

M. ANDREW MOSHIER

CONTEMPORARY  
DISCRETE  
MATHEMATICS

M&H PUBLISHING



Copyright © 2025 M. Andrew Moshier

PUBLISHED BY M&H PUBLISHING



# Contents

	<i>What is this about?</i>	7
	<i>I Natural Numbers and Induction</i>	13
1	<i>The Natural Numbers</i>	15
	<i>The Basic Picture</i>	15
	<i>Narrowing our pictures to actual counting</i>	17
	<i>The Axiom of Induction</i>	18
	<i>Basic Arithmetic Operations</i>	20
	<i>Precedence, association, and infix notation</i>	24
2	<i>Laws of Arithmetic</i>	26
	<i>Monoids</i>	28
	<i>Inductive Proofs</i>	30
3	<i>Orderings of the Natural Numbers</i>	41
	<i>The Standard Order</i>	42
	<i>Divisibility</i>	50
	<i>Laws of Ordered Arithmetic</i>	54
	<i>Subtraction and Division in Natural Numbers</i>	57
	<i>Quotient and remainder</i>	60

4	<i>Examples of Recursion and Induction</i>	65
	<i>Factorial and related operations</i>	66
	<i>Path finding</i>	71
	<i>Binomial coefficients</i>	74
5	<i>Lists</i>	76
	<i>List Basics</i>	76
	<i>List Indexing</i>	86
	<i>Homogeneous lists</i>	88
	<i>Slices</i>	92
	<i>Prefix order</i>	93
	<i>Sorted Lists</i>	94
	<i>Lazy lists, streams, sequences and coinduction</i>	97
	 <i>II Structure</i>	 104
6	<i>Composition</i>	107
	<i>Combinational Boolean Circuits</i>	108
	<i>Components and interfaces</i>	109
	<i>Sequential Composition</i>	109
	<i>Isomorphisms: Lossless conversion</i>	112
	<i>External Diagrams</i>	113
	<i>Parallel composition</i>	115
	<i>Non-strictness</i>	120
7	<i>An Overview of Sets, Functions, Predicates, and Relations</i>	122
	<i>Sets</i>	123
	<i>Functions</i>	130
	<i>Predicates and Relations</i>	133
	<i>Chickens and eggs</i>	141

8	<i>Injections and Surjections</i>	142
	<i>Injections and Surjections</i>	143
	<i>Bijections</i>	148
9	<i>The Size of Sets</i>	149
	<i>The Size of Sets</i>	149
	<i>Characteristic Functions and the Size of the Powersets</i>	157
	<i>Embeddings, size comparison, and the Cantor-Bernstein Theorem</i>	160
10	<i>Operations on Sets</i>	166
	<i>The Finitary Structure of Powersets</i>	166
	<i>Cartesian Products and Binary Relations</i>	171
	<i>Binary Relations and Quantification</i>	174
	<i>Quantifiers and Completeness</i>	176
	<i>Singletons and Atomicity</i>	178
	<i>Images</i>	180
11	<i>Classifications on Sets</i>	183
	<i>Equivalence Relations</i>	183
	<i>Partitions</i>	188
	<i>Quotients</i>	192
12	<i>The Integers and Rationals</i>	195
	<i>Integer Numbers</i>	195
	<i>Rational numbers</i>	203
	<i>III Applications</i>	206
13	<i>Minimum and Maximum</i>	207
14	<i>Greatest Common Divisors and Least Common Multiples</i>	211
	<i>Relative Primality</i>	217

15 *Prime Numbers* 220

16 *Counting* 228

A *Proofs* 229

*The purpose of a proof* 229

*Structure of Proofs* 231

*Context of a proof* 233

*Proof tactics* 235

*Imperative sentences in a proof* 236



## *What is this about?*

DISCRETE MATHEMATICS consists of many individual topics that, imprecisely, contrast with *continuous mathematics*, topics like calculus, real analysis, differential equations, and so on. But a sharp contrast between discrete and continuous mathematics is mainly fictional and convenient only for putting some things in one course syllabus, and some things in another.

Discrete mathematics texts tend to be disorganized (partly a symptom of the artificial separation from continuous mathematics), featuring disparate topics that start looking like an Island of Misfit Math — stuff that simply does not fit anywhere else. Some people call it “math your math teachers neglected to teach you.”

We reject that low view, regarding discrete mathematics as a coherent set of ideas centered around some key themes:

- Discrete mathematics is mainly about structure and information.
- The natural numbers and closely related inductive structures are central partly because they provide a bridge between mathematics and computation.
- Mathematical data is naturally “typed”. Natural numbers constitute a type of data; real numbers constitute a different type of data; polynomials with integer coefficients constitute yet another type of data, and so on.
- Careful use of mathematical notation and careful use of English convey information about structure. Consider “For some real number  $x$ ,  $x^2 = 4$ ” versus “For any real number  $x$ ,  $x^2 = 4$ .” One is true, the other is not. Yet they differ only by a single word. This is typical of mathematical writing.
- Algebraic (equational and inequational) reasoning provides key insights about how structures are be related.
- Mathematics itself is computational — at least it is extremely productive to think that way as much as possible.
- Abstraction and use of analogy provide a crucial means of discovering new ideas, and of tying what we know together.

In these lecture notes, we deal head on with mathematics as the study of abstract structure. This can be frustrating at first because the concrete applications are not always obvious. The pay off comes when we see that generalization from particulars leads to much wider applications than we could have anticipated.

For example,  $154133 + 11^{11^{11}}$  (a number with more than 100 trillion digits) is not divisible by 11. This number is far too big to calculate explicitly as a numeral in base 10. We can not actually hope to perform the exponentiations and addition, and check whether the result is divisible by 11 by using the standard

long division algorithm. Yet, we are completely confident that *if* we could perform the calculations, the remainder of the division would be different from 0. But, why? Well, because of the following reasoning.

We should be able to convince ourselves that  $m^n$  is always divisible by  $m$  whenever  $n$  is a positive integer, not just by taking that for granted. Likewise, we can check (by performing a long division) that 154133 is not evenly divisible by 11. Finally, we should also be able to convince ourselves that if  $m$  is not divisible by some number, let's call it  $p$ , and  $n$  is divisible by  $p$ , then  $m + n$  is not divisible by  $p$ . So, we can reason about this particular example indirectly by working out more general principles. As a bonus, we have just discovered an infinity of numbers that are not divisible by 11, namely, all of the form  $154133 + 11^n$ , where  $n$  is any natural number.

To show a slightly more intricate example, we can be completely certain that  $5^{12^{19}+1} + 2^{12^{19}+1}$  is divisible by 7, reasoning that the numbers of the form  $5^{2n+1} + 2^{2n+1}$  are all divisible by 7, no matter what natural number  $n$  we choose. Figuring out how to convince someone that something like this must be true is actually one of the main activity of mathematics.

### *Persuasion*

Mathematicians spend a lot of time thinking about what is or isn't true. But that would be pretty useless, if they did not then communicate the result of their thinking. It is not enough to know how to solve the problem. To actually *put mathematics to work*, you must learn how to persuade someone else to accept a proposed solution to a problem.

The standards for persuasion in mathematics are very high. We do not settle for "preponderance of evidence" or "beyond reasonable doubt", or "that seems reasonable." We aim for "beyond any doubt." In most human endeavors (including in the sciences), that standard would be paralyzing, but in mathematics, where abstract structure is the object of investigation, it is not only within reach, but it is the standard that makes sense.

In mathematics, a **proof** is a convincing argument. So it falls into the genre of persuasive writing. This point of view is important for understanding how mathematicians work. You will do yourself a favor if you start thinking of a mathematical proof as a (usually very short) essay.

### *Example: The irrational nature of square root of two*

Let's prove that a square root of 2 is not a rational number. Usually, we say this by claiming the square root of 2 is *irrational*. But irrational simply means not rational.

A *rational number* is a number that is formed by dividing an integer by a non-zero integer. So a rational number is  $\frac{n}{d}$  for some integer numerator  $n$  and some integer denominator  $d$  ( $d$  must not equal 0).

The claim is that it is impossible for a rational number to be equal to the square root of 2. That means that the square of any given rational number cannot equal 2. So really what we want to prove is that  $(\frac{n}{d})^2 \neq 2$ , no matter what integers  $n$  and  $d$  are.

Let's suppose  $n$  and  $d$  are integers and  $d \neq 0$ . Our goal is to figure out why  $(\frac{n}{d})^2 \neq 2$  or, equivalently, why  $\frac{n^2}{d^2} \neq 2$ , or why  $n^2 \neq 2d^2$ .

Notice that any fraction of the form  $\frac{2m}{2k}$  can be "reduced" to  $\frac{m}{k}$ . This implies that, after a finite number of steps, we can find another fraction equal to  $\frac{n}{d}$  whose numerator and denominator are not both even. We can assume that  $\frac{n}{d}$  is already at this stage. Remember also that the square of an odd number is again an odd number, and the square of an even number is again even.

Suppose then that  $n$  is an odd number, and hence  $n^2$  is also odd. Then certainly  $n^2 \neq 2d^2$ , because the  $2d^2$  is even. On the other hand, suppose  $n$  is even, that is,  $n = 2k$  for some integer  $k$ . Then  $d$  is odd (because we set things up so that  $n$  and  $d$  can not both be even) and therefore  $d^2$  is odd too. If  $n^2 = 2d^2$ , then we would have that  $4k^2 = 2d^2$ , which implies  $2k^2 = d^2$ . That is not possible because  $d^2$  is odd. Therefore,  $n^2 \neq 2d^2$ . In summary, whether  $n$  is odd or even makes no difference, in both cases we have that  $\left(\frac{n}{d}\right)^2 \neq 2$ .

The point is that this is a convincing argument (at least it should be convincing to anyone willing to follow the details) that a certain thing is not possible. The emphasis on *writing an argument* is important. After all, no one cares what one of us knows about square roots. What matters is that we can explain it.

Most exercises in these notes are *not* of the sort that are familiar in other mathematics textbooks: “Solve for  $x$ :  $4x - 3 = x^2$ .” Rather, an exercise might say “Show that the equation  $4x - 3 = x^2$  has exactly two solutions.” The former would tempt you simply to write (with a box around the answer, of course) something like  $x = 3$ . The latter asks you to *convince* the reader that these two values ( $x = 1$  and  $x = 3$ ) are solutions and that there are no other possibilities. That is a more interesting thing to do.

Particularly in the first part of the text, we emphasize somewhat formulaic writing. This is meant to give you practice with precise mathematical writing. Think of it as practicing basic scales and arpeggios.

## *Roughly, an Outline*

These notes start by investigating two ideas that are fundamental to all of mathematics.

- First, we look carefully at the idea of counting things. You might think there is not much to say about something you have been doing reliably for most of your life. But arithmetic (addition and multiplication, for example) is tightly related to counting. Understanding arithmetic relies on understanding how counting works. This “counting leads to arithmetic” approach gives you practice reasoning carefully in a topic with which you are already familiar.
- Second is the idea of putting things into lists. Again, you have been doing this a long time. You write down shopping lists, you take two lists and merge them into one list, and so on. Lists are crucial for everything in mathematics, even how we write a number in base ten. After all, what exactly does 3429 mean? A *base ten numeral* is formed by putting symbols (digits) next to each other into a certain order. So 3429 is not the same numeral as 3492. Lists provide a general way to think about things like this.

After looking at the basics of counting and of lists, we turn to the foundational languages of contemporary mathematics that allow us to investigate structure: composition, sets, functions and relations. By “composition”, we simply mean to look very generally at how things can be put together (how they can be composed). This leads to very general kinds of mathematical structures called *categories*. Sets are simply collections of things, such as  $\mathbb{N}$  — the collection of natural numbers. Mathematicians frequently use sets similar to the way computer scientists use datatypes, though there are technical differences. A function is a correlation of the things in one set with things in another set. A familiar example is  $f(x) = x^2$  that correlates with any real number  $x$ , the square  $x^2$ . Functions may be composed, so they form an important example of a category. A relation is what it says it is: a way to relate things in one set to things in another. Relations (at least certain kinds) can also be composed. So they form a different category.

To make sense of sets, functions and relations, we need to consider some basic questions:

- What does it mean to say two sets are equal?

- What does it mean to say two functions are equal?
- What does it mean to say two relations are equal?
- How can we specify or “build” particular sets?
- How can we specify or “build” particular functions?
- How can we specify or “build” particular relations?

The middle part of this text concerns these questions.

In the last part of the text, we investigate a variety of useful mathematical ideas using the techniques developed in the first two parts. The emphasis is mostly on practical ideas that either emerge from computing, or are directly useful there.

### *Some basics about notation*

Before we launch into mathematics proper, we need to sort out some basic notation and discuss how to read this text. As usual, we typically use single letters, for example,  $x$  in  $x^2 - 2x + 1$ , as placeholders for values. These are frequently called “variables”, but there may or may not be anything “varying” about the idea. For example, in the equation  $0 = x^2 - 2x + 1$ , there is exactly one real number (1) that can be the value of  $x$ . So sometimes folks refer to  $x$  as an “unknown”. But in a situation like this, we know perfectly well that  $x$  must refer to 1. Calling it “unknown” seems a bit silly. A neutral term for something like this is “Identifier”. So “ $x$  is an identifier” just means that we are using  $x$  as a name for something. Maybe it varies; maybe it is not known. Whatever.

There is nothing special about the name ‘ $x$ ’. I might as well have written  $y^2 - 2y + 1$  or  $a^2 - 2a + 1$ . The difference is only apparent when we make assumptions about the identifier. For example, look at these two:

- “Suppose  $x$  is a positive real number. Consider  $x^2 + 3x + 1$ .”
- “Suppose  $a$  is a positive real number. Consider  $x^2 + 3x + a$ .”

Clearly, they mean different things. In the first one, the polynomial expression is guaranteed to be positive. In the second one, the polynomial in  $x$  has an extra parameter. Contrast this with these two:

- “Suppose  $x$  is a positive real number. Consider  $x^2 + 3x + 1$ .”
- “Suppose  $a$  is a positive real number. Consider  $a^2 + 3a + 1$ .”

Evidently, these two really do mean the same thing. It is important to keep details like this in mind. To understand how a variable should be interpreted look for the context in which it is first mentioned.

Frequently, we use letters in certain parts of the alphabet to refer to specific types of values. For example, we mostly use  $m$ ,  $n$  and  $p$  to refer to natural numbers, and keep using  $x$ ,  $y$  and  $z$  for generic values (not from a particular type of value). Usually  $f$ ,  $g$ ,  $h$  stand for functions. Later on, other types of entities will be needed. We also use Greek letters for some things. It is in your interest to be familiar with the Greek alphabet — at least, some commonly used letters.

Sometimes, a datum should have no meaning apart from its name. **Symbols** are just names with no interpretation. We typically write a symbol in a bold type face like so: **waffles**. Thus  $x$  is a symbol, whereas  $x$  is an identifier. In handwritten text, a symbol might be indicated by underlining.

Three other type faces will come in handy: `san serif`, `C`, and `SMALL CAPS`. Usage will not make much sense now. We can wait to explain how to use them when the time is right.

Parentheses are horribly confusing in mathematics because they are used in several distinct ways. You need to get used to the confusion. Here are some of the ways they are used.

- Function application: We frequently write  $f(5)$  to mean “apply the function  $f$  to the value 5.”
- Pairs, triples and so on:  $(4, 5)$  means “the ordered pair consisting of 4 then 5.” This is what you are used to when you think about cartesian coordinates.
- Ranges: In some situations, the notation  $(0, 1)$  means “the collection of real numbers strictly between 0 and 1. Watch out for this one. It is very easy to confuse the pair  $(0, 1)$  with the range  $[0, 1]$ .”
- Grouped sub-expressions: In an expression such as  $a + b \cdot c$ , you know to evaluate  $b \cdot c$  before adding that to  $a$ . But to get the result of multiplying  $a + b$  with  $c$ , you would write  $(a + b) \cdot c$ .

Usually, you will know what is meant if you pay close attention.

Throughout the text, we refer to certain parts as *algorithms* or *definitions*. The distinction is not precise. The idea is that an algorithm is meant to convey a way of calculating something; a definition is meant to convey a way to talk about something. For example, addition of two natural numbers is presented as an algorithm because there is a precise sequence of concrete steps to take when adding. But  $\leq$  (the relation of being less than or equal to) is presented as a definition. Though there may be a calculation to determine whether  $m \leq n$ , the calculation is not the main idea.

An algorithm in the text is not a program in a particular programming language. Rather, an algorithm is just a clear description of a procedure that might be carried out by hand or by computer. In principle, algorithms in the text could be programmed in a suitable actual programming language, but our emphasis is not coding.

### Margin notes

Margin notes provide side discussions that are not directly in the main course of development. Two special symbols are used in margin notes.



☕ indicates a note that you should slow down to think about carefully. It is important, and is not necessarily easy to grasp. Go get a cup of coffee, or take a walk, while you think.



✎ indicates an easy way to locate exercises.



*Part I*  
*Natural Numbers and Induction*

OUR EARLIEST MATHEMATICAL EXPERIENCE is learning to count. Arithmetic on the natural numbers, simple as it seems, exhibits many of the features of higher contemporary mathematics that concern us throughout this course. By starting with a careful look at arithmetic, you prepare yourself for what comes in this course and in later mathematics.

An important technique for reasoning about natural numbers and many other mathematical structures is called **induction**. Induction is the main theme of Part I, and is used throughout the rest of the text.



# 1

## The Natural Numbers

*Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.*

— Leopold Kronecker

ORDER THREE TACOS and get two. You know something is wrong. Ten comes right after nine. If you have counted up to some humungous number  $m$ , you know that you could count one more if you needed to. In short, you know how to count. In fact, you have known how to count very reliably since you were about four or five years old. You even had most of the ingredients by the time you were two or three. Now, stop and think about that.

There were a whole lot of things you couldn't do at that age, but you had already mastered an important bit of mathematical machinery. As you will see shortly, the parts of that machinery are actually quite sophisticated. Indeed, a theme of this text is that a vast range of advanced mathematics originates in simple ideas.

Given the really young age at which we start to figure out how to count, the ability to count seems to be *innate*. We seem to be born with it, even if it takes a little time to figure out how to use it. That, we think, is part of what Leopold Kronecker was getting at when he quipped that the natural numbers are made by the loving God, and the rest (of math) is humans' work. The ability to count seems to be cooked in. So, let's look at the ingredients.

### The Basic Picture

Counting amounts to answering a question "How many are there?". The answer could be 1, 2, 3 and so on. The answer could also be 0. For example, how many great Country Western songs are there? Zeeero.

On the other hand, "how many?" cannot be answered with  $\frac{1}{2}$ ,  $\pi$ ,  $-5$  and so on. How many Austin Powers movies are there? Twelve

---

#### CHAPTER GOALS

---

Investigate the structure of counting and explain that structure via postulates for the natural numbers.

---

Not all philosophers of mathematics agree that the ability to count is innate, citing some interesting anthropological observations, particularly the existence of some human languages that seem not to have any words corresponding to counting words. Their arguments don't convince us, but that's fine. This text doesn't depend on who is right.

---

#### AN INFORMAL DEFINITION OF NATURAL NUMBERS

---

Counting amounts to answering a question "How many are there?". A *counting* or **natural** number is a number that can be used to answer a question of the form "How many X's are there?"

---

and one quarter? No! Square root of 10? No! Negative 2? No! Three? Yes.

Natural numbers may be pictured like stepping stones as in Figure 1.1.

Not all pictures of “stepping stones” and “arrows” are acceptable as pictures of the natural numbers. Whereas Figure 1.1 depicts a good picture of natural numbers, Figures 1.2, 1.3 and 1.4 illustrate three ways we know *not* to imagine them. These can be ruled out by explaining the basic *vocabulary* of counting.

---

### VOCABULARY 1: Basic Vocabulary of Natural Numbers

---

The *natural numbers* have the following features.

- There is a special natural number, called **zero**, denoted by 0.
- For any natural number  $n$ , there is a unique next natural number, called the **successor** of  $n$ . In these notes, the successor of  $n$  is denoted by  $n^\frown$ .

The set of natural numbers is denoted by  $\mathbb{N}$ . To indicate that an identifier  $n$  is intended to range over natural numbers, we write  $n \in \mathbb{N}$ . So for example, to say “Every  $n \in \mathbb{N}$  is either odd or even” is the same as to say “Every natural number is either odd or even.” We also use  $\in$  to indicate a particular object is a member of the collection. So “ $5 \in \mathbb{N}$ ” is a true statement, and “ $\frac{1}{2} \in \mathbb{N}$ ” is false.

---

According to Vocabulary 1, expressions like 0,  $0^\frown$ ,  $0^{\frown\frown}$ ,  $0^{\frown\frown\frown}$  denote natural numbers. Of course, you would rather abbreviate these by writing 0, 1, 2, 3. But the characters 1, 2, 3, etc., are not related to each other. They are just squiggly shapes that we have all learned to associate with corresponding numbers. They don’t help us understand *counting*, though they unquestionably help make things more efficient.

In contrast, the notation  $0^\frown$  suggests that the number is meant to be the number after 0. Likewise,  $0^{\frown\frown}$  is meant to be the number after the number after 0, and so on. You can always switch between the familiar “decimal” notation and “successor” notation whenever it is convenient. But let’s leave it informal for now because the conversion is kind of obvious. Later on, you’ll look at the details of how that switch is done as an example of something more general. For now, though, let’s just agree that it is “obvious” how to go from  $0^{\frown\frown\frown\frown}$  to 4, or in the other direction from 1213 to  $0^{\frown\frown\cdots\frown}$ . You get the idea. We sometimes say 4 **abbreviates**  $0^{\frown\frown\frown\frown}$  if we need to be clear about it. We might also say 4 abbreviates  $3^\frown$ , if that helps keep things clear.

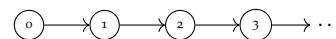


Figure 1.1: A picture of the natural numbers

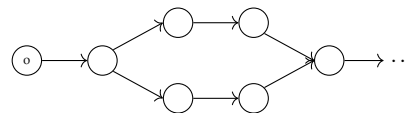


Figure 1.2: Forks in the path

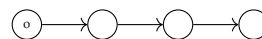


Figure 1.3: A path with nowhere to go

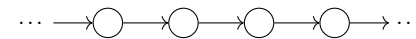


Figure 1.4: A path with nowhere to start

The notation  $n^\frown$  is not standard. I use it to distinguish the successor of  $n$  from  $n + 1$ , because the latter presupposes that we already know what  $+$  means. Another common notation is to write  $S(n)$  or  $S\ n$  for “the successor of  $n$ .”

The symbol “ $\mathbb{N}$ ” is very common usage. Any mathematician will know immediately that you mean the **set of natural numbers** when you use this symbol.

What we mean by the *set of natural numbers* is that  $\mathbb{N}$  is a mathematical object in its own right. Each natural number is an object, and  $\mathbb{N}$  is the collection consisting of all of them. In Chapter 7, we will make the idea of sets more precise.

Note that  $\frown$  is an operation that takes a natural number and produces another natural number. We can indicate this by writing  $\frown: \mathbb{N} \rightarrow \mathbb{N}$  saying that  $\frown$  is a **function from  $\mathbb{N}$  to  $\mathbb{N}$** . This extra bit of notation would be pretty useless on its own, but we will define many other functions. So having a convenient way to specify the “interface” will be helpful. Sometimes it is more convenient to write  $\mathbb{N} \xrightarrow{\frown} \mathbb{N}$  instead of  $\frown: \mathbb{N} \rightarrow \mathbb{N}$ , but these mean the same thing.

Parentheses frequently are used to indicate an order of evaluation. For example, once we have addition and multiplication sorted out,  $m \cdot (n + p)$  will mean that  $m$  is to be multiplied by the result of  $n + p$ . In expressions involving  $\smile$ , parentheses are usually redundant. For example,  $(0^\smile)^\smile$  means the same as  $0^{\smile\smile}$ .

At the end of a section, the usual professorly thing to do is to give you some exercises. For this section, the exercises would involve converting between successor notation and standard base ten notation. There are no exercises for this section. Thank us later.

