DANIEL GONZALEZ CEDRE

DISCRETE MATHEMATICS

UNIVERSITY OF NOTRE DAME

January 24, 2024

These notes are intended for students of CSE 20110 Discrete Mathematics at the University of Notre Dame.

https://daniel-gonzalez-cedre.github.io

Contents

Foreword

COLOR	INTERPRETATION
	Emphasis
	Pronunciation
	Definition
	External link
	Internal link

Table 1: Color legend.

SYNTAX	TRANSLITERATION	SEMANTICS
$\top$	*"True."*	A true sentence; a tautology.
$\bot$	*"False."*	A false sentence; a contradiction.
$x \coloneqq y$	*"x is, by definition, y."*	The name x has been assigned to the object referenced by y.
$x = y$	*"p equals q."*	p and q refer to the same object.
$p \equiv q$ $p \Leftrightarrow q$	*"p is equivalent to q."* *"p if, and only if, q."*	The sentence p is logically equivalent to the sentence q.
$p \vdash q$ $p \Rightarrow q$	*"p proves q."* *"p implies q."*	By assuming the sentence p, we can prove the sentence q.
$\varnothing$	*"The empty set."*	The set containing no elements.
$\{a, b, c\}$	*"The set containing a, b, and c."*	The object whose elements are a, b, c.
$\{x \mid \varphi(x)\}$	*"The set of all x such that $\varphi(x)$."*	The object whose elements consist of all possible objects x for which the sentence $\varphi(x)$ is true.
$\{x \in \mathcal{A} \mid \varphi(x)\}$	*"The set of all x in $\mathcal{A}$ such that $\varphi(x)$."*	The object consisting of all x from $\mathcal{A}$ for which the sentence $\varphi(x)$ is true.
$f : \mathcal{A} \to \mathcal{B}$	*"f is a function from $\mathcal{A}$ to $\mathcal{B}$."*	A function named f with domain $\mathcal{A}$ and codomain B.
$f(x)$	*"f of x."*	The output of f on the input x, where $x \in \mathcal{A}$ and $f(x) \in \mathcal{B}$.
$\mathbb{N}$	*"enn"*	The set of natural numbers.
$\mathbb{Z}$	*"zee"*	The set of integers.
$\mathbb{Q}$	*"queue"*	The set of rational numbers.
$\mathbb{R}$	*"arr"*	The set of real numbers.

Table 2: An overview of some important notation. Note that some expressions, like $p \equiv q$ and $p \vdash q$, have more than one equivalent notation. The middle column gives some common ways of *reading* each notation in English. The last column provides the *meaning* of each expression.

GLYPH	NAME	IPA
$A\ \alpha$	Alpha	[a]
$B\ \beta$	Beta	[v]
$\Gamma\ \gamma$	Gamma	[ɣ]
$\Delta\ \delta$	Delta	[ð]
$E\ \varepsilon$	Epsilon	[e]
$Z\ \zeta$	Zeta	[z]
$H\ \eta$	Eta	[ɛː]
$\Theta\ \theta$	Theta	[θ]
$I\ \iota$	Iota	[iː]
$K\ \kappa$	Kappa	[k]
$\Lambda\ \lambda$	Lambda	[l]
$M\ \mu$	Mu	[m]
$N\ \nu$	Nu	[n]
$\Xi\ \xi$	Xi	[ks]
$O\ o$	Omicron	[o]
$\Pi\ \pi$	Pi	[p]
$P\ \rho$	Rho	[r]
$\Sigma\ \sigma$	Sigma	[s]
$T\ \tau$	Tau	[t]
$\Upsilon\ \upsilon$	Upsilon	[yː]
$\Phi\ \varphi$	Phi	[f]
$X\ \chi$	Chi	[k^h]
$\Psi\ \psi$	Psi	[ps]
$\Omega\ \omega$	Omega	[ɔː]

Table 3: The Greek alphabet.

Logic

0

Language

> *"No language is justly studied merely as an aid to other purposes. It will in fact better serve other purposes, philological or historical, when it is studied for love, for itself."*
>
> – J. R. R. Tolkien

We communicate our thoughts to others with the use of language. This is worth reflecting on. You are probably reading this because you have some interest in computation, mathematics, logic, or are incurably bored; the goal of these notes is—in part—to provide the mathematical background necessary to study these fields at a higher level. This is particularly true for aspiring *computer scientists,* who may have some misconceptions about their field because of its misleading name,[1] and who may not be aware that the field properly and historically falls under the grand umbrella of *mathematics.*

This ambitious undertaking must therefore involve engaging with the tumultuous and violent history of mathematics. Although modern computer science is now richly interdisciplinary, the field was born during a particularly turbulent period in the late 19th and early 20th centuries AD[2] agitated by an existential crisis in mathematics: a crisis caused by our flagrant use of language. Here's a short summary.

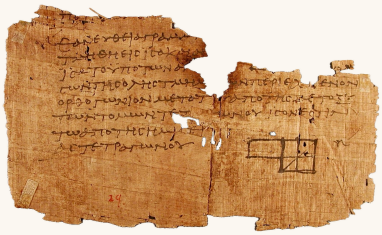

Figure 1: A fragment of book 2 from Euclid's *Elements* taken from the Oxyrhynchus papyri, dated ca. 100 AD.

[1] It's *not* about computers *nor* is it science.

[2] We will see later that its roots span at least to the time of Euclid in 300 BC.

0.1 A Brief History of...

The serious study of rhetoric—the art of argumentation and persuasion—as a subject in its own right dates back to at least the 5th century BC.[3] Around the 3rd century BC, Euclid's 13 books of the *Elements* heralded the birth of geometry, algorithmic computation, and the first theory of numbers,[4] where he *proved* certain statements followed from a list of *axiomatic* assumptions. This was a great achievement, establishing mathematical *proof* as a form of *argumentation* that logically deduces conclusions from a list of common assumptions. The contemporaneous Greek philosopher Theophrastus further pushed the envelope by describing the *form* of these arguments and establishing their validity.

[3] The time of the ancient Greek sophists, who were notably opposed by Socrates, Plato, and Aristotle.

[4] The only evidence of algorithms before this time—for multiplying, factoring, and finding square roots—dates back to Egypt and Babylon before 1600 BC.

The ancient Greeks laid the foundation for the two instrumental aspects of mathematical thought: *abstraction* and *argumentation*. Euclid abstracted what were thought to be the fundamental truths of geometry into a list of 12 *axioms*[1] so that, instead of thinking about *that* particular wall or *that* particular stick or *that* particular roof, he could make statements and observations about *quadrilaterals*, and *lines*, and *triangles* in general. These axioms were meant to encode the *universal truths* of geometry: the nature of what it fundamentally means to construct and measure distances, angles, and (simple) shapes. The last of these axioms would quickly become infamous.

axiom

[1] An *axiom* is a statement that we assume is true without justification nor proof.

Axiom (Parallel Postulate).
If two straight lines meet a third straight line making two interior angles that are each less than right angles, then the two lines—if they were to be extended—must intersect on that side where the interior angles are.

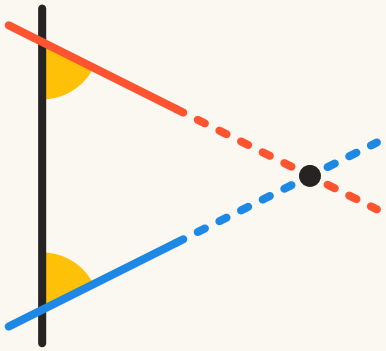

Figure 2: The parallel postulate says that any two lines and that make acute interior angles and with a third line must intersect at a point.

If you stop to think for a moment, this postulate says something very obvious. Assuming all of Euclid's other axioms, there are a few *equivalent* ways to restate the parallel postulate:

1. For any line L and point P not on L, there is exactly one line parallel to L passing through P.
2. The sum of interior angles in any triangle is 180 degrees.
3. A right triangle with side lengths A, B, C satisfies $A^2 + B^2 = C^2$.

You'll recognize this third statement as the Pythagorean *theorem*,[2] which is not merely an assumption![3] For the next 2000 years, the mathematical community was haunted by the thought that it was possible to *prove* the parallel postulate using the other axioms. It seemed like the rest of the axioms did such a perfectly good job of characterizing geometry that the parallel postulate *must necessarily* follow from the other axioms.

[2] A *theorem* is a statement that has a proof.

[3] The first two are called Playfair's axiom and the triangle postulate respectively.

