Discrete Structures

I always wondered why math textbooks put strange abstract art on the cover. I get it now – it's just a prank, bro.

Someone
Somewhere

Justin Goodman
University of Maryland
Dept. Computer Science
Dept. Mathematics
jugoodma@umd.edu

Updated
February 24, 2021

This one goes out to the homies

Justin

Contents

Chapter 0

Introduction

> In the beginning the Universe was created. This has made a lot of people very angry and been widely regarded as a bad move.
>
> Douglas Adams

Welcome! This book is essentially my own typed-up notes/explanations from the *Discrete Structures* (CMSC250) course at UMD. There may be typos – my apologies. My hope is to improve your overall understanding of the course content. If you're willing to put forth the effort to *understand* the course material, and think mathematically, then you will succeed.

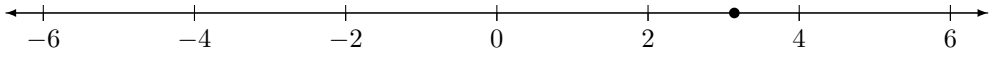
0.1 What is Discrete Mathematics?

Sometimes the actual name of the course is never addressed in-class. The 'mathematics' part hopefully should be familiar – numbers and stuff. What does 'discrete' mean? Well, Merriam-Webster provides two good definitions:

Definition 0.1.1 Discrete

- consisting of distinct or unconnected elements
- taking on or having a finite or countably infinite number of values

Consider the real number line:



We can point to any position on the line and find some real number to represent that point. It may have some long string of decimal digits, however we can do it. In the example above, the point is $\pi \approx 3.14159$. We refer to this real line as *continuous*. Broken up in-between all of these real numbers is a set of discrete points $\{\cdots, -2, -1, 0, 1, 2, \cdots\}$. These points are not continuous, and hence discrete, since we must make a jump (on the real line) from number to number. Our goal in this class is to examine these discrete numbers (which we call the *Integers* $\mathbb{Z}$) and structures.

0.2 Why Should You Care?

It seems somewhat pointless just to learn this whole proof crap (that you figured you never had to do again – see high school Geometry) when you will probably never use it again. So, here (in no particular order) is a list of reasons for and against learning this material.

FOR	AGAINST
• You become a free-thinker, and you become more skeptical	• Proofs are boring and math sucks
• You begin to question your world and your own perceptions	• Other people can do the proofs and math for me – I am willing to accept their work
• The mathematical mindset will help you throughout your life, and will be especially useful while coding	• Discrete math is just a weed-out class
• Your creativity comes back, and you may end up enjoying the course	• I do not need math to know how to code
• Computer science is just applied mathematics	• Discrete math will not improve my programming

If you disagree with any of the FOR reasons, then you should strongly consider why you chose computer science. We now provide counter-arguments to each of the above AGAINST points.

First, proofs and math are very akin to problem-solving. Programming is entirely problem-solving. If you do not enjoy problem-solving, then you are going to hate any programming job you get and you should really consider why you chose computer science.

Second, having other people do the leg-work will only get you so far in life. Eventually, someone is going to ask you to do the hard part too. Plus, if you blindly accept other people's work without checking it for yourself, you become

vulnerable to mistakes. Unchecked work leads to mistakes, which leads to wasted money and lost revenue, which leads to you getting fired from your job.

Third, discrete math is a weed-out class in the sense of the previous point. It is a basis for computer science, and if you do poorly in it then that may be an indicator of your future success in the field.

Fourth, yeah right.

Fifth, you keep telling yourself that while the top programmers at Google use mathematical reasoning to create super-fast mapping algorithms.

0.3 An Exercise in Proof

Maybe you are not convinced that proof is useful and necessary. Well, consider the following problem.

> Draw a circle and place n points on the edge of the circle. Draw a line between each possible pair of points. This will yield closed regions within the circle. How many such regions are there for $n \geq 1$ points?

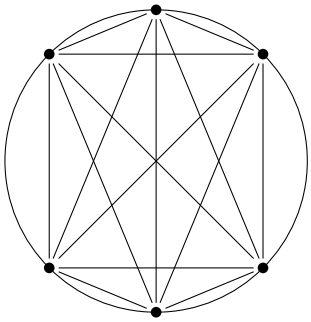
Try it for $n = 1, 2, 3, 4, 5$. We will ignore $n = 0$ for this exercise. You should get:

n	1	2	3	4	5
Regions	1	2	4	8	16

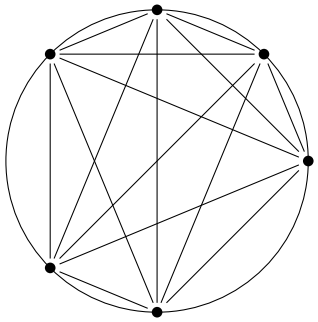
There's a clear pattern here – the number of regions for n points looks like 2^{n-1}. Go ahead and try this formula for the values of n we have – it works!

How confident are you in this formula?

Draw out a circle with $n = 6$ points, and compare your answer to the formula we came up with.



30 Regions



31 Regions

Are you still not convinced that our formula is wrong? Try drawing a circle for $n = 7$ and $n = 8$, and compare your regions to your formula.

This problem is sometimes called *Moser's circle problem*, and is often used to show the flaws of generalizing from observations without sufficient basis.

0.4 Thinking Mathematically

Our goal is to teach you how to think mathematically. We want you to ask and answer questions based on logic. We want you to have sound reasoning. We want you to be able to back-up your answers with solid proof. We want your explanations to make sense, and we want to be convinced. In the real-world, no one is going to just *accept* things you say unless you have good *evidence* to back your claims. Our goal is to teach you how to (1) formally write your claims, and (2) provide proof for your claims. Mathematical thinking comes naturally for some, however for others it may take a whole semester. Keep working at it until you succeed.

0.5 How to Succeed in this Course

This course will provide you a toolkit of concepts from which you can pull to solve problems. Unlike methodological/algorithm-based courses (e.g. Calculus III), this course highly depends on your solid *understanding* of the material. To succeed in this course, you will need to immerse yourself in the material. You will need to struggle through it until you understand. For some of you, this will be quick and easy. For others, not so much. That is okay. You will succeed so long as you put in the effort.

0.6 Summary

In order to succeed, you must:

- Immerse yourself in the course content
- Struggle until you understand
- Think mathematically

Chapter 1

Logic

> All opinions are not equal. Some are a very great deal more robust, sophisticated and well supported in logic and argument than others.
>
> Douglas Adams

Contents

1.1 Introduction

Let's spell out two logic puzzles:

Sam has 1 cow. If Sam has at least 2 cows, then Sam can breed the cows to make one more cow. Assuming Sam has access to infinite resources and time, how many cows can Sam make?

Two givens: knights always tell the truth, and knaves always lie. On the island of knights and knaves, you are approached by two people. The first one says to you, "we are both knaves." What are they actually? (from Popular Mechanic's Riddle of the Week #43: Knights and Knaves, Part 1)

Thinking logically about these puzzles will help you – think about what can and cannot happen; what can and cannot be true. Solving these problems is left as an exercise. There are only two possibilities for Boolean statements – True or False. Here's a formal definition of logic, from Merriam-Webster:

Definition 1.1.1 Logic
A science that deals with the principles and criteria of validity of inference and demonstration : the science of the formal principles of reasoning

This chapter includes a wealth of topics. We will touch on propositional logic and its corollaries, as well as predicate logic and it's applications to the rest of this course. Thinking logically should be a natural process, so we hope this section is relatively straightforward.

1.2 Propositional Logic

Propositional logic is a branch of logic that deals with simple propositions and logical connectives. Sometimes propositional logic is referred to as **zeroth-order logic**, as it lays the foundations for *predicate logic*, also known as *first-order logic*.

Definition 1.2.1 Proposition
A statement that is exclusively either true or false. A proposition must *have* a true or false value, and it cannot have *both*. Propositions are usually represented as variables. These variables can represent a single statement, or large compound statements. We will see examples later

Definition 1.2.2 Logical Connective
An operation that connects two propositions. We study these below

The motivation behind propositional logic is that we want to represent basic logical statements as an expression of variables and operators. Propositional logic also lays the groundwork for higher-order logic.

1.2.1 Truth Tables and Logical Connectives

Before we dive into the logical connectives, let's study the notion of a truth table. This will help us fully understand the logical connectives.

Definition 1.2.3 Truth Table
A table that shows us all truth-value possibilities. For example, with two propositions p and q:

p	q	*some compound proposition*
F	F	T/F
F	T	T/F
T	F	T/F
T	T	T/F

Now we can begin our study of the logical connectives. The following definitions explain the intuition behind the logical connectives, and present their associated truth tables.

Definition 1.2.4 And $\wedge$
Also known as the *conjunction*. Logical connective that evaluates to true when the propositions that it connects are both true. If either proposition is false, then *and* evaluates to false. To remember: *prop 1* **and** *prop 2* must *both* be true. Truth table:

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

Notice the only row that evaluates to true is when both propositions are true

Definition 1.2.5 Or $\vee$
Also known as the *disjunction*. Logical connective that evaluates to true when either of the propositions that it connects are true (at least 1 of the connected propositions is true). If both propositions are false, then *or* evaluates to false. **Note**: if both propositions are true, then *or* still evaluates to true. To remember: either *prop 1* **or** *prop 2* must be true. Truth table:

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

Notice the only row that evaluates to false is when both propositions are false

Definition 1.2.6 Not $\neg$, $\sim$
Also known as the *negation*. Logical connective that flips the truth value of the proposition to which it is connected. Unlike *and* and *or*, *not* only affects 1 proposition. Truth table:

p	$\neg p$
F	T
T	F

Definition 1.2.7 Implication/Conditional $\Rightarrow$, $\rightarrow$
Logical connective that reads as an *if-then* statement. The implication must be false if the first proposition is true and the implied (connected/second) proposition is false. Otherwise it is true. Truth table:

p	q	$p \Rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

Note: the direction of an implication can be flipped: $p \Leftarrow q$ is the same as $q \Rightarrow p$

Let's try to understand the truth table for the implication statement before we continue. We present two examples that attempt to form an intuitive analogy to the implication.

Example: Think of the implication as a vending machine. p is the statement *we put money into the vending machine*, and q is the statement *we received a snack from the vending machine*. Notice that the statements do not necessarily depend on each other. We examine the four cases and see when we are *unhappy*:

1. p is **false** and q is **false** – we did not put in money, and we did not get a snack, so we remain happy (normal operations)
2. p is **false** and q is **true** – we did not put in money, and we did get a snack, so we are very very happy (free snack!)
3. p is **true** and q is **false** – we did put in money, and we did not get a snack, so we are very very unhappy (we got robbed!)
4. p is **true** and q is **true** – we did put in money, and we did get a snack, so we are happy (normal operations)

When we are unhappy, then the implication statement is false. Otherwise it is true (we are not *unhappy*).

Example: Think of the implication in the lens of a program. You want to evaluate whether your program *makes sense*. Here is the example program from the statement $p \Rightarrow q$:

```
(...)
if (p is true) {
        <run body code if q is true>
}
(...)
```

The body code is run only if q is true. Now let's examine the 4 cases and see whether the program makes sense:

1. p is **false** and q is **false** – the program does not go into the body of the if-statement and hence makes sense
2. p is **false** and q is **true** – the program again does not go into the body of the if-statement and hence makes sense (regardless of the value of q)
3. p is **true** and q is **false** – the program goes into the body of the if-statement but since q is false the program does not evaluate the body code. This does not make sense
4. p is **true** and q is **true** – the program goes into the body of the if-statement and evaluates the body code. This makes sense

When the code evaluator makes sense, then the implication statement is true.

Definition 1.2.8 Bi-conditional $\Leftrightarrow$, $\leftrightarrow$
Logical connective that reads as an *if and only if* statement. This means that both propositions must imply each other. For the bi-conditional to be true, both propositions must either be true or false. Truth table:

p	q	$p \Leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Some more complicated ones:

Definition 1.2.9 Exclusive Or (Xor) $\oplus$
Logical connective that evaluates to true when *only* one of the two propositions that it connects is true. If both propositions are true or false, then *xor* evaluates to false. The exclusive part means we *exclude* the *or* case when both propositions are true. Truth table:

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

Definition 1.2.10 Exclusive Nor ($Xnor$) $\otimes$
Logical connective that negates the *xor*. It is just an *xor* connective appended with a *not* connective. Truth table:

p	q	$\neg(p \oplus q) \equiv (p \otimes q)$
F	F	T
F	T	F
T	F	F
T	T	T

Now an example:

Example: Translate the following statement into a propositional logic statement: *exclusively either the weather rains or students wear rain jackets.*

Solution: Let r be the proposition *the weather rains* and j be the proposition *students wear rain jackets.* Then the statement becomes $r \oplus j$

1.2.2 Boolean Algebra

The motivation behind Boolean algebra is that we want to take complicated compound propositional statements and simplify them. If we notice that a variable does not affect the final output, then getting rid of that variable cuts the amount of truth-value possibilities (truth-table rows) in half.

Definition 1.2.11 Boolean Statement
A statement that is exclusively either true or false

Some handy notations:

Definition 1.2.12 Equivalence $\equiv$
Logical equivalence says that the two connected statements are logically the same. You can think of this notation as the *equals* sign. Equality is poorly defined for Boolean expressions, so we use the equivalence notation instead

Definition 1.2.13 Tautology t *or* **T**
A proposition that is always true

Definition 1.2.14 Contradiction c *or* **F**
A proposition that is always false

We provide a handful of helpful theorems to aid in your Boolean algebra simplifications. You do not need to memorize these theorems – they will be given to you as a table.

Theorem 1.2.1 Commutativity
*For any propositions p and q the **and** and **or** operations are commutative:*

$$p \vee q \equiv q \vee p$$

$$p \wedge q \equiv q \wedge p$$

Theorem 1.2.2 Associativity
*For any propositions p, q, and r the **and** and **or** operations are associative:*

$$(p \vee q) \vee r \equiv p \vee (q \vee r)$$

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$$

Theorem 1.2.3 Distributivity
*For any propositions p, q, and r the **and** and **or** operations are distributive:*

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

Theorem 1.2.4 Identity
For any proposition p the following hold:

$$p \vee c \equiv p$$

$$p \wedge t \equiv p$$

Theorem 1.2.5 Negation
For any proposition p the following hold:

$$p \vee \neg p \equiv t$$

$$p \wedge \neg p \equiv c$$

Theorem 1.2.6 Double Negation
For any proposition p the following holds:

$$\neg(\neg p) \equiv p$$

Theorem 1.2.7 Idempotence
For any proposition p the following hold:

$$p \vee p = p$$

$$p \wedge p = p$$

Theorem 1.2.8 De Morgan's
For any propositions p and q the following hold:

$$\neg(p \vee q) = \neg p \wedge \neg q$$

$$\neg(p \wedge q) = \neg p \vee \neg q$$

Theorem 1.2.9 Universal Bound
For any proposition p the following hold:

$$p \vee t \equiv t$$

$$p \wedge c \equiv c$$

Theorem 1.2.10 Absorption
For any propositions p and q the following hold:

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

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

Theorem 1.2.11 Negation of Tautology and Contradiction
The following hold:

$$\neg t \equiv c$$

$$\neg c \equiv t$$

Here are two important theorems that you will use throughout your proofs in this course. The first theorem says that for a bi-conditional both propositions must imply each other. The second theorem gives an equivalence between the implication and *or* connective. The proofs of these theorems are left as an exercise to the reader.

Theorem 1.2.12 Bi-conditional to Implication
For any propositions p and q the following hold:

$$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$$

Theorem 1.2.13 Implication to Disjunction
For any propositions p and q the following holds:

$$p \Rightarrow q \equiv \neg p \vee q$$

Now some examples of Boolean statement simplification.

Example: Simplify the following expression: $(p \Rightarrow q) \Rightarrow r$

Solution:

$$
\begin{aligned}
(p \Rightarrow q) \Rightarrow r &\equiv \big(\neg(p \Rightarrow q)\big) \vee r && \text{Implication to Disjunction} \\
&\equiv \big(\neg((\neg p) \vee q)\big) \vee r && \text{Implication to Disjunction} \\
&\equiv ((\neg(\neg p)) \wedge (\neg q)) \vee r && \text{De Morgan's} \\
&\equiv (p \wedge (\neg q)) \vee r && \text{Double Negation}
\end{aligned}
$$

Notice how we reference a theorem in each step. This allows us to fully explain our equivalence, keeps us from making mistakes, and ensures our equivalence is valid.

Example: Simplify the following expression: $(p \otimes q) \wedge p$

Solution:

$$
\begin{aligned}
(p \otimes q) \wedge p &\equiv \neg(p \oplus q) \wedge p && \text{XNOR equivalence} \\
&\equiv \neg\big((p \wedge (\neg q)) \vee ((\neg p) \wedge q)\big) \wedge p && \text{XOR equivalence} \\
&\equiv \big(\neg(p \wedge (\neg q)) \wedge \neg((\neg p) \wedge q)\big) \wedge p && \text{De Morgan's} \\
&\equiv \big(((\neg p) \vee \neg(\neg q)) \wedge (\neg(\neg p) \vee (\neg q))\big) \wedge p && \text{De Morgan's} \\
&\equiv \big(((\neg p) \vee q) \wedge (p \vee (\neg q))\big) \wedge p && \text{Double Negation} \\
&\equiv ((\neg p) \vee q) \wedge \big((p \vee (\neg q)) \wedge p\big) && \text{Associativity} \\
&\equiv ((\neg p) \vee q) \wedge \big(p \wedge (p \vee (\neg q))\big) && \text{Commutativity} \\
&\equiv ((\neg p) \vee q) \wedge p && \text{Absorption} \\
&\equiv p \wedge ((\neg p) \vee q) && \text{Commutativity} \\
&\equiv (p \wedge (\neg p)) \vee (p \wedge q) && \text{Distributivity} \\
&\equiv c \vee (p \wedge q) && \text{Negation} \\
&\equiv p \wedge q && \text{Identity}
\end{aligned}
$$

Note in the preceding example that we used equivalences between the *xnor* to the *xor*, and the *xor* to the *or*. We will discuss later exactly how we deduced these equivalences.

Now that we understand some equivalences, we can motivate some definitions relating to the implication.

Definition 1.2.15 Converse
The converse of $p \Rightarrow q$ is $q \Rightarrow p$. Obtain this by reversing the arrow direction.

Definition 1.2.16 Inverse
The inverse of $p \Rightarrow q$ is $(\neg p) \Rightarrow (\neg q)$. Obtain this by negating both propositions.

Definition 1.2.17 Contrapositive
The contrapositive of $p \Rightarrow q$ is $(\neg q) \Rightarrow (\neg p)$. Obtain this by reversing the arrow direction, and negating both propositions. Or, take both the converse and inverse.

Definition 1.2.18 Negation
The negation of $p \Rightarrow q$ is $\neg(p \Rightarrow q)$. Obtain this by negating the entire implication.

Now we can use our equivalencies to show some important facts about these definitions.

Theorem 1.2.14 Contraposition Equivalence

$$p \Rightarrow q \equiv (\neg q) \Rightarrow (\neg p)$$

We leave the proof as an exercise. This theorem will serve us well in our study of Number Theory. The contrapositive indirect proof technique relies on this fact.

Theorem 1.2.15 The Converse Error

$$(p \Rightarrow q) \not\equiv (q \Rightarrow p)$$

Proof. We want to show that if you have $p \Rightarrow q$, then you do not necessarily have $q \Rightarrow p$. There are many ways to show this, but we will start by simply examining cases of p and q.
Consider $p \equiv \mathbf{F}$ and $q \equiv \mathbf{T}$. Then $p \Rightarrow q \equiv \mathbf{F} \Rightarrow \mathbf{T} \equiv \mathbf{T}$. But then $q \Rightarrow p \equiv \mathbf{T} \Rightarrow \mathbf{F} \equiv \mathbf{F}$. Then overall we have $(p \Rightarrow q) \Rightarrow (q \Rightarrow p) \equiv \mathbf{T} \Rightarrow \mathbf{F} \equiv \mathbf{F}$. So we've shown that if you have $p \Rightarrow q$, then you do not necessarily have $q \Rightarrow p$. □

Theorem 1.2.16 The Inverse Error

$$(p \Rightarrow q) \not\equiv ((\neg p) \Rightarrow (\neg q))$$

Remark 1.2.19
A similar proof can be made for this theorem. Find one truth-value pairings for p and q such that $p \Rightarrow q$ is true, but $(\neg p) \Rightarrow (\neg q)$ is false.

1.2.3 Circuits

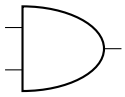
Propositions have two possible values: true and false. If we set true to mean *on* and false to mean *off*, then we can translate our Boolean statements into logical circuits. To do this, think of each logical connective as a *gate*. Similar to the logical connectives, a gate takes 2 (or more) inputs and returns some output. The inputs and outputs are all 1s and 0s (*on*s and *off*s – true and false). Circuits are the bare-bones to computers, so it is necessary you understand the basics. For computer engineers, you must know circuits by heart.

Definition 1.2.20 Circuit
Representations of Boolean statements into electronic components

Boolean variables can be exclusively either true or false; this is analogous to electric wires being exclusively either on or off. We let a tautology be equivalent to a *power source*, which is a wire that is always **on**. Similarly we let a contradiction be equivalent to **off**, or a wire receiving no power.

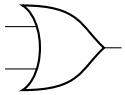
The following gates are exactly equivalent to their logic counterparts. We thus only include the gate picture.

Definition 1.2.21 And Gate – *conjunction*
Picture:

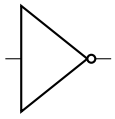


To remember the *and* gate, note that the picture looks like a **D**, which corresponds to the D in AN**D**

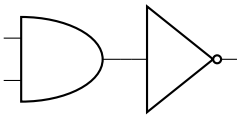
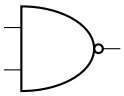
Definition 1.2.22 Or Gate – *Disjunction*
Picture:



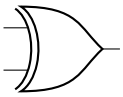
Definition 1.2.23 Not Gate – *Negation*
Picture:



Remark 1.2.24
When we have a gate that has a *not* after it, we can simplify the gate by just adding a circle to the output. Example (the *nand* gate is the negation of the *and* gate):


becomes


Definition 1.2.25 Xor Gate (*eXclusive Or gate*)
Picture:



We can think of truth tables in an equivalent fashion, where 1 is true and 0 is false.

Example: Write the truth table for the *nand* gate.

Solution:

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

You can use 1/0 or T/F, whichever you prefer. If an assignment specifies you use a specific key, then use the one specified. Later when we discuss different number systems, you will notice a correlation between binary numbers and truth tables with 1/0s. This makes it easy to construct a truth table very quickly with a given amount of inputs.

We can also create circuits that add numbers. We will come back to this after our discussion of different number bases in section 3.2.9.

1.2.4 Translations

You must know how to translate between circuits, Boolean statements, and truth tables.

We start by motivating the translation between truth tables and Boolean statements. First, notice that we have already discussed how to make a truth table from a Boolean statement – simply draw the truth table and fill in each row. Now we can focus on the reverse.

We attempt to first motivate a process for this translation by showing a few examples.

Example: First recall the truth table for the conjunction:

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

Recall that we can extend the conjunction to take multiple inputs – in this case, *each* input *must* be **True** for the output to be **True**.
Now consider the following unknown table:

p	q	unknown
F	F	F
F	T	T
T	F	F
T	T	F

This table is similar to the conjunction table in that it has *one row* that outputs **True**. In the conjunction case, the assignments to p and q were both **True**. What are the assignments to the previous truth table? Well, the p input is **False** and the q input is **True**, as per the table. If we apply the conjunction to this row we have, we get the following table:

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

In the previous table, if we look at the middle two columns, we recover the conjunction with the p variable negated.

This is only half the story though.

Example: First recall the truth table for the disjunction:

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

For this example, we only care about the middle two rows where *either* variable is **True**.

These bring light to a simple 3-step process to translate from truth tables to Boolean statements:

1. Collect the rows whose *output* is **True**
2. For each of those rows
   (a) Look at the truth value assigned to the *input*

(b) Construct a Boolean statement where, for each input

- If the assignment is **True** then use the variable itself
- If the assignment is **False** then use the *negation* of the variable

(c) Chain each input with **And** ($\wedge$) connectives

3. Chain each row's statement with **Or** ($\vee$) connectives

Example: Find the unknown Boolean formula corresponding to the following truth table:

p	q	unknown
F	F	T
F	T	F
T	F	T
T	T	F

Solution: We follow the algorithm as before. The first and third rows return true in the output. In the first row, we see neither input is true, so we AND the negation of each input: $(\neg p) \wedge (\neg q)$. In the third row, we see the first input is true, while the second input is false, thus we get: $p \wedge (\neg q)$. OR-ing each statement together thus yields our unknown formula:

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

Remark 1.2.26
In our Boolean formula algorithm, we only focus on the true rows. If we wanted to also include the false rows, then we would need the statement in each false row to return false. We know how to get a statement that returns true, so we can simply take that statement and negate it! What happens, though, if we include the false rows then?

Example: Include the false rows in the Boolean formula from the previous example.

Solution: We negate the second and fourth row returned by the algorithm: $\neg((\neg p) \wedge q)$ and $\neg(p \wedge q)$
So our entire statement is now

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

Let's simplify this statement (rules omitted):

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

Nice, we have recovered the original statement!

Remark 1.2.27
Including each false row in the Boolean statement generated by the algorithm – so long as their true versions are negated – does not logically change the Boolean statement

We have a name for the type of statement generated from our algorithm above:

Definition 1.2.28 Disjunctive Normal Form
Describes a Boolean statement that is a conjunction of disjunctions – abbreviated DNF

If we flip each gate (AND goes to OR, OR goes to AND), then we get another important type of statement:

Definition 1.2.29 Conjunctive Normal Form
Describes a Boolean statement that is a disjunction of conjunctions – abbreviated CNF

Interestingly, any Boolean statement can be translated to an equivalent statement in CNF. Namely, statements in DNF, which are easy to generate, can be translated into CNF. This is important for computational complexity theory – specifically NP-completeness. There exists a problem in computer science which entails finding a set of truth-value assignments for n different Boolean variables which makes a Boolean statement in CNF return true (or, become *satisfiable*).

1.2.5 Reasoning/Deductions

Propositional logic also allows us to *reason* about things.

Definition 1.2.30 Knowledge Base
A group of information that you know is true

Definition 1.2.31 Reasoning
The process of deriving new information from a given knowledge base

See the introduction of this chapter for an example.

Classical Rules of Deduction

We may refer to *deductions* as *inferences*. They are the same.

Definition 1.2.32 Deductions
Using previously-known knowledge in your knowledge base to obtain/create new knowledge

We have a whole list of useful deductions that are provably valid. As with our Boolean algebra theorems, you do not need to memorize these theorems – they will be given to you as a table.

Theorem 1.2.17 Modus Ponens

$$\begin{array}{l} p \\ \underline{p \Rightarrow q} \\ \therefore q \end{array}$$

Theorem 1.2.18 Modus Tollens

$$\begin{array}{l} \neg q \\ \underline{p \Rightarrow q} \\ \therefore \neg p \end{array}$$

Theorem 1.2.19 Disjunctive Addition

$$\begin{array}{l} \underline{p} \\ \therefore p \vee q \end{array}$$

Theorem 1.2.20 Conjunctive Addition

$$\begin{array}{l} \underline{p, q} \\ \therefore p \wedge q \end{array}$$

Theorem 1.2.21 Conjunctive Simplification

$$\begin{array}{l} \underline{p \wedge q} \\ \therefore p, q \end{array}$$

Theorem 1.2.22 Disjunctive Syllogism

$$\begin{array}{l} p \vee q \\ \underline{\neg p} \\ \therefore q \end{array}$$

Theorem 1.2.23 Hypothetical Syllogism

$$\begin{array}{l} p \Rightarrow q \\ \underline{q \Rightarrow r} \\ \therefore p \Rightarrow r \end{array}$$

Theorem 1.2.24 Resolution

$$\begin{array}{l} p \vee q \\ \underline{(\neg q) \vee r} \\ \therefore p \vee r \end{array}$$

Theorem 1.2.25 Division Into Cases

$$\begin{array}{l} p \vee q \\ p \Rightarrow r \\ \underline{q \Rightarrow r} \\ \therefore r \end{array}$$

Theorem 1.2.26 Law of Contradiction

$$\begin{array}{l} \underline{(\neg p) \Rightarrow \mathbf{c}} \\ \therefore p \end{array}$$

You may be wondering how to prove these deductions are *valid*. We have two equivalent methods:

1. Tautological implication
2. Critical-row identification

Consider an arbitrary deduction:

$$\begin{array}{l} P_1 \\ P_2 \\ \cdots\cdots\cdots \\ \underline{P_n} \\ \therefore Q \end{array}$$

To prove it valid,

Tautological implication

1. Construct a new proposition $A \Rightarrow B$ where A is a **conjunction of the premises** and B is the conclusion. For our arbitrary deduction, we would have
$$\big(P_1 \wedge P_2 \wedge \cdots \wedge P_n\big) \Rightarrow Q$$

2. Inspect this proposition in a truth-table. If the proposition is a tautology, then the deduction is valid. Otherwise, the deduction is invalid. So, to be valid we must have
$$(A \Rightarrow B) \equiv \big(P_1 \wedge P_2 \wedge \cdots \wedge P_n\big) \Rightarrow Q \equiv t$$

Critical-row identification

1. Construct a truth-table with columns for each proposition and for the conclusion. *You may need extra columns, that is fine.* For our arbitrary deduction, we would have

P_1	P_2	$\cdots$	P_n	Q
F/T	F/T	$\cdots$	F/T	F/T
		$\vdots$		
T/T	F/T	$\cdots$	F/T	F/T

2. Identify the rows in which each **premise** is true. We call these rows **critical-rows**.

3. For each critical-row, inspect the conclusion. If the conclusion is true in **every** critical-row, then the deduction is valid. Otherwise, the deduction is invalid.

Examples following the above steps.

Example: Show that Modus Ponens is a valid rule of inference.

Solution: *method 1 – tautological implication*
Our proposition we care about is $(p \wedge (p \Rightarrow q)) \Rightarrow q$, so we build the following truth-table with extraneous rows:

p	q	p	$p \Rightarrow q$	$p \wedge (p \Rightarrow q)$	q	$(p \wedge (p \Rightarrow q)) \Rightarrow q$
F	F	F	T	F	F	T
F	T	F	T	F	T	T
T	F	T	F	F	F	T
T	T	T	T	T	T	T

Inspect the last column.

p	q	p	$p \Rightarrow q$	$p \wedge (p \Rightarrow q)$	q	$(p \wedge (p \Rightarrow q)) \Rightarrow q$
F	F	F	T	F	F	**T**
F	T	F	T	F	T	**T**
T	F	T	F	F	F	**T**
T	T	T	T	T	T	**T**

In the case of Modus Ponens, the final column is a tautology, hence the deduction is valid.

Example: Show that Modus Ponens is a valid rule of inference.

Solution: *method 2 – critical-row identification*
Construct a truth table with each premise and conclusion:

p	q	p	$p \Rightarrow q$	q
F	F	F	T	F
F	T	F	T	T
T	F	T	F	F
T	T	T	T	T

Identify the critical-rows.

p	q	p	$p \Rightarrow q$	q
F	F	F	T	F
F	T	F	T	T
T	F	T	F	F
T	**T**	**T**	**T**	T

Inspect the conclusion in each critical-row.

p	q	p	$p \Rightarrow q$	q
F	F	F	T	F
F	T	F	T	T
T	F	T	F	F
T	**T**	**T**	**T**	**T**

In the case of Modus Ponens, the conclusion is true in each critical row, hence the deduction is valid.

We leave it to the reader to understand why the two methods are equivalent.

Deducing Things

In a later section, we will see that this order of logic is not powerful enough to prove mathematical statements. For now, we can still do interesting things with a given knowledge base.

Example: Given the following knowledge base, deduce as much new information as possible using the following rules of inference: Modus Ponens, Modus Tollens, Hypothetical Syllogism, and Disjunctive Syllogism.

$a \Rightarrow b$ $\quad$ $b \Rightarrow (\neg d)$ $\quad$ e $\quad$ $d \vee (\neg e)$

Solution:

By Hypothetical Syllogism $a \Rightarrow b$, $b \Rightarrow (\neg d)$, $\therefore a \Rightarrow (\neg d)$
By Disjunctive Syllogism $d \vee (\neg e)$, e, $\therefore d$
By Modus Tollens d, $a \Rightarrow (\neg d)$, $\therefore \neg a$
By Modus Tollens d, $b \Rightarrow (\neg d)$, $\therefore \neg b$

Remark 1.2.33
From the above example, we restricted the rules you could have used. We did this mainly because Disjunctive Addition can allow you to generate any new knowledge you like – so long as you have one thing that is true, then you can add in a disjunction infinitely-many times.

Remark 1.2.34
In the above example, we could have translated $d \vee (\neg e) \equiv e \Rightarrow d$ and concluded d by Modus Ponens. This somewhat tells you that Disjunctive Syllogism and Modus Ponens are equivalent.

Remark 1.2.35
From an inconsistent database, anything follows.

This is due to the law of contradiction. An *inconsistent database* is one that contains a contradiction. Recall from the law of contradiction that $(\neg p) \Rightarrow c \equiv (\neg(\neg p)) \vee c \equiv p$ using the Identity Boolean algebra theorem. Using Disjunctive Addition, we have the contradiction c, $\therefore A \vee c$, and by Identity, $\therefore A$. A can be *Anything*.

In Artificial Intelligence, there exists an algorithm called **The Resolution Algorithm**. Essentially, it says to take a given knowledge base, translate each statement into disjunctive normal form, then apply the Resolution rule of inference as many times as possible.

1.3 Predicate Logic

Sometimes basic propositions are not enough to do what you want. In programming we can have functions that return true or false. We can do the same thing with logic – we call this *first-order* logic, or *predicate* logic. Predicate logic includes all of propositional logic, however it adds predicates and quantifiers.

Definition 1.3.1 Predicate
A property that the subject of a statement can have. In logic, we represent this sort-of like a function. A predicate takes, as input, some element, and returns whether the inputted element has the specific property.

Example: We could use the predicate $EVEN(x)$ to mean x is an even number. In this case, the predicate is $EVEN(\cdot)$

Example: We could use the predicate $P(y)$ to mean y is an integer multiple of 3. In this case, the predicate is $P(\cdot)$

Remark 1.3.2
Predicates take **elements**. They do **not** take in other predicates. This is because predicates say whether the input element *has* the property specified by the predicate – true and false cannot have properties.

In terms of programming, you can think of a predicate as a program method. For example, the $EVEN(x)$ predicate might be implemented as follows:

```
func EVEN(Entity x) -> bool {
        if IS_INTEGER(x) {
                return x % 2 == 0
        }
        return false
}
```

In this case, entities are *objects* and true/false are *Boolean primitives* (or, propositional statements, which can only be true or false). In contrast to, say, Java, a compiler for this code would not allow *true/false* to be an object.

A better example,

```
class Foo extends Entity {
        bool isInteger
        bool isOdd

        Foo(Integer i) {
                this.isInteger = true
                this.isOdd = i % 2 == 1
        }
}

func ODD(Entity x) -> bool {
        if x has type Foo {
                return x.isOdd
```

```
        }
        if IS_INTEGER(x) {
                return x % 2 == 1
        }
        return false
}
```

Definition 1.3.3 Quantifier
A way to select a specific range of elements that get inputted to a predicate. We have two quantifiers:

- The Universal quantifier $\forall$
- The Existential quantifier $\exists$

The universal quantifier says to select **all** elements, and the existential quantifier says to select **at least one** element.

Definition 1.3.4 Quantified Statement
A logical statement involving predicates and quantifiers. Syntax:

$$(\text{quantifier } var \in D)[\text{statement involving predicates}]$$

And now we can define:

Definition 1.3.5 Predicate Logic
Also called *first-order logic*, is a logic made up of quantified statements.

Example: Translate the following statements to predicate logic:
1. All people are mortal
2. Even integers exist
3. If an integer is prime then it is not even

Solution:
1. Denote P as the domain of people, and the predicate $M(x)$ to mean x is mortal. Then the statement translates to
$$(\forall p \in P)[M(p)]$$
2. Denote $\mathbb{Z}$ as the domain of integers, and the predicate $EVEN(x)$ to mean x is even. Then the statement translates to
$$(\exists x \in \mathbb{Z})[EVEN(x)]$$
3. Denote the predicate $PRIME(y)$ to mean y is prime. Then the

statement translates to

$$(\forall a \in \mathbb{Z})[PRIME(a) \Rightarrow \neg EVEN(a)]$$

1.3.1 Negating Quantified Statements

One may find useful to negate a given quantified statement. We present here how to do this, first with an English example, followed by a quantified example, followed by an algorithm.

Example: The following statement

There is no student who has taken calculus.

is equivalent to

All students have not taken calculus.

Example: The following statement

Not all students have taken calculus.

is equivalent to

There is a student who has not taken calculus.

Example: The following statement

$$\neg(\exists x \in D)[C(x)]$$

is equivalent to

$$(\forall x \in D)[\neg C(x)]$$

Example: The following statement

$$\neg(\forall x \in D)[C(x)]$$

is equivalent to

$$(\exists x \in D)[\neg C(x)]$$

The generic algorithm for pushing the negation into a quantified statement:

1. Flip each quantifier $\forall \to \exists$ and $\exists \to \forall$
2. Apply the negation to the propositional part of the quantified statement, and simplify
   (a) If the inside contains another quantified statement, then recursively apply this algorithm

Remark 1.3.6
The domain and variable attached to any quantifier are **not** changed.

