Unfortunately, the second step depends on knowing that humanity implies mortality for everybody, not just Socrates. If we are unlucky in our choice of axioms, we may not know this. What we would like is a general way to say that humanity implies mortality for everybody, but with just propositional logic, we can't write this fact down.

2.3.1 Variables and predicates

The solution is to extend our language to allow formulas that involve variables. So we might let x, y, z, etc. stand for any element of our **universe of discourse** or **domain**—essentially whatever things we happen to be talking about at the moment. We can now write statements like:

- "x is human."
- "x is the parent of y."
- "$x + 2 = x^2$."

These are not propositions because they have variables in them. Instead, they are **predicates**; statements whose truth-value depends on what concrete object takes the place of the variable. Predicates are often abbreviated by single capital letters followed by a list of **arguments**, the variables that appear in the predicate, e.g.:

- $H(x)$ = "x is human."
- $P(x, y)$ = "x is the parent of y."
- $Q(x)$ = "$x + 2 = x^2$."

We can also fill in specific values for the variables, e.g. $H(\text{Spocrates})$ = "Spocrates is human." If we fill in specific values for all the variables, we have a proposition again, and can talk about that proposition being true (e.g. $Q(2)$ and $Q(-1)$ are true) or false ($Q(0)$ is false).

In **first-order logic**, which is what we will be using in this course, variables always refer to things and never to predicates: any predicate symbol is effectively a constant. There are higher-order logics that allow variables to refer to predicates, but most mathematics accomplishes the same thing by representing predicates with sets (see Chapter 3).

2.3.2 Quantifiers

What we really want is to be able to say when H or P or Q is true for many different values of their arguments. This means we have to be able to talk about the truth or falsehood of statements that include variables. To do this, we **bind** the variables using **quantifiers**, which state whether the claim we are making applies to all values of the variable (**universal quantification**), or whether it may only apply to some (**existential quantification**).

2.3.2.1 Universal quantifier

The **universal quantifier** $\forall$ (pronounced "for all") says that a statement must be true for all values of a variable within some *universe* of allowed values (which is often implicit). For example, "all humans are mortal" could be written $\forall x : \text{Human}(x) \to \text{Mortal}(x)$ and "if x is positive then $x + 1$ is positive" could be written $\forall x : x > 0 \to x + 1 > 0$.

If you want to make the universe explicit, use set membership notation.[7] An example would be $\forall x \in \mathbb{Z} : x > 0 \to x + 1 > 0$. This is logically equivalent to writing $\forall x : x \in \mathbb{Z} \to (x > 0 \to x + 1 > 0)$ or to writing $\forall x : (x \in \mathbb{Z} \wedge x > 0) \to x + 1 > 0$, but the short form makes it more clear that the intent of $x \in \mathbb{Z}$ is to restrict the range of x.[8]

The statement $\forall x : P(x)$ is equivalent to a very large AND; for example, $\forall x \in \mathbb{N} : P(x)$ could be rewritten (if you had an infinite amount of paper) as $P(0) \wedge P(1) \wedge P(2) \wedge P(3) \wedge \ldots$. Normal first-order logic doesn't allow infinite expressions like this, but it may help in visualizing what $\forall x : P(x)$ actually means. Another way of thinking about it is to imagine that x is supplied by some adversary and you are responsible for showing that $P(x)$ is true; in this sense, the universal quantifier chooses the *worst case* value of x.

2.3.2.2 Existential quantifier

The **existential quantifier** $\exists$ (pronounced "there exists") says that a statement must be true for at least one value of the variable. So "some human is mortal" becomes $\exists x : \text{Human}(x) \wedge \text{Mortal}(x)$. Note that we use AND rather than implication here; the statement $\exists x : \text{Human}(x) \to \text{Mortal}(x)$ makes the much weaker claim that "there is some thing x, such that if x is human, then x is mortal," which is true in any universe that contains an immortal purple penguin—since it isn't human, $\text{Human}(\text{penguin}) \to \text{Mortal}(\text{penguin})$ is true.

[7] See Chapter 3.

[8] Programmers will recognize this as a form of *syntactic sugar*.

As with $\forall$, $\exists$ can be limited to an explicit universe with set membership notation, e.g., $\exists x \in Z : x = x^2$. This is equivalent to writing $\exists x : x \in Z \wedge x = x^2$.

The formula $\exists x : P(x)$ is equivalent to a very large OR, so that $\exists x \in \mathbb{N} : P(x)$ could be rewritten as $P(0) \vee P(1) \vee P(2) \vee P(3) \vee \ldots$. Again, you can't generally write an expression like this if there are infinitely many terms, but it gets the idea across.

2.3.2.3 Negation and quantifiers

The following equivalences hold:

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

These are essentially the quantifier version of De Morgan's laws: the first says that if you want to show that not all humans are mortal, it's equivalent to finding some human that is not mortal. The second says that to show that no human is mortal, you have to show that all humans are not mortal.

2.3.2.4 Restricting the scope of a quantifier

Sometimes we want to limit the universe over which we quantify to some restricted set, e.g., all positive integers or all baseball teams. We've previously seen how to do this using set-membership notation, but can also do this for more general predicates either explicitly using implication:

$$\forall x : x > 0 \rightarrow x - 1 \geq 0$$

or in abbreviated form by including the restriction in the quantifier expression itself:

$$\forall x > 0 : x - 1 \geq 0.$$

Similarly

$$\exists x : x > 0 \wedge x^2 = 81$$

can be written as

$$\exists x > 0 : x^2 = 81.$$

Note that constraints on $\exists$ get expressed using AND rather than implication.

The use of set membership notation to restrict a quantifier is a special case of this. Suppose we want to say that 79 is not a perfect square, by which

we mean that there is no integer whose square is 79. If we are otherwise talking about real numbers (two of which happen to be square roots of 79), we can exclude the numbers we don't want by writing

$$\neg\exists x \in \mathbb{Z} : x^2 = 79$$

which is interpreted as

$$\neg\exists x : (x \in \mathbb{Z} \wedge x^2 = 79)$$

or, equivalently

$$\forall x : x \in \mathbb{Z} \rightarrow x^2 \neq 79.$$

Here $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ is the standard set of integers.
For more uses of $\in$, see Chapter 3.

2.3.2.5 Nested quantifiers

It is possible to nest quantifiers, meaning that the statement bound by a quantifier itself contains quantifiers. For example, the statement "there is no largest prime number" could be written as

$$\neg\exists x : (\text{Prime}(x) \wedge \forall y : y > x \rightarrow \neg\text{Prime}(y))$$

i.e., "there does not exist an x that is prime and any y greater than x is not prime." Or in a shorter (though not strictly equivalent) form:

$$\forall x \exists y : y > x \wedge \text{Prime}(y)$$

which we can read as "for any x there is a bigger y that is prime."

To read a statement like this, treat it as a game between the $\forall$ player and the $\exists$ player. Because the $\forall$ comes first in this statement, the for-all player gets to pick any x it likes. The exists player then picks a y to make the rest of the statement true. The statement as a whole is true if the $\exists$ player always wins the game. So if you are trying to make a statement true, you should think of the universal quantifier as the enemy (the **adversary** in algorithm analysis) who says "nya-nya: try to make *this* work, bucko!", while the existential quantifier is the friend who supplies the one working response.

As in many two-player games, it makes a difference who goes first. If we write $\text{likes}(x, y)$ for the predicate that x likes y, the statements

$$\forall x \exists y : \text{likes}(x, y)$$

and

$$\exists y \forall x : \text{likes}(x, y)$$

mean very different things. The first says that for every person, there is somebody that that person likes: we live in a world with no complete misanthropes. The second says that there is some single person who is so immensely popular that everybody in the world likes them. The nesting of the quantifiers is what makes the difference: in $\forall x \exists y : \text{likes}(x, y)$, we are saying that no matter who we pick for x, $\exists y : \text{likes}(x, y)$ is a true statement; while in $\exists y \forall x : \text{likes}(x, y)$, we are saying that there is some y that makes $\forall x : \text{likes}(x, y)$ true.

Naturally, such games can go on for more than two steps, or allow the same player more than one move in a row. For example

$$\forall x \forall y \exists z : x^2 + y^2 = z^2$$

is a kind of two-person challenge version of the Pythagorean theorem where the universal player gets to pick x and y and the existential player has to respond with a winning z. (Whether the statement itself is true or false depends on the range of the quantifiers; it's false, for example, if x, y, and z are all natural numbers or rationals but true if they are all real or complex. Note that the universal player only needs to find one bad (x, y) pair to make it false.)

One thing to note about nested quantifiers is that we can switch the order of two universal quantifiers or two existential quantifiers, but we can't swap a universal quantifier for an existential quantifier or vice versa. So for example $\forall x \forall y : (x = y \rightarrow x + 1 = y + 1)$ is logically equivalent to $\forall y \forall x : (x = y \rightarrow y + 1 = x + 1)$, but $\forall x \exists y : y < x$ is not logically equivalent to $\exists y \forall x : y < x$. This is obvious if you think about it in terms of playing games: if I get to choose two things in a row, it doesn't really matter which order I choose them in, but if I choose something and then you respond it might make a big difference if we make you go first instead.

One measure of the complexity of a mathematical statement is how many layers of quantifiers it has, where a layer is a sequence of all-universal or all-existential quantifiers. Here's a standard mathematical definition that involves three layers of quantifiers, which is about the limit for most humans:

$$\left[\lim_{x \to \infty} f(x) = y\right] \equiv \left[\forall \epsilon > 0 : \exists N : \forall x > N : |f(x) - y| < \epsilon\right].$$

Now that we know how to read nested quantifiers, it's easy to see what the right-hand side means:

1. The adversary picks ϵ, which must be greater than 0.
2. We pick N.
3. The adversary picks x, which must be greater than N.
4. We win if $f(x)$ is within ϵ of y.

So, for example, a proof of

$$\lim_{x\to\infty} 1/x = 0$$

would follow exactly this game plan:

1. Choose some $\epsilon > 0$.
2. Let $N > 1/\epsilon$. (Note that we can make our choice depend on previous choices.)
3. Choose any $x > N$.
4. Then $x > N > 1/\epsilon > 0$, so $1/x < 1/N < \epsilon \rightarrow |1/x - 0| < \epsilon$. QED!

2.3.2.6 Examples

Here we give some more examples of translating English into statements in predicate logic.

All crows are black. $\forall x : \text{Crow}(x) \rightarrow \text{Black}(x)$

The formula is logically equivalent to either of

$$\neg\exists x \text{Crow}(x) \wedge \neg\text{Black}(x)$$

or

$$\forall x : \neg\text{Black}(x) \rightarrow \neg\text{Crow}(x).$$

The latter is the core of a classic "paradox of induction" in philosophy: if seeing a black crow makes me think it's more likely that all crows are black, shouldn't seeing a logically equivalent non-black non-crow (e.g., a banana yellow AMC Gremlin) also make me think all non-black objects are non-crows, i.e., that all crows are black? The paradox suggests that logical equivalence works best for true/false and not so well for probabilities.

Some cows are brown.	$\exists x : \text{Cow}(x) \wedge \text{Brown}(x)$
No cows are blue.	$\neg\exists x : \text{Cow}(x) \wedge \text{Blue}(x)$

Some other equivalent versions:

$$\forall x : \neg(\text{Cow}(x) \wedge \text{Blue}(x)$$
$$\forall x : (\neg\text{Cow}(x) \vee \neg\text{Blue}(x))$$
$$\forall x : \text{Cow}(x) \rightarrow \neg\text{Blue}(x)$$
$$\forall x : \text{Blue}(x) \rightarrow \neg\text{Cow}(x).$$

All that glitters is not gold.	$\neg\forall x : \text{Glitters}(x) \rightarrow \text{Gold}(x)$

Or $\exists x : \text{Glitters}(x) \wedge \neg\text{Gold}(x)$. Note that the English syntax is a bit ambiguous: a literal translation might look like $\forall x : \text{Glitters}(x) \rightarrow \neg\text{Gold}(x)$, which is *not* logically equivalent. This is an example of how predicate logic is often more precise than natural language.

No shirt, no service.	$\forall x : \neg\text{Shirt}(x) \rightarrow \neg\text{Served}(x)$
Every event has a cause.	$\forall x \exists y : \text{Causes}(y, x)$

And a more complicated statement: Every even number greater than 2 can be expressed as the sum of two primes.

$$\forall x : (\text{Even}(x) \wedge x > 2) \rightarrow (\exists p \exists q : \text{Prime}(p) \wedge \text{Prime}(q) \wedge (x = p + q))$$

The last one is **Goldbach's conjecture**. The truth value of this statement is currently unknown.

2.3.3 Functions

A **function symbol** looks like a predicate but instead of computing a truth value it returns an object. Function symbols may take zero or more arguments. The special case of a function symbol with zero arguments is called a **constant**.

For example, in the expression $2 + 2 = 5$, we've got three constants 2, 2, and 5, a two-argument function $+$, and a predicate $=$, which has a special role in predicate logic that we'll discuss in more detail below.

