17 \operatorname{mod} 3 = 1$ since $17 = -6 \cdot 3 + 1$.

3.1.2. Modular arithmetic.

Definition 3.1.13. Let a , b , and m be integers, with $m > 0$. We say a is *congruent* to b modulo m , denoted $a \equiv b \pmod{m}$, if $m \mid (a - b)$.

Remark 3.1.14. This notion is different (but related) to the mod representing the remainder from the division algorithm. For example, if $a = b \operatorname{mod} m$, then a is an integer with $0 \leq a < m$. In particular, $b \operatorname{mod} m$ is an integer. On the other hand, $a \equiv b \pmod{m}$ asserts a relationship between a and b . We never write $b \pmod{m}$ by itself.

Theorem 3.1.15. Let $a, b, m \in \mathbb{Z}$, with $m > 0$. The following are equivalent.

- (1) $a \equiv b \pmod{m}$.
- (2) There exists $k \in \mathbb{Z}$ such that $a = b + km$.
- (3) $a \operatorname{mod} m = b \operatorname{mod} m$.

Proof. It is enough to prove $1 \rightarrow 2$, $2 \rightarrow 3$, and $3 \rightarrow 1$.

$1 \rightarrow 2$: Suppose $a \equiv b \pmod{m}$. Then $m \mid (a - b)$. Then there exists an integer k such that $a - b = mk$. Solving for a , we have $a = b + km$ as desired.

2 \rightarrow 3: Suppose there exists $k \in \mathbb{Z}$ such that $a = b + km$. Using the Division algorithm, we write $a = qm + r$ and $b = q'm + r'$, for some integers q, q', r, r' with $0 \leq r, r' < m$. We want to show that $r = r'$. We compute

$$a = b + km = (q'm + r') + km = (q' + k)m + r'.$$

By the uniqueness of the integers in the Division algorithm, we must have $q' + k = q$ and $r' = r$.

3 \rightarrow 1: Suppose $a \bmod m = b \bmod m$. Let $r = a \bmod m = b \bmod m$. Then by the Division algorithm, there exist integers q and q' such that $a = qm + r$ and $b = q'm + r$. We compute

$$a - b = (qm + r) - (q'm + r) = m(q - q').$$

Since q and q' are integers, we have $m \mid (a - b)$. Thus $a \equiv b \pmod{m}$.

□

Definition 3.1.16. The set of integers that are congruent to a modulo m is called the *congruence class* of a modulo m , and is denoted $[a]$. In particular,

$$[a] = \{a + mk \mid k \in \mathbb{Z}\}.$$

Example 3.1.17. The congruence class of 3 modulo 7 is

$$[3] = \{3 + 7k \mid k \in \mathbb{Z}\} = \{\dots, -11, -4, 3, 10, 17, \dots\}$$

This is equal to the congruence class of 10 since

$$[10] = \{10 + 7k \mid k \in \mathbb{Z}\} = \{\dots, -11, -4, 3, 10, 17, \dots\}$$

In fact, the congruence class of 3 is equal to the congruence class of any integer that is congruent to 3 modulo 7,

$$\dots = [-4] = [3] = [10] = [17] = \dots$$

Example 3.1.18. There are five congruence classes modulo 5,

$$[0] = \{0 + 5k \mid k \in \mathbb{Z}\} = \{\dots, -10, -5, 0, 5, 10, \dots\}$$

$$[1] = \{1 + 5k \mid k \in \mathbb{Z}\} = \{\dots, -9, -4, 1, 6, 11, \dots\}$$

$$[2] = \{2 + 5k \mid k \in \mathbb{Z}\} = \{\dots, -8, -3, 2, 7, 12, \dots\}$$

$$[3] = \{3 + 5k \mid k \in \mathbb{Z}\} = \{\dots, -7, -2, 3, 8, 13, \dots\}$$

$$[4] = \{4 + 5k \mid k \in \mathbb{Z}\} = \{\dots, -6, -1, 4, 9, 14, \dots\}.$$

The following result says that we can define arithmetic on congruence classes.

Theorem 3.1.19. Let $a, \hat{a}, b, \hat{b}, m \in \mathbb{Z}$, with $m > 0$. If $a \equiv \hat{a} \pmod{m}$ and $b \equiv \hat{b} \pmod{m}$, then

- (1) $a + b \equiv \hat{a} + \hat{b} \pmod{m}$;
 (2) $ab \equiv \hat{a}\hat{b} \pmod{m}$.

Proof. Suppose $a \equiv \hat{a} \pmod{m}$ and $b \equiv \hat{b} \pmod{m}$. Then there exist integers k and ℓ such that $a = \hat{a} + km$ and $b = \hat{b} + \ell m$. We compute

$$a + b = (\hat{a} + km) + (\hat{b} + \ell m) = (\hat{a} + \hat{b}) + m(k + \ell).$$

Since k and ℓ are integers, $k + \ell$ is an integer. Thus $a + b \equiv \hat{a} + \hat{b} \pmod{m}$ by Theorem 3.1.15.

Similarly, we compute

$$ab = (\hat{a} + km)(\hat{b} + \ell m) = (\hat{a}\hat{b}) + m(k\hat{b} + \ell\hat{a} + k\ell m).$$

Since $\hat{a}, \hat{b}, k, m,$ and ℓ are integers, $(k\hat{b} + \ell\hat{a} + k\ell m)$ is an integer. Thus $ab \equiv \hat{a}\hat{b} \pmod{m}$ by Theorem 3.1.15. \square

The result says that we can take any element in the congruence class of a and add it to any element in the congruence class of b , and we will get an element of the congruence class of $a + b$, and the analogous result for multiplication. The division algorithm ensures that each congruence class modulo m will contain a unique representative $0 \leq a < m$, so we can transfer the arithmetic on congruence classes to an arithmetic on congruence class representatives.

Example 3.1.20. Suppose $a \equiv 5 \pmod{10}$ and $b \equiv 2 \pmod{10}$. Then

$$a^2 - b \equiv 5^2 - 2 \equiv 23 \equiv 3 \pmod{10}.$$

Definition 3.1.21. Let m be a positive integer. The *integers mod m* , denoted \mathbb{Z}_m , as a set is

$$\mathbb{Z}_m = \{0, 1, \dots, m - 1\}.$$

We endow this set with arithmetic. For a and b in \mathbb{Z}_m , we define the sum and product by

$$a +_m b = (a + b) \bmod m \quad \text{and} \quad a \times_m b = (ab) \bmod m.$$

Example 3.1.22. We compute the addition and multiplication table for \mathbb{Z}_6 .

$+_6$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

\times_6	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

3.1.3. Primitive roots.

Definition 3.1.23. A *primitive root* modulo a prime p is an integer r in \mathbb{Z}_p such that every nonzero element of \mathbb{Z}_p is a power of r .

Example 3.1.24. Find a primitive root modulo 7.

We need to find an element r in \mathbb{Z}_7 such that every element in \mathbb{Z}_7 is expressible as a power of r . Rather than figuring out which power to use to represent each element of \mathbb{Z}_7 , it is easier to just compute powers of r and see which elements of \mathbb{Z}_7 they represent.

We compute in \mathbb{Z}_7 .

$$3^0 = 1$$

$$3^1 = 3$$

$$3^2 = 3^1 \times_m 3 = 3 \times_m 3 = 2$$

$$3^3 = 3^2 \times_m 3 = 2 \times_m 3 = 6$$

$$3^4 = 3^3 \times_m 3 = 6 \times_m 3 = 4$$

$$3^5 = 3^4 \times_m 3 = 4 \times_m 3 = 5$$

This shows $1 = 3^0$, $2 = 3^2$, $3 = 3^1$, $4 = 3^4$, $5 = 3^5$, and $6 = 3^3$. Since $\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$, we have represented all the nonzero elements of \mathbb{Z}_7 as powers of 3. Thus 3 is a primitive root modulo 7.

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) divides
 - (b) factor
 - (c) multiple
 - (d) divisor
 - (e) dividend

-
- (f) quotient
 - (g) remainder
 - (h) two integers are congruent modulo an integer
 - (i) congruence class of an integer modulo an integer
 - (j) \mathbb{Z}_m
 - (k) primitive root
2. State precisely the Division algorithm. Be sure to set up any notation that is required.
 3. Does 13 divide these numbers? Justify.
 - (a) 39
 - (b) -26
 - (c) 0
 - (d) 1
 - (e) 57
 4. What is the quotient and remainder for these?
 - (a) 44 divided by 7
 - (b) -123 divided by 12
 - (c) 253 divided by 15
 - (d) 0 divided by 17
 - (e) -100 divided by 100
 5. Suppose a and b are integers such that $a \equiv 11 \pmod{17}$, and $b \equiv 3 \pmod{17}$. Find the integer c with $0 \leq c \leq 16$ such that satisfies these.
 - (a) $c \equiv 13a \pmod{17}$
 - (b) $c \equiv 5a \pmod{17}$
 - (c) $c \equiv -2a + 3b \pmod{17}$
 - (d) $c \equiv ab \pmod{17}$
 - (e) $c \equiv a^3 - b^2 \pmod{17}$
 6. Evaluate these quantities.
 - (a) $-15 \pmod{6}$
 - (b) $-2 \pmod{12}$
 - (c) $124 \pmod{7}$
 - (d) $244 \pmod{24}$
 - (e) $1245 \pmod{3}$
 7. Find a primitive root modulo 11.
 8. What is the congruence class of 5 modulo 7?
 9. Compute the addition and multiplication table for \mathbb{Z}_5 .
 10. Let a, b, c be integers, where $a, b \neq 0$. Follow the steps below to prove that if $a \mid b$ and $b \mid c$, then $a \mid c$.
 - (a) First, we recognize what we want to prove is in the form “if p , then q .” What is hypothesis p ? What is the conclusion q ?
 - (b) When we want to prove a statement of the form “if p , then q ” directly, we assume p is true and try to show q . This is commonly where we

- set some notation as well. e.g., **Suppose a, b, c are integers such that $a \mid b$ and $b \mid c$.**
- (c) Next we need to recall what $x \mid y$ or *divides* means. It should have something to do with the existence of an integer. We say $x \mid y$ or x *divides* y if there exists an integer k such that . . .
- (d) Now apply the definition to our situation where we have $a \mid b$ and $b \mid c$. e.g., **Then, there exist(s)**
- (e) Check back above to be sure that the two integers whose existence is guaranteed have different names. They need different names because they need not be the same integer.
- (f) Now look back to the goal q that we set above. We should have something like “ $a \mid c$ ”. Use the definition to re-express this goal. **We want to show there exists an integer . . . such that**
- (g) Now use the two equations from (d) to produce the integer whose existence we want from (f). Use equations or complete sentences. Don’t just write sentence fragments of expressions.
- (h) Draw the conclusion.
11. Follow the steps below to prove that $n^3 - n$ is divisible by 3 for every integer n .
- (a) Remember that divisibility statements can be rewritten as congruence conditions. e.g., a is divisible by b can be written as $a \equiv 0 \pmod{b}$. What is the congruence condition for this problem?
- (b) Rewrite the above, alerting the proof reader that our statement is equivalent to the desired statement. Be sure to specify that we will show it for every integer n . e.g., **It suffices to show . . . for every . . .**
- (c) When we want to prove a universal statement, we can fix a generic one to consider. e.g., **Let n be an integer.**
- (d) Turn this into a finite check by looking at each congruence class separately. e.g., **There are . . . cases to consider.**
- (e) For each case, write down the assumption that n is in the congruence class under consideration. Use properties of modular arithmetic from class to prove the result in that case. Repeat for each case.
12. Let a, b, c, d, m be integers, with $m > 1$. Follow the steps below to prove that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$.
- (a) First, recognize what we want to prove is in the form “if p , then q .” What is p ? What is q ?
- (b) As before, to prove a statement of the form “if p , then q ” directly, we assume p is true and try to show q . This is commonly where we set some notation as well. **Suppose We want to show**
- (c) Above are some statements about congruences. Use the definition or theorems to state our assumption and our goal in terms of divisibility or existence of integers.

- (d) Use arithmetic to get from the assumptions to the goal.
 (e) Draw the conclusion.
13. Prove that if n is an odd, positive integer, then $n^2 \equiv 1 \pmod{8}$.
14. Find a counterexample to the following statement about congruences.
 Let a, b, c , and m be integers with $m > 2$. If $ac \equiv bc \pmod{m}$, then $a \equiv b \pmod{m}$.

3.2. Integer representations and applications

Goals. To study representations of integers in different bases, including binary and hexadecimal representations, and to introduce algorithms and applications involving integers based on these representations.

3.2.1. Integer representations. We typically use *decimal* (base 10) notation to represent integers. e.g., 253 means $2 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0$. This is most likely due to the fact that we have 10 fingers. There is no good mathematical reason to use base 10, and in fact we can use any integer $b > 1$ as a base.

Working with other bases is important. By abstracting to a general base b , we find that we understand the usual decimal operations better. Working in other bases also has applications. For example, *binary* (base 2) expansions are used to develop fast exponentiation techniques. This is a vital component of modern computer encryption schemes necessary for today's digital world. Without secure means of communication, things such as online shopping would be impossible. We use base 256 for ASCII encoding, a character encoding standard for electronic communication.

Theorem 3.2.1. Let b be an integer greater than 1. If n is a positive integer, it can be expressed uniquely in the form

$$n = \sum_{i=0}^k a_i b^i,$$

where k is a non-negative integer, the a_i are all non-negative integers strictly less than b , and $a_k \neq 0$.

Definition 3.2.2. The expression $\sum_{i=0}^k a_i b^i$ of the theorem is known as a *base b expansion*, denoted $(a_k a_{k-1} \dots a_1 a_0)_b$.

Example 3.2.3.

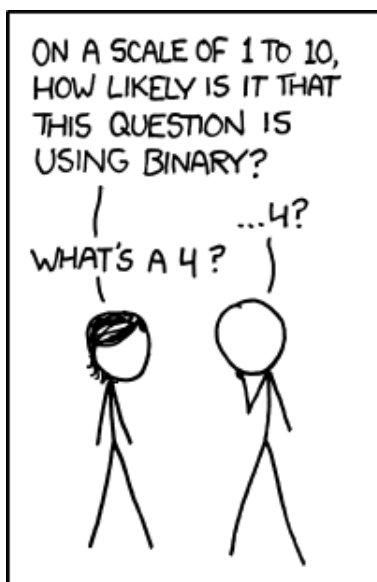


Figure 3.2.1. xkcd: 1 to 10. (<https://xkcd.com/953/>) If you get an 11/100 on a CS test, but you claim it should be counted as a ‘C’, they’ll probably decide you deserve the upgrade.

- $(253)_{10} = 2 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0 = 253$.
- $(10)_2 = 1 \cdot 2^1 + 0 \cdot 2^0 = 2$.

For $b > 10$, we move to other symbols to represent the digits. For example, in *hexadecimal* (base 16), the digits are

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.$$

Example 3.2.4.

$$\begin{aligned} (1FAD)_{16} &= 1 \cdot 16^3 + F \cdot 16^2 + A \cdot 16^1 + D \cdot 16^0 \\ &= 1 \cdot 16^3 + 15 \cdot 16^2 + 10 \cdot 16^1 + 13 \cdot 16^0 \\ &= 8109. \end{aligned}$$

How do we go the other way? Namely, given an integer n in decimal form and a positive base b , how can we compute the base b -expansion of n ?

We can compute the base b expansion using div and mod. See the following examples. The first writes out the steps. The second is more condensed, given as a table.

Example 3.2.5. Compute the *binary* (base 2) expansion of 67.

$$\begin{aligned} 67 &= 33 \cdot 2 + \boxed{1} \\ 33 &= 16 \cdot 2 + \boxed{1} \\ 16 &= 8 \cdot 2 + \boxed{0} \\ 8 &= 4 \cdot 2 + \boxed{0} \\ 4 &= 2 \cdot 2 + \boxed{0} \\ 2 &= 1 \cdot 2 + \boxed{0} \\ 1 &= 0 \cdot 2 + \boxed{1}. \end{aligned}$$

The boxed terms give the base 2 expansion of 67 as

$$67 = (1000011)_2.$$

Python Code Snippet 3.2.6. Below is the Python code to compute the base b expansion of n .

```
def base_expansion(n,b):
    '''
    Return the digits of the base b expansion of n.
    [a_k, ..., a_1, a_0], where a_k b^k + ... + a_0 = n.
    '''
    q = n
    digit_list = [] # used to store digits a_0, ..., a_k
    while q != 0:
        q, r = divmod(q,b)
        digit_list.append(r)
    # now reverse the digits so that a_k is first
    digit_list.reverse()
    return digit_list
```

With the algorithm, here is a more condensed solution.

Example 3.2.7. Compute the base 5 expansion of 325.

i	q	a_i
0	325	0
1	65	0
2	13	3
3	2	2
4	0	—

Thus, we have $325 = (2300)_5$.

Similarly, one can compute that $325 = (505)_8$ and $325 = (101000101)_2$.

3.2.2. Text encoding. ASCII is a standard way to represent characters as numbers. For example, a space is represented by 32, a comma is 44, and a period is 46. The capital letters are also 2 digit integers, starting with 65 for A and going to 90 for Z. The Python functions `chr` and `ord` to convert the ASCII to characters. e.g., `chr(66)` returns the string A. If we want to go the other way, `ord('A')` returns the integer 65. This is known as *encoding*.

In order to encode messages longer than one character, we will view each number as a digit in a base 256 expansion of an integer M .

Example 3.2.8. Suppose I want to encode the message `Help!` using ASCII. We have

$$\text{ord}(H) = 72, \quad \text{ord}(e) = 101, \quad \text{ord}(l) = 108, \quad \text{ord}(p) = 112, \quad \text{ord}(!) = 33,$$

so the encoded message is

$$M = (72, 101, 108, 112, 33)_{256}.$$

That means

$$M = 72 \cdot 256^4 + 101 \cdot 256^3 + 108 \cdot 256^2 + 112 \cdot 256^1 + 33 \cdot 256^0 = 310,939,250,721.$$

Remark 3.2.9. ASCII allows use to convert messages written as strings into sequences of integers. We take this a step further interpreting the sequence as the base 256 expansion of a single integer. We will use this later when we look at RSA encryption §3.5.2.

Example 3.2.10. We can *decode* a message by computing the base 256 expansion of the encoded message and decoding each character. For example, suppose the encoded message is $M = 310,939,249,775$. The base 256 expansion of M is

$$M = (72, 101, 108, 108, 111)_{256}.$$

Then

$$\text{chr}(72) = H, \quad \text{chr}(101) = e, \quad \text{chr}(108) = l, \quad \text{chr}(108) = l, \quad \text{chr}(111) = o,$$

so the decoded message is 'Hello'.

Watch the video

<https://youtu.be/23J2doHaJ6U>

for additional details about implementation.

3.2.3. Fast exponentiation. Let b , n , and m be positive integers. For cryptographic applications, it is important to compute $b^n \bmod m$ efficiently. Note that in theory, the computation is easy. For example, we can compute

$$b^n = \underbrace{b \cdot b \cdots b}_{n \text{ times}}.$$

Then divide the result by m to get the remainder. There are at least two problems with this in practice. The first is that computing b^n in this way requires $n - 1$ multiplications. If n is large, this is slow. We can get around this using *fast exponentiation*. We will develop the algorithm by first looking at an example.

Example 3.2.11. Compute 3^{11} .

First, we compute the binary expansion of $11 = (1011)_2$. Then

$$3^{11} = 3^{2^3+2^1+1} = 3^8 \cdot 3^2 \cdot 3.$$

Notice that we can compute 3^8 as $(3^4)^2$. Similarly, $3^4 = (3^2)^2$. We have by repeated squarings,

$$\begin{aligned} 3 &= 3 \\ 3^2 &= 9 \\ 3^4 &= 9^2 = 81 \\ 3^8 &= 81^2 = 6561. \end{aligned}$$

Thus $3^{11} = 6561 \cdot 9 \cdot 3 = 177,147$. In this computation, we cut the number of required multiplications in half. For larger exponents n , the savings are even more extreme.

We can save even more time when we only care about the remainder modulo some integer.

Example 3.2.12. Compute $3^{11} \bmod 7$.

As in Example 3.2.11, we will use the square and multiply technique to get the exponentiation faster. As before, we compute the binary expansion of $11 = (1011)_2$. Then

$$3^{11} \bmod 7 = 3^{2^3+2^1+1} \bmod 7 = (3^8 \cdot 3^2 \cdot 3) \bmod 7.$$

We make additional time and memory savings by keeping the integers small. We achieve this by reducing mod 7 at every stage, since we only want the result modulo 7.

We have by repeated squarings,

$$\begin{aligned} 3 \bmod 7 &= 3 \\ 3^2 \bmod 7 &= 2 \\ 3^4 \bmod 7 &= 2^2 \bmod 7 = 4 \\ 3^8 \bmod 7 &= 4^2 \bmod 7 = 2. \end{aligned}$$

Thus $3^{11} \bmod 7 = 2 \cdot 2 \cdot 3 \bmod 7 = 5$.

The **fast exponentiation** algorithm allows one to efficiently compute $a^n \bmod m$. We give a slightly different formulation here than the standard that is easier to remember for computations by hand.

Theorem 3.2.13 (Fast exponentiation). *Given positive integers a , n , and m , with $m > 1$, compute $a^n \bmod m$ efficiently by following these steps.*

- (1) Compute the base 2 expansion of the exponent n

$$n = (a_k a_{k-1} \dots a_1 a_0)_2.$$

- (2) Make a list of successive squares p_i , where $p_0 = a \bmod m$ and $p_i = p_{i-1}^2 \bmod m$ for $1 < i \leq k$.

- (3) Compute

$$p_0^{a_0} p_1^{a_1} \dots p_k^{a_k} \bmod m,$$

while remembering the following.

- Keep reducing modulo m when computing the product in order to keep the numbers small.
- When $a_i = 0$, the term $p_i^{a_i} = 1$ and so does not contribute to the product.

Example 3.2.14. Let's compute $3^{67} \bmod 253$.

First we compute the base 2 expansion of $67 = (1000011)_2$ as in Example 3.2.5. Note that this is seven digits ($k = 6$), so we need to compute the seven successive squares p_0, \dots, p_6 .

$$p_0 = 3 \bmod 253 = 3$$

$$p_1 = 3^2 \bmod 253 = 9$$

$$p_2 = 9^2 \bmod 253 = 81$$

$$p_3 = 81^2 \bmod 253 = 236$$

$$p_4 = 236^2 \bmod 253 = 36$$

$$p_5 = 36^2 \bmod 253 = 31$$

$$p_6 = 31^2 \bmod 253 = 202.$$

We summarize the computations so far.

i	a_i	p_i
0	1	$3 \bmod 253 = \boxed{3}$
1	1	$3^2 \bmod 253 = \boxed{9}$
2	0	$9^2 \bmod 253 = 81$
3	0	$81^2 \bmod 253 = 236$
4	0	$236^2 \bmod 253 = 36$
5	0	$36^2 \bmod 253 = 31$
6	1	$31^2 \bmod 253 = \boxed{202}$

The terms corresponding to a 1 in the binary expansion of 67 have been boxed. It is these terms that we multiply together. It follows that

$$3^{67} \bmod 253 = 3 \cdot 9 \cdot 202 \bmod 253 = 141.$$

Python Code Snippet 3.2.15. The Python code given below shows the general algorithm for fast exponentiation. It utilizes the `base_expansion` code above.

```
def fast_power_mod(b, n, m):
    '''
    Return b^n mod m.
    '''
    x = 1 # initialize answer
    power = b % m # initialize power
    digit_list = base_expansion(n, 2)
    for digit in digit_list[::-1]:
        if digit == 1:
            x = (x * power) % m
            power = (power * power) % m
    return x
```

Let's compute another fast exponentiation $x = b^n \bmod m$ example, but using the algorithm in Python Code Snippet 3.2.15. The main difference is that instead of multiplying the relevant values together at the end, we keep track and update the value of x as the computation progresses.

Example 3.2.16. Let's compute $3^{67} \bmod 253$ again. Compare with Example 3.2.14, but this time we will use the algorithm described in Python Code Snippet 3.2.15.

As before, we have compute the base 2 expansion of $67 = (1000011)_2$. We compute p_i by squaring p_{i-1} and reducing mod 253. We initialize $p_0 = 3 \bmod 253 = 3$ and $x = 1$. Whenever the i th digit in the binary expansion of 22 is a 1, we update x by multiplying x by p_i . These rows are marked with a left arrow (\leftarrow) below. The result is the last value of x . Thus $3^{67} \bmod 253 = 141$.

i	a_i	p_i	x
0	1	$3 \bmod 253 = 3$	$1 \cdot 3 \bmod 253 = 3$ \leftarrow
1	1	$3^2 \bmod 253 = 9$	$3 \cdot 9 \bmod 253 = 27$ \leftarrow
2	0	$9^2 \bmod 253 = 81$	27
3	0	$81^2 \bmod 253 = 236$	27
4	0	$236^2 \bmod 253 = 36$	27
5	0	$36^2 \bmod 253 = 31$	27
6	1	$31^2 \bmod 253 = 202$	$27 \cdot 202 \bmod 253 = \boxed{141}$ \leftarrow

3.2.4. Casting out nines.

Theorem 3.2.17. *A positive integer is congruent to the sum of its decimal digits modulo 9.*

Proof. Let the decimal expansion of the integer N be given by

$$N = (a_n a_{n-1} \dots a_1 a_0)_{10}$$

so that

$$N = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10 + a_0.$$

It follows immediately that

$$N \equiv a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10 + a_0 \pmod{9}.$$

The next key fact is that $10 \equiv 1 \pmod{9}$, and so

$$10^k \equiv 1^k \equiv 1 \pmod{9}, \quad \text{for all } k.$$

Since $10^k \equiv 1 \pmod{9}$, we have

$$N \equiv a_n + a_{n-1} + \dots + a_3 + a_2 + a_1 + a_0 \pmod{9}.$$

In other words, N is congruent to the sum of its digits modulo 9. \square

Remark 3.2.18. This result is the reason the *casting out nines* technique for checking arithmetic works. See the video *Casting Out Nines - Numberphile* for additional details.

<https://youtu.be/FlndIiQa20o>

Since an integer is divisible by 9 if and only if it is congruent to 0 modulo 9, Theorem 3.2.17 immediately gives the following divisibility criterion.