Example: Push the negation in as far as possible:

$$\neg(\forall x, y \in \mathbb{Z})[(x < y) \Rightarrow (\exists m \in \mathbb{Q})[x < m < y]]$$

Solution:

$$\begin{aligned}
&\neg(\forall x, y \in \mathbb{Z})[(x < y) \Rightarrow (\exists m \in \mathbb{Q})[x < m < y]] \\
\equiv&(\exists x, y \in \mathbb{Z})\neg[(x < y) \Rightarrow (\exists m \in \mathbb{Q})[x < m < y]] \\
\equiv&(\exists x, y \in \mathbb{Z})\neg[\neg(x < y) \vee (\exists m \in \mathbb{Q})[x < m < y]] \\
\equiv&(\exists x, y \in \mathbb{Z})[\neg\neg(x < y) \wedge \neg(\exists m \in \mathbb{Q})[x < m < y]] \\
\equiv&(\exists x, y \in \mathbb{Z})[(x < y) \wedge (\forall m \in \mathbb{Q})\neg[x < m < y]] \\
\equiv&(\exists x, y \in \mathbb{Z})[(x < y) \wedge (\forall m \in \mathbb{Q})\neg[x < m \wedge m < y]] \\
\equiv&(\exists x, y \in \mathbb{Z})[(x < y) \wedge (\forall m \in \mathbb{Q})[x \geq m \vee m \geq y]]
\end{aligned}$$

Remark 1.3.7
We typically expect the final statement to contain no $\neg$ operators.

1.3.2 Quantified Rules of Inference

Again, *deductions* and *inferences* are the same. We present a handful of important *quantified* rules of inference.

Theorem 1.3.1 Universal Instantiation
For a predicate $P(\cdot)$ and some domain D with $c \in D$,

$$\therefore \frac{(\forall x \in D)[P(x)]}{P(c)}$$

As an example, if our domain consists of all dogs and Fido is a dog, then the above rule can be read as

"All dogs are cuddly"

"Therefore Fido is cuddly"

Theorem 1.3.2 Universal Generalization
For a predicate $P(\cdot)$ and some domain D for an arbitrary $c \in D$,

$$\therefore \frac{P(c)}{(\forall x \in D)[P(x)]}$$

This is most-often used in mathematics. As an example, if our domain consists of all dogs, then the above rule can be read as

"An arbitrary dog is cuddly" (which in-turn applies to all dogs)

"Therefore all dogs are cuddly"

Theorem 1.3.3 Existential Instantiation
For a predicate $P(\cdot)$ and some domain D for some element $c \in D$,

$$\begin{underline{(\exists x \in D)[P(x)]}} \\ \therefore P(c)$$

As an example, if our domain consists of all dogs, then the above rule can be read as

"There is a dog who is cuddly"

"Let's call that dog c, and so c is cuddly"

Theorem 1.3.4 Existential Generalization
For a predicate $P(\cdot)$ and some domain D for some element $c \in D$,

$$\begin{underline{P(c)}} \\ \therefore (\exists x \in D)[P(x)]$$

As an example, if our domain consists of all dogs and Fido is a dog, then the above rule can be read as

"Fido is cuddly"

"Therefore there is a dog who is cuddly"

Theorem 1.3.5 Universal Modus Ponens
For two predicates $P(\cdot)$ and $Q(\cdot)$, and some domain D with $a \in D$,

$$\begin{underline{P(a) \\ (\forall x \in D)[P(x) \Rightarrow Q(x)]}} \\ \therefore Q(a)$$

Theorem 1.3.6 Universal Modus Tollens
For two predicates $P(\cdot)$ and $Q(\cdot)$, and some domain D with $a \in D$,

$$\begin{underline{\neg Q(a) \\ (\forall x \in D)[P(x) \Rightarrow Q(x)]}} \\ \therefore \neg P(a)$$

1.3.3 Proving Things

Our familiar rules of inference are not strong enough to prove abstract mathematical statements. Typically we want our proof to apply to a whole *set* of things (numbers). Now that we know about *predicate logic*, we can apply our

more powerful *quantified* rules of inference to prove real mathematical statements.

Example: Using Universal Modus Ponens, verify the validity of the following proof:

> *Proof.* Let $m, n \in \mathbb{Z}$, and let m be even. Then $m = 2p$ for some integer p.(1) Now,
>
> $$\begin{aligned} m \cdot n &= (2p)n && \text{by substitution} \\ &= 2(pn)^{(2)} && \text{by associativity} \end{aligned}$$
>
> Now, $pn \in \mathbb{Z}$,(3) so by definition of even $2(pn)$ is even.(4) Thus mn is even. □

Solution:

(1) If an integer is even, then it equals twice some integer.
m is a particular integer, and it is even.
$\therefore$ m equals twice some integer p.

(2) If a quantity is an integer, then it is a real number.
p and n are both particular integers.
$\therefore$ p and n are both real numbers.

For all a, b, c, if $a, b, c \in \mathbb{R}$ then $(ab)c = a(bc)$.
2, p, and n are all particular real numbers.
$\therefore$ $(2p)n = 2(pn)$.

(3) For all u, v, if $u, v \in \mathbb{Z}$ then $uv \in \mathbb{Z}$.
p and n are both particular integers.
$\therefore$ $pn \in \mathbb{Z}$.

(4) If a number equals twice some integer, then that number is even.
$2(pn)$ equals twice the integer pn.
$\therefore$ $2(pn)$ is even.

Of course, we would never do a mathematical proof like this. In reality, you do this in your head automatically. Seeing this form, however, allows you to easily verify the **validity** of the proof.

1.4 Summary

- Propositional logic contains the entirety of Boolean algebra and logic connectives, with True and False as the only inputs/outputs
- Predicate logic contains the entirety of propositional logic and uses functions along with entities
- Deriving knowledge from familiar rules entails mathematical proof

1.5 Practice

1. Answer the two logic puzzles presented in the introduction of this chapter.
2. Translate the following statement into propositional logic: *turn right then turn left.*
3. Translate the following statement into propositional logic: *if it is raining then everyone has an umbrella.*
4. How can you quickly construct a truth table with all row possibilities? Use your technique to construct a truth table with 4 variables.
5. How many rows does a truth table with n variables have?
6. Prove theorem 1.2.12.
7. Prove theorem 1.2.13.
8. Draw the circuit representation of theorem 1.2.12.
9. Prove the following rule valid or invalid:

$$\begin{array}{l} (a \wedge d) \Rightarrow b \\ e \\ b \Rightarrow (\neg e) \\ (\neg a) \Rightarrow f \\ (\neg d) \Rightarrow f \\ \hline \therefore f \end{array}$$

10. Prove the following rule valid or invalid:

$$\begin{array}{l} (a \wedge d) \Rightarrow b \\ e \\ b \Rightarrow (\neg e) \\ (\neg a) \Rightarrow f \\ (\neg d) \Rightarrow f \\ \hline \therefore b \Rightarrow e \end{array}$$

11. Push the negation inside the following statement as far as possible:

$$\neg(\forall x \in \mathbb{R})(\exists m \in \mathbb{Z})[(0 \leq x - m < 1) \Leftrightarrow (m = \lfloor x \rfloor)]$$

Chapter 2

Set Theory

Sets are wild fam.

Justin Goodman

Contents

2.1 Introduction

Sets were introduced by Georg Cantor in the late 1800s. Cantor is the grandfather of set theory and continuity. We will see continuity topics later, including Cantor's famous diagonal argument. We can define all of discrete mathematics using sets. What is a set though?

Definition 2.1.1 Set
An unordered collection of unique objects. We denote sets using curly braces $\{\}$ with objects appearing in them – e.g. $\{\circ, 3, \pi, \blacktriangle\}$

Each part of this definition is important – we do not have a set unless it satisfies the entire definition. We now examine the definition.

- A set is a **collection** of stuff. Think of a set like a box. You can put things in your box, and you can take them out. Your box is special – it can expand/contract to fit anything you like.
- A set is **unordered**. This simply means that any different ordering we give to a set does not change the equality property of the set – $\{1, 2\} = \{2, 1\}$
- A set contains **objects**. A set can contain anything you want.
- A set contains **unique** objects. For any two distinct objects in a set, the objects cannot be equal. Sometimes we see books describe the sets $\{1, 1, 2, 3\}$ and $\{1, 2, 3\}$ as equal, however we argue that the first set is not even a set![1] We recommend you ask your instructor about this distinction, and follow what they prefer.

This chapter includes an overview of set theory – sets, operations, binary relations, and theorems.

2.2 Building Sets

Before we dive into set theory concepts, we first introduce a few ways to denote sets.

To start, you can simply denote a set by just writing each element inside some curly braces. For example,

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

describes the set containing 1, 2, and 3. This method is not useful when describing big sets, though. If your set has 2^{64} elements, you would never be able to write them all out!

To solve this, we can use *ellipses* – $\cdots$ – three dots in a row. Ellipses inside a set simply mean that you take the implicit pattern described in the set already, and continue it (possibly indefinitely). For example,

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

Here, the pattern is described in the $1, 2, 3$ part – increase by 1. Now,

So far, the presented notations lack power. To combat this, we introduce set-builder notation. As the name implies, this is a way of building sets.

Definition 2.2.1 Set-builder Notation
A way of defining sets. Syntax: $\{$ element $\mid$ condition(s) $\}$, read as, "element *such that* condition(s) is (are) satisfied." For example, the set of even integers (which we will learn about soon) can be written as: $\{e \mid (\exists k \in \mathbb{Z})[e = 2k]\}$

[1] We would classify the set $\{1, 1, 2, 3\}$ as a *multi-set*

Remark 2.2.2
You may also see the : character instead of the | character in set-builder notation. These are equivalent. $\{e \mid (\exists k \in \mathbb{Z})[e = 2k]\} = \{e : (\exists k \in \mathbb{Z})[e = 2k]\}$.

It is pivotal that you know how to read and create sets this way. The syntax is very flexible because the conditions can be almost anything you like. The conditions should, however, relate to the set itself (otherwise the set would be trivially pointless).

Example: Build a set that contains all square roots of even integers (for now, use the definition of even integers given previously).

Solution:

$$S = \{x \mid (\exists k \in \mathbb{Z})[x^2 = 2k]\}$$

Example: Build a set that contains everything except for the object $\star$.

Solution:

$$S = \{y : y \neq \star\}$$

With set-builder notation, we can describe the following set short-cut:

$$[n] = \{i \in \mathbb{Z} \mid 0 \leq i \leq n\}$$

. We do not know what $\mathbb{Z}$ is just yet, but we will get there soon enough.

Remark 2.2.3
Some authors denote $[3] = \{1, 2, 3\}$ since they do not include 0 as part of the natural numbers. More on this later. For now, just follow whatever your professor is doing.

Remark 2.2.4
Some authors also reserve $[x]$ to mean the floor function – the greatest integer smaller than x. As with anything in math, context is key. We will not use brackets to indicate the floor function, but you should be aware that different notation conventions exist.

Finally, we can describe continuous-interval sets. You may be familiar with the real number line – this is a continuous line because there are no "breaks" between any two numbers you pull from it. Given any two numbers on the real

number line, we can always take the midpoint to get another real number! How do we denote these intervals? We use interval notation:

- $(x, y) = \{r \in \mathbb{R} : x < r < y\}$
- $(x, y] = \{r \in \mathbb{R} : x < r \leq y\}$
- $[x, y) = \{r \in \mathbb{R} : x \leq r < y\}$
- $[x, y] = \{r \in \mathbb{R} : x \leq r \leq y\}$

A parenthesis means we *exclude* the associated number from the continuous interval, and a bracket means we *include* the associated number.

2.3 Definitions

We include a handful of definitions and notations we use in our study of set theory.

Binary relations:

Definition 2.3.1 Member/Element
An object that is part of a set. Symbol: $\in$. Example: $5 \in \{1, 2, 3, 4, 5\}$

Definition 2.3.2 Subset
A set of elements that are all members of another set. A subset CAN be equal to its parent set. Symbol: $\subseteq$. Example: for sets S and T, $S \subseteq T \Leftrightarrow (\forall s \in S)[s \in T]$

Definition 2.3.3 Proper Subset
A set of elements that are all members of another set, but the subset is NOT equal to the parent set. Symbol: $\subset$. Example: for sets S and T, $S \subset T \Leftrightarrow (\forall s \in S)[(s \in T) \wedge (S \neq T)]$

Definition 2.3.4 Superset
A flipped version of subset. Symbol: $\supseteq$. Example: for sets S and T, $S \supseteq T \Leftrightarrow (\forall t \in T)[t \in S]$

Definition 2.3.5 Proper Superset
A flipped version of proper subset. Symbol: $\supset$. Example: for sets S and T, $S \supset T \Leftrightarrow (\forall t \in T)[(t \in S) \wedge (S \neq T)]$

Definition 2.3.6 Equality
Two sets are equal if and only if both sets are subsets of each other. Notation: for sets S and T, $S = T \Leftrightarrow (S \subseteq T) \wedge (T \subseteq S)$. To prove this, you can use set-builder notation & theorems (given later in this chapter) to show equivalence, or you can prove $(e \in S \Rightarrow e \in T) \wedge (e \in T \Rightarrow e \in S)$

Things:

Definition 2.3.7 Cardinality
The number of elements in a set. Notation: for a set S, cardinality is denoted by $|S|$. For example, $|\{2, 3, 4, 5, 6\}| = 5$

Definition 2.3.8 Empty/Null Set
The set containing zero elements. Notation: $\emptyset$ or $\{\}$ – these symbols are *interchangable*. **Note**: $|\emptyset| = 0$

Definition 2.3.9 Universal Set
The set of all possible sets. Notation: U is the Universal Set

Definition 2.3.10 Power Set
The set of all possible subsets of a set. Notation: of a set S, the power set is denoted $\mathcal{P}(S)$. For example, $\mathcal{P}(\{1, 2, 3\}) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. **Note**: for any set S, $|\mathcal{P}(S)| = 2^{|S|}$

Definition 2.3.11 Disjoint
Two sets are disjoint if and only if both sets have no members in common. For two sets S and T, this is equivalent to $S \cap T = \emptyset$

Definition 2.3.12 Partition
(of a set) A set of sets T where the union of every element in T equals the original set, and all elements in T are disjoint. For example, $\{\{1, 2\}, \{3\}, \{4, 5, 6\}\}$ is a partition of $\{1, 2, 3, 4, 5, 6\}$

Operations:

Definition 2.3.13 Union
(of two sets) A set that includes all elements from both sets (discounting duplicates, since a set must contain unique elements). Notation: for sets S and T, $S \cup T = \{e \mid (e \in S) \vee (e \in T)\}$. For example, $\{1, 2, 3\} \cup \{3, 4, 5\} = \{1, 2, 3, 4, 5\}$

Definition 2.3.14 Intersection
(of two sets) A set that includes only elements from both sets. Notation: for sets S and T, $S \cap T = \{e \mid (e \in S) \wedge (e \in T)\}$. For example, $\{1, 2, 3\} \cap \{3, 4, 5\} = \{3\}$

Definition 2.3.15 Subtraction
(of two sets) A set representing the elements of the second set taken out of the first set. **Note**: subtraction order *matters* (i.e. subtraction is not commutative). Notation: for sets S and T, $S - T = \{e \mid (e \in S) \wedge (e \notin T)\}$. You may also see set subtraction represented as $S \setminus T$. For example, $\{1, 2, 3\} \setminus \{3, 4, 5\} = \{1, 2\}$

Definition 2.3.16 Compliment
(of a set) A set containing every element from the universal set that is NOT

contained in the original set. **Note**: you can take the compliment with respect to different universes so long as you specify which one – by default the universal set is assumed. Notation: for a set S, the complement is noted as $S^{\complement}$, or $S^{'}$, or $\overline{S}$; with the universe U, $S^{'} = \{e \mid e \in (U - S)\}$

Definition 2.3.17 Cross Product
(of two sets) The set of all ordered pairings of two sets. Notation: for sets S and T, $S \times T = \{(s,t) \mid s \in S \wedge t \in T\}$.

Note: you can take the cross product of multiple sets. For sets $A_1 \cdots A_n$, the cross product $A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \cdots, a_n) \mid a_i \in A_i\}$. When $A_1 = \cdots = A_n = A$ then we let $A^n = A_1 \times A_2 \times \cdots \times A_n$

2.4 Theorems

We will not go into depth on the axioms of Zermelo–Fraenkel set theory. We do include a handful of nice theorems that will aid in proving statements about sets. Sometimes these are referred to as axioms, however we argue that the following statements can be derived from ZF set theory axioms and should hence be called theorems. It does not really matter though.

You are not required to memorize these theorems – they will be given to you as a table.

Theorem 2.4.1 Commutativity
For any sets A and B the union and intersection operations are commutative:

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

Theorem 2.4.2 Associativity
For any sets A, B, and C the union and intersection operations are associative:

$$(A \cup B) \cup C = A \cup (B \cup C)$$

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

Theorem 2.4.3 Distributivity
For any sets A, B, and C the union and intersection operations are distributive:

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

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

Theorem 2.4.4 Identity
For any set A and universe U the following hold:

$$A \cup \emptyset = A$$

$$A \cap U = A$$

Theorem 2.4.5 Inverse
For any set A and universe U the following hold:

$$A \cup A^{\mathsf{c}} = U$$

$$A \cap A^{\mathsf{c}} = \emptyset$$

Theorem 2.4.6 Double Compliment
For any set A the following holds:

$$(A^{\mathsf{c}})^{\mathsf{c}} = A$$

Theorem 2.4.7 Idempotence
For any set A the following hold:

$$A \cup A = A$$

$$A \cap A = A$$

Theorem 2.4.8 De Morgan's
For any sets A and B the following hold:

$$(A \cup B)^{\mathsf{c}} = A^{\mathsf{c}} \cap B^{\mathsf{c}}$$

$$(A \cap B)^{\mathsf{c}} = A^{\mathsf{c}} \cup B^{\mathsf{c}}$$

Theorem 2.4.9 Universal Bound (Domination)
For any set A and universe U the following hold:

$$A \cup U = U$$

$$A \cap \emptyset = \emptyset$$

Theorem 2.4.10 Absorption
For any sets A and B the following hold:

$$A \cup (A \cap B) = A$$

$$A \cap (A \cup B) = A$$

Theorem 2.4.11 Absolute Compliment
For a given universe U the following hold:

$$\emptyset^{\mathsf{c}} = U$$

$$U^{\mathsf{c}} = \emptyset$$

Theorem 2.4.12 Set Subtraction Equality
For any sets A and B the set subtraction operation satisfies the following:

$$A - B = A \cap B^{\mathsf{c}}$$

This establishes a relationship between the relative and absolute compliment

The aforementioned theorems are helpful for simplifying complicated sets.

Example: Simplify the following expression:

$$((A \cup B) \cap C) \cup ((A^{\mathsf{c}} \cap B^{\mathsf{c}}) \cup D^{\mathsf{c}})^{\mathsf{c}}$$

Solution: Lots of compliments is a good indication for using De Morgan's.

$$\begin{aligned}
&((A \cup B) \cap C) \cup ((A^{\mathsf{c}} \cap B^{\mathsf{c}}) \cup D^{\mathsf{c}})^{\mathsf{c}} && \\
&= ((A \cup B) \cap C) \cup ((A \cup B)^{\mathsf{c}} \cup D^{\mathsf{c}})^{\mathsf{c}} && \text{De Morgan's} \\
&= ((A \cup B) \cap C) \cup ((A \cup B) \cap D) && \text{De Morgan's} \\
&= (A \cup B) \cap (C \cup D) && \text{Distributivity}
\end{aligned}$$

Example: Show the following equivalence:

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

Solution: Sometimes just trying random things works out in your favor.

$$\begin{aligned}
A \cup (B \cup (A \cap C)) &= A \cup ((A \cap C) \cup B) && \text{Commutativity} \\
&= (A \cup (A \cap C)) \cup B && \text{Associativity} \\
&= A \cup B && \text{Absorption}
\end{aligned}$$

Remark 2.4.1
There is a stark similarity between set and Boolean simplification.

2.5 Important Sets

We introduce some notation for a handful of important sets.

Definition 2.5.1 The Natural Numbers – $\mathbb{N}$
The standard discrete numbers with which you count. Some math classes start the naturals at 1, however in computer science we start the naturals at 0. Just remember simply that arrays utilize 0-indexing, so we do the same. The set looks like so: $\{0, 1, 2, 3, 4, 5, \cdots\}$

In mathematics, we define the natural numbers *inductively* – we will discuss induction in a few chapters. We introduce the inductive definition here. You do not need to know this, however we think it is interesting.

Proposition 2.5.1 Inductive Definition of $\mathbb{N}$
Define a set $S \subseteq \mathbb{R}$ to be inductive if and only if the following conditions hold:

- $0 \in S$
- *if* $x \in S$ *then* $x + 1 \in S$

Define $\mathbb{N}$ as the intersection of all inductive sets.

Understandably you may be confused on the notation of $\mathbb{R}$ – we will come back to this in a moment. For now, we continue to the integers.

Definition 2.5.2 The Integers – $\mathbb{Z}$
The standard *signed* discrete numbers with which you count. The integers include all of the natural numbers as well as all of their negatives[2]. The set looks like so: $\{\cdots, -3, -2, -1, 0, 1, 2, 3, \cdots\}$

The next set, which you may be familiar with, is the rationals.

Definition 2.5.3 The Rationals – $\mathbb{Q}$
The set of numbers that can be written as a ratio ($\mathbb{Q}$uotient) of integers. $\mathbb{Q} = \{x = \frac{a}{b} \mid a \in \mathbb{Z} \wedge b \in \mathbb{Z}^{\neq 0}\}$

Naturally we can define somewhat of an 'opposite' to the rationals.

Definition 2.5.4 The Irrationals – $\mathbb{R} \setminus \mathbb{Q}$
The set of real numbers that do not satisfy the rational property. A provable example is that $\sqrt{2} \in \mathbb{R} \setminus \mathbb{Q}$

The real numbers are not the main focus in discrete mathematics, however we still provide a definition.

[2] 0 is neither positive nor negative, so we cannot take its negation, however we let $0 \in \mathbb{Z}$

Definition 2.5.5 The Reals – $\mathbb{R}$
The continuous interval $(-\infty, \infty)$. Any number that does not have the form $a + bi$, where $i = \sqrt{-1}$

In an analytical mathematics course, the reals and irrationals are more strongly defined. The above definitions are enough for this course.

2.6 Sets to Logic

We can use set-builder notation along with our familiar logic rules to prove things about sets. This proof technique can be used to prove the theorems we showed earlier.

Example: Prove theorem 2.4.12:

$$A - B = A \cap B^{\mathsf{c}}$$

Proof.

$$\begin{aligned} A - B &= \{x \mid x \in A \wedge x \notin B\} && \text{defn of subtraction} \\ &= \{x \mid x \in A \wedge x \in B^{\mathsf{c}}\} && \text{defn of compliment} \\ &= A \cap B^{\mathsf{c}} && \text{defn of intersection} \end{aligned}$$

□

Example: Prove example 2.4

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

Proof.

$$\begin{aligned} & A \cup (B \cup (A \cap C)) \\ &= \{x \mid x \in A \vee (x \in B \vee (x \in A \wedge x \in C))\} && \text{defn of } \cup/\cap \\ &= \{x \mid x \in A \vee ((x \in A \wedge x \in C) \vee x \in B)\} && \text{Commutativity (logic)} \\ &= \{x \mid (x \in A \vee (x \in A \wedge x \in C)) \vee x \in B\} && \text{Associativity (logic)} \\ &= \{x \mid x \in A \vee x \in B\} && \text{Absorption (logic)} \\ &= A \cup B && \text{defn of } \cup \end{aligned}$$

□

2.7 Summary

- Sets are unordered collections of unique objects
- Many operations and theorems exist in set theory
- $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R} \setminus \mathbb{Q}, \mathbb{R}$ are all important sets

2.8 Practice

1. Provide an example of a valid set, and an invalid set.
2. Build a set of natural numbers that are all multiples of 3.
3. Explain why the cardinality of a finite set is a natural number.
4. For two sets A and B, is it necessarily the case that $|A \cup B| = |A| + |B|$?
5. Explain intuitively why $|\mathcal{P}(A)| = 2^{|A|}$.
6. Is $\{\mathbb{Z}^-, 0, \mathbb{Z}^+\}$ a valid partition of $\mathbb{Z}$? Explain. If it is not, change the partition to make it valid.
7. Is $\{\{1, 3, 5, 7, 9, \cdots\}, \{0, 2, 4, 6, 8, \cdots\}\}$ a valid partition of $\mathbb{N}$? Explain. If it is not, change the partition to make it valid.
8. Provide a set T which is a valid partition of $\mathbb{N}$ such that $|T| = 5$.
9. For each of the following, answer true or false:
   (a) $\mathbb{Z} \supset \mathbb{N}$
   (b) $\mathbb{N} \in \mathbb{Z}$
   (c) $0 \in \emptyset$
   (d) $\emptyset \in \emptyset$
   (e) $\emptyset \subseteq \emptyset$
   (f) $\emptyset \subset \emptyset$
   (g) $\mathcal{P}(\emptyset) = \emptyset$
10. Simplify the expression $A^\mathsf{c} \cup (B \cup A)^\mathsf{c}$.
11. Prove the two Absorption theorem statements are equivalent.

Chapter 3

Number Theory

Time to get number *freaky*.

Justin Goodman

This chapter is hefty.

Contents

3.1 Introduction

Definition 3.1.1 Number Theory
The mathematical study of integers and their properties

This chapter is intended to introduce you to the most important tool in number theory – proofs. We will further explain how to execute a proof, however we must begin with basic definitions. It is important to see what is involved in a proof, so we will show you some proofs while introducing the basic principles.

3.2 Basic Principles

3.2.1 Summations and Products

We first introduce a form of notation that helps us simplify and express long equations.

Definition 3.2.1 Summation
The sum over a sequence of numbers. Example:

$$\sum_{\substack{i=0 \\ i \text{ even}}}^{3} \frac{i+2}{i+1} = \frac{0+2}{0+1} + \frac{2+2}{2+1} + \frac{4+2}{4+1} + \frac{6+2}{6+1}$$

Sometimes there is a condition (placed under the sum), which means you only evaluate the summation when the condition is *satisfied*. Example:

$$\sum_{\substack{i=0 \\ i \text{ even}}}^{6} \frac{i+2}{i+1} = \frac{0+2}{0+1} + \frac{2+2}{2+1} + \frac{4+2}{4+1} + \frac{6+2}{6+1}$$

The empty sum sums to zero and is denoted, for $a > b$:

$$\sum_{i=a}^{b} a_i = 0$$

Definition 3.2.2 Product
The product over a sequence of numbers. Example:

Again, sometimes there is a condition (placed under the product), the same rules apply. Example:

$$\prod_{\substack{i=1 \\ i \text{ odd}}}^{8} i^2 = 1^2 \times 3^2 \times 5^2 \times 7^2$$

The empty product multiplies to 1 and is denoted, for $a > b$:

$$\prod_{i=a}^{b} a_i = 1$$

Remark 3.2.3
We can think of the indices in a sum or product as being elements of a set. Since the + operator is commutative, then the order that we evaluate the sum with respect to the indices *does not matter*. For example, the sum $\sum_{i=2}^{6} i$ iterates over the indices $i \in \{2, 3, 4, 5, 6\}$, and equals $2 + 3 + 4 + 5 + 6$ which can be re-ordered however we like (eg $4 + 5 + 2 + 3 + 6$). The condition placed on a sum or product simply becomes a restriction on the set. For example, the sum $\sum_{i\equiv 0 \pmod 3}^{6} i$ iterates over the indices $\{i \in \{2, 3, 4, 5, 6\} \mid i \equiv 0 \pmod 3\}$.

Abstractly, we can write the sum (and similarly, the product) as a sum over elements of a set. We write this as follows:

$$\sum_{\substack{i=a \\ P(i)}}^{b} a_i = \sum_{i \in I} a_i$$

where

$$I = \{i \in \mathbb{Z} \mid (a \leq i \leq b) \wedge P(i)\}$$

and $P(i)$ is the condition placed on the sum (or product).

Interestingly, the condition placed on the summation (or product) can depend on the actual *values* in the sum (or product). We could represent this as $P(i) \equiv Q(a_i)$.

Another interesting point, if we ever have $a > b$ then the condition $a \leq i \leq b$ is never satisfied, which means $I = \emptyset$. We also have that if $P(i)$ is false for all values of i such that $a \leq i \leq b$, then again $I = \emptyset$. If this is the case, then we sum (or product) over no indices. This is where the term **empty sum (or product)** comes from.

The empty sum is equal to the additive identity 0, and the empty product is equal to the multiplicative identity 1 (as we have seen before).

Example: Re-write the following sum as a summation, then evaluate it for $n = 2$:

$$\frac{1}{1} + \frac{2}{2} + \frac{3}{4} + \frac{4}{8} + \frac{5}{16} + \cdots + \frac{n+1}{2^n}$$

Solution: The sum equals

$$\sum_{i=0}^{n} \frac{i+1}{2^i}$$

and for $n = 2$ evaluates to

$$\frac{1}{1} + \frac{2}{2} + \frac{3}{4} = 2.75$$

Definition 3.2.4 Factorial
The factorial is the product, denoted $n!$ for $n \in \mathbb{Z}^+$,

$$n! = \prod_{i=1}^{n} i$$

3.2.2 Sequences and Series

We discuss sequences and series in a later chapter, however for now we introduce the basic ideas.

Definition 3.2.5 Sequence
An ordered list of numbers, each one associated with a specific *position* (index)

Example: The following is a sequence:

$$1, 2, 3, 4, 5, 6, \cdots$$

We can also write this surrounded by curly braces:

$$\{1, 2, 3, 4, 5, 6, \cdots\}$$

We can also give this sequence as a function, which takes indices as the input:

$$f(i) = i$$

We can also denote the sequence with a name:

$$a_i = i$$

Definition 3.2.6 Series
The sum of all terms in a sequence. Typically a series is infinite, however it can be finite.

Example: Let the sequence $a_i = 2i$. Then we denote the infinite series:

$$\sum_{i=1}^{\infty} a_i$$

Methods from calculus can help us evaluate and understand infinite series, however that is out of scope of this course.
If we give an upper bound, like $n = 4$, then we can denote and evaluate finite series (which just becomes a summation):

$$\sum_{i=1}^{4} a_i = 2(1) + 2(2) + 2(3) + 2(4) = 20$$

3.2.3 Logarithms

Definition 3.2.7 Exponentiation
The act of raising a fixed number n to a *power* x. You may be familiar with exponentiation with $n, x \in \mathbb{Q}$ (for example, $0.25^{0.5} = \frac{1}{\sqrt{4}} = 0.5$), but in-fact we can define exponentiation on real inputs.

Denote $\exp x = \lim_{n \Rightarrow \infty}(1 + \frac{x}{n})^n = e^x$ with $e^0 = 1$ and $e^1 = e$. After defining logarithms below, we see that $b = e^{\log b}$ so $b^x = (e^{\log b})^x = e^{x \log b}$ which fully defines exponentiation.

$e = \sum_{n=0}^{\infty} \frac{1}{n!}$, $e = \lim_{n \to \infty}(1 + \frac{1}{n})^n$, and e^x is the unique solution to the differential equation $f(x) = f'(x)$ with $f(0) = 1$.

Definition 3.2.8 Logarithm
The function $\log x$ is the inverse function of the exponential function e^x. The logarithm satisfies $\log 1 = 0$ and $\log e = 1$. You may know this as the *natural logarithm* $\ln x = \log_e x$. If no base is given, assume the base is Euler's Number e.

Denote the *base* b of a logarithm as

$$\log_b x = \frac{\log x}{\log b}$$

Proposition 3.2.1

$$\log_b x + \log_b y = \log_b(x \cdot y)$$

Proposition 3.2.2

$$a \log_b x = \log_b(x^a)$$

Proposition 3.2.3

$$\log_b x - \log_b y = \log_b(xy^{-1}) = \log_b(\frac{x}{y})$$

Proposition 3.2.4

$$a^{\log_b x} = x^{\log_b a}$$

3.2.4 Floors and Ceilings

Definition 3.2.9 Floor
The greatest integer less than or equal to the input real number. For $x \in \mathbb{R}$, we denote the floor of x as $\lfloor x \rfloor = m \in \mathbb{Z}$ where $m \leq x$ and $m + 1 > x$. So

$$\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$$

Equivalently,

$$0 \leq x - \lfloor x \rfloor < 1$$

which means we can write $x = \lfloor x \rfloor + \epsilon$ with some $\epsilon \in [0, 1)$.

Definition 3.2.10
The smallest integer greater than or equal to the input real number. For $x \in \mathbb{R}$, we denote the ceiling of x as $\lceil x \rceil = n \in \mathbb{Z}$ where $n \geq x$ and $n - 1 < x$. So

$$\lceil x \rceil - 1 < x \leq \lceil x \rceil$$

Equivalently,

$$-1 < x - \lceil x \rceil \leq 0$$

which means we can write $x = \lceil x \rceil + \epsilon$ with some $\epsilon \in (-1, 0]$. By negating ϵ, then $x = \lceil x \rceil - \epsilon$ with some $\epsilon \in [0, 1)$.

Example: Show that $\lfloor x + n \rfloor = \lfloor x \rfloor + n$ for all $x \in \mathbb{R}$ and $n \in \mathbb{Z}$.

Proof. We can use any definition of the floor to prove this. The easiest proof comes from the second form.
Denote $\lfloor x \rfloor = m \in \mathbb{Z}$. Then $\lfloor x \rfloor + n = m + n$ and $\lfloor x + n \rfloor = \lfloor m + \epsilon + n \rfloor = \lfloor m + n + \epsilon \rfloor = m + n$ since $m + n \in \mathbb{Z}$ by closure of integers under addition (this is discussed below). □

Remark 3.2.11
Similarly, $\lceil x + n \rceil = \lceil x \rceil + n$.

As a closing note, we should consider an interesting example in real analysis that uses floors to break intuition.

Example: $\lfloor 0.9 \rfloor = 0$, $\lfloor 0.99 \rfloor = 0$, $\lfloor 0.999 \rfloor = 0$, $\lfloor 0.999 \cdots 9 \rfloor = 0$. So what about $\lfloor 0.\bar{9} \rfloor$?

Solution: Intuitively, you might think that $0.\bar{9} < 1$ which would mean $\lfloor 0.\bar{9} \rfloor = 0$. This is incorrect, because by definition $0.\bar{9} = 1$. This comes from how we define the bar operator, which means an infinite repetition of whatever is under it.

$$\begin{aligned}
0.\bar{9} &= 0.9 + 0.09 + 0.009 + 0.0009 + \cdots \\
&= 9(0.1 + 0.01 + 0.001 + 0.0001 + \cdots) \\
&= 9(\frac{1}{10} + \frac{1}{100} + \frac{1}{1000} + \frac{1}{10000} + \cdots) \\
&= 9\sum_{k=1}^{\infty}(\frac{1}{10})^k \\
&= 9 \lim_{n\to\infty} \sum_{k=1}^{n}(\frac{1}{10})^k \\
&= 9 \lim_{n\to\infty} [\sum_{k=0}^{n}(\frac{1}{10})^k - 1] \\
&= 9 \lim_{n\to\infty} [\frac{1 - (\frac{1}{10})^{n+1}}{1 - \frac{1}{10}} - 1] \\
&= 9 \lim_{n\to\infty} [\frac{10}{9}(1 - (\frac{1}{10})^{n+1}) - 1] \\
&= 9 \lim_{n\to\infty} [\frac{10}{9} - \frac{10}{9}(\frac{1}{10})^{n+1} - 1] \\
&= 9[\frac{10}{9} - \frac{9}{9} - \frac{10}{9}\frac{1}{10} \lim_{n\to\infty} (\frac{1}{10})^n] \\
&= 1 - \lim_{n\to\infty} \frac{1}{10^n} = 1 - 0 = 1
\end{aligned}$$

Another way to see this,

$$0.\bar{9} = 9 \cdot 0.\bar{1} = 9 \cdot \frac{1}{9} = 1$$

Thus

$$\lfloor 0.\bar{9} \rfloor = \lfloor 1 \rfloor = 1$$

Remark 3.2.12
This example uses a definition from real analysis. This book is on discrete mathematics. You do not need to know the definition of $0.\bar{x}$

3.2.5 Closure

Closure has two parts: a **set**, and an **operation**. Closure is a *property* of a set paired with the given operation. The basic idea is that if you take **any** two elements (or however are required for the operation – typically two) and apply the given operation to those elements, then the result will be an element also in the set. Some examples will be helpful to illustrate the point here.

Definition 3.2.13 Closure
For a given set S and operation $\circ$, S is *closed under* $\circ$ if and only if

$$(\forall a, b \in S)[a \circ b \in S]$$

Note, the given operator could be unary, binary, trinary, etc, and the amount of variables pulled from the set correspond accordingly.

Theorem 3.2.5
The natural numbers $\mathbb{N}$ *are closed under addition. Formally,*

$$(\forall n, m \in \mathbb{N})[n + m \in \mathbb{N}]$$

> **Example:** Try and think of some natural numbers where if you add them you *do not* get a natural number. If you came up with an answer, it would be a *counterexample* to the theorem.
> You should not be able to come up with a counterexample.

Remark 3.2.14
For this book, we will accept the above theorem as a fact, since the proof relies on proposition 2.5.1 (the inductive definition of $\mathbb{N}$) which we are not requiring you to know. The proof is relatively straightforward though, so we include it in a few sections.

Okay, onto more examples of closure.