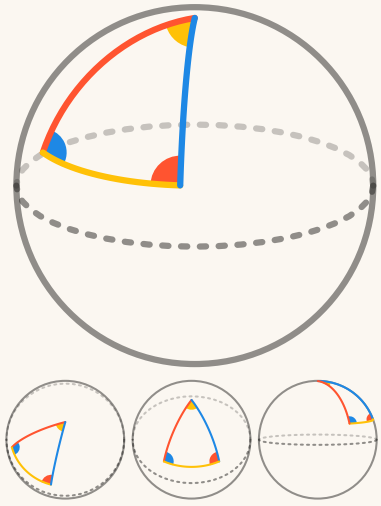

Figure 3: Four views of the same triangle whose angles sum to 270 degrees. Notice how the notions of *straight* and *parallel* differ on the surface of a sphere.

However, between 1810–1832 AD, no less than *three* papers on *hyperbolic* geometry were published, and by 1854 Bernhardt Riemann had developed a theory of *Riemannian* geometry on manifolds. These were all different examples of consistent models of geometry that *denied* the parallel postulate! These ideas were intensely contested: many mathematicians and natural philosophers of the time refused to accept the notion that geometry could be non-Euclidean because *it went against their intuitive notion of how geometry should behave.*

This whole ordeal was only foreshadowing what would come at the turn of the century. In 1874, Georg Cantor would make a series of discoveries[4] surrounding the nature of infinity so fundamentally opposed to common mathematical thought that he would be antagonized and ostracized for decades, causing him to suffer serious depressive crises.

[4] We will study these later.

Once again, mathematicians' *intuitive notions of how infinity should behave* were shown to be wrong. Cantor's discoveries sparked not only a civil war within the mathematical community but also a concerted effort by many mathematicians and logicians in the early 20th century to *fix* mathematics by establishing it on a firm *foundation.*

The cause of all this turmoil was, fundamentally, a *lack of precision and rigor* in the way people would communicate mathematical ideas and arguments. What does it *mean* for a line to be straight, or for two straight lines to be parallel? What does it *mean* to have two lines, or to have infinitely many lines? What *is* infinity? Is infinity a number? What *are* numbers? How do we *know* we are saying anything *true* at all?

If we hope to answer any of these questions, we must first develop a language for *precise* mathematical communication. This necessarily begins with a systematic deconstruction and analysis of *language* itself.

0.2 Syntax and Semantics

semantics

syntax

variable

term

Languages encode ideas into sequences of symbols.[1] These symbols represent objects, ideas, actions, and concepts. The *meaning* behind a particular cluster of symbols is called its *semantics*. The *form* the language takes, dictated by its *grammatical rules* for composing symbols into valid sentences, is called its *syntax*. We refer to objects by giving them names. A *variable* is a symbol[2] that stands in place for an object that has not been determined yet.[3] We can assign name to a *particular* object with the $:=$ symbol. We call these the *terms* of an expression.

[1] For our purposes, we will focus only on written—as opposed to spoken or signed—languages.

[2] We typically denote variables using single Latin or Greek letters, though there are no strict universal rules. Some common examples are listed below.

- $a, b, c, i, j, k, \ell, m, n, p, q, u, v, w, x, y, z$
- $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{G}, \mathcal{H}, \mathcal{M}, \mathcal{N}, \mathcal{R}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$
- $\alpha, \beta, \gamma, \delta, \epsilon, \eta, \theta, \lambda, \mu, \pi, \sigma, \tau, \varphi, \psi, \omega$

[3] A variable does not *necessarily* refer one particular object, or even any object at all.

***Definition* 0.1 (Sentences).**

sentence

atomic

A *sentence* is the expression of a complete thought or idea in accordance with the syntactic and grammatical rules of a given language. A statement is called *atomic* if it can't be broken down into smaller semantic components in any way that obeys the language's syntax and grammar.

1. A *declarative* sentence is one that describes something. They typically consist of a *subject* being described and a *predicate* property it has.
2. An *interrogative* sentence asks a non-rhetorical question.
3. An *imperative* sentence heralds a command or request.

"Oft hope is born when all is forlorn."

"What has it got in its pocketses?"

"Keep your forked tongue behind your teeth."

Mathematical practice principally involves *making and justifying observations about mathematical objects.*[4] As such, we are only really interested in crafting *declarative* sentences—sentences that describe *terms*. We will systematically deconstruct and analyse these kinds of sentences, extract their *logical essence*, and build up a new language for precisely communicating mathematical arguments.

[4] We leave the problem of *what* a mathematical object actually *is* for later.

0.3 A Recurring Theme

Before going any further, we should make a brief detour to discuss a topic that lies at the *heart* of computing, logic, and the 20th century foundational crisis in mathematics: *recursion*. In a very strong sense, what we *mean* when we say that some *thing* is *computable* is that there is a *recursive procedure* that produces that *thing*.

Idea (Church-Turing Thesis). Informally, we say something is *computable* if it can be obtained as the result of a *recursive process*, or as the output of a *Turing machine*, or expressed as a *term in the λ-calculus*.

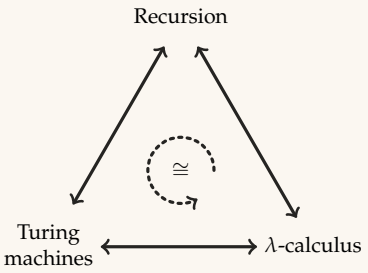

Figure 4: The Church-Turing thesis states that these three concepts—which are all *formally* equivalent—correspond with our *informal* notion of computability. In modern times, many people now take this as a definition for computability.

[1] We will leave Turing machines and the λ-calculus for a future time.

Actually, the three concepts described above are all *equivalent* to each other. It should then be no surprise that *recursion* (and its twin *induction*) will play a central role in our studies, so we will take this brief moment to quickly describe the fundamental idea at behind recursion.[1]

First, an example: how do we *compute* the sum of a list of n numbers?

$$3+5+9+2$$

With some hard work and determination access to the internet, we can see that $3+5+9+2=19$, but *how* did we get that answer? At the most basic level, we started by taking two of the numbers, 3 and 5 say, computing their sum $3+5=8$, and adding this intermediate result to another number from the list, 9 say, to get $8+9=17$, and adding that again to yet another element of the list—in this case, only 2 remains—to finally arrive at $17+2=19$.

$$\begin{aligned} 3+5+9+2 &= 3+5+9+2 && (1)\\ &= 8+9+2 && (2)\\ &= 8+9+2 && (3)\\ &= 17+2 && (4)\\ &= 17+2 && (5)\\ &= 19 && (6) \end{aligned}$$

This might seem so obvious it physically hurts, but let's analyse what we just did more closely. Suppose we have a list of n arbitrary numbers.[2]

$$x_0+x_1+x_2+\cdots+x_{n-2}+x_{n-1}$$

Once again, we begin by taking the first two numbers and computing x_0+x_1, then adding *this* result to x_2, then adding *that* result to x_3, then adding *that* result to x_4, and so on until we reach the end of the list. So, in order to compute $x_0+x_1+x_2+\ldots x_{n-2}+x_{n-1}$, we *first* need to compute $x_0+x_1+x_2+\ldots x_{n-2}$ and then add that result to x_{n-1}.

[2] Notice that, being the sophisticates we are, we start counting at 0, so that a list of n numbers will be indexed starting at 0 and ending at $n-1$.

But wait, isn't $x_0 + x_1 + x_2 + \ldots x_{n-2}$ also the sum of a list? It is, it's just that the list has one less element! So *how do we compute the sum of elements in a list*? We first *compute the sum of elements in a list*, and then add one more element to that result. So, it seems like in order to do what we want, we need to already know how to do what we want; the key here is that we only need to know how to sum the elements of a *smaller* list in order to get the result we want for the *larger* list. As long as we can *eventually* get a result for one of these *"smaller"* sums, we will be able to build up a solution to our original problem by passing this result *"back up"* the chain of computation. Back to our first example.

$$
\begin{aligned}
3+5+9+2 &= 3+5+9+2 && (1)\\
&= 3+5+9+2 && (2)\\
&= 3+5+9+2 && (3)\\
&= 8+9+2 && (4)\\
&= 17+2 && (5)\\
&= 19 && (6)
\end{aligned}
$$

This paragraph describes the *recurrence.*

Steps (1) through (3) continually decompose the given list into sublists on the left until we have no more lists we can break up. Each one of these lists is a smaller version of the original problem, and we compute the sums of these smaller lists by breaking them down and computing *their* sublists' sums, recombining these results at the end.