The nice thing about function symbols is that they let us populate our universe without having to include a lot of axioms about various things

existing. The convention is that anything we can name exists. An example is the construction of the natural numbers $0, 1, 2, \ldots$ used with the Peano axioms: these are represented using the constant 0 and the **successor function** S, so that we can count $0, S0, SS0, SSS0$, and so on.

Note however that there is no guarantee that two objects constructed in different ways are actually distinct ($2 + 2 = 4$ after all). To express whether objects are the same as each other or not requires a dedicated equality predicate, discussed below.

2.3.4 Equality

The **equality** predicate $=$, written $x = y$, is typically included as a standard part of predicate logic. The interpretation of $x = y$ is that x and y are the same element of the domain. Equality satisfies the **reflexivity axiom** $\forall x : x = x$ and the **substitution axiom schema**:

$$\forall x \forall y : (x = y \to (Px \leftrightarrow Py))$$

where P is any predicate. This immediately gives a **substitution rule** that says $x = y, P(x) \vdash P(y)$. It's likely that almost every proof you ever wrote down in high school algebra consisted only of many applications of the substitution rule.

Example: We'll prove $\forall x \forall y : (x = y \to y = x)$ from the above axioms (this property is known as **symmetry**). Apply substitution to the predicate $Pz \equiv z = x$ to get $\forall x \forall y : (x = y \to (x = x \leftrightarrow y = x))$. Use reflexivity to rewrite this as $\forall x \forall y : (x = y \to (1 \leftrightarrow y = x))$, which simplifies to $\forall x \forall y : (x = y \to y = x)$.

Exercise: Prove $\forall x \forall y \forall z : (x = y \wedge y = z \to x = z)$. (This property is known as **transitivity**.)

2.3.4.1 Uniqueness

The abbreviation $\exists! x P(x)$ says "there exists a *unique* x such that $P(x)$." This is short for

$$\exists x (P(x) \wedge (\forall y : P(y) \to x = y)),$$

which we can read as "there is an x for which $P(x)$ is true, and any y for which $P(y)$ is true is equal to x."

An example is $\exists! x : x + 1 = 12$. To prove this we'd have to show not only that there is some x for which $x + 1 = 12$ (11 comes to mind), but that if we have any two values x and y such that $x + 1 = 12$ and $y + 1 = 12$, then

$x = y$ (this is not hard to do, assuming we have at our disposal the usual axioms of arithmetic). So the exclamation point encodes quite a bit of extra work, which is why we often hope that $\exists x : x + 1 = 12$ is good enough and pull out $\exists!$ only if we have to.

There are several equivalent ways to expand $\exists! x P(x)$. Applying contraposition to $P(y) \to x = y$ gives

$$\exists! x P(x) \equiv \exists x (P(x) \wedge (\forall y : x \neq y \to \neg P(y))),$$

which says that any y that is not x doesn't satisfy P. We can also play some games with De Morgan's laws to turn this into

$$\exists! x P(x) \equiv \exists x (P(x) \wedge (\neg \exists y : x \neq y \wedge P(y))).$$

This says that there is an x with $P(x)$, but there is no $y \neq x$ with $P(y)$. All of these are just different ways of saying that x is the only object that satisfies P.

2.3.5 Models

In propositional logic, we can build truth tables that describe all possible settings of the truth-values of the literals. In predicate logic, the analogous concept to an assignment of truth-values is a **structure**. A structure consists of a set of objects or elements (built using set theory, as described in Chapter 3), together with a description of which elements fill in for the constant symbols, which predicates hold for which elements, and what the value of each function symbol is when applied to each possible list of arguments (note that this depends on knowing what constant, predicate, and function symbols are available—this information is called the **signature** of the structure). A structure is a **model** of a particular **theory** (set of statements), if each statement in the theory is true in the model.

In general we can't hope to find all possible models of a given theory. But models are useful for two purposes: if we can find some model of a particular theory, then the existence of this model demonstrates that the theory is consistent; and if we can find a model of the theory in which some additional statement S doesn't hold, then we can demonstrate that there is no way to prove S from the theory (i.e. it is not the case that $T \vdash S$, where T is the list of axioms that define the theory).

2.3.5.1 Examples

- Consider the axiom $\neg \exists x$. This axiom has exactly one model (it's empty).

- Now consider the axiom $\exists!x$, which we can expand out to $\exists x \forall y y = x$. This axiom also has exactly one model (with one element).

- We can enforce exactly k elements with one rather long axiom, e.g. for $k = 3$ do $\exists x_1 \exists x_2 \exists x_3 \forall y : y = x_1 \vee y = x_2 \vee y = x_3 \wedge x_1 \neq x_2 \wedge x_2 \neq x_3 \wedge x_3 \neq x_1$. In the absence of any special symbols, a structure of 3 undifferentiated elements is the unique model of this axiom.

- Suppose we add a predicate P and consider the axiom $\exists x Px$. Now we have many models: take any nonempty model you like, and let P be true of at least one of its elements. If we take a model with two elements a and b, with Pa and $\neg Pb$, we see that $\exists x Px$ is not enough to prove $\forall x Px$, since $\exists x Px$ is true in the model but $\forall x Px$ isn't. Conversely, an empty model satisfies $\forall x Px \equiv \neg \exists x \neg Px$ but not $\exists x Px$.

- Now let's bring in a function symbol S and constant symbol 0. Consider a stripped-down version of the Peano axioms that consists of just the axiom $\forall x \forall y : Sx = Sy \rightarrow x = y$. Both the natural numbers $\mathbb{N}$ and the integers $\mathbb{Z}$ are a model for this axiom, as is the set $\mathbb{Z}_m$ of integers mod m for any m (see §8.3). In each case each element has a unique predecessor, which is what the axiom demands. If we throw in the first Peano axiom $\forall x : Sx \neq 0$, we eliminate $\mathbb{Z}$ and $\mathbb{Z}_m$ because in each of these models 0 is a successor of some element. But we don't eliminate a model that consists of two copies of $\mathbb{N}$ sitting next to each other (only one of which contains the "official" 0), or even a model that consists of one copy of $\mathbb{N}$ (that includes the official 0 with no predecessor) plus any number of copies of $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Z}_m$.

- A practical example: The family tree of the kings of France is a model of the theory containing the two axioms $\forall x \forall y \forall z \text{Parent}(x, y) \wedge \text{Parent}(y, z) \rightarrow \text{GrandParent}(x, z)$ and $\forall x \forall y \text{Parent}(x, y) \rightarrow \neg \text{Parent}(y, x)$. But this set of axioms could use some work, since it still allows for the possibility that there are some x and y for which $\text{Parent}(x, y)$ and $\text{GrandParent}(y, x)$ are both true.

2.4 Proofs

A proof is a way to derive statements from other statements. It starts with **axioms** (statements that are assumed in the current context always to be true), **theorems** or **lemmas** (statements that were proved already; the difference between a theorem and a lemma is whether it is intended as a final

result or an intermediate tool), and **premises** P (assumptions we are making for the purpose of seeing what consequences they have), and uses **inference rules** to derive Q. The axioms, theorems, and premises are in a sense the starting position of a game whose rules are given by the inference rules. The goal of the game is to apply the inference rules until Q pops out. We refer to anything that isn't proved in the proof itself (i.e., an axiom, theorem, lemma, or premise) as a **hypothesis**; the result Q is the **conclusion**.

When a proof exists of Q from some premises $P_1, P_2, \ldots$, we say that Q is **deducible** or **provable** from $P_1, P_2, \ldots$, which is written as

$$P_1, P_2, \ldots \vdash Q.$$

If we can prove Q directly from our inference rules without making any assumptions, we may write

$$\vdash Q$$

The **turnstile** symbol $\vdash$ has the specific meaning that we can derive the conclusion Q by applying inference rules to the premises. This is not quite the same thing as saying $P \to Q$. If our inference rules are particularly weak, it may be that $P \to Q$ is true but we can't prove Q starting with P. Conversely, if our inference rules are too strong (maybe they can prove anything, even things that aren't true) we might have $P \vdash Q$ but $P \to Q$ is false.

For propositions, most of the time we will use inference rules that are just right, meaning that $P \vdash Q$ implies that $P \to Q$ is a tautology, (**soundness**) and $P \to Q$ being a tautology implies that $P \vdash Q$ (**completeness**). Here the distinction between $\vdash$ and $\to$ is whether we want to talk about the existence of a proof (the first case) or about the logical relation between two statements (the second).

Things get a little more complicated with statements involving predicates. For predicate logic, there are **incompleteness theorems** that say that if our system of axioms is powerful enough (basically capable of representing arithmetic), then there are are statements P such that neither of P or $\neg P$ are provable unless the theory is inconsistent.

2.4.1 Inference Rules

Inference rules let us construct **valid** arguments, which have the useful property that if their premises are true, their conclusions are also true.

The main source of inference rules is tautologies of the form $P_1 \wedge P_2 \ldots \to Q$; given such a tautology, there is a corresponding inference rule that allows

us to assert Q once we have $P_1, P_2, \ldots$. Given an inference rule of this form and a goal Q, we can then look for ways to show $P_1, P_2, \ldots$ all hold, either because each P_i is an axiom/theorem/premise or because we can prove it from other axioms, theorems, or premises.

The most important inference rule is **modus ponens**, based on the tautology $(p \wedge (p \to q)) \to q$; this lets us, for example, write the following famous argument:[9]

1. If it doesn't fit, you must acquit. [Axiom]
2. It doesn't fit. [Premise]
3. You must acquit. [Modus ponens applied to 1+2]

There are many named inference rules in classical propositional logic. We'll list some of them below. You don't need to remember the names of anything except modus ponens, and most of the rules are pretty much straightforward applications of modus ponens plus some convenient tautology that can be proved by truth tables or stock logical equivalences. (For example, the "addition" rule below is just the result of applying modus ponens to p and the tautology $p \to (p \vee q)$.)

Inference rules are often written by putting the premises above a horizontal line and the conclusion below. In text, the horizontal line is often replaced by the symbol $\vdash$, which means exactly the same thing. Premises are listed on the left-hand side separated by commas, and the conclusion is placed on the right. We can then write

$p \vdash p \vee q.$	Addition
$p \wedge q \vdash p.$	Simplification
$p, q \vdash p \wedge q.$	Conjunction
$p, p \to q \vdash q.$	Modus ponens
$\neg q, p \to q \vdash \neg p.$	Modus tollens
$p \to q, q \to r \vdash p \to r.$	Hypothetical syllogism
$p \vee q, \neg p \vdash q.$	Disjunctive syllogism
$p \vee q, \neg p \vee r \vdash q \vee r.$	Resolution

Of these rules, addition, simplification, and conjunction are mostly used to pack and unpack pieces of arguments. Modus ponens "the method of affirming" (and its reversed cousin modus tollens "the method of denying")

[9] Maybe not as famous as it once was.

let us apply implications. You don't need to remember modus tollens if you can remember the contraposition rule $(p \to q) \equiv (\neg q \to \neg p)$. Hypothetical syllogism just says that implication is transitive; it lets you paste together implications if the conclusion of one matches the premise of the other. Disjunctive syllogism is again a disguised version of modus ponens (via the logical equivalence $(p \vee q) \equiv (\neg p \to q)$); you don't need to remember it if you can remember this equivalence. Resolution is almost never used by humans but is very popular with computer theorem provers.

An argument is **valid** if the conclusion is true whenever the hypotheses are true. Any proof constructed using the inference rules is valid. It does not necessarily follow that the conclusion is true; it could be that one or more of the hypotheses is false:

1. If you give a mouse a cookie, he's going to ask for a glass of milk. [Axiom]
2. If he asks for a glass of milk, he will want a straw. [Axiom]
3. You gave a mouse a cookie. [Premise]
4. He asks for a glass of milk. [Modus ponens applied to 1 and 3.]
5. He will want a straw. [Modus ponens applied to 2 and 4.]

Will the mouse want a straw? No: Mice can't ask for glasses of milk, so Axiom 1 is false.

2.4.2 Proofs, implication, and natural deduction

Recall that $P \vdash Q$ means there is a proof of Q by applying inference rules to P, while $P \to Q$ says that Q holds whenever P does. These are not the same thing: provability ($\vdash$) is outside the theory (it's a statement about whether a proof exists or not) while implication ($\to$) is inside (it's a logical connective for making compound propositions). But most of the time they mean almost the same thing.

For example, suppose that $P \to Q$ is provable without any assumptions:

$$\vdash P \to Q.$$

Since we can always ignore extra premises, we get

$$P \vdash P \to Q$$

and thus

$$P \vdash P, P \to Q,$$

which gives

$$P \vdash Q$$

by applying modus ponens to the right-hand side.

So we can go from $\vdash P \to Q$ to $P \vdash Q$.

This means that provability is in a sense weaker than implication: it holds (assuming modus ponens) whenever implication does. But we usually don't use this fact much, since $P \to Q$ is a much more useful statement than $P \vdash Q$. Can we go the other way?