> **Example:** The natural numbers are closed under multiplication. Formally,
> $$(\forall n, m \in \mathbb{N})[n \cdot m \in \mathbb{N}]$$

Proof. Informal sketch: since m is a discrete counting number, we can think of $n \cdot m = \underbrace{n + \cdots + n}_{m \text{ times}}$. Since $\mathbb{N}$ is closed under addition, and multiplication here is just repeated addition, we are done. □

Example: The natural numbers are **not** closed under subtraction. Formally,

$$(\exists n, m \in \mathbb{N})[n - m \notin \mathbb{N}]$$

Proof. Take any natural numbers n, m such that $n < m$. Then $n < m \Rightarrow n - m < 0$ and we get a negative number, which is by definition not a natural number. □

Remark 3.2.15
This is somewhat rigorous. All we need here is a counterexample to $(\forall n, m \in \mathbb{N})[n - m \in \mathbb{N}]$. So, just take any numbers that disprove the statement: $5 - 6 = -1 \notin \mathbb{N}$, done!

Theorem 3.2.6

$$(\forall a, b \in \mathbb{Z})[a + b \in \mathbb{Z}]$$

Remark 3.2.16
For the scope of this book, we can take it by definition that integers are closed under addition (the previous theorem).

Example: Prove or disprove

$$(\forall a, b \in \mathbb{Z})[a \div b \in \mathbb{Z}]$$

Proof. Consider $1, 4 \in \mathbb{Z}$ but $1 \div 4 = 0.25 \notin \mathbb{Z}$, thus we have dis-proven the statement. □

Example: Let $S = \{1, 2, 3, 4\}$. Show that S is not closed under addition.

Proof. Well, $3, 4 \in S$ but $3 + 4 = 7 \notin S$. Another counterexample: $4 + 4 = 8 \notin S$ (this is just to illustrate that you can pull **the same element** from the set since closure applies to **all** elements). □

Closure can be of **any** set and **any** operation. We can get fancy with this, as with the previous example and the next example.

Example: Let $S = \{a, b, c, d\}$. Define the following table as the outcome of performing a ♦ operation:

♦	a	b	c	d
a	d	c	b	a
b	b	d	a	c
c	c	a	d	b
d	a	b	c	d

For example, $a♦d = a$ (the first operand comes from the row, and the second operand comes from the column).
Is S closed under ♦? Justify.

Proof. S is closed under ♦ since every possible outcome (as listed in the table) of the ♦ operation results in an element in S. □

Finally, we present an example of a unary operator.

Example: Show that the integers are closed under taking the negative (multiplying by -1),

$$(\forall x \in \mathbb{Z})[-x \in \mathbb{Z}]$$

Proof. Consider that $-1 \in \mathbb{Z}$ and integers are closed under multiplication. □

3.2.6 Parity

We introduce the idea with the definition.

Definition 3.2.17 Parity
An inherit property of the integers which says that *every* integer is *exclusively*

either **even** or **odd** (note the *exclusive or* – an integer must be even or odd, however it cannot be both and it cannot be neither)

This is an important property and will be useful for later proofs. Imagine you want to show some statement is true for *all* integers. Sometimes it is easier to break the statement into two cases – odd integers and even integers – and prove both cases separately. If both cases turn out to be true, then you can use parity to conclude the original statement is true.

We will show this idea in later proofs, however we should first explain what is even and odd.

Definition 3.2.18 Even
An integer n is even if and only if $(\exists k \in \mathbb{Z})[n = 2k]$

Definition 3.2.19 Odd
An integer m is odd if and only if $(\exists l \in \mathbb{Z})[m = 2l + 1]$

By *parity*, we can reformat the definitions of even and odd to depend on each other. The following statement is true:

An integer is even if and only if it is **not** odd

Equivalently:

An integer is odd if and only if it is **not** even

These statements both require that you know one of the two definitions. If this was not true, then the statements would be circular and would be useless.

We will come back to these statements soon.

3.2.7 Divisibility

Previously we showed that the integers are **not** closed under division. However, some integers when divided output an integer, like 3/1 and 4/2. It thus might make sense for us to study division.

Definition 3.2.20 Divisibility
The property that a number can be evenly divided (with no remainder). a is divisible by b if and only if $a \div b \in \mathbb{Z}$. Note that $a \div b = \frac{a}{b}$.

We say that b **divides** a if and only if a **is divisible by** b, and we note this as $b \mid a$. We will always use this notation.

$$b \mid a \Leftrightarrow \frac{a}{b} \in \mathbb{Z} \Leftrightarrow (\exists x \in \mathbb{Z})[a = bx]$$

Remark 3.2.21
We like to think of $b \mid a \Leftrightarrow \frac{a}{b}$ as a rotation counterclockwise:

$$\overset{\curvearrowleft}{b \mid a} \mapsto \frac{a}{b}$$

Definition 3.2.22 Remainder
An alternate way to represent (integer) division. When $b \nmid a$ then $\exists r \in \mathbb{Z}$ where $0 < r < a$ is the *remainder* and $b \mid (a - r)$. Sometimes we write the division as qRr – this is just the integer part of the division $q = (a - r)/b$ appended with R appended with the remainder

Theorem 3.2.7
For any $a, b \in \mathbb{Z}^+$, if $b \mid a$ then $b \leq a$

Proof. $b \mid a$ so by definition $(\exists k \in \mathbb{Z})[\frac{a}{b} = k]$. Since $a, b > 0$ we conclude that $k > 0$. Since $\frac{a}{b} = k$ then $a = bk$. Since $b, k > 0$ we conclude that $bk \geq b$ and thus $a = bk \geq b \Rightarrow b \leq a$ □

These definitions are important, however soon we will see that they fit into a wider scope of number theory. First, some examples.

Example: Give all natural numbers that divide 6.

Solution: We solve this by checking numbers. We do not want to check **all** numbers, though. Fortunately, we only have to check natural numbers, and the previous theorem tells us we only have to check numbers ≤ 6. Thus, we only need to check $0, 1, 2, 3, 4, 5, 6$.

- We cannot divide by 0
- 1 divides everything
- 6 is even so $2 \mid 6$
- $\frac{6}{3} = 2 \in \mathbb{N}$
- $\frac{6}{4} = 1.5 \notin \mathbb{N}$
- $\frac{6}{5} = 1.2 \notin \mathbb{N}$
- Any integer divides itself, so $6 \mid 6$

Thus $1, 2, 3, 6$ divide 6.

Example: Find the remainder of 10 divided by 4.

Solution: We can repeatedly take away 4 from 11 until we're left with r such that $0 < r < 4$.
$11 - 4 = 7$
$7 - 4 = 3$
We did this process 2 times, so $\frac{11}{4} = 2R3$. Also notice that $\frac{11-3}{4} = 8/4 = 2$

3.2.8 Modular Arithmetic

Modular arithmetic gives us a unified way to think about division between integers. Recall our definition of a remainder – this is essentially what a modulus is, but with an important difference. We show some notation first, then the definitions.

$$a \equiv b \pmod{m}$$

a, b, m are all integers here, with $m > 0$. Notice that we use the $\equiv$ symbol, which we have seen is used for *logical equivalence*. We will come back to this.

You may have seen modular arithmetic previously in a programming lens. In this case, the previous notation may be translated to the following.

$$a \ \% \ m = b$$

Still a, b, m are all integers but here we must have $b < m$. This is exactly taking the remainder.

Here is the subtle difference – in modular arithmetic, b **is not necessarily** less than m.

Definition 3.2.23 Congruence
Two integers a and b are congruent with respect to an integer m if and only if

$$\frac{b-a}{m}$$

is an integer. This may be re-written as

$$\big(\exists k \in \mathbb{Z}\big)\left[\frac{b-a}{m} = k\right]$$

Definition 3.2.24 Modulus
The divisor m in the congruence of two integers

Example: Find an integer congruent to 5 with respect to 3.

Solution: We can take the remainder of 5/3, which is 2. We will verify that $2 \equiv 5 \pmod 3$. Notice that

$$\frac{5-2}{3} = \frac{3}{3} = 1$$

which is an integer. Also notice that the order of a and b does not matter:

$$\frac{2-5}{3} = \frac{-3}{3} = -1$$

which is still an integer
The problem asked for *an* integer, so maybe other integers satisfy the congruence. We need to find an integer b such that $\frac{b-5}{3} \in \mathbb{Z}$. We choose some random integer, maybe 8, and set it equal to the previous fraction, and solve for b.

$$\frac{b-5}{3} = 8 \Leftrightarrow b - 5 = 24 \Leftrightarrow b = 29$$

so 29 is congruent to 5, which is also congruent to 2, mod 3

You may have noticed how two integers are *congruent*, not equal. We save *equality* for things that are actually equal. Clearly $29 \neq 2 \neq 5$ yet all of these integers are *congruent* mod 3. This motivates a further expansion on congruence.

Definition 3.2.25 Congruence Class
The set of all integers that are congruent to each other with respect to a given modulus m. For a given integer a, the congruence class of a modulo m is the set of all integer solutions to $\frac{b-a}{m} \in \mathbb{Z}$

Example: Write the set of all integers congruent to 3 modulo 5.

Solution: We need all integer solutions to $\frac{b-3}{5}$, or equivalently $b = 5k+3$ for any integer k. This is the set

$$\{b \in \mathbb{Z} \mid (\exists k \in \mathbb{Z})[b = 5k + 3]\}$$

which, if we plug in values for k, equals

$$\{\cdots, -12, -7, -2, 3, 8, 13, \cdots\}$$

Remark 3.2.26
The congruence class of a modulo m is the set of all integers b such that $a \equiv b \pmod{m}$.

Now that you understand what *is* modular arithmetic, we now discuss the difference between *equivalence* and *congruence*. In logic, as you have seen before, an *equivalence* between two statements means that the two statements output the same truth values for the same inputs (the statements are *logically equivalent*). In modular arithmetic, however, a *congruence* between two integers means they fall in the same *congruence class* modulo some number. We overload the $\equiv$ symbol with these two meanings. Think of the $\equiv$ operator as a function that takes two arguments. In the *equivalence* definition the arguments are logical statements, however in the *congruence* definition the arguments are integers. Logical statements and integers will never overlap, so you should never be confused by our usage of the operator.

Recall that we can use integer parity to relate our definitions of even and odd. We will see now *why* it works.

Theorem 3.2.8
If two numbers are in different modulo classes, then they cannot be the same number. Formally,

$$(\forall a, b \in \mathbb{Z})(\forall m \in \mathbb{Z}^+)[a \not\equiv b \pmod{m} \Rightarrow a \neq b]$$

Proof. This is directly the contrapositive of an intuitive fact: $a = b \Rightarrow a \equiv b \pmod{m}$. Consider that $a \equiv a \pmod{m}$ and since $a = b$ then $a \equiv b \pmod{m}$ □

Example: Show that an even number cannot be an odd number.

Proof. Take an even number $2k$ and an odd number $2l+1$ for some $k, l \in \mathbb{Z}$. Now mod-2 those numbers: $2k \equiv 0 \pmod{2}$ and $2l + 1 \equiv 1 \pmod{2}$. Clearly $0 \neq 1$ and hence by the previous theorem $2k \neq 2l + 1$. Thus an even number cannot be an odd number □

Finally, three important theorems.

Theorem 3.2.9
For all integers a, b, c, d, m with $m > 0$, if

$$a \equiv b \pmod{m}$$

and

$$c \equiv d \pmod{m}$$

then

$$a + c \equiv b + d \pmod{m}$$

Proof. $a \equiv b \pmod{m} \Leftrightarrow (\exists k \in \mathbb{Z})[a = b + mk]$, $c \equiv d \pmod{m} \Leftrightarrow (\exists l \in \mathbb{Z})[c = d + ml]$. Then, we can add these two equations to get $a + c = b + mk + d + ml = b + d + m(k + l)$. By closure of integers under addition, $k + l \in \mathbb{Z}$, therefore $a + c \equiv b + d \pmod{m}$ □

Theorem 3.2.10
For all integers a, b, c, d, m with $m > 0$, if

$$a \equiv b \pmod{m}$$

and

$$c \equiv d \pmod{m}$$

then

$$ac \equiv bd \pmod{m}$$

Proof. $a \equiv b \pmod{m} \Leftrightarrow (\exists k \in \mathbb{Z})[a = b + mk]$, $c \equiv d \pmod{m} \Leftrightarrow (\exists l \in \mathbb{Z})[c = d + ml]$. Then, we can multiply these two equations to get $ac = (b+mk)(d+ml) = bd+bml+dmk+m^2lk = bd+m(bl+dk+mlk)$. By closure of integers under addition and multiplication, $bl + dk + mlk \in \mathbb{Z}$, therefore $ac \equiv bd \pmod{m}$ □

Theorem 3.2.11
For all integers a, b, m, n with $m > 0$ and $n \geq 0$, if

$$a \equiv b \pmod{m}$$

then

$$a^n \equiv b^n \pmod{m}$$

Proof. For $n = 0$ we have $a^0 = 1$ and $b^0 = 1$ and clearly $1 \equiv 1 \pmod{m}$. For $n = 1$ we have as given that $a \equiv b \pmod{m}$. Finally $n > 1$ follows by repeatedly taking the previous theorem letting $c = a$ and $d = b$. □

Remark 3.2.27
The above theorem is a one-way implication. We cannot go the reverse direction. For example, $2^4 = 16 \equiv 1 = 1^4 \pmod{3}$ yet $2 \not\equiv 1 \pmod{3}$

Remark 3.2.28
In all of the above theorems, the modulo m is the same. That is, we cannot do the following: $8 \equiv 2 \pmod 6$ and $7 \equiv 1 \pmod 2$ therefore $8+7 \equiv 2+1 \pmod{???}$

3.2.9 Number Bases

Our familiar decimal system is known as *base-10*, since we have 10 possible digits spanning 0 to 9. We can represent base-10 numbers as an addition of numbers multiplied by the associated decimal position (the ones place, tens, hundreds, thousands, etc...).

Example: $123 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$

There are three other bases important to computer science: base-2, base-8, and base-16.

Definition 3.2.29 Binary (base-2)
Representing numbers with two digits, 0 and 1

Definition 3.2.30 Octal (base-8)
Representing numbers with digits 0 to 7

Definition 3.2.31 Hexadecimal (base-16)
Representing numbers with 16 digits. After the 9th digit, we move to letters A through F

We implicitly denote any number as base-10. If we want to denote a number x as base-b, then we write x_b. For example, the base-10 number 24_{10} equals the base-2 number 11000_2.

Techniques for Translation

We generally work in base-10, so we will start with going from base-10 to any arbitrary base. For an arbitrary number x in base-10, translating it into a number in base b entails taking the least positive integer of x modulo b, integer dividing with respect to the base (by taking $\lfloor x/b \rfloor$, and repeating until the number becomes zero. This will yield a sequence of integers, which make up the digits of the number in base-b from least to most significant (right to left).

Example: Find the base-5 representation of the base-10 number 12345.

Solution: Follow the given algorithm. Step-wise, we get:
1. $12345 \equiv 0 \pmod 5$, and $\lfloor 12345/5 \rfloor = 2469$

2. $2469 \equiv 4 \pmod 5$, and $\lfloor 2469/5 \rfloor = 493$
3. $493 \equiv 3 \pmod 5$, and $\lfloor 493/5 \rfloor = 98$
4. $98 \equiv 3 \pmod 5$, and $\lfloor 98/5 \rfloor = 19$
5. $19 \equiv 4 \pmod 5$, and $\lfloor 19/5 \rfloor = 3$
6. $3 \equiv 3 \pmod 5$, and $\lfloor 3/5 \rfloor = 0$

So $12345_{10} = 343340_5$

We care more about the special cases of b here – 2, 8, and 16. We start with base-2.

A binary number's digits represent powers of 2 in base-10. For example, $18_{10} = 10010_2 = 16_{10} + 2_{10} = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$

$\cdots$	16	8	4	2	1
$\cdots$	1	0	0	1	0

Translating base-10 to base-2: Associate each binary digit with a power of 2 – $\{1, 2, 4, 8, 16, 32, \cdots\}$ – then repeatedly, until the input number becomes zero, subtract the highest power of 2 lower than the input number. The powers of 2 subtracted get a 1 for their digit, and the others get a 0.

Example: Translate 43 to binary.

Solution:

1. The highest power of 2 lower than 43 is 32, so the 32 digit gets a 1. $43 - 32 = 11$
2. The highest power of 2 lower than 11 is 8, so the 8 digit gets a 1. $11 - 8 = 3$
3. The highest power of 2 lower than 3 is 2, so the 2 digit gets a 1. $3 - 2 = 1$
4. The highest power of 2 lower than 1 is 1, so the 1 digit gets a 1. $1 - 1 = 0$

So $43_{10} = 101011_2$

Translating base-10 to base-16: First, translate the number to binary. Append zeros to the left-side of the binary number until the number of digits is a multiple of 4. Next, partition the number into groups of

four (4) digits (bits). Finally, translate each 4-digit binary number group into a hexadecimal digit 0-15 (0-F).

Example: Translate 43 to hexadecimal.

Solution: From before, $43_{10} = 101011_2$. Left-fill with zeros: $= 00101011_2$. Partition into groups of 4: $= 0010 \mid 1011$. Translate each group into a hex number: $0010_2 = 2_{10} = 2_{16}$ and $1011_2 = 11_{10} = B_{16}$. So $43_{10} = 2B_{16}$

Translating base-10 to base-8: (*similar to the previous method*) First, translate the number to binary. Append zeros to the left-side of the binary number until the number of digits is a multiple of 3. Next, partition the number into groups of three (3) digits (bits). Finally, translate each 3-digit binary number group into an octal digit 0-7.

Example: Translate 43 to octal.

Solution: From before, $43_{10} = 101011_2$. Left-fill with zeros (none required): $= 101011_2$. Partition into groups of 3: $= 101 \mid 011$. Translate each group into an octal number: $101_2 = 5_{10} = 5_8$ and $011_2 = 3_{10} = 3_8$. So $43_{10} = 53_8$

Remark 3.2.32
We need four (4) binary digits to represent numbers 0-15, and three (3) binary digits to represent numbers 0-7.

Now, how do we go back from base-b to base-10? Well, associate each digit into an index position i, then multiply that digit by the base b raised to the power i.

Example: $2B_{16} = 2 \cdot 16^1 + B \cdot 16^0 = 2 \cdot 16 + 11 = 32 + 11 = 43$

Finally, how do we arbitrarily translate between bases? Well, usually the fastest way is to translate to base-10 or base-2, then translate to the other base you need.

Fast Mods

Taking mods with certain modulo can be made very easy, so long as you understand the previous modular arithmetic theorems and base-10 expansion.

Mod 1: Just return 1, since everything is divisible by 1.

Mod 2: Return 0 if the integer is even, and 1 if the integer is odd.

Mod 3: Notice that $10^k \equiv 1 \pmod 3$ for all $k \in \mathbb{N}$. So we expand an input number as the base-10 representation $a_n \cdot 10^n + \cdots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0$ where the a_i's make up the associated digits of the input number. Then we take this summation $\pmod 3$ and we get

$$a_n \cdot 10^n + \cdots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0 \equiv a_n + \cdots + a_2 + a_1 + a_0 \pmod 3$$

Thus, any integer is congruent to the sum of its digits $\pmod 3$.

Mod 5: Notice that $10^k \equiv 0 \pmod 5$ for all $k \in \mathbb{N}$ with $k > 0$. So, any digit past the one's place is equivalent to 0 $\pmod 5$,

$$a_n \cdot 10^n + \cdots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0 \equiv 0a_n + \cdots + 0a_2 + 0a_1 + 1a_0 \equiv a_0 \pmod 5$$

Thus, any integer is congruent to its one's place digit $\pmod 5$.

Mod 9: Works exactly the same way as $\pmod 3$, since $10^k \equiv 1 \pmod 9$ for all $k \in \mathbb{N}$.

Mod 10: Return the digit in the one's place (the last digit) of the input integer.

Applications to Circuits

One application which we promised to discuss is binary addition and its relation to circuits.

We can build a circuit that adds two single-digit binary numbers? There are only 4 possibilities for adding single-digit numbers, so let's examine what a truth table for this process might look like (all numbers are in base-2):

input a	input b	output $a+b$
0	0	0
0	1	1
1	0	1
1	1	10

In the output, we can always append zeros to the beginning of the binary number without changing the actual number:

input a	input b	output $a+b$
0	0	00
0	1	01
1	0	01
1	1	10

Now let's separate our output column into two columns – the sum-bit (right) and the carry-bit (left):

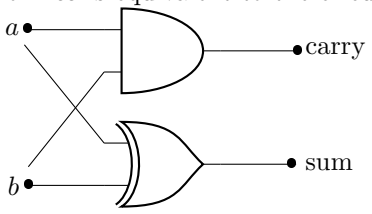
input a	input b	carry	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

We know how to generate Boolean expressions for the output columns:

$$\text{sum} \equiv a \oplus b$$
$$\text{carry} \equiv a \wedge b$$

Which gives us the **half adder**:

Adding two bits $a + b = cs$ is equivalent to the circuit



Now we can add two 1-bit numbers together. What if we want to add multiple-bit numbers together? Consider adding two 2-bit numbers:

$$\begin{array}{ccc} & c & \\ & x & y \\ + & z & w \\ \hline o_2 & o_1 & s \end{array}$$

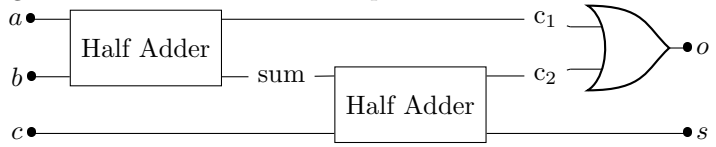
We see here that after adding $y+w$ we are left with a carry bit c for our addition $x+z$. How do we add three bits $c+x+z$? Well, we can separate it into two 1-bit additions: $c+x$ which, via a half-adder, yields a sum bit s_1 and carry bit c_1, then s_1+z which, via another half-adder, yields a sum bit s_2 and carry bit c_2. Then, we can let $o_1 = s_2$. Unfortunately this leaves us two carry bits that somehow need to be combined into a final carry bit.

Consider now the truth-table for adding three bits:

c	x	z	$c+x+z$	carry$(c+x)$	sum$(c+x)=s_1$	carry(s_1+z)
0	0	0	00	0	0	0
0	0	1	01	0	0	0
0	1	0	01	0	1	0
0	1	1	10	0	1	1
1	0	0	01	0	1	0
1	0	1	10	0	1	1
1	1	0	10	1	0	0
1	1	1	11	1	0	0

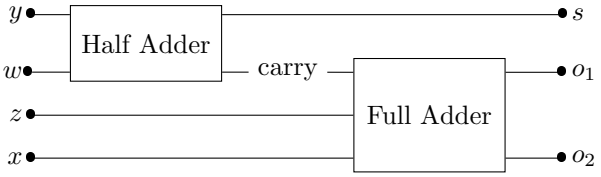
Now notice the carry bit in the output column $c+x+z$ is the same as OR-ing the two carry bits from $c+x$ and s_1+z. This completes our circuit for adding three bits. We call this the **full adder**:

Adding three bits $a+b+c=os$ is equivalent to the circuit



Putting these two structures, the half adder and full adder, together, we can construct a 2-bit adder, which solves our previous problem of doing $xy+zw = o_2o_1s$:

Adding three bits $a+b+c=os$ is equivalent to the circuit



This solves our 2-bit addition problem. What if we want to add 3-bit numbers? 4-bit numbers? n-bit numbers? Well, after the first bit column (adding two bits) we must add three bits. After this second column, we get a sum bit and a carry bit. If we tack on another column, which would make a 3-bit adder, then we add in the previous carry to the two new bits. This same process repeats for

all further columns. So, tacking on another bit solely entails tacking on another full adder! *Picture omitted.*

Example: How many half adders are required for an n-bit adder?

Solution: Note, here we implicitly assumed $n > 0$.
We need 1 half adder to start, and we need $n - 1$ full adders which each contain 2 half adders. This totals to $2(n-1)+1 = 2n-1$ half adders.

3.3 Proofs – Introduction

Recall that number theory studies integers *and their properties*. We want to think about certain properties and be able to explain why they hold. We want our explanations, *proofs*, to be valid. We want to take our knowledge base of basic principles and logically derive true/false statements. The whole idea behind a proof is to creatively apply the set of rules you know to show whether a statement hold or not.

Definition 3.3.1 Proof
A formal and logical explanation for why a statement is true or false.

Let's examine this definition. We want to explain why a statement is *true* or *false*. Recall that in first-order logic (predicate logic) we have *existential* statements and *universal* statements. This gives us four proof "bins":

	Proving True	Proving False
Existential Statement	Find an example that makes the statement True	Show the statement is False for every element
Universal Statement	Show the statement is True for every element	Find an example that makes the statement False

Proving an existential statement true, as well as proving a universal statement false, is relatively easy – all you must do is search the statement domain for an example. Comparatively, proving an existential statement false, as well as proving a universal statement true, is much harder since it takes some form of creativity – that is, if you believe $P \neq NP$[1].

[1] You will learn about this in an Algorithms course. Here is the gist: problems in the *N*on-deterministic *P*olynomial (NP) class are ones whose solution is *easy* to verify in Polynomial time. *P*olynomial class problems have Polynomial-time (relatively fast) algorithms. If $P = NP$ then problems whose solution is easy to check *also* have fast algorithms. Typically you

Before we continue, we must show you how to format your proofs. The structure in this book may be different than what your instructor prefers, so make sure you pay attention in class and follow what they do. Our structure, however, is quite flexible with minimal rules, so it should be easy to adapt. You have likely already seen our style of proof in this book. We always follow a simple structural format that puts "*Proof.*" at the start, followed by what technique we are using (typically), followed by our explanation, finished by a simple square signifying the end of the proof.

Proof. ← this signifies the start of a proof

... put logically-sound proof here ...

..... and this signifies the end of a proof → □

The "*Proof.*" header clearly signifies the following text will be a proof and should be read as such.

The ending square is shorthand for the latin phrase *quod erot demonstrandum* which, used in this context, roughly means *it is shown/demonstrated*. You may put any of the following at the end of your proofs: QED, ■, □.

This general structure is easy to read, easy to adapt, and is easily formal. It is *absolutely fine* to deviate from our structure, **so long as** your proof is still easy to read and makes clear it **is** a proof. We have seen proofs before, but here is another example to get us started.

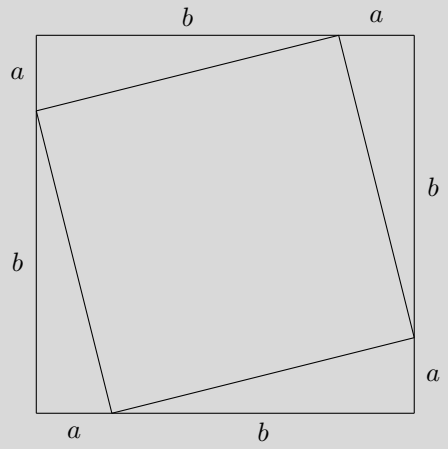
Example: Pythagoras' Theorem states that for any right-triangle, the square of the hypotenuse equals the sum of the squares of the other two sides. More familiarly,

$$a^2 + b^2 = c^2$$

Prove it.

Proof. Make 4 copies of a right triangle, and arrange them like so:

must use creativity to come up with an algorithm, however if it is just as easy to check a solution as it is to come up with a solution generator (algorithm), then creativity becomes meaningless. Suggested reading.



The area of the inner square is then

$$(a+b)^2 - 4(\frac{1}{2}ab) = a^2 + 2ab + b^2 - 2ab = a^2 + b^2$$

Denote the side length of each edge of the inner square c – which is also the hypotenuse of each right-triangle. Then c^2 is the area of the inner square, so

$$c^2 = a^2 + b^2$$

□

Remark 3.3.2
We do not expect you to know how to prove Pythagoras' Theorem – this is just an interesting proof example.

One final remark before we jump into proof techniques. There are almost always multiple ways to proof a statement. Some ways are better than others. Sometimes you may spend hours on a very long, but valid, proof, yet another proof requires less than three lines[2]. Here is a cute example.

Example: Show that

$$(\forall n \in \mathbb{N})[n^4 \mid 10 \Rightarrow n \mid 10]$$

Proof. $n \in \mathbb{N}$. For $n = 0$, $0^4 = 0 \mid 10$ and $0 \mid 10$. For $n = 1$, $1^4 = 1 \mid 10$ and $1 \mid 10$. For $n \geq 2$, $n^4 \geq 16 > 10$ so $n^4 \nmid 10$. In all cases, the $\Rightarrow$ holds.

[2]Yes, this actually happened to me in my Abstract Algebra class. No big deal.

Alternatively, we can use the transitivity of divisibility: $n \mid n^4 \mid 10$. □

3.4 Proofs – Techniques

We have a handful of proof techniques that will help you structure your proofs.

3.4.1 Direct Proof

Direct proofs are classical applications of valid reasoning (see chapter 1.2.5 about deductions). We start with the easier techniques: examples and counter-examples.

Definition 3.4.1 Proof by Example
Showing a statement is true by presenting an example. This is used for proving existential statements true

Example: Prove that there is an integer that divides 6.

Proof. $\frac{6}{2} = 3 \in \mathbb{Z}$ so by definition $2 \mid 6$, so 2 is an example. □

Example: Show that

$$(\exists z \in \mathbb{Z})(\forall n \in \mathbb{N})[z \leq n]$$

Proof. The naturals are all positive, so any negative integer works. Let $z = -1 \leq n$ for all $n \in \mathbb{N}$ be our example. □

Definition 3.4.2 Proof by Counterexample
Showing a statement is false by presenting a counterexample. This is used for proving universal statements false

Example: Prove false that all integers are divisible by 3.

Proof. $3 \nmid 2$, so 2 is a counterexample. □

Remark 3.4.3
Proof by example and counterexample are the same. To see this, consider a general existential statement we want to prove true:

$$(\exists x)[P(x)] \equiv t$$

Now consider a general universal statement we want to prove false, and negate the entire thing:

$$\neg((\forall x)[Q(x)] \equiv f)$$

$$(\exists x)[\neg Q(x)] \equiv t$$

By letting $Q(x) \equiv \neg P(x)$ then we recover a proof by example.

Definition 3.4.4 Proof by Exhaustion
Proving a statement by reasoning about every possibility. Typically used for statements about small sets

Example: Prove that $(\forall e \in \{3, 6, 9\})[3 \mid e]$

Proof. We proceed via proof by exhaustion.
There are three possible cases in this statement: $3 \mid 3$, $3 \mid 6$, $3 \mid 9$
Case 1: $e = 3$, well $3 \mid 3$ since $\frac{3}{3} = 1 \in \mathbb{Z}$
Case 2: $e = 6$, well $3 \mid 6$ since $\frac{6}{3} = 2 \in \mathbb{Z}$
Case 3: $e = 9$, well $3 \mid 9$ since $\frac{9}{3} = 3 \in \mathbb{Z}$
All cases cover the domain, and all are true, therefore the statement is true. □

Definition 3.4.5 Proof by Construction
Proving an existential statement by giving an algorithm or method to create the object in the statement.

Example: Show that there exist irrational numbers a, b such that a^b is rational.

Proof. We will show this by constructing an actual example. First note that $\sqrt{2}$ is irrational. We will prove this later. Next, note that $\log_2(9)$ is irrational. We will also prove this later. □

Example: Prove **false** that $(\forall n \in \mathbb{Z})[n \equiv 1 \pmod 4]$.

Proof. This is probably obvious, since $m = 6 \in \mathbb{Z}$ has $m \equiv 2 \not\equiv 1 \pmod 4$. Consider another, constructive argument. We will construct a counterexample. So suppose $n \equiv 1 \pmod 4$, then $\exists k \in \mathbb{Z}$ such that $n = 1 + 4k$. Then $2n = 2(1 + 4k) = 2 + 8k = 2 + 4(4k) \equiv 2 \not\equiv 1 \pmod 4$. So to construct a counterexample, we can simply double our input number n. □

Definition 3.4.6 Proof by Non-Construction
Proving an existential statement *without* providing an example, or a way to construct an example.

The following is a classic non-constructive proof of the previous example (theorem).

Example: Show that exist irrational numbers a, b such that a^b is rational.

Proof. This time, we will do this without necessarily fixing some precise values for a and b.
Note again (we will prove later) that $\sqrt{2}$ is irrational. Then consider the number $\sqrt{2}^{\sqrt{2}}$. This number is either *rational* or *irrational* (there are no other options, since it is a real number).
If $\sqrt{2}^{\sqrt{2}}$ is *rational*, then, since $\sqrt{2}$ is irrational, we have found a, b.
Else, $\sqrt{2}^{\sqrt{2}}$ is *irrational*. Let $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$, which are two irrational numbers. Then

$$a^b = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\sqrt{2}} = \sqrt{2}^2 = 2 \in \mathbb{Q}$$

So in either case, the statement holds. *Notice* that we have not constructed an actual example – whichever case holds is dependent on whether $\sqrt{2}^{\sqrt{2}}$ is rational or not! □

Definition 3.4.7 General Direct Proof
Any proof that directly uses implications and/or bi-conditionals to directly show a statement is true. These proofs are done as a straightforward application of known-to-be-true facts, rules, propositions, and theorems

Proposition 3.4.1

$$(\forall a, b, c \in \mathbb{Z})[\big((a \mid b) \wedge (a \mid c)\big) \Rightarrow (\forall s, t \in \mathbb{Z})[a \mid (sb + tc)]]$$

Before we prove it, let's understand what it means. The statement says that if a number divides two other numbers, then it must also divide any linear combination of those two other numbers. We will use this proposition in a later example.

Proof. By definition of divisibility, $a \mid b \Leftrightarrow b = ak$ for some $k \in \mathbb{Z}$ and $a \mid c \Leftrightarrow c = al$ for some $l \in \mathbb{Z}$. Take any (all) $s, t \in \mathbb{Z}$. By substitution, and our previous definitions, $sb + tc = sak + tal = a(sk + tl)$. By closure of integers under addition and multiplication, $sk + tl = m \in \mathbb{Z}$. Thus $sb + tc = am \Leftrightarrow a \mid (sb + tc)$ □

Proposition 3.4.2

$$(\forall a, b, c \in \mathbb{Z})[\big((a \mid b) \wedge (b \mid c)\big) \Rightarrow (a \mid c)]$$

Proof. By definition of divisibility, $\exists k, l \in \mathbb{Z}$ such that $b = ak$ and $c = bl$. Then $c = bl = akl = a(kl)$. $kl \in \mathbb{Z}$ by closure of integers under multiplication. Then by definition, $a \mid c$ □

3.4.2 Indirect Proof

Advanced applications of valid reasoning. These techniques are "direct proofs in disguise" because you do, in-fact, prove *a statement* directly, however the original statement you *want to prove* is proven *indirectly* as a result.

We have two main techniques for indirect proof – contraposition and contradiction.

Definition 3.4.8 Proof by Contraposition
A proof technique that allows you to somewhat *reverse* an implication statement:

$$(p \Rightarrow q) \equiv ((\neg q) \Rightarrow (\neg p))$$

In a proof by contraposition, with the goal of proving $p \Rightarrow q$, one would prove the equivalent statement $((\neg q) \Rightarrow (\neg p))$

The proof by contraposition is useful in proving implication statements when it might be easier to go the opposite direction.

Proposition 3.4.3
For all integers a, if a^2 is even then a is even.

Proof. By way of contraposition, we prove the equivalent statement: if a is odd then a^2 is odd.
a is odd so by definition $a = 2k + 1$ for some integer k. Thus,

$$\begin{aligned} a^2 &= (2k+1)^2 \\ &= 4k^2 + 2k + 1 \\ &= 2(2k^2 + k) + 1 \end{aligned}$$

By closure of integers under addition and multiplication, $2k^2+k = m \in \mathbb{Z}$. Then, $a^2 = 2m + 1$ is odd by definition.
The contrapositive of the statement is true, hence the original statement is true. □

We will use this proposition later.

Definition 3.4.9 Proof by Contradiction
A proof technique based on the following fact:

$$\begin{aligned} (\neg p) &\Rightarrow c \\ \therefore\ & p \end{aligned}$$

where c is some *logical contradiction* (ie something that is always false – for example, $p \wedge \neg p$).

A proof by contradiction exploits this fact – if one can show that the negation of a statement *must* lead to a logical contradiction, then that negation of a statement must be false.

Consider the contrapositive of the proof by contradiction:

$$(\neg p) \Rightarrow c \equiv t \Rightarrow p$$

where t is a tautology (ie something that is always true – for example, $p \vee \neg p$). Consider the truth table:

p	$\neg p$	t	$t \Rightarrow p$
F	T	T	F
T	F	**T**	**T**

The contrapositive statement is logically equivalent to p. Thus, a proof by contradiction is equivalent to directly proving the original statement.

The proof by contradiction is useful for proving statements that may "seem obvious" yet have no seemingly obvious direct proof.

Example: Prove there is no greatest integer.

Proof. By way of contradiction, assume there is a greatest integer. Call it $N \in \mathbb{Z}$. By closure of integers under addition, $N+1 \in \mathbb{Z}$. $N+1 > N$, so we have found an integer greater than N. But this contradicts our assumption that N is the greatest integer (and clearly, $N+1 \neq N$). Our assumption has lead to a contradiction, therefore our assumption must be false. Hence there is no greatest integer. □

The following argument, called the Euclidean Argument, is an important example of a proof by contradiction.

Example: Prove that $\sqrt{2} \notin \mathbb{Q}$

Proof. By way of contradiction, assume $\sqrt{2} \in \mathbb{Q}$. Then, by definition $\sqrt{2} = \frac{a}{b}$ for some $a, b \in \mathbb{Z}$ with $b \neq 0$. We necessarily assume that the ratio $\frac{a}{b}$ is in its most-reduced form (that a and b share no common divisors).
Then, $\sqrt{2} = \frac{a}{b} \Rightarrow 2 = \frac{a^2}{b^2} \Leftrightarrow 2b^2 = a^2$. b^2 is an integer by closure, thus a^2 is even. By proposition 3.4.3, it follows that a is even, so by definition $a = 2k$ for some $k \in \mathbb{Z}$.
Then, $2b^2 = a^2 = (2k)^2 = 4k^2$ so by dividing out 2, $b^2 = 2k^2$. k^2 is an integer by closure, thus b^2 is even. Again, by proposition 3.4.3, it follows that b is even, so by definition $b = 2l$ for some $l \in \mathbb{Z}$.
Then, $\sqrt{2} = \frac{a}{b} = \frac{2k}{2l} = \frac{k}{l}$ is reducible. This contradicts our assumption, hence $\sqrt{2} \notin \mathbb{Q}$. □

There is a point in this proof that may be confusing – why exactly do we need the assumption that $\sqrt{2} = \frac{a}{b}$ is in its most-reduced form? An excellent question. Let us do the proof **without** this assumption, and see what happens.