This paragraph encounters the *basis.*

This now brings us to an important point: *we can't decompose* 3 *any further,* because this list only has one element in it. Do we know what the sum of all numbers in a list with one element is? Of course we do: it's just *that* number. Now we can return this result *back up* to the 5 that was waiting to be added to it, and when we add them together, we can return *that* result back to the 9 that was waiting, and then return *that* result to the 2 that was waiting, finally letting us conclude that the sum over the whole list is 19. The *recurrence relation* below summarizes this.[1]

recurrence relation

$$
\texttt{sum}(x_0, x_1, \ldots x_{n-1}) = \begin{cases} 0 & \text{if } n = 0 \\ \texttt{sum}(x_0, x_1, \ldots x_{n-2}) + x_{n-1} & \text{if } n \geqslant 1 \end{cases}
$$

[1] Notice that this is actually written slightly differently than the procedure we've just described; think about *how* this is different and whether or not it actually computes the same result as the procedure we were just analysing.

We've exposed here a *recurrence* and a *basis*—the two key components underlying recursion (and, later, induction). The *recurrent* part of this procedure explains how to express a problem in terms of *"smaller"* instances of the *same problem,* describing how to combine the solutions to those subproblems into a solution for the original problem. Obviously, though, if you just keep decomposing problem into subproblems forever, you'll never be able to actually generate an answer to anything. Eventually, you need to *stop* and actually say what the answer to something is. The *basis*, does exactly this by providing explicit answers to the *smallest* versions of the problem.[2]

recurrence

basis

[2] This is sometimes called the *base case.*

1

Zeroth-Order Logic

"The limits of my language means the limits of my world."
– Ludwig Wittgenstein

Figure 1.1: Illustration by Rockwell Kent from "Moby Dick: or, The Whale."

As we saw in the previous chapter, sentences can be broadly classified based on the kind of information they convey—their *functional role* in language. How do we begin deconstructing the descriptive fragment of our language? Naturally, we can think to classify the descriptive sentences by asking the fundamental question: *is this description true?*

1.1 *Truth Values*

Let's consider the following declarative sentence.

"Ahab is a captain." (1.1)

Here we have a descriptive sentence about the term *Ahab*—a man and thus an object of our discourse—asserting he *is a captain.* In the context of Herman Melville's *Moby Dick,* this is an accurate description. Referring to the above sentence as $\sigma_{1.1}$, we would then say $\sigma_{1.1}$ is *true*. We introduce the symbol $\top$ to denote these kinds of sentences.

true
$\top$

"Ishmael is a whale." (1.2)

The above sentence, however, which we will name $\sigma_{1.2}$, immediately furrows the brow and strikes at the heart of our conscience. We know from the story that Ishmael is a sailor, and thus human, and therefore *not* a whale! We should then want to say that $\sigma_{1.2}$ is *false*, reserving the symbol $\bot$ for sentences of this kind.

false
$\bot$

The symbols $\top$ and $\bot$ are also sometimes called *"top"* and *"bot"* respectively.

The attributes *true* and *false* that we are attaching to these sentences are what we call *truth values*, and they are *the* essential component of the kinds of sentences we want to express. Sentences that are *true* all exhibit a quality that makes them similar to each other but dissimilar to *false* sentences, regardless what the actual sentences themselves *mean*

truth value

semantically. What we've just done is *abstract* the fundamental concept of truth value from descriptive sentences. This abstraction allows us to notice that *all true sentences are essentially the same as each other,* at least from the perspective of their truth values, with the same applying to *false* sentences. On the other hand, *true* and *false* sentences are complete opposites. This relationship inspires our first definition below.

Definition 1.1 (Propositional Equivalence).
equivalence
$\equiv$
We say that two sentences φ and ψ are *logically equivalent* when they have the same truth value. We denote this by writing $\varphi \equiv \psi$.[1]

[1] *"φ is logically equivalent to ψ."*

With this new definition, we can formalize our observations from the preceding paragraph as $\sigma_{1.1} \equiv \top$ and $\sigma_{1.2} \equiv \bot$ as well as $\sigma_{1.1} \not\equiv \sigma_{1.2}$. Notice that each of these three expressions is a complete sentence describing properties[2] held by some objects.[3] In fact, these statements were themselves *true* declarative sentences. Now, let's ponder the following sentence, which we will call $\sigma_{1.3}$.

[2] being (or not) logically equivalent

[3] the sentences $\sigma_{1.1}$ and $\sigma_{1.2}$

"Colorless green ideas sleep furiously." (1.3)

Like the previous examples, this is a grammatically correct, declarative sentence, but what does this sentence *mean?* Is it *true?* Is it *false?* Taking the normal English definitions for each of the words in this sentence, it doesn't seem to make any sense. We then clearly can't call it an accurate description of anything, so it can't possibly be *true.* Does that mean it must be *false?* Well, if we assume it is *false,* then what about the following sentence?

"Colorless green ideas *do not* sleep furiously." (1.4)

This one, which we will call $\sigma_{1.4}$, seems to be saying the opposite of whatever $\sigma_{1.3}$ was saying, so if the other one is *false,* then this one must be *true.* The question then becomes: what is $\sigma_{1.4}$ accurately describing? *This* sentence seems to make just as little sense as the original! This should lead us to conclude that $\sigma_{1.3}$ could not have been *false* either, so
nonsense
that sentence *has no truth value!* We call expressions like this *nonsensical* because they *carry no semantic meaning.*

Let's now analyse the following statement, which we will call $\sigma_{1.5}$.

"This sentence is *false.*" (1.5)

Expressed a little more *formally,* this is the sentence—named $\sigma_{1.5}$—that says $\sigma_{1.5} \equiv \bot$. This certainly doesn't seem like nonsense; it says something clear about a well-understood object. So, what is the truth value of this sentence? We can try reasoning about this like we did before by examining the two possible truth values the $\sigma_{1.5}$ can take.

First, let's assume $\sigma_{1.5}$ is *true*, which we write formally as $\sigma_{1.5} \equiv \top$. By definition, this would imply $\sigma_{1.5}$ is an accurate description of some object, so we should believe what the sentence says about that object. In this case, the object is $\sigma_{1.5}$ and the description is that $\sigma_{1.5} \equiv \bot$. This *contradicts* our initial assumption! ↯ Therefore, $\sigma_{1.5}$ is *not true!*[1]

[1] We conclude this because this is the opposite of our initial assumption, which lead us to a contradiction.

That rules out one truth value. What happens then if we assume $\sigma_{1.5}$ is *false?* Again, we can write this formally as $\sigma_{1.5} \equiv \bot$. By definition, this implies we should *reject* what $\sigma_{1.5}$ is asserting, leaving us with $\sigma_{1.5} \not\equiv \bot$. As before, a *contradiction* emerges! ↯ Therefore, $\sigma_{1.5}$ is *not false* either!

From this simple analysis, we can see that $\sigma_{1.5}$ *does not have a truth value!* Sentences that *contradict themselves* like this are called *paradoxes*.[2] In the preceding analysis, we relied on the idea that $\top$ and $\bot$ are opposed to each other, so that the same sentence can't meaningfully be both $\top$ and $\bot$ at the same time. This should be intuitive based on our natural understanding and usage of the words *true* and *false*, but we will make it a point to *formally* introduce this idea now.

paradox

[2] The word *paradox* is unfortunately overload and context-dependent. When referring to specific sentences, we will use it to specifically mean a self-contradictory sentence such as $\sigma_{1.5}$, but it is also commonly used in some contexts to refer to situations that are simply *unintuitive* rather than outright contradictory.

Axiom (Principle of Bivalence).
Any sentence expressing a truth value is either *true* or *false* but not both.

What this analysis has hopefully shown us is that *not every* well-formed, declarative sentence expresses a truth value. In order for a sentence to express a truth value, it must satisfy the following three properties.

1. The sentence must be grammatically well-formed.
2. The sentence must be declarative.
3. The sentence must be semantically meaningful.

These are the kinds of statements are *eligible to carry a truth value*—the ones for which *it would make sense* to say they are either *true* or *false*—so they will form the foundation of our new language. We will eventually call these *propositions*, but beware that this is not (yet) a *formal* definition of what a proposition is. First, we need to get a better sense of *what* propositions are linguistically and *how* they are formed.

1.2 *Logical Connectives*

The examples of sentences we've seen so far have all been *atomic*—meaning they can't be broken down into simpler sentences that themselves are complete thoughts—but we can obviously express thoughts that are more than merely atomic. These *compounded* propositions are formed by taking smaller propositional sentences and *connecting* them together based on what our intended meaning is.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \to q$	$p \leftrightarrow q$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$

Table 1.1: A truth table summarizing the basic connectives of classical logic. The two left-most columns represent the *input* values of the propositions p and q. The remaining columns describe the *output* of each expression given the corresponding inputs on each row.

Each of these different ways of connecting sentences together suggests a different way of *transforming* between truth values by combining the truth values of the component propositions into a truth value for the compound expression.

logical connective

In this section, we will uncover these different transformations—which we will call *logical connectives*—and encode them using *truth tables*, which specify the output truth values for every combination of inputs.