### *Narrowing our pictures to actual counting*

FIGURES 1.5 AND 1.6 show correct pictures of Vocabulary 1. That is, each picture has a designated 0 and in each picture, every “stepping stone” has a next one. Obviously, these are not pictures of the natural numbers. So the vocabulary is not enough. We also need somehow to rule these bad pictures out.

Figure 1.5 is flawed because 0 has a “predecessor” but for counting, nothing should come before 0. Figure 1.6 is flawed because an element has two distinct predecessors:  $0^\smile = 0^{\smile\smile\smile}$ . We can stop that nonsense with two postulates.

---

#### POSTULATE 1: Nothing Precedes 0

---

*For every natural number  $n$ , it is the case that  $n^\smile \neq 0$ .*

---



---

#### POSTULATE 2: Predecessors are Unique

---

*For any natural numbers  $m$  and  $n$ , if  $m^\smile = n^\smile$  then  $m = n$ .*

---

Suppose you are told that “ $n$  has a predecessor.” Then you know  $n = k^\smile$  for some  $k$  (though you may still not know which  $k$ ). By the two postulates, you immediately also know two more things. First,  $n \neq 0$  because 0 does not have a predecessor. Second,  $k$  is unique. You can speak about *the* predecessor of  $n$ . Natural numbers that possess a predecessor are therefore special.



Be ware that  $0^{(\smile\smile)}$  doesn't really make sense.

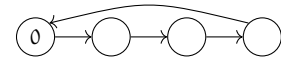


Figure 1.5: A strange way to count:

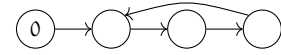


Figure 1.6: Another strange way to count

Does a Monopoly board satisfy this postulate?

Does a Monopoly board satisfy this postulate? What about Chutes and Ladders?

---

**DEFINITION 1:** Predecessors and Positive Natural Numbers
 

---

For two natural numbers  $k$  and  $n$ , we say  $k$  is the **predecessor** of  $n$  if and only if  $n$  is the successor of  $k$ , that is,  $n = k^\circ$ . A natural number  $m$  is **positive** if it has a predecessor. In particular,  $0$  is not positive. Adapting the notation  $\mathbb{N}$  for the set of all natural numbers,  $\mathbb{N}^+$  denotes the collection of all positive natural numbers. So  $0 \in \mathbb{N}$ , but  $0 \notin \mathbb{N}^+$ .

For a positive natural number  $n$ , let  $\text{pred}(n)$  denote its predecessor. The defining facts about  $\text{pred}$  are that

$$\text{pred}(n)^\circ = n$$

for every positive natural number  $n$ , and

$$\text{pred}(m^\circ) = m$$

for every natural number  $m$ .

---

Pay close attention to this definition. It says a natural number is positive if and only if it has a predecessor. It does not say a natural number is positive if it is *greater than*  $0$  because right now “greater than” doesn’t seem to mean anything. We don’t have a definition. Of course, we know intuitively what “greater than” should mean. But we need a way to think about the concept formally. That comes in Chapter 3.

### The Axiom of Induction

Although Postulates 1 and 2 eliminate Figures 1.5, 1.6 and similar bad pictures, there is still a problem illustrated in Figure 1.7. The stepping stone labelled  $\star$  has a unique predecessor because  $\star^\circ = \star$ . So  $\star$  is *positive* according to the definition. But this seems wrong. The extra stepping stone does not belong because no matter how long you *count* starting from  $0$ , you will never reach  $\star$ . It is extra. It can not participate in counting,

The picture in Figure 1.7 models the basic vocabulary correctly because  $0$  is there, and every element has a successor. The picture also satisfies the first two postulates because  $0$  does not have a predecessor, and every positive element has a unique predecessor. The trouble is that  $\star$  is a *cheat*. It comes out of nowhere.

The remedy is to require that no extra numbers are allowed. This leads to the last postulate for natural numbers.

A diagonal strike-through indicates “not”. So the notation  $0 \notin \mathbb{N}^+$  is analogous to  $6 \neq 7 + 1$ .

Notice that  $\text{pred}(0)$  is nonsense because  $0$  is not positive. It has no predecessor. Of course, if we thought we were talking about integers instead of natural numbers, then  $\text{pred}(0)$  would make sense, it would be  $-1$ . But for natural numbers, there simply is no such thing as a predecessor of  $0$ .

It is helpful to be able to spell out the fact that  $\text{pred}$  denotes an operation “turning” a positive natural number  $n$  into a natural number. Like successor, the interface information is summarized by writing  $\text{pred}: \mathbb{N}^+ \rightarrow \mathbb{N}$ . This says that  $\text{pred}$  is an operation that turns any element of  $\mathbb{N}^+$  into an element of  $\mathbb{N}$ . This does not tell us precisely what  $\text{pred}$  does, but just tells us that if  $n$  is a positive natural number, then  $\text{pred}(n)$  makes sense and is guaranteed to be a natural number.

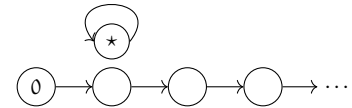


Figure 1.7: A model of the natural numbers?

---

**POSTULATE 3:** The Axiom of Induction (the “No Cheating” Axiom)
 

---

*Natural numbers cannot be invented willy-nilly. Anything that is not required to exist by Vocabulary 1 is not a natural number. To be more precise, no natural numbers can be eliminated without violating the basic vocabulary.*

---

Believe it or not, the basic vocabulary and these three postulates completely characterize the picture of the natural numbers. Any picture that satisfies these will look the same. A rigorous proof of this is possible, but not necessary for now.

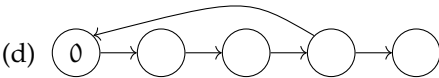
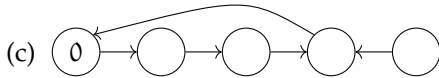
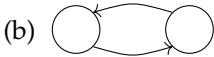
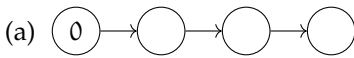



---

**EXERCISES:**


---

1. Each of the following pictures violates either the vocabulary or one or more of the postulates. For each, explain what is violated.



2. Suppose Pat has in mind a picture for Vocabulary 1 and Postulates 1 and 2. Furthermore, in the picture, Pat has in mind an element  $n$  for which (a)  $n \neq 0$  and (b)  $n$  has no predecessor (that is,  $n \neq k^{\sim}$  for every  $k$  in the picture). Convince Pat that the picture fails to satisfy Postulate 3.
3. Draw three different pictures of situations that satisfy all the postulates except that they fail Postulate 1. So there will be an arrow from some stepping stone into the stepping stone labelled 0. The result must satisfy all other postulates including the Axiom of Induction.
- 

### Pattern matching

Everything we have agreed upon regarding the natural numbers can be summarized in terms of a simple rule of pattern matching that we make more precise later in the text.

When I was little, I argued with my brother Rick a lot. It was entertaining, especially on long car ride. The following is a transcript of an actual argument (to the best of my recall):

*Me* Look at that bird. Is that an eagle?

*Rick* That's not even a bird.

*Me* Yes it is.

*Rick* No it isn't.

*Me* Yes it is.

*Rick* No it isn't.

⋮

*Me* Yes it is.

*Rick* No it isn't, times infinity.

At that point, I knew I had been cheated. Arguing with your brother is a game of stamina. The whole point is to outlast your opponent, who should give up in exhaustion and boredom. But Rick cheated by just making up some crazy new move out of nowhere. Bah! I was not having it.

Nobody gets to pull natural numbers out of thin air. And just saying “yes it is, times infinity” does not mean he said “yes it is” infinitely many times. Rick cheated.

Exercise 2 shows that in the natural numbers, if  $n \neq 0$  then  $n$  is positive. In other words, every  $n \in \mathbb{N}$  falls into exactly one of two mutually exclusive cases: either  $n = 0$  or  $n \in \mathbb{N}^+$ .

The postulates (including Induction) are summarized by saying that  $\mathbb{N}$  (the collection of natural numbers) is defined **inductively** by the rule for  $n \in \mathbb{N}$ ,

$$n \longleftrightarrow 0 \mid n^{\frown}.$$

Informally, the rule says that every natural number matches exactly one of the two patterns  $0$  or  $n^{\frown}$  in exactly one way, and that anything that matches one of the two patterns is a natural number (that is the point of the basic vocabulary). In particular,  $0$  is a natural number, if  $n$  is a natural number, then  $n^{\frown}$  is also, and if a natural number matches the pattern  $n^{\frown}$ , then  $n$  is uniquely determined. By saying  $\mathbb{N}$  is defined *inductively*, we assert that the only things that are natural numbers are those things that this rule requires. So  $0$ ,  $0^{\frown}$ ,  $0^{\frown\text{frown}}$ , and so on, are natural numbers, but nothing else is.

Take a moment to think about how to this rule seems to capture the essentials of counting.

### *Basic Arithmetic Operations*

Addition works by counting ahead. To *add*  $4 + 5$ , you could hold out 4 fingers on one hand, and 5 on another, then count all the fingers. This is probably how you first learned to add. A bit later, you figured out that you could just start at 4 and then count ahead five more: 5, 6, 7, 8, 9. Even later, you memorized that  $4 + 5$  is 9.

Multiplication works by counting a number of additions. You first learned to think of  $2 \cdot 3$  as meaning 2 plus 2 plus 2. And later you memorized facts like  $2 \cdot 3$  is 6.

We can spell out these “learner’s” methods of arithmetic.

---

#### ALGORITHM 1: Addition

---

*The sum of  $m \in \mathbb{N}$  and  $n \in \mathbb{N}$  is a natural number  $m + n$ , calculated by the following:*

$$\begin{aligned} m + 0 &= m \\ m + k^{\frown} &= (m + k)^{\frown} \end{aligned} \quad \text{for any } k \in \mathbb{N}.$$


---

To understand what we mean by the word “algorithm,” we need to spell out how to use the algorithms that appear in the text. The idea in Algorithm 1 is that the two equations tell us to replace any expression that matches the pattern “ $m + 0$ ” with “ $m$ ”, and likewise any expression that matches “ $m + k^{\frown}$ ” with “ $(m + k)^{\frown}$ ”, where  $m$

An **algorithm** is a special kind of definition, one that defines how to calculate something. So for example, addition just *is* this calculation, and  $m + n$  is the result of performing the calculation.

So  $+$  has a precise, mechanical, meaning. It will be up to us to check that the algorithm expresses what we mean by true addition.

and  $k$  are any expressions corresponding to a natural number. For example, " $0^\sim + 0^\sim$ " matches the pattern by

$$\boxed{0^\sim}^m + \boxed{0^\sim}^k.$$

So the algorithm specifies that this is replaced by

$$\left( \boxed{0^\sim}^m + \boxed{0^\sim}^k \right)^\sim.$$

So, in principle, a user of this algorithm does not need to know anything except basic pattern matching. But there are still some details to work out.

Consider the expression  $0 + (0^\sim + 0^\sim)^\sim$ . There are two different places to match the pattern  $m + k^\sim$ . So our rule can not be used automatically. We have to decide what to do first, resulting either in  $(0 + (0^\sim + 0^\sim)^\sim)^\sim$  or in  $0 + ((0^\sim + 0)^\sim)^\sim$ . This is a problem because it looks like the algorithm is not specific enough to tell us what to do. It turns out, though, that the way addition is defined here ensures that the order in which the rules are applied doesn't matter. If we simply use the rules (pattern match the left side of an equation and replace with corresponding right side) anywhere the matching works, we will be guaranteed eventually to reach an expression where no further pattern matching is possible. We say the algorithm is **terminating**. Moreover, no matter in what order we do the match-and-replace, the resulting terminal expression will be the same. We say the algorithm is **confluent**. The conditions that guarantee termination and confluence are rather technical. We defer discussing them to later courses.

---

#### EXAMPLE 1:

---

Calculate  $4 + 3$ :

$4 + 3 = 4 + 0^\sim$	— 3 abbreviates $0^\sim$
$= (4 + 0^\sim)^\sim$	— The addition algorithm
$= (4 + 0^\sim)^\sim$	— Same reason
$= (4 + 0)^\sim$	— Same reason
$= 4^\sim$	— Same reason
$= (0^\sim)^\sim$	— 4 abbreviates $0^\sim$
$= 0^\sim$	— Remove unneeded parentheses
$= 7$	— 7 abbreviates $0^\sim$

---

When you need to calculate something "by hand", use this format. Write the first expression followed by  $=$ , followed by the first step in the calculation. Align the equals signs and write subsequent steps on new lines. This makes it much easier to check your own work. Also use another column to write notes explaining why a step is valid. These explanatory notes are not always needed when a step is obvious, but for now get in the habit of writing this way.

The multiplication of two natural numbers is also defined via a confluent and terminating algorithm.

---

**ALGORITHM 2: Multiplication**


---

The product of  $m \in \mathbb{N}$  and  $n \in \mathbb{N}$  is a natural number  $m \cdot n$ , calculated by the following:

$$\begin{aligned} m \cdot 0 &= 0 \\ m \cdot k^{\curvearrowright} &= m + (m \cdot k) \quad \text{for any } k \in \mathbb{N}. \end{aligned}$$


---

The vocabulary and postulates of natural numbers ensure that there are indeed unique operations  $+$  and  $\cdot$  satisfying the equations — the “algorithms” really are algorithms. A detailed proof of this fact is not illuminating right now. Later, you will do better, by seeing that any algorithm written this way really does specify an operation. Then addition and multiplication will turn out just to be particular instances of a general method for specifying algorithms.

---

**EXAMPLE 2:**


---

Calculate  $3 \cdot 2$ :

$$\begin{aligned} 3 \cdot 2 &= 3 \cdot 0^{\curvearrowright\curvearrowright} && \text{— 2 abbreviates } 0^{\curvearrowright\curvearrowright} \\ &= 3 + (3 \cdot 0^{\curvearrowright}) && \text{— The multiplication algorithm} \\ &= 3 + (3 + (3 \cdot 0)) && \text{— Same reason} \\ &= 3 + (3 + 0) && \text{— Same reason} \\ &= 3 + 3 && \text{— The addition algorithm} \\ &= 3 + 0^{\curvearrowright\curvearrowright\curvearrowright} && \text{— 3 abbreviates } 0^{\curvearrowright\curvearrowright\curvearrowright} \\ &= (3 + 0^{\curvearrowright\curvearrowright})^{\curvearrowright} && \text{— The addition algorithm} \\ &= (3 + 0^{\curvearrowright})^{\curvearrowright\curvearrowright} && \text{— Same reason} \\ &= (3 + 0)^{\curvearrowright\curvearrowright\curvearrowright} && \text{— Same reason} \\ &= 3^{\curvearrowright\curvearrowright\curvearrowright} && \text{— Same reason} \\ &= (0^{\curvearrowright\curvearrowright\curvearrowright})^{\curvearrowright\curvearrowright\curvearrowright} && \text{— 3 abbreviates } 0^{\curvearrowright\curvearrowright\curvearrowright} \\ &= 0^{\curvearrowright\curvearrowright\curvearrowright\curvearrowright\curvearrowright\curvearrowright} && \text{— Remove unnecessary parentheses} \\ &= 6 && \text{— 6 abbreviates } 0^{\curvearrowright\curvearrowright\curvearrowright\curvearrowright\curvearrowright\curvearrowright} \end{aligned}$$


---

You certainly will not want to calculate this way in real life. Even a simple calculation like  $3 \cdot 2 = 6$  took thirteen steps. But these examples

In the text, we almost always denote multiplication with an explicit dot. Though it is conventional to write things like  $2m$  instead of  $2 \cdot m$ , or  $(x + y)z$  instead of  $(x + y) \cdot z$ , the multiplication sign “ $\cdot$ ” makes things clearer in most situations.

Both addition and multiplication operate on two natural numbers to produce another natural number. We summarize this by writing

$$+: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N},$$

indicating that  $+$  is the sort of operation that needs two natural numbers ( $\mathbb{N} \times \mathbb{N}$ ) to produce one natural number ( $\mathbb{N}$ ). Likewise,

$$\cdot: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

indicates the same information about the  $\cdot$  operation.



illustrate how addition and multiplication are built mechanistically from counting.

There are some mechanical details that deserve to be ignored for now. We do not spell out rules for swapping between decimal notation (3) and unary notation ( $0^{\frown\circ\circ\circ}$ ). We also do not spell out when to remove unnecessary parentheses (or even what “unnecessary” means). The main reason to ignore these details is that you are not a computer. If we were trying fully to implement these algorithms on a machine, we would need to be precise about every small detail. But for us humans, those details are “obvious” and kind of boring. We count on you being able to deal with things like this so we can skip some of the boring stuff.




---

#### EXERCISES:

---

4. Show that  $n + 1 = n^{\frown}$  for every natural number  $n$ .
  5. Calculate these sums, writing each step of your calculation explicitly. Include the reason for each step (as in the previous example). Take care to lay out the chain of equalities correctly, and do not skip any steps.
    - (a)  $2 + 4$
    - (b)  $4 + 2$
    - (c)  $3 + (3 + 1)$
    - (d)  $(3 + 3) + 1$
    - (e)  $0 + 3$
  6. Notice that it takes more steps to calculate  $2 + 4$  than  $4 + 2$ , even though they produce the same answer. Explain why.
  7. Calculate the following values, writing each step explicitly.
    - (a)  $2 \cdot 3$
    - (b)  $0 \cdot 2$
    - (c)  $2 \cdot (2 \cdot 2)$
    - (d)  $3 \cdot (2 + 1)$
    - (e)  $3 \cdot 2 + 2 \cdot 3$
  8. Write a definition of exponentiation via equations for  $m^0$  and for  $m^{k^{\frown}}$ . Follow the pattern of definition for addition and multiplication.
-

### *Precedence, association, and infix notation*

Exercise 7e is sort of a trick question. Almost surely, you actually understood it to mean “multiply  $3 \cdot 2$ , multiply  $2 \cdot 3$ , then add the two results.” But why? Reading left to right, why not “multiply  $3 \cdot 2$ , then add 2 to the result, then multiply that by 3”? The reason you likely did what you did is tradition. In terms of mechanical calculation, this is one of those obvious, but boring details.

The tradition is that multiplications are evaluated before additions, unless parentheses override that. So  $m + n \cdot p$  is understood to mean “add  $m$  to the product of  $n$  and  $p$ ”. Using parentheses, this is the same as  $m + (n \cdot p)$ .

In general, the rules detailing which operations are to be evaluated earlier are called **precedence rules**. So we say that multiplication **has higher precedence than** addition.

Traditionally, exponentiation has higher precedence than multiplication. If you see  $m \cdot n^p$ , you know it means  $m \cdot (n^p)$ , and not  $(m \cdot n)^p$ . On the other hand, superscripts themselves are implicitly parenthesized. So  $m^{n \cdot p}$  means  $m^{(n \cdot p)}$ . The successor notation has high precedence. So  $m \cdot n^\frown$  means  $m \cdot (n^\frown)$ , not  $(m \cdot n)^\frown$ . As we encounter other operations such as subtraction, division, and even other more exotic ones, we will want to decide their precedence as well.

For familiar binary arithmetic operations like addition and multiplication, tradition also requires that we write the operation between the two operands. So we write  $5 + 7$ . The term for this tradition is **infix notation**. Many other binary operations do not follow this tradition, typically when the operation is given a name rather than a special symbol. For example, the maximum of two numbers is usually written  $\max(m, n)$  even though  $\max$  is a binary operation similar to  $+$ . To distinguish the notation, we say that  $\max$  is written using **prefix notation**. It is also reasonable to consider **postfix notation**, but that is a bit rarer in mathematical writing. Successor in this text uses postfix notation because  $^\frown$  is written *after* the operand.

The general story of “fixity” is this:

*Prefix* An operation is written using *prefix notation* if it is written before all of its operands. Examples are the trigonometric functions:  $\sin \theta$ ,  $\cos x$ , and so on.

*Infix* An operation is written in *infix notation* if it is written after its first operand. Standard arithmetic operations use infix notation:  $5 + 4$ ,  $9 - 6$ , and so on.

*Postfix* An operation is written in *postfix notation* if it is written after all of its operands. You may not have seen many examples of this,

The terms *prefix*, *infix*, *postfix* are used more regularly in computer science than in pure mathematics. We don’t emphasize them in this text. But it is helpful to be aware that notation can be subtle in this way. You will probably encounter prefix and infix notation in other courses (for example, in a logic course if you are a mathematics student, or in a programming languages course if you are a computer science student). So don’t concentrate on this now. Just file it away for later.

One last comment about notation for now. Mathematicians also sometimes use “two dimensional” notation such as  $\frac{3}{4}$  where symbols are stacked up vertically. Exponents and subscripts are also examples of “two dimensional” notation. But generally speaking these tend to be kind of “one off” designs, and most programming languages are designed to use standard one dimensional text. So there is very little common terminology surrounding them. We limit the use of such features to situations where that is the traditional notation (such as fractions and exponents), or where there simply doesn’t seem to be a nice alternative.

but the factorial of a natural number  $n$  is written  $n!$ .

Now consider how to interpret an expression like  $m + n + p$ . We know that addition is associative, so it doesn't matter whether this is meant to be  $(m + n) + p$ , or  $m + (n + p)$ . But in Chapter 3, we will introduce a new operation  $\div$  on natural numbers that is not associative. Since  $(m \div n) \div p$  is not generally equal to  $m \div (n \div p)$ , the parentheses make a difference, even though they tend to clutter more complicated expressions. It would be nice to agree on where the implicit parentheses ought to go so that  $m \div n \div p$  makes sense. In most situations like  $\div$ , it will make sense to "do the first operation, then second, etc." so that  $m \div n \div p$  means  $(m \div n) \div p$ . We say that  $\div$  **associates to the left**. A few operations, notably, the logical operation of implication (written  $\rightarrow$ ), make sense to interpret as associating to the right instead, so that  $A \rightarrow B \rightarrow C$  will mean  $A \rightarrow (B \rightarrow C)$ .

In summary, for any new operation we need to decide three syntactic features. First, decide whether it will be written in prefix notation (e.g.,  $\sin(\theta)$  and  $\max(m, n)$ ), or infix notation (e.g.,  $m + n$  and  $m \cdot n$ ), or postfix notation (e.g.,  $n!$ , and  $n^{\wedge}$ ). Second, if it is written using infix notation, what is its precedence (e.g.,  $\cdot$  has higher precedence than  $+$ )? Finally, if it is written using infix notation, does it associate to the left or to the right?

## 2

# Laws of Arithmetic

*Well, Dr. Shuman tells me that in theory there is nothing the computer can do that the human mind cannot do. The computer merely takes a finite amount of data and performs a finite amount of operations on them. The human mind can duplicate the process.*

— Congressman Brant in *The Feeling Of Power*,  
a short story by Isaac Asimov

WITHOUT A PENCIL AND PAPER, WITHOUT A COMPUTER, using only your brain, calculate

$$221 \cdot 23 + 221 \cdot 17.$$

This looks pretty tedious, and fraught with danger (most mathematicians is really bad at arithmetic ... and counting ... and grammar). But you could just as well calculate

$$221 \cdot (23 + 17).$$

Seeing that  $23 + 17$  is 40, you only need to figure  $221 \cdot 40$ . Boom: 8840!

You are quite sure that is correct because you know several facts about addition and multiplication. To calculate the original problem, just using the algorithms for addition and multiplication would have been mind-numbingly dull. But using what we know, the problem turns out to be very easy.

Thankfully, we do know several laws of arithmetic that allow us to skip explicitly using the algorithms. One of the key laws is that multiplication *distributes over* addition:  $m \cdot n + m \cdot p = m \cdot (n + p)$  for any natural numbers, but you use other laws all the time, such as commutativity —  $m + n = n + m$  — and associativity —  $m + (n + p) = (m + n) + p$ .

What it means to call these patterns “laws” is that they are always true — for any natural numbers — by virtue of how counting works (characterized in the postulates for natural numbers) and how addition and multiplication work (specified by their algorithms).

---

### CHAPTER GOALS

---

Review the basic laws of addition and multiplication, and convince yourself that they are actually true.

---

These laws come up frequently. Many of them show up in non-numerical contexts as well. So you will do yourself a favor by memorizing their names. A few of them will be less familiar to you right now, but they are all important for actually using arithmetic.

The following table summarizes several useful laws of arithmetic. Most of them will be familiar to you.