Corollary 3.2.19 (Divisibility by nine). *A positive integer is divisible by 9 if and only if the sum of its decimal digits is divisible by 9.*

Proof. Therefore $N \equiv 0 \pmod{9}$ if and only if

$$a_n + a_{n-1} + \dots + a_1 + a_0 \equiv 0 \pmod{9}.$$

Since being divisible by 9 is the same as being congruent to 0 (mod 9), we have proved that a positive integer is divisible by 9 if and only if the sum of its decimal digits is divisible by 9. \square

A similar argument gives divisibility rules for 3 and 11.

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) base b expansion
 - (b) binary
 - (c) decimal
 - (d) hexadecimal
2. Compute the base 5 expansion of 253.
3. Watch the following videos
 - *How to count to 1000 on two hands* posted by 3Blue1Brown
<https://youtu.be/1SMmc9gQmHQ>
 - *How high can you count on your fingers? (Spoiler: much higher than 10) - James Tanton* posted by TED-Ed
<https://youtu.be/UixU1oRW64Q>and learn to count in binary on your hands.
4. Compute the integer representing the ASCII encoding of these messages.
 - (a) apple
 - (b) Radiohead
 - (c) discrete math
 - (d) secret
5. These integers represent messages encoded in ASCII as described in §3.2.2. Decode these messages.
 - (a) 6582119
 - (b) 280991720293
 - (c) 311107740793
 - (d) 22107779118197813113556726561
 - (e) The following is an integer that is too long to fit in a line.
1971486178880874921204775823582727213754122745
127854714387549425398830
6. Compute $12^{321} \bmod 456$. Follow the steps below if you get stuck.
 - (a) Compute the base 2 expansion of 321.
 - (b) Check that you got $321 = (101000001)_2$.
 - (c) Compute the successive powers p_i , remembering to reduce mod 456.
Did you notice anything that helps this computation go faster?
 - (d) Multiply the correct terms together to compute $12^{321} \bmod 456$.
7. Compute $7^{447} \bmod 645$.
8. Compute $3^{447} \bmod 645$.
9. Convert the decimal expansion of each of these integers to a binary expansion. i.e., Convert base 10 to base 2.

- (a) 1
 - (b) 156
 - (c) 765
 - (d) 23
 - (e) 245
10. Convert the decimal expansion of each of these integers to a hexadecimal expansion. i.e., Convert base 10 to base 16.
- (a) 123
 - (b) 45326
 - (c) 12
 - (d) 157
 - (e) 149987
11. Convert the binary expansion of each of these integers to a decimal expansion. i.e., Convert base 2 to base 10.
- (a) $(10011)_2$
 - (b) $(111)_2$
 - (c) $(101010)_2$
 - (d) $(111000)_2$
 - (e) $(1011011)_2$
12. Convert the hexadecimal expansion of each of these integers to a decimal expansion. i.e., Convert base 16 to base 10.
- (a) $(A123B)_{16}$
 - (b) $(81C)_{16}$
 - (c) $(ABBA)_{16}$
 - (d) $(DA3)_{16}$
 - (e) $(253)_{16}$
13. Prove that a positive integer is divisible by 5 if and only if the last digit is divisible by 5.
14. Prove that a positive integer is divisible by 3 if and only if the sum of its decimal digits is divisible by 3.
15. Prove that a positive integer is divisible by 11 if and only if the alternating sum of its decimal digits is divisible by 11.
-

3.3. Primes and greatest common divisors

Goals. To introduce some fundamental concepts from number theory, including primality, prime factorization, and greatest common divisors. To introduce some important conjectures about primes.

3.3.1. Primes.

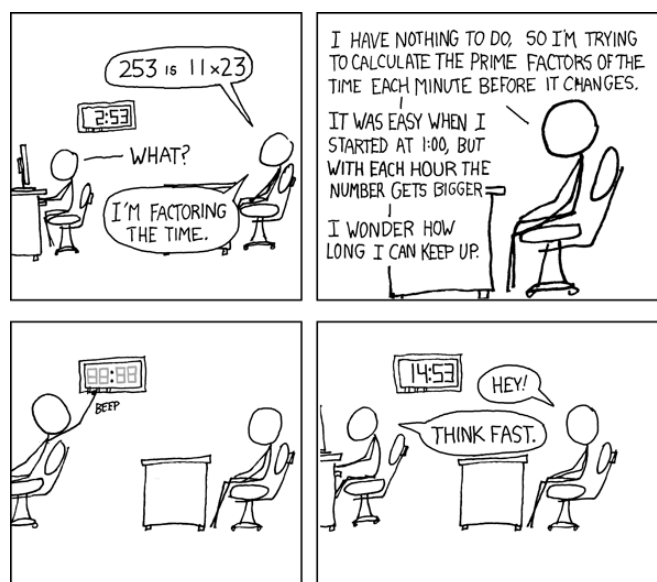


Figure 3.3.1. xkcd: Factoring the Time. (<https://xkcd.com/247/>) I occasionally do this with mile markers on the highway.

Definition 3.3.1. An integer $p > 1$ is *prime* if the only positive factors of p are 1 and p . A positive integer greater than 1 that is not prime is called *composite*.

Remark 3.3.2. In later courses, we extend the notion of prime to all integers. For simplicity, in this course we restrict the prime versus composite distinction to integers greater than 1. Even in the extended notion, 1 is *not* prime.

Example 3.3.3. The first few primes are

$$2, 3, 5, 7, 11, 13, 17, \dots$$

Theorem 3.3.4 (Fundamental Theorem of Arithmetic). *Every integer that is greater than 1 can be written uniquely as a prime or a product of two or more primes, where the primes are written in order of non-decreasing size.*

Example 3.3.5.

$$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 5^2$$

$$253 = 11 \cdot 13$$

$$7007 = 7 \cdot 7 \cdot 11 \cdot 13 = 7^2 \cdot 11 \cdot 13$$

$$23,498,357,349 = 3 \cdot 53 \cdot 397 \cdot 372,263$$

The following Theorem says that if an integer is composite, it must have a “small” prime factor, where “small” means less than or equal to the square root of the number.

Theorem 3.3.6. *If n is composite, then n has a prime factor less than or equal to \sqrt{n} .*

Proof. Let n be composite. Then n can be factored as $n = ab$. If a and b are both greater than \sqrt{n} , then $n = ab > \sqrt{n}\sqrt{n} = n$. Contradiction. Thus n has a divisor less than or equal to \sqrt{n} . Then by the Fundamental Theorem of Arithmetic, that divisor is either prime, or has a prime divisor less than or equal to \sqrt{n} . \square

Theorem 3.3.6 can be used to prove primality of certain integers. The theorem says that if an integer is composite, then it must have a small prime factor. The contrapositive says that if an integer does not have a small prime factor, then it must be prime.

Example 3.3.7. Prove that 523 is prime.

Proof. Theorem 3.3.6 says that if 523 is composite, it will have a prime factor less than $\sqrt{523}$. Since $\sqrt{523} \approx 22.9$, it suffices to show that no prime less than 22 divides 523. That means we need to check 2, 3, 5, 7, 11, 13, 17, 19. A quick computation shows that

$$\begin{aligned} 523 \bmod 2 &= 1 \\ 523 \bmod 3 &= 1 \\ 523 \bmod 5 &= 3 \\ 523 \bmod 7 &= 5 \\ 523 \bmod 11 &= 6 \\ 523 \bmod 13 &= 3 \\ 523 \bmod 17 &= 13 \\ 523 \bmod 19 &= 10. \end{aligned}$$

Thus 523 is prime. \square

We can use the *Sieve of Eratosthenes* to find the primes up to some bound. This is a simple, ancient algorithm for finding all prime numbers up to any given limit. See the WIKIPEDIA page for a nice animation.

https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

The sieve works by iteratively marking as composite (i.e., not prime) the multiples of each prime, starting with the first prime number, 2. See Figure 3.3.1 for an example using the sieve to find the primes up to 49.

Step 1: Numbers from 2 ... 49.

	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Step 2: Eliminated multiples of 2.

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	32	33	34	35	
36	37	38	39	40	41	42	
43	44	45	46	47	48	49	

Step 3: Eliminated multiples of 3.

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2, 3
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	32	33	34	35	
36	37	38	39	40	41	42	
43	44	45	46	47	48	49	

Step 4: Eliminated multiples of 5.

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2, 3, 5
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	32	33	34	35	
36	37	38	39	40	41	42	
43	44	45	46	47	48	49	

Step 5: Eliminated multiples of 7.

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2, 3, 5, 7
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	32	33	34	35	
36	37	38	39	40	41	42	
43	44	45	46	47	48	49	

Step 6: Remaining are prime.

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2, 3, 5, 7,
15	16	17	18	19	20	21	11, 13, 17,
22	23	24	25	26	27	28	19, 23, 29,
29	30	31	32	33	34	35	31, 37, 41,
36	37	38	39	40	41	42	43, 47
43	44	45	46	47	48	49	

Figure 3.3.2. Sieve of Eratosthenes.

Through sieving, we could enumerate the primes up to a bound N . This number is denoted $\pi(N)$. Only 5.0848% of the positive integers less than 1,000,000,000 are prime. See Table 3.3.1 for additional data. It looks like the frequency with which primes occur is diminishing. It might be reasonable to guess that eventually prime numbers become so rare that beyond a certain

Table 3.3.1. Number of primes up to bound.

bound (N)	# of primes ($\pi(N)$)	percentage ($\frac{\pi(N)}{N}$ %)
10	4	40.0000%
100	25	25.0000%
1000	168	16.8000%
10,000	1229	12.2900%
100,000	9592	9.5920%
1,000,000	78,498	7.8498%
10,000,000	664,579	6.6458%
100,000,000	5,761,455	5.7615%
1,000,000,000	50,847,534	5.0848%

bound, they no longer occur. This is not the case, as was known to the Greeks over 2000 years ago. The first known proof is due to Euclid (c. 300 BC). See Theorem 3.3.8. It turns out, the proportion $\frac{\pi(N)}{N}$ of integers up to N that are prime does decrease, but at an ever decreasing rate. If you are interested, you can read more about the *Prime Number Theorem* and its fascinating history.

<http://mathworld.wolfram.com/PrimeNumberTheorem.html>

Theorem 3.3.8 (Infinitude of primes). *There are infinitely many primes.*

Proof. We use proof by contradiction. Suppose there are finitely many primes. Label them as p_1, p_2, \dots, p_n . Consider the integer $q = p_1 p_2 \dots p_n + 1$. By the Fundamental Theorem of Arithmetic, q is prime or can be expressed as a product of two or more primes. Since $q \bmod p_i = 1$ for $i = 1, 2, \dots, n$, we have that q is not divisible by any prime. Thus q is prime. This gives the desired contradiction since q is a prime that is not on our list. Therefore, there are infinitely many primes. \square

Though we were able to prove there are infinitely many primes, there are several open questions about primes.

Definition 3.3.9. *Twin primes* are pairs of primes that differ by 2.

For example, 3 and 5, 5 and 7, 11 and 13, \dots . It is conjectured that there are infinitely many twin primes.

Conjecture 3.3.10 (Twin prime conjecture). *There are infinitely many primes p such that $p + 2$ is also prime.*

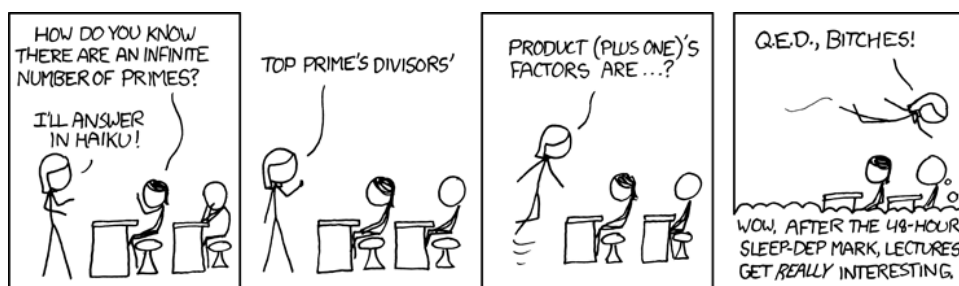


Figure 3.3.3. xkcd: Haiku Proof. (<https://xkcd.com/622/>) After somewhere around 40 hours, there's no academic reason to go to the class. Only go for the hallucinations.

While the twin prime conjecture is still open, there have been several amazing recent advances in this direction. See the video *Gaps between Primes - Numberphile* for some of this story.

<https://www.youtube.com/watch?v=vkMXdShDdtY>

Another famous open problem in number theory concerns decomposing integers into sums of primes. In a letter to Leonhard Euler in 1742, Christian Goldbach conjectured that every odd integer n , $n > 5$, is the sum of three primes. Euler replied that this conjecture is equivalent to the conjecture that every even integer n , $n > 2$, is the sum of two primes. For example, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 3 + 5$, $10 = 3 + 7$, \dots . Although no proof of the Goldbach conjecture has been found, the conjecture has been computationally checked to hold up to $4 \cdot 10^{18}$ [3].

Conjecture 3.3.11 (Goldbach conjecture). *Every even integer n , $n > 2$, is the sum of two primes.*

3.3.2. Greatest common divisor and Euclidean algorithm.

Definition 3.3.12. Let a and b be integers, not both 0. The *greatest common divisor* of a and b , denoted $\gcd(a, b)$ is the largest integer d such that $d \mid a$ and $d \mid b$. We say a is *relatively prime* or *coprime* to b if $\gcd(a, b) = 1$.

Remark 3.3.13. It is easy to see from the definition that $\gcd(a, b) = \gcd(b, a)$.

Example 3.3.14. The greatest common divisor of 10 and 15 is 5 since 5 divides both 10 and 15, and it is the largest integer to do so. We write $\gcd(10, 15) = 5$.

The integers 15 and 22 are relatively prime since $\gcd(22, 15) = 1$.

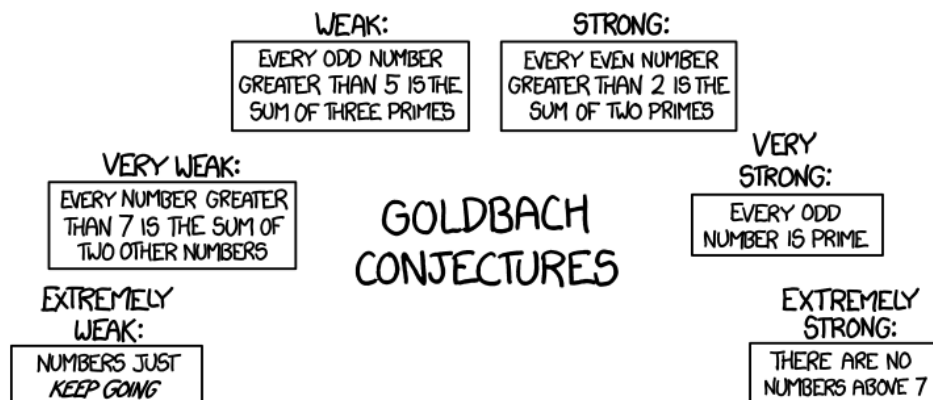


Figure 3.3.4. xkcd: Goldbach Conjectures (<https://xkcd.com/1310/>)

The weak twin primes conjecture states that there are infinitely many pairs of primes. The strong twin primes conjecture states that every prime p has a twin prime $(p + 2)$, although $(p + 2)$ may not look prime at first. The tautological prime conjecture states that the tautological prime conjecture is true.

Definition 3.3.15. For a positive integer n , the *Euler phi function* or *Euler totient function* evaluated at n , denoted $\phi(n)$, is the number of non-negative integers less than n that are relatively prime to n .

Example 3.3.16. There are four positive integers less than 10 that are relatively prime to 10: $\{1, 3, 7, 9\}$. Thus $\phi(10) = 4$.

Example 3.3.17. There are six positive integers less than 7 that are relatively prime to 7: $\{1, 2, 3, 4, 5, 6\}$. Thus $\phi(7) = 6$.

Remark 3.3.18. For p prime, it is easy to show that $\phi(p) = p - 1$. One can also show that for p and q distinct primes, $\phi(pq) = (p - 1)(q - 1)$.

Definition 3.3.19. Let a and b be positive integers. The *least common multiple* of a and b , denoted $\text{lcm}(a, b)$, is the smallest positive integer ℓ such that $a \mid \ell$ and $b \mid \ell$.

For integers we can factor, or for small integers where we can use trial division, it is straightforward to compute greatest common divisors and least common multiples.

Example 3.3.20. Let's compute the greatest common divisor and least common multiple of 24 and 30. We have that $24 = 2^3 \cdot 3$, and $30 = 2 \cdot 3 \cdot 5$. They have $2 \cdot 3$ in common, so greatest common divisor is

$$\gcd(30, 24) = 2 \cdot 3 = 6.$$

For the least common multiple, we need to include all the prime factors that arise and keep the larger exponent, so the least common multiple is

$$\text{lcm}(30, 24) = 2^3 \cdot 3 \cdot 5 = 120.$$

Note that $24 \cdot 30 = 720 = 6 \cdot 120$ so that

$$24 \cdot 30 = \text{gcd}(30, 24) \text{lcm}(30, 24).$$

This turns out to be true in general. See Theorem 3.3.21.

Theorem 3.3.21. *Let a and b be positive integers. Then*

$$ab = \text{gcd}(a, b) \text{lcm}(a, b)$$

Proof. Try this at home. Hint: Use the prime factorizations of a and b guaranteed from the Fundamental Theorem of Arithmetic. Express the gcd and lcm of a and b in terms of the factorizations. Compare the product of these with the prime factorization of the product of a and b . \square

Factoring or trial division works for computing greatest common divisors, but for large a and b , it is horribly inefficient. There is a better method known as the *Euclidean algorithm*. As an additional bonus, the theorem above shows that fast computation of gcd leads to fast computation of lcm. The Euclidean algorithm works because of the following lemma.

Lemma 3.3.22. *Let a and b be positive integers. Then*

$$\text{gcd}(a, b) = \text{gcd}(b, a \bmod b).$$

It may look like the Lemma does not help, as it just turns one gcd computation into another. The real power here comes from two facts:

- $\text{gcd}(a, b) = \text{gcd}(b, a)$, so we can arrange that b is less a ; (Note: if $a = b$, then $\text{gcd}(a, b) = a$ so we would only really use this lemma when $a \neq b$.)
- $a \bmod b$ is strictly less than b .

That means that Lemma 3.3.22 allows us to compute $\text{gcd}(a, b)$ by computing $\text{gcd}(A, B)$, where A and B are smaller than a and b . Nothing prevents us from repeatedly using this result, so we can keep using the result until we are computing $\text{gcd}(d, 0)$, which is equal to d .

Example 3.3.23. Let's compute $\text{gcd}(252, 198)$ using Lemma 3.3.22.

$$\text{Since } 252 \bmod 198 = 54,$$

$$\text{gcd}(252, 198) = \text{gcd}(198, 54).$$

Since $198 \bmod 54 = 36$, we have

$$\text{gcd}(198, 54) = \text{gcd}(54, 36).$$

Since $54 \bmod 36 = 18$, we have

$$\gcd(54, 36) = \gcd(36, 18).$$

Since $36 \bmod 18 = 0$, we have

$$\gcd(36, 18) = \gcd(18, 0) = 18.$$

Thus the greatest common divisor of 252 and 198 is 18.

$$\gcd(252, 198) = 18.$$

Theorem 3.3.24 (Bézout's theorem). *If a and b are positive integers, there exist integers s and t such that*

$$\gcd(a, b) = as + bt.$$

By keeping track of some extra information in the Euclidean algorithm, we get the solution to the Bézout equation as well. The extended version is called *Extended Euclidean Algorithm*.

Example 3.3.23 shows the computation of $\gcd(252, 198)$ using Lemma 3.3.22. In the following example, we go through the additional bookkeeping required to solve the Bézout equation.

Example 3.3.25. Find integers s and t such that

$$\gcd(252, 198) = 252s + 198t.$$

We construct a table to help with the bookkeeping. We have columns q , r , s , and t . The q column keeps track of the quotients. The r column keeps track of the remainder. The s and t columns show integers such that

$$r = 252s + 198t$$

on every row. We proceed using the Euclidean algorithm to reduce r to $\gcd(252, 198)$.

Step 1: Initialize the table.

q	r	s	t
	252	1	0
	198	0	1

Step 2: Compute the quotient

$$252 \operatorname{div} 198 = \left\lfloor \frac{252}{198} \right\rfloor = 1,$$

and enter it in the box in the q column as shown.

q	r	s	t
	252	1	0
1	198	0	1

This step may be easier to understand in a more general setting. The general pattern is repeated throughout the computation, so we go through it in more detail here. We have the table filled out as below. We want to compute the quotient q shown in a box below.

q	r	s	t
	r_1	s_1	t_1
q	r_2	s_2	t_2

To do so, we compute the quotient

$$q = r_1 \operatorname{div} r_2 = \left\lfloor \frac{r_1}{r_2} \right\rfloor,$$

and enter it in the box in the q column as shown.

Step 3: Use the value of q shown in **bold** to compute the next row values for r , s , and t . Compute the next r value: $252 - \mathbf{1} \cdot 198 = 54$. Compute the next s value: $1 - \mathbf{1} \cdot 0 = 1$. Compute the next t value: $0 - \mathbf{1} \cdot 1 = -1$. Enter these three values in their respective boxes as shown.

q	r	s	t
	252	1	0
1	198	0	1
54	1	-1	

This step may be easier to understand in a more general setting. The general pattern is repeated throughout the computation, so we go through it in more detail here. We have the table filled out as below. We just computed the quotient q shown in **bold** below. We want to compute the boxed quantities r_3 , s_3 and t_3 .

q	r	s	t
	r_1	s_1	t_1
q	r_2	s_2	t_2
r_3	s_3	t_3	

To do so, we compute

$$r_3 = r_1 - \mathbf{q}r_2$$

$$s_3 = s_1 - \mathbf{q}s_2$$

$$t_3 = t_1 - \mathbf{q}t_2.$$

You can also think of it as

$$(r_3, s_3, t_3) = (r_1, s_1, t_1) - \mathbf{q}(r_2, s_2, t_2).$$

Enter these values in the boxed spots in the table as shown.

Step 4: Now the bottom two rows look like the table at the end of Step 1, so we proceed to compute the next quotient as in Step 2

$$198 \operatorname{div} 54 = \left\lfloor \frac{198}{54} \right\rfloor = 3,$$

and enter the value in the box in the q column as shown.

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1

Step 5: Now the bottom two rows look like the table at end of Step 2, so we proceed to compute the next values of r , s , and t as in Step 3

$$(198, 0, 1) - 3(54, 1, -1) = (36, -3, 4),$$

and enter the values in their respective boxes as shown.

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1
36	-3	4	

Step 6: Compute the next quotient

$$54 \operatorname{div} 36 = \left\lfloor \frac{54}{36} \right\rfloor = 1.$$

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1
1	36	-3	4

Step 7: Compute the next row of r , s , and t

$$(54, 1, -1) - 1(36, -3, 4) = (18, 4, -5).$$

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1
1	36	-3	4
	18	4	-5

Step 8: Compute the next quotient

$$36 \operatorname{div} 18 = \left\lfloor \frac{36}{18} \right\rfloor = 2.$$

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1
1	36	-3	4
2	18	4	-5

Step 9: Compute the next row of r , s , and t . We can get by with less work in this step because in the r column, $36 - 2 \cdot 18 = 0$. That signals us that the computation is done, and the row above is the one we want. We can signify this in the table by entering 0 in the r column and putting $-$ in the s and t columns, since those values do not matter.

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1
1	36	-3	4
2	18	4	-5
	0	-	-

Step 10: Draw the conclusion. We have the following table computed. How do we interpret it?

q	r	s	t
	252	1	0
1	198	0	1
3	54	1	-1
1	36	-3	4
2	18	4	-5
	0	-	-

We look in the row *above* the row where we have a 0 in the r column. The r value is $\gcd(252, 198)$, and the values of s and t in that row satisfy

$$\gcd(252, 198) = 252s + 198t.$$

Thus we have shown that $\gcd(252, 198) = \boxed{18}$, and that

$$\boxed{18} = 252 \cdot \boxed{4} + 198 \cdot \boxed{-5}.$$

Example 3.3.26. Let's use the Extended Euclidean Algorithm to compute $\gcd(1184339, 137632)$ and to find integers s and t such that

$$\gcd(1184339, 137632) = 1184339s + 137632t.$$

We compute the table, following the steps as in Example 3.3.25.

q	r	s	t
	1184339	1	0
8	137632	0	1
1	83283	1	-8
1	54349	-1	9
1	28934	2	-17
1	25415	-3	26
7	3519	5	-43
4	782	-38	327
2	391	157	-1351
	0	-	-

The computation below shows that $\gcd(1184339, 137632) = \text{span style="border: 1px solid black; padding: 2px;">391}$ and

$$\text{span style="border: 1px solid black; padding: 2px;">391} = 1184339 \cdot \text{span style="border: 1px solid black; padding: 2px;">157} + 137632 \cdot \text{span style="border: 1px solid black; padding: 2px;">-1351}.$$

Python Code Snippet 3.3.27. Here is Python code that will do Extended Euclidean Algorithm to find a solution to the Bézout equation.

```
def XGCD(a,b):
    '''
    Return [d,s,t], where d = (a,b) and s, t are integers such
    that d = as + bt. Uses Extended Euclidean Algorithm.
    '''
    # set up first 2 rows
    r1, s1, t1, r2, s2, t2 = a, 1, 0, b, 0, 1
    r = r2
    while r!= 0: # while remainder is not 0
        q, r = divmod(r1,r2) # compute quotient and remainder
        s = s1 - q*s2
        t = t1 - q*t2
        # now shift everything
        r1, s1, t1, r2, s2, t2 = r2, s2, t2, r, s, t
    # want data just before remainder 0
    return [r1, s1, t1]
```

Lemma 3.3.28. Let a, b, c be positive integers such that $\gcd(a, b) = 1$ and $a \mid bc$. Then $a \mid c$.

Proof. By Bézout's theorem, there exist integers s and t such that

$$1 = sa + tb.$$

Multiply both sides by c to get

$$c = cas + ctb.$$

It is clear that $a \mid csa$, and $a \mid bc$ by assumption, so $a \mid ctb$. Thus a divides the sum c as desired. \square

Although we cannot divide both sides of a congruence by an integer, the following result says that we can if the integer is relatively prime to the modulus.