Table 1.2: Truth table for negations.

p	$\neg p$
$\top$	$\bot$
$\bot$	$\top$

Negations

Suppose we encountered the following sentence, which we call $\sigma_{1.6}$.

"Espresso is not delicious." (1.6)

Immediately, the moral observer will realize the offensive absurdity of this sentence, compelled by the force of conscience to declare $\sigma_{1.6} \equiv \bot$! With this, we could simply carry on with our day; however, pausing to think for a moment, we can see that $\sigma_{1.6}$ is intimately related to the following (much more pleasant) sentence, which we call $\sigma_{1.7}$.

"Espresso is delicious." (1.7)

negation

This sentence is *clearly true*, letting us sigh $\sigma_{1.7} \equiv \top$ in relief. Not only that, it is the saying exactly the opposite of what $\sigma_{1.6}$ asserted! We call propositions like these *negations* of each other. This is our first example of a *transformation* of truth value: the negation of a proposition is another proposition with the opposite truth value. To denote this formally, we introduce the $\neg$ symbol, allowing us to write $\sigma_{1.6} \equiv \neg\sigma_{1.7}$.

$\neg$

We can now think of $\neg$ formally as a *unary function* that operates on truth values.[1] This function works by mapping $\neg\top$ to $\bot$ and by mapping $\neg\bot$ to $\top$. This gives us a way of abstracting negations at the level of truth values, so that we can formally define what it means to *negate* a proposition. We provide this definition now in Table 1.2, where the left-most columns[2] represent the inputs to $\neg$ and the right-most columns show the truth value of the resulting output expression.[3]

[1] A function is *unary* if it takes only one input argument. We will study functions in more detail later.

[2] shown with white backgrounds

[3] shown with colored backgrounds

Conjunctions & Disjunctions

But we can obviously connect two (and sometimes more) sentences together to create larger sentences in English. For example,

"Espresso is delicious, and it nourishes the soul." (1.8)

This sentence is composed of two smaller atomic sentences, namely *"espresso is delicious"* and *"espresso nourishes the soul,"* which we know are both independently *true.* Connecting them together with the word *"and"* should then, based on the way this word works in English, produce another *true* sentence. Conversely, if either of the subexpressions had been *false,* the compound result should also be *false.* This *binary* connective is called the logical *conjunction*, and we denote it using the $\wedge$ symbol. It is defined in Table 1.3.

conjunction
$\wedge$

Table 1.3: Truth table for logical conjunctions and disjunctions.

p	q	$p \wedge q$	$p \vee q$
$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$

There are several distinct ways this connective can appear in English that are nonetheless equivalent. Some examples are listed below.

"Espresso is delicious, *and* it nourishes the soul."
"Espresso is delicious *and* soul-nourishing."
"Espresso is delicious, *but* it nourishes the soul."
"Espresso is delicious, *yet* nourishing to the soul."
"Espresso is delicious; *further*, it nourishes the soul."
"*Although* espresso is delicious, it *also* nourishes the soul."

Table 1.4: These sentences are all logically equivalent to $\sigma_{1.8}$, though this list is obviously not exhaustive.

disjunction
$\vee$

The conjunction has a *logical dual* called the *disjunction*, defined in Table 1.3 using the $\vee$ symbol and exemplified by the following sentence.

"Espresso is delicious, or it nourishes the soul." (1.9)

We all these two connectives *dual* to each other because negating all of the inputs to one of them is equivalent to negating the output of the other. Specifically, the expression $\neg p \vee \neg q$ is equivalent to $\neg(p \wedge q)$ whenever p and q are propositions.

Definition 1.2 (Logical Duality).
logical duality
We say two logical connectives f and g are *logically dual* if negating the inputs of f is always logically equivalent to negating the output of g. Equivalently, we can say f is *logically dual* to g if applying f after $\neg$ always gives the same result as applying $\neg$ after g on any given inputs.

Conjunctions and disjunctions are just one example of a dual connective pair. In fact, every logical connective is dual to some other connective![1] For now, we present this result about $\wedge$ and $\vee$ *without proof;* we will *prove* this statement when we discuss Theorem 1.5 in a short while.

[1] Why might this be? Think about this.

Conditional Statements

We turn our attention now to sentence $\sigma_{1.10}$ below.

"If espresso nourishes the soul, then I will drink it." (1.10)

This is a *conditional* sentence, composed of two subclauses called the *antecedent* and the *consequent*.[1] When we use this sort of linguistic construction, we mean to say that *if* the premise happens, *then* the conclusion must also happen. Said another way: the conclusion must occur *whenever* the premise is satisfied. Notice *we are not asserting anything* about the antecedent or consequent individually! We are only establishing a *relationship* where the consequent occurs *every time* that the premise is satisfied. We call this the *material implication*, denoted by the $\to$ symbol and defined in Table 1.5.

implication
$\to$

$$p_{1.10} := \text{"Espresso nourishes the soul."}$$
$$q_{1.10} := \text{"I will drink espresso."}$$

The antecedent and consequent for $\sigma_{1.10}$ are defined above. With these definitions, we can now write $\sigma_{1.10} \equiv p_{1.10} \to q_{1.10}$ and observe that $\sigma_{1.10}$ simply says: *if* $p_{1.10} \equiv \top$, *then* $q_{1.10} \equiv \top$. Importantly, this is *the only thing* that $\sigma_{1.10}$ is asserting! This sentence *is not saying* that if $p_{1.10} \equiv \bot$, then $q_{1.10} \equiv \bot$. In fact, if the premise is *false*, then $\sigma_{1.10}$ says *nothing* about whether or not $q_{1.10}$ is *true* or *false*.

To make this concrete, suppose I told you the following.

"If you make an A in this class, then I will eat my shoe." (1.11)

If you do happen to make an A in this class, then I'll be forced to physically eat my shoe in order to keep up my end of the bargain; in that case, the sentence was *true*.[2] On the other hand, if you make an A- instead, then I can go home with both shoes and conscience intact; in this case, the sentence was also *true*.[3] However, what if you make the A- but I decide to eat my shoe anyways? Did I lie? No; just because you failed to make an A doesn't mean I *can't* eat my shoe! All I said was that I definitely would if you made an A.[4] That sentence is only a lie when you *do* make an A in the class, but I refuse to eat my shoe, since I'm then breaking my promise.[5]

Table 1.5: Truth table for conditionals.

p	q	$p \to q$	$p \leftrightarrow q$
$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\top$

[1] Synonyms for *antecedent* & *consequent*.

protasis	apodosis
sufficient	necessary
premise	inference
assumption	conclusion
supposition	deduction
implicant	implicand
hypothesis	thesis

[2] $\top \to \top \equiv \top$

[3] $\bot \to \bot \equiv \top$

[4] $\bot \to \top \equiv \top$

[5] $\top \to \bot \equiv \bot$

"I will drink espresso *if* it nourishes the soul."
"Espresso nourishes the soul *only if* I drink it."
"It is *sufficient* that espresso nourish the soul for me to drink it."
"It is *necessary* that I drink espresso for it to nourish the soul."
"I will drink espresso *unless* it doesn't nourish the soul."

Table 1.6: These sentences are *all* logically equivalent to $\sigma_{1.10}$. Pay close attention to *grammar* of each sentence, and make special note of *where* the connectives appear.

biconditional ↔

Finally, the *material equivalence*,[1] also called the *biconditional* and written $p \leftrightarrow q$, is *true* exactly when p and q have the same truth value and is *false* otherwise. With these connectives all defined, we are now ready to formally introduce the *recursive definition* of a proposition.

[1] This is often written *"if and only if"* in English, abbreviated *iff* .

A Formal Proposition

Definition 1.3 (Proposition).

proposition

We say that λ is a *proposition iff* λ satisfies the following recurrence.

1. $\lambda = \top$ or $\lambda = \bot$.
2. $\lambda = \neg(\varphi)$, where φ is a proposition.
3. $\lambda = (\varphi) \wedge (\psi)$ where φ and ψ are propositions.
4. $\lambda = (\varphi) \vee (\psi)$, where φ and ψ are propositions.
5. $\lambda = (\varphi) \rightarrow (\psi)$ where φ and ψ are propositions.
6. $\lambda = (\varphi) \leftrightarrow (\psi)$, where φ and ψ are propositions.

Notice the use of *equality* $=$ rather than *equivalence* $\equiv$ throughout this definition. In each statement here, we are saying that the statement λ is *equal* to the expression on the right-hand side of the $=$ symbol, meaning *they are the same sentence written in the same way.* This gives a *syntactic* definition of what a proposition is.

The use of parentheses in this definition is to avoid issues with order of operations; in situations where the meaning is clear, we can *carefully* drop parentheses.

