

DANIEL GONZALEZ CEDRE

DISCRETE MATHEMATICS

UNIVERSITY OF NOTRE DAME

February 16, 2024

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

Copyright © 2024 Daniel Gonzalez Cedre
<https://daniel-gonzalez-cedre.github.io>

Contents

<i>Logic</i>	7
o <i>Language</i>	8
o.1 <i>A Brief History of...</i>	8
o.2 <i>Syntax and Semantics</i>	10
o.3 <i>A Recurring Theme</i>	11
1 <i>Zeroth-Order Logic</i>	13
1.1 <i>Truth Values</i>	13
1.2 <i>Logical Connectives</i>	16
<i>Negations</i>	16
<i>Conjunctions and Disjunctions</i>	17
<i>Conditional Statements</i>	18
<i>A Formal Proposition</i>	19
<i>Logical Equivalence</i>	20
<i>Logical Nonequivalence</i>	21
1.3 <i>The Propositional Logic</i>	22
<i>Axioms and Proofs</i>	22
<i>Rules of Inference</i>	28
<i>Hilbert's System</i>	30
<i>Classical Syllogisms</i>	31
2 <i>First-Order Logic</i>	33
2.1 <i>A More Expressive Language</i>	34
<i>Forming Formulae Well</i>	36
2.2 <i>Rules of Inference</i>	36

<i>2.3 The Art of Writing Proofs</i>	38
<i>Quantified Formulæ</i>	38
<i>Conditional Statements</i>	38
<i>Junctions</i>	38
<i>Nonconstructive Proofs</i>	39

<i>Mathematics</i>	40
--------------------	----

<i>3 Foundations</i>	41
<i>3.1 Informal Notions</i>	41
<i>Numbers</i>	42
<i>Functions</i>	42
<i>Sets</i>	43
<i>A Note on Notation</i>	43
<i>3.2 Set Theory</i>	44
<i>Infinity</i>	46
<i>Extensionality</i>	46
<i>Pairing</i>	49
<i>Separation</i>	50

<i>Index</i>	52
--------------	----

Notation

SYNTAX	SEMANTICS
\top	"True."
\perp	"False."
$x := y$	" x is, by definition, y ."
$x = y$	" p equals q ."
$p \equiv q$	" p is equivalent to q ."
$p \Leftrightarrow q$	" p if, and only if, q ."
$p \vdash q$	" p proves q ."
$p \Rightarrow q$	" p implies q ."
\emptyset	"The empty set."
$\{a, b, c\}$	"The set containing a, b , and c ."
$\{x \mid \varphi(x)\}$	"The set of all x such that $\varphi(x)$."
$\{x \in \mathcal{A} \mid \varphi(x)\}$	"The set of all x in \mathcal{A} such that $\varphi(x)$."
$f : \mathcal{A} \rightarrow \mathcal{B}$	" f is a function from \mathcal{A} to \mathcal{B} ."
$f(x)$	" f of x ."
$\mathbb{P}(x)$	"The power set of x "
\mathbb{N}	"enn"
\mathbb{Z}	"zee"
\mathbb{Q}	"queue"
\mathbb{R}	"arr"

Table 1: 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.

COLOR	INTERPRETATION
Yellow	<i>Emphasis</i>
Blue	<i>Pronunciation</i>
Magenta	<i>Definition</i>
Green	<i>External link</i>
Black	<i>Internal link</i>

Table 2: Color legend.

MARK	MEANING
定義	<i>definition</i>
直覺	<i>idea</i>
公理	<i>axiom</i>
引理	<i>lemma</i>
定理	<i>theorem</i>
推論	<i>corollary</i>
演算法	<i>algorithm</i>

Table 3: Notation for organizing topics.

GLYPH	NAME	IPA	GLYPH	NAME	IPA
A α	<i>alpha</i>	[a]	N ν	<i>nu</i>	[n]
B β	<i>beta</i>	[v]	Ξ ξ	<i>xi</i>	[ks]
Γ γ	<i>gamma</i>	[y]	O \circ	<i>omicron</i>	[o]
Δ δ	<i>delta</i>	[ð]	Π π	<i>pi</i>	[p]
E ϵ	<i>epsilon</i>	[e]	P ρ	<i>rho</i>	[r]
Z ζ	<i>zeta</i>	[z]	Σ σ	<i>sigma</i>	[s]
H η	<i>eta</i>	[ɛ:]	T τ	<i>tau</i>	[t]
Θ θ	<i>theta</i>	[θ]	Y υ	<i>upsilon</i>	[y:]
I ι	<i>iota</i>	[i:]	Φ φ	<i>phi</i>	[f]
K κ	<i>kappa</i>	[k]	X χ	<i>chi</i>	[kʰ]
Λ λ	<i>lambda</i>	[l]	Ψ ψ	<i>psi</i>	[ps]
M μ	<i>mu</i>	[m]	Ω ω	<i>omega</i>	[ɔ:]

Table 4: The Greek alphabet.

Logic

O

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 AD² agitated by an existential crisis in mathematics: a crisis caused by our flagrant use of language. Here's a short summary.

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.



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.

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.

axiom

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 (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 of the interior angles. 公理

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.

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 discoveries⁴ 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.

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

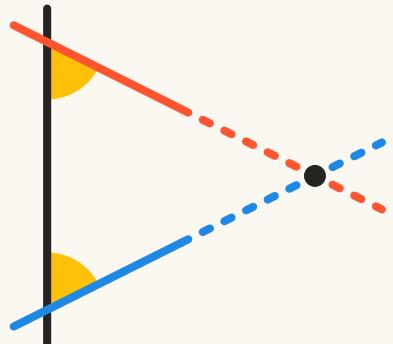


Figure 2: The parallel postulate says that any two lines — and — that make acute interior angles ▶ and ▶ with a third line — must intersect at a point ●.

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

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

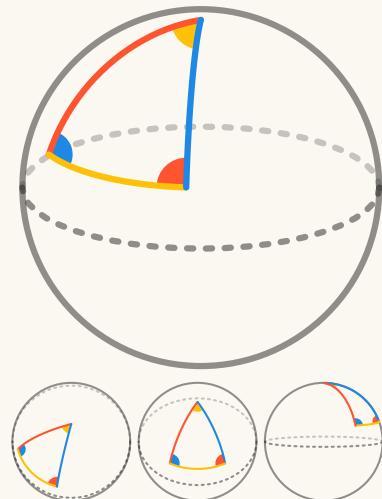


Figure 3: Four views of the same triangle whose angles sum to 270 degrees. Notice how the notions of *straight* and *parallel* differ on the surface of a sphere.

⁴ We will study these later.

Once again, mathematicians' *intuitive* notions of how *infinity* should behave were being contradicted. 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 *logical foundation*.¹

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

Languages encode ideas into sequences of symbols.² 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 symbol³ that stands in place for an object that has not been determined yet.⁴ We can assign a name to a *particular* object with the \coloneqq symbol. We call these the *terms* of an expression.

Definition 0.1 (Sentences).

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.

定義

Mathematical practice principally involves *making and justifying observations about mathematical objects*.⁵ 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.

¹ This ambitious project would eventually fail with the discovery of Kurt Gödel's infamous *incompleteness theorems*.

² For our purposes, we will focus only on written—as opposed to spoken or signed—languages.

³ We typically denote variables using single Latin or Greek letters, though there are no strict universal rules. Some common examples are listed below.

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

⁴ A variable does not *necessarily* refer one particular object, or even any object at all.

“Oft hope is born when all is forlorn.”

“What has it got in its pockets?”

“Keep your forked tongue behind your teeth.”

⁵ 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). We say something is *computable* if it is expressible as a *general recursive process*, is a *term in the λ -calculus*, or could be described by a *Turing machine*. 直覺

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 underlying 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 and 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 $17 + 2 = 19$.

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

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 $x_0 + x_1 + x_2 + \dots + x_{n-2} + x_{n-1}$, we *first* need to compute $x_0 + x_1 + x_2 + \dots + x_{n-2}$ and then add that result to x_{n-1} .

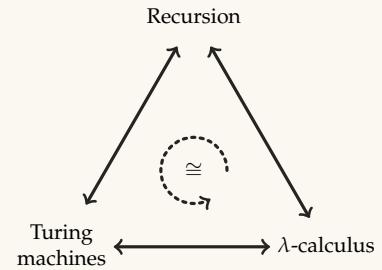


Figure 4: The Church-Turing thesis states that these three concepts—which are all *formally equivalent*—correspond with our *informal* notion of computability. In modern times, many people now take this as a definition for computability.

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

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 $x_0 + x_1 + x_2 + \dots + x_{n-2}$ 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.

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

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 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

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

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*—a.k.a. *base case*—does exactly this by providing explicit answers to the *smallest* versions of the problem.

recurrence
relation

recurrence

basis

This paragraph describes the *recurrence*.This paragraph encounters the *basis*.

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.

1

Zeroth-Order Logic

“The limits of my language means the limits of my world.”

– Ludwig Wittgenstein

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.

“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 \perp for sentences of this kind.

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*

true
 \top

false
 \perp

truth value



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

The symbols \top and \perp are also sometimes called “*top*” and “*bot*” respectively.

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).

We say that two sentences φ and ψ are *equivalent* when they have the same truth value. We denote this by writing $\varphi \equiv \psi$.¹

定義

¹ “ φ is (logically) equivalent to ψ .”

Axiom (Propositional Equivalence is an Equivalence Relation).

We will take the following three properties to be *true* for any sentences φ , ψ , and ζ that are carriers of truth values.

1. $\varphi \equiv \varphi$.
2. If $\varphi \equiv \psi$, then $\psi \equiv \varphi$.
3. If $\varphi \equiv \psi$ and $\psi \equiv \zeta$, then $\varphi \equiv \zeta$.

reflexivity

symmetry

transitivity

This establishes \equiv is an example of an *equivalence relation*.²

公理

With this new definition, we can formalize our observations from the preceding paragraph as $\sigma_{1.1} \equiv \top$ and $\sigma_{1.2} \equiv \perp$ as well as $\sigma_{1.1} \not\equiv \sigma_{1.2}$. Notice that each of these three expressions is a complete sentence describing properties² 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 $\sigma_{1.3}$.