---

LAWS 1: Basic Laws of Arithmetic

---

For any natural numbers,  $m$ ,  $n$  and  $p$ :

**Associativity**

$$\begin{aligned} m + (n + p) &= (m + n) + p \\ m \cdot (n \cdot p) &= (m \cdot n) \cdot p \end{aligned}$$

**Identity**

$$\begin{aligned} m + 0 &= m = 0 + m \\ m \cdot 1 &= m = 1 \cdot m \end{aligned}$$

**Commutativity**

$$\begin{aligned} m + n &= n + m \\ m \cdot n &= n \cdot m \end{aligned}$$

**Positivity**

$$\text{if } m + n = 0 \text{ then } n = 0$$

**Integrality**

$$\text{if } m \cdot n = 1 \text{ then } n = 1$$

**Cancellativity**

$$\begin{aligned} \text{if } m + p &= n + p \text{ then } m = n \\ \text{if } m \cdot (p^\sim) &= n \cdot (p^\sim) \text{ then } m = n \end{aligned}$$

**Distributivity**

$$m \cdot (n + p) = (m \cdot n) + (m \cdot p)$$

**Case Distinction**

$$\text{if } m \neq 0 \text{ then } m \in \mathbb{N}^+$$

**No (Non-trivial) Zero Divisors**

$$m^\sim \cdot n^\sim \neq 0$$


---



This table is organized to emphasize similarities between addition and multiplication. Pay attention to that.

The Law of Case Distinction was the subject of Chapter 1, Exercise 2. Go back and look at that exercise. Let us look at it again.

To prove that the Law of Case Distinction is true, suppose  $m \neq 0$  and  $m$  has no predecessor. Then clearly,  $m$  is not required by the vocabulary of natural numbers: it is not 0 and it is not the successor of some other (required) natural number. But the Axiom of Induction insists that *every* natural number is required by the vocabulary. So if  $m \neq 0$  and  $m \notin \mathbb{N}^+$ , then  $m$  must not be a natural number. In other words, if  $m$  is a natural number, and  $m \neq 0$ , then  $m$  must have a predecessor. By definition, then means  $m \in \mathbb{N}^+$ .

The Law of Positivity says something similar to Case Distinction. Think of it as asserting that there are no “negative” natural numbers. For it is impossible to add two natural numbers to obtain 0, unless one of them was already 0.

The Law of Integrality is analogous to Positivity, but with respect to multiplication. Think of it as asserting that there are no fractional natural numbers.

Actually, if  $m + n = 0$ , both  $m$  and  $n$  must be 0. Try to sort out why.

## Monoids

The laws of associativity and identity are particularly important because there are many other situations where similar laws hold, but that have nothing to do with arithmetic.

Consider words in the standard latin alphabet. For this purpose, let's suppose that a word is any sequence of letters, not just an actual word of English. So **cat** is a word, but so is **xwpfg**. Now for any two words, we can concatenate them to make a new word. Writing this operation with a new symbol  $+$ , for example, **cat**  $+$  **fish** = **catfish**. Apparently, we could concatenate three or more words:

$$(\mathbf{cat} + \mathbf{fish}) + \mathbf{monger} = \mathbf{catfishmonger}.$$

This indicates we are meant to form **catfish** first, then concatenate **monger**. But we could as well form **fishmonger** first, and concatenate **cat** to that as in

$$\mathbf{cat} + (\mathbf{fish} + \mathbf{monger}) = \mathbf{catfishmonger}.$$

Apparently,  $+$  is an associative operation, just like addition and multiplication.

It makes sense to suppose there is an empty word, but since it is empty, it is not so clear how to write it down. We can use a special (non-latin) symbol to indicate the word with no letters. The most common choice is  $\varepsilon$ . Then it is reasonable to suppose that  $\varepsilon + \mathbf{cat} = \mathbf{cat}$ , and  $\mathbf{cat} + \varepsilon = \mathbf{cat}$ . In fact, this should be true for any word. So  $\varepsilon$  plays exactly the same role as 0 for addition, and 1 for multiplication.

To summarize, concatenation behaves like addition and multiplication, insofar as the following laws hold

$$\begin{array}{lll} m + (n + p) = (m + n) + p & m + 0 = m & 0 + m = m \\ m \cdot (n \cdot p) = (m \cdot n) \cdot p & m \cdot 1 = m & 1 \cdot m = m \\ w + (x + y) = (w + x) + y & w + \varepsilon = w & \varepsilon + w = w \end{array}$$

where  $m$ ,  $n$ , and  $p$  are any natural numbers, and  $w$ ,  $x$  and  $y$  are any words.

Suppose you have a very simple robot that can only do things to change its *position* (its location and the direction it is facing). It only responds to two basic commands: move forward by 1 foot, or rotate 90 degrees clockwise. Of course, these basic commands can be sequenced. So the command “move forward; move forward” will tell it to move two feet forward.

Let us write a sequence of commands “do this, then do that” with a semicolon. So  $c_1 ; c_2$  means execute the command  $c_1$ , and then execute  $c_2$ . Evidently sequencing is associative. (Write out what that

means on the nearest piece of paper). Also, suppose there is a trivial command that says “do nothing.” Let’s write that as  $\bullet$ . So  $\bullet; c = c$  and  $c; \bullet = c$ . The command “do nothing, then do  $c$ ” is the same as “do  $c$ ” and likewise “do  $c$ , and then do nothing” is the same as “do  $c$ .”

So it looks like robot commands act rather like natural numbers with respect to  $+$  and  $0$ , and like words with respect to  $+$  and  $\varepsilon$ .

We can abbreviate the two basic robot commands by the letters  $f$  (move forward) and  $r$  (rotate 90 degrees clockwise). So a compounded command like “move forward, then move forward” may be abbreviated as  $ff$ .

As far as the robot’s position is concerned,  $frfrfrfr$  does exactly the same as  $\bullet$ . Before and after executing that command, the robot will be in the same location and the same orientation as before (though it will have moved around the perimeter of a square). For the purposes of positioning the robot  $frfrfrfr = \bullet$ . So sequencing of robot commands is not exactly the same as word concatenation. But sequencing is associative and has an identity element (do nothing). Notice that sequencing is not commutative. For example,  $f;r$  and  $r;f$  are not the same.

You can use your imagination to think of many other systems where putting things together (similar to addition, multiplication, concatenation, command sequencing) is useful and where the laws of associativity and identity should work.

Mathematicians call such a system a **monoid**. This is an unfortunate name because it makes the idea seem technical and obscure. But really, a monoid is just a system in which it is possible to combine items associatively, and in which there is a “nothing” item that satisfies the identity laws. Monoids are everywhere.

To specify a monoid, we merely need to specify three things:

1. a collection of items that can be combined (for example,  $\mathbb{N}$  or the collection of words or the collection of robot commands)
2. a method of combining them (for example, addition or concatenation or command sequencing),
3. an identity item (for example,  $0$  or  $\varepsilon$  or  $\bullet$ ),

and then ensure that the laws of associativity and identity hold. As a synopsis of this, you can say “ $(\mathbb{N}, +, 0)$  is a monoid.”

So if  $W$  is the collection of all latin alphabet words, then  $(W, +, \varepsilon)$  is a monoid. If  $C$  is the collection of all robot commands, then  $(C, ;, \bullet)$  is a monoid. Also  $(\mathbb{N}, \cdot, 1)$  is a monoid.

In some monoids, the combining operation is also commutative. This is the case for  $(\mathbb{N}, +, 0)$  and  $(\mathbb{N}, \cdot, 1)$ . Then we say the obvious thing: the monoid is a **commutative monoid**.

Even etymologically, “monoid” is peculiar. Typically, the suffix “-oid” indicates something “resembling”, or “having the shape of”, as in *paraboloid*. And the root “mono” indicates singularity. How a monoid fits that is a bit of a mystery. It appears to have been used by some mathematicians by around 1954 to mean a collection with *one* associative operation. But the origins are obscure.

---

#### SYNECDOCHE

---

A common linguistic device is to refer to some whole thing by one of its parts, or vice versa. You might say “I don’t have any wheels” to mean “I don’t have a car.” Political reporters frequently say “Moscow” or “the Kremlin” to mean “the Russian state.” Sometimes it goes the other way. You might refer to “management” when you really just mean “my boss.” The linguistic term for this is **synecdoche** (look it up for

The robot command monoid has another feature that the other examples do not. Every command can be reversed. For example, to undo **r** (rotate 90 degrees clockwise) the command **rrr** does the job because **r ; rrr** = **•**. To undo **f**, instruct the robot to do **rrfrr**.

More generally, for every robot command  $c$  there is a command  $c^{-1}$  satisfying  $c ; c^{-1} = \bullet$  and  $c^{-1} ; c = \bullet$ . Then  $c^{-1}$  is called the **inverse** of  $c$ . A monoid in which every item has an inverse is called a **group**.

The additive monoid  $(\mathbb{N}, +, 0)$  is definitely not a group. There is no natural number  $n$  so that  $2 + n = 0$ , for example. If you want inverses for addition, you know what to do: invent the integers. We will do that later.

Words do not form a group. The sequences of robot commands that we have spelled out do form a group.

As you might guess, the term “group” was coined before “monoid.”

### *Inductive Proofs*

You already know that the Identity Laws for addition hold:  $0 + m = m$  and  $m + 0 = m$  for all natural numbers  $m$ . But how do you know? The addition algorithm says explicitly that  $m + 0 = m$ , but not the other way around. You could try to persuade yourself by saying “addition is commutative, so  $0 + m = m + 0$  and by the definition  $m + 0 = m$ .” But wait. How do you know that addition is commutative? The algorithm for addition does not say anything explicitly about  $m + n = n + m$ .

The Axiom of Induction holds the key to reasoning about situations like this. To prove that  $0 + m = m$  is always true, you could try checking that

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$0 + 2 = 2$$

$$\vdots$$

calculating each instance separately. Good luck with that! After checking the first ten thousand cases, there is still the ten thousand and first. An explicit calculation that  $0 + m = m$  is true for each  $m$  will never end.

On the other hand, if you started this never ending process, you would quickly notice a pattern. Suppose you’ve already checked that  $0 + k = k$  is true for some particular natural number  $k$  (perhaps  $k$  is one million, for example). Then checking that  $0 + k^{\wedge} = k^{\wedge}$  would be easy:

$$0 + k^{\wedge} = (0 + k)^{\wedge} \quad \text{— The addition algorithm}$$



$= k^{\frown}$  — We already calculated that  $0 + k = k$ .

So this never ending process of checking for all natural numbers is actually quite repetitive. To check for one trillion, all we have to do is repeat the same boring little two-line calculation one trillion times! Let's not do that. Instead, we can start with 0, checking easily that  $0 + 0 = 0$ . Then we can let repetition of the two-line calculation work to reach any natural number that can be reached starting from 0 and using successors. But the Axiom of Induction says that every natural number is reachable. So (with a lot of patience) we could actually calculate that  $0 + m = m$  for any particular natural number  $m$ . In short,  $0 + m = m$  is a law that holds for all  $m$ .

A proof employing the Axiom of Induction in this way is called a proof **by simple arithmetic induction**, or just a proof *by induction*, for short. The idea is to prove that any natural number that happens to be reachable from 0 using successor has a desired property. Since the axiom tells us all natural numbers are thus reachable, all natural numbers have the desired property.

---

#### PROOF TACTIC 1: Simple Arithmetic Induction

---

To prove that all natural numbers have a certain property, follow this outline:

**Basis** Prove that 0 has the property.

**Inductive Hypothesis** Assume that  $k$  has the property for some (unspecified)  $k \in \mathbb{N}$ .

**Inductive Step** Prove that  $k^{\frown}$  also has the property, using the inductive hypothesis as needed.

From these steps, the natural numbers that have the desired property constitute a model of Vocabulary 1. So by the Axiom of Induction, all natural numbers have the property.

---

The familiar laws of arithmetic (Laws 1) are provable by induction. You do not have to accept them on anyone's authority. For example, let us start with confirming that addition and 0 make  $\mathbb{N}$  into a monoid.

---

#### TACTICS

---

A technique for proving assertions is a **proof tactic**. These are important because they are widely applicable. You will use them in all your mathematics and a good portion of your computer science studies.

On the first reading, go lightly. But return to a tactic when you see the same idea being used again. This is a good way to learn a transferable skill in general — remind yourself, over time, of how it is used. If you try to “study” it without examples, it can fall out of your head as fast as it falls in. But if you come back to it several times, it will start to stick. Be patient.

---

---

PROPOSITION 1: Addition is associative.

---

For any natural numbers  $m$ ,  $n$  and  $p$ ,

$$m + (n + p) = (m + n) + p.$$

*Proof.* Suppose that  $m$  and  $n$  are arbitrary fixed natural numbers. The goal is to show that (with  $m$  and  $n$  fixed), it is the case that

$$m + (n + p) = (m + n) + p$$

holds for all natural numbers  $p$ . By induction on  $p$ :

**Basis** The goal of the basis is to show that  $m + (n + 0) = (m + n) + 0$ .

But  $m + (n + 0) = m + n = (m + n) + 0$  due to the algorithm for  $+$ .

**Inductive Hypothesis** Assume that  $m + (n + k) = (m + n) + k$  for some  $k \in \mathbb{N}$ .

**Inductive Step** The goal is to show that  $m + (n + k^{\frown}) = (m + n) + k^{\frown}$ .

$$\begin{aligned} m + (n + k^{\frown}) &= m + (n + k)^{\frown} && \text{— Algorithm for } + \\ &= (m + (n + k))^{\frown} && \text{— Same reason} \\ &= ((m + n) + k)^{\frown} && \text{— Inductive Hypothesis} \\ &= (m + n) + k^{\frown} && \text{— Algorithm for } + \end{aligned}$$

Therefore (by the Axiom of Induction),  $m + (n + p) = (m + n) + p$  holds for all natural numbers  $p$ . Since this argument does not depend on any extra assumptions about  $m$  and  $n$ , it holds for all natural numbers  $m$  and  $n$  too.  $\square$

---

A common technique is to suppose that some of the values in the proof are fixed, but otherwise not special in any way. A proof that does not use anything special about them is valid for all. This is another tactic, stated below.

The symbol  $\square$  is a punctuation mark to indicate the end of a proof. It is the mathematician's "drop the mic."

The proof of Proposition 1 employs two other very common proof tactics that are worth calling out.

---

**PROOF TACTIC 2: Use of Generic Values, or Universal Generalization**


---

When the goal is to prove that all values of some kind have a desired property:

1. Suppose  $c$  is some arbitrary, fixed value of the kind you need. The name  $c$  must be “fresh.” It cannot be a name you are already using for something else.
2. Prove that  $c$  has the desired property.
3. Conclude that all values of the same kind as  $c$  have the desired property.

“Arbitrary” means that you are not allowed to assume anything special about  $c$ . For example, to prove something is true of all real numbers, you cannot implicitly suppose that  $c$  is positive. You must assume  $c$  is some real number, *and you don’t know anything else about it*.

The conclusion is justified because there is nothing special about  $c$ . The proof that  $c$  has the desired property is valid for any specific value in place of  $c$ . This is why it is crucial that you do not make any extra assumptions about  $c$ . In effect,  $c$  stands in as a *generic* datum.

---

Another tactic, which may seem so obvious it does not need mentioning, is that a lot of the time we simply need to calculate something. The format we used in the foregoing proof is helpful. Later on we will find uses for the format that go beyond proofs of equalities. So let’s make the basic version official now.

---

**PROOF TACTIC 3: Explicit Calculations**


---

When the goal is to prove that two expressions are equal, chain together a sequence of easily checked steps. The proof of Proposition 1 includes this example:

$$\begin{array}{ll}
 m + (n + k)^{\wedge} = m + (n + k)^{\wedge} & \text{— Algorithm for } + \\
 = (m + (n + k))^{\wedge} & \text{— Same reason} \\
 = ((m + n) + k)^{\wedge} & \text{— Inductive Hypothesis} \\
 = (m + n) + k^{\wedge} & \text{— Algorithm for } +
 \end{array}$$

Pay close attention to the layout. The equality signs are all aligned to make it easier to follow the individual steps. Also each line includes a note explaining *why* the step is justified. Sometimes, when a step is really obvious, the justification is not needed. That is a judgement call. For now, it is better for you to get used to providing justifications for each step.

If the calculation only involves two or three steps and their reasons are obvious, you can just chain things together on one line. For

example, in the basis of the inductive proof above, we wrote the calculation  $m + (n + 0) = m + n = (m + n) + 0$  on a single line.

---

The remainder of this section mainly concerns how to use the tactic of simple arithmetic induction to prove other laws of arithmetic.

---

PROPOSITION 2: 0 is the identity for addition.

---

For any  $m$ ,

$$m + 0 = m \quad \text{and} \quad m = 0 + m.$$

*Proof.* The first equality is true by the definition of  $+$ . A proof of the second equality,  $m = 0 + m$ , was suggested in the introduction to induction. To make it explicit now, the proof is by induction on  $m$ .

**Basis** The goal is to show that  $0 + 0 = 0$ . That is clearly true by the algorithm for addition.

**Inductive hypothesis** Suppose that  $0 + k = k$  for some  $k$ .

**Inductive step** The goal is to prove that  $0 + k^\frown = k^\frown$ .

$$\begin{aligned} 0 + k^\frown &= (0 + k)^\frown && \text{— Algorithm for } + \\ &= k^\frown && \text{— Inductive hypothesis} \end{aligned}$$

Therefore by the Axiom of Induction,  $0 + m = m$  is true for all  $m \in \mathbb{N}$ . □

---

With Propositions 1 and 2, we are justified in claiming that  $(\mathbb{N}, +, 0)$  is a monoid, as we had expected. To prove that addition is commutative, we need another fact about how successor and addition interact.

---

LEMMA 1: Successors migrate in additions.

---

For any natural numbers  $m$  and  $n$ ,

$$m + n^\frown = m^\frown + n.$$

*Proof.* Suppose  $m$  is some fixed natural number. By induction on  $n$ :

**Basis** The goal is to show that  $m + 0^\frown = m^\frown + 0$ .

$$\begin{aligned} m + 0^\frown &= (m + 0)^\frown && \text{— Algorithm for } + \\ &= m^\frown && \text{— Algorithm for } + \\ &= m^\frown + 0 && \text{— Algorithm for } + \end{aligned}$$

Because we are currently discussing natural numbers and nothing else, it is safe to write “for any  $m$ ” instead of “for any natural number  $m$ ”, or “for any  $m \in \mathbb{N}$ .”

So we are using the tactic of Universal Generalization here.

**Inductive Hypothesis** Suppose  $m + k^\frown = m^\frown + k$  for some  $k \in \mathbb{N}$ .

**Inductive Step** The goal is to show that  $m + k^{\frown \frown} = m^\frown + k^\frown$ .

$$\begin{aligned} m + k^{\frown \frown} &= (m + k^\frown)^\frown && \text{— Algorithm for } + \\ &= (m^\frown + k)^\frown && \text{— Inductive Hypothesis} \\ &= m^\frown + k^\frown && \text{— Algorithm for } + \end{aligned}$$

So  $m + n^\frown = m^\frown + n$  for all  $n$ . Because the proof does not depend on any assumption about  $m$ , it is valid for all  $m$  too.  $\square$

Roughly speaking this lemma and the definition of  $+$  permit us to move  $^\frown$  anywhere within an addition:  $m^\frown + n = m + n^\frown = (m + n)^\frown$ . So we are free to move a successor “out of the way” whenever we need to. The proof of commutativity illustrates why this is useful.

**PROPOSITION 3:** Addition is commutative.

For any natural numbers  $m$  and  $n$ ,

$$m + n = n + m.$$

*Proof.* Suppose  $n$  is a fixed natural number. The goal is to show that for any natural number  $m$ ,  $m + n = n + m$ . The proof is by induction on  $m$ .

**Basis** The goal is to show that  $0 + n = n + 0$ . But  $0 + n = n = n + 0$  holds because of Proposition 2 ( $0$  is the identity for addition).

**Inductive Hypothesis** Suppose that  $k + n = n + k$  for some  $k$ .

**Inductive Step** The goal is to show that  $k^\frown + n = n + k^\frown$ .

$$\begin{aligned} k^\frown + n &= k + n^\frown && \text{— Migration of successors (Lemma 1)} \\ &= (k + n)^\frown && \text{— Algorithm for } + \\ &= (n + k)^\frown && \text{— Inductive Hypothesis} \\ &= n + k^\frown && \text{— Algorithm for } + \end{aligned}$$

Therefore,  $m + n = n + m$  for all  $m$ . Because the argument does not depend on any assumptions about  $n$ , it is valid for all  $n$ .  $\square$

Now, of course, we know that  $(\mathbb{N}, +, 0)$  is a commutative monoid.

The next law may be less familiar to you. Roughly, it says that we can “subtract” equals and get equals. But because actual subtraction does not generally make sense for natural numbers ( $5 - 7$  means

nothing without introducing negative numbers), cancellativity is as good as it gets. The proof uses two more tactics involving statements “if  $S$  then  $T$ .” See if you can spot the ideas, made explicit after the proof. The proof also employs two new tactics, discussed after the proof.

---

**PROPOSITION 4:** Addition is right cancellative.

---

*For any natural numbers  $m$ ,  $n$  and  $p$ , if  $m + p = n + p$  then  $m = n$ .*

*Proof.* Suppose  $m$  and  $n$  are some fixed natural numbers. The goal is to show that if  $m + p = n + p$ , then  $m = n$  for all natural numbers  $p$ . The proof is by induction on  $p$ .

**Basis** The goal is to show that if  $m + 0 = n + 0$ , then  $m = n$ . Assume that  $m + 0 = n + 0$ . Then the new goal is to show that  $m = n$ . But immediately because  $0$  is identity and using the assumption,

$$m = m + 0 = n + 0 = n.$$

**Inductive hypothesis** Assume that the following statement is true for some  $k \in \mathbb{N}$ : if  $m + k = n + k$  then  $m = n$ .

**Inductive step** The goal is to show that if  $m + k^\frown = n + k^\frown$  then  $m = n$ . Assume that  $m + k^\frown = n + k^\frown$ , calling this assumption (\*) for reference. Now the goal is to prove that  $m = n$ . But the inductive hypothesis says that if  $m + k = n + k$ , then  $m = n$ . So it is enough to prove  $m + k = n + k$ . Calculating, we have

$$\begin{aligned} (m + k)^\frown &= m + k^\frown && \text{— The addition algorithm} \\ &= n + k^\frown && \text{— Assumption (*)} \\ &= (n + k)^\frown && \text{— The addition algorithm} \end{aligned}$$

Thus  $(m + k)^\frown = (n + k)^\frown$ . Because predecessors are unique (Postulate 2),  $m + k = n + k$ . By the inductive hypothesis,  $m = n$ .

Therefore,  $m + p = n + p$  implies  $m = n$  for all  $p$ . Because the argument does not depend on any assumptions regarding  $m$  and  $n$ , it is valid for all  $m$  and all  $n$ .  $\square$

---

This law is more subtle than the previous ones because it involves a statement of the form “if  $S$  then  $T$ .” That is, Proposition 4 is not claiming that  $m = n$  *always* holds. That would be pretty useless. Instead it is claiming that under some special situations (when  $m + p = n + p$ ), we can conclude  $m = n$ . The proof uses two new tactics, precisely to deal with “if - then” statements.

Addition is also left cancellative — if  $p + m = p + n$ , then  $m = n$ . A proof of that is easy, since we now know that addition is commutative.

Postulate 2 is also an implication: if  $m^\frown = n^\frown$  then  $m = n$ . So we use it by recognizing that in order to prove  $m + k = n + k$  (the actual current goal) it is enough to prove  $(m + k)^\frown = (n + k)^\frown$ .

---

**PROOF TACTIC 4: Direct Proof of an Implication**


---

When the goal is to prove a statement of the form “if  $S$  then  $T$ ”,

1. Assume (temporarily) that the statement  $S$  is true.
2. Make  $T$  a new goal.
3. Prove  $T$  using the assumption  $S$  as needed.
4. Conclude that “if  $S$  then  $T$ ” is true without the assumption of  $S$ .