This definition works by first establishing as our *basis* that $\top$ and $\bot$ are propositions in (1). We then, in (2) through (6), specify larger propositions *recursively* by composing together smaller, already-existing propositions using logical connectives. This then lets us verify that statements like $((\neg\top) \wedge (\bot \wedge \top)) \rightarrow \top$ are indeed propositions.

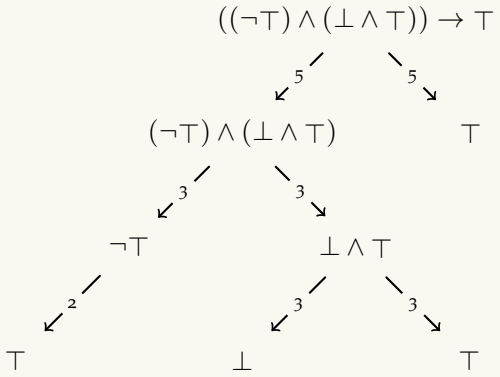

Figure 1.2: In this example, we have dropped some unambiguous parentheses for clarity. Notice, however, that some parentheses *cannot* be dropped: for example, those around the premise of the $\rightarrow$ conditional, and those separating the arguments of the two $\wedge$ conjunctions. If those parentheses had been placed like $((\neg\top) \wedge \bot) \wedge \top$ instead, we would have parsed $\curlywedge$ instead of $\curlyvee$ as in the figure.

Alternatively, think of this as *inductive bootstrapping*.[2] Beginning with $\top$ and $\bot$ from (1) as our initial instances of propositions, we then build larger propositions like $\neg\bot$ and $\top \wedge \bot$, which fall into (2) and (3) respectively. We can then take those expressions, conjunct them again using (2), and place an implication between that result and $\top$ using (5) to arrive at our final expression $((\neg\top) \wedge (\bot)) \rightarrow \top$. By taking basis expressions and connecting them together according to the rules laid out in the definition, we *computed* a way of building the final expression in a way that satisfies the definition, verifying that it is a proposition.

[2] *"Pulling itself up by the bootstraps."*

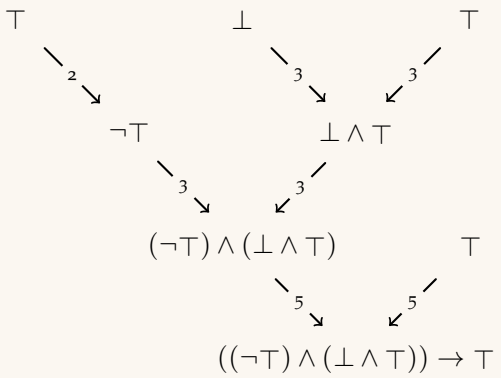

Figure 1.3: The inductive way of building up the expression, as contrasted with the recursive way of tearing down the expression in the previous figure.

***Definition* 1.4 (Propositional Formula).**

propositional formula

A *propositional formula* is an expression that evaluates as a proposition when all of its *variables* are themselves replaced by propositions.

Logical Equivalence

The astute reader may have noticed that some expressions are logically equivalent to each other even if they look different when written out.

p	q	$\neg(p \wedge q)$	$\neg p \vee \neg q$	$p \to q$	$\neg q \to \neg p$
$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$

Table 1.7: A truth table verifying two equivalences. First, that $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are equivalent as predicted by DeMorgan. Second, that $p \to q$ is equivalent to its *contrapositive* $\neg q \to \neg p$.

For example, it's clear that $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$, as the name *"if and only if"* would suggest. We saw another example of an equivalence when we examined the duality of $\wedge$ and $\vee$, illustrated in Table 1.7. We can see that statements like these are logically equivalent because the output truth values are always the same whenever we assign the same input truth values to the variables in these expressions; their output columns are identical in their joint truth table. *Equivalent propositions are essentially the same when viewed through the lens of truth values.*[1]

Following this idea means having to construct a joint truth table whenever we want to check whether or not two formulæ are equivalent. Although it would be a straightforward to automate, doing all of our work by hand would be *extremely* tedious. If we are given two propositions $\varphi(p_1, p_2, \ldots p_n)$ and $\psi(p_1, p_2, \ldots p_n)$ consisting of the same variables, then answering $\varphi(p_1, p_2, \ldots p_n) \stackrel{?}{\equiv} \psi(p_1, p_2, \ldots p_n)$ requires computing truth values for φ and ψ with *all possible combinations* of truth assignments to $p_1, p_2, \ldots p_n$ and checking that they match.

[1] The idea of blurring the lines between objects that are *essentially the same* according to some salient characteristics is a fundamental idea in mathematics that shows up basically everywhere. This is, fundamentally, *why* abstractions are useful and interesting: we abstract in order to draw equivalences between things we previously thought of as distinct.

Now, p_1 can either be $\top$ or $\bot$. For each of these truth values, we then have check both truth values p_2 can take. Then, for each of those, we need to check the two truth values for p_3, and so on until we reach p_n. Each particular assignment of truth values to all of the propositional variables corresponds to one row in our truth table.

If $n = 1$, so our propositions each involve one variable, this means we only need two rows in our truth table to exhaust the entire search space: one row if the variable is $\top$, and one row if it's $\bot$. However, with each new variable we introduce, we *double* the size of our search space because this new variable comes with *two new possible truth values* that we need to check *for each* of the rows we've already computed. We summarize this phenomenon with the following *recurrence relation.*[1]

$$\texttt{rows}(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n = 1 \\ 2 \cdot \texttt{rows}(n-1) & \text{if } n \geqslant 2 \end{cases} \tag{1.12}$$

[1] The degenerate case of $n = 0$, when neither expression has any propositional variables, would just require one row in our truth table since each proposition only has one, unchanging truth value.

This shows us that answering the equivalence question for propositional formulæ of n variables involves computing a truth table with 2^n rows. Obviously, *this doesn't scale;* it quickly becomes infeasible to even *allocate enough space* for our output columns, much less actually compute and check these outputs.

proof

The thinking man's alternative is to instead *prove* that the two expressions are equivalent, constructing a formal, logical argument that derives $\varphi(p_1, p_2, \ldots p_n) \equiv \psi(p_1, p_2, \ldots p_n)$ from assumptions—called *axioms*—using *rules of inference.* We return to this idea in section 1.3.

axiom

Logical Nonequivalence

Showing that two propositional expressions are *not* equivalent is computationally easier than showing that they *are*. Checking that two propositional formulæ are equivalent involves either writing proof or computing *every row* of an exponentially sized truth table. However, checking that two formulæ are *not* equivalent requires *just one example* of a truth assignment on which the propositions disagree. Instead of an entire truth table, all we need is a *single row*.

p	q	$\neg(p \wedge q)$	$\neg p \wedge \neg q$
$\top$	$\top$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\bot$
$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\top$

Table 1.8: A truth table showing negations do not distribute over conjunctions.

For example, to show that $p \to q \not\equiv q \to p$, all we have to do is let $p := \top$ and $q := \bot$. We can then observe that $p \to q \equiv \top \to \bot \equiv \bot$. Meanwhile, $q \to p \equiv \bot \to \top \equiv \top$. Thus, we conclude $p \to q \not\equiv q \to p$.

Definition* 1.5 *(Logical Equivalence & Nonequivalence).
Let φ and ψ be propositional formulæ both consisting of the *same* variables $p_1, \dots p_n$. We say that φ is *equivalent* to ψ if *every* assignment of truth values to the variables of φ and ψ produces the same truth value. In this case, we write $\varphi \equiv \psi$.

logical equivalence

$\equiv$

We say that φ is *not equivalent* to ψ if *there is* an assignment of truth values to the formulæ's variables that makes the truth values of φ and ψ different. In this case, we write $\varphi \not\equiv \psi$.

logical non equivalence

$\not\equiv$

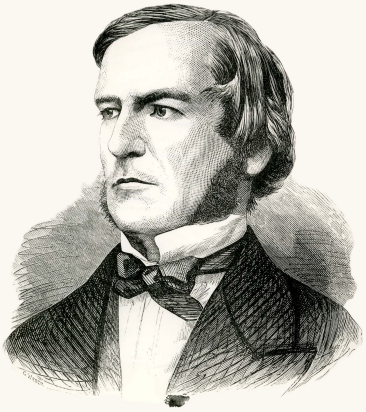

Figure 1.4: George Boole, a largely self-taught mathematician, logician, and philosopher, first described the eponymous *Boolean algebra* in his 1854 monograph *The Laws of Thought*.