“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 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 $\sigma_{1.5}$.

$$\text{"This sentence is } \textit{false}.\text{"} \quad (1.5)$$

Expressed a little more *formally*, this is the sentence—named $\sigma_{1.5}$ —that says $\sigma_{1.5} \equiv \perp$. 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 $\sigma_{1.5} \equiv \perp$. This *contradicts* our initial assumption! ↴ Therefore, $\sigma_{1.5}$ is *not true*¹

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

paradox

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 \perp are opposed to each other, so that the same sentence can't meaningfully be both \top and \perp 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.

Axiom (Principle of Bivalence).

Sentences expressing truth values are either *true* or *false* but not both.
公理

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

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.

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	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	⊥	T	T	T	T
T	⊥	⊥	⊥	T	⊥	⊥
⊥	T	T	⊥	T	T	⊥
⊥	⊥	T	⊥	⊥	T	T

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.

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.

logical
connective

Negations

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

$$\text{"Espresso is not delicious."} \quad (1.6)$$

Immediately, the moral observer will realize the offensive absurdity of this sentence, compelled by the force of conscience to declare $\sigma_{1.6} \equiv \perp$! 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 $\sigma_{1.7}$.

$$\text{"Espresso is delicious."} \quad (1.7)$$

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 $\sigma_{1.6} \equiv \neg\sigma_{1.7}$.

We can now think of \neg formally as a *unary function* that operates on truth values.¹ This function works by mapping \top to \perp and by

negation

→

Table 1.2: Truth table for negations.

p	$\neg p$
T	⊥
⊥	T

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

mapping $\neg\perp$ 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 column represents the inputs¹ to \neg and the right-most column shows the truth values of the resulting output expression.²

Conjunctions and 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*.

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.”
-

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.

Definition 1.2 (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 gives the same result as applying \neg after g on *all possible* inputs. 定義

¹ ...shown with white backgrounds ...

² ...shown with colored backgrounds ...

Table 1.3: Truth table for logical conjunctions and disjunctions.

p	q	$p \wedge q$	$p \vee q$
\top	\top	\top	\top
\top	\perp	\perp	\top
\perp	\top	\perp	\top
\perp	\perp	\perp	\perp

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

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.

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*.² 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 \rightarrow symbol and defined in *table 1.5*.

$p_{1.10} :=$ “Espresso nourishes the soul.”

$q_{1.10} :=$ “I will drink espresso.”

The antecedent and consequent for $\sigma_{1.10}$ are defined above. With these definitions, we can now write $\sigma_{1.10} \equiv p_{1.10} \rightarrow q_{1.10}$ 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 $p_{1.10} \equiv \perp$, then $q_{1.10} \equiv \perp$. 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 \mathcal{A} in this class, then I will eat my shoe.” (1.11)

If you do happen to make an \mathcal{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*.³ On the other hand, if you make an \mathcal{B} instead, then I can go home with both shoes and conscience intact; in this case, the sentence was also *true*.⁴ However, what if you make the \mathcal{B} but I decide to eat my shoe anyways? Did I lie? No; just because you failed to make an \mathcal{A} doesn't mean I *can't* eat my shoe! All I said was that I definitely would if you made an \mathcal{A} .⁵ That sentence is only a lie when you *do* make an \mathcal{A} in the class, but I refuse to eat my shoe, since I really am breaking my promise then.⁶

In *table 1.6*, we list several ways of verbalising $p \rightarrow q$ in English. Since this connective can be worded in so many unintuitive ways; careful attention must be paid to phrases involving conditionals.

¹ Why might this be? Think about this.

Table 1.5: Truth table for conditionals.

p	q	$p \rightarrow q$	$p \leftrightarrow q$
\top	\top	\top	\top
\top	\perp	\perp	\perp
\perp	\top	\top	\perp
\perp	\perp	\top	\top

² Synonyms for *antecedent* & *consequent*.

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

³ $\top \rightarrow \top \equiv \top$

⁴ $\perp \rightarrow \perp \equiv \top$

⁵ $\perp \rightarrow \top \equiv \top$

⁶ $\top \rightarrow \perp \equiv \perp$

"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."

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.

A Formal Proposition

Definition 1.3 (Proposition).

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

1. $\lambda = \top$ or $\lambda = \perp$.
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.

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.

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

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.

定義

This definition works by first establishing as our *basis* that \top and \perp 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 lets us verify statements like $((\neg\top) \wedge (\perp \wedge \top)) \rightarrow \top$ are indeed propositions by recursively decomposing it until we reach the bases.

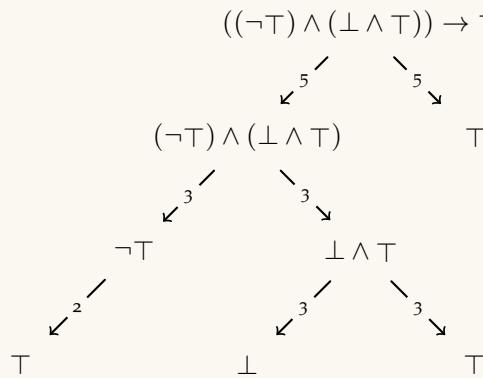
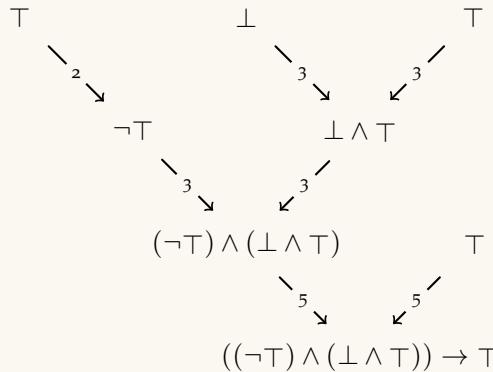


Figure 1.2: In this example, we have dropped some unambiguous parentheses for clarity. Notice, however, that some parentheses *cannot* be dropped: for example, those around the premise of the \rightarrow conditional, and those separating the arguments of the two \wedge conjunctions. If those parentheses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

Alternatively, think of this as *inductive bootstrapping*.¹ Beginning with \top and \perp from (1) as our initial instances of propositions, we then build larger propositions like $\neg\perp$ and $\top \wedge \perp$, 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 $((\neg\top) \wedge (\perp \wedge \top)) \rightarrow \top$. 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.



¹ “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. 定義

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

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 $p \rightarrow q$ is equivalent to its *contrapositive* $\neg q \rightarrow \neg p$.

For example, it's clear that $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$, 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. In their joint truth table, the output columns for the two expressions are identical.

Equivalent propositions are *essentially the same* when we view them 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 $\varphi(p_1, p_2, \dots, p_n)$ and $\psi(p_1, p_2, \dots, p_n)$ consisting of the same variables, then answering $\varphi(p_1, p_2, \dots, p_n) \stackrel{?}{=} \psi(p_1, p_2, \dots, p_n)$ requires computing truth values for φ and ψ with *all possible combinations* of truth assignments to p_1, p_2, \dots, p_n and checking that they match.

Now, p_1 can either be \top or \perp . 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 \perp . 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*.²

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

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

*proof
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*

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.

² 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.

of a truth assignment on which the propositions disagree. Instead of an entire truth table, all we need is a *single row*.

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

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

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

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 $\varphi \equiv \psi$.

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 $\varphi \not\equiv \psi$. 定義

logical equivalence
≡
logical non equivalence
≠

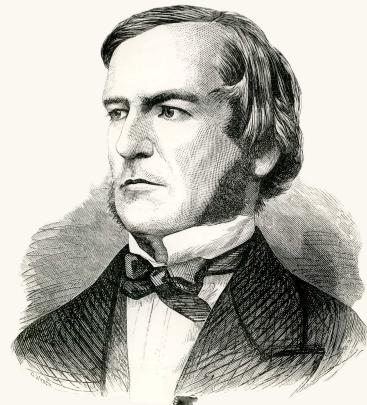


Figure 1.4: George Boole, a largely self-taught mathematician, logician, and philosopher, first described the eponymous Boolean algebra in his 1854 monograph *The Laws of Thought*.

1.3 The Propositional Logic

Axioms and 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	$\top \wedge p \equiv p$	$\perp \vee p \equiv p$
COMPLEMENT	$\neg p \wedge p \equiv \perp$	$\neg p \vee p \equiv \top$
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		$p \rightarrow q \equiv \neg p \vee q$
BICONDITIONAL DISINTEGRATION	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$	

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*

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).

true *a priori*, without any need for verification. Instead, they will form the *basis* upon which we build proofs of *other* statements.

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 \perp ; 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

¹ A *theorem* is a provable proposition.

Theorem 1.1 (Uniqueness of Complements).

If $p \wedge q \equiv \perp$ and $p \vee q \equiv \top$, then $\neg p \equiv q$ for all propositions p, q . 定理

Proof. Let p and q be arbitrary propositions.² Assume $p \wedge q \equiv \perp$ and $p \vee q \equiv \top$.³ 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.

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

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

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

This gives us $q \equiv \neg p \wedge q$. Thus, we conclude $\neg p \equiv \neg p \wedge q \equiv q$. Q.E.D.

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

² 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.

³ 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 “Οπερ ἔδει δεῖξαι. This is called a *tombstone*, and it is a traditional way of denoting the end of a proof. Modern authors might use \square or \blacksquare instead.

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.¹ The first sentence of the proof introduces these two arbitrary propositions.

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 then 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.² What follows next is a sequence of logical statements, each of which is *justified*,³ 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.

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.

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 $\top \equiv \neg\perp$ and $\perp \equiv \neg\top$, based on the way we use the words *true* and *false* in natural language and how \top and \perp are meant to correspond to those truth values. We can now prove this as a *corollary*—a simple consequence—of *theorem 1.1*.

¹ 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.

² If our conclusion were a longer, compound statement, we would continue breaking the problem down *recursively* until we were left with something *atomic*.

³ ...either by a *definition*, an *axiom*, an *assumption* we've made, or a *prior theorem* we've proven ...



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

Corollary 1.1.

$\top \equiv \neg\perp$ and $\perp \equiv \neg\top$.