This conclusion is justified because it is the meaning of “if - then.” That is, *if*  $S$  happens to be true, you are able to prove that  $T$  must also be true. You do not need to know whether  $S$  is true in general. In situations where  $S$  is true, so is  $T$ . A proof like this is called a **direct** proof of “if  $S$  then  $T$ .”

Some alternative phrases that mean the same as “if  $S$  then  $T$ ” are

- “ $S$  implies  $T$ .”
  - “ $S$  only if  $T$ .”
  - “ $T$  follows from  $S$ .”
- 

The other tactic has to do with *using* an implication rather than proving one.

---

**PROOF TACTIC 5: Modus Ponens**


---

When the goal is to prove a statement  $T$ .

1. Find some other related statement  $S$  for which you know that  $S$  implies  $T$ .
2. Set  $S$  as a new goal, and prove it.
3. Conclude that  $T$  is true.

This conclusion is justified because  $S$  is true, and  $S$  implies  $T$ .

---

In the proof of Proposition 4, modus ponens allows us to use the inductive hypothesis “if  $m + k = n + k$  then  $m = n$ ” in proving the goal  $m = n$ . Namely, it allows us to set a new goal to prove  $m + k = n + k$ , and know that this will suffice.

To round out our discussion of the monoid  $(\mathbb{N}, +, 0)$ , we also want to know that it does not include any “accidentally” negative numbers.

The Latin name term **modus ponens** means “method that affirms.”

Modus ponens can be used in other ways. I have described the most common use here.

---

**PROPOSITION 5: The Law of Positivity**


---

*For any natural numbers  $m$  and  $n$ , if  $m + n = 0$ , then  $n = 0$ .*

*Proof.* Suppose  $n \neq 0$ . Then by Case Distinction,  $n$  is a successor. That is,  $n = k^{\wedge}$  for some natural number  $k$ . So  $m + n = m + k^{\wedge} = (m + k)^{\wedge}$ . Hence,  $m + n$  is not equal to 0, by Postulate 1.  $\square$

---

Notice that this proof does not explicitly deal with the stated “if - then” claim. Instead it takes an important indirect approach.

---

**PROOF TACTIC 6: Proof by Contraposition**


---

When the goal is to prove a statement “if  $S$  then  $T$ .”

- Assume that the statement  $T$  is false.
- Set the goal to proving that  $S$  is also false, using the assumption that  $T$  is false as needed.
- Conclude that “if  $S$  then  $T$ ” is true.

The conclusion is justified, because what you have actually shown is that it is impossible to have  $T$  false, without also having  $S$  false. So if  $S$  is true, it is impossible for  $T$  to be false. This is an indirect proof of  $T$ .

A proof like this is called a proof **by contraposition**. It works because a statement “if  $S$  then  $T$ ” is equivalent to the statement “if  $T$  is false then  $S$  is false.” The latter is called the **contrapositive** of “if  $S$  then  $T$ .”

---

To prove analogous facts about multiplication, particularly that it is commutative, we need the following facts (analogous, in some sense, to Proposition 2 and Lemma 1).

Notice that, implicitly, we used Proof Tactic 2, Universal Generalization. To make the tactic explicit, the proof would start out with “Suppose  $m$  and  $n$  are arbitrary natural numbers” and conclude with “Because  $m$  and  $n$  are arbitrary, the proof is valid for all  $m$  and all  $n$ .” We are so used to deploying this tactic that we usually leave it up to the reader.



---

PROPOSITION 6: 0 is an “annihilator”

---

For any natural number  $n$ ,

$$0 \cdot n = 0 = n \cdot 0.$$

*Proof.* The algorithm for multiplication directly specifies that  $n \cdot 0 = 0$ . The proof that  $0 \cdot n = n$  is true for all  $n \in \mathbb{N}$  is by induction on  $n$ .

**Basis** The goal is to show that  $0 \cdot 0 = 0$ . But this is directly from the algorithm for multiplication.

**Inductive Hypothesis** Assume that  $0 \cdot k = 0$  for some  $k \in \mathbb{N}$ .

**Inductive Step** The goal is to show that  $0 \cdot k^\frown = 0$ .

$$\begin{aligned} 0 \cdot k^\frown &= 0 + 0 \cdot k && \text{— Algorithm for } \cdot \\ &= 0 + 0 && \text{— Inductive Hypothesis} \\ &= 0 && \text{— Additive Identity} \quad \square \end{aligned}$$

---

The next fact is a technical property that we will use to prove commutativity of multiplication, which we leave as an exercise.

---

LEMMA 2: Successors migrate in products.

---

For any natural numbers  $m$  and  $n$ ,

$$m^\frown \cdot n = m \cdot n + n.$$

*Proof.* Suppose  $m \in \mathbb{N}$  is fixed. The proof is by induction on  $n$ .

**Basis** The goal is to show that  $m^\frown \cdot 0 = m \cdot 0 + 0$ . Clearly

$$m^\frown \cdot 0 = 0 = 0 + 0 = m \cdot 0 + 0$$

all follow from facts we know about addition and multiplication.

**Inductive Hypothesis** Suppose that  $m^\frown \cdot k = m \cdot k + k$  for some  $k \in \mathbb{N}$ .

**Inductive Step** The goal is to show that  $m^\frown \cdot k^\frown = m \cdot k^\frown + k^\frown$ .

$$\begin{aligned} m^\frown \cdot k^\frown &= m^\frown + m^\frown \cdot k && \text{— Exercise} \\ &= m^\frown + (m \cdot k + k) \\ &= m + (m \cdot k + k)^\frown \\ &= m + (m \cdot k + k^\frown) \\ &= (m + m \cdot k) + k^\frown \\ &= m \cdot k^\frown + k^\frown \end{aligned}$$

By now, you should be used to seeing a formulaic statement such as “Therefore, by the induction axiom,  $0 \cdot n = 0$  for all  $n \in \mathbb{N}$ ”. This is so predictable that most of the time we can omit it.

Therefore,  $m^{\wedge} \cdot n = m \cdot n + n$  for all  $n \in \mathbb{N}$ . Because the proof does not depend on assumptions about  $m$ , it is valid for all  $m \in \mathbb{N}$ .  $\square$

For the record, we skip (even in these exercises) the proof of cancellativity for multiplication by a positive natural number, because it is simpler to prove once we have some other techniques. We return to it in Chapter 3.



#### EXERCISES:

9. Write out the entire proof of Lemma 2 providing the justifications for each line of the calculation in the inductive step.
10. Prove that multiplication is commutative. [Hint: Use Proposition 6 and Lemma 2 that we proved right before these exercises.]
11. Prove that 1 is the identity for multiplication. That is,  $1 \cdot m = m$  and  $m = m \cdot 1$ . In your proof, explicitly identify which tactics are used.
12. Prove that multiplication distributes over addition —  $m \cdot (n + p) = m \cdot n + m \cdot p$  — by induction on  $p$ . You can use any of the lemmas and propositions we have already proved.
  - (a) Prove the basis:  $m \cdot (n + 0) = m \cdot n + m \cdot 0$ .
  - (b) Write the inductive hypothesis.
  - (c) Prove the inductive step:  $m \cdot (n + k^{\wedge}) = m \cdot n + m \cdot k^{\wedge}$
13. Prove that multiplication is associative —  $m \cdot (n \cdot p) = (m \cdot n) \cdot p$  — by induction on  $p$ .
  - (a) Prove the basis:  $m \cdot (n \cdot 0) = (m \cdot n) \cdot 0$ .
  - (b) Write out the inductive hypothesis.
  - (c) Prove the inductive step:  $m \cdot (n \cdot k^{\wedge}) = (m \cdot n) \cdot k^{\wedge}$ . [Hint: Use the Law of Distribution, which you just proved.]
14. Prove the Law of Integrality: if  $m \cdot n = 1$ , then  $n = 1$ . [Hint: Use a proof by contraposition. Suppose  $n \neq 1$ . Show that  $m \cdot n \neq 1$ . Using Case Distinction,  $n \neq 1$  means that either  $n = 0$  or  $n = k^{\wedge\wedge}$  for some  $k$ . No explicit induction is needed.]
15. Prove the Law of No Zero Divisors. [Hint:  $m^{\wedge} \cdot n^{\wedge} = m^{\wedge} + m^{\wedge} \cdot n$  by definition of multiplication. No explicit induction is needed.]

A surprise here is that the proof of associativity of multiplication needs the law of distributivity. The proof that addition is associative was considerably simpler.

### 3

## Orderings of the Natural Numbers

*Good order is the foundation of all great things.*

— Edmund Burke

THE STANDARD ORDERING of natural numbers  $m \leq n$  does not seem to need much explanation. But actually it arises from addition because “ $m \leq n$ ” means “something can be added to  $m$  to reach  $n$ .” So reasoning about the order of natural numbers is precisely reasoning about addition (something we know a lot about now). This suggests useful analogous definitions in terms of multiplication and other operations. The analogies save a lot of effort because many of the most useful facts about  $\leq$  transfer to these other definitions with almost no extra work.

Consider some ways we can read “ $5 \leq 7$ ” out loud:

- Five is smaller than seven
- Five is below seven
- Five precedes seven
- Five is included in seven
- Five is part of seven

Wait. Included? Those sounds strange. But think of it this way. If you walk into a hardware store and see five hammers hanging on the wall, do you think the store has five hammers in stock? No. You only know that the five you see are included in the stock. They might have more hammers in the back of the store.

What about “part of”? Using arithmetic, “five is part of seven” makes sense if you think that  $5 + 2 = 7$  means 5 and 2 are “parts” that combine to make 7. This informal bit of linguistics suggests that “smaller than”, “included in” and “part of” are in some way conceptually related.

---

#### CHAPTER GOALS

---

We look at the familiar order of natural numbers, and derive some useful facts. By an analogy between addition and multiplication, we also discover a second, useful order on natural numbers that corresponds to divisibility.

---

Now suppose you meet up with a Martian mathematician named Glip (it could happen). Glip is justifiably proud of the strong Martian tradition in discrete mathematics, but sometimes English is not Glip's best friend. That's understandable. Anyway, Glip tells you that "four is part of twelve". But also tells you, "and by the way, four is not part of thirteen." You immediately recognize that "part of" cannot mean  $\leq$  as far as Glip is concerned. So you ask a few other questions:

- Is five part of ten? "Of course."
- Is five part of twelve? "Obviously not"
- Is seven part of forty? "Grppt-ng" (which means something rude in Martian)
- Is seven part of thirty-five? "Yes. That's better"

You probably now have a pretty good guess what "part of" means to a Martian mathematician. Evidently, five is part of ten because  $5 \cdot 2 = 10$ . Likewise, seven is part of thirty-five because  $7 \cdot 5 = 35$ . For Glip, "part of" means that one number is a factor of the other. If you think about it, that is just as reasonable as saying that five is part of seven *because*  $5 + 2 = 7$ . Martians evidently think of "parts" of natural numbers differently that we do, but their way is not unreasonable. In this chapter, we concentrate on two orders on natural numbers (Earthling and Martian), and begin an investigation that continues in subsequent chapters regarding how they are related. The first order is the standard one where  $5 \leq 7$  means just what you expect. The second is based on factoring. Of course, this is not really "Martian" ordering because we just made up the whole Glip story, and there are not really Martian mathematicians (though we have occasionally wondered about some of our colleagues). It turns out this other order is called *divisibility*. It has many earthly uses.

### *The Standard Order*

To make the familiar order of natural numbers clear, we can start with the following definition.

---

**DEFINITION 2:** The Standard Order of Natural Numbers
 

---

For natural numbers  $m$  and  $n$ ,  $m$  is **less than or equal to**  $n$ , written  $m \leq n$ , if and only if there is some natural number  $d$  satisfying  $m + d = n$ .

---

For example,  $4 \leq 9$  because  $4 + 5 = 9$ ,  $1 \leq 2$  because  $1 + 1 = 2$ , and  $6 \leq 12015$  because  $6 + 12009 = 12015$ . On the other hand  $5 \not\leq 3$  because there is no natural number  $d$  for which  $5 + d = 3$ .

From the definition, we can immediately infer some useful properties of  $\leq$ .

---

**PROPOSITION 7:** The natural numbers are partially ordered by  $\leq$ .
 

---

The following three properties hold for all natural numbers  $m$ ,  $n$  and  $p$ .

*Reflexivity*             $m \leq m$ ,

*Transitivity*        if  $m \leq n$  and  $n \leq p$ , then  $m \leq p$ ,

*Anti-symmetry*     if  $m \leq n$  and  $n \leq m$ , then  $m = n$ .

*Proof.* Reflexivity holds because of the identity law:  $m + 0 = m$  for any  $m$ .

Transitivity holds because of associativity of addition. Assume that  $m \leq n$  and  $n \leq p$ . The goal then is to prove that  $m \leq p$ . By definition of  $\leq$ , the goal is really to find some natural number  $d$  so that  $m + d = p$ .

By definition of  $\leq$ , the assumptions mean that  $m + e = n$  and  $n + f = p$  for some natural numbers  $e$  and  $f$ . Now let  $d = e + f$ . Then  $m + d = m + (e + f) = (m + e) + f = n + f = p$  by associativity and the assumptions.

Anti-symmetry, a bit more complicated, holds because of identity, associativity, cancellativity and positivity. Assume that  $m \leq n$  and  $n \leq m$ . The goal is to prove that  $m = n$ . By definition of  $\leq$ , it is the case that  $m + e = n$  and  $n + f = m$  for some  $e$  and  $f$ . Hence

$$m + 0 = m = n + f = (m + e) + f = m + (e + f).$$

Cancelling  $m$ , we obtain  $0 = e + f$ . By the Law of Positivity,  $f = 0$ . So  $n = m + f = m$ . □

---

Definitions play a very important role in mathematics. You should treat the defined term as being entirely interchangeable with its definition. So “ $5 \leq 8$ ” really means exactly that “there is some natural number  $d$  satisfying  $5 + d = 8$ .” Nothing more, nothing less.



How can we be so sure that there is no natural number  $d$  for which  $5 + d = 3$ ?

**Reflexivity, transitivity, and anti-symmetry** — the standard names for these three properties — show up in many other situations that are not related to  $\leq$ .

A relation like  $\leq$  that is reflexive, transitive and anti-symmetric is called a **partial order**. A relation that is reflexive and transitive, but possibly not antisymmetric, is called a **pre-order**.

There is no need to know exactly what  $e$  and  $f$  are. All we need is to know that some such numbers exist that satisfy the equations.

Taking a bit of inventory here, notice that reflexivity and transitivity of  $\leq$  only depend on  $(\mathbb{N}, +, 0)$  being a monoid. So automatically, when you encounter some other monoid, you know you will get some sort of “less or equal” relation, defined by analogy with  $\leq$ . It will be guaranteed to be reflexive and transitive. You will see an example in the next section.

The proof of anti-symmetry of  $\leq$  only depends on the monoid  $(\mathbb{N}, +, 0)$  being cancellative and positive. So again, any monoid with those properties has an anti-symmetric “less than or equal” relation.

The proof of Proposition 7 involves the definition of  $\leq$  in two ways because  $\leq$  figures in both the premise and the conclusion of an “if - then” condition. By definition of  $\leq$ , a proof involving this relation really involves statements of the form “for some...”. So tactics for dealing with  $\leq$  are more generally going to be tactics for dealing with “for some...” statements.

Partial order relations arise in many situations that have anything to do with a monoid operation. Consider the relation between people, “P is an ancestor of Q.” Let’s agree (a bit artificially) that every person is a “self-ancestor.” Then “is an ancestor of” is a reflexive, transitive relation. And what we know of biology and time travel suggests that if P is ancestor of Q and Q is ancestor of P, then P and Q are the same person. So ancestorship is also anti-symmetric. But there is no monoid operation on people that gives rise to this relation.

---

#### PROOF TACTIC 7: Existential Instantiation: Calling a Witness

---

To use an assertion that some property holds *for some thing*, you can suppose that such a thing actually exists. That is, you can suppose that  $w$  is some particular object for which the assertion is true. The name  $w$  must be *fresh* — a name you are not already using for some other purpose. And you must take care not to assume any extra properties of  $w$ . This tactic, called **existential instantiation**, is outlined as follows.

When the goal is to prove a statement  $U$ , and a statement “for some  $x$ , it is the case that  $P$  holds” is in the context, do as follows.

- Suppose  $w$  has the property  $P$  — insuring that you’ve chosen a fresh identifier  $w$ .
- Prove  $U$  using  $w$  and the supposition that  $w$  has the property  $P$ .
- Conclude  $U$ .

The conclusion is justified because if there is some  $w$  satisfying  $P$ , and in any situation where such a  $w$  exists,  $U$  also holds, then  $U$  holds. The object  $w$  that is supposed to have the property  $P$  is often called a **witness for  $P$** .

---

This tactic is used to prove that  $\leq$  is transitive and that it is anti-symmetric. The sentence “By definition of  $\leq$ ,  $m + e = n$  and  $n + f = p$  for some natural numbers  $e$  and  $f$ ” reiterates the definition of  $\leq$ . But in the context of the proof, it also introduces fresh names “ $e$ ” and “ $f$ ” for the witnesses.

---

**PROOF TACTIC 8:** Existential Generalization: Finding a Witness
 

---

To prove that a property holds for some value, all you need to do is find a suitable value.

For example, in order to prove that  $m \leq n$ , we merely need to note that this means  $m + d = n$  for some  $d$ , and then produce a suitable value for  $d$ . A proof that  $7 \leq 13$  consists in producing the number 6. So 6 *witnesses* the fact that  $7 \leq 13$ .

When the goal is to prove that there exists some value for which a property  $P$  holds, do the following.

- Describe a specific potential witness  $w$ .
- Prove that  $w$  has the desired property  $P$ .
- Conclude that there exists some value for which  $P$  holds.

Clearly, the conclusion is justified by the plain meaning of a sentence declaring “there is some value for which  $P$  holds.” If a particular value  $w$  has property  $P$ , then *some* value has property  $P$ .

---

Existential generalization is no more mysterious than it seems. It really just says that knowing “Mr. Waffles is a dog” means you also know “Something is a dog.” The tactic is used, for example, to prove transitivity of  $\leq$ . The conclusion is that  $m \leq p$  *because*  $e + f$  is a witness. That is,  $m + (e + f) = p$ .

### *Linearity of the Standard Order*

The “stepping stone” picture of the natural numbers suggests another stronger property, that the natural numbers line up.

---

**PROPOSITION 8:** The natural numbers are linearly ordered by  $\leq$ .
 

---

For any natural numbers  $m$  and  $n$ , either  $m \leq n$  or  $n \leq m$ .

*Proof.* Linearity holds by induction on  $m$ . The inductive goal is to prove that for every  $m$ , it is the case that for every  $n$ , either  $m \leq n$  or  $n \leq m$ .

**Basis** Suppose  $n$  is an arbitrary natural number.  $0 \leq n$  is true because  $0 + n = n$ . Hence the assertion “ $0 \leq n$  or  $n \leq 0$ ” is true for all  $n$ .

**Inductive Hypothesis** Suppose that for all  $n$ , it is the case that  $k \leq n$  or  $n \leq k$ .

This proof involves several of the tactics we have discussed so far. Try to identify them. It also employs another two tactics that you have not seen yet.

**Inductive Step** The goal is to show that for all  $n$ , either  $k^< \leq n$  or  $n \leq k^<$ .

Suppose  $n$  is an arbitrary natural number. By the inductive hypothesis, either  $n \leq k$  or  $k \leq n$ . So the goal breaks down to two subgoals:

1. show that  $n \leq k$  implies that either  $k^< \leq n$  or  $n \leq k^<$ , and
2. show that  $k \leq n$  implies that either  $k^< \leq n$  or  $n \leq k^<$ .

To prove 1, suppose  $n + d = k$  holds for some  $d$ . Then  $n + d^< = k^<$ , and so  $n \leq k^<$ . To prove 2, suppose  $k + d = n$  for some  $d$ . By Case Distinction, either  $d = 0$  or  $d$  is positive. If  $d = 0$ , then  $k = n$ . So  $n \leq k^<$ . If  $d$  is positive, then  $d = \text{pred}(d)^<$ . So

$$\begin{aligned} k^< + \text{pred}(d) &= k + \text{pred}(d)^< && \text{--- Lemma 1} \\ &= k + d && \text{--- Definition of pred} \\ &= n && \text{--- Assumption.} \end{aligned}$$

So  $k^< \leq n$ . □

The property of linearity, unlike other statements so far, asserts an “or” condition: either  $m \leq n$  or  $n \leq m$ . The first tactic associated with “or” assertions is almost trivial.

#### PROOF TACTIC 9: Disjunction Introduction

When the goal is to prove a statement “ $S$  or  $T$ ”, do the following.

1. Either
  - (a) prove  $S$ , or
  - (b) prove  $T$ .
2. Conclude that “ $S$  or  $T$ ” is true.

The conclusion is justified by the plain meaning of the word “or.”

The easiest to spot example in Proposition 8 is in the basis of the induction: The goal is to prove that “ $0 \leq n$  or  $n \leq 0$ ” is true. But “ $0 \leq n$ ” is enough.

Look for other examples in the proof of the proposition.

#### PROOF TACTIC 10: Disjunction Elimination

When the goal is to prove some statement  $U$  in a context where “ $S$  or  $T$ ” holds, do the following:



1. Assume  $S$ , and prove  $U$ .
2. Assume  $T$ , and prove  $U$ .
3. Conclude  $U$  without either assumption.

The conclusion is justified because “ $S$  or  $T$ ” means that at least one of the two is true. So if both of them entail  $U$ , then  $U$  is inescapable.

A special case uses knowing “ $S$  or  $T$ ” to prove  $T$  by proving that  $S$  is false. That is, if “ $S$  or  $T$ ” is known, then one or the other of  $S$  and  $T$  must be true. So if  $S$  is false,  $T$  is forced to be true.

The Latin name for this version is *modus tollendo ponens*, meaning the “method that, by denying, affirms.” By denying  $S$ , it affirms  $T$ .

---

The following proposition completes the “stepping stone” picture of the natural numbers as a linear order whose first element is the number 0, and so that for every number  $n$  there is another number that comes immediately after  $n$ .

---

PROPOSITION 9:  $\leq$  is a discrete order with bottom and no top elements.

---

For all natural numbers  $m$  and  $n$ ,

1.  $0 \leq n$ ,
2.  $n \leq n^\frown$ ,
3.  $m \leq n^\frown$  if and only if  $m \leq n$  or  $m = n^\frown$ ,
4.  $n \leq m \leq n^\frown$  if and only if  $m = n$  or  $m = n^\frown$ .

*Proof.* 1. We have seen in a previous chapter that  $0 + n = n$ , for every natural number  $n$ . Thus, by the definition of  $\leq$ , we obtain that  $0 \leq n$ .

2. For every natural number  $n$ , we have the equalities  $n + 1 = n + 0^\frown = (n + 0)^\frown = n^\frown$ . Hence,  $n \leq n^\frown$ .
3. If  $m \leq n^\frown$  then there is a number  $d$  so that  $m + d = n^\frown$ . If  $d = 0$  then  $m = m + 0 = m + d = n^\frown$ . Otherwise,  $d$  is positive. So, there is a number  $e$  so that  $d = e^\frown$ . Thus,

$$(m + e)^\frown = m + e^\frown = m + d = n^\frown,$$

whence we deduce that  $m + e = n$ , by Postulate 2. That is to say,  $m \leq n$ . This proves that if  $m \leq n^\frown$  then, either  $m \leq n$  or  $m = n^\frown$ .

In order to prove the reverse implication, suppose now that  $m \leq n$  or  $m = n^\frown$ . In the case  $m \leq n$ , notice that since by part 2 we also have that  $n \leq n^\frown$ , we deduce that  $m \leq n^\frown$ , by the transitivity of  $\leq$ . Finally, in the case  $m = n^\frown$ , we also have that  $m = n^\frown \leq n^\frown$ , by the reflexivity of  $\leq$ .

4. We have already seen that  $n \leq n$  and  $n \leq n^\frown$ . Therefore, if either  $m = n$  or  $m = n^\frown$  holds, then  $n \leq m$  also holds.