Proof. This proof is a sketch, and will skip over details noted in the previous proof.
Assume $\sqrt{2} = \frac{a}{b} \in \mathbb{Q}$. Then ... *fill in the previous proof* ... $\sqrt{2} = \frac{a}{b} = \frac{2k}{2l} = \frac{k}{l}$. Okay, then we proceed the same proof where $\sqrt{2} = \frac{k}{l}$.
If we solely examine the numerator, this gives us an infinite sequence of decreasing integers. In fact, since $\sqrt{2} > 0$ we know that all numbers in this sequence are non-negative. The sequence is as follows: $a, \frac{a}{2}, \frac{a}{4}, \frac{a}{8}, \cdots$
From the axioms that build all of mathematics, we can prove that any

decreasing sequence of non-negative integers must stop. This proof is out of scope of this book.
Thus, our numerator, and by a similar argument our denominator, *must* eventually get to a form where they are irreducible. Then, we insert our original proof, and we are done. □

Euclid's Argument can in-fact be extended to show other roots of numbers are irrational.

Example: Prove that $\sqrt{7} \notin \mathbb{Q}$

Proof. This proof will rely on the following lemma, which is given as an exercise to the reader: $(\forall a \in \mathbb{Z})[7 \mid a^2 \Rightarrow 7 \mid a]$
By way of contradiction, assume $\sqrt{7} \in \mathbb{Q}$. Then, by definition $\sqrt{7} = \frac{a}{b}$ for some $a, b \in \mathbb{Z}$ with $b \neq 0$. We necessarily assume that the ratio $\frac{a}{b}$ is in its most-reduced form (that a and b share no common divisors).
Then, $\sqrt{7} = \frac{a}{b} \Rightarrow 7 = \frac{a^2}{b^2} \Leftrightarrow 7b^2 = a^2$. b^2 is an integer by closure, thus a^2 is divisible by 7. By the above lemma, it follows that a is divisible by 7, so by definition $a = 7k$ for some $k \in \mathbb{Z}$.
Then, $7b^2 = a^2 = (7k)^2 = 49k^2$ so by dividing out 7, $b^2 = 7k^2$. k^2 is an integer by closure, thus b^2 is divisible by 7. Again, by the above lemma, it follows that b is divisible by 7, so by definition $b = 7l$ for some $l \in \mathbb{Z}$.
Then, $\sqrt{7} = \frac{a}{b} = \frac{7k}{7l} = \frac{k}{l}$ is reducible. This contradicts our assumption, hence $\sqrt{7} \notin \mathbb{Q}$. □

At this point, you may be able to see a general "machinery" at work here, and you may be wondering whether it works for all square roots.

Example: Spot the logical fallacy in the following proof that $\sqrt{4} \notin \mathbb{Q}$

Proof. Assume the following lemma holds: $(\forall a \in \mathbb{Z})[4 \mid a^2 \Rightarrow 4 \mid a]$
By way of contradiction, assume $\sqrt{4} \in \mathbb{Q}$. Then, by definition $\sqrt{4} = \frac{a}{b}$ for some $a, b \in \mathbb{Z}$ with $b \neq 0$. We necessarily assume that the ratio $\frac{a}{b}$ is in its most-reduced form (that a and b share no common divisors).
Then, $\sqrt{4} = \frac{a}{b} \Rightarrow 4 = \frac{a^2}{b^2} \Leftrightarrow 4b^2 = a^2$. b^2 is an integer by closure, thus a^2 is divisible by 4. By the above lemma, it follows that a is divisible by 4, so by definition $a = 4k$ for some $k \in \mathbb{Z}$.
Then, $4b^2 = a^2 = (4k)^2 = 16k^2$ so by dividing out 4, $b^2 = 4k^2$. k^2 is an integer by closure, thus b^2 is divisible by 4. Again, by the above lemma, it follows that b is divisible by 4, so by definition $b = 4l$ for some $l \in \mathbb{Z}$.
Then, $\sqrt{4} = \frac{a}{b} = \frac{4k}{4l} = \frac{k}{l}$ is reducible. This contradicts our assumption, hence $\sqrt{4} \notin \mathbb{Q}$. □

Solution: This is clearly a false statement, since $\sqrt{4} = 2 = \frac{2}{1} \in \mathbb{Q}$. The fallacy occurs at the beginning – assuming a lemma that does not in-fact hold. Why? Well, $4 \mid 2^2$ but $4 \nmid 2$.

The "machinery" we used before shows that the square root of some numbers (maybe you have spotted a pattern?) is irrational. This is not the only technique we have for showing numbers are irrational. In fact, in a proof by contradiction, as long as we logically arrive at a contradiction then our proof is valid. The next example demonstrates this idea for showing another number is irrational.

Example: Recall earlier we claimed that $\log_2(9)$ is irrational. Prove it.

Proof. By way of contradiction, assume $\log_2(9) \in \mathbb{Q}$. Then $\exists a, b \in \mathbb{Z}$ with $b \neq 0$ such that $\log_2(9) = \frac{a}{b}$. Then using logarithm rules,

$$\begin{aligned} \log_2(9) &= \frac{a}{b} \\ b\log_2(9) &= a \\ \log_2(9^b) &= a \\ 9^b &= 2^a \end{aligned}$$

Then 9^b is odd, and 2^a is even. But we can never have an odd equal to an even, so we have arrived at a contradiction, and thus $\log_2(9) \in \mathbb{R} \setminus \mathbb{Q}$.

Now, here we are glossing over if $a, b < 0$ and if $a = 0$. Consider these cases as an exercise! □

Theorem 3.4.4
The Unique Prime Factorization Theorem Every integer strictly bigger than 1 can be written uniquely as a product of primes. Formally,

$$(\forall n \in \mathbb{N}^{\geq 2})(\exists p_i \in \mathbf{P}, e_i \in \mathbb{N}^{\geq 1})[n = p_1^{e_1} \cdot p_2^{e_2} \cdot \ldots \cdot p_k^{e_k}]$$

where $1 \leq i \leq k$, $\mathbf{P}$ *is the set of prime numbers, and* $p_1 < p_2 < \cdots < p_k$

There are two parts to this proof – namely, that the factorization **exists** and the factorization is **unique**.

Proposition 3.4.5 Euclid's Lemma
For two integers a, b*, if a prime number* p *divides* ab*, then* p *divides* a *or* b *(or both).*

The proof of this is out of scope for this book. It requires fundamental theorems and definitions in number theory (namely, Bezout's Lemma, the Well-ordering Principle, Greatest Common Divisor definition, and Relatively Prime definition). If you are interested, to an internet search for "Euclid's Lemma".

We leave the *existential* part of the Unique Prime Factorization Theorem for later sections in this chapter. Below, we prove the *uniqueness*.

Proof. Assume we have proved the existence part of UPFT.
Suppose the factorization of some integer z is **not** unique, so it has two factorizations. Assume further that z is the *smallest* such integer. To make things easier, let's just write out each prime in both factorizations without powers (for example, we would write $24 = 2 \cdot 2 \cdot 2 \cdot 3$). So we have the factorizations

$$z = p_1 p_2 \cdots p_i = q_1 q_2 \cdots q_j$$

with $i, j \geq 2$.
By definition of divisibility, $p_1 \mid q_1 q_2 \cdots q_j$, so by Euclid's Lemma, p_1 divides one of the q's. WLOG, let $p_1 \mid q_1$. Then, since p_1 and q_1 are both prime, it necessarily follows that $p_1 = q_1$.
In our factorizations, we can then cancel out $p_1 = q_1$ to get $p_2 \cdots p_i = q_2 \cdots q_j$. This is a new non-unique factorization of an integer necessarily smaller than z. This contradicts our assumption that z is the smallest such integer with a non-unique factorization. □

Now we can use the unique factorization theorem:

Example: Prove that there are an infinite number of prime numbers

Proof. By way of contradiction, assume there are a finite number of prime numbers. Then we can list all the prime numbers: $p_1, p_2, \cdots, p_k$. Let $P = p_1 \cdot p_2 \cdot \ldots \cdot p_k$ be the product of all primes in our list. We construct a new number Q by adding 1 to P: $Q = P + 1 = p_1 \cdot p_2 \cdot \ldots \cdot p_k + 1$. Now, either Q is prime or it is not prime. If it is prime, then we have found another prime number not in our original list of primes, which contradicts our assumption.
So Q is not prime. Then, by UPFT, there is some prime factor p that divides Q. p must be in our list of primes, as per our original assumption. Thus, $p \mid P$. But $p \mid (P + 1 = Q)$. By proposition 3.4.1, we must have that $p \mid (Q - P = (P + 1) - P)$, so $p \mid 1$. But by definition no prime number can divide 1. This is a contradiction, so our original assumption must be false. Hence, primes are infinite. □

UPFT gives us another way to prove that $\sqrt{2} \notin \mathbb{Q}$

Example: Prove that $\sqrt{2} \notin \mathbb{Q}$

Proof. Suppose not, then by definition $\sqrt{2} = \frac{a}{b}$ for some $a, b \in \mathbb{Z}$ with $b \neq 0$. In order to use UPFT, we need to show that $a, b > 1$. Since $\sqrt{2} > 0$ then it follows that $a, b > 0$ (if they were both negative, then simplify to make them positive). It suffices to explain why $a, b \neq 1$. Suppose $a = 1$, then if $b = 1$ we have $\sqrt{2} = 1$ which is not true, and if $b > 1$ then $\sqrt{2} < 1 \Rightarrow 2 < 1$ is also not true. Thus $a > 1$, and we need to show that $b \neq 1$. Suppose $b = 1$, then $\sqrt{2} = a \geq 2 \Rightarrow 2 \geq 4$ is not true. So we have both $a, b > 1$.
By UPFT, both a and b have unique prime factorizations:

$$a = p_1^{e_1} \cdots p_k^{e_k}$$

$$b = q_1^{f_1} \cdots q_m^{f_m}$$

with p_i, q_j prime and e_i, f_j positive integers.
Now, since 2 is prime, it appears inside both a and b's factorization, possibly with a zero exponent $2^0 = 1$. So, re-write the factorizations with all of the 2's pulled out:

$$a = 2^e \cdot A$$

$$b = 2^f \cdot B$$

where $e, f \geq 0$ and A, B constitute the rest of the factorization.

Now, we have

$$\begin{aligned}\sqrt{2} &= \frac{a}{b}\\ 2 &= \frac{a^2}{b^2}\\ 2b^2 &= a^2\\ 2(2^f \cdot B)^2 &= (2^e \cdot A)^2\\ 2^{2f+1}B^2 &= 2^{2e}A^2\end{aligned}$$

By UPFT, the only way for this equality to be satisfied is if $2f + 1 = 2e$. This means, by definition, that an odd number equals an even number, which is impossible. We have arrived at a contradiction, so our assumption was wrong, and hence $\sqrt{2} \notin \mathbb{Q}$ □

3.4.3 Induction

Historically induction is difficult for students to learn, *at first*. We have found, though, that as students progress through the course they exclusively either

- Fully understand the principle of induction, or
- Learn how to pattern-match and guess

We hope our explanations allow you to fall in the first category.

Definition 3.4.10 Induction
A proof technique to show that some predicate, mathematical statement, $P(n)$ is true for all n in a discrete lower-bounded increasing set. Typically $n \in \mathbb{N}$.

Induction generally works in two steps. First, we generate some true knowledge about the predicate P. Then, we prove a general method true that given knowledge about P allows us to generate new knowledge about n.

There are four big techniques of induction:

1. Weak induction
2. Strong induction
3. Structural induction
4. Constructive induction

Weak Induction

The proof technique of weak induction is entirely based on the following theorem:

Theorem 3.4.6
The Principle of (weak) Mathematical Induction

$$\begin{array}{ll} & P(0) \\ & (\forall n \in \mathbb{N})[P(n) \Rightarrow P(n+1)] \\ \hline \therefore & (\forall n \in \mathbb{N})[P(n)] \end{array}$$

Included below is an equivalent definition (note the only difference is the second line):

Theorem 3.4.7
The Principle of (weak) Mathematical Induction

$$\begin{array}{ll} & P(0) \\ & (\forall n \in \mathbb{N}^{\geq 1})[P(n-1) \Rightarrow P(n)] \\ \hline \therefore & (\forall n \in \mathbb{N})[P(n)] \end{array}$$

We included both styles because sometimes one style is easier than the other when doing inductive proofs.

A classic inductive proof has 3 parts:

- The base case
- The hypothesis
- The step

Each part has a specific relation to theorems 3.4.6 and 3.4.7. We will explain these relations after the following example.

Example: Prove, $\forall n \in \mathbb{N}$,

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof. We proceed using the principle of mathematical induction (3.4.6).
Base Case: We show the statement is true for $n = 0$. Indeed,

$$\sum_{i=0}^{0} i = 0 \text{ and } \frac{0(0+1)}{2} = 0$$

so the equality holds for $n = 0$
Hypothesis: Assume the statement is true for an arbitrary $n \in \mathbb{N}$. That

is,

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Step: Prove the implication. That is, using the hypothesis, show

$$\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$$

We know that

$$\sum_{i=0}^{n+1} i = (\sum_{i=0}^{n} i) + (n+1)$$

using properties of summation. From the hypothesis we know

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Thus,

$$\begin{aligned}
\sum_{i=0}^{n+1} i &= \big(\sum_{i=0}^{n} i\big) + (n+1) \\
&= \big(\frac{n(n+1)}{2}\big) + (n+1) && \text{hypothesis} \\
&= \big(\frac{n}{2} + 1\big)(n+1) = (\frac{n+2}{2})(n+1) \\
&= \frac{(n+1)(n+2)}{2}
\end{aligned}$$

Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \in \mathbb{N}$ □

Here is the same example, but we use 3.4.7 instead.

Example: Prove, $\forall n \in \mathbb{N}$,

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof. We proceed using the principle of mathematical induction (3.4.7).

Base Case: We show the statement is true for $n = 0$. Indeed,

$$\sum_{i=0}^{0} i = 0 \text{ and } \frac{0(0+1)}{2} = 0$$

so the equality holds for $n = 0$
Hypothesis: Assume the statement is true for an arbitrary $n \in \mathbb{N}^{\geq 1}$. That is,

$$\sum_{i=0}^{n-1} i = \frac{(n-1)((n-1)+1)}{2}$$

Step: Prove the implication. That is, using the hypothesis, show

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

We know that

$$\sum_{i=0}^{n} i = (\sum_{i=0}^{n-1} i) + n$$

using properties of summation. From the hypothesis we know

$$\sum_{i=0}^{n-1} i = \frac{(n-1)((n-1)+1)}{2}$$

Thus,

$$\begin{aligned}
\sum_{i=0}^{n} i &= \big(\sum_{i=0}^{n-1} i\big) + n \\
&= \big(\frac{(n-1)((n-1)+1)}{2}\big) + n && \text{hypothesis} \\
&= \big(\frac{n-1}{2} + 1\big)(n) = (\frac{n+1}{2})(n) \\
&= \frac{n(n+1)}{2}
\end{aligned}$$

Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \in \mathbb{N}$ □

We now explain how each part of an inductive proof fits into the theorem. First notice that the theorem is a rule of inference – we are *deducing* that the theorem holds *for all* natural numbers. So, we need to *prove* the two premises to deduce that the theorem holds. The first premise, $P(0)$, is precisely the **base case**.

We initially need to show that the statement we are proving is true for the lowest possible value of n (typically we use $\mathbb{N}$, but our base case could start at 4 for example). After the base case, we need to show the second premise, that $P(n) \Rightarrow P(n+1)$. If we recall the truth table for $p \Rightarrow q$, then we recover the hypothesis. Recall that if p is *false* then the implication $p \Rightarrow q$ is *true* regardless of what q is. Now, if p is *true* then the implication is dictated entirely by q. The upshot of this is that to prove $P(n) \Rightarrow P(n+1)$ we first assume $P(n)$ is *true* (this is the inductive **hypothesis**), then we use this (since we are proving the $\Rightarrow$) to show that $P(n+1)$ is true (this is the inductive **step**).

So how does induction *work*? Well, suppose we have used induction to prove $P(n)$ for all $n \in \mathbb{N}$. So we have shown that $P(0)$ is true, and $P(n) \Rightarrow P(n+1)$. Using these two facts, we can plug in $n = 0$ into the implication to get $P(0) \Rightarrow P(1)$, and since we know $P(0)$ is true then we deduce that $P(1)$ is true. Again, we can plug in $n = 1$ into the implication to get $P(1) \Rightarrow P(2)$, and since, from before, we know $P(1)$ is true then we deduce that $P(2)$ is true. We can repeatedly do this to generate *any* $P(n)$, and hence we have that $P(n)$ holds for **all** n. Notice that this only worked because we had:

1. the base case
2. an implication, where the domain of input *includes* the base case

One final note about the inductive hypothesis. We want to assume the theorem holds *for some arbitrary* $n \in \mathbb{N}$. We want our inductive step to apply to *any* value of n, however we do **not** want to quantify n with $\forall, \exists$. We want it to be a *generic particular*. Consider the alternative – in the hypothesis you assume $P(n)$ holds for all n. In this case, you have just assumed the entire mathematical statement is true, so why do you need to prove it?

The starting point of induction is arbitrary – we could show some statement is true $\forall n \in \mathbb{N}^{\geq 4}$. The base case and hypothesis should adjust accordingly.

Example: Prove

$$(\forall n \in \mathbb{N}^{\geq 4})[2^n < n!]$$

Proof. We proceed using the principle of mathematical induction (3.4.7).
Base Case: We show the statement is true for $n = 0$. Indeed,

$$2^4 = 16 < 24 = 4!$$

so the inequality holds for $n = 4$
Hypothesis: Assume the statement is true for some arbitrary $n-1 \geq 4$. That is,

$$2^{n-1} < (n-1)!$$

Step: Prove the implication. That is, using the hypothesis, show

$$2^n < n!$$

We know that

$$2^n = 2 \cdot 2^{n-1}$$

and from the hypothesis

$$2^{n-1} < (n-1)!$$

and since $n - 1 \geq 4 \Leftrightarrow n \geq 5 > 2$ then

$$\begin{aligned} 2^n &= 2 \cdot 2^{n-1} \\ &< 2(n-1)! && \text{hypothesis} \\ &< n(n-1)! \\ &= n! \end{aligned}$$

Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \in \mathbb{N}$ □

We can prove any sort of statement with n in a discrete lower-bounded increasing set. (note that sets are by definition un-ordered, however in this case we can place the natural ordering on $\mathbb{N}$ – sorted order)

Example: Prove

$$(\forall n \in \mathbb{N})[8 \mid (2n+1)^2 - 1]$$

Proof. We proceed using the principle of mathematical induction (3.4.6).
Base Case: We show the statement is true for $n = 0$. Indeed,

$$(2 \cdot 0 + 1)^2 - 1 = 1 - 1 = 0$$

and zero is divisible by anything, so the statement holds for $n = 0$
Hypothesis: Assume the statement is true for some arbitrary $n \geq 0$. That is,

$$8 \mid (2n+1)^2 - 1$$

Step: Prove the implication. That is, using the hypothesis, show

$$8 \mid (2(n+1)+1)^2 - 1$$

From the hypothesis we have that

$$8 \mid (2n+1)^2 - 1 \Leftrightarrow (\exists k \in \mathbb{Z})[(2n+1)^2 - 1 = 8k]$$

thus,

$$\begin{aligned}(2(n+1)+1)^2-1 &= (2n+3)^2-1 \\ &= 4n^2+12n+9-1 \\ &= 4n^2+4n+1-1+8n+8 \\ &= (2n+1)^2-1+8n+8 \\ &= 8k+8n+8 && \text{hypothesis} \\ &= 8(k+n+1)\end{aligned}$$

By closure we have $k+n+1 \in \mathbb{Z}$, so by definition $8 \mid 8(k+n+1)$, thus $8 \mid (2(n+1)+1)^2-1$
Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \in \mathbb{N}$ □

Strong Induction

Weak induction is great, however in its current form it is not useful enough to prove some mathematical statements.

Example: Show that $a_n \leq 3^n$ for all $n \in \mathbb{N}$ where

$$a_n = \begin{cases} 1 & n = 0 \\ 2 & n = 1 \\ 3 & n = 2 \\ a_{n-1}+a_{n-2}+a_{n-3} & n \geq 3 \end{cases}$$

Consider how a proof using weak induction might go. There is one base case in a weak induction proof, however there are 3 base cases in the sequence a_n. If we want to prove $P(n)$, then the hypothesis tells us $P(n-1)$ is true, so we have $a_{n-1} \leq 3^{n-1}$. This does not help us, though, since in the inductive step we will have to use $a_n = a_{n-1}+a_{n-2}+a_{n-3}$ but the hypothesis gives us nothing about a_{n-2} and a_{n-3}.
What should we do, then? Strong induction to the rescue!

Theorem 3.4.8
The Principle of (strong) Mathematical Induction

$$\begin{array}{ll} & P(0) \wedge P(1) \wedge \cdots \wedge P(k) \\ & (\forall n \in \mathbb{N}^{>k})[(\bigwedge_{i=0}^{n-1} P(i)) \Rightarrow P(n)] \\ \hline \therefore & (\forall n \in \mathbb{N})[P(n)] \end{array}$$

Remark 3.4.11
We can "go from" n to $n+1$ as we did in Theorem 3.4.6.

As a rule-of-thumb, we use strong induction when we think we might need multiple hypothesis values to prove $P(n)$ instead of needing just $P(n-1)$.

Example: Show Example 3.4.3 using strong induction.

Proof. We proceed using the principle of mathematical induction (3.4.8).
Base Case: We show the statement is true for $n = 0, 1, 2$.
For $n = 0$, $a_0 = 1 \leq 1 = 3^0$
For $n = 1$, $a_1 = 2 \leq 3 = 3^1$
For $n = 2$, $a_2 = 3 \leq 9 = 3^2$
Hypothesis: Assume the statement is true for all $i \in \{0, 1, \cdots, n-1\}$ for some arbitrary $n - 1 \geq 2$. That is,

$$a_i \leq 3^i$$

Step: Prove the implication. That is, using the hypothesis, show that $a_n \leq 3^n$
From the hypothesis, $n - 1 \geq 2$ so $n \geq 3$ and we can apply the recursive definition to get $a_n = a_{n-1} + a_{n-2} + a_{n-3}$. Quite clearly, all of the indices $n-1$, $n-2$, and $n-3$ fall in the induction hypothesis bounds, so $a_{n-1} \leq 3^{n-1}$, $a_{n-2} \leq 3^{n-2}$, and $a_{n-3} \leq 3^{n-3}$. All together, and using rules of inequalities, we have:

$$\begin{aligned}
a_n &= a_{n-1} + a_{n-2} + a_{n-3} \\
&< 3^{n-1} + 3^{n-2} + 3^{n-3} && \text{hypothesis} \\
&= 3^{n-3}(3^2 + 3^1 + 3^0) \\
&= 3^{n-3}(13) \\
&< 3^{n-3}(27) && \text{since } n - 3 > 0 \\
&= 3^{n-3} \cdot 3^3 = 3^n
\end{aligned}$$

Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \in \mathbb{N}$ □

Compare this proof to one using weak induction. Notice how we proved multiple base cases instead of just one, and notice how we used i instead of n in the hypothesis "that is" part. The step was essentially the same, but we used multiple hypothesis substitutions.

Example: Prove the existence part of the UPFT – ie prove that every integer above 1 can be factored into a product of primes.

Proof. We proceed using the principle of mathematical induction (3.4.8).
Base Case: We show the statement is true for $n = 2$. Indeed, 2 is prime so we are done.
Hypothesis: Assume the statement is true for all $i \in \{2, 3, \cdots, n-1\}$ for some arbitrary $n - 1 \geq 2$. That is, i has a prime factorization.
Step: Prove the implication. That is, using the hypothesis, show that n has a prime factorization.
We break into cases. If n is prime, then we are done. Otherwise, n is composite and can be written as $n = ab$ for some integers a and b. Necessarily $2 \leq a, b < n$ so by the induction hypothesis a and b have prime factorizations. Then, n's factorization is the product of a and b's factorizations.
Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n > 1$ □

In our previous weak induction proofs, we typically used the hypothesis without much thought. This is fine when the hypothesis constraints are obviously satisfied. If it is not obvious that some element satisfies the hypothesis constraints, then you need to prove why you can apply the hypothesis. This was done in the above proof, and is showcased in the proof below.

Example: Prove that $a_n \equiv 2 \pmod 5$ for all $n \in \mathbb{N}$ where

$$a_n = \begin{cases} 12 & n = 0 \\ 17 & n = 1 \\ 4a_{\lfloor \log_2 n \rfloor} + 2a_{\lceil n/2 \rceil} & n \geq 2 \end{cases}$$

Proof. We proceed using the principle of mathematical induction (3.4.8).
Base Case: We show the statement is true for $n = 0, 1$.
For $n = 0$, $a_0 = 12 \equiv 7 \equiv 2 \pmod 5$
For $n = 1$, $a_1 = 17 \equiv 12 \equiv 2 \pmod 5$
Hypothesis: Assume the statement is true for all $i \in \{0, 1, \cdots, n-1\}$ for some arbitrary $n - 1 \geq 1$. That is,

$$a_i \equiv 2 \pmod 5$$

Step: Prove the implication. That is, using the hypothesis, show that $a_n \equiv 2 \pmod 5$
From the hypothesis, $n - 1 \geq 1$ so $n \geq 2$ and we can apply the recursive

definition to get $a_n = 4a_{\lfloor \log_2 n \rfloor} + 2a_{\lceil n/2 \rceil}$.
We now need to show that both indices $\lfloor \log_2 n \rfloor$ and $\lceil n/2 \rceil$ fall in the induction hypothesis bounds. Clearly both indices are integers because of the floor and ceiling functions, so it suffices to show that $0 \leq \lfloor \log_2 n \rfloor, \lceil n/2 \rceil \leq n-1$.
First, $2 \leq n \Leftrightarrow \log_2 2 \leq \log_2 n \Rightarrow \lfloor 1 \rfloor \leq \lfloor \log_2 n \rfloor$ and since $0 \leq 1 = \lfloor 1 \rfloor$ then $0 \leq \lfloor \log_2 n \rfloor$.
Second, since $\lfloor \log_2 n \rfloor \leq \log_2 n$ and $n \leq 2^{n-1} \Leftrightarrow \log_2 n \leq n-1$ then $\lfloor \log_2 n \rfloor \leq n-1$. Note: $n \leq 2^{n-1}$ only works when $n > 0$, but we have that $n \geq 2$ so we are good.
Third, $2 \leq n \Leftrightarrow 1 \leq n/2 \Rightarrow \lceil 1 \rceil \leq \lceil n/2 \rceil$ and since $0 \leq 1 = \lceil 1 \rceil$ then $0 \leq \lceil n/2 \rceil$.
Fourth, $\lceil n/2 \rceil < n/2+1 = \frac{n+2}{2} \leq n-1$ if $n+2 \leq 2n-2 \Leftrightarrow 4 \leq n$. But we have $2 \geq n$. So we need to show the original inequality holds for $n = 2, 3$. If $n = 2$ then $\lceil 2/2 \rceil = 1 \leq 2-1 = 1$. If $n = 3$ then $\lceil 3/2 \rceil = 2 \leq 3-1 = 2$. So we are good.
Thus, by the induction hypothesis, $a_{n-1} \equiv 2 \pmod 5$ and $a_{n-2} \equiv 2 \pmod 5$. All together, and using rules of modular arithmetic, we have:

$$\begin{aligned} a_n &= 4a_{\lfloor \log_2 n \rfloor} + 2a_{\lceil n/2 \rceil} \\ &\equiv 4 \cdot 2 + 2 \cdot 2 && \text{hypothesis} \\ &= 12 \equiv 2 \pmod 5 \end{aligned}$$

Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \in \mathbb{N}$ □

Remark 3.4.12
You should prove to yourself that $x \leq y \Rightarrow \lfloor x \rfloor < \lfloor y \rfloor$ and similarly for the ceiling.

Example: Show that any amount of dollars at least 12 can be made exclusively from 3 and 7 dollar bills. In other words show $(\forall n \geq 12)(\exists x, y \in \mathbb{N})[n = 3x + 7y]$

Proof. We can prove this with weak induction, but for this example we will prove it with strong induction.
We proceed using the principle of mathematical induction (3.4.8).
Base Case: We show the statement is true for $n = 12, 13, 14$.
$12 = 3 \cdot 4 + 7 \cdot 0$
$13 = 3 \cdot 2 + 7 \cdot 1$

$14 = 3 \cdot 0 + 7 \cdot 2$
Hypothesis: Assume the statement is true for all $i \in \{12, 13, \cdots, n-1\}$ for some arbitrary $n - 1 \geq 14$. That is,

$$(\exists x, y \in \mathbb{N})[i = 3x + 7y]$$

Step: Prove the implication. That is, using the hypothesis, show that $(\exists x, y \in \mathbb{N})[n = 3x + 7y]$
From the hypothesis, $(\exists x, y \in \mathbb{N})[n-3 = 3x+7y]$. Then $n-3+3 = n = 3x + 7y = 3(x + 1) + 7y$ and since $x + 1 \in \mathbb{Z}$ by closure we are done.
Therefore, by the principle of mathematical induction we conclude the statement holds $\forall n \geq 12$ □

Interestingly, strong induction and weak induction are equivalent techniques. Any proof by weak induction can be done by strong induction, and vice versa.

Theorem 3.4.9 Induction Equivalence
Strong induction $\Leftrightarrow$ *Weak induction.*

Proof. $\Leftarrow$ Assume we have shown that $(\forall n\ in \mathbb{Z}^+)[P(n-1) \Rightarrow P(n)]$. Then necessarily $(P(0) \wedge P(1) \wedge \cdots \wedge P(n-1)) \Rightarrow P(n)$ (logic exercise).
$\Rightarrow$ Assume we can show $P(n)$ using strong induction. Then, let $Q(n) \equiv P(0) \wedge P(1) \wedge \cdots \wedge P(n)$. We know that $(\forall n)[P(n)] \equiv (\forall n)[Q(n)]$ (logic exercise). Then it suffices to prove $Q(n)$ for all n using weak induction.
This is left as an exercise for the reader. □

Usually, though, one technique will be significantly easier than the other. In this case, use the easier technique. In higher level mathematics, we are less formal about using induction. Typically the induction hypothesis is not written, and trivial base cases are hand-waved. We recommend being explicit in your induction proofs until you fully understand what is going on. Once you do, then you can use induction as one simple idea, and most of your induction proofs will under-the-hood be "strong" induction proofs.

Structural Induction

Our next induction technique is structural induction. There is no formal theorem for structural induction since it is essentially equivalent to our previous forms of induction. Interestingly, all forms of induction are equivalent to the *well-ordering principle*, however this is outside the scope of this book.

Structural induction differs from the other forms in that we are proving mathematical statements on recursively-defined *structures* instead of on a sequence of increasing integers. For example, we can induct on binary trees, linked lists,

sets, fractals, and others. Since these structures are recursively defined, they must have base cases, which are the smallest possible sub-structures that build up the main structure.

Example: A **Non-Empty Full Binary Tree** is either:

- a root node r with no children, or
- a root node r with 2 children, T_L and T_R, both Non-Empty Full Binary Trees

Prove $V(T) = E(T) + 1$ for all Non-Empty Full Binary Trees T

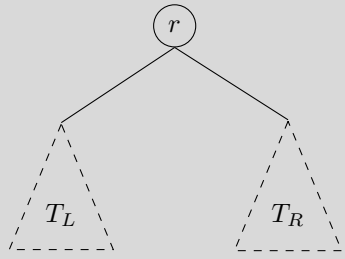
Proof. We show $V(T) = E(T) + 1$ via structural induction on T.
Base Case:

T has a single node r with no children:

T contains one node and no edges, so $V(T) = 1$ and $E(T) = 0$. Plug these into the proposition and we see $1 = 0 + 1$, so the proposition holds for the base case.

Step:

We recursively construct a new tree T_{new} by taking a root node r and pointing the left child to T_L and the right child to T_R. T_L and T_R are both Non-Empty Full Binary Trees that maintain the property $V(T) = E(T) + 1$. Here's how it looks:



We now show that $V(T_{new}) = E(T_{new}) + 1$. Well,

$V(T_{new})$	$=$	$1 + V(T_L) + V(T_R)$	by definition
	$=$	$1 + E(T_L) + 1 + E(T_R) + 1$	by inductive hypothesis
	$=$	$E(T_L) + E(T_R) + 1 + 1 + 1$	re-arranging for readability
	$=$	$E(T_L) + E(T_R) + 2 + 1$	algebra
	$=$	$E(T_{new}) + 1$	by definition

The inductive step shows that by applying the recursive definition of Non-Empty Full Binary Trees the property holds true, hence $V(T) = E(T)+1$ by structural induction. □

Essentially the main idea of induction is the same. We use the recursive defi-

nition, or previous element, to construct the new one, and then show the new thing maintains the proposition.

Example: A **Language** $\mathcal{L}$ has the alphabet $\Sigma = \{a, b, c\}$. Strings in the language are generated as follows:

- the empty string $\epsilon \in \mathcal{L}$
- $(\forall \sigma \in \mathcal{L})[ba\sigma ab^3 \in \mathcal{L} \wedge cb\sigma ba^2c \in \mathcal{L}]$

Prove that all strings in $\mathcal{L}$ contain an even number of a's, b's, and c's.

Proof. We show the proposition via structural induction on σ.
Base Case:

$\sigma = \epsilon$ is the empty string: “”

σ contains zero a's, b's, and c's. Zero is even, so we are done.

Step:

We recursively construct a new string σ_{new} using either of the two rules defined above, where $\sigma \in \mathcal{L}$ is a string that maintains the proposition (ie falls in the induction hypothesis):

1. $\sigma_{new} = ba\sigma ab^3$
2. $\sigma_{new} = cb\sigma ba^2c$

We now show that both new strings have even numbers of a's, b's, and c's.
Denote $N(\sigma, \cdot) \equiv$ the number of input $\cdot$ characters in σ. The induction hypothesis tells us that $(\forall * \in \{a, b, c\})[2 \mid N(\sigma, *)]$.
(1)
$N(\sigma_{new}, a) = 1 + N(\sigma, a) + 1$ is even by definition.
$N(\sigma_{new}, b) = 1 + N(\sigma, b) + 3$ is even by definition.
$N(\sigma_{new}, c) = 0 + N(\sigma, c)$ is even by definition.
(2)
$N(\sigma_{new}, a) = N(\sigma, a) + 2$ is even by definition.
$N(\sigma_{new}, b) = 1 + N(\sigma, b) + 1$ is even by definition.
$N(\sigma_{new}, c) = 1 + N(\sigma, c) + 1$ is even by definition.
The inductive step shows that by applying the recursive definition of $\mathcal{L}$ the property holds true, hence every string in $\mathcal{L}$ has an even numbers of a's, b's, and c's by structural induction. □

Constructive Induction

Our final induction technique is constructive induction. Like structural induction, there is no formal theorem for constructive induction. Instead, constructive induction utilizes the previous induction techniques.

Sometimes you will be presented a proposition with unknowns. In a linear equation, for example $5 = 2x + 3$, you would solve for x. In a proposition, for example

$$(\forall n \in \mathbb{N})[4 \mid (cn - 2)^d]$$

you would solve for c and d. Constructive induction is simply doing a proof by induction to find constants that make a proposition true. In a constructive induction proof, you proceed through a regular induction proof as if your unknowns are *known*. After the inductive step, you will have a handful of constraints on your unknowns. Then, you simply find actual values that satisfy your constraints.

Example: Find $c, d \in \mathbb{Z}^+$ such that $(\forall n \in \mathbb{N})[4 \mid (cn - 2)^d]$.

Proof. We proceed via a proof by constructive weak induction.
Base: $n = 0$, then we need $4 \mid (c \cdot 0 - 2)^d$ which tells us that $4 \mid (-2)^d$, and since the sign has no effect on divisibility, then $4 \mid 2^d$ so (by UPFT) we need $d \geq 2$.
Hypo: Assume for some arbitrary $n \geq 0$ that $4 \mid (cn - 2)^d$.
Step: Show that $4 \mid (c(n + 1) - 2)^d$.
For ease, we guess that $d = 2$. If it does not, then the induction will not work. If it does, then we were lucky!
$(c(n + 1) - 2)^2 = (cn + c - 2)(cn + c - 2) = c^2n^2 + c^2n - 2cn + c^2n + c^2 - 2c - 2cn - 2c + 4 = (c^2n^2 - 4cn + 4) + 2c^2n + c^2 - 4c$ and from the hypothesis $4 \mid c^2n^2 - 4cn + 4$ so thus we need $4 \mid 2c^2n + c^2 - 4c$. Since $4 \mid -4c$ then we only need $4 \mid 2c^2n + c^2$.
Quite easily we can see that $c = 2$ makes the induction work. Therefore, our original guess that $d = 2$ worked!
So our final proposition is $4 \mid (2n - 2)^2$ for all $n \in \mathbb{N}$. □

The above example was easy to guess without constructive induction, however this will not always be the case.