2.4.2.1 The Deduction Theorem

Yes, using the **Deduction Theorem**.

Often we want to package the result of a proof as a **theorem** (a proven statement that is an end in itself) or **lemma** (a proven statement that is intended mostly to be used in other proofs). Typically a proof shows that, given some base assumptions Γ, if certain premises $P_1, P_2, \ldots P_n$ hold, then some conclusion Q holds (with various axioms or previously-established theorems assumed to be true from context). To use this result later, it is useful to be able to package it as an implication $P_1 \wedge P_2 \wedge \ldots P_n \to Q$. In other words, we want to go from

$$\Gamma, P_1, P_2, \ldots, P_n \vdash Q$$

to

$$\Gamma \vdash (P_1 \wedge P_2 \wedge \ldots \wedge P_n) \to Q.$$

The statement that we can do this, for a given collection of inference rules, is the Deduction Theorem:

Theorem 2.4.1 (Deduction Theorem)**.** *If there is a proof of Q from premises $\Gamma, P_1, P_2, \ldots, P_n$, then there is a proof of $P_1 \wedge P_2 \wedge \ldots \wedge P_n \to Q$ from Γ alone.*

The actual proof of the theorem depends on the particular set of inference rules we start with, but the basic idea is that there exists a mechanical procedure for extracting a proof of the implication from the proof of Q assuming P_1 etc.

Caveat: In predicate logic, the deduction theorem only applies if none of the premises contain any free variables (which are variables that aren't bound by a universal or existential quantifier). Usually you won't run into this, but there are some bad cases that arise without this restriction.

2.4.2.2 Natural deduction

In practice, we usually don't refer to the Deduction Theorem directly, and instead adopt a new inference rule:

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \to Q} \quad (\to I) \tag{}$$

This says that if we can prove Q using assumptions Γ and P, then we can prove $P \to Q$ using just Γ. Note that the horizontal line acts like a higher-order version of $\vdash$; it lets us combine one or more proofs into a new, bigger proof.

This style of inference rule, where we explicitly track what assumptions go into a particular result, is known as **natural deduction**. The natural deduction approach was invented by Gentzen [Gen35a, Gen35b] as a way to make inference rules more closely match actual mathematical proof-writing practice than the modus-ponens-only approach that modern logicians had been using up to that point.[10]

The particular rule ($\to I$) is called introducing implication. There is a corresponding rule for eliminating implication that is essentially just modus ponens:

$$\frac{\Gamma \vdash P \to Q \quad \Gamma \vdash P}{\Gamma \vdash Q} \quad (\to E) \tag{}$$

If we want to be really systematic about things, we can rewrite most of our standard inference rules as introduction and elimination rules for particular operators. This can make them a bit easier to remember, since for each Boolean operator there is often an "obvious" introduction and elimination rule for it. See Table 2.4 for a list.

2.4.3 Inference rules for equality

The equality predicate is special, in that it allows for the **substitution rule**

$$x = y, P(x) \vdash P(y).$$

[10] See http://plato.stanford.edu/entries/proof-theory-development/ for a more detailed history of the development of proof theory in general and [Pel99] for a discussion of how different versions of proof theory have been adopted in textbooks.

$$\frac{\Gamma \vdash P}{\Gamma \vdash \neg\neg P} \quad (\neg I)$$

$$\frac{\Gamma \vdash \neg\neg P}{\Gamma \vdash P} \quad (\neg E)$$

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \quad (\wedge I)$$

$$\frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash P} \quad (\wedge E_1)$$

$$\frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash Q} \quad (\wedge E_2)$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \quad (\vee I_1)$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \quad (\vee I_2)$$

$$\frac{\Gamma \vdash P \vee Q \quad \Gamma \vdash \neg Q}{\Gamma \vdash P} \quad (\vee E_1)$$

$$\frac{\Gamma \vdash P \vee Q \quad \Gamma \vdash \neg P}{\Gamma \vdash Q} \quad (\vee E_2)$$

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \to Q} \quad (\to I)$$

$$\frac{\Gamma \vdash P \to Q \quad \Gamma \vdash P}{\Gamma \vdash Q} \quad (\to E_1)$$

$$\frac{\Gamma \vdash P \to Q \quad \Gamma \vdash \neg Q}{\Gamma \vdash \neg P} \quad (\to E_2)$$

Table 2.4: Natural deduction: introduction and elimination rules

If we don't want to include the substitution rule as an inference rule, we could instead represent it as an axiom schema:

$$\forall x : \forall y : ((x = y \wedge P(x)) \rightarrow P(y)).$$

But this is messier.

We can also assert $x = x$ directly:

$$\vdash x = x$$

2.4.4 Inference rules for quantified statements

Universal generalization If y is a variable that does not appear in Γ, then

$$\frac{\Gamma \vdash P(y)}{\Gamma \vdash \forall x : P(x)}$$

This says that if we can prove that some property holds for a "generic" y, without using any particular properties of y, then in fact the property holds for all possible x.

In a written proof, this will usually be signaled by starting with something like "Let y be an arbitrary [member of some universe]". For example: Suppose we want to show that there is no biggest natural number, i.e. that $\forall n \in \mathbb{N} : \exists n' \in \mathbb{N} : n' > n$. Proof: Let n be any element of $\mathbb{N}$. Let $n' = n + 1$. Then $n' > n$. (Note: there is also an instance of existential generalization here.)

Universal instantiation In the other direction, we have

$$\forall x : Q(x) \vdash Q(c).$$

Here we go from a general statement about all possible values x to a statement about a particular value. Typical use: Given that all humans are mortal, it follows that Spocrates is mortal.

Existential generalization This is essentially the reverse of universal instantiation: it says that, if c is some particular object, we get

$$Q(c) \vdash \exists x : Q(x).$$

The idea is that to show that $Q(x)$ holds for at least one x, we can point to c as a specific example of an object for which Q holds. The

corresponding style of proof is called a **proof by construction** or **proof by example**.

For example: We are asked to prove that there exists an even prime number. Look at 2: it's an even prime number. QED.

Not all proofs of existential statements are **constructive**, in the sense of identifying a single object that makes the existential statement true. An example is a well-known **non-constructive** proof that there are irrational numbers a and b for which a^b is rational. The non-constructive proof is to consider $\sqrt{2}^{\sqrt{2}}$. If this number is rational, it's an example of the claim; if not, $\left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \sqrt{2}^2 = 2$ works.[11] Non-constructive proofs are generally not as useful as constructive proofs, because the example used in a constructive proof may have additional useful properties in other contexts.

Existential instantiation $\exists x : Q(x) \vdash Q(c)$ for some c, where c is a new name that hasn't previously been used (this is similar to the requirement for universal generalization, except now the new name is on the right-hand side).

The idea here is that we are going to give a name to some c that satisfies $Q(c)$, and we know that we can get away this because $\exists x : Q(x)$ says that some such thing exists.[12]

In a proof, this is usually signaled by "let x be..." or "call it x." For example: Suppose we know that there exists a prime number greater than 7. Let p be some such prime number greater than 7.

In natural-deduction terms, we can think of these rules as introduction and elimination rules for $\forall$ and E. Table 2.5 shows what these look like.

2.5 Proof techniques

A proof technique is a template for how to go about proving particular classes of statements: this template guides you in the choice of inference

[11] For this particular claim, there is also a constructive proof: $\sqrt{2}^{\log_2 9} = 3$ [Sch01].

[12] This is actually a fairly painful idea to formalize. One version in pure first-order logic is the axiom

$$((\forall x : (Q(x) \to P)) \wedge \exists y : Q(y)) \to P.$$

Nobody but a logician would worry about this.

$$\frac{\Gamma \vdash Pc}{\Gamma \vdash \forall x : Px} \quad (\forall I)$$

$$\frac{\Gamma \vdash \forall x : Px}{\Gamma \vdash Pc} \quad (\forall E)$$

$$\frac{\Gamma \vdash Pc}{\Gamma \vdash \exists x : Px} \quad (\exists I)$$

$$\frac{\Gamma \vdash \exists x : Px}{\Gamma \vdash Pc} \quad (\exists E)$$

Table 2.5: Natural deduction: introduction and elimination rules for quantifiers. For $\forall I$ and $\exists E$, c is a new symbol that does not appear in P or Γ.

rules (or other proof techniques) to write the actual proof. This doesn't substitute entirely for creativity (there is no efficient mechanical procedure for generating even short proofs unless $\mathbf{P} = \mathbf{NP}$), but it can give you some hints for how to get started.

Table 2.6 gives techniques for trying to prove $A \to B$ for particular statements A and B. The techniques are mostly classified by the structure of B. Before applying each technique, it may help to expand any definitions that appear in A or B.

These strategies are largely drawn from [Sol05], particularly the summary table in the appendix, which is the source of the order and organization of the table and the names of most of the techniques. The table omits some techniques that are mentioned in Solow [Sol05]: Direct Uniqueness, Indirect Uniqueness, and various max/min arguments. The remaining techniques mostly follow directly from the inference rules from the preceding section; an exception is induction, which will be discussed in Chapter 5.

For other sources, Ferland [Fer08] has an entire chapter on proof techniques of various sorts. Rosen [Ros12] describes proof strategies in §§1.5–1.7 and Biggs [Big02] describes various proof techniques in Chapters 1, 3, and 4; both descriptions are a bit less systematic than the ones in Solow or Ferland, but also include a variety of specific techniques that are worth looking at.

If you want to prove $A \leftrightarrow B$, the usual approach is to prove $A \to B$ and $A \leftarrow B$ separately. Proving $A \to B$ and $\neg A \to \neg B$ also works (because of contraposition).

Strategy	When	Assume	Conclude	What to do/why it works
Direct proof	Try it first	A	B	Apply inference rules to work forward from A and backward from B; when you meet in the middle, pretend that you were working forward from A all along.
Contraposition	$B = \neg Q$	$\neg B$	$\neg A$	Apply any other technique to show $\neg B \to \neg A$ and then apply the contraposition rule. Sometimes called an *indirect proof* although the term *indirect proof* is often used instead for proofs by contradiction (see below).
Contradiction	When $B = \neg Q$, or when you are stuck trying the other techniques.	$A \wedge \neg B$	False	Apply previous methods to prove both P and $\neg P$ for some P. Note: this can be a little dangerous, because you are assuming something that is (probably) not true, and it can be hard to detect as you prove further false statements whether the reason they are false is that they follow from your false assumption, or because you made a mistake. Direct or contraposition proofs are preferred because they don't have this problem.

Construction	$B = \exists x P(x)$	A	$P(c)$ for some specific object c.	Pick a likely-looking c and prove that $P(c)$ holds.
Counterexample	$B = \neg\forall x P(x)$	A	$\neg P(c)$ for some specific object c.	Pick a likely-looking c and show that $\neg P(c)$ holds. This is identical to a proof by construction, except that we are proving $\exists x \neg P(x)$, which is equivalent to $\neg\forall x P(x)$.
Choose	$B = \forall x(P(x) \to Q(x))$	A, $P(c)$, where c is chosen arbitrarily.	$Q(c)$	Choose some c and assume A and $P(c)$. Prove $Q(c)$. Note: c is a placeholder here. If $P(c)$ is "c is even" you can write "Let c be even" but you can't write "Let $c = 12$", since in the latter case you are assuming extra facts about c.
Instantiation	$A = \forall x P(x)$	A	B	Pick some particular c and prove that $P(c)$ →B. Here you *can* get away with saying "Let $c = 12$." (If $c = 12$ makes B true).
Elimination	$B = C \vee D$	$A \wedge \neg C$	D	The reason this works is that $A \wedge \neg C \to D$ is equivalent to $\neg(A \wedge \neg C) \to D \equiv \neg A \vee C \vee D \equiv A \to (C \vee D)$. Of course, it works equally well if you start with $A \wedge \neg D$ and prove C.

Case analysis	$A = C \vee D$	C, D	B	Here you write two separate proofs: one that assumes C and proves B, and one that assumes D and proves B. A special case is when $D = \neg C$. You can also consider more cases, as long as A implies at least one of the cases holds.
Induction	$B = \forall x \in \mathbb{N} P(x)$	A	$P(0)$ and $\forall x \in \mathbb{N} : (P(x) \rightarrow P(x + 1))$.	If $P(0)$ holds, and $P(x)$ implies $P(x + 1)$ for all x, then for any specific natural number n we can consider constructing a sequence of proofs $P(0) \rightarrow P(1) \rightarrow P(2) \rightarrow \ldots \rightarrow P(n)$. (This is actually a defining property of the natural numbers.)

Table 2.6: Proof techniques (adapted from [Sol05]

2.6 Examples of proofs

Real proofs by actual human mathematicians are usually written in a condensed style that uses ordinary language, without trying to convert everything into logical notation. But in principle it should be possible to translate any such proof into a formal proof. In this section, we give some examples of what a condensed proof might look like, and explain how the steps used in such proofs correspond to inference rules we've already seen.