In order to prove the other implication, suppose that  $n \leq m$  and  $m \leq n^\frown$ . By part 3, either  $m \leq n$ , and therefore  $m = n$ , by the anti-symmetry of  $\leq$ , or  $m = n^\frown$ , as we wanted to see.  $\square$

Part 1 of the previous proposition tells us that 0 is the smallest natural number, according to the standard order. Part 2 shows that there is no largest natural number, since for every natural number  $n$  there is another different one, namely  $n^\frown$ , strictly larger than  $n$ . Part 4 shows that there is no number between  $n$  and  $n^\frown$ . In other words, the number that comes immediately after  $n$  in the standard ordering of natural numbers is  $n^\frown$ . This finally justifies why we call  $n^\frown$  the *successor* of  $n$ . Part 3 is there mainly as a step to proving 4.

In parts 3 and 4 of the previous proposition, we have to prove that two statements are *equivalent* according to the following tactic.

#### PROOF TACTIC 11: Equivalences

An equivalence is a statement of the form “S if and only if T.” Its meaning is that “S implies T” and also that “T implies S.” Hence, a simple strategy to prove an equivalence is to prove these two implications independently. For that, we can use any method, such as Tactics 4 or 6, that we have already learned for implications.

We will show now a very important property of the standard order.

#### PROPOSITION 10: The standard order is a well order.

*Every nonempty collection of natural numbers contains a least one, that is, an element  $m$  in the collection so that for every other element  $n$  of the collection,  $m \leq n$ .*

*Proof.* Suppose that  $A$  is a collection of natural numbers without a least element and consider the collection  $B$  of all the natural numbers  $n$  so that if  $p \leq n$  then  $p$  is not in  $A$ . We will prove that  $B = \mathbb{N}$ .

**Basis** 0 belongs to  $B$ , because there is only one number  $p \leq 0$ , namely  $p = 0$ , and if 0 were an element of  $A$ , then it would be the least element of  $A$ .

**Inductive Hypothesis** Suppose that  $k$  is an element of  $B$ , that is, for all  $p \leq k$ ,  $p$  is not an element of  $A$ .

**Inductive Step** We want to show that  $k^\frown$  is an element of  $B$ . Suppose then that  $p \leq k^\frown$ , and let's prove that  $p$  is not in  $A$ .

By Proposition 9.3, either  $p \leq k$  or  $p = k^\frown$ . If  $p \leq k$ , then  $p$  is not in  $A$ , by the inductive hypothesis, and therefore  $p = k^\frown$ . Hence, we have to prove that  $k^\frown$  is not an element of  $A$  either.

Notice that, by the linearity of the standard order, for every element  $r$  of  $A$ , either  $r \leq k^\frown$  or  $k^\frown \leq r$ . Using Proposition 9.3, this is equivalent to  $r \leq k$ , or  $r = k^\frown$ , or  $k^\frown \leq r$ , what can be also expressed as  $r \leq k$  or  $k^\frown \leq r$ . By the inductive hypothesis,  $r \leq k$  cannot be true, since we are assuming  $r$  to be an element of  $A$ . Hence, we deduce that  $k^\frown \leq r$  for every element  $r$  of  $A$ . If  $k^\frown$  were an element of  $A$ , it would be the least element of  $A$ , against our hypothesis that  $A$  doesn't have a least element. Therefore,  $k^\frown$  is not in  $A$ , as we wanted to show.

We have proven then that  $B = \mathbb{N}$ , since it contains 0 and it also contains the successor  $k^\frown$  of every element  $k$  in  $B$ . Thus, for even number  $n$  and for every  $p \leq n$ , we have that  $p$  is not an element of  $A$ . In particular,  $n$  is not an element of  $A$ . Hence,  $A$  is an empty collection.  $\square$

The standard ordering  $\leq$  of natural numbers has a related relation called a *strict ordering* and denoted  $<$ .

#### DEFINITION 3:

For natural numbers  $m$  and  $n$ ,  $m$  is **strictly less than**  $n$ , written  $m < n$ , if and only if  $m \leq n$  but  $m \neq n$ .

As you can naturally expect from their names, the two relations of natural numbers  $\leq$  and  $<$  are interconnected. But actually, they are related in more than one (more or less obvious) way.

LEMMA 3: The relations *less than or equal* and *less than* are comparable.

For natural numbers  $m$  and  $n$ , we have the following equivalences.

1.  $m < n$  if and only if there is a positive  $d$  so that  $m + d = n$ .
2.  $m < n$  if and only if  $m^\frown \leq n$ .
3.  $m < n$  if and only if  $m \leq \text{pred}(n)$ .

*Proof.* This is an exercise. □

---

In the previous lemma we are giving three different properties that are equivalent to “ $m < n$ .” Notice that none of them is the definition of “ $m < n$ ” as we have given it. So we call these properties *characterizations* of the relation  $<$ . The linearity of  $\leq$  takes the following form for the strict order.

---

#### PROPOSITION 11: Trichotomy

---

For natural numbers  $m$  and  $n$ , one and only one of the following conditions holds:

$$m < n, \text{ or } m = n, \text{ or } n < m.$$

*Proof.* This is an immediate consequence of Proposition 8 and the fact that  $m \leq n$  if and only if  $m < n$  or  $m = n$ . □

---



#### EXERCISES:

---

16. Prove the following characterizations of the strict order  $<$  of natural numbers: For all natural numbers  $m$  and  $n$ ,
    - (a)  $m < n$  if and only if there is a *positive*  $d$  so that  $m + d = n$ ,
    - (b)  $m < n$  if and only if  $m^{\sim} \leq n$ ,
    - (c)  $m < n$  if and only if  $m \leq \text{pred}(n)$ .
- 

### Divisibility

To prove that  $\leq$  is reflexive, transitive, and anti-symmetric, all we needed to know is that addition is associative, has an identity, is cancellative and is positive. All of these laws of addition have exactly analogous laws of multiplication for positive natural numbers. So, without even thinking much about it, there is another order on positive natural numbers defined by multiplication. In other words, we just replace addition with multiplication and see what happens. Surely that should be useful somehow, or at least entertaining.

---

**DEFINITION 4:** Divisibility for positive numbers

---

For two positive natural numbers, say that  $m$  **divides**  $n$ , written  $m \mid n$ , provided that  $m \cdot q = n$  for some  $q \in \mathbb{N}^+$ .

---

Because 0 plays a special role in multiplication, we exclude it from this definition. Later in this section, we will look at what it takes to deal with 0.


---

**EXAMPLE 3:**

---

Divisibility is just what you are expecting. For example, 5 divides 15 because  $5 \cdot 3 = 15$  and 911 divides 689627 because  $911 \cdot 757 = 689627$ . And 5 does not divide 16 because there is no natural number  $q$  satisfying  $5 \cdot q = 16$ .

---

 Why can we be certain that there is no natural number  $q$  so that  $5 \cdot q = 16$ ?

It might seem a little strange at the beginning, but we could think of “ $m \mid n$ ” because “ $m \mid n$ ” and “ $m \leq n$ ” are essentially different. But, there are many ways in which they are analogous. Indeed, the positive numbers are ordered by  $\mid$  in a “nonstandard” way. The next lemma formalizes this statement.

---

---

**LEMMA 4:** The positive natural numbers are partially ordered by divisibility.

---

For all positive natural numbers  $m$ ,  $n$  and  $p$ , the following are true.

*Reflexivity*             $m \mid m$ ,

*Transitivity*        if  $m \mid n$  and  $n \mid p$ , then  $m \mid p$ ,

*Anti-symmetry*     if  $m \mid n$  and  $n \mid m$ , then  $m = n$ .

*Proof.* Proof is left as an exercise. □

---

The proof of this lemma follows, step by step, the proof of Proposition 7. For example, the reflexivity of  $\leq$  is a consequence of the identity law for addition, and the reflexivity of  $\mid$  is a consequence of the identity law for multiplication. The transitivity of  $\leq$  is a consequence of the associativity of addition, while the transitivity of multiplication is a consequence of the associativity of multiplication. As for anti-symmetry, recall that, in the proof of Proposition 7, we used the cancellativity of addition. Analogously, you will need the cancellativity of the product by a positive number in order to proof the anti-symmetry of divisibility. The following lemma closes a gap left in Chapter 2.

---

**PROPOSITION 12:** Multiplication by a positive natural number is cancellative.

---

For all natural numbers  $m, n$ , and every positive natural number  $p$ ,

$$\text{if } m \cdot p = n \cdot p \text{ then } m = n.$$

*Proof.* Suppose  $m$  and  $n$  are natural numbers, and  $p$  is a positive natural number. By definition,  $p = q^\wedge$  for some natural number  $q$ . The goal is to prove that if  $m \cdot p = n \cdot p$  then  $m = n$ . We could proceed by induction on  $q$ , but let's see another way that avoids explicit induction by re-using facts we already know.

Suppose  $m \cdot p = n \cdot p$ . Because of linear ordering of the natural numbers (Proposition 8), either  $m \leq n$  or  $n \leq m$ . So if we can prove that  $m \leq n$  entails  $m = n$ , and that  $n \leq m$  entails  $m = n$ , then by disjunction elimination (Proof Tactic 10), we will be justified in concluding that  $m = n$ .

Suppose  $m \leq n$ . Then  $m + d = n$  for some natural number  $d$ . By Case Distinction,  $d$  is either 0 or  $d$  is positive. If  $d$  is positive, then

$$n \cdot p + 0 = n \cdot p = (m + d) \cdot p = m \cdot p + d \cdot p = n \cdot p + d \cdot p.$$

By cancellativity of addition,  $0 = d \cdot p$ . But we supposed that both  $d$  and  $p$  are positive, so this is impossible. In other words,  $d$  can not be positive. So it must equal 0. Hence  $n = m$ .

The proof that  $n \leq m$  entails  $m = n$  is identical, except for swapping the roles of  $m$  and  $n$ . □

---

Let us see how far the analogy with the standard order takes us. For instance, recall that  $0 \leq m$  for every natural number  $m$  because  $0 + m = m$ . Analogously,  $1 \mid m$  holds for every natural number  $m$  because  $1 \cdot m = m$ . This means that there is a sense in which 1 is the "least" positive natural number.



Is there a "largest" natural number, according to the ordering given by divisibility?




---

### EXERCISES:

---

In each of these problems, you just need to think carefully about the analogous proofs for  $\leq$  and how the relevant properties of addition have counterparts for multiplication.

17. Prove that divisibility is reflexive.
18. Prove that divisibility is transitive.
19. Prove that divisibility is anti-symmetric. That is, prove that for  $m > 0$  and  $n > 0$ , if  $m \mid n$  and  $n \mid m$ , then  $m = n$ . [Hint: Remember Proposition 12.]

20. Prove that divisibility is not linear by finding two natural numbers  $m$  and  $n$  for which  $m \nmid n$  and  $n \nmid m$ .
21. Prove that for any positive natural numbers  $m$  and  $n$ , if  $m \mid n$  then  $m \leq n$ .

The next exercise is somewhat speculative, investigating yet another relation on natural numbers defined in terms of another operation.

22. Let's define yet another relation between natural numbers, based on exponentiation. Say that  $m$  **is a base of**  $n$  provided that  $m^e = n$  for some natural number  $e$ . So for example, 5 is a base of 25 because it is  $5^2$ .
- (a) Prove that this relation is reflexive:  $m$  is a base of  $m$  for any natural number  $m$ .
- (b) Prove that this relation is transitive. [Warning: your proof will not be exactly analogous to the proofs that  $\leq$  and  $\mid$  are transitive.]
- (c) Prove that every  $m$  is a base of 1. [Hint: This exercise is easy.]
- (d) Is this relation anti-symmetric? You don't have to write a proof. But explain informally why you think it is or it isn't.
- 

*What about 0?*

In the definition of divisibility, we only considered positive natural numbers because multiplication by 0 is strange. But we could have defined  $\mid$  for all natural numbers and let the chips fall where they may. So let's suppose we agree that  $m \mid n$  makes sense even when  $m$  or  $n$  is 0. That is,  $m \mid n$  still means that  $m \cdot q = n$  for some natural number  $q$ . Now, when would  $m \mid 0$  be true? Well, any time that we can find a natural number  $q$  satisfying  $m \cdot q = 0$ . That's easy since  $m \cdot 0 = 0$ . So it looks like our definition tells us that every natural number *divides* 0. Does that make sense to you? It seems reasonable, because  $m$  goes 0 times evenly into 0. But wait a minute. This also means that 0 divides 0. It seems peculiar to say that 0 divides anything. Nevertheless, the definition we are using says so. And conveniently, this means that the new definition of  $\mid$  is still a reflexive relation. It is still transitive because in the proof we only needed to know about associativity of multiplication.

Now what about anti-symmetry? That is, suppose  $m \mid n$  and  $n \mid m$  (where either  $m$  or  $n$  might be equal to 0). Is it still the case that  $m = n$ ? We can figure this out easily. Suppose  $m = 0$ . Then  $0 \cdot q = n$  for some  $q$ . But this implies  $n = 0$ . Likewise, if  $n = 0$ , then  $m = 0$ .

And we already know that if neither number is 0, then they are both positive. In every case,  $m = n$ .

So  $|$  makes sense as a partial order even when we consider 0. Some mathematicians don't like to think of 0 as dividing 0. So they stick with the original definition (or at least require  $m$  not to be 0 when they write  $m | n$ ). It is not a big deal, but the distinction can cause confusion in some edge cases. Let's make all this official!

---

**DEFINITION 5:** Divisibility

---

For natural numbers  $m$  and  $n$ , we say that  $m$  *divides*  $n$ , written  $m | n$ , provided that  $m \cdot q = n$  for some  $q \in \mathbb{N}$ .

---

The argument above the previous definition is the proof of the following proposition.

---

**PROPOSITION 13:** The natural numbers are partially ordered by divisibility.

---

For all natural numbers  $m$ ,  $n$  and  $p$ , the following are true.

*Reflexivity*             $m | m$ ,

*Transitivity*        if  $m | n$  and  $n | p$ , then  $m | p$ ,

*Anti-symmetry*     if  $m | n$  and  $n | m$ , then  $m = n$ .

---



Let's repeat the question, is there a largest natural number according to the ordering given by divisibility?

### *Laws of Ordered Arithmetic*

The arithmetic laws for addition and multiplication, as stated in Chapter 2, all deal with equality. But now that we have the standard order and the divisibility partial order of natural numbers, we ought to spend a bit of time looking at how these interact with arithmetic.

---

**LEMMA 5:** Addition is monotonic.

---

For any natural numbers,  $m$ ,  $n$ , and  $p$ , if  $m \leq n$ , then  $m + p \leq n + p$ .

*Proof.* This is an exercise. □

---

Strictly speaking, the statement here only says that addition is monotonic *on the left*. But because of commutativity, it is also monotonic on the right.



---

LEMMA 6: Addition is order-cancellative.

---

*For any natural numbers  $m$ ,  $n$  and  $p$ , if  $m + p \leq n + p$ , then  $m \leq n$ .*

*Proof.* This is an exercise. □

---

Likewise, this statement only says that addition is order-cancellative *on the right*. But because of commutativity, it is also order-cancellative on the left.

---

LEMMA 7: Multiplication is monotonic.

---

*For any natural numbers  $m$ ,  $n$  and  $p$ , if  $m \leq n$ , then  $m \cdot p \leq n \cdot p$ .*

*Proof.* This is an exercise. □

---



---

LEMMA 8: Multiplication by a positive natural number is order-cancellative.

---

*For any natural numbers  $m$  and  $n$ , and any positive natural number  $p$ , if  $m \cdot p \leq n \cdot p$ , then  $m \leq n$ .*

*Proof.* This is an exercise. □

---

The following table summarizes facts we have assembled about arithmetic and order for convenience. Some of the laws involving divisibility that are listed here are not explicitly proved above. This is because the proofs are “trivial” adaptations of analogous proofs involving  $\leq$ . You ought to “spot check” some of them. That’s a good use of a napkin.

---

**LAWS 2: Laws of Arithmetic and Order**


---

For any natural numbers,  $m$ ,  $n$  and  $p$ :

**Reflexivity**  $m \leq m$   
 $m \mid m$

**Transitivity** if  $m \leq n$  and  $n \leq p$ , then  $m \leq p$   
 if  $m \mid n$  and  $n \mid p$ , then  $m \mid p$

**Anti-symmetry** if  $m \leq n$  and  $n \leq m$ , then  $m = n$   
 if  $m \mid n$  and  $n \mid m$ , then  $m = n$

**Linearity** either  $m \leq n$  or  $n \leq m$

**Monotonicity** if  $m \leq n$  then  $m + p \leq n + p$   
 if  $m \leq n$  then  $m \cdot p \leq n \cdot p$   
 if  $m \mid n$  then  $m \cdot p \mid n \cdot p$

**Order cancellativity** if  $m + p \leq n + p$  then  $m \leq n$   
 if  $m \cdot p^{\wedge} \leq n \cdot p^{\wedge}$  then  $m \leq n$   
 if  $m \cdot p^{\wedge} \mid n \cdot p^{\wedge}$  then  $m \mid n$

**Modularity with addition** if  $m \mid n$  and  $m \mid p$ , then  $m \mid (n + p)$   
 if  $m \mid n$  and  $m \mid (n + p)$ , then  $m \mid p$

---

Like the table in Laws 1, this table is organized to emphasize similarities between addition and multiplication, hence between  $\leq$  and  $\mid$ .

Multiplication is monotonic with respect to  $\mid$ .

Multiplication by positive natural numbers is order-cancellative with respect to  $\mid$ .




---

**EXERCISES:**


---

23. Prove that addition is monotonic.
  24. Prove that addition is order-cancellative.
  25. Prove that multiplication is monotonic.
  26. Prove that multiplication by a positive natural number is order-cancellative.
  27. There seems to be missing monotonicity and order-cancellativity laws. The following are not listed in the table. Give examples why they are not laws.
    - (a) If  $m \mid n$ , then  $(m + p) \mid (n + p)$ .
    - (b) If  $(m + p) \mid (n + p)$ , then  $m \mid n$ .
-

### Subtraction and Division in Natural Numbers

Subtraction and division on natural numbers in the usual sense simply don't work properly. For example,  $6 - 9$  is just nonsense. There is no such thing (until someone invents negative numbers). Likewise,  $15/4$  is rubbish. But it turns out that we can make sense of subtraction and division inside the system of natural numbers, provided we adjust our perspective from thinking about *unique* solutions to thinking about *best* solutions.

We typically think of subtraction as the inverse of addition. Consider the equation  $7 = x + 5$ . Subtraction gives us a unique solution:  $x = 7 - 5$ . In fact, we could try to take this as the definition of subtraction:  $a - b$  is the unique number that solves the equation  $a = x + b$ . Putting it another way, we expect  $+$  and  $-$  to be related to each other by the fact that

$$a - b = c \text{ if and only if } a = c + b$$

for any numbers  $a$ ,  $b$  and  $c$ . If  $c$  happens to be a solution for  $a = x + b$ , then  $a - b = c$ . Conversely, since it is true that  $a - b = a - b$ , we are guaranteed that  $a - b$  solves  $a = x + b$ .

Similarly, division captures the inverse of multiplication:  $\frac{a}{b}$  is the unique solution to the equation  $x \cdot b = a$ . Obviously, if  $b$  is 0, then  $a$  has to be 0. But then  $x$  is not unique — any value will do. So  $\frac{a}{0}$  is not defined. Either there are no solutions for  $x \cdot 0 = a$  (when  $a \neq 0$ ) or there are infinitely many (when  $a = 0$ .) You already knew that. Like subtraction, we can capture the relation between multiplication and division by the fact that

$$a/b = c \text{ if and only if } a = c \cdot b$$

for any numbers  $a$ ,  $b$  and  $c$ , provided  $b \neq 0$ .

These equivalences are precisely what *defines* subtraction for integers and division for rational numbers. But neither one works as is for natural numbers.

To find a suitable re-interpretation of subtraction, suppose we relax a bit and ask for an operation  $\dot{-}$  on natural numbers so that

$$m \dot{-} n \leq p \text{ if and only if } m \leq p + n$$

for all natural numbers  $m$ ,  $n$  and  $p$ . This looks exactly like the characterization of subtraction, except that  $\leq$  replaces  $=$ . Is there an actual operation that does this job?

### Monus

Fix two natural numbers  $m$  and  $n$ , and consider the solutions of the inequality  $m \leq x + n$ . Clearly,  $m$  is an example because  $m \leq m + n$ .

This operation is known by the obscure term **monus**.

---

**EXAMPLE 4:**

---

For 5 and 7, apparently 2 solves  $7 \leq x + 5$ . Also any thing bigger that 2 does the job. But anything less than 2 fails. So  $7 \leq x + 5$  if and only if  $2 \leq x$ . So 2, in a sense, is all we need to know. It summarizes all possible solutions in one number because “greater than or equal to 2” means exactly the same thing as “solves  $7 \leq x + 5$ .”

Swapping 5 and 7, we see that  $5 \leq 0 + 7$ . And  $5 \leq x + 7$  if and only if  $0 \leq x$ .

---

The collection  $A$  of all the solutions to the inequation  $m \leq x + n$  is nonempty. For instance,  $m$  would be in  $A$ . Hence, by Proposition 10, there is a least solution of this inequality and the following definition is sound.

---

**DEFINITION 6: Monus**

---

Let  $m \dot{-} n$  denote the least natural number that solves  $m \leq x + n$ . In other words, for every  $m, n, p \in \mathbb{N}$ ,

$$m \dot{-} n \leq p \text{ if and only if } m \leq p + n.$$

---

The following is a very useful result showing that monus works indeed as a truncated difference as we intended. That is, whenever  $n \leq m$ , we have that  $m \dot{-} n$  is the “real difference” between  $m$  and  $n$ , that is,  $m \dot{-} n$  is the unique number  $d$  such that  $n + d = m$ . And, on the other hand, if  $m \leq n$  then  $m \dot{-} n = 0$ .

---

**LEMMA 9:**

---

For all natural numbers  $m$  and  $n$ , the following hold.

1. If  $n \leq m$  then  $n + (m \dot{-} n) = m$ .
2. If  $m \leq n$  then  $m \dot{-} n = 0$ .

*Proof.* Suppose that  $n \leq m$ . Then  $n + d = m$  for some  $d$ . By definition of  $\dot{-}$ , that implies  $m \dot{-} n \leq d$ . But conversely,  $n + d = m \leq n + (m \dot{-} n)$ . Since addition is order cancellative,  $d \leq m \dot{-} n$ . So actually  $d = m \dot{-} n$ .

Suppose that  $m \leq n$ . Then  $m \leq n + 0$  is true. Since 0 is the absolutely smallest natural number, it is the smallest solution for  $m \leq n + x$ . □

The monus operation can also be defined algorithmically:

$$\begin{aligned} m \dot{-} 0 &= m \\ 0 \dot{-} j^{\wedge} &= 0 \\ k^{\wedge} \dot{-} j^{\wedge} &= k \dot{-} j \end{aligned}$$

You ought to take a few minutes to check that this algorithm produces the value that satisfies the specification in this definition.

---

So you can think of  $m \dot{-} n$  as yielding the usual subtraction when  $n \leq m$ , and yielding 0 when that is the best it can do. Let us investigate how  $\dot{-}$  interacts with other arithmetic operations.

---




---

### EXERCISES:

---

28. Prove that  $(m + n) \dot{-} m = n$  for all  $m$  and  $n$ . [Hint: Prove first that  $(m + n) \dot{-} m \leq n$ . It is easier. Then prove that  $n \leq (m + n) \dot{-} m$ .]
  29. Show with a counterexample that  $(m + n) \dot{-} m = m + (n \dot{-} m)$  is not always true.
  30. Prove that  $\dot{-}$  is monotonic in its first argument. That is, if  $m \leq n$ , then  $m \dot{-} p \leq n \dot{-} p$ .
  31. Prove that  $\dot{-}$  is antitonic in its second argument. That is, if  $n \leq p$ , then  $m \dot{-} p \leq m \dot{-} n$ .
  32. Determine whether  $\dot{-}$  is order-right-cancellative, and prove your result. That is, is it the case that  $m \dot{-} p \leq n \dot{-} p$  implies  $m \leq n$ ? If not, provide a counter-example.
  33. Prove that  $(m \dot{-} n) \dot{-} p = m \dot{-} (n + p)$ .
  34. Prove that  $m + (n \dot{-} p)$  is not necessarily equal to  $(m + n) \dot{-} p$ . [Find a counter-example.]
  35. Prove that  $(m \dot{-} n) \cdot p = (m \cdot p) \dot{-} (n \cdot p)$ . [Hint: You are friends with induction now. It's time to ask your friend for help.]
  36. Prove that  $m \dot{-} n = (m + p) \dot{-} (n + p)$  for all  $m, n$  and  $p$ .
- 

The following is a tactic very useful when dealing with monus and quotients. It will also be extensively used in the second part of the course.