推論

Proof. Observe that $\perp \wedge \top \equiv \perp$ by the *identity* axiom. Similarly, we have that $\perp \vee \top \equiv \top \vee \perp \equiv \top$ by *commutativity* and the *identity* axiom again. So, we can apply *theorem 1.1*¹ and conclude $\top \equiv \neg\perp$. Similarly, we can observe that $\top \wedge \perp \equiv \perp \wedge \top \equiv \perp$ by *commutativity* and *identity*, and $\top \vee \perp \equiv \perp$ by the *identity* axiom. Thus, $\perp \equiv \neg\top$ by *theorem 1.1*.

Q.E.D.

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 \perp because this is a fact 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 $\neg p \equiv \neg q$.

推論

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

$$\begin{aligned} q \wedge \neg p &\equiv p \wedge \neg p && \text{because we assumed } p \equiv q \\ &\equiv \perp && \text{by commutativity and complement} \end{aligned}$$

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

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

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 $r \equiv s$, the following.

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

推論

We include *corollary 1.3* above just for completeness, so that some of the basic properties of \equiv are codified somewhere; their proofs are not particularly interesting. We are now ready to tackle the proof of a claim you probably find so obvious as to not even be worth mentioning.

¹ 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, $p := \perp$ and $q := \top$). That means, having satisfied the premises, we get to assert the conclusion, justified by that theorem.

Theorem 1.2 (Double Negation).

For any proposition p , we have that $p \equiv \neg\neg p$.

定理

Proof. Let p be a proposition. We will show p acts like the negation of $\neg p$. Observe $\neg p \wedge p \equiv p \wedge \neg p \equiv \perp$ by *commutativity* and the *complement* axiom. Similarly, $\neg p \vee p \equiv p \vee \neg p \equiv \top$ by *commutativity* and *complement*. Therefore, $p \equiv \neg(\neg p)$ by *theorem 1.1*. Q.E.D.

Our goal for the rest of this section will be to prove *De Morgan's laws*. This will finally establish the establish dual relationship between conjunctions and disjunctions. For this, we first need a few more tools.

Theorem 1.3 (Idempotence).

For any proposition p , we have $p \wedge p \equiv p$ and $p \vee p \equiv p$.

定理

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

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

An analogous chain of reasoning takes us through the disjunctive case.¹

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

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

¹ Notice that we have combined some steps here involving *commutativity*; when it is clear, we can save some space by combining *commutativity* with the step directly proceeding it. We do not yet have the maturity to combine any other steps.

Theorem 1.4 (Domination).

For any proposition p , we have $\top \vee p \equiv \top$ and $\perp \wedge p \equiv \perp$.

定理

Proof. Let p be a proposition and observe.

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

The dual case works out in nearly the same manner.

$$\begin{aligned}
 \perp \wedge p &\equiv p \wedge \perp && \text{by commutativity} \\
 &\equiv p \wedge (p \wedge \neg p) && \text{by complement} \\
 &\equiv (p \wedge p) \wedge \neg p && \text{by associativity} \\
 &\equiv p \wedge \neg p && \text{by idempotence} \\
 &\equiv \perp && \text{by complement}
 \end{aligned}$$

We therefore conclude $p \vee \top \equiv \top$ and $p \wedge \perp \equiv \perp$. Q.E.D.

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 any p and q . 定理

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 $p \wedge q$, satisfying the premise of *uniqueness of negations*. We then apply the theorem to obtain $\neg(p \wedge q) \equiv \neg p \vee \neg q$. We will leave the other half—proving $\neg(p \vee q) \equiv \neg p \wedge \neg q$ —as an exercise to the reader.

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

In the conjunctive branch above, we derived $(p \wedge q) \wedge (\neg p \vee \neg q) \equiv \perp$.

We now prove $(p \wedge q) \vee (\neg p \vee \neg q) \equiv \top$ in the disjunctive branch below.

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

Therefore, by *theorem 1.1*, we conclude $\neg(p \wedge q) \equiv \neg p \vee \neg q$ as desired. Q.E.D.

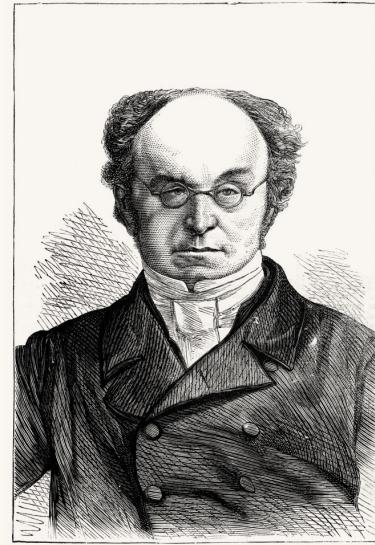


Figure 1.6: [Augustus De Morgan](#), after whom these laws are named, is also notable for his work on logical quantification and mathematical induction.

Rules of Inference

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

Table 1.10: The 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.

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 \perp 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 permit *one-way, inferential* arguments. 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.

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

¹ “ Γ proves φ ” or “ φ follows from Γ .”

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

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.¹

The next rule, named *reductio ad absurdum*,² 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 ($q \wedge \neg q \equiv \perp$). 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.³

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 . 定理

Proof. Let p and q be arbitrary propositions, and suppose $\neg q$ and also $p \rightarrow q$. We know that $p \rightarrow q \equiv \neg q \rightarrow \neg p$,⁴ so we have $\neg q \rightarrow \neg p$. Then, by *modus ponens*, we can conclude $\neg p$. Q.E.D.

¹ *Modus tollens* is short for the Latin phrase *modus tollendo tollens*, literally “*the method of removing by taking away*.”

² *Reductio ad absurdum* is a Latin phrase meaning to “*reduce to absurdity*.” This has also been called *argumentum ad absurdum*.

³ 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.

⁴ This result—that a conditional statement is equivalent to its *contrapositive*—is left as an exercise to the reader.

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*. There are several classical *syllogisms* that have been studied since the time of the ancient Greeks. Before discussing these, we will first prove three important theorems.

Hilbert's System

The logical system we've set up so far—the axioms that establish the propositional calculus as a Boolean algebra, and our comprehensive rules of inference—is very user-friendly, but for this reason it is not *minimal*. We could have made our logical system more “*elegant*”—in some eyes—by choosing a shorter list of axioms and relying on only one rule of inference, at the consequence of having *much uglier* theorems and substantially more tedious proofs. Nonetheless, there is still benefit to be had by studying one of these *more minimal* axiomatizations, as it will provide us invaluable insight into proving a very important theorem: *conjunction elimination*. This alternative axiomatization for the propositional calculus is attributed to Hilbert and Frege. A modern, more condensed version of their system can be written using only two axioms without losing any expressive power. We now prove their axioms as *theorems of our system* below.

Theorem 1.7 (Hilbert's First Axiom).

$$\vdash \varphi \rightarrow (\psi \rightarrow \varphi) \text{ for any propositions } \varphi \text{ and } \psi. \quad \text{定理}$$

Proof. Let φ and ψ be arbitrary propositions and assume φ .

Suppose ψ . We have φ by assumption. Thus, we have $\psi \vdash \varphi$ since we derived φ from ψ . By the *deduction rule*, we then obtain $\psi \rightarrow \varphi$.

We now have $\varphi \vdash (\psi \rightarrow \varphi)$, since we derived $\psi \rightarrow \varphi$ from φ . Therefore, we conclude $\varphi \rightarrow (\psi \rightarrow \varphi)$ by the *deduction rule*. Q.E.D.

Theorem 1.8 (Hilbert's Second Axiom).

$$\vdash (\varphi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)) \text{ for any } \varphi, \psi, \xi. \quad \text{定理}$$

Proof. Let φ , ψ , and ξ be propositions and assume $\varphi \rightarrow (\psi \rightarrow \xi)$. We want to show $(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)$. Towards that goal, assume $\varphi \rightarrow \psi$. We now want to show $\varphi \rightarrow \xi$; so, towards this goal, assume φ . Now,

$$\varphi, (\varphi \rightarrow (\psi \rightarrow \xi)) \vdash \psi \rightarrow \xi$$

by *modus ponens* using our earlier assumption, so we obtain $\psi \rightarrow \xi$. Again, by applying *modus ponens* to our prior assumption, we see that

$$\varphi, (\varphi \rightarrow \psi) \vdash \psi$$

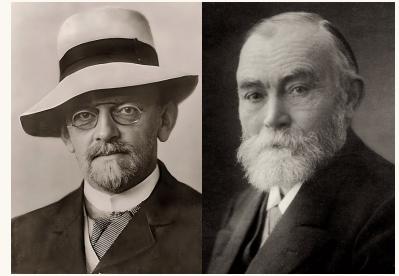


Figure 1.7: **David Hilbert** and **Gottlob Frege** were two of the most influential figures in the *logicist program* that was attempting to reduce mathematics to pure logic. Outside of logic, Hilbert was an extremely accomplished algebraist (maybe you've heard of Hilbert spaces in the context of linear algebra). Frege, while underappreciated during his life, is now recognized as one of the greatest and most profound mathematicians and philosophers of language of human history.

leaves us with ψ . We now take these two intermediate results to deduce

$$\psi, (\psi \rightarrow \xi) \vdash \xi$$

using *modus ponens*. Therefore, we have derived ξ from our initial assumption φ , letting us conclude $\varphi \vdash \xi$.

We now apply the *deduction rule* several times to arrive at the conclusion. From $(\varphi \vdash \xi)$, we deduce $(\varphi \rightarrow \xi)$. Next, from $((\varphi \rightarrow \psi) \vdash (\varphi \rightarrow \xi))$, we deduce $((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi))$. Lastly, we take our expression $((\varphi \rightarrow (\psi \rightarrow \xi)) \vdash ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)))$ and finally derive $((\varphi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)))$. Q.E.D.

Classical Syllogisms

We now follow in the footsteps of classical students of rhetoric, who in antiquity would ponder over these (and other) *syllogisms*—a traditional term referring to an argument where a conclusion is drawn from some collection of premises—as a way to hone our skills in the sister arts of proof-writing and deductive reasoning.

The following theorem allows us to construct and follow extended chains of conditional reasoning. Combined with *modus ponens*, this fundamentally forms the basis for any nontrivial argument.

Theorem 1.9 (Hypothetical Syllogism).

We have $(p \rightarrow q), (q \rightarrow r) \vdash p \rightarrow r$ for any propositions p, q, r . 定理

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

The next theorem is the converse of the *deduction rule*. When these two are taken together, they establish the formal, syntactic equivalence between the \rightarrow and \vdash symbols, which are semantically distinct.