1.3 *The Propositional Logic*

Axioms & Proofs

The axioms of propositional logic encode the *foundational assumptions* we are making about the nature of truth-value-based reasoning. We take these truths to be self-evident *without justification.*

IDENTITY	$\top \wedge p \equiv p$	$\bot \vee p \equiv p$
COMPLEMENT	$\neg p \wedge p \equiv \bot$	$\neg p \vee p \equiv \top$
COMMUTATIVITY	$p \wedge q \equiv q \wedge p$	$p \vee q \equiv q \vee p$
ASSOCIATIVITY	$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$	$p \vee (q \vee r) \equiv (p \vee q) \vee r$
DISTRIBUTIVITY	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
CONDITIONAL DISINTEGRATION		$p \to q \equiv \neg p \vee q$
BICONDITIONAL DISINTEGRATION	$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$	

Table 1.9: The axioms of classical logic. The first five specify a *Boolean algebra*; notice that each of these first five axioms has a conjunctive fragment (left) and a *dual* disjunctive fragment (right).

Each of the statements in this table is a logical equivalence establishing that the two expressions are *interchangeable in all contexts.* We could verify each of these by constructing the appropriate truth table; however, the attitude we will take is that each statement in the table simply *is true a priori.* They will be the basis upon which we build our proofs.

The *complement* axiom in the second row of Table 1.9 shows us two important facts about the negation of any proposition. If we take a proposition p and conjunct it with its negation $\neg p$, that axiom tells us that we get $\bot$; dually, disjuncting p with its negation gives us $\top$. Is this behavior *characteristic* of $\neg p$? The following theorem tells us *yes,* that any proposition that *behaves like* the negation of p must be *indistinguishable* from $\neg p$ through the lens of truth values! With that said, let's try to prove our first *theorem*.[1]

theorem

[1] A *theorem* is a provable proposition.

Theorem 1.1 (Uniqueness of Complements).
For any propositions p and q, if $p \wedge q \equiv \bot$ and $p \vee q \equiv \top$, then $\neg p \equiv q$.

Proof. Let p and q be arbitrary propositions.[1] Assume $p \wedge q \equiv \bot$ and $p \vee q \equiv \top$.[2] We will prove $\neg p \equiv q$ by showing that $\neg p$ and q are both equivalent to the same expression. First, observe the following.

$\neg p \equiv \neg p \wedge \top$	by *identity*
$\equiv \neg p \wedge (p \vee q)$	because $p \vee q \equiv \top$ by assumption
$\equiv (\neg p \wedge p) \vee (\neg p \wedge q)$	by *distributivity*
$\equiv \bot \vee (\neg p \wedge q)$	by *complement*
$\equiv \neg p \wedge q$	by *identity*

As a result, $\neg p \equiv \neg p \wedge q$. Similarly, we can now observe the following.

$q \equiv q \wedge \top$	by *identity*
$\equiv q \wedge (p \vee \neg p)$	by *complement*
$\equiv (q \wedge p) \vee (q \wedge \neg p)$	by *distributivity*
$\equiv (p \wedge q) \vee (\neg p \wedge q)$	by *commutativity*
$\equiv \bot \vee (\neg p \wedge q)$	because $p \wedge q \equiv \bot$ by assumption
$\equiv \neg p \wedge q$	by *identity*

This gives us $q \equiv \neg p \wedge q$. Therefore, we conclude $\neg p \equiv \neg p \wedge q \equiv q$.

Q.E.D.

[1] Since we need to prove this statement *for any two propositions* p and q, we introduce two *arbitrary* propositions at the beginning of our proof.

[2] These assumptions are warranted because they are the *premise* of the *conditional* statement we are proving.

Q.E.D. stands for *Quod Erat Demonstrandum*, which is Latin for "what was to be shown has been demonstrated," after the Greek Ὅπερ ἔδει δεῖξαι. This is called a *tombstone*, and it is a traditional way of denoting the end of a proof. Modern authors might use □ or ■ instead.

Notice how *every* statement in the proof above is written with *purpose*, and much of the proof is inspired by *the form of the theorem* we are trying to prove. Let's analyze what just happened. Before we begin writing the proof, we first read the theorem focussing on two things: the *form* of the statement, and *what* the statement says.

First and foremost, this theorem says something about *any propositions*. We have two options for proving something is true about every single proposition: we can check all of them individually, or we can show that the thing we are trying to prove is an *inherent quality of being a proposition*. The former approach is clearly unworkable whenever we have infinitely many—or even just a large amount of—things to check, as we do here. Instead, we will take the later approach: by taking an *arbitrary* proposition and *making no assumptions, imposing no constraints*, then any argument we make about this particular proposition will also apply to any other proposition we encounter.[3] The first sentence of the proof introduces these two arbitrary propositions.

[3] As an example, suppose we wanted to prove that the square of any positive number is also positive. We obviously can't check all of the positive numbers one-by-one. Instead, we can take an *arbitrary* number x such that $x > 0$, and then argue that $x^2 > 0$. If we do this successfully, then we can take *any* particular number, such as 5, substitute it for x in our argument, and obtain a proof that $5^2 > 0$. However, if we couldn't have written our *original argument* in terms of 5; this would have meant imposing the *additional constraint* that $x = 5$, preventing our argument from generalizing to *all* positive numbers.

Now that we know we are proving something *universal* about propositions, we keep reading the theorem and see that it's a statement of

the form *"if ___, then ___."* This is a *conditional* statement, and the most straight-forward way to show a conditional statement is *true* is to *demonstrate the conclusion is fulfilled whenever the premise is true.* Thus, we can *assume* the premise of the conditional is *true,* and our task them is to derive the conclusion. The second sentence of our proof assumes the premise, which happens to be a conjunction of two statements.

Up to this point, everything we've done has been determined solely by the *form* of the theorem we are trying to prove. Now, our task is to take what we have and show the conclusion.[1] What follows next is a sequence of logical statements, each of which is *justified*,[2] which ends at the conclusion we wanted. *How* you decide to craft this sequence of statements—what statements to make in what order, what proof techniques to use, what intuition inspired your approach—is entirely dependent on *your style* as long as all of the logic is clear, all of the logical rules are followed, and all of the justification is correct.

[1] If our conclusion were a longer, compound statement, we would continue breaking the problem down *recursively* until we were left with something *atomic*.

[2] ...either by a *definition*, an *axiom*, an *assumption* we've made, or a prior *theorem* we've proven...

Proof-writing is an art form in much the same way building a musical instrument is. When a luthier makes a guitar, the process is guided by the particular luthier's traditions, experiences, style, and tastes; so long as the final product is truly a guitar that sounds and plays like a guitar should, the luthier has complete liberty. While two master luthiers might take radically different approaches that lead to guitars with unique aesthetic qualities, they will nonetheless produce two functioning guitars and preference of one over the other will be a matter of judgement and taste. This is much the same when it comes to writing proofs; the analogue to programming should be clear.

Figure 1.5: Examples of three distinct bracing styles for the classical guitar.

Since we proved Theorem 1.1, we can now use this result in the future when proving more complicated statements. For example, it should be easy to see intuitively that $\top \equiv \neg\bot$ and $\bot \equiv \neg\top$, based on the way we use the words *true* and *false* in natural language and how $\top$ and $\bot$ are meant to correspond to those truth values. We can now prove this simple statement by relying on Theorem 1.1. A statement that follows *easily* from a theorem is usually called a *corollary* to that theorem.

Corollary 1.1.
$\top \equiv \neg\bot$ and $\bot \equiv \neg\top$.

Proof. Observe that $\bot \wedge \top \equiv \bot$ by the *identity* axiom. Similarly, we have that $\bot \vee \top \equiv \top \vee \bot \equiv \top$ by *commutativity* and the *identity* axiom again. So, we can apply Theorem 1.1[3] and conclude $\top \equiv \neg\bot$. Similarly, we can observe that $\top \wedge \bot \equiv \bot \wedge \top \equiv \bot$ by *commutativity* and *identity*, and $\top \vee \bot \equiv \bot$ by the *identity* axiom. Thus, $\bot \equiv \neg\top$ by Theorem 1.1.

Q.E.D.

[3] We can invoke the theorem here because we have just *proven* the premises of the theorem are *true* for the particular propositions we are looking at (in this case, $p \coloneqq \bot$ and $q \coloneqq \top$). That means, having satisfied the premises, we get to assert the conclusion, justified by that theorem.

A proof gives us more than just a formal verification of a statement. It tells us that the statement is a *necessary consequence* of the axioms we assumed in setting up our logical system, and every instance of a proof gives us insight into *why* that's the case. These past two proofs show us that we didn't have to explicitly *define* or *assume* $\top$ to be the opposite of $\bot$ because this is a face satisfied by *any* instance of a Boolean algebra.

Let's prove another simple, but useful, theorem.