---

**PROOF TACTIC 12:** Proving equalities using inequalities.

---

If we want to prove that two natural numbers  $m$  and  $n$  are equal, we can prove one of the following.

1. For every number  $x$ , we have that  $m \leq x$  if and only if  $n \leq x$ .
2. For every number  $x$ , we have that  $x \leq m$  if and only if  $x \leq n$ .

It could seem a little odd that in order to prove that  $m = n$ , we would want to involve all the numbers  $x$ . But surprisingly, some times this is the best strategy.

The reason this works for part 1 is the following: obviously, if  $m = n$ , then it is trivially true that for every number  $x$ , we would have  $m \leq x$  if and only if  $n \leq x$ . The other direction is more interesting. Suppose then that we have proven that for every number  $x$ , the equivalence  $m \leq x$  if and only if  $n \leq x$  holds. Then, in particular, this would hold when  $x = m$  and when  $x = n$ . In the first case, if  $x = m$  then we can use the reflexivity of the standard order to deduce that  $m \leq m = x$ , and therefore we would obtain that  $n \leq x = m$ . On the other hand, when we take  $x = n$ , we again by the reflexivity we get  $n \leq n = x$ , and thus  $m \leq x = n$ . Putting together both cases, we obtain  $m \leq n$  and  $n \leq m$ , which entails  $m = n$ , by the anti-symmetry of  $\leq$ .

The reason the condition of part 2 also works is completely analogous and we will skip it.

---

### *Quotient and remainder*

When you were first learning about division, you probably were taught to think of  $16 \div 3$  as “5 with remainder 1.” We can make that idea precise now.

Consider the inequation  $x \cdot 3 \leq 16$ . The only solutions are 0, 1, 2, 3, 4, and 5. We can think of the quotient as the largest solution. Obviously, there is a bit left over that we will need to take into account. The bit left over is called the *remainder*.

---

#### DEFINITION 7: Natural number quotients

---

*For any natural number  $m$  and any positive natural number  $n$ , define  $m // n$  to be the largest natural number solving  $x \cdot n \leq m$ . In other words, for all natural numbers  $m$ ,  $n$  and  $p$  where  $n$  is positive,*

$$p \leq m // n \text{ if and only if } p \cdot n \leq m.$$


---

The notation  $//$  is borrowed from Python.



Why is this definition sound?

---

**EXAMPLE 5:**

---

Evidently,  $17 // 3 = 5$  because  $x \cdot 3 \leq 17$  if and only if  $x \leq 5$ . The natural number quotient  $m // n$  is what you learned in grade school as the quotient ignoring the remainder. Essentially, it counts the number of whole copies of  $n$  that fit in  $m$ . You might be used to writing common fractions. For example,  $17/3 = 5\frac{2}{3}$ . Apparently,  $17 // 3$  is the quotient part of the result, and the numerator 2 is what is left over.

---

Even though  $//$  is not defined in exact analogy with  $\div$  it shares some of its properties. For example, it is quite easy to check that  $//$  is monotonic in its first argument:  $m \leq n$  implies  $m // p \leq n // p$ . In order to prove this, first notice that the inequality  $m // p \leq m // p$  implies that  $(m // p) \cdot p \leq m$ , by the definition of  $//$ . Since we are assuming that  $m \leq n$ , we obtain that  $(m // p) \cdot p \leq n$ , and therefore  $m // p \leq n // p$ , again by the definition of  $//$ .

Recall that  $(m + n) \div n = m$ . The proof of this used the relation between  $\div$  and  $+$  and the fact that addition is order-cancellative. Analogous properties hold for  $//$  and  $\cdot$ . So the proof that  $(m + n) \div n = m$  can be adapted to prove that  $(m \cdot n) // n = m$ .

Because  $n \cdot (m // n) \leq m$ , the value  $m \div n \cdot (m // n)$  is the “true” difference between  $m$  and  $n \cdot (m // n)$ . It is precisely what is left over from division.

---

**DEFINITION 8: Remainder**

---

*For any natural number  $m$  and positive natural number  $n$ , define*

$$m \% n = m \div n \cdot (m // n) \cdot n.$$

*This is called the **remainder** of  $m$  divided by  $n$ .*

---

The notation  $\%$  is borrowed from languages like C and Python. In those languages, “percent” is an otherwise unused character. So the designers of C borrowed it, putting it to use to denote the remainder. Strictly speaking, though, in C  $\%$  is defined on integer data, not only on natural numbers.

By definition of monus, we have an additional law now:

$$m = (m // n) \cdot n + m \% n.$$



## EXERCISES:

37. Prove that  $m = (m // n) \cdot n + (m \% n)$  is true for any  $m \in \mathbb{N}$  and  $n \in \mathbb{N}^+$ .
38. Prove that  $m \% n < n$  for any  $m \in \mathbb{N}$  and  $n \in \mathbb{N}^+$ . [Hint: Prove that  $n \leq m \% n$  leads to a contradiction. So it cannot be true.]

We can summarize what we've discovered by the following fact.

## LEMMA 10: The Division Algorithm or Euclidean Division

*For any natural number  $m$  and positive natural number  $n$ , there are unique natural numbers  $q$  and  $r$  satisfying*

$$\begin{cases} m = q \cdot n + r, \\ r < n. \end{cases}$$

*Proof.* Let  $q = m // n$  and let  $r = m \% n$ . The foregoing Exercises 37 and 38 show that  $m = q \cdot n + r$  and  $r < n$ .

Suppose now that  $s$  and  $t$  satisfy

$$\begin{cases} m = s \cdot n + t, \\ t < n. \end{cases}$$

Our goal is to prove that  $s = m // n$  and  $t = m \% n$ . If  $s < m // n$ , then say  $s + d = m // n$  for some positive  $d$ . Thus

$$s \cdot n + t = m = (m // n) \cdot n + (m \% n) = s \cdot n + d \cdot n + (m \% n).$$

So  $t = d \cdot n + (m \% n)$ . But  $d$  is positive. So  $n \leq d \cdot n$ . This forces  $t$  to be greater than  $n$ . That is a contradiction. So  $s < m // n$  is impossible. By linearity  $m // n \leq s$ .

For similar reasons,  $m // n < s$  is impossible. So  $s = m // n$ . Now it follows directly that  $t = m \% n$ .  $\square$

Although this is called the Division Algorithm historically, this result does not provide an algorithm in our sense. On the other hand natural number division can be given an algorithmic implementation, provided we add tests to our repertoire.

$$\begin{aligned} m // n &= 0 && \text{if } m < n \\ m // n &= ((m \div n) // n)^\wedge && \text{otherwise.} \end{aligned}$$

This lemma says something about uniqueness. It is common to want that. For example, it is useful to know that the equation  $6 = 4 + x$  has one and only one solution. That is what we mean by the word *unique*. Proofs of uniqueness are common enough to deserve some attention in terms of proof tactics.



---

PROOF TACTIC 13: Unique Existential Generalization: Finding a Unique Witness

---

To prove a statement of the form “ $P(x)$  is true for a unique  $x$ ”, do the following.

1. Prove that there exists some  $x$  for which  $P(x)$  is true. For this, you are likely to use Tactic 8.
2. Suppose  $x_0$  and  $x_1$  are values so that both  $P(x_0)$  and  $P(x_1)$  are true. The names  $x_0$  and  $x_1$  must be fresh. Prove that  $x_0 = x_1$ .

This second part of the tactic is sound because it establishes that there cannot be two distinct values satisfying the condition. Hence there is a witness (part 1), and there cannot be more than one witness (part 2). There is a unique witness.

---




---

EXERCISES:

---

39. Find  $83 // 4$  and  $83 \% 4$ .
  40. Prove that  $//$  is monotonic in its first argument. That is, if  $m \leq m'$ , then  $m // n \leq m' // n$ .
  41. Prove that  $//$  is antitonic in its second argument. That is, if  $n \leq n'$ , then  $m // n' \leq m // n$ .
  42. Prove or disprove that  $(m // n) // p = m // (n \cdot p)$  for all  $m \in \mathbb{N}$  and all  $n, p \in \mathbb{N}^+$ .
  43. Prove that  $m // p + n // p$  is not necessarily equal to  $(m + n) // p$ . [Find a counter-example.]
- 

Exercises 40, 41, and 42 are almost parallel to Exercises 30, 31, and 33. Your proofs should be very similar.

Let's close the chapter with a synopsis of laws pertaining to monus and quotient.

---

**LAWS 3: Laws of Monus and Quotients**


---

For any natural numbers,  $m$ ,  $n$ ,  $q$ , and  $d$ , and any positive natural number  $p$ , the following are true.

**Characterization**      $m \dot{-} n \leq d$  if and only if  $m \leq d + n$   
 $q \leq m // p$  if and only if  $q \cdot p \leq m$

**Connection Laws**      $m = (m + n) \dot{-} n$   
 $m \leq (m \dot{-} n) + n$

$m = (m \cdot p) // p$   
 $m \geq (m // p) \cdot p$

**Division algorithm**      $m = (m // p) \cdot p + (m \% p)$   
 $m \% p < p$

**Distributivity**      $(m \dot{-} n) \cdot p = (m \cdot p) \dot{-} (n \cdot p)$

---

# 4

## Examples of Recursion and Induction

ARITHMETIC OPERATIONS ON NATURAL NUMBERS such as addition and multiplication are defined by recursion. In this chapter, we look at several other operations that arise in applications. Some of these, such as factorials, are familiar to you. Others are less so.

As we have seen in the previous chapters, some operations consists in repeatedly applying other operations a certain number of times. For example,  $m + n$  is the result of applying to the number  $m$  the successor operation  $n$  times,  $m \cdot n$  is the result of adding  $m$  with itself  $n$  times,  $m^n$  is the result of multiplying  $m$  with itself  $n$  times. We very often indicate this kind of repetition with ellipses, as follows:

$$m + n = m \overbrace{(\cdot \cdots \cdot)}^n, \quad m \cdot n = \overbrace{m + \cdots + m}^n, \quad m^n = \overbrace{m \cdot \cdots \cdot m}^n.$$

These expressions might be clear enough for us to carry out the computations, but this is because we understand the meaning of ‘...’ in each case. But, how can we make this meaning explicit? It will require a some thought. If an entity, let’s say a machine, knows how to compute an operation, we can tell the machine “do the operation twice,” or “do the operation three times,” or even “do the operation 136” times. But, how do we tell the machine “do the operation  $n$  times”?

The technique that we use in these cases is called *recursion*. This is the idea: we define the meaning of “do the operation  $n + 1$  times” as “to the result of doing the operation  $n$  times, do the operation once more.” In principle, it might seem that there is a danger of circularity, because we are defining “do the operation  $n + 1$  times” in terms of “do the operation  $n$  times.” But, the peculiarity of the natural numbers that every descending sequence will finish eventually, guaranties that we will eventually escape the computation, as long as we also define the meaning of “do the operation 0 times.” Be warned! In general, “do the operation 0 times” may nto seem to be the same as “don’t do

---

### CHAPTER GOALS

---

Define several additional operations on natural numbers using recursion. Prove useful facts about the operations, partly to get more practice deploying the proof tactic of simple arithmetic induction.

---

the operation at all.”

So, this is how we resolve the problem of the ellipses in the previous examples. Let’s start with addition:

$$m + n^{\frown} = m \overbrace{\frown \cdots \frown}^{n+1} = m \overbrace{\frown \cdots \frown}^n \frown = (m + n)^{\frown}.$$

This has to be completed with the clause  $m + 0 = 0$ . This is the reason why we introduced addition as

$$\begin{cases} m + 0 = m, \\ m + n^{\frown} = (m + n)^{\frown}. \end{cases}$$

in the earlier chapter. As for multiplication, we have:

$$m \cdot n^{\frown} = \overbrace{m + \cdots + m}^{n+1} = m + \overbrace{m + \cdots + m}^n = m + m \cdot n,$$

which, together with  $m \cdot 0 = 0$ , justifies the definition of multiplication that we gave previously. The same can be said for exponentials.

### *Factorial and related operations*

The factorial of a positive number  $n$  is the result of multiplying all numbers from 1 up to  $n$ , that is

$$n! = 1 \cdot 2 \cdot \cdots \cdot (n-1) \cdot n.$$

Let’s analyze this expression and see how we can derive a recursive definition, so that we can eliminate these ‘ $\cdots$ ’. We have to think about the result of the operation on  $n^{\frown}$ , which, as we know, is the (immediate) successor of  $n$ .

$$n^{\frown}! = 1 \cdot 2 \cdot \cdots \cdot n \cdot n^{\frown} = (1 \cdot \cdots \cdot n) \cdot n^{\frown} = n! \cdot n^{\frown}.$$

We can use the expression  $n^{\frown}! = n! \cdot n^{\frown}$  as a definition of the factorial, since in the right-hand side we only use the factorial evaluated at an “earlier stage,” and the other operations (multiplication and successor) have been already defined. This has to be completed with a value for  $0!$ . What would be a suitable one? Well, let’s expand, say,  $4!$  according to the given expression:

$$4! = 3! \cdot 4 = 2! \cdot 3 \cdot 4 = 2! \cdot 3 \cdot 4 = 1! \cdot 2 \cdot 3 \cdot 4 = 0! \cdot 1 \cdot 2 \cdot 3 \cdot 4.$$

Notice that, if we want  $4!$  to be  $1 \cdot 2 \cdot 3 \cdot 4$ , there is only one possible value that we can give to  $0!$ , namely, 1.

---

**ALGORITHM 3: Factorial**


---

The factorial of a natural number  $n$ , written  $n!$ , is the natural number defined by

$$\begin{aligned} 0! &= 1 \\ (n^{\curvearrowright})! &= n! \cdot n^{\curvearrowright}. \end{aligned}$$


---

Two closely related computations are known as *rising* and *falling exponents*. Rising exponents are easier to define. So let's start there. The idea, for example, is that 5 to the *rising* 3<sup>rd</sup> power is  $5 \cdot 6 \cdot 7$ , in contrast to the usual exponent  $5 \cdot 5 \cdot 5$ .

Let's find a recursive definition for  $m^{\overline{n}}$ . As before, we start by asking what is the value of the function at  $n^{\curvearrowright}$ , and try to find the same function evaluated at an "earlier stage." We can readily see that there are  $n$  factors in  $m^{\overline{n}}$ . Therefore,

$$\begin{aligned} m^{\overline{n^{\curvearrowright}}} &= m \cdot \overbrace{(m+1) \cdot \dots \cdot (m+n-1)} \cdot (m+n) \\ &= m \cdot (m^{\overline{n}})^{\overline{n}}. \end{aligned}$$

This should be a true identity, but can we use it? Notice that in the right-hand side of this expression, we have the function  $(m^{\overline{n}})^{\overline{n}}$ . This is indeed the evaluation of a function of an "earlier stage," but not the correct one. We should go back to the blackboard.

$$\begin{aligned} m^{\overline{n^{\curvearrowright}}} &= \overbrace{m \cdot (m+1) \cdot \dots \cdot (m+n-1)} \cdot (m+n) \\ &= m^{\overline{n}} \cdot (m+n). \end{aligned}$$

You see that, in the right-hand side, we have  $m^{\overline{n}}$ , which is the same function that we want to define, evaluated at an earlier stage.

---

**ALGORITHM 4: Rising exponent**


---

For natural numbers  $n$  and  $m$ , the rising  $n^{\text{th}}$  exponent of  $m$ , written  $m^{\overline{n}}$ , is a natural number defined by

$$\begin{aligned} m^{\overline{0}} &= 1 \\ m^{\overline{n^{\curvearrowright}}} &= m^{\overline{n}} \cdot (m+n). \end{aligned}$$


---

---

EXAMPLE 6:

---

To calculate  $5^{\overline{3}}$ :

$$\begin{aligned}
 5^{\overline{3}} &= 5^{\overline{2}} \cdot 7 \\
 &= (5^{\overline{1}} \cdot 6) \cdot 7 \\
 &= ((5^{\overline{0}} \cdot 5) \cdot 6) \cdot 7 \\
 &= ((1 \cdot 5) \cdot 6) \cdot 7 \\
 &= (5 \cdot 6) \cdot 7 \\
 &= 30 \cdot 7 \\
 &= 210
 \end{aligned}$$


---

Rising exponent is a hybrid between factorial and standard exponents and, as such, it will enjoy properties from both worlds. Indeed, factorial is a special case of rising exponent, as we will see in the next proposition. But also, rising exponent satisfies equations like  $m^{\overline{0}} = 1 = m^0$ ,  $m^{\overline{1}} = m = m^1$ , and a law similar to the familiar law of standard exponents that  $b^{x+y} = b^x \cdot b^y$ .

---

PROPOSITION 14: Factorial is a special case of rising exponent

---

For any natural number  $n$ ,

$$n! = 1^{\overline{n}}$$

*Proof.* The proof by induction on  $n$ .

**Basis** The goal is to show that  $0! = 1^{\overline{0}}$ . But  $0! = 1$  and  $1^{\overline{0}} = 1$  by definition.

**Inductive hypothesis** Suppose that  $k! = 1^{\overline{k}}$  for some  $k$ .

**Inductive step** The goal is to show that  $(k^{\frown})! = 1^{\overline{k^{\frown}}}$ .

$$1^{\overline{k^{\frown}}} = 1^{\overline{k}} \cdot (1 + k) = k! \cdot k^{\frown} = (k^{\frown})! \quad \square$$


---

---

PROPOSITION 15: The law of rising exponents

---

For any natural numbers  $m$ ,  $n$ , and  $p$ ,

$$m^{\overline{n+p}} = m^{\overline{n}} \cdot (m+n)^{\overline{p}}.$$

*Proof.* Let  $m$  and  $n$  be arbitrary numbers, and let's prove the equality by induction on  $p$ .

**Basis** The goal is to show that  $m^{\overline{n+0}} = m^{\overline{n}} \cdot (m+n)^{\overline{0}}$  for all  $m$  and  $n$ . This is clearly true because  $n+0 = n$  and  $(m+n)^{\overline{0}} = 1$ .

**Inductive hypothesis** Suppose that for some  $k \in \mathbb{N}$ ,

$$m^{\overline{n+k}} = m^{\overline{n}} \cdot (m+n)^{\overline{k}}.$$

**Inductive step** The goal is to show that

$$m^{\overline{n+k^\frown}} = m^{\overline{n}} \cdot (m+n)^{\overline{k^\frown}}.$$

$$\begin{aligned} m^{\overline{n+k^\frown}} &= m^{\overline{(n+k)^\frown}} && \text{— Def. of addition} \\ &= m^{\overline{n+k}} \cdot (m+n+k) && \text{— Def. of rising exp.} \\ &= m^{\overline{n}} \cdot (m+n)^{\overline{k}} \cdot (m+n+k) && \text{— Inductive hypothesis} \\ &= m^{\overline{n}} \cdot (m+n)^{\overline{k^\frown}} && \text{— Def. of rising exp.} \quad \square \end{aligned}$$


---

As we mentioned above, the equation  $m^{\overline{n^\frown}} = m \cdot (m^\frown)^{\overline{n}}$ , which appeared on our search for a recursive definition of rising exponent, must be true. We can indeed derive it from the law of rising exponents.

$$m^{\overline{n^\frown}} = m^{\overline{1+n}} = m^{\overline{1}} \cdot (m+1)^{\overline{n}} = m \cdot (m^\frown)^{\overline{n}}.$$

Falling exponents are defined in the opposite way. For example,  $5^{\overline{3}} = 5 \cdot 4 \cdot 3$ . For the present purpose, we will only define  $m^{\overline{n}}$  when  $n \leq m$ .

Recall that  $n \leq m$  means nothing other than that  $n + d = m$  for some  $d$ , or equivalently  $d + n = m$  for some  $d$ . So  $m^{\overline{n}}$  is the same thing as  $(d+n)^{\overline{n}}$ , where  $n$  and  $d$  can be any two natural numbers. That means we can stop fretting about the extra constraint that  $n \leq m$ . That may seem a bit strange at first, but it illustrates a handy technique.

---

ALGORITHM 5: Falling exponent

---

For natural numbers  $d$  and  $n$ , define the *falling exponent*  $(d + n)^{\underline{n}}$  by the following

$$(d + n)^{\underline{n}} = (d^{\wedge})^{\overline{n}}.$$


---



According to this expression, for which values of  $n$  is  $0^{\underline{n}}$  defined?

---

EXAMPLE 7:

---

To calculate  $5^{\underline{3}}$ :

$$\begin{aligned} 5^{\underline{3}} &= (2 + 3)^{\underline{3}} \\ &= (2^{\wedge})^{\overline{3}} \\ &= 3 \cdot 4 \cdot 5 \\ &= 60 \end{aligned}$$


---

A law of falling exponents is analogous to the law of rising exponents.

---



---

PROPOSITION 16: The law of falling exponents

---

For any natural numbers  $m$ ,  $n$  and  $p$ ,

$$(m + n + p)^{\underline{n+p}} = (m + n + p)^{\underline{p}} \cdot (m + n)^{\underline{n}}.$$

*Proof.* This is an exercise. □

---



---

EXERCISES:

---

44. Prove the law of falling exponents: for any natural numbers  $m$ ,  $n$ , and  $p$ ,

$$(m + n + p)^{\underline{n+p}} = (m + n + p)^{\underline{p}} \cdot (m + n)^{\underline{n}}.$$

45. Show that  $n^{\underline{n}} = n!$  for any natural number  $n$ .

46. Show that the falling exponents satisfy the following recursion: for any natural numbers  $n$  and  $d$ ,

$$\begin{aligned} (d + 0)^{\underline{0}} &= 1 \\ (d + n^{\wedge})^{\underline{n^{\wedge}}} &= (d + n^{\wedge}) \cdot (d + n)^{\underline{n}}. \end{aligned}$$


---



### *Path finding*

Suppose you are in a city with streets in a rectangular grid. You want to walk from one intersection to another that is 5 blocks north and 7 blocks east without wasting steps. Evidently, there are a lot of ways to do that. You could walk, let's say, east 7 blocks and then north 5 blocks. Or you could walk north 5 blocks then east 7 blocks, and end up in the same location. Or you could sort of zig-zag your way there. How many distinct paths do you have?

This is not an easy task. There are a lot of paths. By analyzing the problem more generally for walking  $m$  blocks north and  $n$  blocks east, the original problem will just be one instance of the general story.

First, let us approach the problem algorithmically by designing a procedure for calculating  $R(m, n)$  — the number of distinct paths to travel  $m$  blocks north and  $n$  blocks east.

For starters, think about  $R(0, n)$ . How many paths are available for walking  $n$  blocks due east without wasting steps? Clearly, you don't have any choices. So  $R(0, n) = 1$ .

Now what about  $R(m, 0)$ ? The same idea applies to walking due north. You have no choices. So  $R(m, 0) = 1$ .

To walk  $j$  blocks north and  $k$  blocks east, you could start by walking east one block, or you could start by walking north one block. If you walk east first, then the remainder of the journey requires you to walk  $j$  blocks north and  $k$  blocks east. If you walk north first, the remainder of the journey is to walk  $j$  blocks north and  $k$  blocks east.

So we have our algorithm.

---

#### ALGORITHM 6: Path counting

---

*For any natural numbers  $m$  and  $n$ , the number of distinct paths for walking  $m$  blocks north and  $n$  blocks east is defined recursively by the following.*

$$\begin{aligned} R(0, n) &= 1 \\ R(m, 0) &= 1 \\ R(j, k) &= R(j, k-1) + R(j-1, k) \end{aligned}$$


---

To calculate the original problem  $R(5, 7)$  requires a lot of steps. We will want to find a simpler way to compute  $R(m, n)$  for anything other than small inputs. So for now let's consider a smaller example.