Theorem 1.10 (Implication Elimination).

We have $(p \rightarrow q) \vdash (p \vdash q)$ for any propositions p and q . 定理

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

Theorem 1.11 (Conjunction Introduction).

We have $p, q \vdash p \wedge q$ for any propositions p and q . 定理

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

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

So, we have $p \rightarrow \neg q$, from which we can derive $\neg q$ by *modus ponens*. However, since already we had q , we now have a contradiction. \sharp ²

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

In table 1.11, we summarize these results and some other theorems. We leave the proofs of these as an important list of exercises to the reader.

MODUS TOLLENS	$\neg q, (p \rightarrow q) \vdash \neg p$	
HYPOTHETICAL SYLLOGISM	$(p \rightarrow q), (q \rightarrow r) \vdash p \rightarrow r$	
IMPLICATION ELIM.	$(p \rightarrow q) \vdash (p \vdash q)$	a.k.a. <i>the consolidation rule</i>
CONJUNCTION INTRO.	$p, q \vdash p \wedge q$	a.k.a. <i>adjunction</i>
CONJUNCTION ELIM.	$p \wedge q \vdash p$	a.k.a. <i>simplification</i>
DISJUNCTION INTRO.	$p \vdash p \vee q$	a.k.a. <i>addition</i>
DISJUNCTION ELIM.	$(p \rightarrow r), (q \rightarrow r), (p \vee q) \vdash r$	a.k.a. <i>proof by cases</i>
EX FALSO QUODLIBET	$p, \neg p \vdash q$	a.k.a. <i>explosion</i>
CONSTRUCTIVE DILEMMA	$(\alpha \rightarrow \gamma), (\beta \rightarrow \delta), (\alpha \vee \beta) \vdash \gamma \vee \delta$	

¹ When beginning a *proof by contradiction*, it is good form to explicitly alert the reader to this fact with a phrase like “*towards a contradiction*.”

² The symbol \sharp is useful in *proofs by contradiction* to highlight to the reader where the *contradiction* is and when it is reached.

Table 1.11: Some useful theorems.

2

First-Order Logic

"I am in a charming state of confusion."

– Ada Lovelace

The language we have described so far is often called the *classical logic*—since this is a modern development on Aristotelian logic—or the *propositional logic* because its basic syntactic unit is the proposition. Having the proposition as the most granular accessible referent helps keep this language manageable, but it will hold us back from being as expressive as we'd like to be. For example, suppose we are hungry, and in the course of our ruminations we discover that shepherd's pie is irresistibly delicious. We also happen to know the same thing about paella. Having recognized these facts, no simple substitute will do: we *must* have one of these two meals if we are to be satisfied at all. How might we express this logically? Let's introduce some definitions.

$s := \text{"We eat shepherd's pie."}$

$p := \text{"We eat paella."}$

$n := \text{"We do not eat anything."}$

The claim we are trying to express would formally look as follows.

$$(\neg s \wedge \neg p) \rightarrow n \quad (2.1)$$

From the syntax above, it doesn't seem like there is any relationship between the premise of that conditional statement and its conclusion. In fact, there doesn't even appear to be a relationship between s and p , even though they are both saying something really similar, because *syntactically* they just look like two distinct propositions! Suppose our friend felt the same way as we do about food, but he additionally knew about a *secret third food*: the tostada. Our friend might then resolve to have *that* meal as a fall-back if he can't get his hands on shepherd's pie or paella. He would let $t := \text{"We eat a tostada."}$ and say the following.

$$(\neg s \wedge \neg p) \rightarrow t \quad (2.2)$$



English *shepherd's pie*, as God intended.



The humble *paella*, national dish of Spain.



Tostada & café, a classic Cuban breakfast.

Now, despite our two claims having the *exact same syntactic form*, they express remarkably different ideas. To realize this, think about what it would take to prove (2.1): after verifying $\neg s$ and $\neg p$, we would then need to show we did not eat *any other food!* This is a *universal claim* we are making about *all* possible meals. However, our friend is not making this kind of claim: his conclusion is simply that *there exists* a particular meal he eats if $\neg s$ and $\neg p$ are satisfied. To prove himself right, he simply has to show that he ate that particular meal.

2.1 A More Expressive Language

It will quickly become frustrating for our language to limit our expressivity like this. The missing component in our language is the ability to distinguish the *object* of our speech from the *predicate* description we make about it when we declare a proposition.

Every man is mortal.

Socrates is a man.

\therefore Socrates is mortal.

The argument above seems like a clear, sensible argument; it in fact looks like a simple application of *modus ponens*. Yet, we realize that a proof of this argument in the propositional logic could not actually invoke *modus ponens*. There is no way to symbolize the first sentence in such a way that we obtain a conditional $x \rightarrow y$ where the premise is “Socrates is a man,” and if we can’t do that then we can’t apply *modus ponens*. We fix this issue by augmenting our language with the ability to *syntactically* distinguish between *predicates* and the *terms* they describe.

Definition 2.1 (Term).

term

A *term* is a symbol denoting an object. Specific terms—e.g., the natural number 5, Socrates, shepherd’s pie—are called *constants*. Placeholder terms denoting objects that have not been specifically determined are called *variables*. Notice that *terms, on their own, do not form complete sentences!* A term does not have a truth value! 定義

predicate

Let x_1, \dots, x_n be variable symbols. We say $\varphi(x_1, \dots, x_n)$ is an *n-ary predicate* if replacing each of the n variables x_1, \dots, x_n by terms t_1, \dots, t_n from our results in a *proposition* $\varphi(t_1, t_2, \dots, t_n)$, carrying a truth value. The collection of all terms that our language has referential access to is our *universe of discourse*. 定義

universe of discourse

We’ve now introduced a new problem into our language though. Suppose we have define the predicates $\mu(x) := "x \text{ is a man}"$ and

$\theta(x) := "x \text{ is mortal}"$ in an attempt to translate the previous argument. We can now translate the second premise and conclusion as $\mu(\text{Socrates})$ and $\theta(\text{Socrates})$ respectively. But we still can't translate the first line. For this, we need the ability to express *quantities*.

Let $\varphi(x_1, \dots, x_i, \dots, x_n)$ be an n -ary predicate containing a variable x_i . The *universal quantification* of the variable x_i appearing in φ is denoted $\forall x(\varphi(x_1, \dots, x, \dots, x_n))$ and says *any constant replacing x will satisfy φ* .

```
universal
∀
def forall(universe, predicate):
    for x in universe:
        if not predicate(x):
            return False
    return True
```

"*For all x , $\varphi(x_1, \dots, x, \dots, x_n)$.*"

Figure 2.1: A hypothetical implementation of $\forall x(\varphi(x))$. If it returns `False`, then there is at least one x in `universe` such that `predicate(x) == False`, which is equivalent to $\forall x(\varphi(x)) \equiv \perp$. Otherwise, every x satisfies `predicate(x) == True`, meaning $\forall x(\varphi(x)) \equiv \top$.

existential
 \exists
 x
 $\exists x(\varphi(x_1, \dots, x, \dots, x_n))$

The *existential quantification* of x_i is denoted $\exists x(\varphi(x_1, \dots, x, \dots, x_n))$ and claims that *there is at least one* constant that, in place of x , satisfies φ . The *scope* of a quantifier is denoted by parentheses specifying its variable's lifetime; that variable is *bound* to that quantifier within that scope. A variable that is not bound to any quantifier is called *free*. Statements with free variables *cannot have truth values*, they do not carry *meaning*. If a statement has free variables, those variables need to either be replaced by *terms*, or be bound to a *quantifier*. Because this will be useful in the

```
free variable
def exists(universe, predicate):
    for x in universe:
        if predicate(x):
            return True
    return False
```

"*There exists x such that $\varphi(x_1, \dots, x, \dots, x_n)$.*"

Figure 2.2: A hypothetical implementation of $\exists x(\varphi(x))$. If `True` is returned, then there must be an x in `universe` such that `predicate(x) == True`, which is equivalent to $\exists x(\varphi(x)) \equiv \top$. Otherwise, every x satisfies `predicate(x) == False`, so that $\exists x(\varphi(x)) \equiv \perp$.

unique existential
 $\exists!$
 $\exists! x(\varphi(x_1, \dots, x, \dots, x_n))$

future, we also introduce the *unique existential quantification* of x_i as a way of saying that *there is exactly one* constant satisfying φ in place for x . We use the notation $\exists! x(\varphi(x_1, \dots, x, \dots, x_n))$ to denote this, and read this in English as "*there exists a unique x such that $\varphi(x)$.*"

$$\exists! x(\varphi(x)) \Leftrightarrow \exists x \left(\varphi(x) \wedge \forall y \left(\varphi(y) \Rightarrow (y = x) \right) \right).$$

This is a special case of existential quantification; using the unique existential quantifier means making an existential claim *and additionally asserting that only one such example exists*. So, we define the $\exists!$ quantifier *in terms of* the \exists quantifier. Be careful to note that the $!$ symbol in $\exists!$ does not correspond with negating anything! Do not make the mistake of confusing $!$ with \neg if you have experience with a programming language where the $!$ syntax corresponds to logical negation.

Forming Formulae Well

Definition 2.3 (Formulae).

We say a formula φ is *atomic* if it satisfies the following recurrence.

1. $\varphi = \top$ or $\varphi = \perp$.
2. $\varphi = \psi(t_1, \dots, t_n)$, where ψ is an n -ary predicate, t_1, \dots, t_n are terms.

We say λ is a *well-formed formula*—often abbreviated *wff*—if it satisfies the recurrence relation below.

1. λ is an atomic formula.
2. $\lambda = \neg(\varphi)$, where φ is a wff.
3. $\lambda = (\varphi) \wedge (\psi)$, where φ and ψ are wff.
4. $\lambda = (\varphi) \vee (\psi)$, where φ and ψ are wff.
5. $\lambda = (\varphi) \rightarrow (\psi)$, where φ and ψ are wff.
6. $\lambda = (\varphi) \leftrightarrow (\psi)$, where φ and ψ are wff.
7. $\lambda = \forall x(\varphi)$, where φ is a wff.
8. $\lambda = \exists x(\varphi)$, where φ is a wff.