***Corollary* 1.2.**
For any propositions p and q, if $p \equiv q$, then $\neg p \equiv \neg q$.

Proof. Let p and q be propositions such that $p \equiv q$ and observe.

$$\begin{aligned} q \wedge \neg p &\equiv p \wedge \neg p && \text{because we assumed } p \equiv q \\ &\equiv \bot && \text{by } \textit{complement} \end{aligned}$$

We can do a very similar thing in the disjunctive case.

$$\begin{aligned} q \vee \neg p &\equiv p \vee \neg p && \text{because we assumed } p \equiv q \\ &\equiv \top && \text{by } \textit{complement} \end{aligned}$$

Therefore, applying Theorem 1.1, we conclude that $\neg p \equiv \neg q$.

Q.E.D.

***Corollary* 1.3.**
For any propositions p, q, r, s such that $p \equiv q$ and $r \equiv s$, we have

$$\begin{aligned} p \wedge r &\equiv q \wedge s \\ p \vee r &\equiv q \vee s \\ p \to r &\equiv q \to s \\ p \leftrightarrow r &\equiv q \leftrightarrow s \end{aligned}$$

We include these facts above just for completeness, so that some of the basic properties of $\equiv$ are codified somewhere; their proofs are not particularly interesting. Our goal for the rest of this section will be to prove *De Morgan's laws*. For this, we first need a few more tools.

Theorem* 1.2 *(Double Negation).
For any proposition p, we have that $p \equiv \neg\neg p$.

Proof. Let p be a proposition. We will show that $p \equiv \neg\neg p$ by showing that p acts like the negation of $\neg p$. Observe that $\neg p \wedge p \equiv p \wedge \neg p \equiv \bot$ by *communtativity* and the *complement* axiom. We can similarly see $\neg p \vee p \equiv p \vee \neg p \equiv \top$ by *commutativity* and *complement*. Therefore, we can conclude that $p \equiv \neg(\neg p)$ by Theorem 1.1.

Q.E.D.

Theorem 1.3 (Idempotency).
For any proposition p, we have $p \wedge p \equiv p$ and $p \vee p \equiv p$.

Proof. Let p be a proposition. For the conjunctive statement, observe.

$$\begin{aligned} p \wedge p &\equiv (p \wedge p) \vee \bot && \text{by } \textit{identity} \\ &\equiv (p \wedge p) \vee (p \wedge \neg p) && \text{by } \textit{complement} \\ &\equiv p \wedge (p \vee \neg p) && \text{by } \textit{distributivity} \\ &\equiv p \wedge \top && \text{by } \textit{complement} \\ &\equiv p && \text{by } \textit{identity} \end{aligned}$$

An analogous chain of reasoning takes us through the disjunctive case.

$$\begin{aligned} p \vee p &\equiv (p \vee p) \wedge \top && \text{by } \textit{identity} \\ &\equiv (p \vee p) \wedge (p \vee \neg p) && \text{by } \textit{complement} \\ &\equiv p \vee (p \wedge \neg p) && \text{by } \textit{distributivity} \\ &\equiv p \vee \bot && \text{by } \textit{complement} \\ &\equiv p && \text{by } \textit{identity} \end{aligned}$$

Therefore, we have $p \wedge p \equiv p$ and $p \vee p \equiv p$ as desired.

Q.E.D.

Theorem 1.4 (Domination).
For any proposition p, we have $p \vee \top \equiv \top$ and $p \wedge \bot \equiv \bot$.

Proof. Let p be a proposition and observe.

$$\begin{aligned} p \vee \top &\equiv p \vee (p \vee \neg p) && \text{by } \textit{complement} \\ &\equiv (p \vee p) \vee \neg p && \text{by } \textit{associativity} \\ &\equiv p \vee \neg p && \text{by } \textit{idempotency} \\ &\equiv \top && \text{by } \textit{complement} \end{aligned}$$

The dual case works out similarly.

$$\begin{aligned} p \wedge \bot &\equiv p \wedge (p \wedge \neg p) && \text{by } \textit{complement} \\ &\equiv (p \wedge p) \wedge \neg p && \text{by } \textit{complement} \\ &\equiv p \wedge \neg p && \text{by } \textit{idempotency} \\ &\equiv \bot && \text{by } \textit{complement} \end{aligned}$$

We therefore conclude $p \vee \top \equiv \top$ and $p \wedge \bot \equiv \bot$.

Q.E.D.

We now finally have all the tools we need to prove *De Morgan's laws*.

Theorem 1.5 (De Morgan's Laws).
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ and $\neg(p \vee q) \equiv \neg p \wedge \neg q$ for all propositions p, q,

Proof. Let p and q be propositions. We will prove the first half of this theorem by showing $\neg p \vee \neg q$ acts like the negation of $p \wedge q$ and then applying *uniqueness of negations*. We leave proving $\neg(p \vee q) \equiv \neg p \wedge \neg q$ as an exercise to the reader. First, the conjunctive branch.

$$
\begin{aligned}
(p \wedge q) \wedge (\neg p \vee \neg q) &\equiv p \wedge (q \wedge (\neg p \vee \neg q)) && \text{by } \textit{associativity} \\
&\equiv p \wedge ((q \wedge \neg p) \vee (q \wedge \neg q)) && \text{by } \textit{distributivity} \\
&\equiv p \wedge ((q \wedge \neg p) \vee \bot) && \text{by } \textit{complement} \\
&\equiv p \wedge (q \wedge \neg p) && \text{by } \textit{identity} \\
&\equiv p \wedge (\neg p \wedge q) && \text{by } \textit{commutativity} \\
&\equiv (p \wedge \neg p) \wedge q && \text{by } \textit{associativity} \\
&\equiv \bot \wedge q && \text{by } \textit{complement} \\
&\equiv \bot && \text{by } \textit{domination}
\end{aligned}
$$

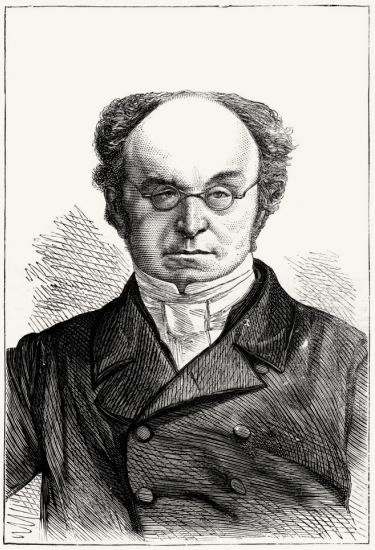

Figure 1.6: Augustus De Morgan, after whom these laws are named, is also notable for his work on logical quantification and mathematical induction.

The disjunctive case works out analogously.

$$
\begin{aligned}
(p \wedge q) \vee (\neg p \vee \neg q) &\equiv ((p \wedge q) \vee \neg p) \vee \neg q && \text{by } \textit{associativity} \\
&\equiv (\neg p \vee (p \wedge q)) \vee \neg q && \text{by } \textit{commutativity} \\
&\equiv ((\neg p \vee p) \wedge (\neg p \vee q)) \vee \neg q && \text{by } \textit{distributivity} \\
&\equiv ((p \vee \neg p) \wedge (\neg p \vee q)) \vee \neg q && \text{by } \textit{commutativity} \\
&\equiv (\top \wedge (\neg p \vee q)) \vee \neg q && \text{by } \textit{complement} \\
&\equiv (\neg p \vee q) \vee \neg q && \text{by } \textit{identity} \\
&\equiv \neg p \vee (q \vee \neg q) && \text{by } \textit{associativity} \\
&\equiv \neg p \vee \top && \text{by } \textit{complement} \\
&\equiv \top && \text{by } \textit{domination}
\end{aligned}
$$

Therefore, $\neg(p \wedge q) \equiv \neg p \vee \neg q$ by Theorem 1.1.

Q.E.D.

Rules of Inference

So far, we've developed a modestly-powerful formal language—capable of expressing some basic logical ideas—founded on *axioms*. This gives us a formal syntactic framework for expressing logical ideas, along with a basic semantics that relates these formal symbols to our natural language. The axioms in Table 1.9 are all *equivalences*—substitution rules between propositions that preserve truth values—and we've now seen several examples of their use in proving some basic theorems.

THE DEDUCTION RULE	$(p \vdash q) \vdash (p \to q)$	If, by assuming p, we can prove q, then we can write $p \to q$.
MODUS PONENS	$p, (p \to q) \vdash q$	If we have $p \to q$ and we know p, then we can deduce q.
MODUS TOLLENS	$\neg q, (p \to q) \vdash \neg p$	If we have $p \to q$ but we also know $\neg q$, then we can infer $\neg p$.
REDUCTIO AD ABSURDUM	$(\neg p \vdash \bot) \vdash p$	If we can derive a contradiction from $\neg p$, then $\neg p$ is absurd, and we conclude p instead.

Table 1.10: The rules of inference.