2.6.1 Axioms for even numbers

Let's define what it means for a number to be even, where we use the Peano-axiom convention for writing numbers as $0, S0, SS0$, etc. We will use the

following axioms for our definition, where Ex means that x is even:

$$A_1 : \forall x : Ex \leftrightarrow (x = 0 \vee (\exists y : Ey \wedge x = SSy))$$
$$A_2 : \forall x : 0 \neq Sx.$$
$$A_3 : \forall x \forall y : Sx = Sy \rightarrow x = y.$$

Here A_1 is the definition of Ex and A_2 and A_3 are general axioms about S that we are throwing in because we will need them in some of our proofs.

2.6.2 A theorem and its proof

Now let's prove this exciting theorem:

Theorem 2.6.1. *All of the following statements are true:*

1. $E0$.
2. $\neg E(S0)$.
3. $E(SS0)$.
4. $\neg E(SSS0)$.
5. $E(SSSS0)$.

Proof. 1. Axiom A_1 says that x is even if it is 0.

2. Suppose $E(S0)$ holds. Then either $S0 = 0$ or $S0 = SSy$ for some y such that Ey holds. The first case contradicts A_2; in the second case, applying A_3 gives that $S0 = SSy$ implies $0 = Sy$, which again contradicts A_2. So in either case we arrive at a contradiction, and our original assumption that $E(S0)$ is true does not hold.

   (This is an example of an indirect proof.)

3. From A_1 we have that $E(SS0)$ holds if there exists some y such that Ey and $SS0 = SSy$. Let $y = 0$.

4. We have previously established $\neg E(S0)$. We also know that $SSS0 \neq 0$, so $E(SSS0)$ is true if and only if $SSS0 = SSy$ for some y with Ey. Applying A_3 twice gives $SSS0 = SSy$ iff $S0 = y$. But we already showed $\neg E(S0)$, so $\neg E(SSS0)$.

5. Since $E(SS0)$ and $SSSS0 = SS(SS0)$, $E(SSSS0)$.

□

The nice thing about proving all of these facts at once is that as we prove each one we can use that fact to prove the later ones. From a purely stylistic point of view, we can also assume that the reader is probably starting to catch on to some of the techniques we are using, which is why the argument for $E(SSSS0)$ is so succinct compared to the argument for $E(SS0)$.

If we had to expand these arguments out using explicit inference rules, they would take longer, but we could do it. Let's try this for the proof of $\neg E(S0)$. We are trying to establish that $A_1, A_2, A_3 \vdash \neg E(S0)$. Abbreviating A_1, A_2, A_3 as Γ, the strategy is to show that $\Gamma \vdash E(S0) \rightarrow Q$ for some Q with $\Gamma \vdash \neg Q$; we can then apply the $\rightarrow E_2$ rule (aka modus tollens) to get $\Gamma \vdash \neg E(S0)$.

Formally, this looks like:

1. $\Gamma \vdash E(S0) \leftrightarrow (S0 = 0 \vee \exists y : (Ey \wedge S0 = SSy))$. ($\forall E$ applied to A_1.)
2. $\Gamma \vdash E(S0) \rightarrow (S0 = 0 \vee \exists y : (Ey \wedge S0 = SSy))$. (Expand $\leftrightarrow$ and use one of the $\wedge$ elimination rules.)
3. $\Gamma, E(S0) \vdash S0 = 0 \vee \exists y : (Ey \wedge S0 = SSy)$. ($\rightarrow E$).
4. $\Gamma, E(S0) \vdash \neg(S0 = 0)$. (Apply $\forall E$ to A_2.)
5. $\Gamma, E(S0) \vdash \exists y : (Ey \wedge S0 = SSy)$. (Combine last two steps using $\vee E_1$.)
6. $\Gamma, E(S0) \vdash Ez \wedge S0 = SSz$. (This is $\exists E$. In the condensed proof we didn't rename y, but calling it z here makes it a little more obvious that we are fixing some particular constant.)
7. $\Gamma, E(S0) \vdash S0 = SSz$. ($\wedge E_1$.)
8. $\Gamma, E(S0) \vdash S0 = SSz \leftrightarrow 0 = Sz$. (Apply $\forall E$ to A_3).
9. $\Gamma, E(S0) \vdash S0 = SSz \rightarrow 0 = Sz$. (Another expansion plus $\wedge E$).
10. $\Gamma, E(S0) \vdash 0 = Sz$. (Apply $\rightarrow E_1$ to $S0 = SSz$ and $S0 = SSz \rightarrow 0 = Sz$.)
11. $\Gamma \vdash E(S0) \rightarrow 0 = Sz$. ($\rightarrow I$.)
12. $\Gamma \vdash \neg(0 = Sz)$. ($\forall E$ and A_2.)
13. $\Gamma \vdash \neg E(S0)$. ($\rightarrow E_2$.)

One thing to notice about the formal argument is how $E(S0)$ moves in and and out of the left-hand side of the turnstile in the middle of the proof. This is a pretty common trick, and is what is going on whenever you read a proof that says something like "suppose P holds" or "consider the case where P holds." Being able to just carry P (in this case, $E(S0)$) around as an assumption saves a lot of writing "if P" over and over again, and more formally is what allows us to unpack $P \to Q$ and apply inference rules to Q.

2.6.3 A more general theorem

So far we have only proved results about a few specific numbers. Can we say anything about all numbers? Let's try to prove the following theorem:

Theorem 2.6.2. *For all x, if x is even, $SSSSx$ is even.*

Proof. Let x be even. Then SSx is even (Axiom A_1), and so $SS(SSx) = SSSSx$ is also even. □

Written out using natural-deduction inference rules (with some of the more boring steps omitted), the proof would look like this:

1. $\Gamma, Ex \vdash (\exists y : Ey \wedge SSx = SSy) \to E(SSx)$. (Axiom A_1, $\forall E$, $\vee E_1$.)
2. $\Gamma, Ex \vdash Ex$.
3. $\Gamma, Ex \vdash SSx = SSx$. (Reflexivity of $=$.)
4. $\Gamma, Ex \vdash Ex \wedge SSx = SSx$. ($\wedge I$ applied to previous two steps.)
5. $\Gamma, Ex \vdash \exists y : Ey \wedge SSy = SSx$. (Let $y = x$.)
6. $\Gamma, Ex \vdash E(SSx)$. (Modus ponens!)
7. $\Gamma, Ex \vdash E(SSSSx)$. (Do it all again to show $E(SSx) \to E(SSSSx)$. This is the boring part we promised to omit.)
8. $\Gamma \vdash Ex \to E(SSSSx)$. ($\to I$.)
9. $\Gamma \vdash \forall x : Ex \to E(SSSSx)$. ($\forall I$).

If we had to write all the boring parts out, it might make sense to first prove a lemma $\forall x : Ex \to E(SSx)$ and then just apply the lemma twice.

The instruction "let x be even" is doing a lot of work in the condensed proof: it is introducing both a new name x that we will use for the Universal Generalization rule $\forall E$, and the assumption that x is even that we will use

for the Deduction Theorem $\rightarrow E$. Note that we can't apply $\forall E$ until we've moved the assumption Ex out of the left-hand side of the turnstile, because Universal Generalization only works if x is not a name mentioned in the assumptions.

2.6.4 Something we can't prove

One thing we probably know about the natural numbers is that if x is even, then $x + 1$ is odd, and vice versa. As a theorem this would look like

Claim 2.6.3. *For all x, $Ex \leftrightarrow \neg E(Sx)$.*

Unfortunately our axiom system is not strong enough to prove this claim. Here is a model that satisfies the axioms but for which the claim fails:

1. Include the ordinary natural numbers $0, S0, SS0$, etc. with $E0, \neg E(S0), E(SS0)$, etc.

2. Include an extra unnatural number u such that $u = Su$ and Eu holds.

It turns out that adding u doesn't violate any of the axioms. Axiom A_1 is happy, because $Eu \leftrightarrow E(SSu)$ since both u and SSu are even. Axiom A_2 is happy because $0 \neq Su$. Axiom A_3 is happy because $Sx = Sy \leftrightarrow x = y$ whenever x and y are both natural or both u, and also if one of x and y is natural and the other is u (because in this case $Sx \neq Sy$ and $x \neq y$).

But: with u in the model, we have an object for which Eu and $E(Su)$ are both true, contradicting the claim! So if we want the successor to any even number to be odd, we are going to need a bigger set of axioms.

What we are really missing here is the Axiom Schema of Induction, which says that if $P(0)$ and $\forall x : P(x) \rightarrow P(Sx)$, then $\forall x : P(x)$. Note that throwing in the Axiom Schema of Induction actually requires adding *infinitely many* axioms, since we get a distinct axiom for each choice of formula P.

Chapter 3

Set theory

Set theory is the dominant foundation for mathematics. The idea is that everything else in mathematics—numbers, functions, etc.—can be written in terms of sets, so that if you have a consistent description of how sets behave, then you have a consistent description of how everything built on top of them behaves. If predicate logic is the machine code of mathematics, set theory would be assembly language.

The nice thing about set theory is that it requires only one additional predicate on top of the standard machinery of predicate logic. This is the **membership** or **element** predicate $\in$, where $x \in S$ means that x is an element of S. Here S is a set—a collection of elements—and the identity of S is completely determined by which x satisfy $x \in S$. Every other predicate in set theory can be defined in terms of $\in$.

We'll describe two versions of set theory below. The first, **naive set theory**, treats any plausible collection of elements as a set. This turns out to produce some unfortunate paradoxes, so most mathematics is built on a more sophisticated foundation known as **axiomatic set theory**. Here we can only use those sets whose existence we can prove using a standard list of axioms. But the axioms are chosen so that all the normal things we might want to do with sets in naive set theory are explicitly possible.

3.1 Naive set theory

Naive set theory is the informal version of set theory that corresponds to our intuitions about sets as unordered collections of objects (called **elements**) with no duplicates. An element of a set may also be a set (in which case it contains its own elements), or it may just be some object that is not a set

(also known as an **urelement**, which is German for "primitive element").

A set can be written explicitly by listing its elements using curly braces:

- $\{\}$ = the **empty set** $\emptyset$, which has no elements.
- $\{\text{Moe}, \text{Curly}, \text{Larry}\}$ = the **Three Stooges**.
- $\{0, 1, 2, \ldots\} = \mathbb{N}$, the **natural numbers**. Note that we are relying on the reader guessing correctly how to continue the sequence here.
- $\{\{\}, \{0\}, \{1\}, \{0, 1\}, \{0, 1, 2\}, 7\}$ = a set of sets of natural numbers, plus a stray natural number that is directly an element of the outer set.

Membership in a set is written using the $\in$ symbol (pronounced "is an element of," "is a member of," or just "is in"). So we can write Moe $\in$ the Three Stooges or $4 \in \mathbb{N}$. We can also write $\notin$ for "is not an element of," as in Moe $\notin \mathbb{N}$, and the reversed symbol $\ni$ for "has as an element," as in $\mathbb{N} \ni 4$.

A fundamental axiom in set theory (the **Axiom of Extensionality**; see §3.4) is that the only distinguishing property of a set is its list of members: if two sets have the same members, they are the same set.

For nested sets like $\{\{1\}\}$, $\in$ represents only direct membership: the set $\{\{1\}\}$ only has one element, $\{1\}$, so $1 \notin \{\{1\}\}$. This can be confusing if you think of $\in$ as representing the English "is in," because if I put my lunch in my lunchbox and put my lunchbox in my backpack, then my lunch is in my backpack. But my lunch is *not* an element of $\{\{\text{my lunch}\}, \text{my textbook}, \text{my slingshot}\}$. In general, $\in$ is not transitive (see §9.3): it doesn't behave like $\leq$ unless there is something very unusual about the set you are applying it to. There is also no standard notation for being a deeply-buried element of an element of an element (etc.) of some set.

In addition to listing the elements of a set explicitly, we can also define a set by **set comprehension**, where we give a rule for how to generate all of its elements. This is pretty much the only way to define an infinite set without relying on guessing, but can be used for sets of any size. Set comprehension is usually written using **set-builder notation**, as in the following examples:

- $\{x \mid x \in \mathbb{N} \wedge x > 1 \wedge (\forall y \in \mathbb{N} : \forall z \in \mathbb{N} : yz = x \to y = 1 \vee z = 1)\}$ = the prime numbers.
- $\{2x \mid x \in \mathbb{N}\}$ = the even numbers.
- $\{x \mid x \in \mathbb{N} \wedge x < 12\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$.

$\{ x \mid 0 \leq x \leq 100, x = 1 \pmod 2 \}$
```
[ x | x <- [0..100], x `mod` 2 == 1 ]
[ x for x in range(0,101) if x % 2 == 1 ]
```

Table 3.1: Set comprehension vs list comprehension. The first line gives the set of odd numbers between 0 and 100 written using set-builder notation. The other lines construct the odd numbers between 0 and 100 as ordered list data structures in Haskell and Python respectively.

Some very high-level programming languages like Haskell or Python have a similar mechanism called **list comprehension** which does pretty much the same thing except the result is an ordered list. Table 3.1 gives some examples of what this looks like.