Theorem 3.3.29. *Let m be a positive integer. Let a , b , and c be integers. If $ac \equiv bc \pmod{m}$, and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.*

Proof. Suppose $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$. Then $m \mid (ac - bc)$, so $m \mid c(a - b)$. Since $\gcd(c, m) = 1$, Lemma 3.3.28 implies $m \mid (a - b)$. Then $a \equiv b \pmod{m}$ as desired. \square

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) prime
 - (b) composite
 - (c) twin primes
 - (d) greatest common divisor of integers
 - (e) relatively prime or coprime integers
 - (f) least common multiple
 - (g) Euler phi function
2. State precisely the Fundamental Theorem of Arithmetic. Be sure to set up any notation that is required.
3. State precisely Bézout's theorem. Be sure to set up any notation that is required.
4. Compute $\phi(15)$.
5. Compute $\phi(24)$.
6. Which positive integers less than 12 are relatively prime to 12?
7. Which positive integers less than 25 are relatively prime to 25?
8. Determine whether these integers are prime.
 - (a) 21
 - (b) 100

- (c) 101
 - (d) 253
 - (e) 91
9. Compute the greatest common divisor and least common multiple of the following integers.
- (a) 131 and 19
 - (b) 260 and 77
 - (c) 46 and 34
 - (d) 132 and 192
 - (e) 293 and 37
 - (f) 15 and 87
 - (g) 57 and 93
10. Compute the greatest common divisor and least common multiple of the following integers.
- (a) 36503 and 5017
 - (b) 8479 and 12017
 - (c) 15089 and 16999
 - (d) 11371 and 10541
 - (e) 14453 and 26671
11. Compute the greatest common divisor and least common multiple of the following integers.
- (a) 3599 and 5917
 - (b) 9701 and 8633
 - (c) 23707 and 5809
 - (d) 8413 and 12709
 - (e) 19303 and 5917
12. Show that $\gcd(75, 53) = 1$. Find integers s and t such that

$$75s + 53t = 1.$$

Use these to find an integral solution to

$$75x + 53y = 13.$$

Check your solutions by plugging back in.

13. Show that $\gcd(75, 10) = 5$. Use this to show that

$$75x + 10y = 13$$

has no integral solutions.

14. Explain why there are no integers s and t such that $25s + 30t = 1$.
15. Find integers s and t such that $2018s + 253t = 1$.
16. Use the Extended Euclidean Algorithm to find integers s and t such that

$$4321s + 12367t = 149.$$

Check your solution by plugging back in.

17. Use the Extended Euclidean Algorithm to find integers s and t such that

$$5293s + 8509t = 67.$$

Check your solution by plugging back in.

18. Use the Extended Euclidean Algorithm to find integers s and t such that

$$27263s + 44377t = 199.$$

Check your solution by plugging back in.

19. Use the Sieve of Eratosthenes to find primes up to 100. (You should find 25 primes less than 100.)

3.4. Solving congruences

Goals. To learn how to solve linear congruences and simultaneous systems of linear congruences.

3.4.1. Linear congruences.

Definition 3.4.1. A *linear congruence* is a congruence of the form

$$ax \equiv b \pmod{m},$$

where m is a positive integer, a and b are integers, and x is a variable.

We want to solve linear congruences. As motivation, note that if we wanted to solve $ax = b$, and a has a multiplicative inverse a^{-1} , then we would just multiply both sides by a^{-1} to solve, and get $x = a^{-1}b$. e.g., The solution to $7x = 3$ is $x = \frac{3}{7}$. The same idea works for linear congruences if there exists an integer \bar{a} such that $a\bar{a} \equiv 1 \pmod{m}$. Such an integer is called an *inverse* of a modulo m .

Definition 3.4.2. Let a and m be integers, with $m > 1$. An *inverse* of a modulo m is an integer \bar{a} such that $a\bar{a} \equiv 1 \pmod{m}$.

Example 3.4.3. 5 is an inverse of 7 modulo 17, since $5 \cdot 7 \equiv 1 \pmod{17}$. Note that an inverse is not unique. 22 is another inverse of 7 modulo 17, since $22 \cdot 7 \equiv 1 \pmod{17}$. Inverses can be negative. For example, -12 is an inverse of 7 modulo 17. In fact, there are infinitely many inverses of 7 modulo 17. Every integer in the congruence class of 5 modulo 17 is an inverse of 7 modulo 17

$$[5] = \{5 + 17k \mid k \in \mathbb{Z}\}.$$

In the example above, we see that the inverses of 7 modulo 17 form a congruence class. If an integer a has an inverse modulo m , this is what happens in general.

Theorem 3.4.4. *If $\gcd(a, m) = 1$, then the inverse of a modulo m exists. Furthermore, it is unique modulo m .*

Proof. Suppose $\gcd(a, m) = 1$. Then by Bézout, there exists integers s and t such that $as + mt = 1$. It follows that $mt = 1 - as$, so $as \equiv 1 \pmod{m}$. Thus s is an inverse of a modulo m .

Next, we show uniqueness. Suppose s and s' are inverses of a modulo m . We need to show $s \equiv s' \pmod{m}$. Since s and s' are inverses of a modulo m , there exists integers k and k' such that $as = 1 + km$ and $as' = 1 + k'm$. Then

$$a(s - s') = as - as' = (1 + km) - (1 + k'm) = m(k - k').$$

Then m divides $a(s - s')$ and $\gcd(a, m) = 1$, so Lemma 3.3.28 implies m divides $s - s'$. It follows that $s \equiv s' \pmod{m}$, as desired. \square

Using modular arithmetic, we can see a few things that will simplify computations. Suppose $\gcd(a, m) = 1$ so that a has an inverse modulo m . An inverse of a will be an inverse for every integer in the congruence class of a modulo m . We can use this to simplify our problem when $|a| > m$. Furthermore, once an inverse is found, every integer in the congruence class of the inverse is also an inverse. This is helpful when we want to find an inverse with additional properties. For example, we may want to find an inverse that is positive and “small” since we are doing computations by hand.

Example 3.4.5. Let's find a positive integer that is an inverse of 253 modulo 8.

Since $253 \pmod{8} = 5$, this reduces to finding an inverse of 5 modulo 8. Since $3 \cdot 5 \equiv 15 \equiv -1 \pmod{8}$, we have that -3 is an inverse of 5 modulo 8. Every integer in the congruence class $[-3]$ is also an inverse. We want a positive inverse, so we can take $-3 + 8 = 5$ as an inverse. In fact, since the inverse is unique modulo 8, we just proved that this is the smallest positive integer that is an inverse of 253 modulo 8.

We can use the modular inverse to solve linear congruences $ax \equiv b \pmod{m}$ when $\gcd(a, m) = 1$. Namely, $x \equiv \bar{a}b \pmod{m}$. When $\gcd(a, m) \neq 1$, the solution is a bit more subtle.

For small m , the inverse of a is easy to compute by inspection. For larger m , use Extended Euclidean algorithm. Specifically, if $\gcd(a, m) = 1$, there exists integers s and t such that

$$1 = as + tm.$$

Then

$$1 \equiv as \pmod{m}.$$

3.4.2. Chinese Remainder Theorem.

Theorem 3.4.6 (Chinese Remainder Theorem). *Let $m = m_1 m_2 \cdots m_r$ with $\gcd(m_i, m_j) = 1$ for all $i \neq j$. Then the system*

$$\left\{ \begin{array}{l} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \vdots \\ x \equiv a_r \pmod{m_r} \end{array} \right\}$$

has a unique solution modulo m , given by

$$x = a_1 e_1 + a_2 e_2 + \cdots + a_r e_r, \quad \text{where } e_i = w_i \cdot t_i,$$

with

$$w_i = \frac{m}{m_i} \quad \text{and} \quad t_i w_i \equiv 1 \pmod{m_i}.$$

In other words, the solution set is exactly the congruence class of x modulo m ,

$$[x] = \{x + km \mid k \in \mathbb{Z}\}.$$

Remark 3.4.7. Roughly speaking, the e_i is 1 in the mod m_i direction and 0 in the mod m_j direction for $i \neq j$.

Steps to solve CRT problem:

- (1) Identify a_i and m_i .
- (2) Compute m and w_i .
- (3) Compute t_i , the inverse of w_i modulo m_i .
- (4) There is a unique solution modulo m

$$x = a_1 t_1 w_1 + a_2 t_2 w_2 + \cdots + a_r t_r w_r.$$

Example 3.4.8. Use the Chinese Remainder Theorem to find all of the solutions to following the system of congruences.

$$\left\{ \begin{array}{l} x \equiv 7 \pmod{12} \\ x \equiv 3 \pmod{5} \end{array} \right\}$$

Following the notation above, we have $a_1 = 7$, $m_1 = 12$, $a_2 = 3$, and $m_2 = 5$. Since $\gcd(12, 5) = 1$, we can use CRT. We are guaranteed a unique solution modulo $m = 12 \cdot 5 = 60$. It has the form

$$x \equiv 7 \cdot e_1 + 3e_2 \pmod{60}.$$

Recall that e_1 is an integer such that $e_1 \equiv 1 \pmod{12}$ and $e_1 \equiv 0 \pmod{5}$. We can compute e_1 as $e_1 = w_1 t_1$, where $w_1 = m/m_1$, and t_1 is the inverse

of w_1 modulo m_1 . Then $w_1 = 5$. By inspection, we can take $t_1 = 5$, since $5 \cdot w_1 \equiv 1 \pmod{12}$. Then $e_1 = 5 \cdot 5 = 25$.

Similarly, $e_2 = w_2 t_2$, where $w_2 = m/m_2 = 12$. We compute the inverse of 12 modulo 5, but this is the same as the inverse of 2 modulo 5, since $12 \equiv 2 \pmod{5}$. By inspection, we can take $t_2 = 3$, so $e_2 = 12 \cdot 3 = 36$.

Then

$$x \equiv 7 \cdot 25 + 3 \cdot 36 \equiv 283 \equiv 43 \pmod{60}.$$

Equivalently,

$$x = 43 + 60k, \quad \text{for } k \in \mathbb{Z}.$$

Example 3.4.9. Find all the solutions to

$$\left\{ \begin{array}{l} x \equiv 1 \pmod{2} \\ x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 4 \pmod{11} \end{array} \right\}.$$

Since 2, 3, 5, 11 are pairwise coprime, we can use CRT. There is a unique solution modulo $m = 2 \cdot 3 \cdot 5 \cdot 11 = 330$. Let $w_i = \frac{m}{m_i}$. Then

$$w_1 = 165$$

$$w_2 = 110$$

$$w_3 = 66$$

$$w_4 = 30.$$

We compute t_i such that $t_i w_i \equiv 1 \pmod{m_i}$. Then t_1 is the inverse of 165 modulo 2. We take $t_1 = 1$. Similarly, t_2 is inverse of 110 modulo 3, which is the same as the inverse of 2 modulo 3. We pick $t_2 = 2$. Similarly, we can take $t_3 = 1$ and $t_4 = 7$. Then

$$e_1 = t_1 w_1 = 1 \cdot 165 = 165$$

$$e_2 = t_2 w_2 = 2 \cdot 110 = 220$$

$$e_3 = t_3 w_3 = 1 \cdot 66 = 66$$

$$e_4 = t_4 w_4 = 7 \cdot 30 = 210.$$

Then

$$\begin{aligned} x &\equiv 1 \cdot e_1 + 2e_2 + 3e_3 + 4e_4 \pmod{330} \\ &\equiv 1 \cdot 165 + 2 \cdot 220 + 3 \cdot 66 + 4 \cdot 210 \pmod{330} \\ &\equiv 323 \pmod{330}. \end{aligned}$$

Thus $x = 323 + 330k$, for $k \in \mathbb{Z}$. Equivalently, $x \equiv 323 \pmod{330}$.

Example 3.4.10. My son had 500 action figures before we moved to Greensboro. He has not bought any new ones, but he lost a few in the process of the move, and he wants to know how many he has now. He can only count accurately to 10, but he knows that you are a number theorist, and he has

faith in you. He reports that there is an odd number left. When you tell him that is not enough information, he reports that there is 1 left over if he lines them up 5 at a time, 2 left over if he lines the up 7 at a time, and 3 left over if he lines them up 9 at a time. How many action figures does he have?

Let x be the number of action figures my son has. Then

$$\begin{array}{ll} x \equiv 1 \pmod{2} & \text{since } x \text{ is odd} \\ x \equiv 1 \pmod{5} & \text{since there is 1 left over in rows of 5} \\ x \equiv 2 \pmod{7} & \text{since there is 2 left over in rows of 7} \\ x \equiv 3 \pmod{9} & \text{since there is 3 left over in rows of 9} \end{array}$$

Note that we can solve for x using CRT since

$$\gcd(2, 5) = \gcd(2, 7) = \gcd(2, 9) = \gcd(5, 7) = \gcd(5, 9) = \gcd(7, 9) = 1.$$

Let's follow the steps to solve a CRT problem:

(1) We identify a_i and m_i . We have

$$a_1 = 1, m_1 = 2 \quad a_2 = 1, m_2 = 5 \quad a_3 = 2, m_3 = 7 \quad a_4 = 3, m_4 = 9.$$

(2) We compute

$$\begin{aligned} m &= m_1 m_2 m_3 m_4 = 630 \\ w_1 &= m_2 m_3 m_4 = 315 \\ w_2 &= m_1 m_3 m_4 = 126 \\ w_3 &= m_1 m_2 m_4 = 90 \\ w_4 &= m_1 m_2 m_3 = 70. \end{aligned}$$

- (3) t_1 : The inverse of 315 modulo 2 is the same as the inverse of 1 modulo 2, which is 1 by inspection. Specifically, we choose $t_1 = 1$.
- t_2 : The inverse of 126 modulo 5 is the same as the inverse of 1 modulo 5, which is 1 by inspection. Specifically, we choose $t_2 = 1$.
- t_3 : The inverse of 90 modulo 7 is the same as the inverse of -1 modulo 7, which is -1 . Specifically, we choose $t_3 = -1$ (Note: Some of you will instead say the inverse of 90 modulo 7 is the same as the inverse of 6 modulo 7, which is 6. That is fine as well. My way just keeps the numbers smaller if you are willing to use negative numbers.)
- t_4 : The inverse of 70 modulo 9 is the same as the inverse of 7 modulo 9, which is 4 by inspection. Specifically, $t_4 = 4$.

(4) We compute

$$\begin{aligned} x &\equiv a_1t_1w_1 + a_2t_2w_2 + a_3t_3w_3 + a_4t_4w_4 \pmod{630} \\ &\equiv (1 \cdot 1 \cdot 315) + (1 \cdot 1 \cdot 126) + (2 \cdot (-1) \cdot 90) + (3 \cdot 4 \cdot 70) \pmod{630} \\ &\equiv 1101 \pmod{630} \\ &\equiv 471 \pmod{630}. \end{aligned}$$

In other words, $x = 471 + 630k$ for some integer k . Since my son has less than 500 action figures, he must have 471 action figures.

3.4.3. Fermat's little theorem and Euler's generalization.

Theorem 3.4.11 (Fermat's little theorem and Euler's generalization). *If p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p}$. Furthermore, $a^p \equiv a \pmod{p}$. More generally, if $\gcd(a, m) = 1$, then $a^{\phi(m)} \equiv 1 \pmod{m}$.*

Example 3.4.12. Compute $8^{222} \pmod{11}$.

Note that 11 is prime and $\gcd(11, 8) = 1$. Since $11 - 1 = 10$, Fermat's little theorem implies $8^{10} \equiv 1 \pmod{11}$. We rewrite the exponent 222 as

$$222 = 22 \cdot 10 + 2$$

and use properties of exponents to get

$$8^{222} \equiv 8^{22 \cdot 10 + 2} \equiv 1^{22} \cdot 8^2 \pmod{11} \equiv 9 \pmod{11}.$$

Thus $8^{222} \pmod{11} = 9$.

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) linear congruence
 - (b) inverse of an integer modulo an integer
2. State precisely the Chinese Remainder Theorem. Be sure to set up any notation that is required.
3. State precisely the Fermat's little theorem and Euler's generalization. Be sure to set up any notation that is required.
4. Show that 25 is an inverse of 13 modulo 36.
5. Find an inverse of 5 modulo 7 by inspection.
6. Use the Extended Euclidean Algorithm to find an inverse of 68 modulo 253.

7. Use Chinese Remainder Theorem to find the smallest positive solution to the system of congruences

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 1 \pmod{4} \\ x \equiv 3 \pmod{5} \end{cases}.$$

8. Use Fermat's Little Theorem to compute $7^{222} \pmod{11}$.
 9. Use Fermat's little theorem to compute $3^{302} \pmod{5}$.
 10. Find all the solutions to

$$\begin{cases} x \equiv 2 \pmod{13} \\ x \equiv 6 \pmod{15} \end{cases}.$$

What is the smallest positive solution?

11. Find all the solutions to

$$\begin{cases} x \equiv 2 \pmod{7} \\ x \equiv 3 \pmod{8} \\ x \equiv 4 \pmod{5} \end{cases}.$$

What is the smallest positive solution?

12. The problems below are related. Namely, a computation in one portion may be used in more than one part.
 (a) Compute $\gcd(2468, 357)$.
 (b) Use the Extended Euclidean Algorithm to solve the Bézout equation for 2468 and 357. Namely, find integers s and t such that

$$\gcd(2468, 357) = 2468s + 357t.$$

- (c) Compute an inverse of 357 modulo 2468, or explain why one does not exist.
 (d) Find an inverse of 19 modulo 241.
 (e) Show that 937 is an inverse of 13 modulo 2436.
 (f) Solve the linear congruence $357x \equiv 123 \pmod{2468}$.
13. I have an unknown number of Easter candies. When I arrange them in groups of 20, there are 3 left over. When I arrange them in groups of 41, there are 26 left over. What is the minimum number of candies that I could have?
14. Solve the congruence $2x \equiv 3 \pmod{13}$ by inspection.
15. Solve the linear congruence $19x \equiv 11 \pmod{141}$.
16. Solve the congruence $200x \equiv 5 \pmod{1357}$ using modular inverses.
17. Split up the positive integers less than 13, except 1 and 12 into pairs of integers such that each pair consists of integers that are inverses of each other modulo 13. Use this to show $12! \equiv -1 \pmod{13}$.



Figure 3.5.1. xkcd: Code Talkers. (<https://xkcd.com/257/>) As far as I can tell, Navajo doesn't have a common word for 'zero'. do-neh- lini means 'neutral'.

18. Suppose you collected shells on the beach with your daughter. When you arrange them in piles of 15, there are 13 left over. When you arrange them in piles of 19, there are 6 leftover. Use the Chinese Remainder Theorem to figure out how many shells you collected. Give the smallest positive solution.

3.5. Cryptography

Goals. To introduce the basic notions of cryptography and cryptographic protocols. To explain both classical and modern encryption methods.

3.5.1. Shift ciphers. We want a method to encrypt, or make secret, a plaintext message. One of the earliest know approaches was by Julius Caesar. The idea was to basically shift the alphabet by 3 spots.

Definition 3.5.1. The *Caesar cipher* is the shift cipher, where we shift forward by three.

Here are the steps to encrypt using the Caesar cipher.

Table 3.5.1. Caesar cipher lookup table.

A	B	C	D	E	F	G	H	I	J	K	L	M
D	E	F	G	H	I	J	K	L	M	N	O	P
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- (1) Replace each letter in the message by an element of \mathbb{Z}_{26} equal to 1 less than its position in the alphabet. e.g., A is replaced by 0, B is replaced by 1, ..., Z is replaced by 25. In this step, we are *encoding* the message as a list of integers.
- (2) Replace each number p by $(p + 3) \bmod 26$. Equivalently, apply the function $f: \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26}$ defined by $f(x) = x + 3 \bmod 26$ to each number. This step does the *encrypting*.
- (3) Translate the numbers back to letters. This step does the *decoding*.

Example 3.5.2. Let's encrypt CAT using the Caesar cipher.

First, we change CAT to numbers, we get 2 0 19. Apply f to get 5 3 22. Translate back to letters to get FDW.

$$\begin{aligned} C &\xrightarrow{\text{encode}} 2 \xrightarrow{\text{encrypt}} 5 \xrightarrow{\text{decode}} F \\ A &\xrightarrow{\text{encode}} 0 \xrightarrow{\text{encrypt}} 3 \xrightarrow{\text{decode}} D \\ T &\xrightarrow{\text{encode}} 19 \xrightarrow{\text{encrypt}} 22 \xrightarrow{\text{decode}} W \end{aligned}$$

If we were going to encrypt longer messages, it would be faster to pre-compute a lookup table as shown in Table 3.5.1.

Remark 3.5.3. Instead of using f , which represents a shift of 3, we can shift by any integer amount

$$f(x) = (x + k) \bmod 26$$

to yield a cipher known as a *shift cipher*. To decode, we use a function

$$g(x) = f^{-1}(x) = x - k \bmod 26.$$

More generally, the encryption function f can be any invertible function $f: \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26}$.

3.5.2. RSA. First, we give a bit of history. Ron Rivest, Adi Shamir, and Leonard Adleman (shown in Figure 3.5.2) first publicly described this algorithm for public key encryption in 1978¹[4]. They posted one of the first

¹Clifford Cocks described an equivalent system in 1973, but it was classified by the UK intelligence agency GCHQ until 1997

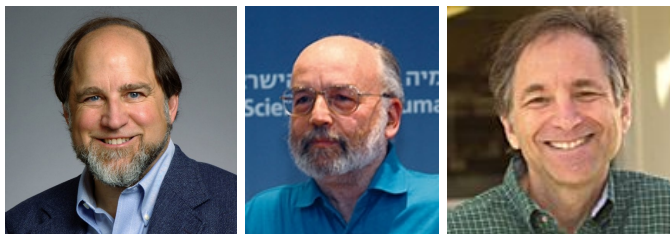


Figure 3.5.2. Rivest, Shamir, and Adleman. Ron Rivest, Adi Shamir, and Leonard Adleman (left to right), inventors of the RSA encryption scheme.

public-key encryption messages using a 129 digit number which later became known as RSA-129 [2].

$$\begin{aligned}
 \text{RSA-129} &= 1143816257578888676692357799761466120102182967212 \\
 &\quad 4236256256184293570693524573389783059712356395870 \\
 &\quad 5058989075147599290026879543541 \\
 &= 3490529510847650949147849619903898133417764638493 \\
 &\quad 387843990820577 \\
 &\times 3276913299326670954996198819083446141317764296799 \\
 &\quad 2942539798288533.
 \end{aligned}$$

They offered a \$100 prize and remarked that using technology and factoring techniques available at that time, it would take 40 quadrillion years to crack. Advances in factoring techniques and computers cracked the code in April 1994 [1] to find that the secret message was:

The Magic Words are Squeamish Ossifrage²

Suppose Alice wants to send Bob an encrypted message. Bob lets her know his public key, a pair of integers (N, e) .

Definition 3.5.4. The *RSA public encryption key* consists of a pair of integers (N, e) , where N is the product of two distinct primes p and q , and the *encryption exponent* e is relatively prime to $\phi(N) = (p - 1)(q - 1)$.

The *message space* is set of integers $\{1, 2, \dots, N\}$. To encrypt a message M from the message space, Alice computes the integer C in the message space that satisfies

$$C \equiv M^e \pmod{N}.$$

Notice that with fast exponentiation, this is fast.

²According to WIKIPEDIA, Ossifrage is an older name for the lammergeier, a scavenging vulture that is famous for dropping animal bones and live tortoises onto rocks to crack them open. It might perhaps be considered among the least squeamish of creatures.

Remark 3.5.5. RSA encryption works when the messages are integers between 1 and N . As we saw in §3.2.2, ASCII encoding allows us to encode a message string into an integer. After the message is encoded as an integer, we can encrypt it with RSA encryption.

If a Eve captures C while it is being transmitted, she will have a hard time computing the original message M , since taking e th roots is a hard problem known as the *discrete log problem*.

How is it any easier for Bob? The trick is that Bob has a bit of extra information. When constructing the key, Bob chooses N to be a product of two distinct primes p and q , so that $N = pq$. The exponent e is chosen so that the greatest common divisor $\gcd(e, \phi(N)) = 1$. Since Bob knows the factorization of N , he can easily compute the Euler *phi* function $\phi(N) = (p - 1)(q - 1)$. Then using the Euclidean algorithm, Bob can compute an inverse to e modulo $\phi(N)$, an integer d such that $ed \equiv 1 \pmod{\phi(N)}$. By Bézout's Theorem, there is an integer k so that $ed = 1 + k\phi(N)$. Now Euler's generalization to Fermat's little theorem says that if $\gcd(C, N) = 1$, we have

$$\begin{aligned} C^d &\equiv (M^e)^d \pmod{N} \\ &\equiv M^{1+k\phi(N)} \pmod{N} \\ &\equiv M \cdot (M^{\phi(N)})^k \pmod{N} \\ &\equiv M \pmod{N}. \end{aligned}$$

In fact, this happens even if $\gcd(C, N) \neq 1$. We show this using CRT in Theorem 3.5.8.

Thus, to decrypt the message, Bob does not need to take an e th root of C modulo N . Instead, he can raise C to the d th power and achieve the same result. Thank you, Euler! Again, with fast exponentiation, this is fast.

Note that if Eve can factor N , then she can also decrypt the message. For that reason, Bob must choose p and q “large” enough.

Definition 3.5.6. A positive integer d is a *decryption exponent* for RSA public key (N, e) , d is an inverse of e modulo $\phi(N)$.

Before proving that RSA works in general, let's look at a small example.

Example 3.5.7. Suppose Bob has public key $(N, e) = (55, 3)$. For the purposes of this example, pretend 55 is so large that Eve cannot factor it. Then Alice can encrypt any number from 1 to 55. Suppose Alice wants to send $M = 18$. She computes

$$C = M^e \pmod{N} = 18^3 \pmod{55} = 2.$$

How does Bob decrypt C ? He knows that $\phi(55) = 40$, since he created the key. The decryption exponent d is the inverse of e modulo $\phi(N)$, so

it satisfies $ed \equiv 1 \pmod{\phi(N)}$. Bob can compute $d \equiv -13 \pmod{40}$ by Extended Euclidean Algorithm

q	r	s	t
	40	1	0
13	3	0	1
3	1	1	-13
	0	-3	40

or by observing that $3 \cdot 13 = 39 \equiv -1 \pmod{40}$. Thus $d = -13 \pmod{40} = 27$.

To decrypt $C = 2$, Bob computes

$$M = C^d \pmod{N} = 2^{27} \pmod{55} = 18.$$

What are some of the problems with this example? i.e., What kind of attacks should Eve try to decrypt the message?

First, the value $N = 55$ is too easy to factor. Once Eve knows $55 = 5 \cdot 11$, CRT tells her that

$$\phi(55) = \phi(5)\phi(11) = (5 - 1)(11 - 1) = 40.$$

Then she can compute d using Euclidean algorithm just as Bob did to decrypt any intercepted message.