Yet, you may have noticed that *some* of our reasoning in those proofs *was not based on equivalences*. This is most apparent in the proof of Theorem 1.1, our very first theorem. We began that proof by introducing two arbitrary propositions and then *immediately assuming* that their conjunction was $\bot$ and their disjunction as $\top$. Making those assumptions was not justified on any of the equivalence axioms we'd introduced, so why were we allowed to say that in our proof? By a similar token, in the proof of Corollary 1.1, we apply Theorem 1.1 by saying that, since we'd satisfied the premises of that theorem, we were allowed to write down the conclusion of that theorem. Why were we allowed to say that? In short: *because it makes sense!* The problem, of course, is that nothing yet in our system *formally* gives us the right or power to do these things, even though they make logical sense.

This then calls for the introduction of more axioms—ones that will allow us to construct these kinds of *one-way*, *inferential* arguments alongside our equivalence-based reasoning. We call these the *rules of inference*.

The rules in Table 1.10 each take the form $\Gamma \vdash \varphi$,[1] where Γ represents a set of assumptions and φ is the conclusion that follows from them. The $\vdash$ symbol, sometimes called a turnstile, signifies that we can *prove* φ by assuming the statements in Γ and using the equivalence axioms, the rules of inference, and any theorems we've already proven. If there is nothing written to the left of the $\vdash$ symbol, this simply means that the conclusion φ can be derived *without* any additional assumptions.

prove

[1] *"Γ proves φ"* or *"φ follows from Γ."*

The most important of the rules of inference is *modus ponens*, enabling us to *follow through* on chains of conditional reasoning.[2] *Modus ponens* is, in a sense, the essence of classical rhetoric. Without it, the conclusion of a conditional statement's conclusion would not be meaningfully *conditioned* on its premise. There would be no point in establishing hypothetical arguments because the conditional chains of reasoning would never actually have any point to work towards. This rule has a sister—*modus tollens*—which conversely allows *breaking down* arguments counterfactually, denying antecedents with false consequents.[3]

[2] *Modus ponens* is short for the Latin phrase *modus ponendo ponens*, literally *"the method of putting by placing."*

[3] *Modus tollens* is short for the Latin phrase *modus tollendo tollens*, literally *"the method of removing by taking away."*

The next rule, named *reductio ad absurdum*,[1] gives us the ability to construct proofs by contradiction. Suppose we are interested in proving some proposition p. One way to reason about the validity of p is to think about what would happen if p were not the case. Hypothetically, assuming $\neg p$, if we were able to derive both q and also $\neg q$, then we would have derived a falsity ($q \wedge \neg q \equiv \bot$). If we were starting from *true* premises, this would be impossible since all of our axioms and rules of inference are *truth-preserving*. Clearly, this must mean that our assumption $\neg p$ was *not true*, leaving p as the only logical conclusion. This form of argumentation is like *"is like arguing with a hammer,"* according to a dear professor of mine from undergrad. It is incredibly powerful and has been in use since at least the year 400 BC.[2]

[1] *Reductio ad absurdum* is a Latin phrase meaning to *"reduce to absurdity."* This has also been called *argumentum ad absurdum*.

[2] In Plato's dialogues, Socrates frequently engages in this sort of reasoning by showing his opponents' seemingly-sensible statements can be systematically dismantled to absurdity.

Finally, the *deduction rule* is a technical rule of inference that ties together the meta-symbol $\vdash$ with the logical $\rightarrow$ symbol. It enshrines the parallel between a deductive *"q follows from p"* statement and a formal *"if p then q"* statement. If this distinction is confusing, just keep in mind that we are constructing a formal language to express mathematical ideas with; the *propositions* we express are written in our language, but we write our *proofs* of these propositions in our natural language, and our natural language is what we use to write down the rules and axioms that our language must obey. The *deduction rule* tells us that the result of our *proofs* can be converted into statements *within the formal language.*

Although this is a rather small collection of rules, it is capable of representing any kind of expressible propositional rhetoric. Despite that, it's not a *minimal* set of rules for the zeroth-order logic. In fact, it's possible to have an even smaller set of rules without sacrificing the rhetorical strength of our language. *Modus tollens*, for instance, could actually be shown to follow from the other rules of inference as a *theorem*, reducing our total number of assumptions. Let's prove it now.

Theorem 1.6 (Modus Tollens).
We have $\neg q, (p \rightarrow q) \vdash \neg p$ for any propositions p and q.

Proof. Let p and q be arbitrary propositions, and suppose $\neg q$ and also $p \rightarrow q$. We know that $p \rightarrow q \equiv \neg q \rightarrow \neg p$,[3] so we have $\neg q \rightarrow \neg p$. Then, by *modus ponens*, we can conclude $\neg p$.

Q.E.D.

[3] This result—that a conditional statement is equivalent to its *contrapositive*—is an exercise in your problem set.

We don't provide a *minimal* set of inference rules in order to keep this topic somewhat manageable, but the interested reader might be excited to learn that *all* of propositional logic can be encoded using *just two connectives* ($\neg$ and $\rightarrow$) and *just three axioms* along with *modus ponens*.

In addition to these rules, there are several classical *syllogisms*[4]—studied

[4] A *syllogism* is a traditional way to refer to a deductive argument that draws a conclusion from a small set of premises.

since times of ancient Greece—that have formed the historical bedrock for argumentation in philosophy. We can prove some of these now.

Theorem 1.7 (Hypothetical Syllogism).
We have $(p \to q), (q \to r) \vdash p \to r$ for any propositions p, q, and r.

This theorem allows us to construct and follow chains of reasoning, a fundamental aspect of any nontrivial argument.

Proof. Let p, q, and r be arbitrary propositions, and suppose $p \to q$ and $q \to r$. We will first show that $p \vdash r$. Assume p. Since $p \to q$, we have q by *modus ponens*. Further, since we have $q \to r$, we get r by *modus ponens*. Thus, $p \vdash r$. Therefore, by applying the *deduction theorem*, we can conclude $p \to r$.

Q.E.D.

Theorem 1.8 (Conditional Elimination).
We have $(p \to q) \vdash (p \vdash q)$ for any propositions p and q.

This theorem, when combined with the *deduction theorem*, establishes the syntactic equivalence between the semantically-distinct $\to$ and $\vdash$ symbols.

Proof. Let p and q be arbitrary propositions, and suppose $p \to q$. We will now show that $p \vdash q$. Assume p. Then, since we have $p \to q$, we can derive q by *modus ponens*. Thus, $p \vdash q$.

Q.E.D.

Theorem 1.9 (Conjunction Introduction).
We have $p, q \vdash p \wedge q$ for any propositions p and q.

This is known as *adjunction*.

Proof. Let p and q be arbitrary propositions. Assume p, and also separately assume q. Towards a contradiction,[1] suppose $\neg(p \wedge q)$. We can plainly see

$$\begin{aligned} \neg(p \wedge q) &\equiv \neg p \vee \neg q && \text{by } \textit{De Morgan's laws} \\ &\equiv p \to \neg q && \text{by } \textit{conditional disintegration}. \end{aligned}$$

So, we have $p \to \neg q$, from which we can derive $\neg q$ by *modus ponens*. This shows us that $\neg(p \wedge q) \to \neg q$ by the *deduction theorem*. However, since we also had q by assumption, we can derive $\neg(p \wedge q) \to q$ using the *deduction theorem* again. ↯ [2]

Therefore, we can conclude $p \wedge q$ by *reductio ad absurdum*.

Q.E.D.

[1] When beginning a *proof by contradiction*, it is good form to explicitly alert the reader to this fact when you make your problematic assumption.

[2] The symbol ↯ is useful for highlighting to the reader where, in a proof by contradiction, the actual *contradiction* is.

Theorem 1.10 (Conjunction Elimination).
We have $(p \wedge q) \vdash p$ for any propositions p and q.

This is known as *simplification*.

Theorem 1.11 (Disjunctive Syllogism).
We have $(p \vee q), \neg p \vdash q$ for any propositions p and q.

Theorem 1.12 (Disjunction Introduction).
We have $p \vdash (p \vee q)$ for any propositions p and q.

This is known as *addition*.

Theorem 1.13 (Disjunction Elimination).
We have $(p \to r), (q \to r), (p \vee q) \vdash r$ for any propositions p, q, r.

This is known as *proof by cases*.

Theorem 1.14 (Biconditional Elimination).
We have $(p \leftrightarrow q), p \vdash q$ for any propositions p, q.

Theorem 1.15 (Constructive Dilemma).
We have $(p \to s), (q \to t), (p \vee q) \vdash (s \vee t)$ for any p, q, s, t.

Theorem 1.16 (Ex Falso Quodlibet).
We have $p, \neg p \vdash q$ for any propositions p, q.

This is known as the *principle of explosion*.

Index