A well-formed formula with no free variables is called a *sentence in the first-order logic*. Looking at the above definitions, a *wff* that has no free variables will boil down to a *proposition*, meaning it will have a definite, unambiguous truth value. Sentences will be our primary mode for expressing conjectures, theorems, and proofs.

定義

2.2 Rules of Inference

UNIVERSAL INTRODUCTION	$\varphi(t)$ for an <i>arbitrary</i> $t \vdash \forall x(\varphi(x))$	If we know $\varphi(t)$ and t is <i>arbitrary</i> , then we can say $\forall x(\varphi(x))$.
UNIVERSAL ELIMINATION	$\forall x(\varphi(x)) \vdash \varphi(t)$ for <i>any term</i> t	If we have $\forall x(\varphi(x))$, then we can pick <i>any</i> t and say $\varphi(t)$.
EXISTENTIAL INTRODUCTION	$\varphi(t)$ for a <i>particular</i> $t \vdash \exists x(\varphi(x))$	If we know $\varphi(t)$ for a <i>specific</i> term t , then we can say $\exists x(\varphi(x))$.
EXISTENTIAL ELIMINATION	$\exists x(\varphi(x)) \vdash \varphi(t)$ for a <i>new term</i> t	If we have $\exists x(\varphi(x))$, then we have $\varphi(t)$ for some t that has not yet appeared.

When we were building the propositional logic, we first defined a *syntax* for our logic by introducing the logical connectives and some other special symbols; we then gave it an *algebraic semantics* when we introduced the equivalence axioms and the rules of inference. Now that we are augmenting our language with *terms*, *predicates*, and *quantifiers*, we have a similar need to establish semantics for interpreting our

Table 2.1: The rules of inference for quantified expressions involving predicates. Note that the “*new term*” referred to by *existential elimination* must be a symbol that has not yet appeared in your proof.

new symbols. We introduce these rules in *table 2.1*. In addition, we have three important theorems involving quantified expressions, each containing a *universal* fragment and an *existential* fragment. This first theorem establishes a form of *De Morgan duality* between the \forall and \exists quantifiers: *negating* a quantified sentence is equivalent to quantifying the *negated* sentence using the *other* quantifier.

Theorem 2.1 (Negation of Quantifiers).

If φ is a predicate of at most one free variable, these equivalences hold.

$$\neg\forall x(\varphi(x)) \equiv \exists x(\neg\varphi(x)) \quad \neg\exists x(\varphi(x)) \equiv \forall x(\neg\varphi(x))$$

定理

The next theorem illustrates a sort of *distributive law* for quantifiers. Be sure to *pay careful attention to the parentheses* in the following theorem.

Theorem 2.2 (Distribution of Quantifiers).

Let φ be a predicate of at most one free variable and p be a proposition. The four equivalences below are then satisfied; mind the parentheses.

$$\begin{aligned} \forall x(\varphi(x)) \wedge p &\equiv \forall x(\varphi(x) \wedge p) & \exists x(\varphi(x)) \wedge p &\equiv \exists x(\varphi(x) \wedge p) \\ \forall x(\varphi(x)) \vee p &\equiv \forall x(\varphi(x) \vee p) & \exists x(\varphi(x)) \vee p &\equiv \exists x(\varphi(x) \vee p) \end{aligned}$$

Further, if ψ is also a predicate with at most one free variable and t is a term, then the following four one-way inferences hold.

$$\begin{aligned} \forall x(\varphi(x) \wedge \psi(x)) &\vdash \forall x(\varphi(x) \wedge \psi(t)) & \exists x(\varphi(x) \wedge \psi(t)) &\vdash \exists x(\varphi(x) \wedge \psi(x)) \\ \forall x(\varphi(x) \vee \psi(x)) &\vdash \forall x(\varphi(x) \vee \psi(t)) & \exists x(\varphi(x) \vee \psi(t)) &\vdash \exists x(\varphi(x) \vee \psi(x)) \end{aligned}$$

However, those inferences above are *not* equivalences, as shown below.

$$\begin{aligned} \forall x(\varphi(x) \wedge \psi(t)) &\nvDash \forall x(\varphi(x) \wedge \psi(x)) & \exists x(\varphi(x) \wedge \psi(x)) &\nvDash \exists x(\varphi(x)) \wedge \psi(t) \\ \forall x(\varphi(x) \vee \psi(t)) &\nvDash \forall x(\varphi(x) \vee \psi(x)) & \exists x(\varphi(x) \vee \psi(x)) &\nvDash \exists x(\varphi(x)) \vee \psi(t) \end{aligned}$$

Finally, the following four equivalences hold for conditional statements.

$$\begin{aligned} \forall x(\varphi(x) \rightarrow p) &\equiv \exists x(\varphi(x)) \rightarrow p & \forall x(p \rightarrow \varphi(x)) &\equiv p \rightarrow \forall x(\varphi(x)) \\ \exists x(\varphi(x) \rightarrow p) &\equiv \forall x(\varphi(x)) \rightarrow p & \exists x(p \rightarrow \varphi(x)) &\equiv p \rightarrow \exists x(\varphi(x)) \end{aligned}$$

定理

The third and final theorem concerns the *order of quantifiers*, importantly pointing out that *quantifiers don't necessarily commute with each other*.

Theorem 2.3 (Quantifier Shift).

If φ is a predicate of at most two free variables, then the following hold.

$$\begin{aligned} \forall x\forall y(\varphi(x, y)) &\equiv \forall y\forall x(\varphi(x, y)) & \exists x\exists y(\varphi(x, y)) &\equiv \exists y\exists x(\varphi(x, y)) \\ \forall x\exists y(\varphi(x, y)) &\nvDash \exists y\forall x(\varphi(x, y)) & \exists x\forall y(\varphi(x, y)) &\vdash \forall y\exists x(\varphi(x, y)) \end{aligned}$$

定理

2.3 The Art of Writing Proofs

The way approach a proof of a statement principally depends on the *form* of the what we're trying to prove. Depending on what the statement *looks* like, a valid proof may be allowed to take certain liberties or be required to satisfy certain constraints. We will end this chapter with some words of advice for writing proofs based on the rules of inference we have established and the semantic interpretation we have attached to our various logical symbols. Since propositions and sentences in the first-order logic are *recursive* constructions, the first thing we should do when presented a statement to prove is to *recursively* analyze its *form*.

Quantified Formulae

If we are trying to prove a statement like $\forall x(\varphi(x))$, we can *check* $\varphi(t)$ for all possible values of t . This is usually not possible, as our universe of discourse often contains infinitely many objects. The natural alternative is to *introduce an arbitrary term t* and, without making any assumptions about t , to show that t satisfies φ . If we manage to do this without relying on any details pertaining to t specifically, then our argument *will generalize universally*. On the other hand, to prove a statement of the form $\exists x(\varphi(x))$, the task is to *find a specific object t* that we can prove satisfies φ . Existential claims are often the *most difficult* kind to prove because there is, generally, no clear strategy for *how t* should be found.

Conditional Statements

Suppose we have a statement we want to prove that takes the form of a conditional $p \rightarrow q$. These are *by far the most common* kinds of statements we will be interested in proving. This involves showing we can derive q from p , so we first *assume p* in order to get to q . After assuming p is the case, we can think of how to derive q based on its *form* by again going through this analysis. Alternatively, instead of showing $p \rightarrow q$ directly, we can always think to prove $\neg q \rightarrow \neg p$ and apply our knowledge that a conditional statement is always equivalent to its contrapositive.

Junctions

Statements that look like $p \wedge q$ are relatively straight-forward: we have to show that *both p and q* are true. Similarly, showing $p \vee q$ requires deriving *one of either p or q*, but we are free to choose which one to pursue. Naturally, this will depend on what forms p and q take.

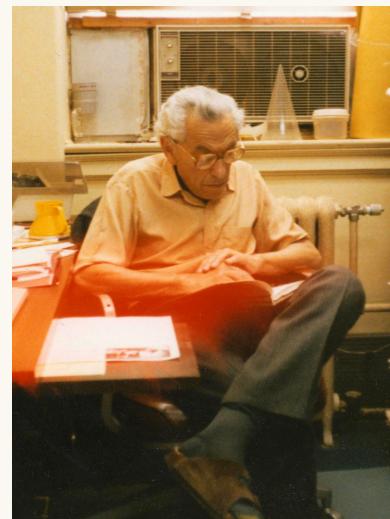


Figure 2.3: “The purpose of life is to prove and to conjecture.” – Paul Erdős



Figure 2.4: “Another roof, another proof.” – Paul Erdős

Nonconstructive Proofs

When, in the course of human events, it becomes necessary for one people to encounter a *contradiction*, a decent respect to the opinions of mankind requires that they should *reject the assumptions* that impelled them there. What we mean by this is: if you are ever feeling like a proposition p is *obviously true*, but its proof feels insurmountable, try *assuming $\neg p$* and seeing what happens. If this leads you to a *contradiction*, then you can invoke *reductio ad absurdum* and conclude p , washing your hands of the situation.

Ex falso quodlibet can be treated as a cousin to *reductio ad absurdum*. It is nowhere near as commonly used as a mode of reasoning, and to many it is far less intuitive than a simple proof by contradiction would be, but there are situations when it can be used to shortcut a proof in only a couple of lines. Keep an eye out for situations in which you are asked to prove a conditional statement $p \equiv \perp \rightarrow q$ with a *false premise* because this rule will let you immediately reach your conclusion.

Mathematics

3

Foundations

"Finally I am becoming stupider no more."

– Paul Erdős

With the development of the first-order logic, we finally have a formal language for rigorous communication. This language has several incredibly nice properties: it's sufficiently expressive to prove any *universal* truth, while not being so unwieldy as to admit falsehoods or contradictions. The development of the first-order logic—along with Gödel's completeness and incompleteness theorems—marks one of humanity's greatest intellectual achievements, which would have ramifications throughout nearly every field of philosophy and natural science. With this language in hand, we are now ready to embark on our studies of *mathematics* proper. The natural first question we have to answer is: *what is our universe of discourse?* What *are* mathematical objects?