Sometimes the original set that an element has to be drawn from is put on the left-hand side of the vertical bar:

- $\{n \in \mathbb{N} \mid \exists x, y, z \in \mathbb{N} \setminus \{0\} : x^n + y^n = z^n\}$. This is a fancy name for $\{1, 2\}$, but this fact is not obvious [Wil95].

Using set comprehension, we can see that every set in naive set theory is equivalent to some predicate. Given a set S, the corresponding predicate is $x \in S$, and given a predicate P, the corresponding set is $\{x \mid Px\}$. But watch out for **Russell's paradox**: what is $\{S \mid S \notin S\}$?

3.2 Operations on sets

If we think of sets as representing predicates, each logical connective gives rise to a corresponding operation on sets:

- $A \cup B = \{x \mid x \in A \vee x \in B\}$. The **union** of A and B.
- $A \cap B = \{x \mid x \in A \wedge x \in B\}$. The **intersection** of A and B.
- $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$. The **set difference** of A and B.
- $A \triangle B = \{x \mid x \in A \oplus x \in B\}$. The **symmetric difference** of A and B.

(Of these, union and intersection are the most important in practice.)
Corresponding to implication is the notion of a **subset**:

- $A \subseteq B$ ("A is a subset of B") if and only if $\forall x : x \in A \to x \in B$.

As with $\in$, $\subseteq$ can be reversed: $A \supseteq B$ means that A is a **superset** of B, which is the same as saying $B \subseteq A$. We can also write $A \not\subseteq B$ to say that A is a not a subset of B, and the rather awkward-looking $A \subsetneq B$ to say that A is a **proper subset** of B, meaning that $A \subseteq B$ but $A \neq B$. (The standard version $A \subseteq B$ allows the case $A = B$.)

Sometimes one says A is **contained in** B if $A \subseteq B$. This is one of two senses in which A can be "in" B—it is also possible that A is in fact an element of B ($A \in B$). For example, the set $A = \{12\}$ is an element of the set $B = \{\text{Moe}, \text{Larry}, \text{Curly}, \{12\}\}$, but A is not a subset of B, because A's element 12 is not an element of B. Usually we will try to reserve "is in" for $\in$ and "is contained in" for $\subseteq$, but it's safest to use the symbols (or "is an element/subset of") to avoid any possibility of ambiguity.

Finally we have the set-theoretic equivalent of negation:

- $\bar{A} = \{x \mid x \notin A\}$. The set $\bar{A}$ is known as the **complement** of A.

If we allow complements, we are necessarily working inside some fixed **universe**, since the complement $U = \bar{\emptyset}$ of the empty set contains all possible objects. This raises the issue of where the universe comes from. One approach is to assume that we've already fixed some universe that we understand (e.g. $\mathbb{N}$), but then we run into trouble if we want to work with different classes of objects at the same time. The set theory used in most of mathematics is defined by a collection of axioms that allow us to construct, essentially from scratch, a universe big enough to hold all of mathematics without apparent contradictions while avoiding the paradoxes that may arise in naive set theory. However, one consequence of this construction is that the universe is (a) much bigger than anything we might ever use, and (b) *not* a set, making complements not very useful. The usual solution to this is to replace complements with explicit set differences: $U \setminus A$ for some specific universe U instead of $\bar{A}$.

3.3 Proving things about sets

We have three predicates so far in set theory, so there are essentially three positive things we could try to prove about sets:

1. Given x and S, show $x \in S$. This requires looking at the definition of S to see if x satisfies its requirements, and the exact structure of the proof will depend on what the definition of S is.

2. Given S and T, show $S \subseteq T$. Expanding the definition of subset, this means we have to show that every x in S is also in T. So a typical

proof will pick an arbitrary x in S and show that it must also be an element of T. This will involve unpacking the definition of S and using its properties to show that x satisfies the definition of T.

3. Given S and T, show $S = T$. Typically we do this by showing $S \subseteq T$ and $T \subseteq S$ separately. The first shows that $\forall x : x \in S \to x \in T$; the second shows that $\forall x : x \in T \to x \in S$. Together, $x \in S \to x \in T$ and $x \in T \to x \in S$ gives $x \in S \leftrightarrow x \in T$, which is what we need for equality.

There are also the corresponding negative statements:

1. For $x \notin S$, use the definition of S as before.
2. For $S \not\subseteq T$, we only need a counterexample: pick any one element of S and show that it's not an element of T.
3. For $S \neq T$, prove one of $S \not\subseteq T$ or $T \not\subseteq S$.

Note that because $S \not\subseteq T$ and $S \neq T$ are existential statements rather than universal ones, they tend to have simpler proofs.

Here are some examples, which we'll package up as a lemma:

Lemma 3.3.1. *The following statements hold for all sets S and T, and all predicates P:*

$$S \supseteq S \cap T \tag{3.3.1}$$
$$S \subseteq S \cup T \tag{3.3.2}$$
$$S \supseteq \{x \in S \mid P(x)\} \tag{3.3.3}$$
$$S = (S \cap T) \cup (S \setminus T) \tag{3.3.4}$$

Proof.

- (3.3.1) Let x be in $S \cap T$. Then $x \in S$ and $x \in T$, from the definition of $S \cap T$. It follows that $x \in S$. Since x was arbitrary, we have that for all x in $S \cap T$, x is also in T; in other words, $S \cap T \subseteq T$.
- (3.3.2). Let x be in S. Then $x \in S \vee x \in T$ is true, giving $x \in S \cup T$.
- (3.3.3) Let x be in $\{x \in S \mid P(x)\}$. Then, by the definition of set comprehension, $x \in S$ and $P(x)$. We don't care about $P(x)$, so we drop it to just get $x \in S$.
- (3.3.4). This is a little messy, but we can solve it by breaking it down into smaller problems.

  First, we show that $S \subseteq (S \setminus T) \cup (S \cap T)$. Let x be an element of S. There are two cases:

1. If $x \in T$, then $x \in (S \cap T)$.
2. If $x \notin T$, then $x \in (S \setminus T)$.

In either case, we have shown that x is in $(S \cap T) \cup (S \setminus T)$. This gives $S \subseteq (S \cap T) \cup (S \setminus T)$.

Conversely, we show that $(S \setminus T) \cup (S \cap T) \subseteq S$. Suppose that $x \in (S \setminus T) \cup (S \cap T)$. Again we have two cases:

1. If $x \in (S \setminus T)$, then $x \in S$ and $x \notin T$.
2. If $x \in (S \cap T)$, then $x \in S$ and $x \in T$.

In either case, $x \in S$.

Since we've shown that both the left-hand and right-hand sides of (3.3.4) are subsets of each other, they must be equal.

□

Using similar arguments, we can show that properties of $\wedge$ and $\vee$ that don't involve negation carry over to $\cap$ and $\cup$ in the obvious way. For example, both operations are commutative and associative, and each distributes over the other.

3.4 Axiomatic set theory

The problem with naive set theory is that unrestricted set comprehension is too strong, leading to contradictions. **Axiomatic set theory** fixes this problem by being more restrictive about what sets one can form. The axioms most commonly used are known as **Zermelo-Fraenkel set theory with choice** or **ZFC**. We'll describe the axioms of ZFC below, but in practice you mostly just need to know what constructions you can get away with.

The short version is that you can construct sets by (a) listing their members, (b) taking the union of other sets, (c) taking the set of all subsets of a set, or (d) using some predicate to pick out elements or subsets of some set.[1] The starting points for this process are the empty set $\emptyset$ and the set $\mathbb{N}$ of all natural numbers (suitably encoded as sets). If you can't construct a set in this way (like the Russell's Paradox set), odds are that it isn't a set.

These properties follow from the more useful axioms of ZFC:

[1] Technically this only gives us Z, a weaker set theory than ZFC that omits Replacement (Fraenkel's contribution) and Choice.

Extensionality Any two sets with the same elements are equal.[2]

Existence The empty set $\emptyset$ is a set.[3]

Pairing Given sets x and y, $\{x, y\}$ is a set.[4]

Union For any set of sets $S = \{x, y, z, \ldots\}$, the set $\bigcup S = x \cup y \cup z \cup \ldots$ exists.[5]

Power set For any set S, the power set $\mathcal{P}(S) = \{A \mid A \subseteq S\}$ exists.[6]

Specification For any set S and any predicate P, the set $\{x \in S \mid P(x)\}$ exists.[7] This is called **restricted comprehension**, and is an **axiom schema** instead of an axiom, since it generates an infinite list of axioms, one for each possible P. Limiting ourselves to constructing subsets of existing sets avoids Russell's Paradox, because we can't construct $S = \{x \mid x \notin x\}$. Instead, we can try to construct $S = \{x \in T \mid x \notin x\}$, but we'll find that S isn't an element of T, so it doesn't contain itself but also doesn't create a contradiction.

Infinity There is a set that has $\emptyset$ as a member and also has $x \cup \{x\}$ whenever it has x.[8] This gives an encoding of $\mathbb{N}$ where $\emptyset$ represents 0 and $x \cup \{x\}$ represents $x + 1$. Expanding out the $x + 1$ rule shows that each number is represented by the set of all smaller numbers, e.g. $3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$, which has the nice property that each number n is represented by a set with exactly n elements, and that $a < b$ can be represented by $a \in b$.[9]

Without this axiom, we only get finite sets.

(Technical note: the set whose existence is given by the Axiom of Infinity may also contain some extra elements outside of $\mathbb{N}$, but we can strip them out—with some effort—using Specification.)

---

[2] $\forall x : \forall y : (x = y) \leftrightarrow (\forall z : z \in x \leftrightarrow z \in y)$.

[3] $\exists x : \forall y : y \notin x$.

[4] $\forall x : \forall y : \exists z : \forall q : q \in z \leftrightarrow q = x \vee q = y$.

[5] $\forall x : \exists y : \forall z : z \in y \leftrightarrow (\exists q : z \in q \wedge q \in x)$.

[6] $\forall x : \exists y : \forall z : z \in y \leftrightarrow z \subseteq x$.

[7] $\forall x : \exists y : \forall z : z \in y \leftrightarrow z \in x \wedge P(z)$.

[8] $\exists x : \emptyset \in x \wedge \forall y \in x : y \cup \{y\} \in x$.

[9] Natural numbers represented in this way are called **finite von Neumann ordinals**. These are a special case of the **von Neumann ordinals**, discussed in §3.5.5.4, which can also represent values that are not finite.

There are three other axioms that don't come up much in computer science:

Foundation Every nonempty set A contains a set B with $A \cap B = \emptyset$.[10] This rather technical axiom prevents various weird sets, such as sets that contain themselves or infinite descending chains $A_0 \ni A_1 \ni A_2 \ni \dots$. Without it, we can't do induction arguments[11] once we get beyond $\mathbb{N}$.

Replacement If S is a set, and $R(x, y)$ is a predicate with the property that $\forall x : \exists! y : R(x, y)$, then $\{y \mid \exists x \in S : R(x, y)\}$ is a set.[12] Like comprehension, replacement is an axiom schema. Mostly used to construct astonishingly huge infinite sets.

Choice For any set of nonempty sets S there is a function f that assigns to each x in S some $f(x) \in x$. This axiom is unpopular in some circles because it is **non-constructive**: it tells you that f exists, but it doesn't give an actual definition of f. But it's too useful to throw out.

Like everything else in mathematics, the particular system of axioms we ended up with is a function of the history, and there are other axioms that could have been included but weren't. Some of the practical reasons for including some axioms but not others are described in a pair of classic papers by Maddy [Mad88a, Mad88b].

3.5 Cartesian products, relations, and functions

Sets are unordered: the set $\{a, b\}$ is the same as the set $\{b, a\}$. Sometimes it is useful to consider **ordered pairs** (a, b), where we can tell which element comes first and which comes second. These can be encoded as sets using the rule $(a, b) = \{\{a\}, \{a, b\}\}$, which was first proposed by Kuratowski [Kur21, Definition V].[13]

[10] $\forall x \neq \emptyset : \exists y \in x : x \cap y = \emptyset$.

[11] See Chapter 5.

[12] $(\forall x : \exists! y : R(x, y)) \to \forall z : \exists q : \forall r : r \in q \leftrightarrow (\exists s \in z : R(s, r))$.

[13] This was not the only possible choice. Kuratowski cites a previous encoding suggested by Hausdorff [Hau14] of (a, b) as $\{\{a, 1\}, \{b, 2\}\}$, where 1 and 2 are tags not equal to a or b. He argues that this definition "seems less convenient to me" than $\{\{a\}, \{a, b\}\}$, because it requires tinkering with the definition if a or b turn out to be equal to 1 or 2. This is a nice example of how even though mathematical definitions arise through convention, some definitions are easier to use than others.

Given sets A and B, their **Cartesian product** $A \times B$ is the set $\{(x, y) \mid x \in A \wedge y \in B\}$, or in other words the set of all ordered pairs that can be constructed by taking the first element from A and the second from B. If A has n elements and B has m, then $A \times B$ has nm elements.[14] For example, $\{1, 2\} \times \{3, 4\} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$.