Example: Find constants $a, b, c, d \in \mathbb{R}$ such that

$$(\forall n \in \mathbb{N})[\sum_{i=0}^{n} i^2 = an^3 + bn^2 + cn + d]$$

Proof. We proceed via a proof by constructive weak induction.
Base: $n = 0$, then we have $\sum_{i=0}^{0} i^2 0^2 = 0$ and $a0^3 + b0^2 + c0 + d = d$. For the induction to work, we need the left-hand side to equal the right-hand side. Thus, $d = 0$
Hypo: Assume for some arbitrary $n \in \mathbb{N}$ that $\sum_{i=0}^{n} i^2 = an^3 + bn^2 + cn$
Step: Show that $\sum_{i=0}^{n+1} i^2 = a(n+1)^3 + b(n+1)^2 + c(n+1)$.
We proceed like normal weak induction.

$$\begin{aligned}\sum_{i=0}^{n+1} i^2 &= (\sum_{i=0}^{n} i^2) + (n+1)^2 \\ &= (an^3 + bn^2 + cn) + (n+1)^2 \qquad \text{hypothesis}\end{aligned}$$

At this point, we would do some algebra and simplify to get our equation to $= a(n+1)^3 + b(n+1)^2 + c(n+1)$. Unfortunately, we do not know what a, b, c are. This is where constructive induction comes to play – we set the equality in question and find values of a, b, c that make the equality work!

$$an^3 + bn^2 + cn + (n+1)^2 = a(n+1)^3 + b(n+1)^2 + c(n+1)$$

If we expand out terms we get:
LHS: $an^3 + bn^2 + cn + n^2 + 2n + 1$
RHS: $a(n^3 + 3n^2 + 3n + 1) + b(n^2 + 2n + 1) + c(n+1)$
Then if we match up the n^3, n^2, n^1, and n^0 terms, we get a system of equations:

$$\begin{aligned} an^3 &= an^3 \\ bn^2 + n^2 &= a3n^2 + bn^2 \\ cn + 2n &= a3n + b2n + cn \\ 1 &= a + b + c \end{aligned}$$

If we divide out the n's and solve, we get:

$$\begin{aligned} 1 &= 1 \\ b + 1 &= 3a + b && \Rightarrow a = \frac{1}{3} \\ c + 2 &= 3a + 2b + c && \Rightarrow b = \frac{2 - 3a}{2} = \frac{1}{2} \\ 1 &= a + b + c && \Rightarrow c = 1 - a - b = \frac{1}{6} \end{aligned}$$

So we get

$$\sum_{i=0}^{n} i^2 = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$$

□

Remark 3.4.13
The reader should verify that the above solution agrees with the known-solution

$$\sum_{i=0}^{n} i^2 = \frac{(n)(n+1)(2n+1)}{6}$$

Constructive induction is most valuable for proving upper and lower bounds on recurrences. We will discuss this in a later chapter. For now, problems like this will look as follows.

Example: Find the smallest possible $c \in \mathbb{R}^+$ and $d \in \mathbb{N}$ such that $a_n \leq cd^n$ where

$$a_n = \begin{cases} 2 & n = 0 \\ 4 & n = 1 \\ 4a_{n-1} + 3a_{n-2} & n \geq 2 \end{cases}$$

Proof. We proceed via a proof by constructive strong induction.
Base:
$n = 0$, then we need $a_0 \leq cd^0$ so we get $2 \leq c$
$n = 1$, then we need $a_1 \leq cd^1$ so we get $4 \leq cd$
Hypo: Assume for all $0 \leq i < n$ for some arbitrary $n > 1$ that $a_i \leq cd^i$
Step: Show that $a_n \leq cd^n$.

$$\begin{aligned} a_n &= 4a_{n-1} + 3a_{n-2} \\ &\leq 4cd^{n-1} + 3cd^{n-2} && \text{hypothesis} \end{aligned}$$

To make the induction work, we then need $4cd^{n-1} + 3cd^{n-2} \leq cd^n$. So, we simplify this inequality and minimize our constants.

$$\begin{aligned} 4cd^{n-1} + 3cd^{n-2} &\leq cd^n \\ 4d + 3 &\leq d^2 \\ 0 &\leq d^2 - 4d - 3 \end{aligned}$$

Since $d \in \mathbb{N}$ then we can plug in values of d into our polynomial until the

inequality holds:

d	0	1	2	3	4	5	6
d^2-4d-3	-3	-6	-7	-6	-3	2	9

So to minimize $d \in \mathbb{N}$ and still make the inequality $0 \leq d^2 - 4d - 3$ work we let $d = 5$.
Knowing this, we go back to the base case and satisfy the constraints $2 \leq c$ and $4 \leq cd \Rightarrow \frac{4}{5} \leq c$. So the minimal $c \in \mathbb{R}^+$ that satisfies both constraints is $c = 2$.
So we get

$$a_n \leq 2 \cdot 5^n$$

□

Remark 3.4.14
We could have used the quadratic formula to find the exact roots, then chosen the ceiling of the positive root.

Sometimes the unknown may be in a different place.

Example: Find the largest possible $x \in \mathbb{N}$ such that $a_n \leq 20n$ where

$$a_n = \begin{cases} 0 & n = 0 \\ a_{\lfloor n/3 \rfloor} + a_{\lfloor 2n/5 \rfloor} + xn & n > 0 \end{cases}$$

Proof. We proceed via a proof by constructive strong induction.
Base:
$n = 0$, then we have $a_0 = 0 \leq 0 = 20 \cdot 0$. We learn nothing from the base case.
Hypo: Assume for all $0 \leq i < n$ for some arbitrary $n > 0$ that $a_i \leq 20i$
Step: Show that $a_n \leq 20n$. During the process, we will find a value for x.

$$\begin{aligned} a_n &= a_{\lfloor n/3 \rfloor} + a_{\lfloor 2n/5 \rfloor} + xn \\ &\leq 20\lfloor n/3 \rfloor + 20\lfloor 2n/5 \rfloor + xn && \text{hypothesis} \\ &\leq 20\frac{n}{3} + 20\frac{2n}{5} + xn && \text{floor defn.} \\ &\leq 20n \end{aligned}$$

To make the induction work, we simplify the inequality:

$$\begin{aligned} 20\frac{n}{3} + 20\frac{2n}{5} + xn &\leq 20n \\ \frac{20}{3} + \frac{40}{5} + x &\leq 20 \\ 20 + 3 \cdot 8 + 3x &\leq 60 \\ 3x &\leq 60 - 20 - 24 \\ x &\leq \frac{16}{3} = 5 + \frac{1}{3} \end{aligned}$$

Since $x \in \mathbb{N}$ the largest value of x that works is 5. So we get

$$x = 5$$

□

Induction Templates

Here are some templates we use for induction proofs. These should be used as a starting point until you fully understand what is going on. Some professors may do induction proofs differently – follow their style.

Weak Induction

Style 1

Proof. Let $P(n)$ be the proposition *insert statement here*. We proceed via W.M.I. on n.
Base: Prove $P($*lowest base case index*$)$
insert math here that shows the proposition is true for the lowest base case
Hypo: Assume $P(n)$ is true for arbitrary $n \geq$ *lowest base case index*, i.e.: *plug n into the proposition*
Step: Show $P(n) \Rightarrow P(n+1)$, i.e.: *plug* $n+1$ *into the proposition*
insert math here, making sure you ***use the IH***
The IS is true, hence by the P.M.I. *insert full proposition* is true. □

Style 2

Proof. Let $P(n)$ be the proposition *insert statement here*. We proceed via W.M.I. on n.
Base: Prove $P($*lowest base case index*$)$
insert math here that shows the proposition is true for the lowest base

case
Hypo: Assume $P(n-1)$ is true for arbitrary $n >$ *lowest base case index*, i.e.: *plug* $n-1$ *into the proposition*
Step: Show $P(n-1) \Rightarrow P(n)$, i.e.: *plug* n *into the proposition*
insert math here, making sure you ***use the IH***
The IS is true, hence by the P.M.I. *insert full proposition* is true. □

Strong Induction

Proof. Let $P(n)$ be the proposition *insert statement here.* We proceed via S.M.I. on n.
Base: Prove $P(k)$ for all k base case indices
insert math here that shows the proposition is true for every base case
Hypo: Assume $P(i)$ is true for all *lowest base index* $\leq i \leq n-1$ for arbitrary $n-1 \geq$ *highest base case index*, i.e.: *plug* i *into the proposition*
Step: Show $P(n)$, i.e.: *plug* n *into the proposition*
insert math here, making sure you ***use the IH***
The IS is true, hence by the P.M.I. *insert full proposition* is true. □

Structural Induction

Proof. We show *insert statement here* via Structural Induction on *the variable that represents the structure.*
Base: *show the statement holds for every base case laid out in the recursively-defined structure*
insert math here
Hypo: *okay to omit*
Step: We recursively construct a new *object* by *precisely follow the recursive definition.* The *recursive* part(s) of the new object fall in the inductive hypothesis, so satisfy the property *plug those recursive parts into the statement. show a picture of the new structure.* We now show the property holds for our new structure.
insert math here, making sure you ***use the IH***
The IS is true, hence by Structural Induction *insert full proposition* is true. □

Constructive Induction

Just follow the previous techniques. At the end of the step, usually you will be solving some form of algebraic system.

3.4.4 Combination of Techniques

We can combine our previously-defined proof techniques to accomplish any goal we want.

Definition 3.4.15 Proof by Cases
A more general proof technique where you break a statement into multiple sub-statements, and prove each sub-statement individually. The sub-statements **must** fully cover the original statement (there cannot be anything missing)

Example: Explain, for the statement $(\forall x \in \mathbb{Z})[2 \mid x^2 + x]$, why the following proof is invalid.

Proof. When x is even then $\exists k \in \mathbb{Z}$ such that $x = 2k$, and thus $x^2 + x = (2k)^2 + 2k = 4k^2 + 2k = 2(2k^2 + k) = 2q$ where $q = 2k^2 + k \in \mathbb{Z}$ by closure of integers under addition and multiplication. So $x^2 + x$ is divisible by 2 by definition □

Solution: It is invalid because the proof does not fully cover all possibilities of the statement. The proof covers all even integers, however odd integers are nowhere found in the proof. Clearly $\mathbb{Z} \neq \mathbb{Z}^{\text{even}}$.
This proof should have been a proof by cases. One case is the proof already given, and the other case should be when x is odd.

Example: Finish the previous proof.

Proof. (continued)
Case 2: x is odd, then $\exists l \in \mathbb{Z}$ such that $x = 2l + 1$, then $x^2 + x = (2l+1)^2 + 2l + 1 = 4l^2 + 4l + 1 + 2l + 1 = 4l^2 + 6l + 2 = 2(2l^2 + 3l + 1) = 2r$ where $r = 2l^2 + 3l + 1 \in \mathbb{Z}$ by closure of integers under addition and multiplication. So $x^2 + x$ is divisible by 2 by definition
Both cases cover the entirety of the integers by parity, thus the statement holds for all integers □

Remark 3.4.16
Both cases in the previous proof were direct proofs, however in other examples there could be cases that are proven indirectly. The previous proof used two cases, however there could be examples using more (e.g. if you want to prove $(\forall a \in \mathbb{Z})[7 \mid a^2 \Rightarrow 7 \mid a]$).

3.5 Summary

- There are many definitions and concepts in number theory.

- There are many proof techniques in number theory.
- To be good at proofs, you must internalize all definitions, think outside the box, and practice a lot.

3.6 Practice

1. Re-write the following sum as a summation:

$$1 - 4 + 7 - 10 + 13 - 16 + 19 - 22 + \cdots \pm (3n - 2)$$

2. Re-write the following product as a product:

$$n^{1/2} \times (2n)^{1/4} \times (3n)^{1/6} \times (4n)^{1/8} \times \cdots$$

3. Calculate $\log_2 3 \times \log_3 4 \times \cdots \times \log_{31} 32$ without a calculator.
4. Show that $\lfloor x \rfloor + \lfloor y \rfloor \leq \lfloor x + y \rfloor \leq \lfloor x \rfloor + \lfloor y \rfloor + 1$ for all $x, y \in \mathbb{R}$.
5. Show that $\lfloor x \rfloor = x \Leftrightarrow x \in \mathbb{Z}$.
6. Explain the difference between $\equiv$ (congruence) and % (remainder).
7. Show that $(\forall m, n \in \mathbb{Z}^{\neq 0})[(n \mid m \wedge m \mid n) \Rightarrow m = \pm n]$.
8. Come up with a rule for $\pmod 4$ and $\pmod{11}$.
9. Fill in the following number-base translation table:

Base 2	Base 8	Base 10	Base 16
10110			
	703		
		1000	
			0xDAB

10. Prove there is no smallest integer.
11. Prove $7 \mid a^2 \Rightarrow 7 \mid a$ for any $a \in \mathbb{Z}$.
12. Prove $2 \mid a^3 \Rightarrow 2 \mid a$ for any $a \in \mathbb{Z}$. Then prove that $\sqrt[3]{2} \notin \mathbb{Q}$ by the Euclidean Argument and by UPFT.
13. Generalize the lemmas used in the root-irrationality proofs – namely, prove that for any prime p, integer a, and integer $n > 1$, $p \mid a^n \Rightarrow p \mid a$. Then, prove the general claim that $\sqrt[n]{p} \notin \mathbb{Q}$ by the Euclidean Argument and by UPFT.
14. Show that $2 \mid n(n+1)$ using a direct proof and using weak induction.

15. Show that for any $n \times n$ matrix A with $n > 1$ that

$$\sum_{i=2}^{n}\sum_{j=1}^{i-1} A_{ij} = \sum_{j=1}^{n-1}\sum_{i=j+1}^{n} A_{ij}$$

where A_{ij} is the value of A in the i-th row and j-th column.

16. Prove that $\sqrt[n]{n} < 2 - \frac{1}{n}$ for all $n \geq 2$.

17. Prove $\forall x \neq 0,\ \forall n \in \mathbb{N}^{>1}$ that $x^2 \mid ((x+1)^n - nx - 1)$

18. Prove the coin problem in Example 3.4.3 using weak induction.

19. Prove $\left(\bigwedge_{i=1}^{n-1}(p_i \Rightarrow p_{i+1})\right) \Rightarrow \left((\bigwedge_{i=1}^{n-1} p_i) \Rightarrow p_n\right)$ for all $n > 1$

20. Prove that $(\forall n \in \mathbb{N})[s_n \equiv 0 \pmod 2]$ where

$$s_n = \begin{cases} 2, & n = 0 \\ 4, & n = 1 \\ 3s_{n-1} + s_{n-2}, & n \geq 2 \end{cases}$$

21. Solve the quadratic equation $x^2 - x - 1 = 0$ – name the larger root ϕ and the smaller root ψ. Then, justify why $\phi^2 = \phi + 1$ and $\psi^2 = \psi + 1$. Finally, use these to show that $f_n = \frac{\sqrt{5}}{5}(\phi^n - \psi^n)$ where

$$f_n = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ f_{n-1} + f_{n-2} & n \geq 2 \end{cases}$$

22. Let

$$l_n = \begin{cases} 2 & n = 0 \\ 1 & n = 1 \\ l_{n-1} + l_{n-2} & n \geq 2 \end{cases}$$

Show that $l_n = \phi^n + \psi^n$

23. Let $S \subset \mathbb{Z}^2$ be defined recursively as:

- $(0,0) \in S$
- $(\forall (x,y) \in S)[(x, y+1) \in S \wedge (x+1, y+1) \in S \wedge (x+2, y+1) \in S]$

Prove that $(\forall (x,y) \in T)[x \leq 2y]$

24. Show that every Non-Empty Full Binary Tree T satisfies

(a) $l(T) = I(T) + 1$

(b) $EXT(T) = INT(T) + N(T) - 1$

where

- $l(T)$ = the number of leaf nodes (nodes without children) in T
- $I(T)$ = the number of internal nodes (nodes with children) in T
- $EXT(T)$ = the sum of all external path lengths in T. An external path is a path from the root node to any leaf node, and the length is the number of edges in that path
- $INT(T)$ = the sum of all internal path lengths in T. An internal path is a path from the root node to any internal node

25. Show that the number of nodes in a perfect k-ary tree is $\sum_{i=0}^{h} k^i$ (which, by the geometric series $= \frac{k^{h+1}-1}{k-1}$) where $h =$ the height of the tree, and k is the branching factor of the tree. Define height as the maximal external path length. Define a perfect k-ary tree as follows:
    - $h = 0$: a single root node
    - $h > 0$: a root node with exactly k children (each of which is a perfect k-ary tree of height $h - 1$
26. Prove that $(\forall n \in \mathbb{N}^{\geq n_0})[2^n > \frac{1}{24}(n^4 - 6n^3 + 23n^2 - 18n + 24)]$, then use your proof to find $n_0 \in \mathbb{N}$
27. Prove $(\exists a, b \in \mathbb{N})(\forall n \in \mathbb{N})[7 \mid a^n + b]$
28. Find constants $a, b, c, d, e \in \mathbb{R}$ such that
$$(\forall n \in \mathbb{N})[\sum_{i=0}^{n} i^3 = an^4 + bn^3 + cn^2 + dn + e]$$
29. Prove that $t_n \leq n \log_b n$ where
$$t_n = \begin{cases} 0, & n = 0 \\ 2t_{\lfloor \frac{n}{2} \rfloor} + n - 1, & n > 0 \end{cases}$$
you will need to select a base $b \in \mathbb{Z}^+$ during the step that makes the induction easy
30. Find the lower bound for the amount of dollars we can make using strictly 5 and 6 dollar bills. Do this with constructive induction. Verify your result is correct by proving it with both weak and strong induction.
31. Repeat the previous problem with strictly 3, 7, and 15 dollar bills.

Chapter 4

Combinatorics and Probability

What?!

Lil Jon

Contents

4.1 Introduction

Combinatorics and probability offer us both a way of thinking and a handful of tools to solve interesting problems. These topics are more real-world focused since they have more direct and obvious applications. The combinatorics mindset can be difficult for some students – hang in there, you can do it.

This section discusses combinatorics, or *counting arguments*, followed by probability theory and basic statistics. The two topics are related, as you will see later. Probability theory includes discussions on *continuous* probability, however we are exclusively concerned with *discrete* probability (re: the name of this course).

4.2 Combinatorics

Imagine you are a software developer for a large company, *Macrosoft*. Your manager assigned you to a project that needs testing. Your test cases must cover all execution paths. At first this seems like a daunting task, however you took discrete mathematics and learned about combinatorics. You can calculate exactly how many test cases you need. And so you do – only to find out that you need $9! = 362880$ test cases. In the fine words of JonTron, "that's a lot of damage."

Definition 4.2.1 Combinatorics
The area of mathematics dealing with counting

Example: Count the number of elements in the following set:

$$\{\emptyset, \{a\}, \{b\}, \{a, b\}\}$$

Solution: There are two ways to solve this problem. First, you can visually count the number of elements – 1, 2, 3, 4. Otherwise, you can notice that the given set is the powerset of $\{a, b\}$, which is a set of size 2. We know that $|\mathcal{P}(A)| = 2^{|A|}$, so the number of elements is $2^2 = 4$.

Remark 4.2.2
There are almost always multiple valid ways to solve combinatorics problems. Some ways are better than others, though, because they can abstract out to larger inputs.

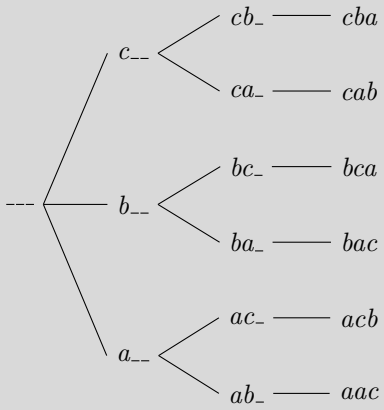
Example: How many different orderings of the letters *abc* can we create?

Solution: We can list all the possible orderings and count them: *abc*, *acb*, *bac*, *bca*, *cab*, *cba*.
Alternatively, we can view this as a 3-letter word where we need to place letters in each of the three positions.

For the first position, we have three possibilities: *a*, *b*, or *c*. Then, we place that letter and move on to the second position where we can place one of the two remaining letters. In the last position, we place the remaining letter.

So, *for each* possibility in the first position, we have all possibilities for the second and third positions. So there are $3\times$ *second/third position possibilities* total orderings. Then, journey into the second and third positions. There are two remaining letters, so there are $2\times$ *third position possibilities* sub-orderings. There is one option in the last position, so putting this all together we get $3 \times 2 \times 1 = 3! = 6$ total orderings.

Another way to view this problem is as a decision tree. For each step in the tree, we decide which letter to place in the permuted string. Then we need to count the number of leaf nodes in the tree.



Definition 4.2.3 Multiplication Rule
Given some procedure E that 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 those ways there are n_2 ways to do the second task, then the total number of ways to do procedure E is $n_1 \times n_2$

Remark 4.2.4
This is a highly technical definition that just abstracts the procedure we did the previous example.

Definition 4.2.5 Permutations
The number of **ordered** arrangements of *all* elements from a set of size n. By the multiplication rule, the total amount of permutations equals

$$n!$$

Remark 4.2.6
$0! = 1$ – come back to this later and see if you can make sense of this combinatorially!

Example: How many permutations can we make of the string "rock"?

Solution: By definition, there are $4! = 24$ permutations. If you do not believe me, list them out! In fact, let's do that right now.

rock, *rokc*, *rcok*, *rcko*, *rkoc*, *rkco*,
orck, *orkc*, *ocrk*, *ockr*, *okrc*, *okcr*,
crok, *crko*, *cork*, *cokr*, *ckro*, *ckor*,
kroc, *krco*, *korc*, *kocr*, *kcro*, *kcor*

Feel free to check this against the multiplication rule!

Remark 4.2.7
In larger examples, we will have analogously large answers. For example, the string "hardest" has $7! = 5040$ permutations. *You do not have time to list out that many permutations.*

This is not the whole story of permutations. What happens when we have repeat objects in our set (which, would not be a set by definition, but bear with us)?

Example: How many permutations can we make of the string "rook"?

Solution: Let's list them out, the same way as we did before with "rock".

rook, *roko*, *rook*, *roko*, *rkoo*, *rkoo*,
orok, *orko*, *oork*, *ookr*, *okro*, *okor*,
orok, *orko*, *oork*, *ookr*, *okro*, *okor*,
kroo, *kroo*, *koro*, *koor*, *koro*, *koor*

Notice that the first row and last row each contain three, instead of six, unique strings – half the amount. Further, the second and fourth row are duplicates of each other. Now, we will not count *roko* and *roko* (seen as ro_1ko_2 and ro_2ko_1) as different strings. So there are $\frac{4!}{2} = 12$ permutations of "rook".

Remark 4.2.8
Notice that we have two instances of the letter 'o'. In the original string, we can order the 'o' letters in two ways – o_1 comes first in the string, and o_2 second, or o_2 comes first and o_1 second. This ordering is what we are dividing out, since each of these different orderings of o_1 and o_2 yields the same string.

Example: Now consider the string "ooopp". How many permutations can we make of this string?

Solution: Recall the idea of dividing out the number of orderings of each repeated letter. We group those duplicate strings together, and count that as one unique permutation. We have 3 'o' letters, and 2 'p' letters. There are 5 total letters, so we have $\frac{5!}{3!2!}$ permutations.
Let's examine one uniquely defined permutation, "opopo". We know that op_1op_2o is the same as op_2op_1o. But now we have three instances of 'o'. For each of the previous orderings, we have $o_1po_2po_3$, $o_1po_3po_2$, $o_2po_1po_3$, $o_2po_3po_1$, $o_3po_1po_2$, $o_3po_2po_1$. So by the multiplication rule, we have $2 \times 6 = 2! \times 3!$ different "orderings" of the same word. So for each permutation of "ooopp", we can group $2! \times 3! = 12$ of those permutations into the same string. So we divide out this amount from the entire permutation.

Definition 4.2.9 r-Permutation
Ordered arrangements of r elements from a set of size n. The total amount of r-permutations is denoted, and equals,

$$P(n, r) = \frac{n!}{(n-r)!}$$

Why does this formula work? Well, use our previous definition of a permutation to permute a set of size n. Then, consider our r elements as unique letters in a string, and consider the other $n - r$ elements as the same letter. Then use the previous idea to conclude that there are $\frac{n!}{(n-r)!}$ permutations.

Remark 4.2.10
In the previous paragraph, we gave a *combinatorial argument*.

Example: We are a combinatorics photographer. A group of 7 students walked into our office asking for all different possible pictures of them with only 3 people. Since we charge per-photo, how many such photos will we take?

Solution: We *could* count out all of the options, but that might be too much work. Instead, we can translate the problem into a string problem like before. We want all possibilities of 3 people in a row:

In the first position, we have 7 possibilities. Then, in the second position, we have 6 possibilities. Finally, in the third position, we have 5 possibilities. Thus, in total, we can make $7 \times 6 \times 5$ photos.
Alternatively, we could have used r-permutations. In this case, $r = 3$ and $n = 7$. So we are finding all 3-permutations of a set of size 7. This is

$$P(7,3) = \frac{7!}{(7-3)!} = \frac{7!}{4!} = 7 \times 6 \times 5$$

Definition 4.2.11 r-Combinations
(or just *Combinations*) **Unordered** arrangements of r elements from a set of size n. Alternatively, it is the number of r-sized subsets of a n-sized set. The total amount of combinations is denoted and equals

$$C(n,r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Remark 4.2.12
$\binom{n}{r}$ is often called a *binomial coefficient*.

Example: How many student committees of size 3 can we make from a student population of 7?

Solution: There is no notion of *order* within a committee, so we want the number of 3-combinations from a set of size 7. So the answer is

$$\binom{7}{3} = \frac{7!}{3!4!}$$

Alternatively, we could first solve the problem as-if we care about the committee ordering, and then *divide out* all of the different possible orderings per committee. For example, the ordered committees a, b, c and b, c, a should be counted as the same. So, how many orderings *per-committee* can we make? 3!. Then, the total amount of ordered committees is $P(7,3) = \frac{7!}{4!}$. So if we divide out 3! we get

$$\frac{P(7,3)}{3!} = \frac{7!}{3!4!} = \binom{7}{3}$$

Remark 4.2.13

The previous example shows the relationship between permutations and combinations by showing an easy-to-abstract combinatorial argument.

There are slight differences between the notion of *with replacement* and *without replacement*. In the former, one would sample a set, record their observation, and put their sample back and keep going. In the latter, one would sample a set, record their observation, **not** put the sample back, and continue. Keeping this difference in mind will help you tremendously in choosing the correct counting technique to solve a given problem.

Example:

1. How many passwords of length 10 can we create if we can only use lowercase English characters?
2. What if we are not allowed to use the same character more than once?

Solution: In the first problem, we can write out 10 slots with 26 options per slot. The multiplication rule tells us we have 26^{10} possible passwords. In the second problem, we cannot re-use characters. So we choose the first character (26 options), then choose the second character (25 options – we cannot use the first chosen character again!), and keep going. This gives us $26 \times 25 \times \cdots \times 17 = \frac{26!}{(26-10)!}$ possible passwords.

In the first problem, we are allowed to *re-use* each character – we sample characters *with replacement*. In the second, we are not – *without replacement*.

	With Replacement	Without Replacement
Order Matters	n^k	$P(n,k) = \frac{n!}{(n-k)!}$
Order Does Not Matter	$\binom{n+k-1}{n-1}$	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$

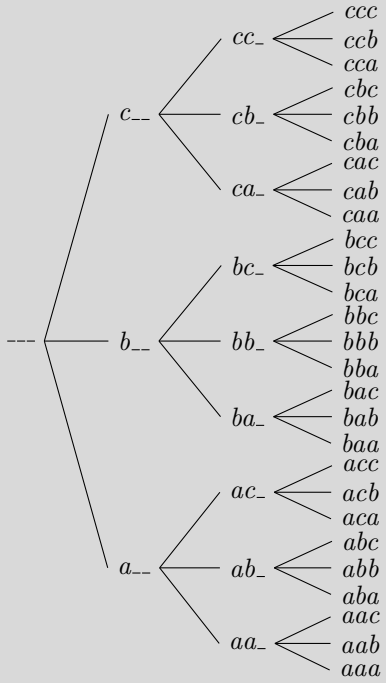
Figure 4.1: Ordering and Replacement – picking k elements from n total elements

Remark 4.2.14
For explanation on unordered sampling with replacement, see here.

Example: How many strings of length 3 can we make using the characters $\Sigma = \{a, b, c\}$?

Solution: We are counting elements with replacement where order matters. Our main set $n = 3$ is the number of character choices, and our subset $k = 3$ is the string length. By the table, we can make $3^3 = 27$ strings.

Another viable solution to this problem is to use a decision tree, where each branch represents placing one of the three characters in the string.



Definition 4.2.15 Combinatorial Argument
A counting proof that shows a cominatorial identity is true. The most common approach is to show two seemingly different quantities *count* the same thing.

Remark 4.2.16
There exist two forms of combinatorial argument.

1. Double counting method
2. Bijective method

We will focus exclusively on the first.

One can show that $X = Y$ via combinatorial argument using the following steps,

1. Find/create a counting problem that can be solved in at least two ways
2. Show that we can count the problem using quantity X
3. Show that we can count the problem using quantity Y
4. Conclude that, since both ways count same problem, the quantities X and Y are equal

Example: Use a cominatorial argument to show

$$2^n = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{i=0}^{n} \binom{n}{i}$$

Proof. Consider the problem of counting the number of bit-strings of length n.
We can solve this problem one way by writing out n slots and for each slot choosing one of two options: $\{0, 1\}$. The multiplication rule tells us this yields 2^n bit-strings.
We can also solve this problem by counting the number of bit-strings that contain zero 1s, one 1, two 1s, three 1s, all the way to n 1s. This indeed counts all of the bit-strings. There are $\binom{n}{0}$ bit-strings of length n with zero 1s. There are $\binom{n}{1}$ bit-strings of length n with one 1. Et cetera, there are $\binom{n}{n}$ bit-strings of length n with n 1s. The denominator in each binomial represents the number of slots in the string we are selecting to be a 1. Each of these groups are disjoint, so the total number of bit-strings is their sum: $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n}$.
Both counting techniques solve the same problem, so they must be equivalent. Thus $2^n = \sum_{i=0}^{n} \binom{n}{i}$. □

Definition 4.2.17 Algebraic Argument
A proof that argues the correctness of (typically) a combinatorial identity using algebra.

Example: Show that

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Proof.

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

□

The previous example can be shown using a combinatorial argument, however the reader should try this on their own. Usually both argument techniques can show a counting-type identity, however often one technique is better than the other.

Example: Explain why $\binom{n}{0} = \binom{n}{n} = 1$.

Proof. We will first show a combinatorial argument, followed by an algebraic argument.
Consider the *meaning* of the binomial coefficient. $\binom{n}{0}$ means we are choosing 0 elements out of a set of size n. How many ways can we do this? 1 way – we just pick nothing and leave. Now consider $\binom{n}{n}$, choosing n elements out of a set of size n. How many ways can we do this? 1 way – pull out the entire set (unordered!) and leave. Notice that both outcomes are the same, so they are equal.
Alternatively, consider that $\binom{n}{0} = \frac{n!}{0!(n-0)!} = \frac{n!}{n!} = 1 = \frac{n!}{n!} = \frac{n!}{n!0!} = \frac{n!}{n!(n-n)!} = \binom{n}{n}$ □

Theorem 4.2.1 The Binomial Theorem

For $n \in \mathbb{N}$,

$$(x+y)^n = \binom{n}{0}x^n + \binom{n}{1}n^{n-1}y + \binom{n}{2}x^{n-2}y^2 + \cdots + \binom{n}{n-1}xy^{n-1} + \binom{n}{n}y^n$$
$$= \sum_{i=0}^{n}\binom{n}{i}x^{n-i}y^i$$

Proof. We offer a quick proof by induction on n. Consider $n = 0$, then $(x+y)^0 = 1$, and $\sum_{i=0}^{0}\binom{n}{i}x^{n-i}y^i = \binom{0}{0}x^0y^0 = 1$, so they are the same. Let our hypothesis be for $n > 0$, $(x+y)^{n-1} = \sum_{i=0}^{n-1}\binom{n-1}{i}x^{n-1-i}y^i$. The step is a bit tricky, so hold on tight. Notice when we use our identity $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$.

$$\begin{aligned}
\sum_{i=0}^{n}\binom{n}{i}x^{n-i}y^i &= \binom{n}{0}x^n + \sum_{i=1}^{n-1}\binom{n}{i}x^{n-i}y^i + \binom{n}{n}y^n \\
&= x^n + (\binom{n}{1}x^{n-1}y + \cdots + \binom{n}{n-1}xy^{n-1}) + y^n \\
&= x^n + ((\binom{n-1}{0} + \binom{n-1}{1})x^{n-1}y + \\
&\cdots + (\binom{n-1}{n-2} + \binom{n-1}{n-1})xy^{n-1}) + y^n \\
&= x^n + ((\binom{n-1}{0}x^{n-1}y + \binom{n-1}{1}x^{n-1}y) + \\
&\cdots + (\binom{n-1}{n-2}xy^{n-1} + \binom{n-1}{n-1}xy^{n-1})) + y^n \\
&= x^n + (\sum_{i=0}^{n-2}\binom{n-1}{i}x^{n-1-i}y^{i+1}) \\
&+ (\sum_{i=1}^{n-1}\binom{n-1}{i}x^{n-1-i+1}y^i) + y^n \\
&= x^n + y(\sum_{i=0}^{n-2}\binom{n-1}{i}x^{n-1-i}y^i) \\
&+ x(\sum_{i=1}^{n-1}\binom{n-1}{i}x^{n-1-i}y^i) + y^n \\
&\overset{\text{IH}}{=} x^n + y((x+y)^{n-1} - \binom{n-1}{n-1}y^{n-1}) \\
&+ x((x+y)^{n-1} - \binom{n-1}{0}x^{n-1}) + y^n
\end{aligned}$$

$$
\begin{aligned}
&= x^n + y(x+y)^{n-1} - y^n + x(x+y)^{n-1} - x^n + y^n \\
&= y(x+y)^{n-1} + x(x+y)^{n-1} \\
&= (y+x)(x+y)^{n-1} = (x+y)(x+y)^{n-1} \\
&= (x+y)^n
\end{aligned}
$$

□

Remark 4.2.18
With this theorem, we can prove that $2^n = \sum_{i=0}^{n} \binom{n}{i}$. *Hint*: set explicit values for x and y.

Example: Expand $(x-2)^3$.

Solution: $(x-2)^3 = (x+(-2))^3 = \binom{3}{0}x^3(-2)^0 + \binom{3}{1}x^2(-2)^1 + \binom{3}{2}x^1(-2)^2 + \binom{3}{3}x^0(-2)^3 = x^3 - 6x^2 + 12x - 8$

The Binomial Theorem is helpful in computing the coefficients in a binomial expansion, as seen above. However, remembering the formula may be a hassle. This is where the idea of *Pascal's Triangle* comes into play. We will use the previously-shown identity $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ to help us explain the triangle.