3.1 Informal Notions

Thanks to insights made throughout the 20th and 21st centuries, there are actually several competing ways to answer this question (though the most *modern* and “*computer science*” of these formalisms would have required us to take a different logical foundation than the one we did). We will be taking a mainstream perspective that is fundamentally based on the concept of a *set*, but we will introduce two other useful kinds of objects in this section for convenience. *Technically speaking*, *every* object in our universe of discourse will be, or could be, *implemented* as a set, but it's often distracting to think of things like numbers as sets. As an analogy, think about the files on your computer. The PDF file you're reading these notes from is, fundamentally, a long binary number stored somewhere in your computer's memory. That number *represents* this PDF in the same way a set can represent a function, or the number 15, but if all you want to do is read these notes then it wouldn't be useful to interact with the binary *implementation* of the PDF.

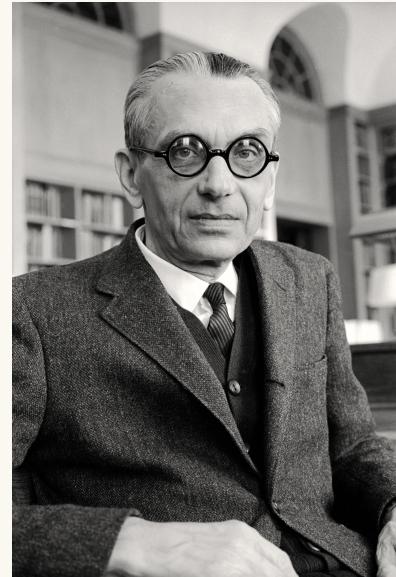


Figure 3.1: Kurt Gödel was an absolutely monumental figure in mathematical logic. He famously showed all universal truths in the first-order logic are provable (a property known as *completeness*). Despite this, he then infamously demonstrated there are *mathematical* truths that *cannot* be proven (the *incompleteness* theorems).

Numbers

The most natural kinds of objects we should feel impelled to discuss are the *numbers*, and the most fundamental kind of number is, naturally, the *natural number*. Informally, these correspond precisely with the *non-negative whole numbers*. We can elegantly characterize these kinds of numbers with the following recurrence.

1. 0 is a natural number.
2. If n is a natural number, then $S(n)$ is a natural number.

In the above recurrence, the notation $S(n)$ —read “*the successor of n* ”—is referring to the “*next (whole) number after n* .” This is the defining characteristic of the natural numbers, from which every other arithmetical property springs forth: begin somewhere (*i.e.*, at zero), and proceed by taking steps (*i.e.*, if n is a natural number, then so its the *next one*).

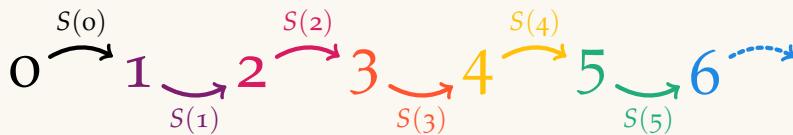


Figure 3.2: An initial segment of the natural number line, which begins at zero.

These will be a very important class of object for us to talk about, so we introduce them into our universe of discourse here. For now, we will be philosophical Platonists in the sense that we will simply believe the natural numbers exist “*out there, somewhere, in the ideal platonic realm of forms*.” After we develop a bit more theory, we will be able to be more concrete about *what* precisely a number *is* formally-speaking.

Functions

A crucial part of the description of the natural numbers we just made is this notion of the *successor* of a natural number n . This idea is usually expressed in terms of the *successor function*, which begs us to define what a *function* is. For the moment, we will say a *function* is an object that *maps inputs from a domain to outputs in a codomain* in a *deterministic way*. Specifically, a function *must produce exactly one output* for each of its valid inputs—the output will not change unless the input changes.¹ If we have a function named f and a valid input x , then the notation we will use to denote the output value f realizes on the input x is $f(x)$.²

$$\forall x \forall y (x = y \Rightarrow f(x) = f(y))$$

With this notation, we express this idea more formally above, taking note that the quantifiers range over the collection of valid inputs for f . We throw function into our universe for now and revisit this later.

¹ Think about this informal definition and see if it agrees with the kinds of things you have been calling “*functions*” throughout your life so far.

² “ *f of x* ,” or “ *f at x* .”

Sets

Since functions are maps that transform inputs into outputs, we are finally driven to ask “*inputs from where?*” All roads eventually lead to the idea of *a collection of things*. Functions map *collections of inputs to outputs*. Polygons are *collection of points*. Numbers measure the sizes of *collection of things*. In fact, any form of speech will find it hard to avoid invoking the concept of *a collection of things* eventually.

A notion of such *fundamental* importance to mathematics should therefore have a central place in our universe of discourse. In the same way binary numbers form a foundation for the files in your computer, we will be building our mathematical universe using *collections* as our fundamental unit of reference. We will call these collections *sets*, and refer to the objects they contain as their *elements*. For example, we might say that the number 0 is an element of the set of all natural numbers.

As the most fundamental and basic object in our universe, we will study these first and *encode* their behavior in the form of *axioms*. Each axiom will incorporate some *intuitive property* that we would expect to be *true* about sets based on their inspiration as “*abstract collections of things*.” This system of axioms—which we will study in the next section—is called *Zermelo-Fraenkel set theory*.

A Note on Notation

	ENTAILMENT	EQUIVALENCE
Language	\rightarrow	\leftrightarrow
Metalanguage	\vdash	\equiv
Mathematics	\Rightarrow	\Leftrightarrow



Figure 3.3: Ernst Zermelo (left) produced one of the first axiomatizations of set theory in 1908, which was augmented in 1922 by Abraham Fraenkel (right) and also, independently, by Thoralf Skolem.

Table 3.1: With the rules of inference and the theorems in table 1.11, we recognize the equivalence between \rightarrow , \leftrightarrow syntactically and \vdash , \equiv semantically. To simplify our notation, we replace these symbols with \Rightarrow , \Leftrightarrow respectively.

As a final note, *we will be simplifying our notation from this section forward*. We had previously been introduced to the symbols \rightarrow and \leftrightarrow for expressing conditional statements *within* the language of the first-order logic. In the *metalanguage*—the language we are using right now to *talk about* the formal system we built—we used the \vdash symbol to denote that some conclusions are derivable from some premises, and we used \equiv to denote that two statements were logically indistinguishable. Given the theorems we proved in the last few chapters, the line between these two classes of symbols has been made blurrier, and it’s typical in mainstream mathematical practice to ignore this distinction entirely. So, we now introduce the symbol \Rightarrow to denote entailment as a replacement for the \rightarrow and \vdash symbols. Similarly, we introduce \Leftrightarrow as a replacement for \leftrightarrow and \equiv , denoting logical equivalence in all contexts.

3.2 Set Theory

A *set* is an abstraction of the idea of a *collection of objects*. This idea, carried forward, naturally implies the need to communicate two kinds of relationships between objects: *equality* and *elementhood*. These will be the two basic predicate symbols of our *theory of sets*.

In order to identify objects that are the same, we introduce the binary *equality* predicate: given two objects x and y , we say $x = y$ precisely when x is *identical* to y . If you've seen the $=$ symbol before in your life, this is exactly the same symbol you're used to, and it has the natural properties you would expect of a predicate called "*equality*".

- 1. $\forall x(x = x)$ *reflexivity*
- 2. $\forall x\forall y((x = y) \Rightarrow (y = x))$ *symmetry*
- 3. $\forall x\forall y\forall z(((x = y \wedge y = z)) \Rightarrow (x = z))$ *transitivity*

You'll notice that these are precisely the same three properties that *logical equivalence* had; these are both examples of *equivalence relations*. We will assume these three statements about equality axiomatically.

The second, and more interesting, predicate relates sets to the elements they contain. We call this predicate *elementhood* and denote it with the \in symbol. These two predicates are enough to express anything we could possibly want about sets. As an example, suppose that \mathcal{A} is a set. By saying $(0 \in \mathcal{A}) \wedge (1 \in \mathcal{A})$, we are saying that \mathcal{A} contains both 0 and 1 as elements, and by saying $2 \notin \mathcal{A}$ we claim that 2 is *not* an element of \mathcal{A} . However, saying $(0 \in \mathcal{A}) \wedge (1 \in \mathcal{A})$ doesn't prevent \mathcal{A} from possibly containing *more* elements. If we wanted to say that \mathcal{A} contains *only* the elements 0 and 1, we would have to assert that $0 \in \mathcal{B}$ and that $1 \in \mathcal{B}$, but we would also need to say $\forall x(x \in \mathcal{B} \Rightarrow (x = 0 \vee x = 1))$. This asserts that not only are 0 and 1 among the elements of \mathcal{A} , but that any element of \mathcal{A} *must* be one of those two. Now, notice that we can rewrite $0 \in \mathcal{A} \wedge 1 \in \mathcal{A}$ as $\forall x((x = 0 \vee x = 1) \Rightarrow x \in \mathcal{A})$. So, saying that \mathcal{A} contains *exactly* the elements 0 and 1 tells us that being 0 or being 1 is an *equivalent condition* for being an element of \mathcal{A} . We can write this as $\forall x(x \in \mathcal{A} \Leftrightarrow (x = 0 \vee x = 1))$. This would be a lot to write every time, so let's introduce some *notation*.

Definition 3.1 (Set Notation).

Given finitely many terms x_0, x_1, \dots, x_{n-1} , we denote by $\{x_0, x_1, \dots, x_{n-1}\}$ the set whose elements are *exactly* the objects x_0, x_1, \dots, x_{n-1} . We write out each element of the set explicitly, separating the elements with commas, with the understanding that the following is *true* for any z .

"The set containing x_0, x_1, \dots, x_{n-1} ."

$$z \in \{x_0, x_1, \dots, x_{n-1}\} : \Leftrightarrow (z = x_0) \vee (z = x_1) \vee \dots \vee (z = x_{n-1})$$

This is often called *set builder notation*. From *figure 3.5*, we can use this notation to say $\mathcal{A} = \{0, 0, 1, 2\}$ and $\mathcal{B} = \{0, 2, 1\}$ whereas $\mathcal{C} = \{0, 1, 3, 2\}$. Notice that set builder notation is extremely restrictive; it only lets us describe sets with *finitely many elements*, and it forces us to *write them all out*. What if we want to talk about a set with so many elements that it would be annoying—or impossible—to write them all down? How would we write down the set of *even* natural numbers, or the set of *prime* numbers, or even the set of natural numbers *itself*? To solve this problem, we introduce *set comprehension notation*.

$$z \in \{x \mid \varphi(x)\} \Leftrightarrow \varphi(z)$$

With this notation, we can pick a predicate φ and refer to the collection of all those things that satisfy that predicate by writing $\{x \mid \varphi(x)\}$. In this way, we can refer to the set of even natural numbers by writing $\{x \mid x \in \mathbb{N} \wedge "x \text{ is even}"\}$. We don't yet have a formal way of expressing "*x is even*," but once we do, that will make $x \in \mathbb{N} \wedge "x \text{ is even}"$ a predicate.