Because of the ordering, Cartesian product is not commutative in general. We usually have $A \times B \neq B \times A$. (Exercise: when are they equal?)

The existence of the Cartesian product of any two sets can be proved using the axioms we already have: if (x, y) is defined as $\{\{x\}, \{x, y\}\}$, then $\mathcal{P}(A \cup B)$ contains all the necessary sets $\{x\}$ and $\{x, y\}$, and $\mathcal{P}(\mathcal{P}(A \cup B))$ contains all the pairs $\{\{x\}, \{x, y\}\}$. It also contains a lot of other sets we don't want, but we can get rid of them using Specification.

A special class of relations are **functions**. A function from a **domain** A to a **codomain**[15] B is a relation on A and B (i.e., a subset of $A \times B$) such that every element of A appears on the left-hand side of exactly one ordered pair. We write $f : A \to B$ as a short way of saying that f is a function from A to B, and for each $x \in A$ write $f(x)$ for the unique $y \in B$ with $(x, y) \in f$.[16]

The set of *all* functions from A to B is written as B^A: note that the order of A and B is backwards here from $A \to B$. Since this is just the subset of $\mathcal{P}(A \times B)$ consisting of functions as opposed to more general relations, it exists by the Power Set and Specification axioms.

When the domain of a function is finite, we can always write down a list of all its values. For infinite domains (e.g. $\mathbb{N}$), almost all functions are impossible to write down, either as an explicit table (which would need to be infinitely long) or as a formula (there aren't enough formulas). Most of the time we will be interested in functions that have enough structure that we can describe them succinctly, for obvious practical reasons. But in a sense these other, ineffable functions still exist, so we use a definition of a function that encompasses them.

Often, a function is specified not by writing out some huge set of ordered pairs, but by giving a rule for computing $f(x)$. An example: $f(x) = x^2$.

[14] In fact, this is the most direct way to *define* multiplication on $\mathbb{N}$, and pretty much the only sensible way to define multiplication for infinite cardinalities; see §11.1.5.

[15] The codomain is sometimes called the *range*, but most mathematicians will use **range** for $\{f(x) \mid x \in A\}$, which may or may not be equal to the codomain B, depending on whether f is or is not surjective.

[16] Technically, knowing f alone does not tell you what the codomain is, since some elements of B may not show up at all. This can be fixed by representing a function as a pair (f, B), but it's not something most people worry about.

Particular trivial functions can be defined in this way anonymously; another way to write $f(x) = x^2$ is as the anonymous function $x \mapsto x^2$.

3.5.1 Examples of functions

- $f(x) = x^2$. Note: this single rule gives several different functions, e.g. $f : \mathbb{R} \to \mathbb{R}$, $f : \mathbb{Z} \to \mathbb{Z}$, $f : \mathbb{N} \to \mathbb{N}$, $f : \mathbb{Z} \to \mathbb{N}$. Changing the domain or codomain changes the function.
- $f(x) = x + 1$.
- Floor and ceiling functions: when x is a real number, the **floor** of x (usually written $\lfloor x \rfloor$) is the largest integer less than or equal to x and the **ceiling** of x (usually written $\lceil x \rceil$) is the smallest integer greater than or equal to x. E.g., $\lfloor 2 \rfloor = \lceil 2 \rceil = 2$, $\lfloor 2.337 \rfloor = 2$, $\lceil 2.337 \rceil = 3$.
- The function from $\{0, 1, 2, 3, 4\}$ to $\{a, b, c\}$ given by the following table:

0	a
1	c
2	b
3	a
4	b

3.5.2 Sequences

Functions let us define sequences of arbitrary length: for example, the infinite sequence $x_0, x_1, x_2, \ldots$ of elements of some set A is represented by a function $x : \mathbb{N} \to A$, while a shorter sequence (a_0, a_1, a_2) would be represented by a function $a : \{0, 1, 2\} \to A$. In both cases the **subscript** takes the place of a function argument: we treat x_n as **syntactic sugar** for $x(n)$. Finite sequences are often called **tuples**, and we think of the result of taking the Cartesian product of a finite number of sets $A \times B \times C$ as a set of tuples (a, b, c), even though the actual structure may be $((a, b), c)$ or $(a, (b, c))$ depending on which product operation we do first.

We can think of the Cartesian product of k sets (where k need not be 2) as a set of sequences indexed by the set $\{1 \ldots k\}$ (or sometimes $\{0 \ldots k - 1\}$). Technically this means that $A \times B \times C$ (the set of functions from $\{1, 2, 3\}$ to $A \cup B \cup C$ with the property that for each function $f \in A \times B \times C$, $f(1) \in A$, $f(2) \in B$, and $f(3) \in C$) is not the same as $(A \times B) \times C$ (the set of all ordered pairs whose first element is an ordered pair in $A \times B$ and whose second element is in C) or $A \times (B \times C)$ (the set of ordered pairs whose first

element is in A and whose second element is in $B \times C$). This distinction has no practical effect and so we typically ignore it; the technical justification for this is that the three different representations are all **isomorphic** in the sense that a translation exists between each pair of them that preserves their structure.

A special case is the Cartesian product of no sets. This is just the set containing a single element, the empty sequence.

Cartesian products over indexed collections of sets can be written using product notation (see §6.2), as in

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

or even

$$\prod_{x \in \mathbb{R}} A_x.$$

3.5.3 Functions of more (or less) than one argument

If $f : A \times B \to C$, then we write $f(a, b)$ for $f((a, b))$. In general we can have a function with any number of arguments (including 0); a function of k arguments is just a function from a domain of the form $A_1 \times A_2 \times \ldots A_k$ to some codomain B.

3.5.4 Composition of functions

Two functions $f : A \to B$ and $g : B \to C$ can be **composed** to give a **composition** $g \circ f$. This is a function from A to C defined by $(g \circ f)(x) = g(f(x))$. Composition is often implicit in definitions of functions: the function $x \mapsto x^2 + 1$ is the composition of two functions $x \mapsto x + 1$ and $x \mapsto x^2$.

3.5.5 Functions with special properties

We can classify functions $f : A \to B$ based on how many elements x of the domain A get mapped to each element y of the codomain B. If every y is the image of at least one x, f is **surjective**. If every y is the image of at most one x, f is **injective**. If every y is the image of exactly one x, f is **bijective**. [17] These concepts are formalized below.

[17] These terms, which are generally attributed to the group of mathematicians who published under the name Bourbaki [Bou70], are now pretty well established and have the advantage of being hard to confuse with each other. An older convention in English was

3.5.5.1 Surjections

A function $f : A \to B$ that covers every element of B is called **onto**, **surjective**, or a **surjection**. This means that for any y in B, there exists some x in A such that $y = f(x)$. An equivalent way to show that a function is surjective is to show that its **range** $\{f(x) \mid x \in A\}$ is equal to its codomain.

For example, the function $f(x) = x^2$ from $\mathbb{N}$ to $\mathbb{N}$ is not surjective, because its range includes only perfect squares. The function $f(x) = x + 1$ from $\mathbb{N}$ to $\mathbb{N}$ is not surjective because its range doesn't include 0. However, the function $f(x) = x + 1$ from $\mathbb{Z}$ to $\mathbb{Z}$ *is* surjective, because for every y in $\mathbb{Z}$ there is some x in $\mathbb{Z}$ such that $y = x + 1$.

3.5.5.2 Injections

If $f : A \to B$ maps distinct elements of A to distinct elements of B (i.e., if $x \neq y$ implies $f(x) \neq f(y)$), it is called **one-to-one**, **injective**, or an **injection**. By contraposition, an equivalent definition is that $f(x) = f(y)$ implies $x = y$ for all x and y in the domain. For example, the function $f(x) = x^2$ from $\mathbb{N}$ to $\mathbb{N}$ is injective. The function $f(x) = x^2$ from $\mathbb{Z}$ to $\mathbb{Z}$ is *not* injective (for example, $f(-1) = f(1) = 1$). The function $f(x) = x + 1$ from $\mathbb{N}$ to $\mathbb{N}$ is injective.

3.5.5.3 Bijections

A function that is both surjective and injective is called a **one-to-one correspondence**, **bijective**, or a **bijection**. Any bijection f has an **inverse** function f^{-1}; this is the function $\{(y, x) \mid (x, y) \in f\}$.

Of the functions we have been using as examples, only $f(x) = x + 1$ from $\mathbb{Z}$ to $\mathbb{Z}$ is bijective.

3.5.5.4 Bijections and counting

Bijections let us define the size of arbitrary sets without having some special means to count elements. We say two sets A and B have the same **size** or **cardinality** if there exists a bijection $f : A \leftrightarrow B$.

Often it is convenient to have standard representatives of sets of a given cardinality. A common trick is to use the **von Neumann ordinals**, which are sets that are constructed recursively so that each contains all the smaller

---

to call surjective functions **onto**, injective functions **one-to-one**, and bijective functions **one-to-one correspondences**. This can lead to confusing between injective and bijection functions, so we'll stick with the less confusing French-derived terminology.

ordinals as elements.[18] The empty set $\emptyset$ represents 0, the set $\{0\}$ represents 1, $\{0, 1\}$ represents 2, and so on. The first infinite ordinal is $\omega = \{0, 1, 2, \ldots\}$, which is followed by $\omega + 1 = \{0, 1, 2, \ldots; \omega\}$, $\omega + 2 = \{0, 1, 2, \ldots; \omega, \omega + 1\}$, and so forth; there are also much bigger ordinals like ω^2 (which looks like ω many copies of ω stuck together), ω^ω (which is harder to describe, but can be visualized as the set of infinite sequences of natural numbers with an appropriate ordering), and so on. Given any collection of ordinals, it has a smallest element, equal to the intersection of all elements: this means that von Neumann ordinals are **well-ordered** (see §9.5.6). So we can define the cardinality $|A|$ of a set A formally as the unique smallest ordinal B such that there exists a bijection $f : A \leftrightarrow B$.

This is exactly what we do when we do counting: to know that there are 3 stooges, we count them off $0 \to$ Moe, $1 \to$ Larry, $2 \to$ Curly, giving a bijection between the set of stooges and $3 = \{0, 1, 2\}$.

Because different infinite ordinals may have the same cardinality, infinite cardinalities are generally not named for the smallest ordinal of that cardinality, but get their own names. So the cardinality $|\mathbb{N}|$ of the naturals is written as $\aleph_0$, the next largest possible cardinality as $\aleph_1$, etc. See §3.7.1 for more details.

3.6 Constructing the universe

With power set, Cartesian product, the notion of a sequence, etc., we can construct all of the standard objects of mathematics. For example:

Integers The integers are the set $\mathbb{Z} = \{\ldots, -2, -1, 0, -1, 2, \ldots\}$. We represent each integer z as an ordered pair (x, y), where $x = 0 \vee y = 0$; formally, $\mathbb{Z} = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid x = 0 \vee y = 0\}$. The interpretation of (x, y) is $x - y$; so positive integers z are represented as $(z, 0)$ while negative integers are represented as $(0, z)$. It's not hard to define addition, subtraction, multiplication, etc. using this representation.

[18] The formal definition is that S is an ordinal if (a) every element of S is also a subset of S; and (b) every subset T of S contains an element x with the property that $x = y$ or $x \in y$ for all $y \in T$. In other words, every subset T of S has a **minimal element** with respect to $\in$. If we treat $\in$ as $<$, this property makes S **well-ordered** (see §9.5.6). The fact that every subset of S has a minimal element means that we can do induction on S, since if there is some property that does not hold for all x in S, there must be some minimal x for which it doesn't hold. So if we can prove that $\forall y < x : P(y)$ implies $P(x)$, then it must be the case that P holds for every element of S, because otherwise we get a contradiction at the minimal x for which P does not hold.

Rationals The rational numbers $\mathbb{Q}$ are all fractions of the form p/q where p is an integer, q is a natural number not equal to 0, and p and q have no common factors. Each such fraction can be represented as a set using an ordered pair (p, q). Operations on rationals are defined as you may remember from grade school.

Reals The real numbers $\mathbb{R}$ can be defined in a number of ways, all of which turn out to be equivalent. The simplest to describe is that a real number x is represented by pair of sets $\{y \in \mathbb{Q} \mid y < x\}$ and $\{y \in Q \mid y \geq x\}$; this is known as a **Dedekind cut** [Ded01]. Formally, a Dedekind cut is any pair of subsets (S, T) of $\mathbb{Q}$ with the properties that (a) S and T **partition** $\mathbb{Q}$, meaning that $S \cap T = \emptyset$ and $S \cup T = \mathbb{Q}$; (b) every element of S is less than every element of T ($\forall s \in S \, \forall t \in T : s < t$); and (c) S contains no largest element ($\forall x \in S \, \exists y \in S : x < y$). Note that real numbers in this representation may be hard to write down.

A simpler but equivalent representation is to drop T, since it is just $\mathbb{Q} \setminus S$: this gives use a real number for any proper subset S of $\mathbb{Q}$ that has no largest element and is **downward closed**, meaning that $x < y \in S$ implies $x \in S$. Real numbers in this representation may still be hard to write down.