---

EXAMPLE 8:

---

The number of routes to walk 3 blocks east and 3 blocks north is calculated (using some algebra to simplify many of the steps).

$$\begin{aligned}
 R(3,3) &= R(2,3) + R(3,2) \\
 &= R(1,3) + R(2,2) + R(2,2) + R(3,1) \\
 &= R(1,3) + 2 \cdot R(2,2) + R(3,1) \\
 &= R(0,3) + R(1,2) + 2 \cdot (R(1,2) + R(2,1)) + R(2,1) + R(3,0) \\
 &= 1 + 3 \cdot R(1,2) + 3 \cdot R(2,1) + 1 \\
 &= 1 + 3 \cdot (R(0,2) + R(1,1)) + 3 \cdot (R(1,1) + R(1,0)) + 1 \\
 &= 1 + 3 + 6 \cdot R(1,1) + 3 + 1 \\
 &= 1 + 3 + 6 \cdot (R(0,1) + R(1,0)) + 3 + 1 \\
 &= 1 + 3 + 6 + 6 + 3 + 1 \\
 &= 20
 \end{aligned}$$


---

As you can see, calculating  $R(5,7)$  would involve a lot of repetition, were you to follow the actual algorithm. Even in the example, we avoided a lot of repetition by consolidating some computations with algebra, but it would be even better to have a simple formula for  $R(m,n)$  that involves only easy-to-understand operations.

To investigate, let's cook up a table:

	0	1	2	3	4	5	...
0	1	1	1	1	1	1	...
1	1	2	3	4	5	6	...
2	1	3	6	10	15	21	...
3	1	4	10	20	35	56	...
4	1	5	15	35	70	126	...
5	1	6	21	56	126	252	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 4.1: Number of paths for walking  $m$  blocks east and  $n$  blocks north.

To get entry  $n$  in a row  $m$ , add the entries in the previous row up to that column. So there seems to be a simple relation between each row and the previous row, which for now we can write informally as

$$R(m, n) = \sum_{i=0}^n R(m, i).$$

This is a useful relation. But it still does not provide a direct way to calculate. Here is a much more useful fact.

---

**LEMMA 11:** Path formula
 

---

For any natural numbers  $m$  and  $n$ ,

$$R(m, n) \cdot m! \cdot n! = (m + n)!.$$

*Proof.* The proof is by induction on  $m$ . So the goal is to prove that for all  $m$ , it is the case that for all  $n$ ,  $R(m, n) \cdot m! \cdot n! = (m + n)!$ .

This proof will involve a nested induction.

**Outer Basis** The goal is to prove that for all  $n$ ,

$$R(0, n) \cdot 0! \cdot n! = (0 + n)!.$$

This is easy. You can supply the details.

**Outer Inductive Hypothesis** Suppose that for some  $j \in \mathbb{N}$ , it is the case that for all  $n$ ,  $R(j, n) \cdot j! \cdot n! = (j + n)!$ .

**Outer Inductive Step** The goal is to prove that  $R(j^\frown, n) \cdot j^\frown! \cdot n! = (j^\frown + n)!$  for all  $n$ . The proof of the outer inductive step is by induction on  $n$ .

**Inner Basis** The goal is to prove that  $R(j^\frown, 0) \cdot j^\frown! \cdot 0! = (j^\frown + 0)!$ . Again, this is easy.

**Inner Inductive Hypothesis** Suppose that for some  $k \in \mathbb{N}$  it is the case that  $R(j^\frown, k) \cdot j^\frown! \cdot k! = (j^\frown + k)!$ .

**Inner Inductive Step** The goal now is to show that

$$R(j^\frown, k^\frown) \cdot j^\frown! \cdot k^\frown! = (j^\frown + k^\frown)!.$$

$$\begin{aligned} R(j^\frown, k^\frown) \cdot j^\frown! \cdot k^\frown! &= (R(j, k^\frown) + R(j^\frown, k)) \cdot j^\frown! \cdot k^\frown! && \text{— By definition of } R \\ &= j^\frown! \cdot R(j, k^\frown) \cdot j! \cdot k^\frown! + R(j^\frown, k) \cdot j^\frown! \cdot k! \cdot k^\frown! && \text{— By algebra and definition of } ! \\ &= j^\frown! \cdot (j + k^\frown)! + R(j^\frown, k) \cdot j^\frown! \cdot k! \cdot k^\frown! && \text{— By the outer inductive hypothesis (1)} \\ &= j^\frown! \cdot (j + k^\frown)! + (j^\frown + k)! \cdot k^\frown! && \text{— By the inner inductive hypothesis} \\ &= (j^\frown + k)^\frown! \cdot (j^\frown + k)! && \text{— By algebra} \\ &= (j^\frown + k)^\frown! && \text{— By definition of } ! \\ &= (j^\frown + k^\frown)! && \text{— By definition of } + \quad \square \end{aligned}$$


---

In practice, the equation  $R(m, n) \cdot m! \cdot n! = (m + n)!$  is used to find  $R(m, n)$ . So it is typically written

$$R(m, n) = \frac{(m + n)!}{m! \cdot n!}.$$

The fraction notation is acceptable, because the division has no remainder and the denominator is never 0. The result is guaranteed to be a natural number.

### Binomial coefficients

The values  $R(i, j)$  show up in many situations that are not obviously related to counting the number of paths for walking in a grid. So even though the original problem may seem artificial, the solution is valuable. Problems like this are typically called **counting problems**. We will discuss counting problems in more detail in Chapter 16. For now, let us look at an algebraic problem that seems unrelated to paths, but turns out to be almost identical.

A *binomial* is a term of the form  $(x + y)^n$  for some natural number  $n$ . What  $x$  and  $y$  are meant to represent is not important, but we can suppose they are intended to be real numbers. The important thing is that addition and multiplication are associative and commutative, and multiplication distributes over addition. This means that binomials can be “multiplied out.” So  $(x + y)^2$  is equal to  $x^2 + 2xy + y^2$ . Likewise,  $(x + y)^3$  is equal to  $x^3 + 3x^2y + 3xy^2 + y^3$ . By putting the “hidden” coefficient 1 in front of  $x^n$  and  $y^n$ , and by writing all exponents explicitly, we get Table 4.2.

$n$	$(x + y)^n$					
0	$1x^0y^0$					
1	$1x^1y^0$	$+1x^0y^1$				
2	$1x^2y^0$	$+2x^1y^1$	$+1x^0y^2$			
3	$1x^3y^0$	$+3x^2y^1$	$+3x^1y^2$	$+1x^0y^3$		
4	$1x^4y^0$	$+4x^3y^1$	$+6x^2y^2$	$+4x^1y^3$	$+1x^0y^4$	
5	$1x^5y^0$	$+5x^4y^1$	$+10x^3y^2$	$+10x^2y^3$	$+4x^1y^4$	$+1x^0y^5$

Table 4.2: The first several binomials

The coefficients in Table 4.2 match the entries in Table 4.1 provided they are re-arranged slightly. The binomial  $(x + y)^n$  consists of  $n + 1$  terms, and the coefficient for the  $k^{\text{th}}$  term is  $R(k, n \div k)$ . This only makes sense when  $k \leq n$ .

Most of the practical uses for  $R$  are in situations like this, where instead of being handed  $i$  and  $j$  explicitly, we are handed  $i$  and  $n$  (the sum of  $i$  and  $j$ ). This leads to some standard notation.

---

#### DEFINITION 9: Binomial Coefficients

---

For natural numbers  $n$  and  $k$  where  $k \leq n$ , the **binomial coefficient**, written  $\binom{n}{k}$ , is defined by

$$\binom{n}{k} = R(k, n \div k).$$

Using the formula for  $R$ , most folks write this as

$$\binom{n}{k} = \frac{n!}{k! \cdot (n - k)!}$$

We could also have defined this by saying  $\binom{i+j}{i} = R(i, j)$ .

noting that  $n - k$  is safe to write in place of  $n \div k$  because  $k \leq n$ .

From the recursion of Algorithm 6, we immediately derive the following:

$$\binom{n}{0} = 1, \quad \binom{n}{n} = 1, \quad \binom{n}{k} = \binom{n}{k-1} + \binom{n}{k}.$$

EXAMPLE 9:

To compute  $\binom{5}{3}$ , we can use this recursion:

$$\begin{aligned} \binom{5}{3} &= \binom{4}{2} + \binom{4}{3} \\ &= \binom{3}{1} + \binom{3}{2} + \binom{3}{2} + \binom{3}{3} \\ &= \binom{2}{0} + \binom{2}{1} + \binom{2}{1} + \binom{2}{2} + \binom{2}{1} + \binom{2}{2} + \binom{3}{3} \\ &= \binom{2}{0} + \binom{1}{0} + \binom{1}{1} + \binom{1}{0} + \binom{1}{1} + \binom{2}{2} + \binom{1}{0} + \binom{1}{1} + \binom{2}{2} + \binom{3}{3} \\ &= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ &= 10. \end{aligned}$$

But this is much easier to calculate using the formula we have discovered:

$$\binom{5}{3} = \frac{5!}{3! \cdot 2!} = \frac{120}{6 \cdot 2} = 10.$$



#### EXERCISES:

47. Calculate  $\binom{4}{2}$  by explicit steps as in Example 9, without using the formula that involves factorial.
48. Calculate  $\binom{10}{4}$  any way you want.
49. Show that  $n! = n^k \cdot (n \div k)!$  for any natural numbers  $n$  and  $k$  such that  $k \leq n$ , and deduce that  $\binom{n}{k} = \frac{n^k}{k!}$ . [Hint: Recall the law of the falling exponents.]

# 5

## Lists

*Dresser des listes est l'une des plus anciennes activités du poète.  
[Making lists is one of the poet's oldest activities.]*

— Jorge Luis Borges

NATURAL NUMBERS constitute an example of a general process whereby “bigger” objects are built up from “smaller” ones. The Axiom of Induction captures the idea of building up natural numbers, providing a method for proving facts about them. In this chapter, we develop analogous ways to think about *lists*.

### List Basics

The idea of a *list* is really meant to be the familiar one, so that a list of “to do” items is a list. The alphabetized names on a class roster is a list. Even a base ten numeral such as 30295 is a list of five digits used to represent the number thirty thousand two hundred ninety five. Similarly, words are lists of letters, sentences are list of words, paragraphs are lists of sentences, and so on. You can think of many other natural examples of lists.

Most of the time, an informal point of view is just fine. But because lists are so common, when we want to think carefully about them as lists, it is helpful to deal with them more formally. We will use square brackets, following notation in many common programming languages. So for example,  $[1, 3, 2, 1]$  is a list of natural numbers. Importantly, order matters in a list. So the list  $[1, 2, 3, 1]$  differs from  $[1, 3, 2, 1]$ .

In many practical situations, we do not really need brackets, or even commas to separate the items. The list of digits  $[1, 0, 2]$  is much more compactly written 201. Dropping the brackets and commas makes strings of digits easier to read. When we do this, we need to be clear what we mean (for example, listing the digits in reverse order is a choice we can make).

---

#### CHAPTER GOALS

---

Introduce a formalization of the familiar idea of *list*. Following the investigation of natural numbers, introduce list induction and prove basic facts about lists.

---

The difference in order is a choice we can make. It happens that it is more helpful to think of the numeral 201 as the list of digits starting from the least significant digit to the most.

For now, a list is just a list, so interpretation is not an issue. And because we are currently interested in general list behavior, in this chapter I'll stick with the “bracket and comma” notation.

Given two lists, say  $[2, 5, 1]$  and  $[3, 0, 9]$ , it is natural to think of gluing them together to obtain  $[2, 5, 1, 3, 0, 9]$ . This operation is called **concatenation**. There are many different common notations for concatenation. The programming language Haskell uses  $++$ ; Python uses  $+$ . Some mathematicians use  $\otimes$ , or even  $\cdot$ . We will adapt Haskell's notation, denoting the operation of concatenation by  $+$ . So  $[2, 5, 1] + [3, 0, 9]$  is the list  $[2, 5, 1, 3, 0, 9]$ .

First, let us consider some properties of concatenation that seem obvious based on an informal idea of lists.

- Concatenation ought to be *associative*. That is, for any three lists,  $L$ ,  $M$  and  $N$ , it ought to be the case that

$$L + (M + N) = (L + M) + N.$$

- There ought to be an empty list  $[]$  that is an *identity* for concatenation:

$$[] + L = L = L + [].$$

- Concatenation ought to be *cancellative*. Suppose that  $L + M = L + N$ . Then by “erasing” the items in  $L$ , it ought to be true that  $M = N$ . Likewise if  $L + N = M + N$  then  $L = M$  ought to hold.
- Suppose  $[x] = [y]$ . That is, the list consisting of the single item  $x$  is the same as the list consisting of the single item  $y$ . Then  $x$  ought to equal  $y$ .
- A list ought to be finite.

We could just point to the above properties and declare that lists are the things that satisfy all of them. This would be like defining the natural numbers by specifying how addition works. It could be done, but would require work to convince you that the above properties are enough. Perhaps we missed something.

A safer route is to describe how lists work in close analogy with the natural numbers and then argue that the resulting objects behave as we want them to. First, we formulate how to build a list one item at a time (similar to how successor is used to “build” a natural number up from zero). Second, we postulate basic facts about lists that correspond to the postulates for natural numbers, including a “no cheating” postulate analogous to the Axiom of Induction for natural numbers. Finally, we define concatenation and other operations algorithmically and prove that the result satisfies the desired properties.

Computer scientists usually think of building lists one item at a time by *prepending* items — sticking them on the front. Mathematicians can afford to be less picky about that because they do not need to worry about how to implement things on a machine. So they do not have a particular favored convention. We will follow the computer science convention. For example, to get the list  $[a, b, e]$ , we will start with the empty list  $[]$ , then put  $e$  on the front to get  $[e]$ , then put  $b$  on that to get  $[b, e]$ , and finally put  $a$  on the front to get  $[a, b, e]$ .

Adapting notation from Haskell,  $x : L$  indicates prepending. For historical reasons it is sometimes pronounced “cons” as in “5 cons  $[3, 4]$  is  $[5, 3, 4]$ .” This is a shortening of “construct” because this is the basic way to construct a list.

---

#### EXAMPLE 10:

---

Here are some examples.

- $5 : 6 : [4, 5]$  is the same as  $5 : [6, 4, 5]$ , which is the same as  $[5, 6, 4, 5]$
- $1 : []$  is the same as  $[1]$
- $1 : 2 : 3 : 4 : []$  is the same as  $[1, 2, 3, 4]$ .
- $1 : 2$  is *nonsense*.

You cannot prepend to 2 because it is not a list.

---

To start making the analogy with natural numbers clear, let’s spell out this vocabulary.

---

#### VOCABULARY 2: Basic Vocabulary of Lists

---

- There is a list, called **the empty list** and denoted by  $[]$ .
- For any thing  $x$  and any list  $L$ , there is another list, having  $x$  as the **head** and  $L$  as the **tail**. The result is denoted as  $x : L$ .

You can still use square brackets to abbreviate lists that are built up, just as you use base ten numerals to abbreviate natural numbers. So for example,  $[2, 3]$  is an abbreviation for  $2 : 3 : []$ . The  $:$  operation associates right to left. So  $2 : 3 : []$  means  $2 : (3 : [])$ . The alternative  $(2 : 3) : []$  is nonsense — because 3 is not a list.

---

It is really important to note that the notation  $a : b$  makes sense exactly when  $b$  is a list. For example,  $[3, 4] : [5, 6]$  is a list (it is  $[[3, 4], 5, 6]$ ). But  $[3, 4] : 5$  is rubbish. 5 is not a list.



A list may consist of items that are all similar. For example, a list may consist entirely of natural numbers, or entirely of alphabetic characters, or entirely of lists of natural numbers, and so on. But in general, the vocabulary permits *heterogeneous* lists. For example,  $[4, a]$  is a list consisting of two items that are not of the same kind. That is permitted by the vocabulary.

As with the natural numbers, we need postulates that prevent unwanted behavior.

---

POSTULATE 4: Empty list has no head or tail.

---

*For any thing  $x$  and any list  $L$ ,  $[] \neq x : L$ .*

---

Compare to Postulate 1, namely,  $0 \neq n^+$ . The list  $[]$  is analogous to the natural number 0.

Likewise, a list that is not empty can only be built one way.

---

POSTULATE 5: Uniqueness of head and tail

---

*For any things  $x$  and  $y$  and any lists  $L$  and  $M$ , if  $x : L = y : M$ , then  $x = y$  and  $L = M$ .*

---

Compare to Postulate 2. So  $x : L$  is analogous to a successor. But the list  $L$  has many distinct successors, one for each thing that could be prepended.

For example, if you know that  $[2, 3, 4, 5] = x : L$ , then you also know immediately that  $x$  must equal 2 and  $L$  must equal  $[3, 4, 5]$ . No other interpretation is possible.




---

#### EXERCISES:

---

50. Are  $(1 : [2]) : [3]$  and  $1 : ([2] : [3])$  the same list?
51. For each of following lists determine whether the list has a head or tail, and if so, indicate what is the head and what is the tail.
- |                     |                   |
|---------------------|-------------------|
| (a) $[4, 3, 0]$     | (d) $[[2, 3], 4]$ |
| (b) $[0, 6, 10]$    | (e) $[]$          |
| (c) $5 : [9] : [4]$ | (f) $[] : []$     |
52. Prove that if  $[x] = [y]$ , then  $x = y$ .
- 

Lists also need an induction axiom that ensures that all lists are finite.

---

**POSTULATE 6:** The Axiom of List Induction
 

---

*Anything that is not required to exist by Vocabulary 2 is not a list. To be more precise, no lists can be eliminated without violating the basic vocabulary.*

---

Compare to Postulate 3 in Chapter 1.

An immediate consequence of the axiom of list induction is that every nonempty list has a head and a tail, which by Postulate 5 uniquely determines that list. This compares to the fact that every nonzero natural number has a predecessor. That's what we called Case Distinction.

---

**PROPOSITION 17:** Every nonempty list has a head and a tail
 

---

*If  $L$  is a nonempty list, then there is an  $x$  and a list  $N$  so that  $L = x : N$ .*

*Proof.* We will prove this statement by contraposition. That is, we suppose that  $L$  is an object so that  $L \neq []$  and  $L$  doesn't have a head or a tail, and we will see that actually  $L$  is not a list. If it was, upon removal of  $L$  from the collection of all lists, we realize that  $[]$  is still in the collection, because  $L \neq []$ , and for any thing  $x$  and list  $N$ , the list  $x : N$  is still in the collection, because  $L \neq x : N$ , since  $x : N$  has head and tail, while  $L$  doesn't, by assumption. This means that  $L$  is not required by the vocabulary to exist, and therefore it is not a list, by Postulate 6. □

---

To be honest, I have swept a pretty big pile of dirt under the rug here. The vocabulary for lists declares that *any thing* can be an item on a list. Presumably, this means any *mathematical* thing, such as a natural number, a real number, a polynomial, another list, a list consisting of lists,  $\mathbb{N}$  (the set of natural numbers) and so on.

We will discuss this in detail later, but it turns out that with very reasonable assumptions about how mathematics works, the very idea of mathematical things has to be open-ended. I don't just mean we haven't thought of everything yet. I mean that it is actually impossible to say, once and for all, what constitutes "everything". So it is also impossible to say, once and for all, what is allowed to be an item on a list. And thus, it is impossible to say, once and for all, what all possible lists are. I know I am being vague here. I will make it up to you later.

The axiom of list induction, nevertheless, justifies proofs about lists using a scheme almost identical to simple arithmetic induction.

---

**PROOF TACTIC 14:** List Induction
 

---

To prove some property is true for all lists, proceed as follows.

**Basis** Prove that the property is true for  $[]$ .

**Inductive hypothesis** Assume that the property is true for some list  $K$ .

**Inductive step** Prove that for any thing  $x$ , the property is true for the list  $x : K$ .

Conclude that the property is true for all lists.

---

Lists also enjoy a kind of recursion, which is specially useful to define all sorts of concepts associated to lists. For instance, every list has a length, which is a natural number. The length of  $[2, 4, 3]$  is 3. The length of  $[2, [3, 4]]$  is 2. A precise definition is easy to formulate by recursion.

---

**ALGORITHM 7:** Length of a List
 

---

For a list  $L$ , the **length** of  $L$ , denoted by  $\text{len}(L)$ , is the natural number defined by

$$\begin{aligned}\text{len}([]) &= 0, \\ \text{len}(x : K) &= \text{len}(K)^\frown.\end{aligned}$$


---

**EXAMPLE 11:**


---

$$\begin{aligned}\text{len}([2, 3, 4]) &= \text{len}(2 : [3, 4]) \\ &= \text{len}([3, 4])^\frown \\ &= \text{len}(3 : [4])^\frown \\ &= \text{len}([4])^\frown^\frown \\ &= \text{len}(4 : [])^\frown^\frown \\ &= \text{len}([])^\frown^\frown^\frown \\ &= 0^\frown^\frown^\frown \\ &= 3\end{aligned}$$


---

The main operation on lists, remember, is meant to be *concatenation*. This is defined precisely by the following.

---

**ALGORITHM 8:** List Concatenation
 

---

For lists  $L$  and  $M$ , their **concatenation**, denoted by  $L \mathbin{+} M$ , is a list, defined by

$$\begin{aligned} [] \mathbin{+} M &= M, \\ (x : K) \mathbin{+} M &= x : (K \mathbin{+} M) \quad \text{for any thing } x \text{ and any list } K. \end{aligned}$$


---

---

**EXAMPLE 12:** List concatenation
 

---

To calculate  $[4, [5, 2], 1] \mathbin{+} [3, 4, 1]$ , follow the algorithm:

$$\begin{aligned} [4, [5, 2], 1] \mathbin{+} [3, 4, 1] &= (4 : [5, 2] : 1 : []) \mathbin{+} [3, 4, 1] && \text{--- } [4, [5, 2], 1] \text{ abbreviates } 4 : [5, 2] : 1 : [] \\ &= 4 : ([5, 2] : 1 : []) \mathbin{+} [3, 4, 1] && \text{--- Def. of } \mathbin{+} \\ &= 4 : [5, 2] : ([1 : []] \mathbin{+} [3, 4, 1]) && \text{--- Same} \\ &= 4 : [5, 2] : 1 : ([[] \mathbin{+} [3, 4, 1]) && \text{--- Same} \\ &= 4 : [5, 2] : 1 : [3, 4, 1] && \text{--- Same} \\ &= [4, [5, 2], 1, 3, 4, 1] && \text{--- Abbreviation} \end{aligned}$$


---

Observe that both in the definition of *length* of a list and *concatenation* of two lists, we have distinguished two cases. This is not exclusive of just these two concepts, but is part of a more general method that parallels recursion for natural numbers and is therefore also called *recursion* for lists. Thus, if we want to define an operation on lists by recursion, first we need to specify how this operation affects the empty list  $[]$ , and then we have to specify how this operation affects the rest of the lists, that is, lists of the form  $x : M$  with head and tail. And in this part of the definition, we are allowed to use the same operation that we are defining acting on  $M$ , since this would be an “early stage” of the operation.

The method of induction works very well with concepts that are defined using recursion. Now we can prove several useful facts about lists.

---

PROPOSITION 18:  $[]$  is the identity for  $+$ .

---

For list  $L$ ,

$$[] + L = L \quad \text{and} \quad L = L + [].$$

*Proof.* By definition  $[] + L = L$  is always true. The second equality is proved by list induction on  $L$ .

The proof should look familiar (see the proof of Lemma 2).

**Basis:**  $[] + [] = []$  is true by definition of  $+$ .

**Inductive hypothesis:** Assume  $K + [] = K$  for some list  $K$ .

**Inductive step:** Suppose  $x$  is some thing. The goal is to show the equality  $(x : K) + [] = x : K$ .

$$\begin{aligned} (x : K) + [] &= x : (K + []) && \text{— by definition of } + \\ &= x : K && \text{— by the Inductive Hypothesis} \end{aligned}$$

Thus (by the Axiom of List Induction),  $L + [] = L$  holds for all lists.  $\square$

---



---

PROPOSITION 19: Concatenation is associative.