Next, the message space is too small. Notice that if Eve could solve $x^3 \equiv 2 \pmod{55}$, she can find Alice's message. The message space $\{1, 2, \dots, 55\}$ is small enough that Eve could just compute $x^3 \pmod{55}$ for several values of x and quickly find an answer.

Theorem 3.5.8. *Let N , e , and d be positive integers such that*

- (1) *N is the product of two distinct primes, i.e., there are primes $p \neq q$ such that $N = pq$;*
- (2) *e is relatively prime to $\phi(N)$, i.e., $\gcd(e, \phi(N)) = 1$;*
- (3) *d is an inverse of e modulo $\phi(N)$, i.e., $ed \equiv 1 \pmod{\phi(N)}$.*

Let M be any integer, and let $C = M^e$. Then $C^d \equiv M \pmod{N}$.

Why is RSA secure? What numbers should I pick? The encryption exponent e is typically chosen to be $e = 2^{16} + 1 = 65537$. This is not for security, but for speed. Because of our fast powering algorithm, this choice of e allows encryption to be done even more quickly.

If p and q are chosen to be *large* primes (for today's technology 100-200 digit primes are large enough), then the claim is that this encryption is secure. To decrypt a message, we either have to be able to take eth roots (i.e., solve $x^e \equiv C \pmod{N}$), or compute d . The first problem is hard, and one of the main ways to try to solve it is via finding d . In order to find d , we must know

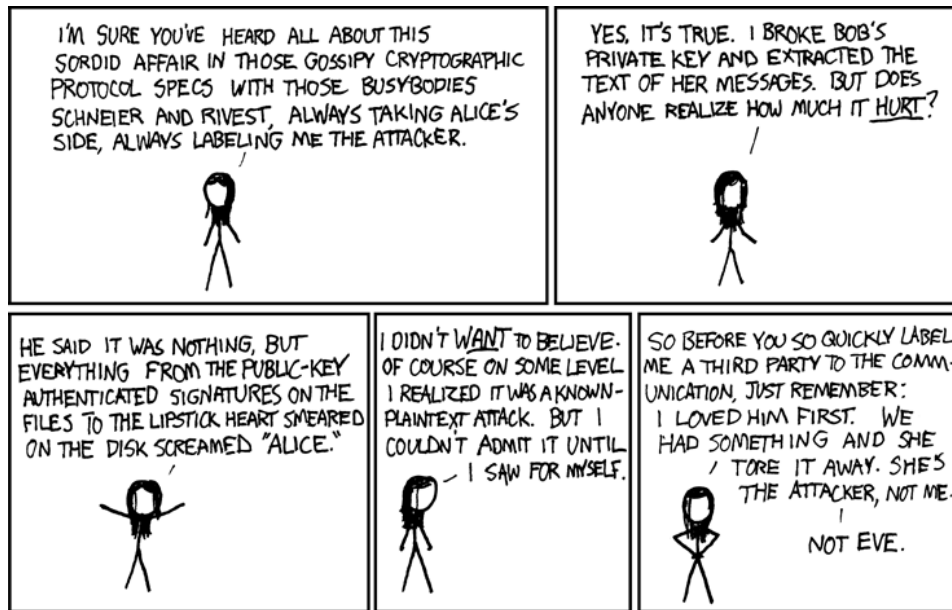


Figure 3.5.3. xkcd: Alice and Bob. (<https://xkcd.com/177/>) Yet one more reason I'm barred from speaking at crypto conferences.

$\phi(N)$. As long as Bob keeps the factors p and q secret then computing $\phi(N)$ is hard.

In the days of early commercial cryptography, many companies offered “challenges” to measure the state of progress in practical cryptanalysis. RSA used a *Factoring Challenge*. More information can be found at

https://en.wikipedia.org/wiki/RSA_Factoring_Challenge.

They posted a set of eight challenge numbers, ranging in size from 576 bits (174 decimal digits) to 2048 bits (617 decimal digits) that made up the challenge³. Each number is the product of two large primes, similar to the modulus of an RSA key pair. The first person to submit a correct factorization for any of the challenge numbers was eligible for a cash prize. To date, only four of the eight challenge numbers have been factored.

³The RSA numbers were generated on a computer with no network connection of any kind. The computer's hard drive was subsequently destroyed so that no record would exist, anywhere, of the solution to the factoring challenge. Not even the people at RSA knew the factorizations.

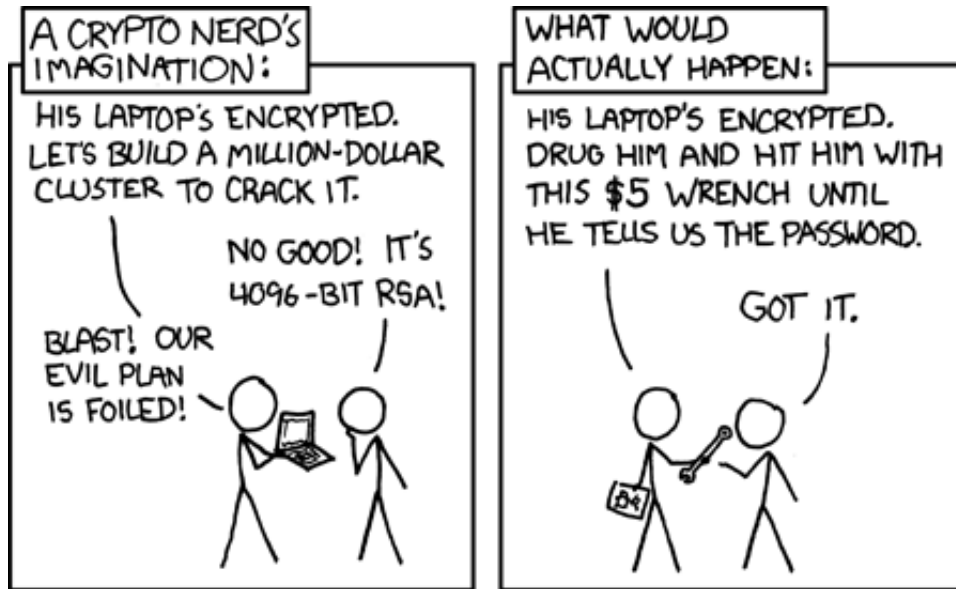


Figure 3.5.4. xkcd: Security. (<https://xkcd.com/538/>) Actual actual reality: nobody cares about his secrets. (Also, I would be hard-pressed to find that wrench for \$5.)

A reasonable RSA key to use is 1024 bits. Their RSA-1024 has 309 decimal digits.

```
N = 135066410865995223349603216278805969938881475605667027524485
143851526510604859533833940287150571909441798207282164471551
373680419703964191743046496589274256239341020864383202110372
958725762358509643110564073501508187510676594629205563685529
475213500852879416377328533906109750544334999811150056977236
890927563
```

Just how large is N ? There is a nice video by Vsauce explaining to see how large $52!$ is. If you don't have 20 minutes, skip to around the 15:56 minute mark. Even then, we are not done, since $52!$ is MUCH smaller than N . In fact,

$$N > (52!)^4,$$

so if we repeat the process $52!$ times, then repeat all of that $52!$ times, then repeat all of that $52!$ times, we are still not done.

<https://www.youtube.com/watch?v=ObiqJzfyACM>

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) Caesar cipher
 - (b) RSA public encryption key
 - (c) RSA encryption exponent
 - (d) RSA decryption exponent
2. Encrypt the message UNCG SPARTANS by translating the letters into numbers, applying the given encryption function, then translating the numbers back into letters.
 - (a) $f(p) = p + 3 \pmod{26}$
 - (b) $f(p) = p + 22 \pmod{26}$
 - (c) $f(p) = -3p \pmod{26}$
 - (d) $f(p) = 5p + 7 \pmod{26}$
3. Encrypt the message MIDNIGHT by translating the letters into numbers, applying the given encryption function, then translating the numbers back into letters.
 - (a) $f(p) = p + 17 \pmod{26}$
 - (b) $f(p) = p - 4 \pmod{26}$
 - (c) $f(p) = -7p \pmod{26}$
 - (d) $f(p) = 3p + 12 \pmod{26}$
4. Decrypt these messages that were encrypted using the Caesar cipher.
 - (a) KHOS
 - (b) VSDUWDQV
 - (c) VHFUHW
 - (d) DEVTXDWXODWH
 - (e) VXUUHSWLWLRXV
 - (f) DOLFH ORYHV ERE
5. Decrypt these messages that were encrypted using the encryption function $f(p) = 5p - 3 \pmod{26}$.
 - (a) OGERR
 - (b) LKYLJLCAR
 - (c) XKPKNFPTJ
 - (d) CPC APYRJ RYR
6. Decrypt these messages that were encrypted using the encryption function $f(p) = p + 10 \pmod{26}$.
 - (a) LVEO
 - (b) LKXKXK
 - (c) CYVSNKBSDI
 - (d) OFO VYFOC KVSMO

7. Suppose the ciphertext HVS ZONM RCU XIADG CJSF HVS EIWQY PFCKB TCL was produced by encrypting a plaintext message using a shift cipher. What is the original plaintext message?
 8. My RSA public key is $(N, e) = (85, 3)$. Encrypt the number $M = 23$.
 9. Encrypt the message $M = 253$ using the RSA encryption scheme with public key $(N, e) = (391, 17)$.
 10. Encrypt the current year using the RSA encryption scheme with public key $(N, e) = (343751, 23)$.
 11. Show that the RSA public key $(N, e) = (527, 13)$ is too small by computing the decryption exponent.
 12. Alice encodes her birthday as an 8-digit number $yyymmdd$. Suppose she encrypts it using the RSA encryption scheme with public key
$$(N, e) = (25736197, 29),$$
resulting in the ciphertext $C = 8141408$. What is her birthday?
-

Induction

Many times, a universal quantification can be reinterpreted as a nice family of propositions, indexed by positive integers. For example, consider the statement “the sum of the first n positive integers is $\frac{n(n+1)}{2}$.” This can be rephrased as the universally quantified statement “for every positive integer n , $1 + 2 + \cdots + n = \frac{n(n+1)}{2}$.” This can be seen as a family of infinitely many simpler propositions:

$$\begin{aligned} 1 &= 1 \\ 1 + 2 &= \frac{2(2+1)}{2} \\ 1 + 2 + 3 &= \frac{3(3+1)}{2} \\ &\vdots \end{aligned}$$

Mathematical induction is a powerful proof technique that allows us to prove certain universally quantified statements by examining closely the related family of more basic propositions. Despite its name, mathematical induction is an example of deductive, not inductive, reasoning.

4.1. Mathematical induction

Goals. To explain how to construct proofs of a variety of theorems using mathematical induction.

Mathematical induction can be used to prove statements that assert $P(n)$ is true for all positive integers n .

In its most basic form, there are two parts:

- (1) Prove $P(1)$ is true.
- (2) Prove for all positive integers k , if $P(k)$ is true, then $P(k + 1)$ is true.

Then symbolically, mathematical induction says

$$(P(1) \wedge \forall k(P(k) \rightarrow P(k + 1))) \rightarrow \forall nP(n).$$

A useful analogy to keep in mind is stacking dominoes to topple. How can we prove that the n th domino topples for all n ? First, we need to know that the first domino falls. A bit of thought shows that this is not enough. We need to additionally know that the dominoes are stacked expertly. Namely, they are stacked in such a way that if the k th domino falls, then the $(k + 1)$ st domino falls.

Theorem 4.1.1 (Principle of Mathematical Induction). *Let $\{P(n): n \geq n_0\}$ be a family of propositions such that*

- (1) $P(n_0)$ is true.
- (2) $P(k)$ implies $P(k + 1)$, for $k \geq n_0$.

Then $P(n)$ is true for all $n \geq n_0$.

Follow these steps to write a proof by induction.

- (1) State that we are using induction. To be extra clear, tell the reader on what we are doing induction. A common way of saying this is “We proceed by induction on . . .”
- (2) Define $P(n)$.
- (3) **basis step** or **base case**: Prove $P(n_0)$ is true.
- (4) **inductive step**: For fixed (generic) $k \geq n_0$, assume $P(k)$ is true. This is known as assuming the **inductive hypothesis**. Prove $P(k + 1)$ is true.

Example 4.1.2. Prove

$$1 + 2 + \cdots + n = \frac{n(n + 1)}{2},$$

for all $n \geq 1$.

Proof. Let $P(n)$ be the proposition

$$P(n) = “1 + 2 + \cdots + n = \frac{n(n + 1)}{2}.”$$

We proceed by induction on n .

Basis step: $P(1)$ is the statement $1 = \frac{1(1+1)}{2}$, which is clearly true.

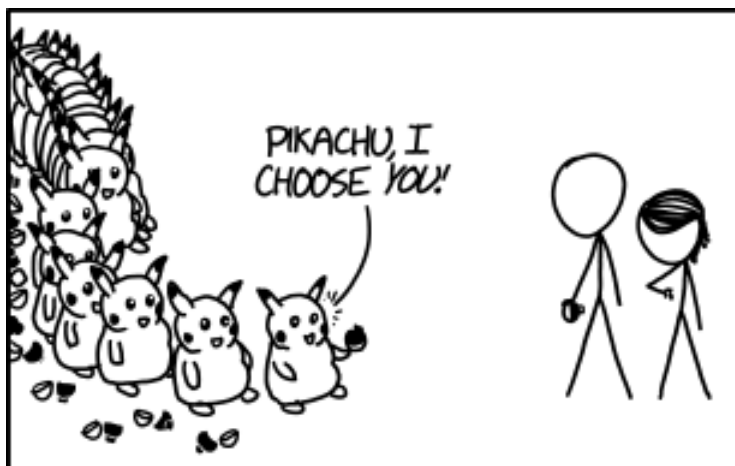


Figure 4.1.1. xkcd: Win By Induction. (<https://xkcd.com/1516/>) This would be bad enough, but every 30th or 40th pokéball has TWO of them inside.

Inductive step: Fix $k \geq 1$. Assume $P(k)$ is true. This is the inductive hypothesis. Specifically, we assume

$$1 + 2 + \cdots + k + (k + 1) = \frac{k(k + 1)}{2}.$$

Then add $k + 1$ to both sides.

$$\begin{aligned} 1 + 2 + \cdots + k + (k + 1) &= \frac{k(k + 1)}{2} + (k + 1) \\ &= \frac{k(k + 1)}{2} + \frac{2(k + 1)}{2} \\ &= \frac{k(k + 1) + 2(k + 1)}{2} \\ &= \frac{(k + 1)(k + 2)}{2}. \end{aligned}$$

Thus $P(k + 1)$ is true.

Therefore by mathematical induction,

$$1 + 2 + \cdots + n = \frac{n(n + 1)}{2},$$

for all $n \geq 1$. □

Example 4.1.3. Prove $n^3 - n$ is divisible by 3 for all $n \geq 1$.

Proof. Let $P(n)$ be the proposition

$$P(n) = "n^3 - n \text{ is divisible by 3.}"$$

We proceed by induction on n .

Basis step: $P(1)$: We see that $1^3 - 1 = 0$, and 0 is divisible by 3.

Inductive step: Fix $k \geq 1$. Assume $P(k)$ is true. Then $k^3 - k$ is divisible by 3, so there exists an integer ℓ such that $k^3 - k = 3\ell$. We want to show $(k+1)^3 - (k+1)$ is divisible by 3.

$$\begin{aligned} (k+1)^3 - (k+1) &= k^3 + 3k^2 + 3k + 1 - k - 1 \\ &= k^3 - k + 3k^2 + 3k - k \\ &= 3\ell + 3(k^2 + k) && \text{by inductive hypothesis} \\ &= 3(\ell + k^2 + k). \end{aligned}$$

Since ℓ and k are integers, $\ell + k^2 + k$ is an integer. Thus $(k+1)^3 - (k+1)$ is divisible by 3.

Therefore by mathematical induction, $n^3 - n$ is divisible by 3 for all positive integers n . \square

Example 4.1.4 (Tower of Hanoi). The *Tower of Hanoi* is a puzzle game consisting of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

- (1) Only one disk can be moved at a time.
- (2) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e., a disk can only be moved if it is the uppermost disk on a stack.
- (3) No disk may be placed on top of a smaller disk.

See the WIKIPEDIA article for more details.

https://en.wikipedia.org/wiki/Tower_of_Hanoi

Let $\text{Hanoi}(n)$ denote the Tower of Hanoi game with n disks. Once we have a winning strategy for $\text{Hanoi}(k)$ for some positive integer k , we can get a winning strategy for $\text{Hanoi}(k+1)$ as follows.

- (1) Use strategy to move the smallest k disks to an empty rod.
- (2) Move largest disk to the remaining empty rod.
- (3) Use strategy to move the smallest k disks on top of the largest disk.

If there are m_k moves in the $\text{Hanoi}(k)$ winning strategy, then there are $m_k + 1 + m_k = 2m_k + 1$ moves in the $\text{Hanoi}(k+1)$ strategy described above. It is clear that $m_1 = 1$. Then it's easy to see that $m_2 = 3$, $m_3 = 7$, $m_4 = 15$, \dots . It looks like $m_k = 2^k - 1$. Let's prove it by induction.

Proof. Let $P(n)$ be the proposition “ $m_n = 2^n - 1$.”

We proceed by induction on n .

Basis step: $P(1)$: It is clear that $m_1 = 1$.

Inductive step: Fix $k \geq 1$. Assume $P(k)$ is true so that $m_k = 2^k + 1$. Then

$$\begin{aligned} m_{k+1} &= 2m_k + 1 \\ &= 2(2^k - 1) + 1 && \text{inductive hypothesis} \\ &= 2^{k+1} - 1. \end{aligned}$$

Thus $P(k + 1)$ is true.

Therefore, by mathematical induction, $m_n = 2^n - 1$ for all $n \geq 1$. \square

Remark 4.1.5. A proof almost identical to the one above can be used to prove that the winning strategy for Hanoi(n) with the minimum number of moves has $2^n - 1$ moves.

Example 4.1.6. Find the flaw in the following proof that any set with n people are all the same age.

Proof. We proceed by induction. Let $P(n)$ be the proposition

$$P(n) = \text{“Any set of } n \text{ people are all the same age.”}$$

We proceed by induction on n .

Basis step: $P(1)$: It is clear that in any set with 1 person, they are the same age.

Inductive step: Fix $k \geq 1$. Assume $P(k)$ is true so that any set of k people are the same age. Let S be a set of $k + 1$ people. We have to show that everyone in S is the same age. Pick one person X in S . Let $T = S - \{X\}$. Then $|T| = k$, so by the inductive hypothesis everyone in T is the same age, call it a . Pick another person X' in S , $X \neq X'$. Let $T' = S - \{X'\}$. Then $|T'| = k$, so by the inductive hypothesis everyone in T' is the same age, call it a' . Notice that we have $a = a'$ since T and T' have some members in common. Furthermore, $T \cup T' = S$. Thus everyone in S is the same age.

Therefore by mathematical induction, any set of n people are all the same age. \square

What is the error in the “proof”? When we choose $X' \neq X$, we are assuming $k \geq 2$. Furthermore, when we assert T and T' have members in common, we assume $k \geq 3$. This shows that we need to be really careful that the basis step is correctly chosen so that the inductive step can work.

Exercises

1. State precisely the Principle of Mathematical Induction. Be sure to set up any notation that is required.
2. Complete the steps below to prove that for every positive integer n ,

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

I have written in **bold** part of what you need to write. You should complete the proof.

- (a) What is the proposition $P(n)$? **Let $P(n)$ be the proposition**
 - (b) **Basis step:**
 - (i) What is the statement $P(1)$?
 - (ii) Prove $P(1)$ is true.
 - (c) **Inductive step:**
 - (i) State the inductive hypothesis. **Fix $k \geq 1$. Assume**
 - (ii) What is it we want to prove, assuming the inductive hypothesis? **We want to show**
 - (iii) Prove it. Be clear where you are using the inductive hypothesis.
 - (iv) Write **This completes the inductive step.**
 - (d) State the conclusion. **By mathematical induction,**
3. For each positive integer n , let $P(n)$ be the proposition

$$2^0 + 2^1 + 2^2 + \cdots + 2^n = 2^{n+1} - 1.$$

Follow the steps below to prove $P(n)$ is true for all positive integers n .

- (a) **Basis step:**
 - (b) **Inductive step:**
 - (c) **Conclusion:**
4. Prove this extension of De Morgan's law. Let p_1, p_2, \dots, p_n be propositions. Prove that

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n.$$

5. Find the flaw in the following proof that $n^2 + 5n + 1$ is even for all $n \geq 0$.

Proof. Let $P(n)$ be the proposition " $n^2 + 5n + 1$ is even." Fix $k \geq 0$. Assume $P(k)$ is true so that $k^2 + 5k + 1$ is even. We want to show $(k+1)^2 + 5(k+1) + 1$ is even. We compute

$$\begin{aligned} (k+1)^2 + 5(k+1) + 1 &= k^2 + 2k + 1 + 5k + 5 + 1 \\ &= (k^2 + 5k + 1) + 2(k+3). \end{aligned}$$

Since $k^2 + 5k + 1$ is even by inductive hypothesis and $2(k+3)$ is visibly even, $(k+1)^2 + 5(k+1) + 1$ is even.

Thus by induction, $n^2 + 5n + 1$ is even for all $n \geq 0$. □

6. Find the flaw in the following proof that $2^n \leq n + 1$ for all $n \geq 1$.

Proof. Let $P(n)$ be the proposition “ $2^n \leq n + 1$ ”.

We proceed by induction on n .

Base case: $2^1 = 2$ and $1 + 1 = 2$, so $2^1 \leq 1 + 1$.

Inductive step: Fix $k \geq 1$. Assume $P(k)$ is true. i.e., assume $2^k \leq k + 1$.

$$2^{k+1} \leq (k + 1) + 1$$

$$2 \cdot 2^k \leq k + 2$$

$$2^k \leq \frac{k}{2} + 1 \leq k + 1,$$

which is true by inductive hypothesis. Thus by induction, $2^n \leq n + 1$ for all $n \geq 1$. \square

7. Prove that for all positive integers n ,

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

More generally, fix real numbers a and r , with $r \neq 1$. Prove that for every positive integer n ,

$$a + ar^j + \cdots + ar^n = \frac{ar^{n+1} - a}{r - 1}.$$

8. If you have had calculus, prove the power rule for positive exponents. Specifically, prove that for every positive integer n ,

$$\frac{d}{dx}(x^n) = nx^{n-1}.$$

(Hint: Use induction on n and the Product rule, writing $x^n = x \cdot x^{n-1}$.)

9. Prove that for every positive integer n ,

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

10. Prove that for every positive integer n ,

$$1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}.$$

11. Prove for every positive integer n ,

$$1^3 + 2^3 + 3^3 + \cdots + n^3 = \left(\frac{n(n+1)}{2}\right)^2.$$

12. Prove that for every positive integer n ,

$$1 \cdot 1! + 2 \cdot 2! + \cdots + n \cdot n! = (n+1)! - 1.$$

13. Prove that $n < 2^n$, for every positive integer n .

14. Prove that $11^n - 4^n$ is divisible by 7, for all $n \geq 0$. Redo the proof using congruences instead of induction.

15. Prove that for every positive integer n ,

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

4.2. Strong induction

Goals. To explain how to construct proofs of a variety of theorems using strong induction and the well-ordering property.

Strong induction is another form of mathematical induction that can often be used when we cannot easily use mathematical induction. For **strong induction**, replace the inductive step in the Principle of Mathematical Induction by a stronger assumption. In its most basic form, there are two parts:

- (1) Prove $P(1)$ is true.
- (2) Prove that for all positive integers k , if $P(j)$ is true for all $1 \leq j \leq k$, then $P(k+1)$ is true.

Then symbolically, mathematical induction says

$$(P(1) \wedge \forall k((P(1) \wedge P(2) \wedge \cdots \wedge P(k)) \rightarrow P(k+1))) \rightarrow \forall n P(n).$$

Theorem 4.2.1 (Principle of Strong Induction). *Let $\{P(n) : n \geq n_0\}$ be a family of propositions such that*

- (1) $P(n_0)$ is true.
- (2) $P(n_0) \wedge P(n_0+1) \wedge \cdots \wedge P(k)$ implies $P(k+1)$, for $k \geq n_0$.

Then $P(n)$ is true for all $n \geq n_0$.

For the inductive step, we need to show “ $P(n_0) \wedge P(n_0+1) \wedge \cdots \wedge P(k)$ implies $P(k+1)$.” For fixed (generic) $k \geq n_0$, assume $P(j)$ is true for all $n_0 \leq j \leq k$. Prove $P(k+1)$ is true.

Example 4.2.2. Prove that every positive integer greater than 1 can be written as a product of primes.

Proof. Let $P(n)$ be the proposition “ n can be written as a product of primes.” We want to show $P(n)$ is true for all $n \geq 2$. We proceed by induction on n .

Basis step: $P(2)$ is true since 2 is prime.

Inductive step: Fix $k \geq 2$. Suppose $P(j)$ is true for all $0 \leq j \leq k$ so that j can be written as a product of primes when $0 \leq j \leq k$. We want to show $k+1$ can be written as a product of primes. There are two cases to consider.

$k + 1$ is prime: In this case, $P(k + 1)$ is trivially true.

$k + 1$ is composite: In this case, there exist integers a and b such that $a \geq 2$, $b \geq 2$, $ab = k + 1$. This implies $a < k + 1$ and $b < k + 1$, so $2 \leq a \leq k$ and $2 \leq b \leq k$. Then by the inductive hypothesis, a and b can be written as products of primes. Thus $k + 1 = ab$ can be written as a product of primes.

Therefore by strong induction, every integer greater than or equal to 2 can be written as a product of primes. \square

Example 4.2.3. When we use the inductive hypothesis, it is important to verify that we are in the range where we can use it. See the example below.

Find the flaw in the following proof that $3^n = 1$ for every nonnegative integer n .

Let $P(n)$ be the proposition $3^n = 1$.

Basis step: $P(0)$ is true since $3^0 = 1$.

Inductive step: (Uses strong induction.) Fix $k \geq 0$. Assume $P(j)$ is true for all $0 \leq j \leq k$. In particular, we assume that $3^j = 1$ for all $0 \leq j \leq k$. We wish to show that $3^{k+1} = 1$. We compute

$$3^{k+1} = 3^k \cdot 3^1 = \frac{3^k \cdot 3^k}{3^{k-1}} = \frac{1 \cdot 1}{1} = 1,$$

since by the inductive hypothesis, $3^k = 1$ and $3^{k-1} = 1$. Thus, by induction, $3^n = 1$ for all $n \geq 0$.

The induction principle follows from the *well-ordering property* of the integers, which says that every non-empty subset of the non-negative integers has a least element.