"The set of all x such that $\varphi(x)$."

定義

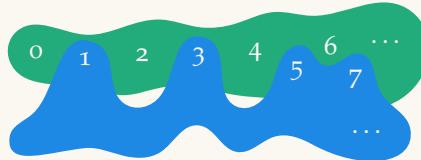


Figure 3.4: The set of *all natural numbers*, the smallest set \mathbb{N} for which $0 \in \mathbb{N}$ and $\forall x(x \in \mathbb{N} \Rightarrow S(x) \in \mathbb{N})$, shown with the subset of *odd natural numbers*.

The elementhood predicate is our fundamental relational symbol (apart from equality) between sets, but this predicate naturally implies another interesting relationship that two sets can share. For example, the blue set in *figure 3.4* represents the set of all odd natural numbers, which we just learned can be written as $\{x \mid x \in \mathbb{N} \wedge "x \text{ is odd}"\}$ using set comprehension notation. It should be pretty clear that every element of this set is a natural number. The same is true about $\{0, 1, 2\}$ and $\{0, 1, 2, 3\}$. Every element of each one of these sets is *also* an element of \mathbb{N} . Taking this further, the elements of $\{0, 1, 2\}$ are each $\{0, 1, 2, 3\}$. This emergent relationship is captured by the definition below.

Definition 3.2 (Subset).

Given two sets \mathcal{X} and \mathcal{Y} , we say that \mathcal{X} is a subset of \mathcal{Y} , denoted with the notation $\mathcal{X} \subseteq \mathcal{Y}$, when every element of \mathcal{X} is also an element of \mathcal{Y} .

$$\mathcal{X} \subseteq \mathcal{Y} \Leftrightarrow \forall z(z \in \mathcal{X} \Rightarrow z \in \mathcal{Y})$$

We can now see $\{x \mid x \in \mathbb{N} \wedge "x \text{ is odd}"\} \subseteq \mathbb{N}$ and $\{0, 1, 2\} \subseteq \{0, 1, 2, 3\}$. However, $\{0, 1, 2, 3\} \not\subseteq \{0, 1, 2\}$ because $3 \in \{0, 1, 2, 3\}$ but $3 \notin \{0, 1, 2\}$.

定義

Infinity

We should keep one thing clear: these *definitions do not assert anything!* Just because we now have the ability to write something down with this new notation *doesn't* mean the notation *refers* to an existing object. To *formally* have sets to talk about, we need to introduce them with either an *axiom* or a *proof*. There is, of course, a set that has been looming over us this whole time—the set of *natural numbers*—that we certainly want to exist. Towards that goal, we introduce one more definition.

Definition 3.3 (Inductive Set).

inductive

We say a set \mathcal{I} is *inductive* if $0 \in \mathcal{I}$ and $\forall x(x \in \mathcal{I} \Rightarrow \mathcal{S}(x) \in \mathcal{I})$. 定義

Axiom 0 (Infinity).

$$\exists x(x \text{ is inductive} \wedge \forall y(y \text{ is inductive} \Rightarrow x \subseteq y)). \quad \text{公理}$$

The set described by *axiom 0* is—in a sense—the “smallest” *inductive set*, which is precisely the set of natural numbers.¹ Therefore, *axiom 0* establishes the *existence of the set of natural numbers*. Once this understanding is clear, it is common to make the following recursive declaration.

$$\mathbb{N} := \left\{ x \mid x = 0 \vee \exists y(y \in \mathbb{N} \wedge x = \mathcal{S}(y)) \right\}$$

The rest of this chapter will introduce *seven axioms* for set theory, each encoding a particular piece of *intuition* about *how sets should behave*.

Extensionality

Sets are entirely determined by their elements. Because sets abstract the idea of *a collection of objects*, everything we need to know about a set should be determined by the elements it contains. Naturally—when we asking “*when are two sets equal?*”—we should expect to answer this question by looking *inside* the sets and comparing their elements.

In *figure 3.5*, we have the sets \mathcal{A} , \mathcal{B} , \mathcal{C} , and \mathcal{D} . We can see that $0 \in \mathcal{A}$, $1 \in \mathcal{A}$, and $2 \in \mathcal{A}$, and we also have that $0 \in \mathcal{B}$, $1 \in \mathcal{B}$, and $2 \in \mathcal{B}$. Even though the elements appear with different frequencies and in different positions between the two sets, it must be that $\mathcal{A} = \mathcal{B}$ because they have all the same elements. However, we can see that $3 \in \mathcal{C}$ while $3 \notin \mathcal{A}$, implying that $\mathcal{A} \neq \mathcal{C}$. In general, we should then expect sets to be equal precisely when they have the same elements, and that sets with the same elements should always be equal.

Axiom 1 (Extensionality).

$$\forall x \forall y((x = y) \Leftrightarrow \forall z(z \in x \Leftrightarrow z \in y)). \quad \text{公理}$$

¹ The fact that the smallest inductive set is actually the set of natural numbers is a *theorem*, but we will not take the time nor effort to prove it here.

This relationship between $=$ and \in is exactly what the *axiom of extensionality* encodes. In our example above, we can now use this axiom to prove that $\mathcal{A} = \mathcal{B}$ by showing that $\forall z(z \in \mathcal{A} \Leftrightarrow z \in \mathcal{B})$. In fact, this is essentially what we've done in the preceding paragraph; because \mathcal{A} and \mathcal{B} are both small, finite sets, by listing all the elements of each set and showing that they're all the same, we have a proof of $\forall z(z \in \mathcal{A} \Leftrightarrow z \in \mathcal{B})$. *Extensionality* tells us this means $\mathcal{A} = \mathcal{B}$.

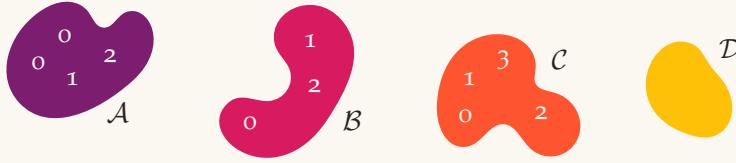


Figure 3.5: A visual representation of two sets. The *purple* set has the same elements as the *red* set, the figures refer to the same set, letting us infer $\mathcal{A} = \mathcal{B}$. The *orange* set contains an element not present in the other two sets, so $\mathcal{C} \neq \mathcal{A}$ and $\mathcal{C} \neq \mathcal{B}$. The *yellow* set has no elements, so it is empty.

By the same token, if we wanted to show that $\mathcal{A} \neq \mathcal{C}$, we would need to show $\neg\forall z(z \in \mathcal{A} \Leftrightarrow z \in \mathcal{C})$. We decompose this statement below.

$$\begin{aligned}\neg\forall z(z \in \mathcal{A} \Leftrightarrow z \in \mathcal{C}) &\equiv \exists z \neg(z \in \mathcal{A} \Leftrightarrow z \in \mathcal{C}) \\ &\equiv \exists z \neg((z \in \mathcal{A} \Rightarrow z \in \mathcal{C}) \wedge (z \in \mathcal{C} \Rightarrow z \in \mathcal{A})) \\ &\equiv \exists z (\neg(z \in \mathcal{A} \Rightarrow z \in \mathcal{C}) \vee \neg(z \in \mathcal{C} \Rightarrow z \in \mathcal{A})) \\ &\equiv \exists z (\neg(z \notin \mathcal{A} \vee z \in \mathcal{C}) \vee \neg(z \notin \mathcal{C} \vee z \in \mathcal{A})) \\ &\equiv \exists z ((z \in \mathcal{A} \wedge z \notin \mathcal{C}) \vee (z \in \mathcal{C} \wedge z \notin \mathcal{A}))\end{aligned}$$

So, what we would need to do is *find* an element z that's *in* one of the two sets but *not in* the other. Since we saw that $3 \in \mathcal{C}$ but $3 \notin \mathcal{A}$, that's exactly what it means for $\mathcal{A} \neq \mathcal{C}$ according to the *axiom of extensionality*.

Lemma 3.1.

$$\forall x \forall y (x = y \Leftrightarrow (x \subseteq y) \wedge (y \subseteq x)). \quad \text{引理}$$

Proof. Let x and y be sets. Observe the following chain of equivalences.

$$\begin{aligned}x = y &\Leftrightarrow \forall z(z \in x \Leftrightarrow z \in y) \quad \text{by extensionality} \\ &\Leftrightarrow \forall z((z \in x \Rightarrow z \in y) \wedge (z \in y \Rightarrow z \in x)) \\ &\Leftrightarrow \forall z(z \in x \Rightarrow z \in y) \wedge \forall z(z \in y \Rightarrow z \in x) \\ &\Leftrightarrow (x \subseteq y) \wedge (y \subseteq x) \quad \text{by definition}\end{aligned}$$

Therefore, $x = y \Leftrightarrow (x \subseteq y) \wedge (y \subseteq x)$. Q.E.D.

What about the set \mathcal{D} ? In the figure, it would seem like \mathcal{D} has no elements at all. *Extensionality* reveals to us that this means \mathcal{D} cannot equal any of \mathcal{A} , \mathcal{B} , nor \mathcal{C} . In fact, \mathcal{D} can't be equal to any set containing any elements because that set would contain something \mathcal{D} doesn't.

Definition 3.4 (Empty Set).

empty We say that a set \mathcal{X} is *empty* iff $\forall x(x \notin \mathcal{X})$. We also define the following.

$$\emptyset := \{x \mid x \neq x\}$$

\emptyset The referent of the \emptyset symbol above is called *the empty set*. 定義

If we think of sets as *abstract containers*, it should be easy to conceptualize an empty container, which is exactly what \emptyset would correspond to. With such a suggestive name, we should be able to say that \emptyset is empty, right? Let's prove this as our first real theorem of set theory.