More conventionally, a real number can be written as an infinite decimal expansion like

$$\pi \approx 3.14159265358979323846264338327950288419716939937510582\ldots,$$

which is a special case of a **Cauchy sequence** that gives increasingly good approximations to the actual real number the further along you go.

We can also represent standard objects of computer science:

Deterministic finite state machines A **deterministic finite state machine** is a tuple $(\Sigma, Q, q_0, \delta, Q_{\text{accept}})$ where Σ is an **alphabet** (some finite set), Q is a **state space** (another finite set), $q_0 \in Q$ is an **initial state**, $\delta : Q \times \Sigma \to Q$ is a **transition function** specifying which state to move to when processing some symbol in Σ, and $Q_{\text{accept}} \subseteq Q$ is the set of **accepting states**. If we represent symbols and states as natural numbers, the set of all deterministic finite state machines is then just a subset of $\mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N}) \times \mathbb{N} \times \left(\mathbb{N}^{\mathbb{N} \times \mathbb{N}}\right) \times \mathcal{P}(\mathbb{N})$ satisfying some consistency constraints.

3.7 Sizes and arithmetic

We can compute the size of a set by explicitly counting its elements; for example, $|\emptyset| = 0$, $|\{\text{Larry}, \text{Moe}, \text{Curly}\}| = 3$, and $|\{x \in \mathbb{N} \mid x < 100 \wedge x \text{ is prime}\}| = |\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97\}| = 25$. But sometimes it is easier to compute sizes by doing arithmetic. We can do this because many operations on sets correspond in a natural way to arithmetic operations on their sizes. (For much more on this, see Chapter 11.)

Two sets A and B that have no elements in common are said to be **disjoint**; in set-theoretic notation, this means $A \cap B = \emptyset$. In this case we have $|A \cup B| = |A| + |B|$. The operation of **disjoint union** acts like addition for sets. For example, the disjoint union of 2-element set $\{0, 1\}$ and the 3-element set $\{\text{Wakko}, \text{Jakko}, \text{Dot}\}$ is the 5-element set $\{0, 1, \text{Wakko}, \text{Jakko}, \text{Dot}\}$.

The size of a Cartesian product is obtained by multiplication: $|A \times B| = |A| \cdot |B|$. An example would be the product of the 2-element set $\{a, b\}$ with the 3-element set $\{0, 1, 2\}$: this gives the 6-element set $\{(a, 0), (a, 1), (a, 2), (b, 0), (b, 1), (b, 2)\}$. Even though Cartesian product is not generally commutative, swapping each pair (a, b) to (b, a) is a bijection, so $|A \times B| = |B \times A|$.

For power set, it is not hard to show that $|\mathcal{P}(S)| = 2^{|S|}$. This is a special case of the size of A^B, the set of all functions from B to A, which is $|A|^{|B|}$; for the power set we can encode $\mathcal{P}(S)$ using 2^S, where 2 is the special set $\{0, 1\}$, and a subset T of S is encoded by the function that maps each $x \in S$ to 0 if $x \notin T$ and 1 if $x \in T$.

3.7.1 Infinite sets

For infinite sets, we take the above properties as *definitions* of addition, multiplication, and exponentiation of their sizes. The resulting system is known as **cardinal arithmetic**, and the sizes that sets (finite or infinite) might have are known as **cardinal numbers**.

The finite cardinal numbers are just the natural numbers: $0, 1, 2, 3, \ldots$. The first infinite cardinal number is the size of the set of natural numbers, and is written as $\aleph_0$ (**aleph-zero**, **aleph-null**, or **aleph-nought**). The next infinite cardinal number is $\aleph_1$ (**aleph-one**): it might or might not be the size of the set of real numbers, depending on whether you include the **Generalized Continuum Hypothesis** in your axiom system.[19]

[19]The generalized continuum hypothesis says (essentially) that there aren't any more cardinalities out there in between the ones whose existence can be deduced from the other axioms of set theory. A consequence of this is that there are no cardinalities between $|\mathbb{N}|$ and $|\mathbb{R}|$. An alternative notation exists if you don't want to take a position on GCH:

Infinite cardinals can behave very strangely. For example:

- $\aleph_0 + \aleph_0 = \aleph_0$. In other words, it is possible to have two sets A and B that both have the same size as $\mathbb{N}$, take their disjoint union, and get another set $A + B$ that has the same size as $\mathbb{N}$. To give a specific example, let $A = \{2x \mid x \in \mathbb{N}\}$ (the **even numbers**) and $B = \{2x + 1 \mid x \in \mathbb{N}\}$ (the **odd numbers**). These have $|A| = |B| = |\mathbb{N}|$ because there is a bijection between each of them and $\mathbb{N}$ built directly into their definitions. It's also not hard to see that A and B are disjoint, and that $A \cup B = \mathbb{N}$. So $|A| = |B| = |A| + |B|$ in this case.

  The general rule for cardinal addition is that $\kappa + \lambda = \max(\kappa, \lambda)$ if at least one of κ and λ is infinite. Sums of finite cardinals behave exactly the way you expect.

- $\aleph_0 \cdot \aleph_0 = \aleph_0$. Example: A bijection between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$ using the **Cantor pairing function** $\langle x, y \rangle = (x+y+1)(x+y)/2+y$. The first few values of this are $\langle 0, 0 \rangle = 0, \langle 1, 0 \rangle = 2 \cdot 1/2 + 0 = 1, \langle 0, 1 \rangle = 2 \cdot 1/2 + 1 = 2, \langle 2, 0 \rangle = 3 \cdot 2/2 + 0 = 3, \langle 1, 1 \rangle = 3 \cdot 2/2 + 1 = 4, \langle 0, 2 \rangle = 3 \cdot 2/2 + 2 = 5$, etc. The basic idea is to order all the pairs by increasing $x+y$, and then order pairs with the same value of $x + y$ by increasing y. Eventually every pair is reached.

  The general rule for cardinal multiplication is that $\kappa \cdot \lambda = \max(\kappa, \lambda)$ if at least one of κ or λ is infinite. So $\kappa \cdot \lambda = \kappa + \lambda$ if either is infinite (or both are zero).

- $\mathbb{N}^* = \{\text{all finite sequences of elements of } \mathbb{N}\}$ has size $\aleph_0$. One way to do this to define a function recursively by setting $f([]) = 0$ and $f([\text{first}, \text{rest}]) = 1 + \langle \text{first}, f(\text{rest}) \rangle$, where first is the first element of

this writes $\beth_0$ ("beth-0") for $|\mathbb{N}|$, $\beth_1$ ("beth-1") for $|\mathbb{R}| = |\mathcal{P}(\mathbb{N})|$, with the general rule $\beth_{i+1} = 2^{\beth_i}$. This avoids the issue of whether there exist sets with size between $\mathbb{N}$ and $\mathbb{R}$, for example. In my limited experience, only hard-core set theorists ever use $\beth$ instead of $\aleph$: in the rare cases where the distinction matters, most normal mathematicians will just assume GCH, which makes $\beth_i = \aleph_i$ for all i.

the sequence and rest is all the other elements. For example,

$$\begin{aligned} f(0,1,2) &= 1 + \langle 0, f(1,2)\rangle \\ &= 1 + \langle 0, 1 + \langle 1, f(2)\rangle\rangle \\ &= 1 + \langle 0, 1 + \langle 1, 1 + \langle 2, 0\rangle\rangle\rangle \\ &= 1 + \langle 0, 1 + \langle 1, 1 + 3\rangle\rangle = 1 + \langle 0, 1 + \langle 1, 4\rangle\rangle \\ &= 1 + \langle 0, 1 + 19\rangle \\ &= 1 + \langle 0, 20\rangle \\ &= 1 + 230 \\ &= 231. \end{aligned}$$

This assigns a unique element of $\mathbb{N}$ to each finite sequence, which is enough to show $|\mathbb{N}^*| \leq |\mathbb{N}|$. With some additional effort one can show that f is in fact a bijection, giving $|\mathbb{N}^*| = |\mathbb{N}|$.

3.7.2 Countable sets

The sets $\mathbb{N}$, $\mathbb{N}^2$, and $\mathbb{N}^*$ all have the property of being **countable**, which means that they can be put into a bijection with $\mathbb{N}$ or one of its subsets. Countability of $\mathbb{N}^*$ means that anything you can write down using finitely many symbols (even if they are drawn from an infinite but countable alphabet) is countable. This has a lot of applications in computer science: one of them is that the set of all computer programs in any particular programming language is countable.

3.7.3 Uncountable sets

Exponentiation is different. We can easily show that $2^{\aleph_0} \neq \aleph_0$, or equivalently that there is no bijection between $\mathcal{P}(\mathbb{N})$ and $\mathbb{N}$. This is done using **Cantor's diagonalization argument**, which appears in the proof of the following theorem.

Theorem 3.7.1. *Let S be any set. Then there is no surjection $f : S \to \mathcal{P}(S)$.*

Proof. Let $f : S \to \mathcal{P}(S)$ be some function from S to subsets of S. We'll construct a subset of S that f misses, thereby showing that f is not a surjection. Let $A = \{x \in S \mid x \notin f(x)\}$. Suppose $A = f(y)$. Then $y \in A \leftrightarrow y \notin A$, a contradiction.[20] □

[20] Exercise: Why does A exist even though the Russell's Paradox set doesn't?

Since any bijection is also a surjection, this means that there's no bijection between S and $\mathcal{P}(S)$ either, implying, for example, that $|\mathbb{N}|$ is strictly less than $|\mathcal{P}(\mathbb{N})|$.

(On the other hand, it is the case that $\left|\mathbb{N}^{\mathbb{N}}\right| = \left|2^{\mathbb{N}}\right|$, so things are still weird up here.)

Sets that are larger than $\mathbb{N}$ are called **uncountable**. A quick way to show that there is no surjection from A to B is to show that A is countable but B is uncountable. For example:

Corollary 3.7.2. *There are functions $f : \mathbb{N} \to \{0, 1\}$ that are not computed by any computer program.*

Proof. Let P be the set of all computer programs that take a natural number as input and always produce 0 or 1 as output (assume some fixed language), and for each program $p \in P$, let f_p be the function that p computes. We've already argued that P is countable (each program is a finite sequence drawn from a countable alphabet), and since the set of all functions $f : \mathbb{N} \to \{0, 1\} = 2^{\mathbb{N}}$ has the same size as $\mathcal{P}(\mathbb{N})$, it's uncountable. So some f gets missed: there is at least one function from $\mathbb{N}$ to $\{0, 1\}$ that is not equal to f_p for any program p. □

The fact that there are more functions from $\mathbb{N}$ to $\mathbb{N}$ than there are elements of $\mathbb{N}$ is one of the reasons why set theory (slogan: "everything is a set") beat out **lambda calculus** (slogan: "everything is a function from functions to functions") in the battle over the foundations of mathematics. And this is why we do set theory in CPSC 202 and lambda calculus (disguised as Scheme) in CPSC 201.

3.8 Further reading

See [Ros12, §§2.1–2.2], [Big02, Chapter 2], or [Fer08, §1.3, §1.5].

Chapter 4

The real numbers

The **real numbers** $\mathbb{R}$ are the subject of high-school algebra and most practical mathematics. Some important restricted classes of real numbers are the **naturals** $\mathbb{N} = 0, 1, 2, \ldots$, the **integers** $\mathbb{Z} = \ldots, -2, -1, 0, 1, 2, \ldots$, and the **rationals** $\mathbb{Q}$, which consist of all real numbers that can be written as ratios of integers p/q, otherwise known as **fractions**.

The rationals include $1, 3/2, 22/7, -355/113$, an so on, but not some common mathematical constants like $e \approx 2.718281828\ldots$ or $\pi \approx 3.141592\ldots$. Real numbers that are not rational are called **irrational**. There is no single-letter abbreviation for the irrationals.

The typeface used for $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ is called **blackboard bold** and originates from the practice of emphasizing a letter on a blackboard by writing it twice. Some writers just use ordinary boldface: $\mathbf{N}$, etc., but this does not scream out "this is a set of numbers" as loudly as blackboard bold. You may also see blackboard bold used for the **complex numbers** $\mathbb{C}$, which are popular in physics and engineering, and for some more exotic number systems like the **quaternions** $\mathbb{H}$,[1] which are sometimes used in graphics, or the **octonions** $\mathbb{O}$, which exist mostly to see how far complex numbers can be generalized.

Like any mathematical structure, the real numbers are characterized by a list of **axioms**, which are the basic facts from which we derive everything we know about the reals. There are many equivalent ways of axiomatizing the real numbers; we will give one here. Many of these properties can also be found in [Fer08, Appendix B]. These should mostly be familiar to you from high-school algebra, but we include them here because we need to know

[1] Why $\mathbb{H}$? The rationals already took $\mathbb{Q}$ (for "quotient"), so the quaternions are abbreviated by the initial of their discoverer, William Rowan Hamilton.

what we can assume when we want to prove something about reals, and also because it lets us sneaky in definitions of various algebraic structures like groups and fields that will turn out to be useful later.