Remark 4.2.4. This well-ordering property is not true for subsets of the real numbers, so we cannot use these induction techniques to prove things like $P(x)$ is true for all real positive numbers x .

Example 4.2.5. Prove that every amount of postage of 12 cents or more can be formed using 4-cent and 5-cent stamps.

Proof. Let $P(n)$ be the proposition

$P(n) =$ “ n cents can be formed using 4-cent and 5-cent stamps.”

We want to prove $P(n)$ is true for $n \geq 12$. We proceed by induction on n .

Basis step:

$$12 = 3 \cdot 4 + 0 \cdot 5$$

$$13 = 2 \cdot 4 + 1 \cdot 5$$

$$14 = 1 \cdot 4 + 2 \cdot 5$$

$$15 = 0 \cdot 4 + 3 \cdot 5$$

Inductive step: Fix $k \geq 15$. Suppose $P(j)$ is true for all $12 \leq j \leq k$ so that j cents can be formed using 4-cent and 5-cent stamps for each j such that $12 \leq j \leq k$. We want to form $k + 1$ cents. Consider $j = (k + 1) - 4 = k - 3$. Then $j \leq k$. Since $k \geq 15$, we have $j = k - 3 \geq 15 - 3$. Thus $12 \leq j \leq k$. Then by the inductive hypothesis, we can form j cents using 4-cent and 5-cent stamps. Just add another 4-cent stamp to get $j + 4 = k - 3 + 4 = k + 1$ cents. Thus $P(k + 1)$ is true.

Therefore by strong induction, $P(n)$ is true for all $n \geq 12$. \square

Example 4.2.6. Recall the Fibonacci sequence is defined by $f_0 = 0$, $f_1 = 1$, and

$$f_{n+1} = f_n + f_{n-1}, \quad \text{for } n \geq 2.$$

Prove that $f_n \leq 2^n$ for all $n \geq 0$.

Proof. Let $P(n)$ be the proposition " $f_n \leq 2^n$." We proceed by induction on n .

Basis step: It is clear that $f_0 = 0 \leq 2^0$ and $f_1 = 1 \leq 2^1$, so $P(0)$ and $P(1)$ are true.

Inductive step: Fix $k \geq 1$. Assume $P(j)$ is true for $0 \leq j \leq k$ so that $f_j \leq 2^j$ for $0 \leq j \leq k$. We want to prove $f_{k+1} \leq 2^{k+1}$. We have

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} \\ &\leq 2^k + 2^{k-1} \\ &\leq 2 \cdot 2^k \\ &= 2^{k+1}. \end{aligned}$$

Thus $P(k + 1)$ is true.

Therefore by strong induction, $f_n \leq 2^n$ for all $n \geq 0$. \square

The validity of mathematical induction and strong induction follows from a fundamental axiom of the integers known as the **well-ordering property**. Note that while we list the well-ordering property below as a theorem, it is in fact an axiom (self-evident truth) that we assume in the precise construction of the set of integers.

Theorem 4.2.7 (Well-Ordering Property). *Every nonempty set of nonnegative integers has a least element.*

Why does the well-ordering property imply that induction is valid?

Proof. For simplicity, just think about the most basic form, as described before Theorem 4.1.1. We proceed by contradiction. Suppose not. Then the set S of positive integers for which $P(n)$ is false is nonempty. By the well-ordering property, S must have a least element. Let m be the least



Figure 4.2.1. xkcd: Set Theory. (<https://xkcd.com/982/>) Proof of Zermelo's well-ordering theorem given the Axiom of Choice: 1: Take S to be any set. 2: When I reach step three, if S hasn't managed to find a well-ordering relation for itself, I'll feed it into this wood chipper. 3: Hey, look, S is well-ordered.

element in S . Then m is not 1, since $P(1)$ is true by the basis step. Because m is positive and not 1, we must have that m is at least 2. Then $m - 1$ is a positive integer less than m , and $P(m - 1)$ is true. By the inductive step, we know that $P(m - 1)$ implies $P(m)$, so $P(m)$ must be true as well. Contradiction! Thus induction is valid.

The proof the more general case as well as strong-induction is similar. \square

Exercises

1. State precisely the Principle of Strong Induction. Be sure to set up any notation that is required.
2. State precisely the Well-Ordering Property. Be sure to set up any notation that is required.
3. Complete the steps below to prove that every amount of postage of 12 cents or more can be formed using 4-cent and 5-cent stamps.
 - (a) What is the proposition $P(n)$? **Let $P(n)$ be the proposition**
 - (b) **Basis step:**
 - (i) What is the statement $P(12)$?
 - (ii) Prove $P(12)$ is true.

- (iii) In this case, the proof is simpler if we use strong induction. State and prove $P(13)$, $P(14)$, and $P(15)$.

(c) **Inductive step:**

- (i) State the inductive hypothesis. **Fix** $k \geq \dots$. **Assume** \dots for all $\dots \leq j \leq k$. Be careful here. We gain the additional advantage of additional cases in the basis step by forcing k to be larger, and allowing j to go down as much as possible.
- (ii) What is it we want to prove, assuming the inductive hypothesis? **We want to show** \dots .
- (iii) Prove it. Be clear where we are using the inductive hypothesis. Verify that we are working in a range where the inductive hypothesis is assumed to hold.
- (iv) Write **This completes the inductive step.**

(d) State the conclusion. **By mathematical induction, \dots**

4. Suppose $a_1 = 10$, $a_2 = 5$, and $a_n = 2a_{n-1} + 3a_{n-2}$ for $n \geq 3$. Use strong induction to prove that 5 divides a_n for $n \geq 1$.
5. Let n be a positive integer. Show that any $2^n \times 2^n$ chessboard with 1 square removed can be tiled using L-shaped pieces, where these pieces cover three squares at a time, as shown below.



6. Let f_n denote the n^{th} Fibonacci number. Prove that

$$f_n = \frac{\phi^n - (1 - \phi)^n}{\sqrt{5}}, \quad \text{where } \phi = \frac{1 + \sqrt{5}}{2}.$$

(Hint: Since ϕ satisfies $\phi^2 - \phi - 1 = 0$, we have $\phi^2 = \phi + 1$ and $(1 - \phi)^2 = 2 - \phi$.)

7. Prove that the power set $\mathcal{P}(S)$ of a finite set S has cardinality

$$|\mathcal{P}(S)| = 2^{|S|}.$$

(Hint: Use induction on the size $|S|$ of S . Fix an element a in S , and count the subsets of S containing a and the subsets not containing a . Note: A direct proof of this is given in Theorem 5.1.14.)

8. Prove that if a sequence $\{a_n\}$ satisfies

$$a_{n+1} = \frac{a_n}{a_n + 1}$$

then

$$a_n = \frac{a_0}{na_0 + 1}.$$

Counting

When angry count to ten before you speak. If very angry, count to one hundred.

Thomas Jefferson (1743–1826)

We learn to count at a young age. It is something that is fundamental and basic. There is archaeological evidence suggesting humans have been counting for over 50,000 years. In this chapter, we define precisely what it means to *count* something. Then we develop advanced counting techniques and look at some of the implications.

5.1. Basics of counting

Goals. To introduce basic counting rules and to show how they are used to solve a variety of counting problems.

Combinatorics is the study of arrangements of objects. We want to count the number of ways to do certain things.

First we make precise what it means to *count* something.

Definition 5.1.1. Let A a finite nonempty set. To *count* A means to construct a bijection $\phi: A \rightarrow \{1, 2, \dots, |A|\}$, where $|A|$ is the cardinality of A .

Recall that the cardinality of a finite set is defined earlier in §2.1.3. This definition of counting is exactly what we show children when we teach them to count to answer “how many” type questions, though not in this language.

Specifically, we teach them to construct a bijection in order to find the size of a set. For example, imagine teaching a child to count the number of hearts (♥) shown below:



Most likely, you pointed to each heart, one-by-one, and recited: “one, two, three, four, . . . , thirteen, fourteen.” In doing so, you created a bijection between the set of hearts and $\{1, 2, \dots, 14\}$. From this, you can deduce that there are 14 hearts.

Notice that we do not need to have constructed the same bijection in order to get the right number for the cardinality. The key point is that we create a function that is both injective and surjective. Do you see why?

We can generalize the definition for counting and cardinality to include some infinite sets. An infinite set is any set that does not have a finite cardinality. It turns out, there are different sizes of infinity.

Definition 5.1.2. Two sets A and B have the same *cardinality*, denoted $|A| = |B|$ if there is a bijection from A to B .

Definition 5.1.3. A *countable* set is either finite or has the same cardinality as the set of positive integers. An *uncountable* set is any set that is not countable.

Notice that this gives at least two different “sizes” for infinite sets. Some sets are infinite, but we can count them by constructing a bijection to the positive integers. Other infinite sets are so large that we cannot even construct such a bijection to the positive integers.

Example 5.1.4. The set of integers \mathbb{Z} is a countable set. To show this, we need to construct a bijection from \mathbb{Z} to $\{1, 2, \dots\}$. Consider the function $f: \mathbb{Z} \rightarrow \{1, 2, \dots\}$ defined by

$$f(x) = \begin{cases} 2x + 1 & \text{if } x \geq 0, \\ -2x & \text{if } x < 0. \end{cases}$$

So $f(0) = 1$, $f(1) = 3$, $f(3) = 7$, . . . , and $f(-1) = 2$, $f(-2) = 4$, $f(-3) = 6$, In other words, f sends the nonnegative integers to the odd integers, and the negative integers are sent to the even integers.

Let’s show that f is bijective. First, to see that f is surjective, fix a generic element n in $\{1, 2, \dots\}$. If n is even, then $-\frac{n}{2}$ is a negative integer, and so

$$f\left(-\frac{n}{2}\right) = -2\left(-\frac{n}{2}\right) = n.$$


If n is odd, then $\frac{n-1}{2}$ is a positive integer, and so

$$f\left(\frac{n-1}{2}\right) = 2\left(\frac{n-1}{2}\right) + 1 = n.$$

Thus f is surjective.

To see f is injective, fix generic integers a and \hat{a} such that $f(a) = f(\hat{a})$. If $f(a)$ is even, then $f(\hat{a})$ is also even, and so both a and \hat{a} are negative. Then $f(a) = -2a$, and $f(\hat{a}) = -2\hat{a}$. Since $f(a) = f(\hat{a})$, we have $-2a = -2\hat{a}$ so $a = \hat{a}$. Now suppose $f(a)$ is odd. Then $f(\hat{a})$ is also odd, and so both a and \hat{a} are nonnegative. Then $f(a) = 2a + 1$, and $f(\hat{a}) = 2\hat{a} + 1$. Since $f(a) = f(\hat{a})$, we have $2a + 1 = 2\hat{a} + 1$ so $a = \hat{a}$. Thus f is injective.

Since f is injective and surjective, f is bijective and so \mathbb{Z} is countable.

 Example 5.1.4 is a bit counterintuitive. It shows that \mathbb{Z} can be put in bijection with a proper subset of itself. In particular, \mathbb{Z} has the same size as $\{1, 2, \dots\}$. This hints at some of the strange things that can happen with counting infinite sets.

Since the inverse of a bijective function is bijective, we can show an infinite set is countable by constructing a bijection from the set of positive integers to the set. Such a bijection is a sequence that includes all of the elements of A , listed with no repeats. In other words, an infinite set is countable if and only if it is possible to list all of the elements in a sequence with no repeats. In fact, we can relax the “no repeats” part, since if we do have a sequence with repeats, we can easily extract a subsequence without repeats. For example, if we are given the sequence

$$1, 2, 3, 4, 5, 5, 6, 7, 7, 8, 9, 9, 10, \dots$$

we instead consider the sequence

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \dots$$

In Example 5.1.4 that shows the set of integers is countable, the sequence that is inverse the bijection given is

$$0, -1, 1, -2, 2, -3, 3, \dots$$

Sometimes it is easier to think of a sequence instead of a formula for the function.

Since the composition of bijective functions is bijective, to show an infinite set A is countable it is enough to find a bijection from A to \mathbb{Z} . If we have such a bijection, we compose it with the bijection from \mathbb{Z} to $\{1, 2, \dots\}$ given in Example 5.1.4 to get the desired bijection from A to $\{1, 2, \dots\}$. Using this idea, it is clear that the set of even integers $2\mathbb{Z}$ is countable, since $f: 2\mathbb{Z} \rightarrow \mathbb{Z}$ given by $f(x) = \frac{x}{2}$ is a bijection from $2\mathbb{Z}$ to \mathbb{Z} . A similar argument shows the set of odd integers is countable.

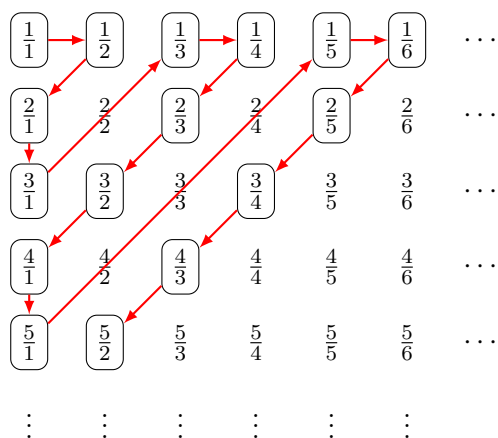


Figure 5.1.1. The positive rationals are countable.

What about the set of rational numbers? Surely there are more rational numbers than integers? It turns out, the set of rational numbers is also countable.

Theorem 5.1.5. *The set of rational numbers is countable.*

Proof. It suffices to show the set of positive rational numbers is countable. (Why? Exercise.)

Every positive rational number is the quotient $\frac{p}{q}$ of two positive integers p and q . We arrange the positive rational numbers by listing those with denominator $q = 1$ in the first row, those with denominator 2 in the second row, and so on. The rational numbers with denominator k are listed in the k th row, as shown in Figure 5.1.1. From the figure, we create a sequence by looking at the diagonals: list $\frac{p}{q}$ with $p + q = 2$, followed by those with $p + q = 3$, followed by with $p + q = 4$, and so on as shown in Figure 5.1.1. The numbers we keep for the sequence are circled. The uncircled numbers are those we leave out because they are already listed. Because all the positive rational numbers are listed once, we have shown the set of positive integers is countable. The exercise completes the proof by extending this bijection to a bijection from \mathbb{Z} to \mathbb{Q} . \square

One may start to think that every infinite set is countable. In fact, there are sets that are larger. In 1879, Georg Cantor produced a proof using a technique now known as the *Cantor diagonalization argument*, that proves the set of real numbers is not countable.

Theorem 5.1.6. *The set of real numbers is uncountable.*

Proof. It suffices to prove the interval $e(0, 1)$ is uncountable. We prove this by contradiction. Suppose the interval $(0, 1)$ is countable. Then there is a sequence that contains all of the real numbers in $(0, 1)$.

$$\begin{aligned} a_1 &= 0.a_{1,1}a_{1,2}a_{1,3}a_{1,4}a_{1,5} \dots \\ a_2 &= 0.a_{2,1}a_{2,2}a_{2,3}a_{2,4}a_{2,5} \dots \\ a_3 &= 0.a_{3,1}a_{3,2}a_{3,3}a_{3,4}a_{3,5} \dots \\ &\vdots \quad \vdots \end{aligned}$$

The i th term in the sequence has a decimal expansion with $a_{i,j}$ in the j th digit after the decimal point. For example, if the third term was $0.536831\dots$, then $a_{3,1} = 5$, $a_{3,2} = 3$, $a_{3,3} = 6$, \dots

By assumption, this sequence contains every real number in the interval $(0, 1)$. To reach a contradiction, it suffices to produce a real number in the interval $(0, 1)$ that is provably not a term in the sequence. Consider the number

$$d = 0.d_1a_2a_3a_4a_5 \dots,$$


where d_j is defined by

$$d_j = \begin{cases} 3 & \text{if } a_{j,j} \neq 3, \\ 4 & \text{if } a_{j,j} = 3. \end{cases}$$

We have that $d \neq a_1$, since they differ in the first digit after the decimal, $d_1 \neq a_{1,1}$. Similarly, $d \neq a_2$, since they differ in the second digit after the decimal, $d_2 \neq a_{2,2}$, and so on. In general, d cannot be the k th term in the sequence, since $d_k \neq a_{k,k}$. See boxed digits below.

$$\begin{aligned} a_1 &= 0.\boxed{a_{1,1}}a_{1,2}a_{1,3}a_{1,4}a_{1,5} \dots \\ a_2 &= 0.a_{2,1}\boxed{a_{2,2}}a_{2,3}a_{2,4}a_{2,5} \dots \\ a_3 &= 0.a_{3,1}a_{3,2}\boxed{a_{3,3}}a_{3,4}a_{3,5} \dots \\ &\vdots \quad \vdots \end{aligned}$$

In other words, d is not a term in the sequence. This contradicts the assumption that we have a sequence containing every real number. Thus there is no such sequence, and hence the interval $(0, 1)$ is uncountable. It follows that the set of real numbers is uncountable. \square

 There is a subtle point that not every real number has a unique decimal representation. For example, $0.5000\dots$ is the same real number as $0.4999\dots$. We avoid this issue by choosing d to have digits only involving 3 and 4 to be sure we are producing a number that has only one representation, so that we can be sure it is not in the sequence.

5.1.1. Product Rule.

Theorem 5.1.7 (Product Rule). *Suppose a procedure can be broken down into a sequence of two tasks. If there are n_1 ways to do the first task, and for each of these ways there are n_2 ways to do the second task, then there are n_1n_2 ways to do the procedure.*

Example 5.1.8. Suppose auditorium chairs are labelled with an uppercase letter followed by a positive integer not exceeding 100. How many different labels are possible?

We can view this problem as counting the ways to assign a label to a chair. We proceed by constructing tasks.

Task 1. Assign a letter: There are 26 ways to assign the letter.

Task 2. Assign an integer: For each choice of letter, there are 100 ways to assign the integer.

By the Product Rule, there are $26 \cdot 100 = 2600$ different ways to assign a label. Hence there are 2600 different labels.

The Product Rule generalizes to m tasks. Suppose a procedure can be carried out by performing tasks T_1, T_2, \dots, T_k tasks in sequence. If each task T_i can be done in n_i ways regardless of how the previous tasks were done, then there are $n_1n_2 \cdots n_m$ ways to carry out the procedure.

Example 5.1.9. How many bit strings of length 8 are there?

We just have to choose each bit, so there are 8 tasks. Each bit can be 0 or 1, so there are 2 ways to choose each bit. The Product Rule says there are $2^8 = 256$ such bit strings.

Example 5.1.10. How many bit strings of length 8 that start with 1 are there?

The first bit must be 1. We are left with choosing the remaining bits, so there are 7 tasks. There are 2 ways to choose each of the remaining 7 bits. The Product Rule says there are $2^7 = 128$ such bit strings.

Example 5.1.11. How many bit strings of length 8 that end with 00 are there?

The last two bits must be 00. We are left with choosing the remaining bits, so there are 6 tasks. There are 2 ways to choose each of the remaining 6 bits. The Product Rule says there are $2^6 = 64$ such bit strings.

Example 5.1.12. How many bit strings of length 8 that start with 1 and end with 00 are there?

The first bit must be 1 and the last two must be 00. We are left with choosing the remaining bits, so there are 5 tasks. There are 2 ways to choose

each of the remaining 5 bits. The Product Rule says there are $2^5 = 32$ such bit strings.

Example 5.1.13. Suppose there are 22 people in the class. How likely is it that two share a birthday?

First, let's count the number of ways the birthdays could be arranged, assuming no one was born on February 29. We order the 22 people, and break this own into 22 tasks. The i th task is choosing a birthday for the i th person. There are 365 ways to complete each task, regardless of how previous tasks were done. By the Product Rule, the number of birthday arrangements is

$$365^{22} \approx 2.346621351 \times 10^{56}.$$

Next, let's compute the number of ways the birthdays could be arranged, assuming no one was born on February 29 and no two people share a birthday. Again, we break this into 22 tasks, where the i th task is choosing the birthday for the i th person. The difference is in the number of ways to complete each task. There are 365 ways to complete the first task. There are only 364 ways to complete the second task, since we cannot choose the same birthday as the first person. Notice that the number of ways does not depend on what day was chosen for the first person. Similarly, there are 353 ways to complete the second task, and so on. In general, there are $365 - (i - 1)$ ways to complete the i th task. By the Product Rule, the number of birthday arrangements so that no two share a birthday is

$$\underbrace{365 \cdot 364 \cdot 363 \dots 344}_{22 \text{ terms}} \approx 1.230344586 \times 10^{56}.$$

The likelihood of an event happening (assuming all possible events are equally likely) is the ratio of the number of ways the particular event can occur divided by the total number of events. We can compute that

$$\frac{1.230344586 \times 10^{56}}{2.346621351 \times 10^{56}} \approx 0.5243.$$

That means if birthdays are arranged randomly, there is a 52% chance that no two people in the class share a birthday. This is the largest gathering where that is greater than 50%. If we do the same computation with twice as many people, there is only a 6.7% chance that no one shares a birthday.

Theorem 5.1.14. *If a set A has cardinality n , then the cardinality of the power set of A is $|\mathcal{P}(A)| = 2^n$.*

Proof. Label the elements of A ,

$$A = \{a_1, a_2, \dots, a_n\}.$$

Note that we uniquely determine a subset of A by specifying the elements that are in the subset.

We specify a subset of A by completing n tasks, where the i th task is deciding whether or not a_i is in the subset. There are two ways to complete each task, regardless of how previous tasks are completed. By the Product Rule, there are 2^n different subsets of A . \square

Example 5.1.15. How many license plates can be made, if the license plate consists of 3 upper case letters followed by 4 one digit numbers.

We view this as a sequence of 7 tasks: choose a letter, choose a letter, choose a letter, choose a digit, choose a digit, choose a digit, and choose a digit. There are 26 letters and 10 digits, so there are 26 ways to choose a letter and 10 ways to choose a digit. The number of ways to complete each task does not depend on how the previous tasks were completed. Thus by the Product Rule, there are

$$26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 \cdot 10 = 175,760,000$$

different license plates.

Example 5.1.16. Let $A = \{a, b, c\}$, and let $B = \{1, 2, 3, 4, 5\}$.

- (1) How many functions $f: A \rightarrow B$ are there?

We need to choose a value for $f(x)$ for each x in A . There are 5 choices for $f(a)$, 5 choices for $f(b)$, and 5 choices for $f(c)$. By the Product Rule, there are $5 \cdot 5 \cdot 5 = 125$ functions from A to B .

- (2) How many injective functions $f: A \rightarrow B$ are there?

We need to choose a value for $f(x)$ for each x in A , but we need to do so in a way that produces an injective function. There are 5 choices for $f(a)$. There are 4 choices for $f(b)$, since we cannot choose the same value as was chosen for $f(a)$. Similarly, there are 3 choices for $f(c)$, since we must avoid $f(a)$ and $f(b)$. By the Product Rule, there are $5 \cdot 4 \cdot 3 = 60$ injective functions from A to B .

- (3) How many surjective functions $f: A \rightarrow B$ are there?

Since $|B| > |A|$, there are no surjective functions from A to B .

5.1.2. Sum Rule.

Theorem 5.1.17 (Sum Rule). *If a task can be done in either one of n_1 ways or n_2 ways, where none of the set of n_1 ways is the same as any of the set of n_2 ways. Then there are $n_1 + n_2$ ways to do the task.*

The Sum Rule can be generalized. If a task can be done in one of n_1 ways, or in one of n_2 ways, \dots , or in one of n_m ways, where none of the set of n_i ways is the same as any of the set of n_j ways, for all $i \neq j$, then there are $n_1 + n_2 + \dots + n_m$ ways to complete the task.

Example 5.1.18. Suppose there are 10 goats and 15 sheep in a certain village.

- (1) Suppose a gift of one animal (goat or sheep) is to be given to a visiting dignitary. How many ways are there to select a gift?

There are 10 ways to select a goat and 15 ways to select a sheep, and none of the ways that select a goat is the same as any of the ways that select a sheep. By the Sum Rule, there are $10 + 15 = 25$ ways to select the gift.

- (2) Suppose it is decided instead that the gift should consist of one goat and one sheep. How many ways are there to select a gift?

The procedure of selecting a gift can be broken down into two tasks: select the goat and select the sheep. The number of ways to select the sheep does not depend on which goat was selected. By the Product rule, there are $10 \cdot 15 = 150$ ways to select a gift.

5.1.3. Principle of Inclusion-Exclusion.

Definition 5.1.19 (Principle of Inclusion-Exclusion). If a task can be done in n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2 - k$, where k is the number of ways to do the task that is common to the n_1 and n_2 ways.

The Principle of Inclusion-Exclusion is sometimes called the **Subtraction Rule**.

Example 5.1.20. How many bit strings of length 8 are there that start with 1 or end with 00?

From Example 5.1.9, there are 2^7 bit strings of length 8 that start with 1. From Example 5.1.11, there are 2^6 bit strings of length 8 that end with 00. From Example 5.1.12, there are 2^5 bit strings that start with 1 and end with 00. By the Principle of Inclusion-Exclusion, there are $2^7 + 2^6 - 2^5 = 160$ bit strings that start with 1 or end with 00.

Example 5.1.21. Suppose there are 350 undergraduates in an auditorium. Of these, 220 are computer science majors, 147 are math majors, and 51 are double majoring in computer science and math. How many are neither computer science nor math majors?

First count the number of computer science or math majors. By the Principle Inclusion-Exclusion, there are $220 + 147 - 51 = 316$ computer science or math majors. Then there are $350 - 316 = 34$ students that are neither computer science nor math majors.

5.1.4. Division Rule.

Theorem 5.1.22 (Division Rule). Suppose we can complete a task using a procedure that can be carried out in n ways, and for each of these ways exactly d of the ways corresponds to the same way. Then there are $\frac{n}{d}$ ways to do the task.

Example 5.1.23. Seat four people at a circular table. Two seating arrangements are the same if each person has the same left and right neighbor. In other words, two seating arrangements are the same if one can be rotated into the other.

By the Product Rule there are $4 \cdot 3 \cdot 2 \cdot 1 = 24$ ways to arrange the four people around the table. For each way, exactly four give the same seating arrangement. By the Division Rule, there are $\frac{24}{4} = 6$ different seating arrangements.