Definition 4.2.19 Pascal's Triangle
Construct a discrete triangle of numbers as follows:

- The zero$^{\text{th}}$ row is the single number 1
- The n^{th} row consists of n numbers, where the first and last numbers are a 1, and every in-between number is the sum of the two adjacent numbers in the row above it

The first 5 rows look like this:

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
```

A keen eye will notice that each number can be expressed as a binomial coefficient dependent on its row and column position:

$$\begin{array}{ccccccccccc} & & & & \binom{0}{0} & & & & & & \\ & & & \binom{1}{0} & & \binom{1}{1} & & & & & \\ & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & & \\ & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & & \\ \binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & & \end{array}$$

This pattern is explained by our identity $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$. Think about what this identity says – the current binomial coefficient is the sum of two adjacent binomial coefficients in the row above it. To see this, write Pascal's Triangle as a triangular matrix, let the rows be n and columns be k, and let the first column be $\binom{n}{0} = 1$ and the diagonal be $\binom{n}{n} = 1$:

$$\begin{array}{ccccccc} \binom{0}{0} & & & & & & 1 \\ \binom{1}{0} & \binom{1}{1} & & & & & 1 \quad 1 \\ \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & & = & 1 \quad 2 \quad 1 \\ \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & & & 1 \quad 3 \quad 3 \quad 1 \\ \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} & & 1 \quad 4 \quad 6 \quad 4 \quad 1 \end{array}$$

Theorem 4.2.2 The Multinomial Theorem

$$(x_1 + x_2 + \cdots + x_k)^n = \sum_{\substack{i_1, i_2, \cdots, i_k \geq 0 \\ i_1 + i_2 + \cdots + i_k = n}}^{n} \frac{n!}{i_1! i_2! \cdots i_k!} x_1^{i_1} x_2^{i_2} \cdots x_k^{i_k}$$

We omit the proof on this one, however it is essentially a generalization of the proof for the Binomial Theorem.

Remark 4.2.20
You might see $\frac{n!}{i_1! i_2! \cdots i_k!}$ written as $\binom{n}{i_1, i_2, \cdots, i_k}$. The latter is called the *multinomial coefficient*, and is equal to the former.

Example: Expand $(x - 1 + y)^3$.

Solution: Our exponent possibilities are $(0,0,3), (0,3,0), (3,0,0), (1,2,0), (1,0,2), (2,1,0), (0,1,2), (0,2,1), (2,0,1), (1,1,1)$. Thus
$(x - 1 + y)^3 = \frac{3!}{0!0!3!} x^0 (-1)^0 y^3 + \frac{3!}{0!3!0!} x^0 (-1)^3 y^0 + \frac{3!}{3!0!0!} x^3 (-1)^0 y^0 + \frac{3!}{1!2!0!} x^1 (-1)^2 y^0 + \frac{3!}{1!0!2!} x^1 (-1)^0 y^2 + \frac{3!}{2!1!0!} x^2 (-1)^1 y^0 + \frac{3!}{0!1!2!} x^0 (-1)^1 y^2 + \frac{3!}{0!2!1!} x^0 (-1)^2 y^1 + \frac{3!}{2!0!1!} x^2 (-1)^0 y^1 + \frac{3!}{1!1!1!} x^1 (-1)^1 y^1$
$= y^3 - 1 + x^3 + 3x + 3xy^2 - 3x^2 - 3y^2 + 3y + 3x^2 y - 6xy$

Example: Calculate the coefficient of the term $x^4y^{1000}z^{200}w^{90}$ in the expansion of $(x-y+2z-2w^3)^{1234}$.

Solution: The coefficient is $\frac{1234!}{4!1000!200!30!}\cdot 1^4\cdot(-1)^{1000}\cdot 2^{200}\cdot(-2)^{30}$. This follows from the Multinomial Theorem. Note that $w^{90}=(w^3)^{20}$. We set $(x_1,x_2,x_3,x_4)=(x,-y,2z,-2w^3)$ and $(i_1,i_2,i_3,i_4)=(4,1000,200,30)$. Then the term in the expansion is $\frac{1234!}{4!1000!200!30!}\cdot x^4\cdot(-y)^{1000}\cdot(2z)^{200}\cdot(-2w^3)^{30}$. By properties of powers, we pull the constant terms *out* and yield $\frac{1234!}{4!1000!200!30!}(1)^4(-1)^{1000}(2)^{200}(-2)^{30}\cdot x^4y^{1000}z^{200}w^{90}$.

Now, before we move on, we should have a quick discussion as to why $\binom{n}{k}$ is an integer. It is very tempting to say that, since we can always construct a situation in which $\binom{n}{k}$ *counts something*, then it is an integer. Indeed, this is a valid *combinatorial argument*. $\binom{n}{k}$ counts exactly the number of possible k-subsets from a set of size n. Can we show this a different way? Well, an algebraic argument might not lead anywhere, but an inductive proof may.

Example: Show that $\binom{n}{k}\in\mathbb{Z}$ for $n,k\in\mathbb{N}$ and $n\geq k$.

Proof. First, let us show the following lemma:

$$(\forall m\in\mathbb{N})(\forall o\in\mathbb{N})[m!\mid\prod_{i=o}^{o+m-1}i]$$

All this says is that $m!$ divides the product of m consecutive integers. We use o as an *offset*. We will show this with a *double induction* proof on m and o. Note that o could be any integer, but we will restrict to naturals for simplicity.

We start by inducting on m. For $m=0$ then we have $0!=1$ and 1 divides everything. So the base case holds.

For a hypothesis, assume $(m-1)!\mid\prod_{i=o}^{o+m-2}i$ for an arbitrary $m-1\geq 0$.

For the inductive step, we aim to show $m!\mid\prod_{i=o}^{o+m-1}i$. We will now show this by inducting on o.

Note that if $o=0$ then the product is 0, and anything divides 0. We do not have to, but we will also consider $o=1$. In this case, $\prod_{i=1}^{1+m-1}i=m!$, and $m!\mid m!$. So the base case holds.

For a hypothesis, assume $m!\mid\prod_{i=o-1}^{o-1+m-1}i$ for an arbitrary $o-1\geq 0$.

We aim to show that $m!\mid\prod_{i=o}^{o+m-1}i=(o)(o+1)\cdots(o+m-1)$. Well $(o)(o+1)\cdots(o+m-1)=(o-1)(o)\cdots(o-1+m-1)+(o)(o+1)\cdots(o+m-2)\big((o+m-1)-(o-1)\big)=\prod_{i=o-1}^{o-1+m-1}i+m\prod_{i=o}^{o+m-2}i$. The first term in the sum is divisible by $m!$ by the induction hypothesis for o. The

second term $\prod_{i=o}^{o+m-2} i$ is divisible by $(m-1)!$ by the induction hypothesis for m. Then $(m-1)! \mid \prod_{i=o}^{o+m-2} i \Rightarrow m(m-1)! \mid m\prod_{i=o}^{o+m-2} i$. Thus our entire original sum is divisible by $m!$.
So the induction step for o holds, and thus the induction step for m also holds.
Then $\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n\cdot(n-1)\cdots(n-k+1)}{k!}$. Then the numerator is a product of $n-(n-k+1)+1 = k-1+1 = k$ consecutive integers, so by the previous lemma $\binom{n}{k} = \frac{n\cdot(n-1)\cdots(n-k+1)}{k!} \in \mathbb{Z}$. □

4.3 The Inclusion-Exclusion Principle

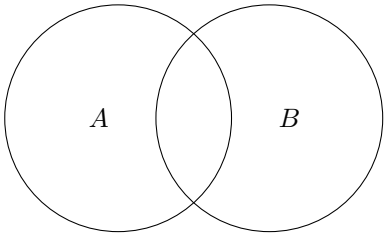
Combinatorics deals with counting things. Often we care about counting sizes of sets. Sometimes even when sets overlap, as in a set intersection. The Inclusion-Exclusion Principle helps us tackle situations when we know *almost* all information about two sets.

Example: As a motivating example, let's suppose we wish to count the number of binary strings of length 3 *starting with a 0* **or** *ending with a 1.* A naive approach might be to first calculate the number of strings that start with a 0, and add to that the number of strings that end with a 1.

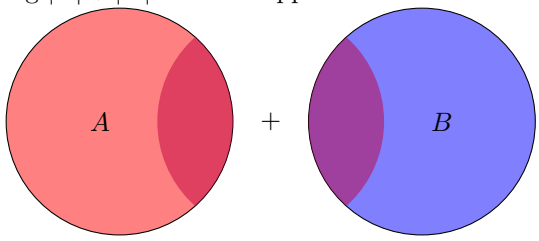
Start with 0: 000, 001, 010, 011 Start with 1: 100, 101, 110, 111

If we do this, then we would yield the solution $4+4=8$. Yet, indeed *not all length-3 binary strings start with a 0 or end with a 1.* For example, 100 and 110 satisfy neither condition. Indeed, the correct answer is $4+4-2=6$. In our naive approach, we forgot to account for the *overlap* between our two cases.

Example: Suppose we have 30 students in a class. 10 of them received an A on the first assignment, and 15 of them received an A on the second assignment. 13 students received an A on *both* assignments. How many students have received *at least one* A?
We can visualize this by drawing a Venn-diagram.



We have $A =$ the students who received an A on the first assignment, and $B =$ the students who received an A on the second. Then $|A| = 10$, $|B| = 15$, and their intersection $|A \cap B| = 13$. But we are interested in the union – how can we calculate it?
Well, from the Venn-diagram, we want to find the size of the entire diagram shaded in. How can we use what we know to get *close* to this? Let's try adding $|A| + |B|$ – what happens?



Since $A \cap B \subseteq A$ and $A \cap B \subseteq B$, then when we add $|A| + |B|$ we get the entire shaded Venn-diagram $A \cup B$! ... plus an extra copy of the middle bit $A \cap B$ ....
So just subtract off that extra copy! Then we get

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Now plug in and find that the number of students who received an A is

$$10 + 15 - 13 = 12$$

The Inclusion-Exclusion Principle is precisely a generalization of this exact situation. The idea is that we *include* some sets, while we *exclude* some other sets.

Theorem 4.3.1 The Inclusion-Exclusion Principle
The size of the union of n sets equals

1. *the inclusion of the sizes of every set*
2. *the exclusion of the sizes of every pairwise set intersection*
3. *the inclusion of the sizes of every triple-wise set intersection*
4. *and so on*

More abstractly, for the sets $A_1, A_2, \cdots, A_n$ we have

$$|\bigcup_{i=1}^{n} A_i| =$$

$$\sum_{1\leq i\leq n} |A_i| - \sum_{1\leq i,j\leq n} |A_i \cap A_j| + \sum_{1\leq i,j,k\leq n} |A_i \cap A_j \cap A_k|$$
$$- \cdots + (-1)^{n-1}|A_1 \cap \cdots \cap A_n|$$

Remark 4.3.1
The proof of this theorem is essentially the 2-set case but generalized.

Example: Calculate the number of students who received *below* an A in all of their first three assignments, given the following information:

- There are 30 students in the class
- 8 received an A on assignment 1
- 8 received an A on assignment 2
- 8 received an A on assignment 3
- 6 received an A on assignment 1 and 2
- 2 received an A on assignment 1 and 3
- 4 received an A on assignment 2 and 3
- 1 received an A on all three assignments

Solution: We apply The Inclusion-Exclusion Principle to first calculate the number of people who received an A in any of the first three assignments.

$$\begin{aligned} |A_1 \cup A_2 \cup A_3| &= |A_1| + |A_2| + |A_3| \\ &\quad - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3| \\ &\quad + |A_1 \cap A_2 \cap A_3| \\ &= 8 + 8 + 8 - 6 - 2 - 4 + 1 = 13 \end{aligned}$$

Then the number of people who *did not* get an A is just the total number of students minus those who received an A. $|(A_1 \cup A_2 \cup A_3)^c| = 30 - 13 = 17$

4.4 Pigeonhole Principle

Let us imagine a situation where we have 6 books and 4 bookshelves. Try placing the books on the bookshelves such that each bookshelf only has one book.

Go ahead, I'll wait.

You can try and try and try, but to no avail. You cannot arrange the books on the shelves such that each shelf contains only one book! Or, equivalently, you know that at least one bookshelf holds more than one book. We formalize this idea as the Pigeonhole Principle – the books become pigeons and the bookshelves become pigeonholes.

Theorem 4.4.1 The Pigeonhole Principle
If n pigeons are placed into k pigeonholes, and $n > k$, then at least one pigeonhole contains more than one pigeon.

Remark 4.4.1
This theorem is somewhat intuitive. Start by placing one pigeon into each hole, then you will have some leftover pigeons. Specifically, you are left with $n - k$ pigeons. Since $n > k$ then $n - k > 0$. So your leftover pigeons have to go somewhere, but all pigeonholes already have a pigeon!

Proof. By contraposition, let all k containers hold ≤ 1 pigeon. Then we add up the total number of pigeons. Denote H_i to be the number of pigeons in pigeonhole i. The total number of pigeons is $n = \sum_{i=1}^{k} H_i$. But $H_i \leq 1$, so $n = \sum_{i=1}^{k} H_i \leq \sum_{i=1}^{k} 1 = (k - 1 + 1) = k$. So $n \leq k$. □

Example: Your birth-month is the month in which you were born. There are 12 birth-months. How many people do we need to gather before we are guaranteed that two of them share a birth-month?

Solution: We use the Pigeonhole Principle. The pigeons are the people we need to gather, and the pigeonholes are the birth-months. Then we need the number of people $n > 12$. So we need at least 13 people.

Theorem 4.4.2 Generalized Pigeonhole Principle
If n objects are placed into k boxes, then there is at least one box containing at least $\lceil \frac{n}{k} \rceil$ objects.

Proof. By contradiction, assume all k boxes contain $< \lceil \frac{n}{k} \rceil$ objects. This is the same as saying that all boxes contain $\leq \lceil \frac{n}{k} \rceil - 1$ objects. Denote the number of objects in each box B_i. Then the total number of objects is $n = \sum_{i=1}^{k} B_i \leq \sum_{i=1}^{k} (\lceil \frac{n}{k} \rceil - 1) = (\lceil \frac{n}{k} \rceil - 1)(k - 1 + 1) = k(\lceil \frac{n}{k} \rceil - 1) < k(\frac{n}{k} + 1 - 1) = n$. So $n < n$, which is a contradiction. □

Remark 4.4.2
Suppose you place n objects into k boxes one-by-one, wrapping around to the first box once you reach the end. Then if $n > k$ we must have that the first box contains more than $\frac{n}{k}$ objects – all we have done is grouped the objects into k groups. Then the ceiling $\lceil \frac{n}{k} \rceil$ corresponds to that *plus one* as in the original Pigeonhole Principle.

Example: Suppose we have a standard deck of cards – 13 ranks, 4 suits, which yields $13 \times 4 = 52$ cards. How many cards must we draw to guarantee that our hand contains at least three cards of all the same suit?

Solution: Intuitively, we can try to "minimize" the number of like-suit cards we draw. There are four suits, so first pick 4 cards each with a different suit. Then do it again. Then the suit of the next card will already have been chosen twice, which means we will have three cards of the same suit. This is $2 \times 4 + 1 = 9$ cards.
This is not rigorous though! It might make intuitive sense, but we should apply our proven theorems. In this case, we will use the Generalized Pigeonhole Principle. Let the number of cards we draw be the objects, and the suits be the boxes. Then we need one of our boxes to contain at least 3 objects. So that box contains at least $\lceil \frac{n}{4} \rceil = 3$ objects. Then the smallest n that satisfies this equation is $n = 9$, because $\lceil \frac{9}{4} \rceil = \lceil 2.25 \rceil = 3$ but $\lceil \frac{8}{4} \rceil = \lceil 2 \rceil = 2 \neq 3$.

Example: Assume the same deck of cards as before. How many cards must we draw to guarantee that our hand contains cards from all four suits?

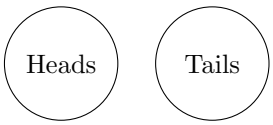
Solution: Intuitively, we can try to "minimize" the number of suits we draw, until we are forced to draw the final fourth suit. This means we would draw every card from the first three suits, and then the next card we pick must be from the fourth suit. This yields $13 \times 3 + 1 = 40$ cards.
Formally, as in the Generalized Pigeonhole Principle, let the objects be the number of cards we draw, and the holes be the *ranks*. Then we need one of our boxes to contain at least four objects (each a different suit of that box's rank). So that box contains at least $\lceil \frac{n}{13} \rceil = 4$ objects. Then the smallest n that satisfies this equation is $n = 40$, because $\lceil \frac{40}{13} \rceil = \lceil 3.0769 \cdots \rceil = 4$ but $\lceil \frac{39}{13} \rceil = \lceil 3 \rceil = 3 \neq 4$.

4.5 Discrete Probability

We have discussed techniques for counting. We can apply this in a statistical sense when we start counting *events*. Often when we discuss probability, we care about some smaller portion, or subset, of the total amount of possible events.

An easy way to introduce probability is by working through a well-known example – flipping a coin. Assuming the coin is fair, and coin flips are truly random, then we can start to reason about, given any coin flip, whether the coin will land heads-up or tails-up. Later we will see that we do not need these assumptions to reason about which side lands face-up, but we will assume them now for simplicity.

Suppose we flip a fair and truly random coin. We know that there are only two possible outcomes – the coin lands heads-up, or the coin lands tails-up. Since the coin is truly *random*, we have no way to predict which side will land face-up. Since the coin is *fair*, then both sides are equally likely to land face-up. We refer to this as a 50/50 chance – a 50% chance of landing tails, and a 50% chance of landing heads. Note that $50\% = 0.5 = \frac{1}{2}$, which is the likelihood that any *one* of our *two* (heads or tails) events occurs.



A subtle note on **percents** – the % symbol. A *percent* is a way to represent *proportions*, or fractions, of a whole. In percent-land, we are concerned with fractions of a single whole, 1. All percents represent numbers in the continuous interval $[0, 1]$. To translate a number $0 \leq r \leq 1$ into a percent, multiply the number by 100 and append a % on the end. So $r \mapsto 100r\%$. For example, $\frac{1}{4} = 0.25 = 25\%$. Usually percents are approximations for their real-number counterpart – as such, percents necessarily approximate irrational numbers and any fraction with an infinite decimal expansion. For example, $\frac{1}{9} = 0.\bar{1} = 0.111111\cdots \approx 11.11\%$. The cutoff point, and whether you round the result, is dependent on the situation – research *significant figures* if you are interested, but this is irrelevant for this course.

Definition 4.5.1 Probability
The likelihood (percent chance) that an event occurs.

Example: The probability of flipping tails on a fair 2-sided coin is $\frac{1}{2}$.

Example: The probability of rolling a 2 on a fair 6-sided dice is $\frac{1}{6}$.

The fractions in a probability are formulated as:

$$\frac{\#\text{ events wanted}}{\#\text{ events possible}}$$

Our previous examples had single events in the numerator, however this fraction lets us extend this to different combinations of events. The numerator and denominator are *counts*, so we can apply any of our previously learned combinatoric methods as well.

Example: What is the probability of rolling a 2 or a 3 on a 6-sided die?

Solution: We cannot roll a 2 and a 3 at the same time, so we can add their individual probabilities. So the probability is $\frac{1}{6} + \frac{1}{6} = \frac{1}{3}$.
Alternatively, there are two total different events we care about, out of 6 possible events, so the probability is $\frac{2}{6}$.

Example: We aim to select a group of size 3, by choosing one-by-one without replacement, from a set of objects $\{A, B, C, D, E, F, G\}$. What is the probability that our selected group is $\{A, C, E\}$?

Solution: Well, how many possible ways can we select exactly those three objects? Note here that the order in which we select the objects does not matter, so long as we end up with $\{A, C, E\}$. We can translate this to a string problem, selecting letters from $\{A, C, E\}$ and placing into a length-3 string. This gives us $3! = 6$ possible selections. Then the i^{th} letter in the string is the i^{th} object we select. Then how many total possible groups can we select? Well, this is equivalent to asking the amount of possible length-3 strings we can make from our original set, or the amount of 3-permutations. This equals $P(7, 3) = \frac{7!}{4!} = 7 \cdot 6 \cdot 5$. All together, this gives a probability of $\frac{\frac{3!}{7!}}{\frac{7!}{4!}} = \frac{3!4!}{7!} = \frac{3 \cdot 2 \cdot 1}{7 \cdot 6 \cdot 5} = \frac{1}{35}$.
Alternatively, we could have treated groups as one meaningful package, and examined how many total groups we can make. We can make $\binom{7}{3}$ groups, one of which is $\{A, C, E\}$. Then the probability is $\frac{1}{\binom{7}{3}} = \frac{3!4!}{7!}$ which is the same as before.

Now we go back to an earlier example.

Example: For a 6-sided die, calculate the following probabilities: rolling a 2 or a 3; rolling a 1, 4, or 5; rolling a 6.

Solution: The first set has probability $\frac{1}{3}$, as we calculated before. The second has probability $\frac{1+1+1}{6} = \frac{1}{2}$. The third has probability $\frac{1}{6}$.

Remark 4.5.2
Notice how $\frac{1}{3} + \frac{1}{2} + \frac{1}{6} = \frac{2}{6} + \frac{3}{6} + \frac{1}{6} = \frac{2+3+1}{6} = 1$

So why does this happen? Well, further notice from the previous example that our three different events are entirely disjoint. Finally, notice that all three events together make up all *possible* events for a 6-sided die. The only possibilities for a 6-sided die is rolling one of the six sides.

Before we formalize this idea, we should establish notation for probabilities.

Definition 4.5.3 Probability Notation
We denote

$$P(E) \in [0, 1]$$

as the probability that event E occurs.

Definition 4.5.4 Sample Space
Denote Ω as the set of all possible events/outcomes for the current situation.

Theorem 4.5.1
Let $E_1, E_2, \cdots, E_n \in \Omega$. *If* $\bigcap_{i=1}^{n} \{E_i\} = \emptyset$ *and* $\bigcup_{i=1}^{n} \{E_i\} = \Omega$ *then*

$$\sum_{i=1}^{n} P(E_i) = 1$$

This intuitively says that the probabilities of disjoint events that cover the entire sample space add to 1. This follows quite easily from the definition of probability – essentially think of our disjoint sample space as a partition of $[0, 1]$, with the lengths of the partition as the probability of the associated event. We saw this happened in the previous example.

Example: Calculate the probability of **not** rolling a 1 from a 6-sided die.

Solution: Note that the event *not rolling a 1* is equivalent to rolling a 2 or a 3 or a 4 or a 5 or a 6. Which we can calculate as having probability $\frac{5}{6}$.
Alternatively, notice how the events *roll a 1* and *not roll a 1* are disjoint. Further notice how these two events entirely make up our sample space of rolling any of the six sides. Then we can apply the previous theorem. Denote the first event as E_1 and the second E_2. We know the probability of rolling a 1, $P(E_1)$, is $\frac{1}{6}$. Then we know $P(E_1) + P(E_2) = 1$, thus $\frac{1}{6} + P(E_2) = 1$ so $P(E_2) = 1 - \frac{1}{6} = \frac{5}{6}$.

This method, subtracting 1 to find the complement probability, is very common and helpful in more difficult problems. In the previous example, we could easily count all possibilities in the complement event. In other examples, this might not be possible.

Example: Suppose we roll three 6-sided dice. Our total value from the roll is the sum of all three die. Calculate the probability that our total value is at least 5.

Solution: There are a *lot* of possible rolls that give us a total of at least 5. However, there are significantly less rolls that give us values *strictly less than* 5. We can count them. We have $\{1, 1, 1\}$, $\{2, 1, 1\}$, $\{2, 2, 1\}$, $\{3, 1, 1\}$. Then we have a total of $\frac{6^3}{3!}$ dice rolls (we divide out the 3! possible orderings). So of those dice rolls, we have four that we care about. So $P(v < 5) = \frac{4}{\frac{6^3}{3!}}$ where v is the dice roll value. Therefore, $P(v \geq 5) = 1 - P(v < 5) = 1 - \frac{4}{\frac{6^3}{3!}}$.

Now that we understand the idea of joint and disjoint events, we can discuss joint and disjoint *probability*.

Definition 4.5.5 Joint Probability
The probability of two (or more) events occurring together. We denote this as

$$P(E_1, E_2, \cdots)$$

We can discuss joint probability for events that are related to each other and for events completely unrelated to each other. This idea of *relation*, that one event may or may not *affect* another, is precisely the idea of *independence*.

Definition 4.5.6 Independence
Two events are independent if the outcome of one event does not affect the outcome of the other. More than two events are independent if all possible pairings of events are independent.

Example: The n events for flipping n coins are independent.

Example: Successive selection events of marbles from a bag, without replacement, are dependent.

Theorem 4.5.2
If E_1 and E_2 are independent then $P(E_1 \wedge E_2) = P(E_1)P(E_2)$

Remark 4.5.7
You may sometimes see $P(E_1 \wedge E_2)$ written as $P(E_1 \cap E_2)$.

This is essentially a consequence of our combinatorics multiplication rule.

Example: Calculate the probability of flipping heads twice in a row. Then calculate the probability of flipping n heads in a row.

Solution: There is a $\frac{1}{2}$ probability we flip heads. We know that the second coin flip is independent of the first, so we multiply to get $\frac{1}{2}\frac{1}{2} = \frac{1}{4}$. Similarly, for n flips we get $\underbrace{\frac{1}{2} \cdot \frac{1}{2} \cdots \frac{1}{2}}_{n \text{ times}} = \frac{1}{2^n}$.

Once events start affecting one another, however, then their probabilities act differently.

Example: Suppose we have a bag of marbles – 3 red marbles, 2 blue marbles, and 5 green marbles. Calculate the probability of selecting (without replacement) a blue marble, then a red marble, then two green marbles.

Solution: Once we select a marble, notice that the total amount of marbles change. We start with $3 + 2 + 5 = 10$ marbles. We have 2 blue marbles, so the probability of selecting a blue marble is $\frac{2}{10}$. Then, we have $3 + 1 + 5 = 9$ marbles left – there is one less blue marble. Of these

marbles, we have a $\frac{3}{9}$ probability of selecting a red marble. Then we have $2+1+5=8$ marbles left. We have a $\frac{5}{8}$ chance of selecting a green marble. Finally, we have $2+1+4=7$ marbles left, so we have a $\frac{4}{7}$ chance of selecting the second green marble.
Now that each probability calculation has accounted for our event dependencies, then we can treat them as independent and multiply to get our resulting probability. This yields $\frac{2}{10} \cdot \frac{3}{9} \cdot \frac{5}{8} \cdot \frac{4}{7} \approx 2.381\%$.

The previous example gives us motivation for a new definition.

Definition 4.5.8 Conditional Probability
The probability of one event occurring given another event occurs. We denote this as

$$P(E_2 \mid E_1)$$

which reads as *the probability that E_2 occurs given E_1 occurred.*

Example: As in the previous example, suppose we have a bag of marbles – 3 red marbles, 2 blue marbles, and 5 green marbles. Calculate the probability of selecting a red marble *given* we first selected a blue marble.

Solution: Let E_1 represent selecting a blue marble and E_2 represent selecting a red marble. Then $P(E_2 \mid E_1) = \frac{3}{9}$. In this case, we take out the already-selected blue marble, and continue as usual. There are $3+1+5=9$ total marbles, and three of those are red.

Remark 4.5.9
Compare this to the probability of selecting a blue marble and then selecting a red marble (which is $\frac{2}{10} \cdot \frac{3}{9}$).

Think of conditional probability as a re-scale. In a standard probability, we divide the size of the event by the size of the sample space. In a *conditional* probability, we change the denominator to the size of the *given* event. More formally:

Proposition 4.5.3

$$P(E_2 \mid E_1) = \frac{P(E_2 \cap E_1)}{P(E_1)}$$

Remark 4.5.10
Note that this definition does not work when $E_1 = \emptyset \Rightarrow P(E_1) = 0$ since we have a divide-by-zero. But, note that intuitively $P(E_2 \mid \emptyset) = P(E_2 \cap \emptyset) = 0$, so it still *somewhat* makes sense. The more common given is

$$P(E_2 \mid E_1)P(E_1) = P(E_2 \cap E_1)$$

Now, what if our events in a conditional probability are independent? So what is $P(E_2 \mid E_1)$ when E_1 and E_2 are independent? Well, the conditional probability focuses on when E_2 occurs *given* E_1 occurred. But E_2 is independent of E_1, so E_1 places no condition on E_2. This gives us our next theorem.

Theorem 4.5.4
If E_1 and E_2 are independent then $P(E_2 \mid E_1) = P(E_2)$

Example: Calculate the probability that we flip heads on a coin given we already flipped heads.

Solution: The probability is $\frac{1}{2}$. In the world where we already flipped heads, well the next flip we do has no dependence on our first flip. So the second flip has the same probability of a standard flip.

Example: What does disjointness tell us about independence?

Solution: For two disjoint events E_1 and E_2 by definition $E_1 \cap E_2 = \emptyset$. Then $P(E_1 \cap E_2) = \emptyset$. For independence to hold, we need $P(E_1 \mid E_2) = P(E_1)$. We know that $P(E_1 \mid E_2)P(E_2) = P(E_1 \cap E_2)$, so for independence to hold we need

$$P(E_1)P(E_2) = P(E_1 \cap E_2) = 0$$

Thus for independence to hold we need one of $E_1, E_2 = \emptyset$.
This may be counter-intuitive at first – one may think that disjoint events are indeed independent. Yet, while a Venn-diagram interpretation of events gives us information about *conditional* probability, it tells us nothing about *independence*.
The intuition is here – when two events are mutually exclusive, then either one or the other occurs. Thus, if one occurs then the other does not occur (and vice versa). This is, in and of itself, a **dependence** relationship.

Sometimes a particular conditional probability is difficult to calculate, but the reverse conditional probability is much easier.

Theorem 4.5.5 Bayes' Theorem
Let $P(B) \neq 0$. *Then*

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Before we prove this, we need a lemma.

Proposition 4.5.6

$$P(A \cap B) = P(A)P(B \mid A)$$

This is simply formalizing our multiplication rule for probabilities, accounting for dependent events. This is also just a definition of conditional probability.

Then Bayes' holds by getting $P(A \cap B) = P(A)P(B \mid A)$ and $P(B \cap A) = P(B)P(A \mid B)$, noticing that $P(A \cap B) = P(B \cap A)$, and setting these equal

$$P(A)P(B \mid A) = P(A \cap B) = P(B)P(A \mid B)$$

Remark 4.5.11
Our events in Bayes' theorem are historically written as A, B, but those are simply placeholders. We could have written them as E_1, E_2, E, F, etc.

Remark 4.5.12
There are a few mnemonics to remember this theorem. You can use the proof's formalization, which requires a simple rearrangement to recover the theorem. You can also rearrange the theorem to get

$$P(A \mid B) = \frac{P(A) \cdot P(B \mid A)}{P(B)}$$

and remember this by remembering $ABABAB$.

Example: Bayes' is often used for false positives and false negatives. Suppose there exists a disease, and a test for this disease. Suppose for people who *have* the disease, the test accurately says **yes** 90% of the time. Suppose for people who *do not have* the disease, the test falsely says **yes** 5% of the time (false positives). Suppose finally that 2% of the world population has this disease. You take the test, and it turns up positive. What is the probability you actually have the disease? Further

suppose the test turned up negative. Do we have enough informaiton to determine if you actually do *not* have the disease?

Solution: We have events D and Y, representing that you have the Disease and representing the test saying Yes. Our goal is to find $P(D \mid Y)$ – the probability you have the disease given the test says you have the disease. We know $P(Y \mid D) = 0.9$ – the probability that the test says yes given you have the disease. We also know $P(D) = 0.02$ – the probability that you have the disease, which is just the amount of people in the world who have the disease. Then we can use Bayes' so long as we have $P(Y)$ – the probability of the test saying yes. Well, we know the test says yes in the case that you do have the disease and the case that you do not – these cover the entirety of our sample space. So we can add them. $P(Y \mid D) = 0.9$ and $P(Y \mid \neg D) = 0.05$, so $P(Y) = P(Y \mid D)P(D) + P(Y \mid \neg D)P(\neg D) = (0.9)(0.02) + (0.05)(1 - 0.02) = 0.067$. Then by Bayes', $P(D \mid Y) = \frac{P(D)P(Y|D)}{P(Y)} = \frac{(0.02)(0.9)}{0.067} \approx 0.2687 \approx 26.87\%$ chance. So just above a 1 in 4 chance that you actually have the disease when the test says you do. I suppose the test is not very good then.

Now suppose the test says you do not have the disease. Then we need the probability that the test says No, $P(N)$. Now, the test either says yes or it says no, and these events are mutually exclusive. So $P(N) = 1 - P(Y) = 1 - 0.067$. Then we aim to find $P(\neg D \mid N)$, which by Bayes' means we need $P(\neg D) = 1 - P(D) = 0.98$ and $P(N \mid \neg D)$. This is the probability that the test says no given you do not have the disease. We know the probability that the test says *yes* given you do not have the disease, 0.05. Then $P(N \mid \neg D) = 1 - P(Y \mid \neg D) = 1 - 0.05 = 0.95$. Indeed we do have enough information! Thus by Bayes' $P(\neg D \mid N) = \frac{P(\neg D)P(N|\neg D)}{P(N)} = \frac{(0.98)(0.95)}{1-0.067} \approx 0.9979 \approx 99.79\%$. I suppose the test accurately says you do *not* have the disease, so maybe it is not so bad then.

Remark 4.5.13
The second part of this example reminds us of an interesting idea. $P(A \mid B) + P(\neg A \mid B) = 1$. However, we know nothing about the relationship between $P(A \mid B)$ and $P(A \mid \neg B)$ (see exercises).

Note that if events A and B are independent then Bayes' tells us that $P(A) = P(A \mid B) = \frac{P(B|A) \cdot P(A)}{P(B)} = \frac{P(B) \cdot P(A)}{P(B)} = P(A)$. This does not super useful, though.

4.6 Basic Statistics

Statistics are an important application of probability. Statistics give us a way to mathematically model probabilities of big events/experiments. We cover three fundamental statistics principles, along with a few related topics.

Remark 4.6.1
Before we continue, you may notice that we also write $\Pr(E)$ to be the probability that event E occurs. Before, we wrote this as $P(E)$. These notations are interchangeable.

Definition 4.6.2 Random Variable
A random variable is an abstraction of a sample space.

We denote X as a random variable. Random variables can be discrete or continuous, but we will focus only on the discrete case. Random variables take on values, or outcomes. Typically the outcomes of a random variable are actual values, but can sometimes be events with associated values. All possible random variable outcomes have an associated probability. In the discrete case, we can enumerate each probability. We know that all of these probabilities must add to 1. This also tells us that the probability of an outcome *not* from the random variable has probability 0.

Since the values of X are numeric values, we can apply our standard mathematical operators. We denote $\Pr(X = x)$ as the probability that the random variable X outputs the value x. Similarly, we denote $\Pr(X > x)$ as the probability that X is greater than x. We can take the probability of any proposition dependent on X.

Definition 4.6.3 Expected Value
$\mathbb{E}[X] = \mu_X = \sum_{x \in X} x \cdot \Pr(X = x)$. This is a generalization of the *arithmetic mean*, which is defined in scenarios where outcomes are equally likely. If we denote $x_1, x_2, x_3, \cdots, x_n$ to be the values outputted by X, and $p_1, p_2, p_3, \cdots, p_n$ to be their respective probabilities, then $\mathbb{E}[X] = \sum_{i=1}^{n} p_i x_i$

Sometimes students confuse *expected value* with *average*. First, the "average" is not well-defined – it could refer to the mean, median, or mode (typically it refers to the mean). Second, as stated in the definition, the expected value is a generalized *mean*. The expected value is precisely the value of the random variable we expect to get.

Note that

$$\mathbb{E}[f(X)] = \sum_{x \in X} f(x) \cdot \Pr(X = x)$$

which means that the expected value of some function of X is the same as the expected value of X but with $f(x)$ substituted for x.

Proposition 4.6.1
$\mathbb{E}$ *is a linear function, i.e.* $\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$

Proof. The expected value of a *value*, not a *variable*, is the value itself: $\mathbb{E}[v] = v$. If we think of a random variable V that always outputs v, then $\Pr(V = v) = 1$, so $\mathbb{E}[V] = \sum_{v \in V} v \cdot \Pr(V = v) = v \cdot 1 = v$.
Then the expected value of the sum of variables is the sum of the expected value of those variables: $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.

$$\begin{aligned}
\mathbb{E}[X+Y] &= \sum_{x+y \in X+Y} (x+y) \cdot \Pr(X+Y = x+y) \\
&= \sum_{x \in X} \sum_{y \in Y} (x+y) \cdot \Pr(X = x \wedge Y = y) \\
&= \sum_{x \in X} \sum_{y \in Y} x \cdot \Pr(X = x \wedge Y = y) \\
&+ \sum_{x \in X} \sum_{y \in Y} x \cdot \Pr(X = x \wedge Y = y) \\
&= \sum_{x \in X} x \sum_{y \in Y} \Pr(X = x \wedge Y = y) \\
&+ \sum_{y \in Y} y \sum_{x \in X} \Pr(X = x \wedge Y = y) \\
&= \sum_{x \in X} x \sum_{y \in Y} \Pr(X = x) \Pr(Y = y \mid X = x) \\
&+ \sum_{y \in Y} y \sum_{x \in X} \Pr(Y = y) \Pr(X = x \mid Y = y) \\
&= \sum_{x \in X} x \cdot \Pr(X = x) \sum_{y \in Y} \Pr(Y = y \mid X = x) \\
&+ \sum_{y \in Y} y \cdot \Pr(Y = y) \sum_{x \in X} \Pr(X = x \mid Y = y) \\
&= \sum_{x \in X} x \cdot \Pr(X = x) + \sum_{y \in Y} y \cdot \Pr(Y = y) \\
&= \mathbb{E}[X] + \mathbb{E}[Y]
\end{aligned}$$

These together give us the result. □

Remark 4.6.4
Linearity is extended to any countable number of sums and products. So

$$\mathbb{E}[c_1 X_1 + c_2 X_2 + \cdots + c_n X_n] = c_1 \mathbb{E}[X_1] + c_2 \mathbb{E}[X_2] + \cdots + c_n \mathbb{E}[X_n]$$

Definition 4.6.5 Variance
$\text{var}(X) = \sigma_X^2 = \mathbb{E}[(X - \mu_X)^2] = \mathbb{E}[(X - \mathbb{E}[X])^2]$

Proposition 4.6.2

$$\sigma_X^2 = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

Proof. $\mathbb{E}$ is linear. Then

$$\begin{aligned}
\sigma_X^2 &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\
&= \mathbb{E}[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2] \\
&= \mathbb{E}[X^2] - 2\mathbb{E}[X]\mathbb{E}[X] + \mathbb{E}[\mathbb{E}[X]^2] \\
&= \mathbb{E}[X^2] - 2\mathbb{E}[X]^2 + \mathbb{E}[X]^2 \\
&= \mathbb{E}[X^2] - \mathbb{E}[X]^2
\end{aligned}$$

□

Variance is a measure for the *spread* of a random variable's sampled output from the variable's expected value. Variance also leads us to an idea of *Covariance* between two (or more) random variables, which leads us to the idea of random variables being *correlated*. If the covariance between two random variables is zero, then the two random variables are *uncorrelated*, or independent. This is somewhat out of scope for this course, so search online if you are interested.

Definition 4.6.6 Standard Deviation
$\text{stdDev}(X) = \sigma_X = \sqrt{\sigma_X^2} = \sqrt{\text{var}(X)}$

The standard deviation is also a measure for the spread of a random variable's outcomes from the expected value. In this case, though, the *unit* for standard deviations is the same unit used for the expected value. With variance, however, the unit is the *square* of the expected value's unit. For example, if we had a random variable whose value outputs are in meters, then the expected value's unit will be in meters, the variance's unit will be in meters squared, and the standard deviation's unit will be in meters. Standard deviation is useful for reporting, since its units are easier to understand, but variance is useful for analysis (proofs). The two are entirely dependent on each other, however. With one, you have the other.

Our formulation for the standard deviation is the *population* standard deviation – when we take a holistic view of the entire output of a random variable. We can also discuss *sample* standard deviation, which estimates the population standard deviation from a random sampling of the random variable. The formula changes, however, since sampling a variable introduces bias in that we could sample

the same value more than once. This is a gross oversimplification, however these ideas are out of scope for this course. We encourage you to read more if interested.

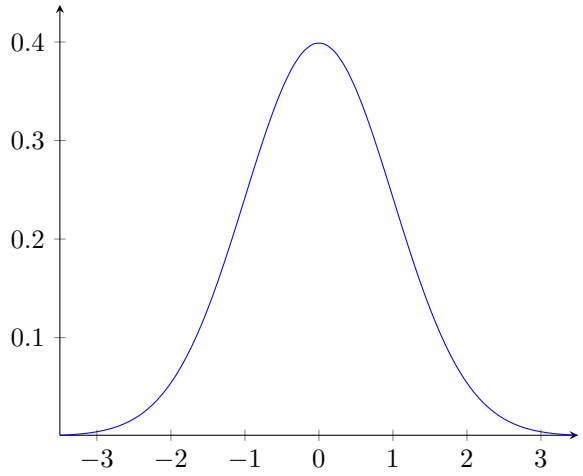
We will end with two examples of random variables, and how they give rise to *probability distributions*.

Remark 4.6.7
Before we begin, let us introduce some notation. In statistics, the notation $X \sim Y(\cdot)$ means that X *is distributed according to* $Y(\cdot)$. $Y(\cdot)$ is some function that describes whatever probability distribution you care about.

Example: Suppose U_n distributes the values $\{1, 2, \cdots, n\}$ each with **equal** probability. We call $X \sim U_n$ a *uniform distribution*, since each value is distributed uniformly (equally) as an output. Examples of this include coin flips, dice rolls, picking marbles with replacement, etc.
The expected value of a uniform distribution is $\mathbb{E}[X] = \sum_{x \in X} x \cdot \Pr(X = x) = \sum_{x=1}^{n} x \cdot \frac{1}{n} = \frac{1}{n} \sum_{x=1}^{n} x = \frac{1}{n} \frac{n(n+1)}{2} = \frac{n+1}{2}$. For example, the expected value of a standard 6-sided die is $\frac{7}{2} = 3.5$.
$\mathbb{E}[X]^2 = \frac{1}{n^2}$, so it suffices to calculate $\mathbb{E}[X^2]$ for the variance. For this, we need to examine X^2, which is the square of each output from X. These values are $\{1^2, 2^2, 3^2, \cdots, n^2\}$. The associated probabilities do not change. Then $\mathbb{E}[X^2] = \sum_{x=1}^{n} x^2 \cdot \frac{1}{n} = \frac{1}{n} \frac{n(n+1)(2n+1)}{6} = \frac{(n+1)(2n+1)}{6}$. Then the variance is $\mathbb{E}[X^2] - \mathbb{E}[X]^2 = \frac{(n+1)(2n+1)}{6} - \left(\frac{n+1}{2}\right)^2 = \frac{(n+1)(2n+1)}{6} - \frac{(n+1)^2}{4} = \frac{2(n+1)(2n+1)}{12} - \frac{3(n+1)^2}{12} = \frac{2(n+1)(2n+1) - 3(n+1)^2}{12} = \frac{(2(2n+1) - 3(n+1))(n+1)}{12} = \frac{(n-1)(n+1)}{12} = \frac{n^2-1}{12}$. For example, the variance of a 6-sided die is $\frac{6^2-1}{12} = 2.91\overline{6}$.
The standard deviation is the square root of the variance, which in the uniform case is $\sqrt{\frac{n^2-1}{12}}$. For example, the standard deviation of a 6-sided die is $\sqrt{2.91\overline{6}} \approx 1.708$.

Remark 4.6.8
You thus may see our uniform distribution noted as $X \sim U(1, n)$, which is abstracted to $X \sim U(a, b)$ where a and b, $a \leq b$, are arbitrary endpoints. In this course, the range between a and b is discrete with unit step sizes, however the uniform distribution can be extended to a continuous range.

Example: Suppose X is a random variable with $\mu = 0$ and $\sigma = 1$. We call X a *standard normal distribution*. It looks like this:



The x-axis is in units of standard deviations away from the mean, and the y-axis is in percentage of values observed. The area under the graph gives us the total percentage of values observed within a given range. Recall $\mu = 0$ and $\sigma = 1$. We note that 68.27% of observations are within $\mu \pm 1\sigma$, 95.45% within $\mu \pm 2\sigma$, and 99.73% within $\mu \pm 3\sigma$. This tells us that if you sample a normal distribution, there is just above a half chance that your value will be somewhat to the mean, and almost definitely within two standard deviations. We also see that the mean is equal to the median (the midpoint of observations), and is equal to the mode (the most amount of observations) – look at the tip of the curve! Oh, we also call the normal distribution a *bell curve*, because it kind-of looks like a bell!

We can also *normalize* a random variable. If μ and σ are the mean and standard deviation of X then $Z = \frac{X-\mu}{\sigma}$ will be roughly normally distributed.

Normal distributions occur in many situations in nature. For example, human height. Another example, student grade data is often normally distributed (this is where the term *curving* comes in, which is when μ is low so the "curve" is shifted to reduce grade cutoffs). Finally, a Galton Board, for large enough trials, approximates a normal distribution.

4.7 Summary

- Combinatorics is a toolbox, a mindset, for counting.
- Probability often utilizes counting techniques.
- Statistics is applied counting and probability.

4.8 Practice

1. We wish to build a sandwich made of bread, vegetables, cheese, and meat. We can select one of each kind of sandwich substance. There are 8 kinds of breads, 3 kinds of vegetables, 4 kinds of cheese, and 4 kinds of meat. How many possible sandwiches can we make?

2. In a standard deck of cards, each card has an associated rank (number) $\in R = \{A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K\}$ and suit (symbol) $\in S = \{\spadesuit, \heartsuit, \diamondsuit, \clubsuit\}$. The set of cards can be abstracted to pairings of ranks and symbols $C = R \times S$. For example, $(A, \clubsuit)$ is the ace of clubs, and $(7, \spadesuit)$ is the 7 of spades. Typically in card games each player has a *hand*, which is simply a set of cards.

   (a) Calculate the total amount of n-card hands. Use this to calculate the total amount of 5-card hands.

   (b) A hand has a *pair* if two cards in the hand have the same rank. Calculate the total amount of 5-card hands that have a pair.

   (c) There are other types of hands you could have (for example, 3-of-a-kind, straight, full house, etc.). A *straight* is a hand where the cards can be ordered, with wrap-around, in ascending (or descending) order. We let $A < 2 < \cdots < 10 < J < Q < K$. For example, $\{(J, \cdot), (Q, \cdot), (K, \cdot), (A, \cdot), (2, \cdot)\}$ is a valid straight. Calculate the total amount of straights. What if we did not allow for wraparound?

   (d) A *full house* is a 5-card hand with a pair and a 3-of-a-kind (three cards with the same rank). Calculate the total number of full houses.

   (e) Notice that our hands with pairs from (b) include full houses. How many 5-card hands have a pair but are **not** full houses? Now, these hands also include 3-of-a-kinds and 4-of-a-kinds. Remove these as well. Do we need to remove straights? Do we need to remove 5-of-a-kinds, or any other n-of-a-kinds?

   (f) These have all been related to ranks, but there are also hands related to suits. For example, calculate the total amount of 5-card hands where each card is the same suit. Need these be removed from our set of 5-card hands that are pairs?

3. How many ways can you represent $n \in \mathbb{Z}^+$ as the sum of $k \in \mathbb{Z}^+$ integers? Solve this by first writing down all cases (as sets) of $n = 5, k = 3$. For example, $\{2, 2, 1\}$ sum to 5. Then re-write these cases as sums of 1. For example, $\{1 + 1, 1 + 1, 1\}$. Then find a pattern with the placement of , (commas) and + (plus signs) within the 1s.

4. How many terms are in the expansion of $(x_1 + x_2 + \cdots + x_k)^n$?

5. Justify why $0! = 1$ makes sense combinatorially.

6. Show that $\binom{n}{k} = \binom{n}{n-k}$ using a cominatorial argument and an algebraic argument.
7. Prove that $\binom{2n}{n} = \sum_{i=0}^{n} \binom{n}{i}^2$.
8. Prove that $\binom{n+2}{3} = \sum_{i=1}^{n} i(n-i+1)$.