Theorem 3.1 (The Empty Set is Empty).

$\forall x(x \notin \emptyset)$. 定理

Proof. Let x be a set. Suppose, towards a contradiction, that $x \in \emptyset$. Then, we know $x \in \{z \mid z \neq z\}$ by definition of the empty set. This further tells us, by the definition of set comprehension notation, that $x \neq x$. However, we know $x = x$. Therefore, $x \notin \emptyset$. Q.E.D.

Based on how \mathcal{D} is drawn in *figure 3.5*, \mathcal{D} empty since $\forall x(x \notin \mathcal{D})$. Does that mean that $\mathcal{D} = \emptyset$, or is it possible to have multiple distinct empty sets? As you might have guessed by what the *axiom of extensionality* says, *there is only one empty set* because all empty sets are equal to each other, justifying the name *the empty set* for \emptyset .

Theorem 3.2 (The Empty Set is Unique).

$\forall x(\forall y(y \notin x) \Rightarrow x = \emptyset)$. 定理

Proof. Let x be a set such that $\forall y(y \notin x)$. We will show x has all the same elements as \emptyset . Let z be a set. We will show $z \in x \Leftrightarrow z \in \emptyset$.

If $z \in x$, notice $z \notin x$ follows from $\forall y(y \notin x)$. Thus, $z \in \emptyset$ by *explosion*. If $z \in \emptyset$, then $z \neq z$ by definition; but, $z = z$. So, $z \in x$ by *explosion*.

Recall *ex falso quodlibet*; anything follows from a contradiction.

Thus, $\forall z(z \in x \Leftrightarrow z \in \emptyset)$. So, $x = \emptyset$ by the *axiom of extensionality*.

Q.E.D.

It's important to note that none of the prior analyses nor theorems *prove* that \emptyset exists, only that there can be at most one empty set. We will need to wait until the *axiom of separation* to discuss this.

As you may have guessed, the empty set is the *smallest* in a precise sense. Given any two sets \mathcal{X} and \mathcal{Y} , we can define an *ordering* by saying that \mathcal{X} is "*less than*" \mathcal{Y} in when $\mathcal{X} \subseteq \mathcal{Y}$. With this notion of ordering induced by the \subseteq relation, we can see that the \emptyset is *ordered below* every other set, making it *minimal* in the \subseteq ordering among all sets.

Theorem 3.3.

$$\forall x(\emptyset \subseteq x).$$

定理

Proof. Let x be a set. Towards a contradiction, suppose $\emptyset \not\subseteq x$. Then, there exists some z such that $z \in \emptyset \wedge z \notin x$ by definition. This implies $z \in \emptyset$; however, we know $\forall w(w \notin \emptyset)$. \blacksquare Therefore, $\emptyset \subseteq x$. Q.E.D.

This is not the only nice property of the *set inclusion* ordering induced by the \subseteq relation. In fact, this relation has all the defining properties of a *partial order*: *reflexivity*, *antisymmetry*, and *transitivity*.

Theorem 3.4 (Set Inclusion is a Partial Order).

The following three statements hold about the \subseteq relation.

- | | |
|--|---------------------|
| 1. $\forall x(x \subseteq x)$ | <i>reflexivity</i> |
| 2. $\forall x \forall y(((x \subseteq y) \wedge (y \subseteq x)) \Rightarrow x = y)$. | <i>antisymmetry</i> |
| 3. $\forall x \forall y \forall z(((x \subseteq y) \wedge (z \subseteq x)) \Rightarrow x \subseteq z)$. | <i>transitivity</i> |

This makes \subseteq an example of a *partial order* on the class of sets. We prove the *reflexive* property below, leaving the rest as exercises. 定理

Proof. Let x be a set. Let z be a set and recall that $(z \in x) \Rightarrow (z \in x)$. Therefore, since z was arbitrary, we have $x \subseteq x$ by definition. Q.E.D.

Pairing

If you remember from our earlier discussion of set notation, we have a way of expressing “*the set containing a , b , and c* ” by writing down $\{a, b, c\}$. However, just having the ability to *say* something doesn’t make what we’re saying *meaningful*. If we want to be sure that $\{a, b, c\}$ actually refers to an object that *exists*, then we will either need *proof* that it exists, or we’ll need to rely on an *axiom* to grant us its existence. This next axiom partially addresses the problem with our set notation by guaranteeing that *set builder notation* always refers to an existing set so long as all of its elements also exist.

Axiom 2 (Pairing).

$$\forall x \forall y \exists z(z = \{x, y\}).$$

公理

By definition, we call a set a *singleton* if it contains exactly *one* element and a *doubleton* if it contains exactly *two* elements. The *pairing axiom* makes the straight-forward assertion that $\{x, y\}$ exists so long as x and y also exist—this set $\{x, y\}$ may be a *singleton* if $x = y$ or a *doubleton* if $x \neq y$. In other words, this axiom lets us construct *unordered pairs*.



Figure 3.6: Given the sets $\{0, 1, 7\}$ and $\{2, 3\}$ exist, the pairing axiom asserts the existence of $\{\{0, 1, 7\}, \{2, 3\}\}$.

Separation

We can similarly express “*the set of all things that make $\varphi(\cdot)$ true*” for any predicate φ by writing $\{x \mid \varphi(x)\}$, but again we have the same problem regarding the existence of the referent. If you think this obsession might be needlessly neurotic, let’s take a moment to see what happens if we pretend that $\{x \mid \varphi(x)\}$ exists for any predicate we feel like; after all, it should be natural to say that “*the set of all x with some property*” exists if a set is simply an abstract collection of objects.

Define the predicate $\beta(r) := r \notin r$. This is a perfectly sensible, unassuming predicate. Just as a sanity check, remind yourself about \mathcal{A} from figure 3.5 and notice that $\mathcal{A} \notin \mathcal{A}$ because \mathcal{A} is not 0, 1, nor 2. This means \mathcal{A} satisfies β , making $\beta(\mathcal{A})$ is *true*, with the point being that β is *true* about some real, tangible sets.

Now, let $\mathfrak{R} := \{x \mid \beta(x)\} = \{x \mid x \notin x\}$, and let’s try to see whether \mathfrak{R} makes $\beta(\cdot)$ *true* or *false*.¹ If $\beta(\mathfrak{R})$ is the case, then $\mathfrak{R} \notin \mathfrak{R}$ by the definition of β . That means that $\mathfrak{R} \in \{x \mid \beta(x)\}$; but then $\mathfrak{R} \in \mathfrak{R}$. ↗ Well, what happens if $\neg\beta(\mathfrak{R})$ is the case instead? Then, we must have $\neg(\mathfrak{R} \notin \mathfrak{R})$, so that $\mathfrak{R} \in \mathfrak{R}$, meaning $\mathfrak{R} \in \{x \mid \beta(x)\}$ by definition. However, if $\mathfrak{R} \in \{x \mid \beta(x)\}$, then \mathfrak{R} satisfies β , so that $\mathfrak{R} \notin \mathfrak{R}$. ↗

It seems like no matter what we do, we run into a problem because the *existence* of an object like \mathfrak{R} is *inherently contradictory*. We can not allow things like \mathfrak{R} into our universe of discourse or our logical system will explode. This observation—that the “*set*” of all sets that don’t contain themselves doesn’t exist—is known as *Russell’s paradox*, and it should point out to us that we can not allow *unrestricted* use of set comprehension notation. However, if we *restrict* ourselves to using comprehension only to filter elements from sets that already exist, we can completely avoid the issue of Russell’s paradox. By being careful to only use this *restricted* form of *comprehension*, we can enshrine the existence of sets whose elements are filtered—or *separated*—from another existing set with a predicate by introducing our next axiom.

Axiom 3 (Schema of Separation).

For any predicate φ with at most one free variable, we have

$$\forall x \exists y \left(y = \{z \mid z \in x \wedge \varphi(z)\} \right).$$

公理

Russell's paradox

¹ Remember that, since $\beta(\cdot)$ is a *predicate*, substituting its free variable with a term yields a *proposition* equivalent to \top or \perp .

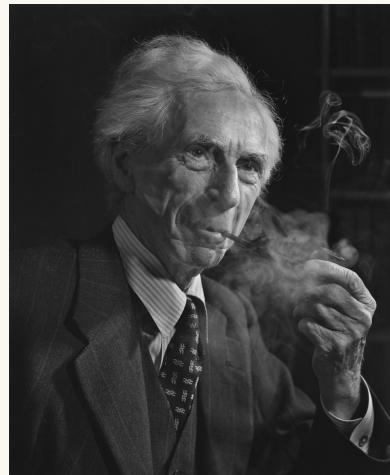


Figure 3.7: Russell’s paradox is named after eminent mathematician and philosopher **Bertrand Russell**. He first mentioned this paradox in a letter to logician and philosopher **Gottlob Frege** as a critique of his “*Basic Law V*,” which was essentially an *unrestricted* form of comprehension for logical functions.

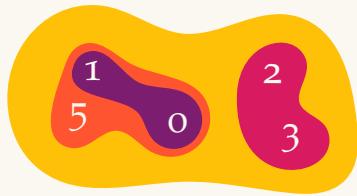


Figure 3.8: The *orange*, *red*, and *purple* sets are all subsets of the *yellow* set. We can see *purple* \subseteq *orange*, but *orange* $\not\subseteq$ *purple*. Further, *orange* $\not\subseteq$ *red*, and *red* $\not\subseteq$ *orange*, implying *purple* $\not\subseteq$ *red*, and *red* $\not\subseteq$ *purple*.

Index

- atomic, 10
- axiom, 21
- conjunction, 17
- contrapositive, 29
- disjunction, 17
- duality
 - logical, 17
- equivalence
 - logical, 22
 - material, 19
 - nonequivalence, 22
 - propositional, 14
- formula
 - propositional, 20
- logical
 - nonequivalence, 22
- material
 - equivalence, 19
 - implication, 18
 - negation, 16
- proof, 21
- proposition
 - formal, 19
- propositional
 - formula, 20
 - variable, 20
- quantifier
 - existential, 35
 - unique existential, 35
 - universal, 35
- sentence, 10
- theorem, 23