Exercises

1. State precisely the following rules. Be sure to set up any notation that is required.
 - (a) Product Rule
 - (b) Sum Rule
 - (c) Principle of Inclusion-Exclusion
 - (d) Division Rule
2. How many bit strings of length eight contain exactly three 0s? Hint: Think about choosing locations for the 0s.
3. How many answer keys are possible for a twenty question multiple choice test, where each question has exactly six choices?
4. Suppose there are 350 undergraduates in an auditorium. Of these, 220 are computer science majors, 147 are math majors, and 51 are double majoring in computer science and math. How many are neither computer science nor math majors?
5. Let $A = \{a, b, c, d\}$, and let $N = \{1, 2, 3, 4, 5, 6, 7\}$.
 - (a) How many functions are there from A to N ? How many functions are there from N to A ?
 - (b) How many injective functions are there from A to N ? How many injective functions are there from N to A ?
 - (c) How many surjective functions are there from A to N ? It is a trickier problem to determine the number of surjective functions from N to A .
6. Suppose a multiple choice exam consists of 20 questions, each with choices A, B, C, D.
 - (a) How many possible answer keys are there?
 - (b) In how many ways can a student answer the questions on the test, if the student answers every question?
 - (c) In how many ways can a student answer the questions on the test, if the student student can leave answers blank?

7. A particular brand of shirt comes in a variety of colors (red, green, blue, black, pink, orange, purple) and sizes (XS, S, M, L, XL). How many different types of shirts are there?
8. How many license plates can be made using either three uppercase English letters (A–Z) followed by three digits (0–9) or four uppercase English letters (A–Z) followed by two digits (0–9)?
9. In how many ways can Alice arrange six of her dolls in a row if she has twenty-three dolls?
10. In how many ways can a photographer at a wedding arrange five people in a row from a group of twenty people, given that the bride must be in the picture?
11. In how many ways can Alice arrange twelve of her fifty-four books on a shelf, given that she must include *Strangers Have the Best Candy* and *The Joy of Chickens*?
12. Each user on a computer system has a password, where each character is an uppercase letter (A–Z) or a digit (0–9).
 - (a) How many possible 8 character passwords are there?
 - (b) How many possible 8 character passwords are there that do not use any digits?
 - (c) How many possible 8 character passwords are there, if each password must contain at least one digit?
13. Count the number of times that the letter F appears in the following sentence:

FINISHED FILES ARE THE RESULT OF YEARS OF SCIENTIFIC STUDY COMBINED WITH THE EXPERIENCE OF YEARS.
14. Let f be a bijection from the set of positive rational numbers $\mathbb{Q}_{>0}$ to the set of positive integers $\mathbb{Z}_{>0}$. Use f to construct a bijection from \mathbb{Q} to \mathbb{Z} . Use this to fill in the gap of the proof in Theorem 5.1.5. Hint: Extend the domain of f to be \mathbb{Q} : For a positive rational number r you know $f(r)$. What is a reasonable choice to pick to define $f(-r)$?

5.2. Pigeonhole Principle

Goals. To introduce the Pigeonhole Principle and show how to use it in enumeration and in proofs.

The basic idea is really simple. For example, if we have three boxes and want to put away four or more toys, there must be at least one box containing two or more toys. The technique can be used cleverly to prove statements that are not so trivial.

Theorem 5.2.1 (Pigeonhole Principle). *If k is a positive integer, and $k + 1$ or more objects are placed into k boxes, then there is at least one box containing two or more objects.*

Example 5.2.2. In any group of 27 English words, at least two must start with the same letter, since there are only 26 letters in the English alphabet.

Example 5.2.3. Assume no human has more than 200,000 hairs on his/her head. Since there are more than 200,000 people in Greensboro, there are at least two people in Greensboro with exactly the same number of hairs on their head.

Theorem 5.2.4. *If A and B are nonempty sets with $|B| = k$ and $|A| \geq k + 1$, then there are no injective functions from A to B .*

Proof. We show that any function from A to B is not injective. Make a box for each b in B . For each a in A , place a in box labelled b if $f(a) = b$. By the Pigeonhole Principle, there is a box with at least two elements in it. Thus there is a b in B with at least two different preimages. Thus f is not injective. \square

Example 5.2.5. For any integer n , there is a positive multiple of n that has only 0s and 1s in its decimal expansion.

n	multiple
2	10
3	111
4	100
5	10
6	1110
7	1001
\vdots	\vdots

How can we prove something like this in general?

There are n congruence classes modulo n . Use these as labels on a box. Consider the $n + 1$ integers

$$1, 11, 111, \dots, \underbrace{111 \cdots 1}_{n+1 \text{ times}}.$$

By the Pigeonhole Principle, there must be a congruence class with at least two of these integers in it. The difference of these two integers is 0 modulo n , hence divisible by n . Additionally, the difference only has 0s and 1s in its decimal expansion.

Theorem 5.2.6 (Extended Pigeonhole Principle). *Suppose n and k are positive integers. If n objects are placed into k boxes, then there is at least one box containing $\lceil \frac{n}{k} \rceil$ objects.*

Example 5.2.7. In a group of 100 people, there are at least $\lceil \frac{100}{12} \rceil = 9$ born in the same month.

Example 5.2.8. Suppose a bowl contains apples, oranges, and bananas. How many random selections are required to ensure we have at least two of the same fruit?

Label boxes by fruit types. Place fruit in the box identifying its type. Then there are 3 boxes, so if we select $3 + 1 = 4$ fruit, the Pigeonhole Principle says we will have at least two of the same type.

If we wanted at least five of the same type, the Extended Pigeonhole Principle says we need $\lceil \frac{n}{3} \rceil \geq 5$. That means we need $\frac{n}{3} > 4$, so $n > 12$. Thus if we select 13 fruit, we are guaranteed at least five of the same type.

Example 5.2.9. How many distinct random selections from $\{1, 2, 3, 4, 5, 6\}$ are required to guarantee at least one pair that adds up to 7?

Note that $1 + 6 = 2 + 5 = 3 + 4 = 7$ are the only ways to get a pair to add to 7. Label 3 boxes: $\{1, 6\}$, $\{2, 5\}$, and $\{3, 4\}$. By the Pigeonhole Principle, if we select $3 + 1 = 4$ numbers, one box must contain two or more elements. Thus we will have a pair that sums to 7 if we select 4 numbers.

Example 5.2.10. Suppose we deal cards from a standard deck of cards (jokers removed). How many cards must be dealt to guarantee that at least 3 of the same suit are chosen?

Imagine 4 boxes, labelled by the 4 suits: hearts (\heartsuit), diamonds (\diamondsuit), clubs (\clubsuit), spades (\spadesuit). By the Extended Pigeonhole Principle, if we deal n cards, we are guaranteed at least one suit has $\lceil \frac{n}{4} \rceil$ cards in it. We want $\frac{n}{4} > 2$, so we need $n > 2 \cdot 4 = 8$. Thus 9 cards will guarantee 3 of the same suit.

Exercises

1. State precisely the Pigeonhole Principle. Be sure to set up any notation that is required.
2. State precisely the Extended Pigeonhole Principle. Be sure to set up any notation that is required.
3. UNCG has 18,502 students. How many students at UNCG *must* share a birthday? Explain using the Extended Pigeonhole Principle. (You may assume no one was born on February 29.)

4. Suppose the final exam is graded on a scale from 0 to 100 points. How many students must be in the class to guarantee that at least two students receive the same score on the final exam?
 5. How many numbers must be selected from the set $\{1, 3, 5, 7, 9, 11, 13, 15\}$ to guarantee at least one pair of these numbers add up to 16?
 6. What is the minimum number of people required to be sure that at least four will have birthdays in the same month?
 7. A standard deck of cards consists of 52 cards. Each card is one of thirteen ranks (A, 2, 3, . . . , J, Q, K) and one of four suits (\clubsuit , \spadesuit , \heartsuit , \diamondsuit).
 - (a) How many cards must be selected to guarantee that at least three of the same suit are chosen?
 - (b) How many cards must be selected to guarantee that at least three of the same rank are selected?
 8. A standard deck of cards consists of 52 cards. Each card is one of thirteen ranks (A, 2, 3, . . . , J, Q, K) and one of four suits (\clubsuit , \spadesuit , \heartsuit , \diamondsuit).
 - (a) How many cards must be selected to guarantee that at least two hearts (\heartsuit) are selected?
 - (b) How many cards must be selected to guarantee that at least three spades (\spadesuit) are selected?
 - (c) How many cards must be selected to guarantee that at least two hearts (\heartsuit) and three spades (\spadesuit) are selected?
 9. Alice selects clips randomly from a bowl that contains ten large paper clips and thirty small paper clips.
 - (a) How many must she select to be sure of having at least three of the same size?
 - (b) How many must she select to be sure of having at least five of the same size?
 10. Show that there are at least seventeen people in Greensboro (population 285,000) with the same three initials, assuming everyone has a first, middle, and last initial.
 11. There are 38 different time periods during which classes can be scheduled. If there are 650 different classes, how many different rooms will be needed?
 12. Show that among any group of five integers, there are at least two with the same remainder when divided by 4.
 13. There are 50 baskets of apples. Each basket contains at least one apple and no more than 24 apples. Show that there are at least 3 baskets containing the same number of apples. If you use the Pigeonhole Principle or its extension, be sure to tell me what are the pigeons and what are the boxes. Hint: The apples are not pigeons. The baskets are not boxes.
-

5.3. Permutations and combinations

Goals. To introduce permutations and combinations, to solve counting problems using them, and to show how theorems are proved by combinatorial arguments.

5.3.1. Permutations.

Definition 5.3.1. A *permutation* of a set of distinct objects is an ordered arrangement of these objects.

Example 5.3.2. Let $S = \{x, y, z\}$. The permutations of S are

$$\{xyz, xzy, yxz, yzx, zxy, zyx\}.$$

Definition 5.3.3. An ordered arrangement of r elements of a set is called an *r -permutation*. The number of r -permutations of a set of size n is denoted $P(n, r)$.

Remark 5.3.4. If $|S| = n$, then an n -permutation is the same thing as a permutation.

Example 5.3.5. Let $S = \{a, b, c, d\}$.

(1) The 1-permutations of S are

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

Thus we see that $P(4, 1) = 4$.

(2) The 2-permutations of S are

$$\{ab, ac, ad, ba, bc, bd, ca, cb, cd, da, db, dc\}.$$

Thus we see that $P(4, 2) = 12$.

(3) The 3-permutations of S are

$$\begin{aligned} &\{abc, acb, bac, bca, cab, cba, \\ &abd, adb, bad, bda, dab, dba, \\ &acd, adc, cad, cda, dac, dca, \\ &bcd, bdc, cbd, cdb, dbc, dcb\}. \end{aligned}$$

Thus we see that $P(4, 3) = 24$.

(4) As an exercise, try to list the permutations of S . There are 24 of them.

Theorem 5.3.6. Let n and r be integers such that $0 \leq r \leq n$. Then

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

Proof. Constructing an r -permutation, can be broken down into a sequence of r tasks. First, we select the first term in the arrangement. There are n ways to do this. Then, we select the second term in the arrangement. There are $n - 1$ ways to do this because one of the ways is no longer valid as it was already used in the previous step. Similarly, there are $n - 2$ ways to select the third term, and so on. The result then follows by the Product Rule. \square

Remark 5.3.7. $0! = 1$, so $P(n, n) = n!$.

Example 5.3.8. Suppose there are eight runners in a race. How many different ways can we Gold/Silver/Bronze finishers, assuming no ties?

We want to count the number of 3-permutations of the runners. Since there are eight runners, we have

$$P(8, 3) = \frac{8!}{(8-3)!} = \frac{8!}{5!} = \frac{8 \cdot 7 \cdot 6 \cdot \cancel{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}}{\cancel{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}} = 336.$$

Remark 5.3.9. For n and k such that $0 \leq k \leq n$, cancellation gives

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

Example 5.3.10. Let $S = \{a, b, c, \dots, x, y, z\}$. The number of permutations of S is

$$P(26, 26) = 26! = 403,291,461,126,605,635,584,000,000.$$

Example 5.3.11. How many permutations of $S = \{a, b, c, \dots, x, y, z\}$ contain abc ?

We can view this as a different problem. We want permutations of

$$S' = \{abc, d, e, f, \dots, x, y, z\}.$$

Then $|S'| = 24$, and

$$P(24, 24) = 24! = 620,448,401,733,239,439,360,000.$$

Example 5.3.12. How many ways are there to arrange seven people in from a group of ten (including me) in a row if I need to be one of the seven?

We break this into tasks. First select the position for me. There are 7 choices. The remaining spots are ordered. We need a 6-permutation of the set of 9 remaining people. There are

$$P(9, 6) = \frac{9!}{(9-6)!} = 9 \cdot 8 \cdot 7 \cdot 6 \cdot 7 \cdot 6 = 60,480$$

such permutations. By the Product Rule, there are $7 \cdot 60480 = 423,360$ arrangements.

5.3.2. Combinations.

Definition 5.3.13. A *combination* of a set of distinct objects is an unordered selection of these objects. An unordered selection of r -elements from a set is called an *r -combination*. The number of r -combinations of a set of size n is denoted $C(n, r)$.

Example 5.3.14. Let $S = \{a, b, c, d\}$.

(1) The 1-combinations of S are

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

Thus we see that $C(4, 1) = 4$.

(2) The 2-combinations of S are

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

Thus we see that $C(4, 2) = 6$.

(3) The 3-combinations of S are

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

Thus we see that $C(4, 3) = 4$.

(4) The 4-combinations of S are

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

Thus we see that $C(4, 4) = 1$.

Remark 5.3.15. There is only one way to select all of the elements of S , so $C(n, n) = 1$, in general.

Theorem 5.3.16. Let n and r be integers such that $0 \leq r \leq n$. Then

$$C(n, r) = \frac{n!}{r!(n-r)!}.$$

Proof. We can carry out the last of listing the r -combinations by listing the r -permutations. By Theorem 5.3.6, there are $\frac{n!}{(n-r)!}$ ways to do this. For each of these ways, exactly $r!$ of the ways correspond to the same way. The result then follows by the Division Rule. \square

Example 5.3.17. How many poker hands (5 cards) are in a standard deck of 52 cards?

We want to count 5-combinations of cards. Since there are 52 cards, the number of 5-combinations is

$$C(52, 5) = \frac{52}{5!(52-5)!} = \frac{52}{5!47!} = \frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 2,598,960.$$

Selecting r elements from a set of size n to be in an r -combination is equivalent to selecting $n - r$ elements not to be in the r -combination. Thus the number of ways to do each must be the same. This gives the following theorem.

Theorem 5.3.18. *Let n and r be integers such that $0 \leq r \leq n$. Then*

$$C(n, r) = C(n, n - r).$$

Example 5.3.19. Suppose there are 9 males and 11 females in MAT 253. How many different final exam committees can be made if a committee consists of 3 males and 4 females?

The procedure of choosing a committee can be broken down into two tasks—selecting the male members and selecting the female members. The number of ways to select male members is

$$C(9, 3) = \frac{9!}{3!(9-3)!} = \frac{9!}{3!6!} = \frac{9 \cdot 8 \cdot 7}{3 \cdot 2 \cdot 1} = 84.$$

The number of ways to select female members is

$$C(11, 4) = \frac{11!}{4!(11-4)!} = \frac{11!}{4!7!} = \frac{11 \cdot 10 \cdot 9 \cdot 8}{4 \cdot 3 \cdot 2 \cdot 1} = 330.$$

Then by the Product Rule, the number of different committees is

$$84 \cdot 330 = 27,720.$$

Example 5.3.20. How many bit strings of length ten have exactly three 0s?

We can describe a bit string with exactly three 0s by specifying the location of the three 0s. There are ten possible places for them to go, and we cannot distinguish the three 0s, so the order does not matter. Thus the number of bit strings with exactly three 0s is

$$C(10, 3) = \frac{10!}{3!(10-3)!} = \frac{10!}{3!7!} = \frac{10 \cdot 9 \cdot 8}{3 \cdot 2 \cdot 1} = 120.$$

Example 5.3.21. Thirteen people on a soccer team show up for a game. Of the thirteen that show up, three are women. How many ways are there to choose ten players to take the field if at least one of these players must be a woman?

Which of the following computations is correct?

- (1) There are $C(3, 1) = 3$ ways to choose a woman. Then there are 12 remaining players, and we need to choose 9 of them. The number of ways to choose 9 from 12 is

$$C(12, 9) = \frac{12!}{9!(12-9)!} = \frac{12!}{9!3!} = \frac{12 \cdot 11 \cdot 10}{3 \cdot 2 \cdot 1} = 220.$$

By the Product Rule, the number of ways to choose a team with at least one woman is the product

$$C(3, 1)C(12, 9) = 3 \cdot 220 = 660.$$

- (2) There are 3 ways to choose a woman. Of the remaining 10 men, we need to choose 9. The number of ways to choose 9 from 10 is $C(10, 9) = 10$. By the Product Rule, the number of ways to choose a team with at least one woman is the product $3 \cdot 10 = 30$.
- (3) To pick a team, independent of gender, we need to choose 10 people from a group of 13. The number of ways to pick 10 from 13 is

$$C(13, 10) = \frac{13!}{10!(13-10)!} = \frac{13!}{10!3!} = \frac{13 \cdot 12 \cdot 11}{3 \cdot 2 \cdot 1} = 286.$$

There is $C(10, 10) = 1$ way to choose a team of all men. Therefore there are $286 - 1 = 285$ ways to choose a team with at least one woman.

The third method is correct. The second method counts the teams with exactly 1 female. It does not allow for more than one female. The first method counts certain teams more than once. For example if Alice is chosen first as part of the $C(3, 1)$, and then Beverly is chosen as part of the $C(12, 9)$, the team created is duplicated when Beverly is chosen first as part of the $C(3, 1)$ and Alice is chosen as part of the $C(12, 9)$. We can see furthermore that the value of 660 cannot be correct, as the total number of teams independent of gender is 286, as shown in the third method.

Example 5.3.22. How many ways are there for eight men and five women to stand in a line so that no two women stand next to each other? (Hint: First position the men and then consider the possible positions for the women.)

There are $P(8, 8) = 8!$ ways to arrange the men in order. There are 9 slots where a woman can go so that the women are separated by at least one man. There are $C(9, 5) = \frac{9!}{5!(9-5)!}$ ways to choose the slots for the women. There are $P(5, 5) = 5!$ ways to arrange the women. By the Product Rule, there are

$$8! \cdot \frac{9!}{5!(9-5)!} \cdot 5! = 609,638,400 \quad \text{ways.}$$

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) permutation
 - (b) r -permutation
 - (c) combination
 - (d) r -combination

2. A local pizza shop offers their pies in small, medium, large, or extra large. For toppings, they offer: pepperoni, sausage, bacon, olives, onions, peppers, and anchovies. How many different pizzas can they make that have exactly three (different) toppings?
 3. Compute the following.
 - (a) $P(7, 3)$
 - (b) $C(7, 3)$
 - (c) $C(8, 0)$
 - (d) $P(8, 5)$
 - (e) $C(8, 3)$
 4. Fifteen people on a softball team show up for a game.
 - (a) How many ways are there to select 9 to take the field?
 - (b) How many ways are there to assign the 9 positions?
 5. How many permutations of the letters ABCDEFGH contain the string ABC?
 6. In how many different orders can ten runners finish a race if no ties are allowed?
 7. List all the permutations of $\{1, 2, 3\}$
 8. List all the 3-combinations of $\{a, e, i, o, u\}$.
 9. List all the 3-permutations of $\{1, 2, 3, 4, 5, 6\}$.
 10. In how many ways can a set of four letters be selected from the English alphabet?
 11. How many ways are there for 10 women and 6 men to stand in a line if so that no two men stand next to each other? (Hint: First position the women and then consider the possible positions for the men.)
 12. Harry, Hermione, Ron, Fred, George, Ginny, Luna, Neville, Seamus, and Hagrid go to some pictures taken.
 - (a) How many ways are there to arrange four people from that group in a row for the picture?
 - (b) Suppose Harry is willing to pay for any picture that he is in. How many ways are there to arrange four people from that group in a row for the picture, if Harry must be one of the four?
-

Relations

Phineas: ...*you believe in us!*

Ferb: *And we believe in you.*

Phineas: *And therefore, through the transitive property of belief, you do believe in yourself!*

Phineas and Ferb Season 2 (2010)

Relationships between sets occur in many different contexts. For example, we deal with relationships between people all the time. e.g., Alice is dating Bob. Bob is friends with Eve. Eve and Alice are sisters. We also deal with relationships between other sets. e.g., Four quarters are worth \$1. Ten dimes are worth \$1.

Relationships between elements of sets are described mathematically using a structure called a *relation*, which is just a subset of the Cartesian product of the sets. For example, in the friendship example above, friendship is the relation F and (Bob, Eve) is an element in F .

6.1. Relations and their properties

Goals. To introduce the concept of a relation and basic properties of relations, including the reflexive, symmetric, antisymmetric, and transitive properties.

6.1.1. Binary relations.

A TABLE OF
SLIGHTLY WRONG
EQUATIONS AND IDENTITIES
USEFUL FOR
APPROXIMATIONS
AND/OR
TROLLING TEACHERS
(FOUND USING A MIX OF TROLLING-ERROR,
MATHEMATICAL AND ROBERT FLEMING /AES TOOL.)
ALL UNITS ARE SI UNLS UNLESS OTHERWISE NOTED.

RELATION:	APPROXIMATE TO: UNITS:	ONE PART IN:
ONE LIGHT-YEAR ^(ly)	99^8	IN 140
EARTH SURFACE AREA ^(m²)	69^8	IN 150
OCEANS VOLUME ^(m³)	91^8	IN 70
SECONDS IN A YEAR	75^{14}	IN 100
SECONDS IN A YEAR (ADVT. MICHIGAN)	$525,000-60$	IN 1100
AGE OF THE UNIVERSE (years)	15^{10}	IN 70
PLANCK'S CONSTANT	$\frac{1}{30^{10}}$	IN 110
FINE STRUCTURE CONSTANT	$\frac{1}{140}$	IN 1000 <small>([500] [1000])</small>
FUNDAMENTAL CHARGE	$\frac{3}{14n^{10}}$	IN 500
WHITE HOUSE SWITCHBOARD	$e^{\sqrt{1+78}}$	
JENNIS CONSTANT	$(7^{1/2}-9)n^2$	
<p>INTERESTING: WORLD POPULATION ESTIMATE WHICH SHOULD STAY CURRENT FOR A DECADE OR SO:</p> <p>THE FIRST TWO DIGITS OF DEC. 2011 (4) VERSION FOR US POPULATION: 2011 (4) 3 (22) (1000)</p> <p>SUBTRACT THE 4th DIGIT OF LAST THREE DIGITS: $11-1=10$ SUBTRACT 10: $10-10=0$</p> <p>ADD A DECIMAL POINT: $10 \rightarrow 1.0$ MULTIPLY BY 3: $1.0 \times 3 = 3.0$</p> <p>ADD 6: $6 + 3.0 = 9.0$ WORLD POPULATION IN BILLIONS: 7.2</p>		
ELECTRON REST ENERGY	$\frac{e}{7^{16}} \text{ J}$	ONE PART IN 1000
LIGHT-YEAR (MILES)	2^{1222}	ONE PART IN 1000
$\sin(60^\circ) = \frac{\sqrt{3}}{2} = \frac{e}{\pi}$		ONE PART IN 1000
$\sqrt{3} = \frac{2e}{\pi}$		ONE PART IN 1000
γ (GAUSS'S SIGMA CONSTANT)	$\frac{1}{73}$	ONE PART IN 4,000
FEET IN A METER	$\frac{5}{9\pi}$	ONE PART IN 4,000
$\sqrt{5} = \frac{2}{3}$		ONE PART IN 7,000
AVOGADRO'S NUMBER	69^{11}	ONE PART IN 25,000
GRAVITATIONAL CONSTANT G	$\frac{1}{e^{(10^{-1})^{10^{-1}})}$	ONE PART IN 25,000
R (GAS CONSTANT)	$(e+1)\sqrt{5}$	ONE PART IN 30,000
PROTON-ELECTRON MASS RATIO	$6\pi^2$	ONE PART IN 50,000
LITERS IN A GALLON	$3 + \frac{\pi}{4}$	ONE PART IN 500,000
g	$6 \ln(4e)$	ONE PART IN 750,000
PROTON-ELECTRON MASS RATIO	$\frac{e^6-10}{\phi}$	ONE PART IN 5,000,000
RED LASER WAVELENGTH	$\frac{1}{1200}$	IN 1000
MEAN EARTH RADIUS	$5^8 6e$	IN 1000
<p>PROTIP - NOT ALL OF THESE ARE WRONG:</p> <p>$\sqrt{2} = \frac{3}{2} + \frac{\pi}{4}$ $\cos \frac{\pi}{4} + \cos \frac{3\pi}{4} + \cos \frac{5\pi}{4} = \frac{1}{2}$</p> <p>$7 = \frac{e}{3} + \frac{e}{5}$ $\sqrt{5} = \frac{13}{21} + \frac{4\pi}{11}$ $\sum \frac{1}{n^2} = \ln(3)^2$</p>		

Figure 6.1.1. xkcd: Approximations. (<https://xkcd.com/1047/>) Two tips: 1) 8675309 is not just prime, it's a twin prime, and 2) if you ever find yourself raising $\log(\text{anything})^e$ or taking the π -th root of anything, set down the marker and back away from the whiteboard; something has gone horribly wrong.

Definition 6.1.1. Let A and B be sets. A *binary relation* or *relation* R from A to B is a subset of $A \times B$, $R \subseteq A \times B$. We use aRb to denote the fact that (a, b) is in R , and say a is *related* to b .

One can think of a relation R as defining a relationship between elements from A to B .

Example 6.1.2. Let P denote the set of people on earth, and let F be the set of foods. Define a relation L from P to F by $(p, f) \in L$ if and only if p likes f . For example, (Dan, potato chips) is an element of L . We can also write that as Dan L potato chips.

Example 6.1.3. Let C be the set of names of cities in the US. Let S be the set of names of states in the US. Let R be the relation from C to S

$$R = \{(a, b) \in C \times S \mid a \text{ is a city in } b\}.$$

Then (Greensboro, North Carolina) and (Dallas, Texas) are both in R .

Definition 6.1.4. A *relation* on a set A is a relation from A to A .

Example 6.1.5. Equality defines a relation on \mathbb{R} , $E \subseteq \mathbb{R} \times \mathbb{R}$.

$$E = \{(a, b) \in \mathbb{R} \times \mathbb{R} \mid a = b\}.$$

Since $2 = 2$, we have $2E2$. Similarly, $\pi E\pi$, $\sqrt{2}E\sqrt{2}$, etc.

Example 6.1.6. Let P denote the set of all people (alive or dead). Let G be the relation on P by xGy if and only if y is the grandfather of x . People have two grandfathers (a paternal grandfather and a maternal grandfather), so G does not define a function from P to P . Functions are not allowed to have this ambiguity.