4.1 Field axioms

The real numbers are a **field**, which means that they support the operations of addition $+$, multiplication $\cdot$, and their inverse operations subtraction $-$ and division $/$. The behavior of these operations is characterized by the **field axioms**.

4.1.1 Axioms for addition

Addition in a field satisfies the axioms of a **commutative group** (often called an **abelian group**, after Niels Henrik Abel, an early nineteenth-century mathematician). These characterize the behavior of the addition operation $+$ and sums of the form $a + b$ ("a **plus** b").

Axiom 4.1.1 (Commutativity of addition)**.** *For all numbers,*

$$a + b = b + a. \tag{4.1.1}$$

Any operation that satisfies Axiom 4.1.1 is called **commutative**. Commutativity lets us ignore the order of arguments to an operation. Later, we will see that multiplication is also commutative.

Axiom 4.1.2 (Associativity of addition)**.** *For all numbers,*

$$a + (b + c) = (a + b) + c. \tag{4.1.2}$$

An operation that satisfies Axiom 4.1.2 is called **associative**. Associativity means we don't have to care about how a sequence of the same associative operation is parenthesized, letting us write just $a + b + c$ for $a + (b + c) = (a + b) + c$.[2]

[2] A curious but important practical fact is that addition is often *not* associative in computer arithmetic. This is because computers (and calculators) approximate real numbers by **floating-point numbers**, which only represent the some limited number of digits of an actual real number in order to make it fit in limited memory. This means that low-order digits on very large numbers can be lost to **round-off error**. So a computer might report $(1000000000000 + -1000000000000) + 0.00001 = 0.00001$ but $1000000000000 + (-1000000000000 + 0.00001) = 0.0$. Since we don't have to write any programs in this class, we will just work with actual real numbers, and not worry about such petty numerical issues.

Axiom 4.1.3 (Additive identity)**.** *There exists a number* 0 *such that, for all numbers* a*,*

$$a + 0 = 0 + a = a. \tag{4.1.3}$$

An object that satisfies the condition $a+0 = 0+a = a$ for some operation is called an **identity** for that operation. Later we will see that 1 is an identity for multiplication.

It's not hard to show that identities are unique:

Lemma 4.1.4. *Let* $0' + a = a + 0' = a$ *for all* a*. Then* $0' = 0$*.*

Proof. Compute $0' = 0' + 0 = 0$. (The first equality holds by the fact that $a = a + 0$ for all a and the second from the assumption that $0' + a = a$ for all a.) □

Axiom 4.1.5 (Additive inverses)**.** *For each* a*, there exists a number* $-a$*, such that*

$$a + (-a) = (-a) + a = 0. \tag{4.1.4}$$

For convenience, we will often write $a + (-b)$ as $a - b$ ("a **minus** b"). This gives us the operation of **subtraction**. The operation that returns $-a$ given a is called **negation** and $-a$ can be read as "**negative** a," "**minus** a",[3] or the "negation of a."

Like identities, inverses are also unique:

Lemma 4.1.6. *If* $a' + a = a + a' = 0$*, then* $a' = -a$*.*

Proof. Starting with $0 = a' + a$, add $-a$ on the right to both sides to get $-a = a' + a + -a = a'$. □

4.1.2 Axioms for multiplication

Multiplication in a field satisfies the axioms of a commutative group, if the additive identity 0 is excluded.

For convenience[4], the multiplication operation $\cdot$ is often omitted, allowing us to write ab for $a \cdot b$. We will use this convention when it will not cause confusion.

[3] Warning: Some people will get annoyed with you over "minus a" and insist on reserving "minus" for the operation in $a - b$. In extreme cases, you may see $-a$ typeset differently: $\text{-}a$. Pay no attention to these people. Though not making the distinction makes life more difficult for calculator designers and compiler writers, as a working mathematician you are entitled to **abuse notation** by using the same symbol for multiple purposes when it will not lead to confusion.

[4] Also called "laziness."

Axiom 4.1.7 (Commutativity of multiplication)**.** *For all numbers,*

$$ab = ba. \tag{4.1.5}$$

Axiom 4.1.8 (Associativity of multiplication)**.** *For all numbers,*

$$a(bc) = (ab)c. \tag{4.1.6}$$

Axiom 4.1.9 (Multiplicative identity)**.** *There exists a number* $1 \neq 0$ *such that, for all numbers* a*,*

$$a \cdot 1 = 1 \cdot a = a. \tag{4.1.7}$$

We insist that $1 \neq 0$ because we want Axiom 4.1.9 to hold in $\mathbb{R} \setminus \{0\}$. This also has the beneficial effect of preventing us from having $\mathbb{R} = \{0\}$, which would otherwise satisfy all of our axioms.

Since the only difference between the multiplicative identity and the additive identity is notation, Lemma 4.1.4 applies here as well: if there is any $1'$ such that $a \cdot 1' = 1' \cdot a = a$ for all a, then $1' = 1$.

Axiom 4.1.10 (Multiplicative inverses)**.** *For every* a *except* 0*, there exists a number* a^{-1}*, such that*

$$a \cdot a^{-1} = a^{-1} \cdot a = 1. \tag{4.1.8}$$

Lemma 4.1.6 applies here to show that a^{-1} is also unique for each a.

For convenience, we will often write $a \cdot b^{-1}$ as a/b or the vertical version $\frac{a}{b}$. This gives us the operation of **division**. The expression a/b or $\frac{a}{b}$ is pronounced "a **over** b" or (especially in elementary school, whose occupants are generally not as lazy as full-grown mathematicians) "a **divided by** b." Some other notations for this operation are $a \div b$ and $a : b$. These are also mostly used in elementary school.[5]

Note that because 0 is not guaranteed to have an inverse,[6] the meaning of $a/0$ is not defined.

The number a^{-1}, when it does exist, is often just called the **inverse** of a or sometimes "inverse a." (The ambiguity that might otherwise arise with the additive inverse $-a$ is avoided by using negation for $-a$.) The multiplicative inverse a^{-1} can also be written using the division operation as $1/a$.

[5] Using a colon for division is particularly popular in German-speaking countries, where the "My Dear Aunt Sally" rule for remembering that multiplication and division bind tighter than addition and subtraction becomes the more direct *Punktrechnung vor Strichrechnung*—"point reckoning before stroke reckoning."

[6] In fact, once we get a few more axioms, terrible things will happen if we try to make 0 have an inverse.

4.1.3 Axioms relating multiplication and addition

Axiom 4.1.11 (Distributive law)**.** *For all a, b, and c,*

$$a \cdot (b + c) = ab + ac \tag{4.1.9}$$

$$(a + b) \cdot c = ac + bc \tag{4.1.10}$$

Since multiplication is commutative, we technically only need one of (4.1.9) and (4.1.10), but there are other structures we will see called **rings** that satisfy the distributive law without having a commutative multiplication operation, so it's safest to include both.

The additive identity 0 also has a special role in multiplication, which is a consequence of the distributive law: it's an **annihilator**:

Lemma 4.1.12. *For all a,*

$$a \cdot 0 = 0 \cdot a = 0. \tag{4.1.11}$$

Proof. Because $0 = 0 + 0$, we have $a \cdot 0 = a \cdot (0 + 0) = a \cdot 0 + a \cdot 0$. But then adding $-(a \cdot 0)$ to both sides gives $0 = a \cdot 0$. □

Annihilation is why we don't want to define 0^{-1}, and thus won't allow division by zero. If there were a real number that was 0^{-1}, then for any a and b we would have:

$$\begin{aligned} a \cdot 0 = b \cdot 0 = 0 \\ (a \cdot 0) \cdot 0^{-1} = (b \cdot 0) \cdot 0^{-1} \\ a \cdot (0 \cdot 0^{-1}) = b \cdot (0 \cdot 0^{-1}) \\ a \cdot 1 = b \cdot 1 \\ a = b. \end{aligned}$$

(Exercise: which axiom is used at each step in this proof?)

In particular, we would get $1 = 0$, contradicting Axiom 4.1.9.

A similar argument shows that

Lemma 4.1.13. *If $a \cdot b = 0$, then $a = 0$ or $b = 0$.*

Proof. Suppose $a \cdot b = 0$ but $a \neq 0$.[7] Then a has an inverse a^{-1}. So we can compute

$$a \cdot b = 0 \tag{4.1.12}$$

$$a^{-1} \cdot a \cdot b = a^{-1} \cdot 0 \tag{4.1.13}$$

$$b = 0. \tag{4.1.14}$$

[7] This is an example of the proof strategy where we show $P \vee Q$ by assuming $\neg P$ and proving Q.

□

Another consequence of the distributive law is that we can determine how multiplication interacts with negation. You may recall being taught at an impressionable age that

$$a \cdot (-b) = -(ab), \tag{4.1.15}$$
$$(-a) \cdot b = -(ab), \tag{4.1.16}$$

and

$$(-a) \cdot (-b) = ab. \tag{4.1.17}$$

Like annihilation, these are not axioms—or at least, we don't have to include them as axioms if we don't want to. Instead, we can prove them directly from axioms and theorems we've already got. For example, here is a proof of (4.1.15):

$$\begin{aligned} a \cdot 0 &= 0 \\ a \cdot (b + (-b)) &= 0 \\ ab + a \cdot (-b) &= 0 \\ -(ab) + (ab + a \cdot (-b)) &= -(ab) \\ (-(ab) + ab) + a \cdot (-b) &= -(ab) \\ 0 + a \cdot (-b) &= -(ab) \\ a \cdot (-b) &= -(ab). \end{aligned}$$

Similar proofs can be given for (4.1.16) and (4.1.17).

A special case of this is that multiplying by -1 is equivalent to negation:

Corollary 4.1.14. *For all a,*

$$(-1) \cdot a = -a. \tag{4.1.18}$$

Proof. Using (4.1.17), $(-1) \cdot a = -(1 \cdot a) = -a$. □

4.1.4 Other algebras satisfying the field axioms

The field axioms so far do not determine the real numbers. They also hold for any number of other fields, including the rationals $\mathbb{Q}$, the complex numbers $\mathbb{C}$, and various finite fields such as the integers modulo a prime p (written as $\mathbb{Z}_p$; we'll see more about these in Chapter 14).

They do not hold for the integers $\mathbb{Z}$ (which don't have multiplicative inverses) or the natural numbers $\mathbb{N}$ (which don't have additive inverses either). This means that $\mathbb{Z}$ and $\mathbb{N}$ are not fields, although they are examples of weaker algebraic structures (a **ring** in the case of $\mathbb{Z}$ and a **semiring** in the case of $\mathbb{N}$).

In order to get the reals, we will need a few more axioms.

4.2 Order axioms

Unlike $\mathbb{C}$ and $\mathbb{Z}_p$ (but like $\mathbb{Q}$), the real numbers are an **ordered field**, meaning that in addition to satisfying the field axioms, there is a relation $\leq$ that satisfies the axioms:

Axiom 4.2.1 (Comparability)**.** *$a \leq b$ or $b \leq a$.*

Axiom 4.2.2 (Antisymmetry)**.** *If $a \leq b$ and $b \leq a$, then $a = b$.*

Axiom 4.2.3 (Transitivity)**.** *If $a \leq b$ and $b \leq c$, then $a \leq c$.*

Axiom 4.2.4 (Translation invariance)**.** *If $a \leq b$, then $a + c \leq b + c$.*

Axiom 4.2.5 (Scaling invariance)**.** *If $a \leq b$ and $0 \leq c$, then $a \cdot c \leq b \cdot c$.*

The first three of these mean that $\leq$ is a **total order** (see §9.5.5). The other axioms describe how $\leq$ interacts with addition and multiplication.

For convenience, we define $a < b$ as shorthand for $a \leq b$ and $a \neq b$, and define reverse operations $a \geq b$ (meaning $b \leq a$) and $a > b$ (meaning $b < a$). If $a > 0$, we say that a is **positive**. If $a < 0$, it is **negative**. If $a \geq 0$, it is **non-negative**. **Non-positive** can be used to say $a \leq 0$, but this doesn't seem to come up as much as non-negative.

Other properties of $\leq$ can be derived from these axioms.

Lemma 4.2.6 (Reflexivity)**.** *For all x, $x \leq x$.*

Proof. Apply comparability with $y = x$. □

Lemma 4.2.7 (Trichotomy)**.** *Exactly one of $x < y$, $x = y$, or $x > y$ holds.*

Proof. First, let's show that at least one holds. If $x = y$, we are done. Otherwise, suppose $x \neq y$. From comparability, we have $x \leq y$ or $y \leq x$. Since $x \neq y$, this gives either $x < y$ or $x > y$.

Next, observe that $x = y$ implies $x \not< y$ and $x \not> y$, since $x < y$ and $x > y$ are both defined to hold only when $x \neq y$. This leaves the possibility that $x < y$ and $x > y$. But then $x \leq y$ and $y \leq x$, so by anti-symmetry, $x = y$, contradicting our assumption. So at most one holds. □