9. Prove that $\binom{m+n}{2} - \binom{m}{2} - \binom{n}{2} = mn$.
10. Use the Binomial Theorem to show that $2^0\binom{n}{0} + 2^1\binom{n}{1} + \cdots + 2^n\binom{n}{n} = 3^n$. Use this idea to abstract out and show that $\sum_{i=0}^{n} r^i\binom{n}{i} = (r+1)^n$ for any $n \in \mathbb{Z}^+$.
11. Show that $\sum_{k=d}^{n} \binom{n}{k}\binom{k}{d} = 2^{n-d}\binom{n}{d}$.
12.
13.
14.
15.
16.
17.
18.
19. 2 of 20 light-bulbs are defective. You select two light-bulbs at random. What is the probability that neither bulb selected is defective?
20.
21.
22. Recall we calculated the probability of flipping n heads in a row. Explain whether this is the same probability as flipping alternating heads and tails. Further explain whether the actual pattern of heads and tails matters.
23. Find a relationship between $P(A \mid B)$ and $P(A \mid \neg B)$ *Hint: use Bayes'.*
24. We say that a random variable X follows a *Bernoulli distribution* if $X = \{0, 1\}$ and $\Pr(X = 1) = p \in [0, 1]$ and $\Pr(X = 0) = 1 - p = q \in [0, 1]$. We can describe this as $X \sim \text{Bernoulli}(p)$ since the Bernoulli is entirely determined by the input probability p. For example, a coin flip is a Bernoulli distribution with $p = 0.5$, $x = 0$ being heads (or tails), and $x = 1$ being tails (or heads). Calculate μ_X, σ_X^2, and σ_X.
25. Suppose you are blindly throwing darts at a dart board. You have a $p \in [0, 1]$ chance of actually hitting the target in each independent trial. You wish to model the amount of shots $k \in \mathbb{Z}^+$ required to successfully hit the target. We say this situation follows a *geometric distribution* with parameter p. Each trial is modeled by a Bernoulli distribution. Let our

situation be $X \sim \text{Geo}(p)$. Then $\Pr(X = k) = (1-p)^{k-1}p$, which is the probability that you succeed on the k^{th} trial.

Justify why the probability function above is correct, and calculate the expected trial number of when you successfully hit the target μ_X. Further, calculate σ_X^2 and σ_X. Now set $p = 0.25$, a 25% chance of hitting the target. After how many thrown darts do you expect to hit the target?

Chapter 5

Functions and Relations

They don't think it be like it is, but it do.

Oscar Gamble

Contents

5.1 Introduction

Sometimes in life you will have two (or more) things that are somehow related. Sometimes people make these relations for you. Sometimes you will be given two things and you have to come up with the relation. Our goal in this section is to give you the tools to solve these problems life throws at you. And remember, when life gets you down, just chuck some lemons back at him.

This chapter begins studying relations, then builds up to define functions.

5.2 Relations

Relations are a formal way for us to map elements together – although this is not apparent at first. You may already be familiar with functions, *mappings*,

from high-school mathematics courses. We will build to functions using relations.

Relations simply give us a *relation*ship between elements. We begin with a formal definition, then dive into relation classification.

Definition 5.2.1 Relation
For sets A and B, a relation R is any subset of the cross product of A and B:

$$R \subseteq \{(a, b) \in A \times B\}$$

To specify that two elements are related under a relation R we say xRy which means $(x, y) \in R$

Here are a few examples:

Example: An *equality* relation $\{(a, b) \mid a = b\} \subseteq \mathbb{R} \times \mathbb{R}$. This is a set of pairs where, for any given pair both elements in the pair are equal.

Example: A *less-than* relation $\{(a, b) \mid a < b\} \subseteq \mathbb{N} \times \mathbb{N}$. This is a set of pairs where, for any given pair the first element is strictly less-than the second element.

Example: The following is a relation, with $A = \{2, 4\}$ and $B = \{6, 8, 10\}$: $R = \{(x, y) \in A \times B \mid \frac{y}{x} \in \mathbb{Z}\} = \{(2, 6), (2, 10), (4, 8)\}$

Remark 5.2.2
We will only consider relations of sets with themselves. We *can* define relations between different sets, however this will make the following definitions significantly more challenging to define.

Definition 5.2.3 Reflexive
A relation R on a set A is reflexive if

$$(\forall a \in A)[(a, a) \in R]$$

Definition 5.2.4 Symmetric
A relation R on a set A is symmetric if

$$(\forall a, b \in A)[(a, b) \in R \Rightarrow (b, a) \in R]$$

Definition 5.2.5 Transitive
A relation R on a set A is transitive if

$$(\forall a, b, c \in A)[(a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R]$$

You will typically be asked to classify whether a relation satisfies the above three properties.

Example: Classify the following:

Relation	refl.	symm.	trans.
$\{(1,1),(1,2),(2,1)\} \subseteq \{1,2\}^2$	◊	◊	◊
$\{(1,1),(1,2),(2,2)\} \subseteq \{1,2\}^2$	◊	◊	◊
$\{(x,y) \in \mathbb{N} \times \mathbb{N} \mid \frac{y}{x} \in \mathbb{Z}\}$	◊	◊	◊
$\{(x,y) \in \mathbb{R}^{\geq 1} \times \mathbb{R}^{\geq 1} \mid x < y^2\}$	◊	◊	◊
$\{(x,y) \in \mathbb{R} \times \mathbb{R} \mid (x = y) \vee (x = -y)\}$	◊	◊	◊

Solution: ♦ indicates YES, ◊ indicates NO

Relation	refl.	symm.	trans.
$\{(1,1),(1,2),(2,1)\} \subseteq \{1,2\}^2$	◊	♦	◊
$\{(1,1),(1,2),(2,2)\} \subseteq \{1,2\}^2$	♦	◊	♦
$\{(x,y) \in \mathbb{N} \times \mathbb{N} \mid \frac{y}{x} \in \mathbb{Z}\}$	◊	◊	♦
$\{(x,y) \in \mathbb{R}^{\geq 1} \times \mathbb{R}^{\geq 1} \mid x < y^2\}$	◊	◊	◊
$\{(x,y) \in \mathbb{R} \times \mathbb{R} \mid (x = y) \vee (x = -y)\}$	♦	♦	♦

Oftentimes in computer science and mathematics we care about relations that satisfy all three properties. We even give these relations a fancy name, *equality relation*.

Definition 5.2.6 Equivalence Relation
A relation that is reflexive, symmetric, and transitive

The equivalence relation gives us a notion of *equality*. If a relation R is an equivalence relation, then if xRy then we can think of x being "equal" to y. The equals sign $=$ gives us an equivalence relation on sets of numbers. In linear algebra, an isomorphism $\cong$ is an equivalence relation on vector spaces. The mod operator $\equiv$ gives us an equivalence relation on $\mathbb{Z}$. As a real-life application, modular arithmetic is used in cryptography!

Here are some other cool definitions that probably will not be tested.

Definition 5.2.7 Antisymmetric
A relation R on a set A is antisymmetric if

$$(\forall a, b \in A)[(a,b) \in R \wedge (b,a) \in R \Rightarrow a = b]$$

Definition 5.2.8 Composite of a Relation
If $R \subseteq A \times B$ and $S \subseteq B \times C$ are both relations, then the *composite* of R and S is

$$S \circ R = \{(a,c) \mid a \in A \wedge c \in C \wedge (b \in B \Rightarrow (a,b) \in R \wedge (b,c) \in S)\}$$

Definition 5.2.9 Power of a Relation
If $R \subseteq A \times A$ then for $n \in \mathbb{Z}^+$ the *power* $R^n = R^{n-1} \circ R$ where $R^1 = R$

Theorem 5.2.1
$R \subseteq A \times A$ is transitive iff $R^n \subseteq R$

The proof of this theorem is left as an exercise.

We have been discussing binary relations – relations between two sets – however we can define the notion of an n-ary relation. An n-ary relation is a subset of the cross product of n sets.

Example: Here is a trinary (or *ternary*) relation:

$$R = \{(1, a, *), (2, a, +), (1, b, +)\} \subseteq \{1, 2, 3\} \times \{a, b, c, d\} \times \{*, +\}$$

The notions of reflexivity, symmetry, and transitivity also abstract to n-ary relations, as does an equivalence relation. This is outside the scope of this course.

5.3 Functions

Now that we understand relations, we can define a function.

Definition 5.3.1 Function
A relation $R \subseteq C \times D$ that satisfies the following properties:

1. $(\forall c \in C)(\exists d \in D)[(c, d) \in R]$
2. $(\forall c \in C)(\forall d_1, d_2 \in D)[((c, d_1) \in R \wedge (c, d_2) \in R) \Rightarrow d_1 = d_2]$

If we recall the *exists unique* symbol $\exists!$, then we can combine the two properties above into one:

1. $(\forall c \in C)(\exists! d \in D)[(c, d) \in R]$

You may be familiar with the following less-formal definition.

Definition 5.3.2 Function
An explicit rule, or set of rules, that can be applied to a specified input to yield a specified output. There are two rules that all functions must follow:

1. Every input must be mapped to an output
2. No input can map to more than one output

Basically, functions give us a *mapping* between sets – a relation! For any relation $R \subseteq C \times D$, we let the first set C be the function *input* and the second set D be the function *output*. If a pairing $(c, d) \in R$, then the function representing the relation R means that the function input c yields the output d.

Example: Determine whether the following relations are functions.

Relation	is a func.	is not a func.
$\{(x,y) \in \mathbb{R} \times \mathbb{R}^{\geq 0} \mid (x = y) \vee (x = -y)\}$	△	△
$\{(x,y) \in \mathbb{N} \times \mathbb{N} \mid \frac{y}{x} \in \mathbb{Z}\}$	△	△
$\{(x,y) \in \mathbb{N} \times \{1\} \mid \frac{x}{y} \in \mathbb{Z}\}$	△	△
$\{(x,y) \in \mathbb{N} \times \{1\} \mid x < y\}$	△	△
$\{(x,y) \in \mathbb{N} \times \{0\} \mid x \geq y\}$	△	△

Solution: ▲ indicates YES, △ indicates NO

Relation	is a func.	is not a func.
$\{(x,y) \in \mathbb{R} \times \mathbb{R}^{\geq 0} \mid (x = y) \vee (x = -y)\}$	▲	△
$\{(x,y) \in \mathbb{N} \times \mathbb{N} \mid \frac{y}{x} \in \mathbb{Z}\}$	△	▲
$\{(x,y) \in \mathbb{N} \times \{1\} \mid \frac{x}{y} \in \mathbb{Z}\}$	▲	△
$\{(x,y) \in \mathbb{N} \times \{1\} \mid x < y\}$	△	▲
$\{(x,y) \in \mathbb{N} \times \{0\} \mid x \geq y\}$	▲	△

Every function is a relation, but not every relation is a function. For any function, you can always write out the $(x, f(x))$ pairings – this is by definition a relation. On the other hand, the above examples show that not every relation is a function.

The above relations are contrived examples. Typically we define functions on sets of numbers (we will restrict it to subsets of $\mathbb{R}$). We use the following notation to denote a function f:

$$f(x) = \text{some rules}$$

and we pair the input/output sets as follows:

$$f : D \to R$$

We might also notate functions by saying $f : D \to R$ given by $f : d \mapsto r$.

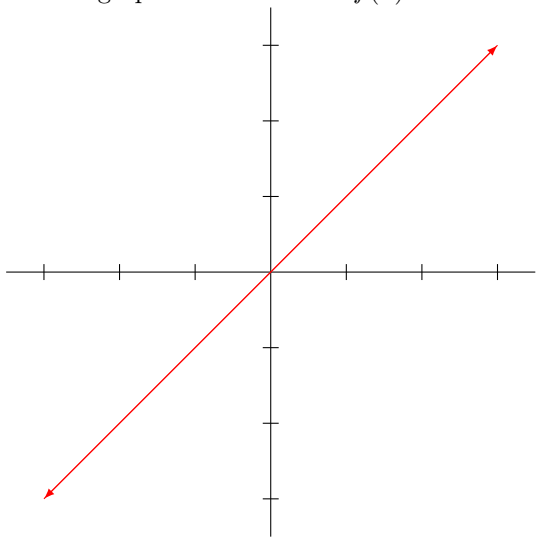
Definition 5.3.3 Domain
The set of inputs to a function

Definition 5.3.4 Range
The set containing all outputs to a function. This may also be called the *co-domain*

Definition 5.3.5 Image
The set of all outputs to a function. This may be a subset of the function's range

Let's look at some basic examples. First are some graphs of functions. Hopefully you learned how to plot 2-dimensional functions in previous math classes.

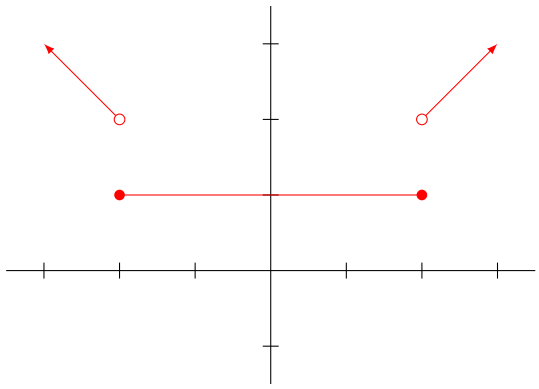
Example: Here is a graph of the function $f(x) = x$ where $f : \mathbb{R} \to \mathbb{R}$



Example: Here is a graph of the function

$$f(x) = \begin{cases} x, & x > 2 \\ 1, & -2 \leq x \leq 2 \\ -x, & x < -2 \end{cases}$$

where $f : \mathbb{R} \to \mathbb{R}$



Next we present examples of classifying whether a function is a function. To do this, recall the two requirements:

1. Every input must be mapped to an output
2. No input can map to more than one output

You should check both of these requirements. For the first, recall that to disprove a universal statement you need a single counter-example. Thus, search the domain and make sure each input maps to an output

Example: Which of the following are functions? For the non-functions, change them such that they become functions.

(a) $a(x) = x$ where $a : \mathbb{Z} \to \mathbb{R}$
(b) $b(x) = \sqrt{x}$ where $b : \mathbb{Z} \to \mathbb{R}$
(c) $c(y) = \log_2(y)$ where $c : \mathbb{N} \to \mathbb{Z}$
(d) $d(y) = y^2$ where $d : \mathbb{Z} \to \mathbb{N}$
(e) $e(z) = \begin{cases} z+1 & z \geq 0 \\ z-1 & z \leq 0 \end{cases}$ where $e : \mathbb{R} \to \mathbb{Z}$

Solution:

(a) Function
(b) Not a function – b does not work for any negative input, however the domain is set to $\mathbb{Z}$. Simply change the domain to $\mathbb{N}$
(c) Not a function – $c(3) \notin \mathbb{Z}$ thus not every input is mapped to an output. Change the co-domain to positive real numbers
(d) Function
(e) Not a function – (1) the input $z = 0$ has two different outputs $e(0) = 1$ and $e(0) = -1$, (2) any real number that is not an integer does not have a mapping in the range (ex: $e(1.5) = 1.5 + 1 = 2.5 \notin \mathbb{Z}$). Change one of the cases to not include 0, then either change the range to $\mathbb{R}$ or change the domain to $\mathbb{Z}$

There exist 3 properties that any function can have.

Definition 5.3.6 Injection (One-to-One)
A function where each co-domain (range) value is mapped to by at most one value in the domain. In other words, each value in the range has either 0 or 1 pointers from the domain. Formally, $f : D \mapsto R$ is injective iff

$$(\forall x_1, x_2 \in D)[f(x_1) = f(x_2) \Rightarrow x_1 = x_2]$$

The horizontal line test can be used to see if a function is injective

Definition 5.3.7 Surjection (Onto)
A function where each co-domain (range) value is mapped to by at least one value in the domain. In other words, there are no un-mapped values in the range. Formally, $f : D \mapsto R$ is surjective iff

$$(\forall y \in R)(\exists x \in D)[y = f(x)]$$

Definition 5.3.8 Bijection (One-to-One Correspondence)
A function that is both an injection and a surjection. Every input is mapped to exactly one unique output

You should know how to identify these properties in functions. To test injectivity, perform the horizontal line test – run a horizontal line vertically along the graph of the function; the function is injective if the horizontal line never touches more than one point. To test surjectivity, ask yourself if you can "get to" every output element. To be bijective, you must both be injective and surjective. A function can be neither injective nor surjective, but still be a function.

We can represent a function by drawing ovals representing the domain and range, drawing points in each oval representing the elements in each set, and drawing arrows between the points representing the function mapping. In this way, we present a visual guide to the above definitions.

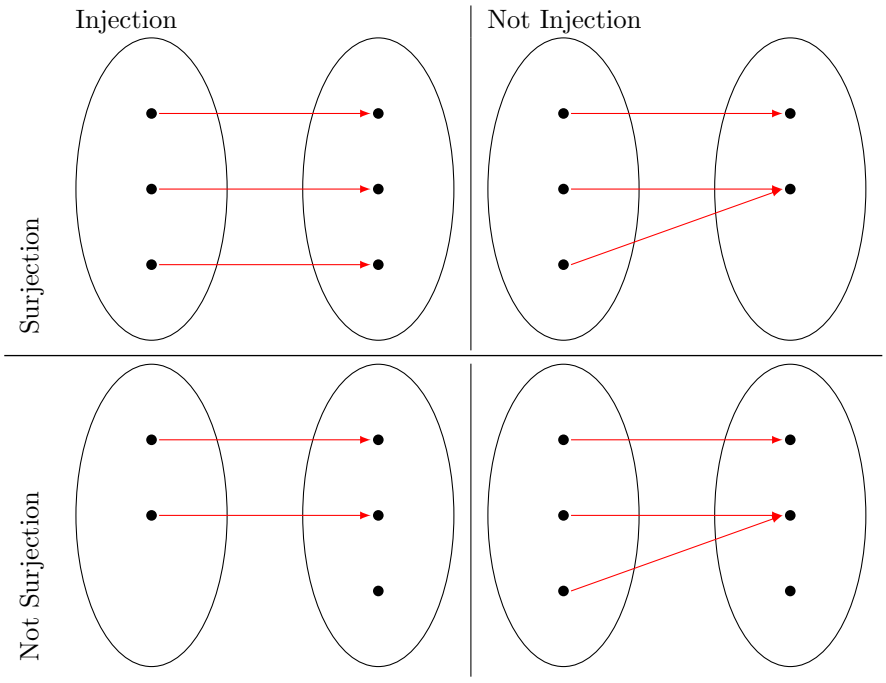


Figure 5.1: Injection and Surjection

Example: Classify the following:

		func.	inj.	surj.	bij.
$f(x) = x$	$f : \mathbb{N} \to \mathbb{Z}$	◊	◊	◊	◊
$g(x) = x$	$g : \mathbb{Z} \to \mathbb{Z}$	◊	◊	◊	◊
$h(x) = x^2$	$h : \mathbb{R} \to \mathbb{N}$	◊	◊	◊	◊
$i(x) = \lfloor x^2 \rfloor$	$i : \mathbb{R} \to \mathbb{N}$	◊	◊	◊	◊
$j(x) = x^3 - x$	$j : \mathbb{Z} \to \mathbb{R}$	◊	◊	◊	◊

Solution: ♦ indicates YES, ◊ indicates NO

		func.	inj.	surj.	bij.
$f(x) = x$	$f : \mathbb{N} \to \mathbb{Z}$	♦	♦	◊	◊
$g(x) = x$	$g : \mathbb{Z} \to \mathbb{Z}$	♦	♦	♦	♦
$h(x) = x^2$	$h : \mathbb{R} \to \mathbb{N}$	◊	◊	◊	◊
$i(x) = \lfloor x^2 \rfloor$	$i : \mathbb{R} \to \mathbb{N}$	♦	◊	♦	◊
$j(x) = x^3 - x$	$j : \mathbb{Z} \to \mathbb{R}$	♦	◊	◊	◊

Hash tables are an important application of functions in computer science. A hash table is a data structure; it is a fancy array. A hash table has an internal array, and a hash function. This hash function maps table elements (some set of objects) to indices ($\mathbb{N}$). The modulus of the hash function output, with respect to the table size, is taken as the input object's key. This is where the object is stored in the table. Ideally, you want your hash function to be an injection on the set of indices.

Theorem 5.3.1
A bijection $f : A \to B$ is invertible, and the inverse, denoted $f^{-1} : B \to A$, is also a bijection.

Proof. First note that *inverse* simply means that $f^{-1}(f(a)) = a$. With this in mind, then we note that $f^{-1}(b) = a$ such that $f(a) = b$. Then since f is a bijection, b is the one-and-only output that a maps to. Then a is the one-and-only output that b maps to in f^{-1}. So it is a bijection. We can show this more formally by showing f^{-1} is an injection and a surjection.
1-1: Let $f^{-1}(b_1) = f^{-1}(b_2)$, then $f(f^{-1}(b_1)) = f(f^{-1}(b_2)) \Rightarrow b_1 = b_2$.
onto: Let $f^{-1}(b) = a$, then $b = f(a)$ which exists because f is a bijection.
You should prove on your own that indeed f^{-1} is a function! □

Theorem 5.3.2
The composition of bijective functions is a bijection.

Proof. Let f, g be the bijections $f : A \mapsto B$ and $g : B \mapsto C$. Then by the previous theorem f, g have inverses which are also bijections.
Define $h : A \mapsto C$ given by $h : a \mapsto g(f(a)) = (g \circ f)(a)$.
1-1: Let $h(a_1) = h(a_2)$, then $g(f(a_1)) = g(f(a_2)) \Rightarrow f^{-1}(g^{-1}(g(f(a_1)))) = f^{-1}(g^{-1}(g(f(a_2)))) \Rightarrow a_1 = a_2$
onto: Let $h(a) = c$, then $g(f(a)) = c$ so $a = f^{-1}(g^{-1}(c))$ exists. □

5.4 Sequences

Sequences make up an interesting sub-part of functions. In calculus, we build up continuous functions as limits of sequences. This is discrete mathematics, however, so sequences will be of more interest to us.

Definition 5.4.1 Sequence
An ordered list of numbers, each one associated with a specific *position* (index)

For the purposes of this class, you can think of a sequence as a mapping between the natural numbers and the associated index in the sequence. Sometimes you can express a sequence as a closed-form function, and sometimes a sequence can have multiple interpretations. Let's look at a few examples to get the notation down.

Example: The following three notations describe the same sequence:
(a) $1, 3, 7, 15, 31, \ldots$
(b) $a_n = \begin{cases} 1 & , \ n = 1 \\ 2a_{n-1} + 1 & , \ n > 1 \end{cases}, \ n \in \mathbb{Z}^+$
(c) $f(n) = 2^n - 1$ where $f : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$

We have explicit names for the notation types in the previous example.

Definition 5.4.2 General Form
A written list of the elements of a sequence, in order, typically with ... at the end (and at the beginning, if your sequence elements start in the middle – typically this does not happen). Sometimes mathematicians put curly braces {} around their general-form sequences – try not to get this confused with a set! Other times mathematicians will give you a rule, written in plain English, for generating sequence terms

Definition 5.4.3 Recursive Form
An explicit formula that depends on previous elements in the sequence, and necessarily has one or more base cases. Typically this is expressed as

$$x_n = \begin{cases} \text{base case(s)} & , n \in \{\text{some set}\} \\ \text{some formula dependent on } n & , n \notin \{\text{the previous set}\} \end{cases}$$

Definition 5.4.4 Closed Form
An explicit formula to generate a specific term in a sequence. This formula is solely dependent on a given index. Typically this is expressed as

$$y_k = \text{some formula dependent on } k$$

Remark 5.4.5
Sequence numbers do not need to be in any particular order, nor do they necessarily have to fit some sort of function. Sequences elements can also be any sort of number (we will restrict to $\mathbb{R}$).

Example: Some sequences:

$$1, 0, -1, 0, 1, 0, -1, 0, \ldots$$

$$8, 6, 4, 2, 8, 6, 4, 2, 8, \ldots$$

$$0, 0.5, 0.333, 0.25, 0.2, 0.125, \ldots$$

Sometimes we will ask you to generate terms of a sequence. This is a relatively straightforward task so long as you avoid simple arithmetic errors.

Example: Generate the first 7 terms of the following sequences:
(a) The nth term is its index tripled (start your indices at 0)
(b) $f_k = \begin{cases} 1 & , k = 1, 2 \\ f_{k-1} + f_{k-2} & , k \geq 3 \end{cases}$
(c) $l_1 = l_2 = l_3 = 1$ and $l_i = l_{i-1} + l_{i-2} + l_{i-3}$

Solution: We will use {} notation here:
(a) A simple rule (one that we could express in closed form) – $\{0, 3, 6, 9, 12, 15, 18, \ldots\}$
(b) This is the Fibonacci sequence – $\{1, 1, 2, 3, 5, 8, 13, \ldots\}$
(c) In this case, the index is implied to be $\in \mathbb{Z}^+$ – $\{1, 1, 1, 3, 5, 9, 17, \ldots\}$

Other times we will ask you to find a closed form for a sequence given in general or recursive form. Typically we ask this about general forms, however occasionally you will see a recursive form. This task is akin to pattern-matching and sometimes requires much more critical thinking. Sometimes sequences can have multiple answers – any correct answer suffices. Answers that are unreasonable – i.e. answers that only fit the provided sequence digits – will receive a good chuckle along with zero marks.

Example: Find closed-forms for the following sequences:
(a) $\{1, 4, 9, 16, 25, \dots\}$
(b) $\{3, 6, 9, 12, 15, \dots\}$
(c) $\{1, 4, 9, 6, 5, 6, 9, \dots\}$
(d) $\{1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, \dots\}$
(e) $\{2, 5, 8, 0, 3, 6, 9, \dots\}$

Solution: We attempt to explain the patterns that led us to the solutions:
(a) The pattern here is squares. Each sequence number is the square of its index
(b) $f(n) = 3(n + 1)$ where $f : \mathbb{N} \to \mathbb{Z}$
(c) Hopefully you still had the first sequence in mind – squares modulo 10
(d) Notice a pattern with the indices – specifically look at the index where there is a switch. Index 1 starts the 1s, index 4 starts the 2s, index 9 starts the 3s, index 16 starts the 4s. What is the relationship between these numbers? Take the square root of the index. Then realize that square roots maintain inequalities – $1 = \sqrt{1} < \{\sqrt{2}, \sqrt{3}\} < \sqrt{4} = 2$, so look towards floors and ceilings to yield integers. Thus, $f(n) = \lfloor \sqrt{n} \rfloor$ where $f : \mathbb{Z}^+ \to \mathbb{Z}^+$
(e) This one can have multiple answers. One solution: $f(n) = 3n - 1 \pmod{11}$ (taking least positive remainder) where $f : \mathbb{Z}^+ \to \mathbb{Z}^+$. Typically if you see a somewhat circular sequence, especially if there is a zero, then consider applying a modulus. Start at the zero and look forward until the sequence reverts. In this case, we have $0, 3, 6, 9$ which is simply $f(n) = 3n$. Then play around with the start point and the modulus

Finding a closed-form function that matches a sequence is difficult. It requires a deep understanding of how all mathematical operators work, and requires a knack for finding patterns. There are multiple methods for estimating functions, including polynomial interpolation, piece-wise functions, floor-ceiling interpolation, and least-squares. These methods usually yield a result significantly more complicated than finding a simpler function. However, there are merits to using these methods as a starting-point.

5.4.1 Series

You may remember series from a calculus course. Series are closely related to sequences.

Definition 5.4.6 Series
The sum of all terms in a sequence

Example: The Geometric Series ($r \neq 1$)

$$a + ar + ar^2 + \cdots ar^{n-1} = \sum_{i=0}^{n-1} ar^i$$

which has the solution

$$= a(\frac{1-r^n}{1-r})$$

Typically we try to understand the limiting behavior of infinite series. There are a number of tests you can use to study this behavior, however this topic is typically taught in Calculus II classes. We will thus omit this topic here, however we do recommend at least understanding the idea behind limits. We will come back to infinite objects when we discuss countability in chapter 6.

Example: Determine whether the series $\sum_{i=1}^{\infty} \frac{1}{3^{i-1}}$ converges or diverges.

Solution: From the geometric series formula above, $\sum_{i=1}^{n} \frac{1}{3^{i-1}} = \sum_{i=0}^{n-1} \frac{1}{3^i} = \sum_{i=0}^{n-1} \frac{1}{3}^i = \frac{1-\frac{1}{3}^n}{1-\frac{1}{3}} = \frac{3}{2}(1 - \frac{1}{3^n})$. Then we can take the limit as $n \to \infty$, which $= \frac{3}{2}(1-0) = \frac{3}{2}$

5.5 Summary

- Relations give us a formal way to write out a mathematical relationship between elements in sets.
- Functions are a special kind of binary relation that satisfy $(\forall c \in C)(\exists! d \in D)[(c,d) \in R]$. A function $f : C \mapsto D$ is then described by the input/output pairs $(c,d) \in R$.
- Sequences have three forms: general, recursive, and closed.

5.6 Practice

1. Write out the *greater than* relation $R \subseteq \{1,2,3\} \times \{-1,1,3,5\}$
2. Classify the following relations as reflexive, symmetric, and/or transitive:
   (a) $\{(x,y) \in \mathbb{N} \times \mathbb{N} \mid x \leq \sqrt{y}\}$
   (b) $\{(x,y) \in \mathbb{N} \times \mathbb{N} \mid x \equiv y^2 \pmod 4\}$

(c) $\{(x, y) \in \mathbb{N} \times \mathbb{N} \mid \sqrt{x} \leq y\}$

3. Prove Theorem 5.2.1.

4. Determine whether the following relations are functions:

   (a) $\{(x, y) \in \mathbb{Z}^- \times \mathbb{Z}^+ \mid x < y\}$

   (b) $\{(x, y) \in \mathbb{Z}^+ \times \mathbb{Z}^- \mid x < y\}$

   (c) $\{(x, y) \in \mathbb{R} \times \mathbb{R} \mid y = x + 2\}$

5. Why is the relation $\{(x, y) \in \mathbb{Z} \times \mathbb{Z} \mid x \equiv y \pmod{m}\}$ **not** a function?

6. Determine whether the following are functions. If they are, then classify them as injective, surjective, and/or bijective.

   (a) $f : \mathbb{Z} \to \mathbb{Z}$ given by $f : z \mapsto \%37$

   (b) $f : \mathbb{Q} \to \mathbb{Q}$ given by $f : q \mapsto \sqrt{q}$

   (c) $f : \mathbb{R} \to \mathbb{Z}$ given by $f : x \mapsto \lfloor x \rfloor$

7. Suppose a function $f : X \to Y$ is injective and $h : X \to X$ is bijective. Is the function $g : X \to Y$ given by $g(x) = f(h(x))$ also injective? What if h is only surjective? What if h is only injective?

8. Find closed-forms, or simple rules, that describe the following sequences:

   (a) $\{-3, -1, 1, 3, 5, \cdots\}$

   (b) $\{-1, 1, 7, 25, 79, \cdots\}$

   (c) $\{2, 3, 5, 0, 4, \cdots\}$

Chapter 6

Countability

$\aleph_0$ is my favorite character!

Justin Goodman

Contents

6.1 Introduction

Our motivation here is that we want to describe the sizes of incredibly large sets. What is the cardinality of $\mathbb{N}$? So far, we have not learned any means to count the natural numbers. That is what this chapter is for!

Additional reading: The Banach-Tarski Paradox – Vsauce (relevant content starts at 2:06)

6.2 Definitions

Definition 6.2.1 Enumeration of a Set
A specified ordering of a set. Enumerations should be described in such a way that given a partial listing, you can always find the next element.

Example: We could enumerate the set of natural numbers as follows:

$$0, 1, 2, 3, 4, 5, 6, 7, \cdots$$

We call this the *natural ordering* of $\mathbb{N}$. Some sets might not have a natural ordering, so we must enumerate it to provide an ordering.

Example: We could enumerate the set of integers as follows:

$$0, 1, -1, 2, -2, 3, -3, 4, -4, \cdots$$

Example: The following is *not* a valid enumeration of the integers:

$$0, 1, 2, 3, 4, 5, \cdots, -1, -2, -3, -4, -5, \cdots$$

This is not valid because we will never know when to switch from positive to negative integers.

The point of enumeration is such that we can create *indices* for each element in a set. As we will see later, it will be helpful if you know that, for example, the index $i = 4$ into the set of integers yields -2.

Definition 6.2.2 Finite
A set is finite if there exists a bound on the number of elements. If you start counting the elements in a finite set, there should be a precise stopping point.

Example: $\{1, 2, 3\}$ and $\mathbb{N}^{<10,000,000,000}$ are finite sets.

Definition 6.2.3 Countably Infinite
A set is countably infinite if you can find a one-to-one correspondence (bijection) between the set and $\mathbb{N}$. Alternatively, a set is countably infinite if you can enumerate the set such that every element appears at a finite position.

More simply, a set is countably infinite if you can index each element.

Remark 6.2.4
We denote $|S| = \aleph_0$ for any (every) countably infinite set S.

There exist other definitions for countably infinite, however they all share the same idea that *countably infinite* sets can be *counted in a finite amount of time.*

Definition 6.2.5 Countable
A set is countable if it is finite or countably infinite.

Example: $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ is countable.

Proof. S is finite, with 10 elements, so it is countable. □

Example: $\mathbb{N}$ is countable.

Proof. The function $f(x) = x$ where $f : \mathbb{N} \mapsto \mathbb{N}$ is a bijection, so $\mathbb{N}$ is countably infinite. □

Example: $\mathbb{Z}$ is countable.

Proof. Enumerate $\mathbb{Z}$ as follows: $\{0, -1, 1, -2, 2, -3, 3, \cdots\}$
Then

$$f(x) = \begin{cases} \frac{n}{2} & x \equiv 0 \pmod 2 \\ -\frac{n+1}{2} & x \equiv 1 \pmod 2 \end{cases}$$

where $f : \mathbb{N} \mapsto \mathbb{Z}$ is a bijection (*why?*). □

Example: $\mathbb{Q}^+$ is countable.

Proof. We use what is called a *snaking argument*. The idea is to build a grid that describes all positive rational numbers, then provide a bijection between that grid and the natural numbers.
Grid:

	1	2	3	4	5	$\cdots$
1	1/1	2/1	3/1	4/1	5/1	$\cdots$
2	1/2	2/2	3/2	4/2	5/2	$\cdots$
3	1/3	2/3	3/3	4/3	5/3	$\cdots$
4	1/4	2/4	3/4	4/4	5/4	$\cdots$
5	1/5	2/5	3/5	4/5	5/5	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Let $f(n)$ yield the rational number at position n along the snake in the table (e.g. $f(0) = 1/1$, $f(1) = 2/1$, $f(2) = 1/2$). Then $f : \mathbb{N} \mapsto \mathbb{Q}^+$ is a bijection (*why?*). □

Remark 6.2.6
This proof does not account for the fact that we can reduce fractions (like how $4/2 = 2/1$). A more rigorous proof uses the Schröder-Bernstein Theorem, which is out of the scope of this course.

Definition 6.2.7 Uncountable
A set that is not countable.

Example: $\mathbb{R}$ is uncountable.

But why? We will come back to this. First, we need a new proof tool to help us, plus some theorems.

6.3 Cantor's Diagonal Argument

We begin our study of this famous proof by diving into the actual proof.

Example: Prove $(0, 1)$ is uncountable.

Proof. By contradiction. Assume $(0, 1)$ is countable. Then, we can enumerate the entire set. One possible enumeration is as follows:

$$\begin{array}{lccccccccc}
r_0 = 0. & 0 & 1 & 2 & 6 & 5 & 9 & 8 & 7 & \cdots \\
r_1 = 0. & 1 & 8 & 4 & 3 & 1 & 3 & 0 & 8 & \cdots \\
r_2 = 0. & 1 & 9 & 9 & 5 & 9 & 9 & 6 & 6 & \cdots \\
r_3 = 0. & 1 & 9 & 0 & 3 & 2 & 5 & 2 & 7 & \cdots \\
r_4 = 0. & 1 & 9 & 0 & 4 & 2 & 0 & 4 & 1 & \cdots \\
r_5 = 0. & 1 & 9 & 0 & 4 & 3 & 3 & 8 & 2 & \cdots \\
r_6 = 0. & 1 & 9 & 0 & 4 & 3 & 4 & 8 & 2 & \cdots \\
r_7 = 0. & 1 & 9 & 0 & 4 & 3 & 4 & 9 & 2 & \cdots \\
r_8 = 0. & 1 & 9 & 0 & 4 & 3 & 4 & 9 & 3 & \cdots \\
\vdots & & & & & & & & &
\end{array}$$

Now, let us construct an element $x \in (0, 1)$:

$$x = 0.x_1x_2x_3x_4\cdots$$

where

$$x_i \equiv r_{i,i} + 1 \pmod{10}$$

and $r_{i,i}$ is the i^{th} digit after the decimal point in r_i. Here, we are using the mod function to give us the least-positive number in the congruence class $r_{i,i} + 1 \pmod{10}$ – otherwise known as the *remainder*.
So we grab the *diagonal* digits,

$$\begin{array}{lccccccccc}
r_0 = 0. & \mathbf{0} & 1 & 2 & 6 & 5 & 9 & 8 & 7 & \cdots \\
r_1 = 0. & 1 & \mathbf{8} & 4 & 3 & 1 & 3 & 0 & 8 & \cdots \\
r_2 = 0. & 1 & 9 & \mathbf{9} & 5 & 9 & 9 & 6 & 6 & \cdots \\
r_3 = 0. & 1 & 9 & 0 & \mathbf{3} & 2 & 5 & 2 & 7 & \cdots \\
r_4 = 0. & 1 & 9 & 0 & 4 & \mathbf{2} & 0 & 4 & 1 & \cdots \\
r_5 = 0. & 1 & 9 & 0 & 4 & 3 & \mathbf{3} & 8 & 2 & \cdots \\
r_6 = 0. & 1 & 9 & 0 & 4 & 3 & 4 & \mathbf{8} & 2 & \cdots \\
r_7 = 0. & 1 & 9 & 0 & 4 & 3 & 4 & 9 & \mathbf{2} & \cdots \\
r_8 = 0. & 1 & 9 & 0 & 4 & 3 & 4 & 9 & 3 & \cdots \\
\vdots & & & & & & & & &
\end{array}$$

and to each of them we add one (mod 10) to construct

$$x = 0.19043493\cdots$$

We claim that this new x is not in the enumeration. Your keen eye may have noticed that $x = r_8$, so our claim is bogus. This is a rightful note, *however* our claim is still valid. Remember that the numbers in the enumeration have an infinite decimal expansion. So $r_8 = 0.19043493d_8d_9d_{10}d_{11}\cdots$, as does $x = 0.19043493x_8x_9x_{10}x_{11}\cdots$.
For decimal numbers to be equal, all of their corresponding digits must be equal. When we construct the digit x_8 in x, do we have that $x_8 = d_8$, the corresponding digit in r_8? Well, by construction, $x_8 \equiv r_{8,8} + 1 \equiv d_8 + 1 \pmod{10}$. It should be clear that $d_8 \not\equiv d_8 + 1 \pmod{10}$, so we conclude that $x_8 \neq d_8$. So x and r_8 are not equal!
Now, for r_9, the previous argument tells us that x_9 will be different than d_9, so $x_9 \neq r_9$. And we can repeat this for every single r_i in the enumeration. Since the enumeration is finite, there has to be a stopping point. Then by repeatedly applying the previous argument, we have that *every* number in the enumeration $r_i \neq x$. This is a contradiction, though, because $x \in (0, 1)$ (this should be obvious).
Thus, $(0, 1)$ is uncountable. □

Remark 6.3.1
Cantor's diagonal argument can be abstracted to show that other sets are uncountable. The general method is:

1. Assume the set is countable

2. Enumerate the set, because it is countable
3. Find an element in the set that is *not* in the enumeration

The last step is the "diagonal" part of the argument – the easiest way to find a new element is by constructing a new element that is different than every element in the set by at least one minute change.

Remark 6.3.2
When you apply Cantor's diagonal argument in your proofs, you can omit the lengthy explanations. All you need is the three parts above – in the third step, though, you should offer justification as to why your *diagonalizer* (the constructor function that gives you the new element) actually gives you a new element.

Now, we just showed that $(0, 1)$ is uncountable. But we also know that $(0, 1) \subseteq \mathbb{R}$. This subset relation intuitively suggests that $|(0, 1)| \leq |\mathbb{R}|$. So it is tempting to conclude from this that $\mathbb{R}$ is indeed uncountable. Luckily, this is exactly what we can conclude, but we need some theorems that will let us do that.

6.4 Useful Theorems

We give some useful theorems of countable and uncountable sets. These will help you in determining whether a set is countable or uncountable.

Definition 6.4.1 Cardinality
Two countable sets have the same cardinality if there exists a bijection between the two sets.

Remark 6.4.2
We had previously defined cardinality as the number of elements in a set. This new definition is more abstract and generalizable. In fact, though, these two definitions are equivalent on finite sets. Indeed, if two finite sets have n elements, then they are countable, and any enumeration of the two sets yields a bijection. On the other hand, if there is a bijection f between the two, then we can associate each input with an index $\leq n$, which yields the same index on the second set – so they have the same number of elements.

Remark 6.4.3
For countably infinite sets we will take this as a definition, since this is the first time we are seeing *number of elements* for countably infinite sets.

Remark 6.4.4
$|\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}| < |\mathbb{N}|$ even though the two sets are countable. The first is finite, but the second is countably infinite.

This tells us that our sets we have already proved countably infinite, $\mathbb{N}, \mathbb{Z}, \mathbb{Q}^+$, all have the same cardinality as $\mathbb{N}$ which is $\aleph_0$ since there is a bijection between $\mathbb{N}$ and those sets. This also implies that they all have the same cardinalities with each other. We can show this as a corollary to the following theorem.

6.4.1 Countable Set Theorems

Theorem 6.4.1
There exists a bijection between any countably infinite set.

Proof. Suppose A, B are countably infinite. Then there exists bijections f, g such that $f : \mathbb{N} \to A$ and $g : \mathbb{N} \to B$. Define the function $h : A \to B$ given by $h : a \mapsto g(f^{-1}(a))$. This is visualized as

$$A \overset{f^{-1}}{\to} \mathbb{N} \overset{g}{\to} B$$

This is a composition of bijective functions, which is a bijection. □

Theorem 6.4.2
Every countably infinite set has cardinality $\aleph_0$.

Proof. This follows from the previous theorem, and that a bijection between two sets implies their cardinalities are equal. □

Remark 6.4.5
It seems natural to also define $|A| \leq |B| \Leftrightarrow$ there exists an injection $f : A \mapsto B$.

Theorem 6.4.3
A *is countable* $\Rightarrow |A| \leq |\mathbb{N}|$.

Proof. A is countable, so there is a bijection from $\mathbb{N}$ to A. Then the inverse is also a bijection, which is also an injection. Then $|A| \leq |\mathbb{N}|$. Alternatively, A is countable so $|A| = |\mathbb{N}|$ so $|A| \leq |\mathbb{N}|$. □

Theorem 6.4.4
If a set B *is countable, then so is any subset* $A \subseteq B$.

Proof. Define a function $f : A \to B$ given by $f(a) = a$. Then this is an injective function (why?), so $|A| \leq |B|$. Then B is countable so $|B| = |\mathbb{N}|$.

Then $|A| \leq |B| = |\mathbb{N}|$. □

Theorem 6.4.5
A is countable $\Leftarrow |A| \leq |\mathbb{N}|$.

Proof. If $|A| \leq |\mathbb{N}|$, then there is an injective function $f : A \to \mathbb{N}$. Then let $B \subseteq \mathbb{N}$ be the image of f. Then by the previous theorem $\mathbb{N}$ is countable so B is countable. Then $|A| = |B| \leq |\mathbb{N}|$. □

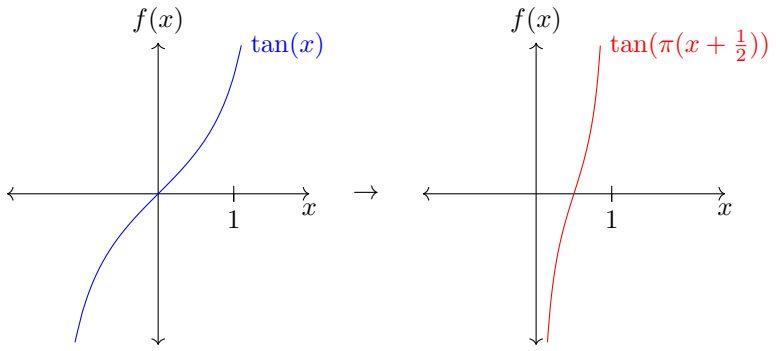
Remark 6.4.6
We conclude that A is countable $\Leftrightarrow |A| \leq |\mathbb{N}|$.

6.4.2 Uncountable Set Theorems

Theorem 6.4.6
If any set A is uncountable, then so is any superset $B \supseteq A$.

Proof. By contradiction, let $B \supseteq A$ and assume B is countable. Then by the previous theorem, $B \subseteq A$. But $A \subseteq B \subseteq A$, so $A = B$. But A is uncountable, so B is uncountable. But we assumed B was countable. This is a contradiction, so B is uncountable. □

Now we come back to $\mathbb{R}$ being uncountable. Since $\mathbb{R} \supseteq (0, 1)$, then since $(0, 1)$ is uncountable our theorems allow us to conclude that $\mathbb{R}$ is uncountable. Interestingly, we could also have given the bijection $f : (0, 1) \to \mathbb{R}$ given by $f : x \mapsto \tan(\pi(x + \frac{1}{2}))$. It looks like this:



6.5 Summary

- A countable set is either finite, or can be enumerated by the natural numbers
-
-

6.6 Practice

1. Explain why the following proof fails:

   > *Proof.* $\mathbb{R}^+ \times \mathbb{R}^+$ is countably infinite because we can set up a grid as we did in the $\mathbb{Q}^+$ argument and apply the same snaking bijection. □

2. Show that the even integers are countable by providing an explicit bijection from $\mathbb{N}$ to $\mathbb{Z}^{\text{even}}$.
3. Show that, given two countable sets A, B, the following are countable:
   (a) $A \cup B$ *Hint: how did we prove the integers were countable?*
   (b) $A \cap B$
   (c) $A \setminus B$
   (d) $A \times B$ *Hint: how did we prove the rationals were countable?*
4. Show that $\mathbb{Q}$ is countable.
5. Show that $\{x \in \mathbb{R} \mid x > 0 \wedge x^2 \in \mathbb{Q}\}$ is countable.
6. Explain why the following proof fails:

   > *Proof.* The set of all polynomials with positive integer coefficients is uncountable. Assume it is countable, and enumerate each polynomial as we did in Cantor's diagonalization argument – the rows are the polynomials, and the columns are the i^{th} coefficient in the polynomial (eg, ax^3, a is the third coefficient). Then the polynomial $f(x) = (p_{1,1} + 1) + (p_{2,2} + 1)x + (p_{3,3} + 1)x^2 + \cdots$, where $p_{i,i}$ is the i^{th} coefficient on polynomial p, is not in our enumeration. □

7. Show that the set of functions from natural numbers to natural numbers $\{f \mid f : \mathbb{N} \to \mathbb{N}\}$ is uncountable.
8. Show that the power set of any countably infinite set is uncountable. Do this by showing that $\mathcal{P}(\mathbb{N})$ is uncountable using a diagnalization argument.

9. Show that the power set of any set is uncountable.
10. Show that, given an uncountable set A and a set B,
    (a) $A \cup B$ is uncountable
    (b) $A \cap B$ is countable if and only if B is countable
    (c) $A \setminus B$ is uncountable if B is countable
    (d) $A \times B$ is uncountable
11. Show that the set of irrational numbers $\{x \in \mathbb{R} \mid x \notin \mathbb{Q}\}$ is uncountable.
12. Find another bijection from $(0, 1) \to \mathbb{R}$ that does not use a trigonometric function.

Chapter 7

Graph Theory

> The origins of graph theory are humble, even frivolous.
>
> Norman L. Biggs

Contents

7.1 Introduction

Graph theory is traditionally introduced using the Seven Bridges of Königsberg (see fig. 7.1). The scene is Königsberg, Prussia, 1700s. Leonhard Euler sees this area – two land masses separated by a river, with two islands in-between, all connected by 7 bridges – and thinks, *hmm, I wonder if I could walk through this city crossing each bridge only once.* Euler mathematically proved that no such walk exists due to the nature of this area. This was the birth of graph theory.

Euler did not brute-force the solution by examining all possible paths. Instead, Euler noticed that he could abstract out the large landmasses into single points, called vertices, and connect them by simple lines, called edges, which represent

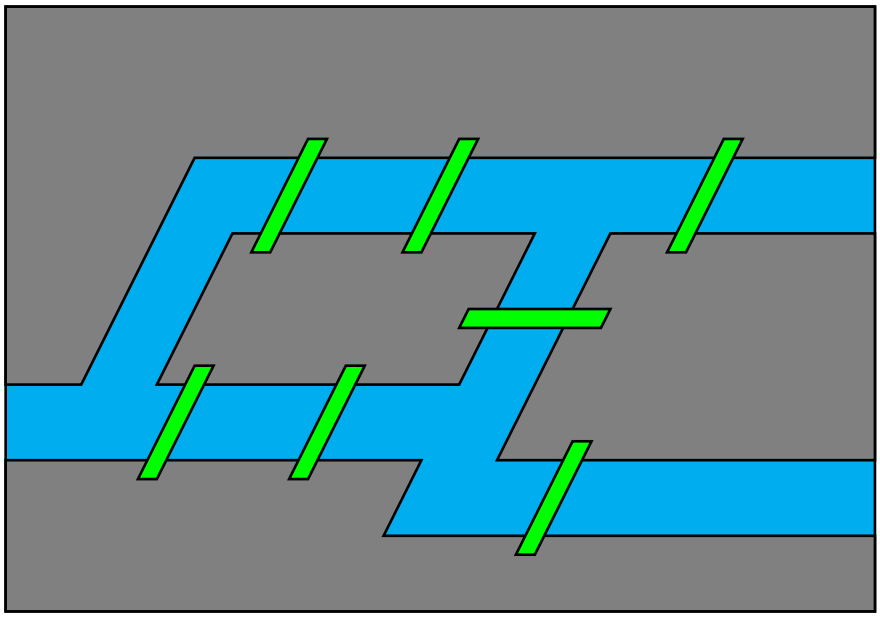


Figure 7.1: The Seven Bridges of Königsberg

the bridges. This mathematical structure is a *graph*. The graph version of the Seven Bridges is presented in figure 7.2.

The beautiful thing about graphs is that we can move the vertices around in space, and so long as the edges remain connected to their respective vertices, without changing the fundamental structure of the graph!

Graphs are applicable in numerous fields, including

- Network (computer, water, electric) flow
- Facebook friendships
- Google Maps path-finding
- Game AI path-finding
- E-commerce similarity predictions
- Neural networks
- ...

7.2 Key Terms

Definition 7.2.1 Graph
A set of abstract points, called vertices V, and connections between those points,

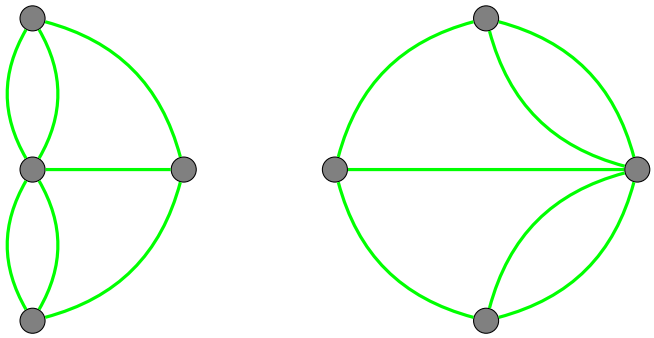


Figure 7.2: Two graphified versions of The Seven Bridges of Königsberg. The left graph is roughly a direct translation of the actual landmasses. The right graph is the same graph with the vertices moved around (take the middle-left vertex in the first graph, move it out to the left, then rotate the resulting graph 180 degrees)

called edges E. We denote a graph as $G = \{V, E\}$ or $G = (V, E)$. The order that you list V and E does not matter, however typically V is listed before E

Definition 7.2.2 Vertex

Definition 7.2.3 Edge
Directed vs Un-directed

Definition 7.2.4 Di-graph
A graph where each edge is directed.

Definition 7.2.5 Cycle
A thing. Only defined for directed edges

Definition 7.2.6 DAG
A Directed Acyclic Graph – a di-graph with no cycles

7.3 Graph Representations

There exist multiple ways to represent a graph. Each way has benefits and drawbacks. Typically algorithmic analysis involving graphs uses the representation that provides the best run-time.

Definition 7.3.1 Adjacency Matrix
An $m \times m$ (square) symmetric matrix M of 0s and 1s. Each row/column represents a node, and the position i, j in M represents the connection. For an edge

Definition 7.3.2 Adjacency List

There are other graph representations, such as the *edge list* and the *incidence matrix*. We encourage you to take a data structures course to learn more about these.

7.4 Problems

7.4.1 Graph Coloring

7.4.2 Network Flow

7.5 Summary

7.6 Practice

Chapter 8

Asymptotic Analysis

In algorithms, as in life,
persistence usually pays off.

Steven S. Skiena

Contents

8.1 Introduction

What is an algorithm?

Definition 8.1.1 Algorithm
A recipe/description of a process that accomplishes a goal

Why study algorithms? First, you will gain tools to *compare* algorithms. Second, studying algorithms will help you to creatively make *new* algorithms.

Why study algorithms in discrete math? Computers are discrete. Computers run programs, which are based on algorithms. Hence, algorithms must be discrete.

8.2 $\mathcal{O}$, Ω, and Θ Notations

Our motivation here is to describe the *growth-rate* of functions. The growth-rate class gives us key insights to what the actual function does on large inputs.

Definition 8.2.1 Big-Oh
An upper-bound on the growth-rate of a function. Formally:

$$f(n) \in \mathcal{O}(g(n)) \Leftrightarrow (\exists n_0 \in \mathbb{N}, \exists c \in \mathbb{R}^+, \forall n \in \mathbb{N}^{\geq n_0})[f(n) \leq c \cdot g(n)]$$

Definition 8.2.2 Big-Omega
A lower-bound on the growth-rate of a function. Formally:

$$f(n) \in \Omega(g(n)) \Leftrightarrow (\exists n_0 \in \mathbb{N}, \exists c \in \mathbb{R}^+, \forall n \in \mathbb{N}^{\geq n_0})[f(n) \geq c \cdot g(n)]$$

Definition 8.2.3 Big-Theta
An exact growth-rate of a function. Formally:

$$f(n) \in \Theta(g(n)) \Leftrightarrow f(n) \in \mathcal{O}(g(n)) \wedge f(n) \in \Omega(g(n))$$

Informally, $f(n) \in \mathcal{O}(g(n))$ just means that **eventually** (for big enough n) we have that (some constant-multiple of) g overtakes f. For example, $2^n \in \mathcal{O}(n!)$ since $2^n < n!$ ($n!$ overtakes 2^n) for $n \geq 4$ ($n \in \mathbb{N}$). So the growth-rate of 2^n is bounded *above* by $n!$. Similarly, $n! \in \Omega(2^n)$ since $n! > 2^n$ for $n \geq 4$. So $n!$ is bounded *below* by 2^n. Oftentimes in algorithm analysis we aim to show an algorithm has exact run-time (Θ), and we do this by showing it has both $\mathcal{O}$ and Ω run-time. Sometimes, though, this is hard to do.

Remark 8.2.4
You may see $f(n) = \mathcal{O}(g(n))$ instead of $f(n) \in \mathcal{O}(g(n))$. Both notations represent the same thing.

8.3 Analyzing Algorithms

We can express algorithms in terms of functions. The function input is typically n, the number of elements inputted into, or input size for, the algorithm, and the function output is typically time. Usually *time* will be the number of certain operations performed during the algorithm.

Example: Come up with a function $T(n)$ that describes the run-time of the following algorithm. Define run-time as the number of conditional evaluations. In the following algorithm, L is a list of integers of length n.