Example 6.1.7. A function $f: A \rightarrow B$ defines a subset of $A \times B$ called the *graph* of f ,

$$\Gamma = \{(a, b) \in A \times B \mid b = f(a)\}.$$

We see that Γ is a relation from A to B .

Example 6.1.8. Relations are more general than graphs of functions. Let C be the set of names of cities in the US. Let S be the set of names of states in the US. Let R be the relation from C to S

$$R = \{(a, b) \in C \times S \mid a \text{ is a city in } b\}.$$

Then (Greenville, North Carolina) and (Greenville, South Carolina) are in R . This cannot happen for the graph of a function, since functions have the property that there is a unique element in the range related to each element in the domain.

Example 6.1.9. The following define relations on \mathbb{Z} .

- (1) $R_1 = \{(a, b) \in \mathbb{Z}^2 \mid a \leq b\}$.
- (2) $R_2 = \{(a, b) \in \mathbb{Z}^2 \mid a = b - 3\}$.

We have $10R_157$ and $10R_213$.

6.1.2. Properties of binary relations.

Definition 6.1.10. A relation R on a set A is *reflexive* if, for every a in A , we must have aRa

Proof Technique 6.1.11 (Show R is reflexive). Suppose R is a relation on a set A , and we want to prove R is reflexive.

- (1) Fix a generic element a in A .
- (2) Deduce aRa .
- (3) Conclude R is reflexive.

Definition 6.1.12. A relation R on a set A is *symmetric* if, for every a and b in A , whenever bRa , we must have aRb .

Proof Technique 6.1.13 (Show R is symmetric). Suppose R is a relation on a set A , and we want to prove R is symmetric.

- (1) Fix generic elements a and b in A such that aRb .
- (2) Deduce bRa .
- (3) Conclude R is symmetric.

Definition 6.1.14. A relation R on a set A is *antisymmetric* if, for any a and b in A , whenever aRb and bRa , we must have $a = b$.

Definition 6.1.15. A relation R on a set A is *transitive* if, for every a , b , and c in A , whenever aRb and bRc , we must have aRc .

Proof Technique 6.1.16 (Show R is transitive). Suppose R is a relation on a set A , and we want to prove R is transitive.

- (1) Fix generic elements a , b , and c in A such that aRb and bRc .
- (2) Deduce aRc .
- (3) Conclude R is transitive.

Example 6.1.17. Let R be the relation on \mathbb{Z} defined by \leq , so that aRb if and only if $a \leq b$.

- (1) Since $a \leq a$ for every integer a , we have aRa for every integer a . Thus, R is reflexive.
- (2) Since $1 \leq 2$, but $2 \not\leq 1$, we have $1R2$ but not $2R1$. Thus, R is not symmetric.
- (3) Whenever $a \leq b$ and $b \leq a$, we must have $a = b$. That means whenever aRb and bRa , we must have $a = b$. Thus R is antisymmetric.
- (4) Whenever $a \leq b$ and $b \leq c$, we must have $a \leq c$. Then, whenever aRb and bRc , we must have aRc . Thus R is transitive.

Example 6.1.18. Let R be the relation on $\{1, 2, 3, 4\}$ that is given by

$$R = \{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}.$$

Then R is not reflexive since $(1, 1)$ is not in R . It is not symmetric since $(2, 4)$ is in R but $(4, 2)$ is not in R . It is not antisymmetric since $(2, 3)$ is in R and $(3, 2)$ is in R , but $2 \neq 3$. It is transitive since if (a, b) is in R and (b, c) is in R , then (a, c) is in R .

Example 6.1.19. Let R be the relation on $\{1, 2, 3, 4\}$ that is given by

$$R = \{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\}.$$

Then R is not reflexive since $(1, 1)$ is not in R . It is not symmetric since $(1, 4)$ is in R , but $(4, 1)$ is not in R . It is not antisymmetric since $(1, 3)$ is in R and $(3, 1)$ is in R , but $1 \neq 3$. It is not transitive since $(1, 3)$ is in R and $(3, 1)$ is in R , but $(1, 1)$ is not in R .

Example 6.1.20. Let R be the relation on $\{1, 2, 3, 4\}$ that is given by

$$R = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}.$$

Then R is reflexive since $(1, 1)$, $(2, 2)$, $(3, 3)$, and $(4, 4)$ are all in R . It is symmetric because whenever (a, b) is in R , we have (b, a) is in R . It is transitive since if (a, b) is in R and (b, c) is in R , then (a, c) is in R . It is not antisymmetric since $(1, 2)$ is in R and $(2, 1)$ is in R , but $1 \neq 2$.

6.1.3. Combining binary relations. Since relations are sets, we can take unions, intersections, differences, and complements of relations.

Example 6.1.21. Let S be the set of students at UNCG, and let C be the set of courses at UNCG. Let T and N be the relations from S to C ,

$$T = \{(s, c) \in S \times C \mid s \text{ has taken } c\};$$

$$N = \{(s, c) \in S \times C \mid s \text{ needs } c \text{ to graduate}\}.$$

Then

$$T \cap N = \{(s, c) \in S \times C \mid s \text{ has taken } c \text{ and needs } c \text{ to graduate}\};$$

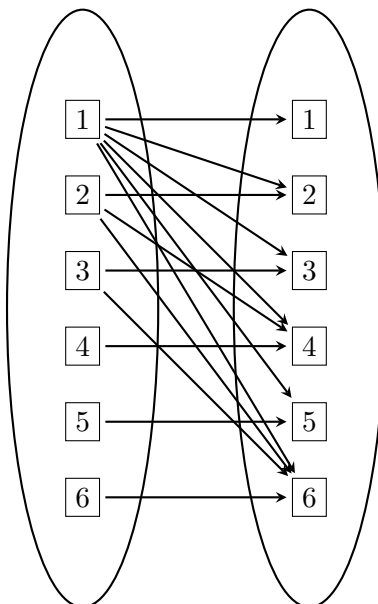
$$T - N = \{(s, c) \in S \times C \mid s \text{ has taken } c \text{ but does not need } c \text{ to graduate}\}.$$

Like functions, we can represent relations in many ways.

Example 6.1.22. Let R be the relation on $\{1, 2, 3, 4, 5, 6\}$ given by aRb if and only if a divides b . Then

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 2), (2, 4), (2, 6), (3, 3), (3, 6), (4, 4), (5, 5), (6, 6)\}.$$

We can represent it graphically as



We can represent R in a table by putting an \times in the a th row and b th column if and only if a divides b .

R	1	2	3	4	5	6
1	\times	\times	\times	\times	\times	\times
2		\times		\times		\times
3			\times			\times
4				\times		
5					\times	
6						\times

Like functions, we can also compose relations.

Definition 6.1.23. Let A , B , and C be sets. Suppose R is a relation from A to B , and S is a relation from B to C . The **composite** or **composition** of R and S , denoted $S \circ R$, is the relation from A to C

$$S \circ R = \{(a, c) \in A \times C \mid \exists b \in B \text{ such that } aRb \text{ and } bSc \}.$$

Example 6.1.24. Let $A = \{1, 2, 3\}$, $B = \{1, 2, 3, 4\}$, and $C = \{0, 1, 2\}$. Let R be the relation from A to B

$$R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\},$$

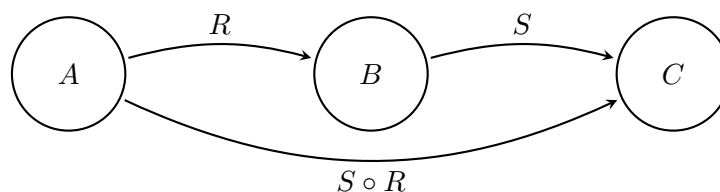


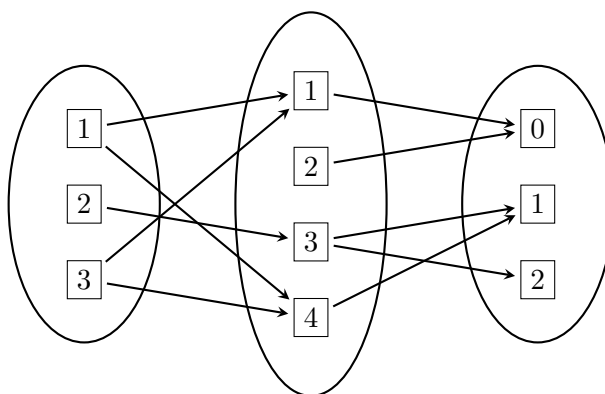
Figure 6.1.2. Composition of two relations.

and let S be the relation from B to C

$$S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}.$$

Compute the composition $S \circ R$.

One approach is to view this graphically.



Then

$$S \circ R = \{(1, 0), (2, 1), (2, 2), (1, 1), (3, 0), (3, 1)\}.$$

Exercises

- Give the definition for these terms. Be sure to set up any notation that is required.
 - binary relation from one set to another
 - relation on a set
 - reflexive relation
 - symmetric relation
 - antisymmetric relation
 - transitive relation
 - composition of relations
- Consider the relation $R = \{(a, b) \mid a \text{ divides } b\}$ on the set $\{1, 2, 3, 4\}$.
 - List all of the ordered pairs in R .

- (b) Is R reflexive? Explain.
 (c) Is R symmetric? Explain.
 (d) Is R antisymmetric? Explain.
 (e) Is R transitive? Explain.
3. Which of these relations on the set of all people are reflexive? Which are symmetric? Which are antisymmetric? Which are transitive? Justify.
- (a) $\{(a, b) \mid a \text{ and } b \text{ are the same height}\}$
 (b) $\{(a, b) \mid a \text{ and } b \text{ live in the same state}\}$
 (c) $\{(a, b) \mid a \text{ and } b \text{ have met}\}$
 (d) $\{(a, b) \mid a \text{ and } b \text{ are blood relatives}\}$ ¹
 (e) $\{(a, b) \mid a \text{ and } b \text{ speak a common language}\}$
4. Which of these relations on $\{1, 2, 3, 4\}$ are reflexive? Which are symmetric? Which are antisymmetric? Which are transitive? Justify.
- (a) $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
 (b) $\{(1, 2), (2, 1), (1, 3), (3, 1), (1, 4), (4, 1)\}$
 (c) $\{(1, 1), (2, 2), (1, 2), (2, 1), (3, 3), (4, 4)\}$
 (d) $\{(1, 1), (2, 2), (1, 2), (2, 1), (1, 3), (3, 1), (3, 3), (4, 4)\}$
5. Consider the following relations R on the set of real numbers. For each of these relations, determine whether or not it is reflexive, symmetric, antisymmetric, and/or transitive. Justify.
- (a) $(x, y) \in R$ if and only if $x + y = 0$
 (b) $(x, y) \in R$ if and only if $x - y \in \mathbb{Z}$
 (c) $(x, y) \in R$ if and only if $x = 253$
 (d) $(x, y) \in R$ if and only if $xy = 0$
 (e) $(x, y) \in R$ if and only if $x = 1$ or $y = 1$
6. Let $A = \{0, 1, 2, 3, 4\}$, and let $B = \{0, 1, 2, 3\}$. Consider the following relations from A to B . For each of these relations, list the ordered pairs in the relation.
- (a) $(a, b) \in R$ if and only if $a = b$
 (b) $(a, b) \in R$ if and only if $a > b$
 (c) $(a, b) \in R$ if and only if $\gcd(a, b) = 1$
 (d) $(a, b) \in R$ if and only if $\text{lcm}(a, b) = 2$
 (e) $(a, b) \in R$ if and only if $a \mid b$
 (f) $(a, b) \in R$ if and only if $a + b = 4$

7. Let R and S be the relations

$$R = \{(1, 2), (2, 3), (3, 4)\}$$

$$S = \{(1, 1), (1, 2), (2, 1), (2, 3), (3, 1), (3, 4)\}$$

Compute the following relations.

- (a) $R \cup S$

¹A person who is related to another through a common ancestor, and not by marriage or adoption.

- (b) $R \cap S$
- (c) $R - S$
- (d) $S - R$

8. Let R and S be relations on $\{1, 2, 3, 4\}$ defined by

$$R = \{(2, 2), (2, 3), (3, 4), (4, 4)\}$$

$$S = \{(1, 2), (2, 1), (2, 4), (3, 1), (3, 4)\}.$$

Compute the following relations.

- (a) $R \circ R$
 - (b) $R \circ S$
 - (c) $R \circ R$
 - (d) $S \circ S$
9. Let P be the relation on the set of people consisting of pairs (a, b) , where a is a parent of b . Let S be the relation consisting of pairs (a, b) , where a and b are siblings (brothers or sisters). Describe the composition relations $P \circ S$ and $S \circ P$.
10. Let B be the relation on the set of states in the US consisting of pairs (a, b) where a shares a land border with b .
- (a) Give three examples of elements in B .
 - (b) Give three examples of elements not in B .
 - (c) Is B symmetric?
 - (d) Is B transitive?
11. Give an example of a relation on a set that is reflexive, but not symmetric.
12. Give an example of a relation on a set that is symmetric, but not reflexive.
13. Give an example of a relation on a set that is reflexive and symmetric, but not transitive.

6.2. Equivalence relations

Goals. To study equivalence relations and their equivalence classes.

6.2.1. A special class of binary relation.

Definition 6.2.1. A relation R on a set A is called an *equivalence relation* if R is reflexive, symmetric, and transitive.

Equivalence relations are important throughout mathematics and computer science. It makes the notion of equivalent objects precise.

Definition 6.2.2. Two elements a and b are *equivalent* if they are related by an equivalence relation.



Figure 6.2.1. xkcd: Soda Sugar Comparisons. (<https://xkcd.com/1793/>) The key is portion control, which is why I've switched to eating smaller cans of frosting instead of full bottles.

This is a notion that makes precise when two things are “the same” up to differences that we are willing to ignore. We have seen this before.

For example, we regularly think of a \$1 note as “the same” as four quarters, though they are different. According to the [United States Mint](#), a quarter is 5.670 grams. That means \$1000 in quarters weighs about 50 pounds. On the other hand, a \$1 note weighs about 1 gram, so \$1000 in \$1 notes weighs about 2.2 pounds. Would you rather run a race carrying \$1000 in \$1 notes or quarters?

For a more mathematical example, we think of $\frac{2}{6}$ as equal to $\frac{1}{3}$. In what sense are they the same? In this case, two objects are equal as rational numbers, but differ in representation.

Remark 6.2.3. We often use \sim instead for equivalence relations, so if a and b are equivalent, we might write $a \sim b$.

How do we prove a relation R on a set A is an equivalence relation? We need to verify that R is reflexive, symmetric, and transitive. Recall that these are characterized by universal conditions. To show something is true “for all x in A ”, we fix a generic element a in A , and show that the conditions are satisfied for the generic element a . Then since the condition is satisfied for the generic element, it is satisfied by all the elements. How do we do this in practice?

Proof Technique 6.2.4 (Show R is an equivalence relation). Suppose R is a relation on a set A , and we want to prove R is an equivalence relation. The proof has three parts.

- (1) Use Proof Technique 6.1.11 to show R is reflexive.
- (2) Use Proof Technique 6.1.13 to show R is symmetric.
- (3) Use Proof Technique 6.1.16 to show R is transitive.
- (4) Conclude R is an equivalence relation.

Example 6.2.5. Let R be the relation on \mathbb{Z} defined by aRb if and only if $|a| = |b|$. Let’s show that R is an equivalence relation.

Proof. We check the three defining characteristics.

Reflexive: Let $a \in \mathbb{Z}$. Then $|a| = |a|$, so aRa . Thus R is reflexive.

Symmetric: Let $a, b \in \mathbb{Z}$. Suppose aRb so that $|a| = |b|$. Then $|b| = |a|$, so bRa . Thus R is symmetric.

Transitive: Let $a, b, c \in \mathbb{Z}$. Suppose aRb and bRc so that $|a| = |b|$ and $|b| = |c|$. Then $|a| = |c|$, so aRc . Thus R is transitive.

Therefore R is an equivalence relation. □

Example 6.2.6. Define a relation R on \mathbb{R} by aRb if and only of $a - b \in \mathbb{Z}$. Is R an equivalence relation?

Proof. We check the three defining characteristics.

Reflexive: Let $a \in \mathbb{R}$. Then $a - a = 0 \in \mathbb{Z}$, so aRa . Thus R is reflexive.

Symmetric: Let $a, b \in \mathbb{R}$. Suppose aRb so that $a - b = k \in \mathbb{Z}$. Then $b - a = -k \in \mathbb{Z}$, so bRa . Thus R is symmetric.

Transitive: Let $a, b, c \in \mathbb{R}$. Suppose aRb and bRc so that $a - b = k \in \mathbb{Z}$ and $b - c = \ell \in \mathbb{Z}$. Then $a - c = k - \ell \in \mathbb{Z}$, since the difference of integers is an integers, so aRc . Thus R is transitive.

Therefore R is an equivalence relation. □

Theorem 6.2.7. Let $m > 1$ be an integer. Let R be the relation on \mathbb{Z}

$$R = \{(a, b) \in \mathbb{Z}^2 \mid a \equiv b \pmod{m}\}.$$

Then R is an equivalence relation.

Proof. We check the three defining conditions.

Reflexive: Let $a \in \mathbb{Z}$. Then $m \mid (a - a)$, so $a \equiv a \pmod{m}$. Then aRa . Thus R is reflexive.

Symmetric: Let $a, b \in \mathbb{Z}$. Suppose aRb so that $a \equiv b \pmod{m}$. Then $a - b = km$ for some integer k . Then $b - a = (-k)m$, so $b \equiv a \pmod{m}$. Then bRa . Thus R is symmetric.

Transitive: Let $a, b, c \in \mathbb{Z}$. Suppose aRb and bRc so that $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$. Then $a = b + km$ for some integer k , and $b = c + \ell m$ for some integer ℓ . Then

$$a = b + km = (c + \ell m) + km = c + (\ell + k)m.$$

Since k and ℓ are integers, $k + \ell$ is an integer. Then $a \equiv c \pmod{m}$, and so aRc . Thus R is transitive.

Therefore R is an equivalence relation. \square

6.2.2. Equivalence classes and partitions.

Definition 6.2.8. Let R be an equivalence relation on A , and let a be an element of A . The *equivalence class* of a , denoted $[a]_R$ or $[a]$ is the set of elements that are related to a . In other words,

$$[a] = \{b \in A \mid aRb\}.$$

Remark 6.2.9. Since equivalence relations are symmetric, this is the same as

$$[a] = \{b \in A \mid bRa\}.$$

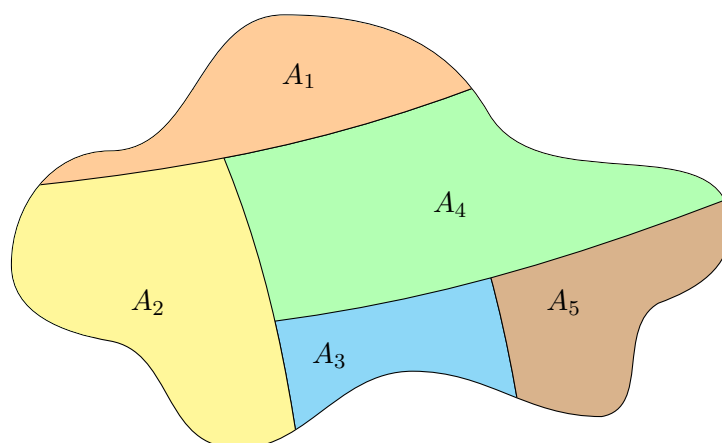
Example 6.2.10. Let R denote the congruence modulo 7 equivalence relation. Then

$$\begin{aligned} [0] &= \{\dots, -14, -7, 0, 7, 14, \dots\} \\ [14] &= \{\dots, -14, -7, 0, 7, 14, \dots\} \\ [3] &= \{\dots, -11, -4, 3, 10, 17, \dots\} \end{aligned}$$

Note that $[a] = [a']$ precisely when aRa' .

Definition 6.2.11. A *partition* of a set A is a collection of non-empty subsets $A_i \subseteq A$, $i \in I$, such that

- (1) $A_i \cap A_j = \emptyset$ for all $i \neq j$;
- (2) $\bigcup_{i \in I} A_i$.



Theorem 6.2.12. Let \sim be an equivalence relation on A . Then for all $a, b \in A$, the following are equivalent.

- (1) $a \sim b$
- (2) $[a] = [b]$
- (3) $[a] \cap [b] \neq \emptyset$

Proof. We prove $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$.

$1 \rightarrow 2$: Suppose $a \sim b$. Let $c \in [a]$. Then $c \sim a$. By transitivity, we have $c \sim b$. Thus $c \in [b]$. Therefore $[a] \subseteq [b]$. Now let $c \in [b]$. Then $b \sim c$. By transitivity, we have $a \sim c$. Thus $c \in [a]$. Therefore $[b] \subseteq [a]$. Then $[a] = [b]$, as desired.

$2 \rightarrow 3$: Suppose $[a] = [b]$. Since \sim is reflexive, $a \in [a]$, so $[a] \neq \emptyset$. Then $[a] \cap [b] = [a] \neq \emptyset$.

$3 \rightarrow 1$: Suppose $[a] \cap [b] \neq \emptyset$. Let $c \in [a] \cap [b]$. Then $a \sim c$ and $c \sim b$. By transitivity, we have $a \sim b$.

□

Theorem 6.2.13. *Let \sim be an equivalence relation on a set A . Then the equivalence classes form a partition of A . Conversely, given a partition $\{A_i \mid i \in I\}$ of a set A , there is an equivalence relation \sim that has the sets A_i as equivalence classes.*

Example 6.2.14. Consider the equivalence relation R on the set of people in the world defined by aRb if and only if a has the same birthday (month and date). Then [Thom Yorke] is the set of all people with a birthday on October 7. The equivalence relation partitions the set of all people into equivalence classes, where each equivalence class contains all the people with a given birthday. Specifically, there are 366 equivalence classes (don't forget February 29!).

Given a partition of a set A , we compute the corresponding equivalence relation R on A using the subsets in the partition as equivalence classes.

Example 6.2.15. Suppose $A = \{1, 2, 3, 4, 5, 6\}$. Given the partition $A_1 = \{1, 2\}$, $A_2 = \{3, 5, 6\}$, and $A_3 = \{4\}$, the ordered pairs in the equivalence relation R produced by this partition is

$$R = \{(1, 2), (2, 1), (3, 5), (5, 3), (3, 6), (6, 3), (5, 6), (6, 5), (4, 4)\}.$$

Example 6.2.16. How many different equivalence relations are there on $A = \{1, 2, 3\}$? By Theorem 6.2.13, it is enough to enumerate partitions of A .

- (1) $A = \{1, 2, 3\}$
- (2) $A = \{1, 2\} \cup \{3\}$
- (3) $A = \{1\} \cup \{2, 3\}$
- (4) $A = \{1, 3\} \cup \{2\}$
- (5) $A = \{1\} \cup \{2\} \cup \{3\}$

Thus there are five equivalence relations on A .

Exercises

1. Give the definition for these terms. Be sure to set up any notation that is required.
 - (a) equivalence relation
 - (b) equivalent elements
 - (c) equivalence class
 - (d) partition of a set
2. Define an equivalence relation on the set of restaurants in Greensboro.
3. Define an equivalence relation on the set of classes offered at UNCG.

4. Let \sim be an equivalence relation on a set A , and let a and b be elements of A . Prove that if a is equivalent to b , then the equivalence class of a is equal to the equivalence class of b . In other words, show if $a \sim b$, then $[a] = [b]$.
5. Consider the relation $R = \{(a, b) \mid a \text{ divides } b\}$ on the set $\{1, 2, 3, 4\}$. Is R an equivalence relation? Justify.
6. Let R be the relation on the set of all people, defined by
- $$R = \{(a, b) \mid a \text{ and } b \text{ were born in the same month}\}.$$
- Prove that R is an equivalence relation.
7. Suppose \sim is an equivalence relation on a set A , and a and b are elements of A . Prove that if $[a] \cap [b] \neq \emptyset$, then $a \sim b$.
8. Let R be the relation on \mathbb{Z}^2 such that $(a, b)R(c, d)$ if and only if $a+d = b+c$. Is R an equivalence relation? Justify.
9. Let R be the relation on \mathbb{Z}^2 such that $(a, b)R(c, d)$ if and only if $ad = bc$. Is R an equivalence relation? Justify.
10. Which of these relations on the set of all people are equivalence relations? Justify.
- $\{(a, b) \mid a \text{ and } b \text{ are the same height}\}$
 - $\{(a, b) \mid a \text{ and } b \text{ live in the same state}\}$
 - $\{(a, b) \mid a \text{ and } b \text{ have met}\}$
 - $\{(a, b) \mid a \text{ and } b \text{ are blood relatives}\}$ ²
 - $\{(a, b) \mid a \text{ and } b \text{ speak a common language}\}$
11. Which of these relations on $\{1, 2, 3, 4\}$ are equivalence relations? Justify.
- $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
 - $\{(1, 2), (2, 1), (1, 3), (3, 1), (1, 4), (4, 1)\}$
 - $\{(1, 1), (2, 2), (1, 2), (2, 1), (3, 3), (4, 4)\}$
 - $\{(1, 1), (2, 2), (1, 2), (2, 1), (1, 3), (3, 1), (3, 3), (4, 4)\}$
12. Let A and B be sets, and let $f: A \rightarrow B$ be a function. Let R be the relation on A defined by
- $$R = \{(a_1, a_2) \in A^2 \mid f(a_1) = f(a_2)\}.$$
- Prove that R is an equivalence relation.
 - Describe the equivalence classes of R .

²A person who is related to another through a common ancestor, and not by marriage or adoption.

Programming assignments

The good news about computers is that they do what you tell them to do. The bad news is that they do what you tell them to do.

Ted Nelson (1937–)

Computers can be used to great benefit in mathematics, both in education and research. Many results in this course lend themselves to computational exploration. The following Python programming exercises have you implementing algorithms from class and using these algorithms to explore mathematics.

The language chosen for these assignments is Python 3. There are several additional resources you may find helpful. Choose based on your programming background and desired difficulty level.

- Python 3 Tutorial: This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python’s most noteworthy features, and will give you a good idea of the language’s flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in The Python Standard Library. This will be the main source of information for the programming assignments in the Appendix.