```
1: function GetMax(L, n)
2:     M ← −∞
3:     i ← 0
4:     while i < n do
5:         if L[i] > M then
6:             M ← L[i]
7:         end if
8:         i ← i + 1
9:     end while
10:    return M
11: end function
```

Solution: We evaluate the if-statement in line 5 exactly n times, and we evaluate the while-loop condition exactly $n+1$ times (the last time is when we break out of the loop). Thus, $T(n) = n + n + 1 = 2n + 1$

Example: Now come up with a function $U(n)$ that outputs the number of $\leftarrow$ operations. Compare $T(n)$ to $U(n)$.

Solution: We have two initial stores from lines 1 and 2. Each while-loop iteration has a store command in line 8. The while-loop runs exactly n times, so this gives us n stores. Line 6 has a store command which is run $\leq n$ times. We can model this using a natural number $c \leq n$ where c is the number of times line 5 evaluates to true. Our function is thus $U(n) = n + c + 2$
Since $c \leq n$ we have that $U(n) \leq T(n)$ for all $n > 1$. Note that when $n = 1$ we have $T(n) = 3$ and $U(n) = 4$ (since $c = 1$). In any case, we have that the growth-rates of both functions are solely dependent on n and are hence the same. Both functions are in $\Theta(n)$

Definition 8.3.1 The Loop-Heuristic
(algorithm analysis) For each nested loop in an algorithm, multiply n by the number of nested levels. This can be helpful in an initial analysis, however it will only take you so far.

Remark 8.3.2
We state the loop-heuristic as a definition, however it can be rephrased as a

provable theorem.

It is important to think about what is going on during the algorithm instead of blindly following the loop-heuristic. We present a few related examples.

Example: Analyze the following algorithm:

```
1: function AnalyzeList(L, n)
2:     M ← 0
3:     for i ← 0; i < n; i ← i + 1 do
4:         k ← L[i]
5:         for j ← i; j < n; j ← j + 1 do
6:             k ← k ∗ L[j]
7:         end for
8:         M ← M + k
9:     end for
10:    return M
11: end function
```

Solution: The algorithm contains 2 nested loops, so by the loop-heuristic our run-time is n^2. We examine further. The outer loop goes through the list n times. For each iteration i, we have a loop that goes through the list $n - i$ times. The actual *work* is done in the inner loop. Thus, the amount of work equals $1+2+3+\cdots+n = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$ by the Gaussian sum. This has a growth rate of n^2, so we see that the loop-heuristic worked and the algorithm runs in $\mathcal{O}(n^2)$. In fact, the amount of work does not change with respect to different input ordering, so the algorithm runs in $\Theta(n^2)$

Example: Examine the following algorithm:

```
1: function AnalyzeGraph(G = (V, E))
2:     c ← 0
3:     for v ∈ V do
4:         for w ∈ UnvisitedNeighbors(v) do
5:             Visit w
6:             c ← c + 1
7:         end for
8:     end for
9:     return c
10: end function
```

Solution: G represents a graph, where V is the set of vertices and E is the set of edges. We usually denote $|V| = n$ and $|E| = m$.
Let us first think about what the algorithm is *doing*. We instantiate $c = 0$, then go through each unvisited vertex we increment c. We consider the amount of work in the inner loop $\Theta(1)$, and we do this exactly $|V| = n$ times. So our algorithm runs in $\Theta(V)$ (or $\Theta(n)$).

Remark 8.3.3
How would this change had we instead iterated through the *neighbors*, instead of *unvisited neighbors*? We will come back to this

Okay, so how does one determine whether an algorithm is *good*? Well, it depends on how you define *good*. Typically we define *good* as having a low run-time. Unfortunately, an algorithm's run-time can change depending on the input. Remember: when writing algorithms, you cannot predetermine the input! This heeds way to the idea of *best-case*, *worst-case*, and *average-case* analysis.

Definition 8.3.4 Best-case Analysis
Analyzing an algorithm based on an optimal input that minimizes its run-time.

Best-case analysis is akin to bounding an algorithm's run-time function below, which is just determining the tightest Big-Omega bound. So, Ω essentially tells you how fast the algorithm will run in the best case.

Definition 8.3.5 Worst-case Analysis
Analyzing an algorithm based on a pathological input that maximizes its run-time.

By contrast, worst-case analysis is akin to bounding an algorithm's run-time function above, which is just determining the tightest Big-Oh bound. So, $\mathcal{O}$ essentially tells you how fast the algorithm will run in the worst case.

Definition 8.3.6 Average-case Analysis
Analyzing an algorithm's run-time based on typical, expected, input.

Remark 8.3.7
You may be tempted to think that, as the analogy would imply, average-case analysis is akin to Θ. This is not true. Θ tells you the *exact* run-time, which is when $\mathcal{O}$ and Ω agree. If you know your algorithm run-time has a Θ bound, then the average-case analysis will be this bound. The converse is not true though. If you have the average-case bound, this does not necessarily mean the worst-case or best-case is the same. An important example of this is the Quicksort algorithm.

We present a pathological example to explain why these different analysis types are important.

Example: Examine the following algorithm:

```
1: function PATHOLOGICAL(L, n)
2:     if n ≡ 0 (mod 2) then
3:
4:     else
5:         return L[0]
6:     end if
7:     c ← 0
8:     for v ∈ V do
9:         for w ∈ UnvisitedNeighbors(v) do
10:            Visit w
11:            c ← c + 1
12:        end for
13:    end for
14:    return c
15: end function
```

Solution:

8.3.1 Solving Summations

8.3.2 Solving Recurrences

8.4 Common Complexities

There are a handful of common run-time complexities that you should be familiar with. In this book, we give them to you in ascending order of growth rate in Big Theta notation.

Definition 8.4.1 Constant

$$\Theta(1)$$

Definition 8.4.2 Logarithmic

$$\Theta(\log n)$$

Definition 8.4.3 Linear

$$\Theta(n)$$

Definition 8.4.4 Polynomial

$$\Theta(n^c),\ c > 1$$

We call $\Theta(n^2)$ Quadratic

Definition 8.4.5 Exponential

$$\Theta(c^n),\ c > 1$$

8.5 P vs NP

There exist many YouTube videos discussing the famous P vs. NP problem (ex 1, ex 2). Moreover, if you solve it, the Clay Mathematics Institute has $1,000,000 waiting for you. So what is it?

Definition 8.5.1 P
The class of computational problems that can be solved in *P*olynomial time.

Definition 8.5.2 NP
The class of computational problems where a given solution can be *verified* in polynomial time.

Remark 8.5.3
NP stands for *Nondeterministic Polynomial*. It does **not** stand for *non-polynomial* as some people think.

Definition 8.5.4 Reductions

Definition 8.5.5 NP-Complete
A computational problem is NP-Complete if it satisfies two conditions:

1. The problem is in NP
2. Every problem in NP is reducible to it in polynomial time

Remark 8.5.6
Problems that satisfy the second condition above, and not necessarily the first, are called **NP-Hard**.

The second condition is tricky

Theorem 8.5.1 The Cook-Levin Theorem
The Boolean satisfiability problem is NP-Complete.

Remark 8.5.7
While at this point in the course you likely can understand the proof, we omit it for scope.

We leave this discussion with one final remark. P = NP will render most (all) cryptographic algorithms and services broken. This is because most modern cryptography is built on the observation that factoring very large primes is *hard*. If you are interested, look into RSA and/or the Diffie-Hellman key exchange.

8.6 The Halting Problem

Infinite loops are a software developer's worst nightmare. They are hard to find, debug, and (sometimes) fix. If only there were a piece of software that could examine your code and tell you whether there are any infinite loops. More abstractly, if it could tell you whether your program continues forever or eventually stops.

Well, we are sorry to break the news to you, but this piece of software can never exist; sort-of.

The halting problem is most commonly used as an introduction to computability theory. What problems can we compute? What does it even mean to be computable? These questions are out of scope of this book, but if the following problem interests you, consider taking a course on theory of computation.

Definition 8.6.1 The Halting Problem
A computational problem of determining whether an arbitrary computer program, with input, will stop or run forever.

Think to yourself whether you can come up with a solution program. It is a difficult problem. Alan Turing proved in 1936 that no such algorithm could exist *for all* pairs of programs and inputs. This proof led to (created) the Turing machine, and arguably all of modern computer science. The proof sketch follows. We call it a sketch because the actual proof is significantly more in-depth.

Proof. Assume that such an algorithm (code description, function, method, whatever you want to call it) h exists. The function $h : P \times I \to B$ given by $h : (p, i) \mapsto b$ tells us whether a given input program $p \in P$ (the set of all programs) with input $i \in I$ (the set of all program inputs) will halt (stop) or not: $b \in B = \{\text{True}, \text{False}\}$.
Then the following procedure $\text{G} \in P$:

```
1: function G(·)
2:     if h(G, ·) then
3:         while True do                    ▷ loop forever
```

```
    end while
  end if
end function
```

We have two cases: $h(\text{G}, \cdot)$ returns True, or $h(\text{G}, \cdot)$ returns False.
Case 1: $h(\text{G}, \cdot)$ = True. Then G stops. But since h returned True, the body of the if-statement will be executed. Then G will loop forever. But this contradicts the output of h.
Case 2: $h(\text{G}, \cdot)$ = False. Then G does not stop. But since h returned False, the body of the if-statement is *not* executed, so G will stop. But this contradicts the output of h.
Thus, the halting problem is undecidable (impossible to answer for all inputs). □

8.7 Summary

-
-
-

8.8 Practice

1. Analyze the following algorithm:
   ```
   function SomeFunc(n)
       something
   end function
   ```
2. Order the following complexities: $\Theta(n \log n)$, $\Theta(n \log(n^2))$, $\Theta(n)$, $\Theta(n^2)$. Explain why your ordering is correct.
3. Order the following complexities: $\Theta(\log(3n))$, $\Theta(n^n)$, $\Theta(n!)$, $\Theta(\log(\log n))$.

Chapter 9

Conclusion

So, that's it?

Justin Goodman

Yes, that is it. That is Discrete Mathematics. You did it, congratulations!

9.1 Closing Remarks

The fun does not stop here – there are many more topics in discrete mathematics! This book is geared towards students in an introductory proof course. One can also go arbitrarily deep in any topic listed in this book. Mathematics is beautiful in this sense – there is no endpoint. There will always be a bigger number. There will always be a more difficult unsolved problem in mathematics. There is no ending in sight, but this infinite journey is worth more than the finale.

9.2 Tips

Here is a compiled list of tips. There are tips for each topic related to taking a course corresponding to this book. Each list is in no particular order.

9.2.1 Studying

- Take advantage of the Spacing Effect – spread your studying out over time.
- Utilize the Pomodoro Technique. Repeat the following process: study for 25 minutes, take a 5 minute break, repeat 4 times, take a longer 15-30 minute break.

- Teach your peers about what you are studying – if you can teach someone a topic, then you understand that topic.

9.2.2 Assignments

- Start early.
- Work with others.
- Struggle until you understand.
- Seek help, not solutions.

9.2.3 Exams

- Get plenty of sleep before the exam. 8 hours of sleep with 2 hours of studying is significantly better than 2 hours of sleep with 8 hours of studying.
- Eat food, drink water, exercise, and use the bathroom before the exam. Avoid drugs.
- Be aware of each concept, even if you forgot to study it. If on the exam you come across something you are rusty with, then write down as much as you know about the problem.
- Write neatly.
- Often, your first instinct answer is the correct answer. Stay away from over-analyzing a question.
- During the exam when you have time, re-read each question and make sure you are actually answering the question.
- If you read a problem and the answer does not immediately jump out at you, or if you get stuck on a problem, mark it and move on. Sometimes other exam questions will give you the insight you need to go back and answer the marked problem.
- Relax.
- Plan out how you want to attack this exam.
- **Remember the big picture of life.**

9.3 Summary

- This is a long and difficult course
- Be proud of yourself
- You got this