<https://docs.python.org/3/tutorial/>

- Non-Programmer's Tutorial for Python 3: The Non-Programmers' Tutorial For Python 3 is a tutorial designed to be an introduction to the Python programming language. This guide is for someone with no programming experience.
https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3
- Python for Non-Programmers: If you've never programmed before, the tutorials on this page are recommended for you; they don't assume that you have previous experience.
<https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>
- Python for Programmers: The tutorials on this page are aimed at people who have previous experience with other programming languages (C, Perl, Lisp, Visual Basic, etc.).
<https://wiki.python.org/moin/BeginnersGuide/Programmers>
- The Python Wiki: This Wiki is a community place to gather and organize all things about Python. Feel free to exercise your editorial skills and expertise to make it a useful knowledge base and up-to-date reference on all Python-related topics.
<https://wiki.python.org/moin/FrontPage>
- Learn Python the Hard Way: This book instructs you in Python by slowly building and establishing skills through techniques like practice and memorization, then applying them to increasingly difficult problems. By the end of the book you will have the tools needed to begin learning more complex programming topics.
<https://learnpythonthehardway.org/book/>

- P0: Hello world
- P1: For and if
- P2: Sets
- P3: Functions
- P4: More functions and lists
- P5: Fast exponentiation
- P6: Extended Euclidean Algorithm and primality testing
- P7: Birthday problem
- P8: Dictionaries and analysis of languages
- Optional: Fun with turtles (optional)
- RSA Exercise

P0: HELLO WORLD

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §1. Whetting Your Appetite
 - §3. An Informal Introduction to Python
- (NP)
 - Hello, World
- Write a script that does the following.
 - (1) Print “Hello, world.”
 - (2) Print your full name.
 - (3) Define `year` to be your year (Freshman, Sophomore, Junior, Senior, Other) and `major` to be your major, both as strings. Then print these values.
 - (4) Print at least three more things I might not know about you. This can include things such as an interesting fact about yourself, a description of your programming background, what you hope to get out of this course, your plans for after graduation,

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P1: FOR AND IF

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §3.1.1. Numbers
 - §3.2 First Steps Towards Programming
 - §4.1 `if` Statements
 - §4.2. `for` Statements
 - §4.3. The `range()` function
- (NP) Read and work through the examples in
 - Count to 10
 - Decisions
 - For Loops
- Write a script that solves the following problems. For each problem, print the problem number before the answer to make it easier to grade.
 - (1) Compute the first 20 positive perfect squares using a `for` or `while` loop. (Note: This is the same as the square of the first 20 positive integers.)
 - (2) Compute the last digit of each of the first 20 positive perfect squares. You can get the last digit of `a` using `a % 10`. As before, use a `for` or `while` loop.
 - (3) Make a conjecture about the digits that can never arise as the last digit of a positive perfect square. Use complete sentences.
 - (4) Verify your conjecture for the first 10,000 positive perfect squares. To avoid human error, use a `for` loop with an `if` statement that tells you when the conjecture fails. Print the statement `Beginning verification...` before the `for` loop and the statement `Verification complete.` at the end of the `for` loop. Print `Conjecture is false!` if you find a counterexample to your conjecture in the `for` loop. Hint: You can define a set `conjectured_set`, and check if the last digit is in there.

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P2: SETS

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §5.1.3 List Comprehensions
 - §5.4. Sets
- (NP) Read and work through the examples in
 - For Loops
 - Boolean Expressions
- Write a script that solves the following problems. For each problem, first print the problem number to make it easier to grade.
 - (1) Let $A = \{0, 1, 2, 3, 5, 10, 12\}$, and let $B = \{2, 4, 6, 8, 10, 12, 14, 16\}$. Use Python to compute $A \cup B$, $A \cap B$, $A - B$, and $B - A$. Put in enough print statements so that the output is clear.
 - (2) Let E be the first 50 positive integer multiples of 2.

$$E = \{2n \mid n \in \{1, 2, \dots, 50\}\}.$$

Construct E in Python using a `for` loop with `update` or as a list comprehension. Print E .

- (3) Let F be the set of set of positive integers n up to 100 that are 0 modulo 2

$$F = \{n \mid n \in \{1, 2, \dots, 100\} \text{ and } n \bmod 2 \equiv 0\}.$$

Construct F in Python using a `for` loop with `update` or as a list comprehension. Note that $n \bmod 2$ can be computed in Python as `n % 2`.

- (4) It should be clear from the definitions that E and F are the same set. Suppose it wasn't clear. Use Python to verify $E = F$ in two ways.
 - Use `for` loops to verify that each element of E is in F **and** each element of F is in E .
 - Use the built-in `==` to check the equality of Python sets.

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P3: FUNCTIONS

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §4.6. Defining Functions
 - §4.7.6. Documentation Strings
- (NP) Read and work through the examples in
 - Defining Functions
- Write the functions described below to your script. Be sure to include some comments and a short docstring for each function.
 - Write a function `triangular(n)` which takes as input an integer n and returns the n^{th} triangular number $T_n = n(n + 1)/2$. Remember to use `//` instead of `/` to keep the result an integer.
 - Write a function `sumupto(n)` which takes as input a positive integer n and returns the sum

$$S_n = 1 + 2 + 3 + \cdots + n.$$

The code should *not* create a list `[1, 2, ..., n]` and then add the terms. Within the function, define a variable `total` that is initialized to 0. Then run a `for` loop to keep adding to `total`. e.g., `total += i` or `total = total + i`.

- Write a function `sumpower(n,k)` which takes as input integers n and k , and returns the sum of the k^{th} powers of the first n positive integers. For example, `sumpower(100,7)` should return 1300583304167500 since

$$1^7 + 2^7 + \cdots + 100^7 = 1300583304167500.$$

- Write a script that solves the following problems. For each problem, first print the problem number followed by the answer to make it easier to grade. Be sure your script includes all the necessary code to produce the results.
 - (1) Use a `for` loop to print n , S_n , and T_n on a line for each $n = 1, 2, \dots, 100$.
 - (2) Make a conjecture about the relationship between S_n and T_n .
 - (3) Compute the sum $T_n + T_{n-1}$ for $n = 1, 2, \dots, 20$, and make a conjecture about the relationship between $T_n + T_{n-1}$ and n .
 - (4) Compute the sum

$$1^5 + 2^5 + 3^5 + \cdots + 2016^5.$$

- (5) Describe what `sumpower(100,-2)` computes.

- (6) Compute `sumpower(500, -2)`, `sumpower(5000, -2)`, and `sumpower(50000, -2)`. Use this to explain why it is reasonable to guess that

$$\frac{1}{1} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots = \frac{\pi^2}{6}.$$

You can use `import math` to teach Python some math. You can get approximation of π with `math.pi`.

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P4: MORE FUNCTIONS AND LISTS

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §4.7. More on Defining Functions
 - §3.1.4. Lists
 - §5.1 More on Lists (§5.1.1–4)
- (NP) Read and work through the examples in
 - Lists
 - More on Lists
- Write the functions described below to your script. Be sure to include some comments and a short docstring for each function.
 - Let f be a function defined on positive integers by

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even,} \\ 3n + 1 & \text{if } n \text{ is odd.} \end{cases}$$

Write a function `hailstone(n)` which takes as input a positive integer n and returns a list called the *hailstone sequence* $[a_0, a_1, \dots]$, where $a_0 = n$ and $a_k = f(a_{k-1})$ for $k > 0$. The sequence terminates whenever it reaches 1. The still unproven *Collatz conjecture* or $3x + 1$ *conjecture* claims that all hailstone sequences have finite length.

- Write a script that solves the following problems. For each problem, first print the problem number followed by the answer to make it easier to grade. Be sure your script includes all the necessary code to produce the results.
 - (1) Compute the hailstone sequence for 1.
 - (2) Compute the hailstone sequence for 27.
 - (3) Which positive integer $n \leq 20,000$ has the longest hailstone sequence? How long is the hailstone sequence of this integer?

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P5: FAST EXPONENTIATION

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §7.1
- Review lectures and textbook: base b expansion and modular exponentiation
- Write the functions described below to your script. Be sure to include docstrings and some comments.
 - Write a function `fast_power_mod(b,n,m)` which takes inputs b, n, m and returns $b^n \bmod m$. Your code should implement the fast exponentiation described in lecture. You can use Python's `pow` function to check your code.
- Write a script that solves the following problems. For each problem, first print the problem number followed by the answer to make it easier to grade. Be sure your script includes all the necessary code to produce the results.
 - (1) Enter `(253**520126) % 123` to compute $253^{520126} \bmod 123$. Compare the time it takes to compute with the time required for `fast_power_mod(253, 520126, 123)`.
 - (2) Use your function `fast_power_mod` to compute
$$1234223432^{56789101112} \bmod 24123456789101112.$$
 - (3) Use your function `fast_power_mod` to compute $a^{10} \bmod 11$ for $a = 1, 2, \dots, 10$. Repeat your experiment, computing $a^{p-1} \bmod p$ for $a = 1, 2, \dots, p-1$ for the primes $p = 13, 17, \text{ and } 19$. You can use formatted printing to make it look nice. Use your findings to conjecture the value of $a^{p-1} \bmod p$ for any prime p .
 - (4) Repeat the experiment above replacing the primes p with composites $n = 15, 256, \text{ and } 765$. Use your function `fast_power_mod` to compute $a^{n-1} \bmod n$ for $a = 1, 2, \dots, n-1$. Does your conjecture from (1) appear to be true for composite moduli as well?

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P6: EXTENDED EUCLIDEAN ALGORITHM AND PRIMALITY TESTING

DIRECTIONS. Name your script 253-yourlastname-#.py . For example, my submission for P3 would be a file 253-yasaki-3.py . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - 3.1.1. Numbers
 - Learn about `divmod` at
<http://docs.python.org/py3k/library/functions.html>
- Review lecture notes on Extended Euclidean algorithm (page 273) and Fermat's Little Theorem (page 281).
- Write the functions described below to your script. Be sure to include docstrings and some comments.
 - Write a function `XGCD(a,b)` which takes as input 2 non-negative integers a, b and returns the $[d, s, t]$, where $d = \gcd(a, b)$ and s and t are integers so that

$$d = sa + tb.$$

Your code should implement the Extended Euclidean algorithm as described in lecture or on page 273 of the textbook or in the lecture notes.

- Recall that Fermat's Little Theorem says that for a prime p and integer a with $\gcd(a, p) = 1$, we have that $a^{p-1} \equiv 1 \pmod{p}$, or equivalently $a^{p-1} \bmod p = 1$. The contrapositive gives us a primality test. Specifically, suppose we want to test if a positive integer n is composite. Fix an integer a such that $\gcd(a, n) = 1$. If $a^{n-1} \bmod n \neq 1$, then n is not prime. Write a function `maybeprime` which takes as input positive integer n and a prime a and outputs `True` if $a = n$ or $a^{n-1} \bmod n = 1$, and `False` otherwise. Be sure to use fast exponentiation.
- Note that for a fixed positive integer n , if `maybeprime(n,a)` returns `True` for several prime values of a , then it is perhaps likely that n is prime. Write a function `probablyprime` that takes as input a positive integer n that returns `true` if `maybeprime(n,a)` returns `True` for $a \in \{2, 3, 5, 7\}$ and `False` otherwise.
- Write a script that solves the following problems. For each problem, first print the problem number followed by the answer to make it easier to grade. Be sure your script includes all the necessary code to produce the results.
 - (1) Use your `XGCD` code on $a = 1873452387876123$ and $b = 2664867585108574408$ to find the greatest common divisor d and integers s, t so that $as + bt = d$.
 - (2) Use the following to find the positive integers n less than or equal to 100,000 that are probably prime.

```
P = [n for n in range(2,100001) if probablyprime(n)]
```

- (3) There are 9,592 actual primes less than or equal to 100,000. How many composite integers faked their way past your `probablyprime` test? Hint: `len(P)` will return the number of elements in `P`.

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P7: BIRTHDAY PROBLEM

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (NP) Read and work through the examples in
 - Advanced Function Example
 - Recursion
- Review lecture notes on the Product Rule for counting.
- Write the functions described below to your script. Be sure to include docstrings and some comments.
 - Write a function `birthday_noclash` which takes as input a positive integer n and returns the probability that no two people in a group of n people share a birthday. For simplicity, assume no one was born on February 29 so that there are 365 different possibilities for birthdays. Hint: It may be helpful to have a `factorial` function. You can write your own or `import math` to teach Python some math, and use `math.factorial`.
- Write a script that solves the following problems. For each problem, first print the problem number followed by the answer to make it easier to grade. Be sure your script includes all the necessary code to produce the results.
 - (1) Compute the probability of no two people from a group of 22 share a birthday. Compare your answer with the lecture notes.
 - (2) Assuming there are 30 people in the class, compute the probability that no two people from the class share a birthday.
 - (3) How large must a group be before the probability that no two people share a birthday is less than 1%?

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

P8: DICTIONARIES AND ANALYSIS OF LANGUAGES

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- (PT) Read and work through the examples in
 - §5.5. Dictionaries
 - §7.2 Reading and Writing Files
- Write the functions described below to your script. Be sure to include some comments.
 - Write a function `count_chars(s)` that takes as input a string `s`, and returns a dictionary that gives the number of occurrences of every character in `s`. It suffices to only count the 26 lowercase letters of the English alphabet. It may help to start with a list of letters to use as keys. Something like the following will produce such a list.

```
alphabet = list('abcdefghijklmnopqrstuvwxyz')
```
- Use your functions to answer the following questions. Put in print statements to make the output more readable.
 - (1) Download the files `english.txt` and `german.txt`. Use `open` and `read` to read the contents in as strings. Use `lower` to convert to all characters to lower case. Something like the following would work for the english file.

```
with open('english.txt') as infile:  
    english = infile.read().lower()
```
 - (2) Use `count_chars` to compare the frequency of each letter in the two texts. You should give the data as percentages, since the texts are not of equal length.
 - (3) Bonus: Name a novel that does not use the letter 'e'.

RUBRIC

- 10–9 pts:** Script runs without errors. All required components are correctly addressed. The difference between 9 and 10 comes from coding style (comments, structure) and writing style (grammar and spelling in responses).
- 8–6 pts:** Script runs without errors but some required component is missing or incorrect. The score in this range depends on the what is missed.
- 5 pts:** Script does not run because of errors.
- 0 pts:** No submission.

POPTIONAL: FUN WITH TURTLES (OPTIONAL)

DIRECTIONS. Name your script `253-yourlastname-#.py` . For example, my submission for P3 would be a file `253-yasaki-3.py` . Each project script should be uploaded to Canvas by clicking the assignment.

The *Python Tutorial* (PT) is available at

<http://docs.python.org/py3k/tutorial/>

the *Non-Programmer's Tutorial for Python 3* (NP) is available at

http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

- Read and work through the examples in
 - `turtle` module

<http://docs.python.org/3.3/library/turtle.html>

Write a script that utilizes the `turtle` module to do something fun. Use some of the functionality of Python and math that we have covered this semester. e.g., functions, recursion, conditional statements, `for` and `while` loops, `if` statements.

In order to receive full credit, your script must satisfy the following criteria:

- Use at least two of the functions in the `turtle` module.
- Use at least one function that you define.
- Utilize at least one of the following: `for` loop, `while` loop, `if` statement, recursion.
- Make me smile when I run your code. I will be grading for creativity, use of colors, and overall awesomeness.

This assignment is optional. Your lowest Homework and Programming Assignment grades will be replaced with the grade on this assignment.

RSA EXERCISE

1. SOME HISTORY

Rivest, Shamir, and Adleman first publicly described this algorithm for public key encryption in 1978¹. They posted one of the first public-key encryption messages using a 129 digit number which later became known as RSA-129.

$$\begin{aligned} \text{RSA-129} &= 114381625757888867669235779976146612010218296721242362562 \\ &561842935706935245733897830597123563958705058989075147599290026879543541 \\ &= 3490529510847650949147849619903898133417764638493387843990820577 \\ &\quad \times 32769132993266709549961988190834461413177642967992942539798288533. \end{aligned}$$

They offered a \$100 prize and remarked that using technology and factoring techniques available at that time, it would take 40 quadrillion years to crack. Advances in factoring techniques and computers cracked the code in April 1994 to find that the secret message was:

The Magic Words are Squeamish Ossifrage

According to WIKIPEDIA,

Ossifrage is an older name for the lammergeier, a scavenging vulture that is famous for dropping animal bones and live tortoises onto rocks to crack them open. It might perhaps be considered among the least squeamish of creatures.

2. THE SET-UP

Suppose Alice wants to send Bob an encrypted message. Bob lets her know his public key.

Definition 2.1. The *RSA public encryption key* consists of a pair of integers (N, e) , where N is the product of two distinct primes.

The set of integers $\{1, \dots, N\}$ is the set of possible messages, but we will see that you do not want the message to be 1 or N . To encrypt a message M , Alice computes

$$C = M^e \pmod{N}.$$

Notice that with fast exponentiation, this is fast.

If a Eve captures C while it is being transmitted, she will have a hard time computing the original message M . See the lecture notes for more information.

How is it any easier for Bob? The trick is that Bob has a bit of extra information. When constructing the key, Bob chooses N to be a product of two distinct primes p and q . Then $\phi(N) = \phi(pq) = (p-1)(q-1)$. This is the Euler-phi function at N , the number of positive integers less than or equal to N that are relatively prime to N . The exponent e is chosen so that $\gcd(e, \phi(N)) = 1$. Then using the Euclidean algorithm, Bob can compute an inverse to e modulo $\phi(N)$, an integer d such that $ed \equiv 1 \pmod{\phi(N)}$. Then there is an integer k

¹Clifford Cocks described an equivalent system in 1973, but it was classified by the UK intelligence agency GCHQ until 1997

so that $ed = 1 + k\phi(N)$. Now Euler's generalization to Fermat's little theorem says that if $\gcd(C, N) = 1$,

$$\begin{aligned} C^d &\equiv (M^e)^d \pmod{N} \\ &\equiv M^{1+k\phi(N)} \pmod{N} \\ &\equiv M \cdot (M^{\phi(N)})^k \pmod{N} \\ &\equiv M \pmod{N}. \end{aligned}$$

In other words, to decrypt the message, Bob does not need to take an e th root of C modulo N . Instead, he can raise C to the d th power and achieve the same result, where d is an inverse of e modulo $\phi(N)$. Thank you, Euler! Again, with fast exponentiation, this is fast. Note: Choose d to be a positive integer since our fast exponentiation algorithm requires the exponent to be positive.

3. ASCII ENCODING

ASCII is a standard way to represent characters as numbers. For example, a space is represented by 32, a comma is 44, and a period is 46. The capital letters are also 2 digit integers, starting with 65 for A and going to 90 for Z. The Python functions `chr` and `ord` convert the ASCII to characters. e.g., `chr(66)` returns the string A. If you want to go the other way, `ord('A')` returns the integer 65. This is known as *encoding*.

In order to encode messages longer than one character, we will view each number as a digit in a base 256 expansion of an integer M . For example, suppose I want to send the message `Help!`. We have

$$\text{ord}(H) = 72, \quad \text{ord}(e) = 101, \quad \text{ord}(l) = 108, \quad \text{ord}(p) = 112, \quad \text{ord}(!) = 33,$$

so the encoded message is

$$M = (72, 101, 108, 112, 33)_{256}.$$

That means

$$M = 72 \cdot 256^4 + 101 \cdot 256^3 + 108 \cdot 256^2 + 112 \cdot 256^1 + 33 \cdot 256^0 = 310939250721.$$

This allows use to convert messages written as strings into integers.

We can *decode* the message by computing the base 256 expansion and decoding each character. For example, suppose the encoded message is $M = 310939249775$. The base 256 expansion of M is

$$M = (72, 101, 108, 108, 111)_{256}.$$

Then

$$\text{chr}(72) = H, \quad \text{chr}(101) = e, \quad \text{chr}(108) = l, \quad \text{chr}(108) = l, \quad \text{chr}(111) = o,$$

so the decoded message is 'Hello'.

Since it is easy to encode and decode string messages, we can now use RSA to encrypt messages.

4. EXERCISES

- (1) Encode your birthday using ASCII as described above to get an integer M .
- (2) Use my public key

$$N = 913336127711006102170609898942942716906241096981411826716803$$

$$e = 65537$$

to encrypt your message to get an integer C . Enter this integer on the assignment in Canvas.

- (3) Go to <http://magma.maths.usyd.edu.au/calc/> and enter Factorization(913336127711006102170609898942942716906241096981411826716803); and click 'Submit' to find a factorization of $N = pq$. Note: The fact that the computer can factor my public key is an indication that the key is too small.
- (4) Use your knowledge of the factorization to compute $\phi(N) = (p - 1)(q - 1)$.
- (5) Use Extended Euclidean Algorithm to find the decryption exponent d , which is a positive integer that is an inverse of e modulo $\phi(N)$.
- (6) Use the decryption exponent to decrypt my secret message.

$$C = 902366426828977962222652187660968416915888050115990141594121$$

- (7) Use the decoding procedure described above to convert the integer to a string. Post this string to the Discussion Board in Python for RSA.

Bibliography

- [1] D. Atkins, M. Graff, A. K. Lenstra, and P. C. Leyland, *The magic words are squeamish ossifrage (extended abstract)*, Advances in cryptology—ASIACRYPT '94 (Wollongong, 1994), Lecture Notes in Comput. Sci., vol. 917, Springer, Berlin, 1995, pp. 263–277.
- [2] M. Gardner, *Mathematical games: A new kind of cipher that would take millions of years to break*, Scientific American (August 1977), 120–124.
- [3] T. Oliveira e Silva, S. Herzog, and S. Pardi, *Empirical verification of the even Goldbach conjecture and computation of prime gaps up to $4 \cdot 10^{18}$* , Math. Comp. **83** (2014), no. 288, 2033–2060.
- [4] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), no. 2, 120–126.
- [5] K. H. Rosen, *Discrete mathematics and its applications*, McGraw-Hill Higher Education, 2012.

Index

- r*-combination, 157
- r*-permutation, 155

- Absorption laws, 19, 53
- and, 4
- antisymmetric, 164
- arithmetic progression, 71
- Associative laws, 19, 53

- base *b* expansion, 89
- base case, 130
- basis step, 130
- biconditional, 7
- bijection, 65
- bijective, 65
- binary, 89, 91
- binary relation, 163

- Caesar cipher, 120
- Cantor diagonalization argument, 144
- cardinality, 44, 142
- Cartesian product, 45
- casting out nines, 96
- ceiling, 60
- Chinese Remainder Theorem, 115
- closed formula, 73
- codomain, 56
- combination, 157
- combinatorics, 141
- common difference, 71
- common ratio, 70

- Commutative laws, 19, 53
- complement, 50
- Complement laws, 53
- Complementation law, 53
- composite, 99, 166
- composition, 67, 166
- compound propositions, 3
- conclusion, 6
- conditional, 6
- congruence class, 84
- congruent, 83
- conjunction, 4
- constructive existence proof, 35
- contingency, 17
- contradiction, 17
- contrapositive, 9
- Contrapositive law, 19
- converse, 9
- coprime, 103
- count, 141
- countable, 142
- counterexample, 23

- dangerous bend, xii
- De Morgan's laws for propositions, 18
- De Morgan's laws for quantifiers, 26
- De Morgan's laws for sets, 54
- decimal, 89
- decoding, 121
- decryption exponent, 123

- difference, 48
- direct proof, 30
- discrete log problem, 123
- disjoint, 50
- disjunction, 5
- Distributive laws, 19, 53
- dividend, 83
- divides, 81
- Division algorithm, 82
- Division Rule, 149
- divisor, 83
- domain, 23, 56
- Domination laws, 19, 53
- Double negation law, 19

- elements, 41
- empty set, 42
- encoding, 121
- encrypting, 121
- encryption exponent, 122
- equal, 42
- equivalence class, 172
- equivalence relation, 169
- equivalent, 169
- Euclidean algorithm, 105
- Euler phi function, 104
- Euler totient function, 104
- Euler's generalization, 118
- even, 31
- exclusive disjunction, 6
- exclusive or, 6
- existential quantification, 24
- existential quantifier, 24
- Extended Pigeonhole Principle, 153
- Extended Euclidean Algorithm, 106

- factor, 81
- fast exponentiation, 92–94
- Fermat's little theorem, 118
- Fibonacci sequence, 73
- finite set, 44
- floor, 59
- function, 56

- geometric progression, 70
- Goldbach conjecture, 103
- graph, 163
- greatest common divisor, 103

- hexadecimal, 90
- hypothesis, 6

- Idempotent laws, 19, 53
- Identity laws, 19, 53
- if and only if, 7
- if then, 6
- iff, 7
- image, 56, 57
- inductive hypothesis, 130
- inductive step, 130
- Infinitude of primes, 102
- initial term, 70, 71
- injection, 61
- injective, 61
- injectivity, 60
- integers, 41
- integers mod m , 85
- intersection, 48
- inverse, 9, 66, 113
- irrational, 34

- least common multiple, 104
- linear congruence, 113
- logically equivalent, 18

- mapping, 56
- mathematical induction, 129, 130
- members, 41
- membership table, 52
- modular arithmetic, 83
- multiple, 81

- negation, 4
- Negation law, 19
- nonconstructive proofs, 36
- number theory, 81

- odd, 31
- one-to-one, 61
- one-to-one correspondence, 65
- onto, 63
- or, 5

- partition, 173
- permutation, 155
- Pigeonhole Principle, 152
- power set, 44
- predicate, 22

- preimage, 56
- prime, 99
- Prime Number Theorem, 102
- primitive root, 86
- Principle of Mathematical Induction, 130
- Principle of Strong Induction, 136
- Product Rule, 146
- proof, 1
- proof by contradiction, 34
- proof by contraposition, 33
- proper subset, 42
- proposition, 2
- propositional equivalence, 16
- propositional function, 22
- propositional logic, 1
- propositional variables, 3

- quantification, 23
- quotient, 83

- range, 57
- rational, 34
- rational numbers, 42
- real numbers, 42
- recurrence relation, 72
- reflexive, 164
- related, 163
- relation, 46, 163
- relatively prime, 103
- remainder, 83
- roster method, 41

- sequence, 70
- set, 41
- set builder notation, 41
- shift cipher, 121
- Sieve of Eratosthenes, 100
- solution, 72
- strong induction, 136
- subject, 22
- subset, 42
- Subtraction Rule, 149
- sum, 74
- Sum Rule, 148
- surjection, 63
- surjective, 63
- surjectivity, 63

- symmetric, 164
- symmetric difference, 49

- tautology, 17
- term, 70
- theorem, 1
- Tower of Hanoi, 132
- transformation, 56
- transitive, 164
- truth table, 9
- truth value, 2
- Twin prime conjecture, 102
- twin primes, 102

- uncountable, 142
- union, 47
- universal quantification, 23
- universal quantifier, 23

- well-ordering property, 137, 138
- witness, 24, 35

- xor, 6