---

For all lists  $L$ ,  $M$  and  $N$ ,

$$L + (M + N) = (L + M) + N.$$

*Proof.* Suppose  $M$  and  $N$  are fixed lists. The proof is by induction on  $L$ .

Again, this should look familiar. It is almost identical to the proofs that addition and multiplication are associative.

**Basis** The goal is to show that  $[] + (M + N) = ([] + M) + N$ . This follows easily from the definition of  $+$ .

**Inductive hypothesis** Suppose  $K + (M + N) = (K + M) + N$  for some list  $K$ .

**Inductive step** The goal is to show that for any  $x$ ,

$$(x : K) + (M + N) = ((x : K) + M) + N.$$

Let  $x$  be any thing. Then

$$\begin{aligned} (x : K) + (M + N) &= x : (K + (M + N)) && \text{— Def. of } + \\ &= x : ((K + M) + N) && \text{— Inductive Hypothesis} \\ &= (x : (K + M)) + N && \text{— Def. of } + \\ &= ((x : K) + M) + N && \text{— Def. of } + \end{aligned}$$

So  $L \oplus (M \oplus N) = (L \oplus M) \oplus N$  is true for all  $L$ . Since the proof does not depend on any special properties of  $M$  and  $N$  (except that they are both lists), the result is true for all lists  $M$  and  $N$ .  $\square$

For a technical reason mentioned earlier, we cannot say that lists with concatenation and  $[]$  form a monoid. To do that, we would need to specify a single collection of all lists. That is not possible because the items of lists (mathematical things) is an open ended collection.

Concatenation is also analogous to addition in the following way.

**PROPOSITION 20:** Concatenation is both left and right cancellative

For lists  $L$ ,  $M$  and  $N$ ,

- $L \oplus M = L \oplus N$  implies  $M = N$ ; and
- $L \oplus N = M \oplus N$  implies  $L = M$ .

*Proof.* Exercise.  $\square$

Here is another useful fact that we can prove relating length to concatenation.

**PROPOSITION 21:**  $\text{len}$  is a homomorphism with respect to concatenation and addition.

- $\text{len}([]) = 0$ ,
- for any lists  $L$  and  $M$ ,  $\text{len}(L \oplus M) = \text{len}(L) + \text{len}(M)$ .

*Proof.* The fact that  $\text{len}([]) = 0$  is true by the definition of the length of a list. Suppose then that  $M$  is some fixed list. The proof of the second part is by induction on  $L$ .

**Basis** The goal is to show that  $\text{len}([]) + \text{len}(M) = \text{len}([] \oplus M)$ . But

$$\text{len}([]) + \text{len}(M) = 0 + \text{len}(M) = \text{len}(M) = \text{len}([] \oplus M)$$

by definition of  $\oplus$  and the fact that 0 is the additive identity.

**Inductive Hypothesis** Suppose  $\text{len}(K \oplus M) = \text{len}(K) + \text{len}(M)$  holds for some list  $K$ .

**Inductive Step** The goal is to show that for any  $x$ ,

$$\text{len}((x : K) \oplus M) = \text{len}((x : K)) + \text{len}(M).$$

We will encounter many operations like  $\text{len}$  that translate one operation into another. In this case,  $\text{len}$  translates  $\oplus$  into  $+$ , and  $[]$  into 0. Such operations are called *homomorphisms*.

Suppose  $x$  is some thing.

$$\begin{aligned}
 \text{len}((x : K) \uplus M) &= \text{len}(x : (K \uplus M)) && \text{--- Def. of } \uplus \\
 &= \text{len}(K \uplus M)^\frown && \text{--- Def. of len} \\
 &= (\text{len}(K) + \text{len}(M))^\frown && \text{--- Inductive Hypothesis} \\
 &= \text{len}(K)^\frown + \text{len}(M) && \text{--- Lemma 1} \\
 &= \text{len}(x : K) + \text{len}(M) && \text{--- Def. of len} \quad \square
 \end{aligned}$$



## EXERCISES:

53. Prove Proposition 20.
54. Show that concatenation is not commutative.
55. Write an algorithm for the list operation of **appending** an item to the end of a list. That is, define an algorithm for  $L \frown x$ , so that the result is  $L$  with  $x$  appended to the end. For example,  $[3, 2, 7] \frown 5$  should result in  $[3, 2, 7, 5]$ .
56. Write an algorithm for list reversal. That is, define an operation  $\text{rev}$  on lists so that, for example,  $\text{rev}([2, 3, 4])$  is  $[4, 3, 2]$ . [Hint: Ask yourself what is the reverse of the empty list, and what is the reverse of  $x : L$ . Consider using  $\frown$  in your algorithm.]
57. Here are two sanity checks for  $\text{rev}$ .
  - (a) Show that  $\text{rev}(\text{rev}(L)) = L$  for any list  $L$ .
  - (b) Show that  $\text{rev}(L \uplus M) = \text{rev}(M) \uplus \text{rev}(L)$  for any lists  $L$  and  $M$ .
58. Prove that  $\frown$  satisfies properties exactly analogous to the first two postulates for  $::$ 
  - $[] \neq L \frown x$  for any list  $L$  and any thing  $x$ .
  - If  $L \frown x = M \frown y$ , then  $L = M$  and  $x = y$ .

The list induction axiom also has an analogue that I am not asking you to confirm.

These exercises are useful because they confirm that building lists by prepending ( $x : L$ ) versus by appending ( $M \frown y$ ) are interchangeable. So if it is easier to think about growing a list using  $:$  — which is the convention in computer science — fine. If in a particular use of lists it is easier to think about by growing lists using  $\frown$  — more common in mathematics — you are just as safe.

For example, suppose had we decided to define concatenation by  $\frown$ . It would read this way:

$$L \uplus [] = L$$

$$L \dot{+} (K \frown x) = (L \dot{+} K) \frown x$$

The result is *provably* the same as Algorithm 8. Other algorithms involving lists work similarly. So mainly the choice to use  $:$  versus  $\frown$  is about which one is more convenient for any particular situation.

### List Indexing

In a list  $L$ , the items are in order. So we can refer to items by their position in the list. There are two standards in mathematics for doing this: start counting from 1 or from 0. Although starting from 0 (meaning that the initial item of a list is item number 0) may take some getting used to, it actually makes many calculations simpler. For that reason, most programming languages use this convention for a lists and arrays. I will consistently start with 0.

---

#### ALGORITHM 9: Indexed items in a List

---

Suppose  $L$  is a list and  $i$  is a natural number so that  $i < \text{len}(L)$ . Note that  $\text{len}([]) = 0$ , so there is no  $i < \text{len}([])$ . We define  $L_i$  as follows.

$$\begin{aligned} & []_i \text{ is never defined because } i \not< \text{len}([]) \\ & (x:L)_0 = x \\ & (x:L)_{k \frown} = L_k \text{ provided that } L_k \text{ is defined} \end{aligned}$$


---

This is a precise way of explaining that in a list, for example  $L = [a, b, c, d, e]$ , we can refer to an item by its *index*, so that  $L_0 = a$ ,  $L_1 = b$  and so on, up to  $L_4 = e$ . For this example,  $L_5$ ,  $L_6$  and so on are not defined.

---

#### EXAMPLE 13:

---

Suppose  $L = [a, b, c, d, e]$ . We can calculate  $L_3$  explicitly step by step.

$$\begin{aligned} L_3 &= [a, b, c, d, e]_3 \\ &= (a:b:c:d:e:[])_3 \\ &= (b:c:d:e:[])_2 \\ &= (c:d:e:[])_1 \\ &= (d:e:[])_0 \\ &= d \end{aligned}$$

Of course, this is just a very careful way to find item number 3 in the list. In every day use, we humans would simply count forward



from the beginning of the list. This definition explains precisely what “simply count forward” means.

### Indexed notation

Itemization allows us to introduce some useful alternative notation. For instance, if  $a$  is a list of length 4, then  $a = [a_0, a_1, a_2, a_3]$ . And, instead of writing  $[a_0, a_1, a_2, a_3]$ , for example, it is sometimes more convenient and succinct to write  $[a_i]_{i < 4}$  or  $[a_i]_{i=0}^3$ . This is especially handy if the list is long:  $[b_j]_{j < 10000}$  is a lot shorter than writing out 10000 terms. It is also valuable when we want to consider a list of indeterminate length:  $[a_i]_{i < n}$  indicates a list  $L$  of length  $n$  so that  $L_i = a_i$  for each  $i < \text{len}(L)$ .

This notation is particularly useful when the items are systematically related to their indices. So for example,  $[i^2]_{i < n}$  the list consisting of the first  $n$  perfect squares.



Beware that  $[a_i]_{i < 0}$  is an empty list because there are no natural numbers  $i$  strictly less than 0.



### EXERCISES:

59. Suppose  $L$  is a list and  $i < \text{len}(L)$ . Then it makes sense to think about the list in which  $L_i$  is removed. For example, for  $L = [a, b, c]$ , removing  $L_1$  results in the list  $[a, c]$ . Let us denote the result of removing item  $i$  from list  $L$  as  $L \setminus i$ . So  $[a, b, c] \setminus 1 = [a, c]$ .

In this exercise, you define  $L \setminus i$  explicitly. Explain your answers for each point.

- Should  $[] \setminus 0$  be defined? If not, why not? If so, what should it be?
- What should  $(x : L) \setminus 0$  be?
- What should  $(x : L) \setminus k^\sim$  be? When should it be defined and undefined?
- Now write an algorithm for  $L \setminus i$  in a layout similar to Algorithm 9.

60. Suppose  $L = [3, 2, 3, 3, 5]$  and  $M = [0, 1, 2, 3, 4, 5]$ . Calculate the following explicitly step by step.

- $\text{len}(L)$
- $L_4$
- $(L + M)_9$
- $L \setminus 4$
- $(M \setminus 3)_5$

61. Do you think the list  $[a_i]_{i < n}$  is different than the list  $[a_k]_{k < n}$ ? Explain your thinking.
- 

### Counting items

Items on a list can be repeated, for example, in  $[3, 4, 3, 2, 5, 1]$  the natural number 3 appears two times. It is sometimes useful to know how many repetitions there are.

---

#### ALGORITHM 10: Counting list items

---

For a list  $L$  and thing  $x$ , define  $\text{count}(x, L)$  to be the number times  $x$  appears as an item on the list  $L$ .

*Specifying an explicit recursive algorithm is an exercise.*

---



#### EXERCISES:

---

62. Define a recursive algorithm implementing  $\text{cnt}$  for Algorithm 10. Use of the algorithm  $\text{count}(x, L)$  should produce a natural number that counts the number of occurrences of  $x$  on  $L$ . For example,  $\text{count}(3, [4, 5, 3, 6, 3])$  should result in 2. But  $\text{count}(7, [4, 5, 3, 6, 3])$  should result in 0.
- 

### Homogeneous lists

We will commonly need to consider homogeneous lists — lists in which all items are drawn from the same collection. When that collection is a clearly defined set, these lists also comprise a clearly defined set. And since we already know that concatenation is an associative operation with identity  $[]$ , this set actually forms a monoid.

A typical example is that  $\mathbb{N}$  is a set. So the collection of lists of natural numbers is also a set. Evidently, if  $L$  and  $M$  are lists of natural numbers, so is  $L \# M$ . And  $[]$  trivially is a list of natural numbers (all of the items on the list are natural numbers). Remember, a monoid is a clearly defined set equipped with an associative operation and an identity for the operation. This leads to a general definition.

---

**DEFINITION 10:** List Monoids
 

---

If  $X$  is a set, then  $(\text{List}[X], +, [])$  is the monoid consisting of lists with items drawn from  $X$ . Precisely, the set  $\text{List}[X]$  is defined by

- $[] \in \text{List}[X]$ .
- If  $x \in X$  and  $L \in \text{List}[X]$ , then  $x : L \in \text{List}[X]$ .
- Nothing else is in  $\text{List}[X]$ .

The monoid unit  $[]$  and the monoid operation  $+$  are defined exactly as for general lists.

---

As usual,  $\text{List}[X]$  refers to the collection of all lists of  $X$ s, and as synecdoche, to the monoid.

---

**EXAMPLE 14:**


---

The monoid  $\text{List}[\mathbb{N}]$  consists of all lists of natural numbers. So  $[4, 5, 0, 1]$  belongs to  $\text{List}[\mathbb{N}]$ . But  $[3, [4, 5]]$  does not because it consists of two items, the second of which is  $[4, 5]$ , not a natural number.

Given that the monoid  $\text{List}[\mathbb{N}]$  consists of a clearly defined collection, we can also consider  $\text{List}[\text{List}[\mathbb{N}]]$ , consisting of lists of lists of natural numbers. For example,  $[[3, 2], [9, 1], []]$  is a list of lists of natural numbers.

---




---

**EXERCISES:**


---

63. Give four examples of an element of  $\text{List}[\text{List}[\text{List}[\mathbb{N}]]]$ . [Bonus for one of your examples being the smallest example possible.]
- 

It is not hard to think of situations where homogeneous lists are useful: lists of natural numbers, lists of names, lists of playing cards, whatever. It is valuable to know how these are related.

---

**DEFINITION 11: Monoid Homomorphisms**


---

We have encountered this idea before. Recall that the law of exponents tells us, for example, that  $2^{m+n} = 2^m \cdot 2^n$  and  $2^0 = 1$ . So  $m \mapsto 2^m$  is an operation that takes the additive monoid  $(\mathbb{N}, +, 0)$  into the multiplicative monoid  $(\mathbb{N}, \cdot, 1)$  by transforming  $+$  into  $\cdot$  and  $0$  into  $1$ .

For two monoids  $(X, \star, e)$  and  $(Y, *, \perp)$ , a homomorphism is a function  $h: X \rightarrow Y$  satisfying

$$\begin{aligned} h(e) &= \perp \\ h(x \star y) &= h(x) * h(y). \end{aligned}$$


---

For a given set  $X$ , the operation that sends an element  $x \in X$  to the singleton list  $[x]$  shows in a precise way how the monoid  $\text{List}[X]$  is related to the set  $X$ . In a sense, the elements of  $X$  translate to these special, very simple, lists. The next fact spells out what makes  $[x]$  special.

---

**THEOREM 1:  $\text{List}[X]$  is a free monoid over  $X$** 


---

Suppose  $X$  is a set,  $(Y, \star, e)$  is a monoid, and  $f: X \rightarrow Y$  is a function. There is a unique monoid homomorphism  $f^\dagger: \text{List}[X] \rightarrow Y$  satisfying  $f^\dagger([x]) = f(x)$  for every  $x \in X$ .

*Proof.* Define  $f^\dagger$  recursively by the following

$$\begin{aligned} f^\dagger([]) &= e \\ f^\dagger(x : L) &= f(x) \star f^\dagger(L). \end{aligned}$$

By definition  $f^\dagger([]) = e$ , and we can check by induction on lists that  $f^\dagger(L + M) = f^\dagger(L) \star f^\dagger(M)$ . So  $f^\dagger$  is a monoid homomorphism. Moreover,  $f^\dagger([x]) = f(x) \star f^\dagger([]) = f(x) \star e = f(x)$ . So  $f^\dagger$  meets our specifications.

Suppose  $h$  is a homomorphism from  $\text{List}[X]$  to  $Y$  for which

$$h([x]) = f(x)$$

for all  $x \in X$ . By induction on lists,  $h(L) = f^\dagger(L)$  for all lists  $L \in \text{List}[X]$ . So  $h$  and  $f^\dagger$  are the same function.  $\square$

---


**EXERCISES:**


---

64. Complete the proof of Theorem 1.
-

### List as a functor

The construction of the monoid  $\text{List}[X]$  from a set  $X$  extends to operations.

---

#### DEFINITION 12: List extended to operations

---

Suppose  $X$  and  $Y$  are sets, and  $f: X \rightarrow Y$  is an operation that transforms elements of  $X$  into elements of  $Y$ . Then define an operation from  $\text{List}[X]$  to  $\text{List}[Y]$  that applies  $f$  to each item on a list by recursion.

$$\begin{aligned}\text{List}[f]([]) &= [] \\ \text{List}[f](x : K) &= f(x) : \text{List}[f](K)\end{aligned}$$


---

For the example of factorial,  $\text{List}[\!:]$  is the function that transforms a list  $L$  of natural numbers into the list of factorials of the items in  $L$ .

$$\begin{aligned}\text{List}[\!:]([4, 2, 5]) &= 4! : \text{List}[\!:]([2, 5]) \\ &= 4! : 2! : \text{List}[\!:]([5]) \\ &= 4! : 2! : 5! : \text{List}[\!:]([]) \\ &= 4! : 2! : 5! : [] \\ &= [4!, 2!, 5!]\end{aligned}$$


---

For example, the factorial operation  $n \mapsto n!$  transforms natural numbers into natural numbers. For another example,  $n \mapsto [n]$  transforms natural numbers to lists of natural numbers.

This definition of extending a function to a function on lists is used commonly in computing, where it is often referred to as *map*. Informally,  $\text{List}[f]$  “maps” the function  $f$  across all the items of a list.




---

#### EXERCISES:

---

In these exercises, you will investigate the behavior of  $\text{List}[-]$  with respect to operations.

65. The successor of natural numbers is an operation  $(\_)^{\circ}$  taking elements of  $\mathbb{N}$  to elements of  $\mathbb{N}$ . For the following lists, calculate  $\text{List}[(\_)^{\circ}](L)$ .
  - (a)  $[3, 5, 1]$
  - (b)  $[0, 9, 99, 999]$
  - (c)  $[]$
66. Squaring is an operation  $(\_)^2$  taking natural numbers to natural numbers. For the same lists as in the previous exercise, calculate  $\text{List}[(\_)^2](L)$ .
67. For each of the lists in the previous two exercises, calculate  $\text{List}[(\_)^2](\text{List}[(\_)^{\circ}](L))$ .

68. Suppose  $f: X \rightarrow Y$  is an operation between sets  $X$  and  $Y$ , and  $g: Y \rightarrow Z$  is an operation between  $Y$  and another set  $Z$ . Furthermore, let  $h: X \rightarrow Z$  be the operation defined by  $h(x) = g(f(x))$ .

For a list  $K$  in  $\text{List}[X]$ , notice that  $\text{List}[f](K)$  is a list in  $\text{List}[Y]$ . So  $\text{List}[g](\text{List}[f](K))$  is a list in  $\text{List}[Z]$ . (See the previous exercise for examples). Show (by induction on lists) that  $\text{List}[g](\text{List}[f](K)) = \text{List}[h](K)$ .

---

Suppose  $(X, \star, e)$  is a monoid. We already know that  $\text{List}[X]$  is also a monoid using the operation  $\oplus$  and identity  $[]$ . These data lead to a homomorphism from  $\text{List}[X]$  to  $X$  defined by “folding”:

$$\begin{aligned}\text{fold}([]) &= e \\ \text{fold}(x : K) &= x \star \text{fold}(K).\end{aligned}$$

For example, the fold operation determined by  $(\mathbb{N}, +, 0)$  works like this

$$\begin{aligned}\text{fold}([4, 3, 7]) &= 4 + \text{fold}([3, 7]) \\ &= 4 + 3 + \text{fold}([7]) \\ &= 4 + 3 + 7 + \text{fold}([]) \\ &= 4 + 3 + 7 + 0.\end{aligned}$$

Clearly, this just adds the elements of the given list together. Sometimes this is written as  $\sum(L)$ . If we apply the same idea to the multiplicative monoid  $(\mathbb{N}, \cdot, 1)$ , then folding results in the product instead of the sum, but the basic procedure is the same. The product of a list is sometimes written as  $\prod(L)$ .

Folding has an important property. It preserves the monoid structure. That is  $\text{fold}(K \oplus L) = \text{fold}(K) \star \text{fold}(L)$  for every list  $K$  and  $L$  in  $\text{List}[X]$ . Likewise,  $\text{fold}([]) = e$  by definition. Now let's apply folding to the monoid  $\text{List}[X]$  itself. This transforms a list of lists of  $X$ s into a list of  $X$ s. For example,  $[[3, 4, 5], [2, 0, 9], [2, 4]]$  is transformed to  $[3, 4, 5, 2, 0, 9, 2, 4]$ .

### Slices

Suppose  $L = [a_0, \dots, a_{n-1}]$  is a list of length  $n$  and that  $i \leq j \leq n$ . We can extract the sublist of  $L$  consisting of items indexed  $i, i+1, \dots, j-1$ , writing  $L[i : j]$  for the list  $[a_i, a_{i+1}, \dots, a_{j-1}]$ . You may think this is strange usage because  $L[i : j]$  stops just one item shy of  $j$ . But it is exactly the same notation used, for example, in Python for what are called “slices” of a list. One reason this notation makes sense is that it works well with concatenation. Namely, if  $i \leq j \leq k \leq n$ , then

$$L[i, j] \oplus L[j, k] = L[i, k].$$

Strictly, this defines *left* folding for  $\star$  and  $e$ . But because we have assumed the monoid laws, left and right are provably the same. Many programming languages have built-in operations of *lfold* and *rfold* that work for any constant and any binary operation regardless of the monoid laws.

An alternative notation is  $[a_k]_{i \leq k < j}$ .

So here is a formal definition of slicing.

---

ALGORITHM 11: List Slicing

---

For a list  $L$  and natural numbers  $i \leq j \leq \text{len}(L)$ ,  $L[i : j]$  is a list defined by the following:

$$\begin{aligned} L[0 : 0] &= [] \\ (x : L)[0 : j^\frown] &= x : (L[0 : j]) \\ (x : L)[i^\frown : j^\frown] &= L[i : j] \end{aligned}$$


---




---

EXERCISES:

---

69. For  $L = [4, 3, 7, 9, 10]$  calculate  $L[2 : 4]$  by explicit steps.
  70. Show that for any list  $L$  and any  $i \leq \text{len}(L)$ , it is the case that  $L[i : i] = []$ .
  71. Show that for any list  $L$  and any  $i < \text{len}(L)$ , it is the case that  $L[i, i^\frown] = [L_i]$ .
  72. Show that for any list  $L$  and any  $i \leq j \leq k \leq \text{len}(L)$ , it is the case that  $L[i : j] \uplus L[j : k] = L[i : k]$ .
- 

### Prefix order

By definition,  $m \leq n$  on natural numbers means that  $m + d = n$  for some  $d$ . Likewise,  $m \mid n$  means that  $m \cdot d = n$  for some  $d$ . Several useful properties of ordering and divisibility come about naturally from these definitions and from the laws of arithmetic. An analogous definition makes sense for lists.

---

**DEFINITION 13:** Prefix ordering of lists
 

---

For lists  $L$  and  $M$ , say that  $L$  is a prefix of  $M$ , writing  $L \sqsubseteq M$ , if  $L \uplus D = M$  for some list  $D$ .

We may also say that  $M$  is an extension of  $L$  when  $L$  is a prefix of  $M$ .

---




---

**EXERCISES:**


---

73. Show that  $\sqsubseteq$  is reflexive.
  74. Show that  $\sqsubseteq$  is transitive.
  75. Show that  $\sqsubseteq$  is anti-symmetric.
  76. Show that  $\sqsubseteq$  is not linear.
  77. Show that  $\sqsubseteq$  is locally linear in the sense that if  $L \sqsubseteq N$  and  $M \sqsubseteq N$ , then either  $L \sqsubseteq M$  or  $M \sqsubseteq L$ .
- 

### Sorted Lists

In this section, we briefly discuss special lists that are comprised of data (such as natural numbers) that are equipped with an order. Consider the list  $[4, 7, 8, 10]$ , in contrast to  $[10, 4, 7, 8]$ . The former is *sorted*, in the sense that the elements appear in increasing order (in the standard  $\leq$  order). This is a useful property for lists that we will need in certain applications. So it is worth a short section to make the idea clear.

---

**DEFINITION 14:** Sorted lists
 

---

Suppose  $X$  is a collection and  $\sqsubseteq$  is a partial order on  $X$  (for example,  $\leq$  on the natural numbers). Then a list  $L \in \text{List}[X]$  is said to be **sorted** if it is the case that  $L_i \sqsubseteq L_j$  for every  $i < j < \text{len}(L)$ .

---



---

**EXAMPLE 15:**


---

Recall that we considered  $W$  to be the collection of all words from the Latin alphabet. They are ordered by the familiar alphabetical order (we could, but don't really need to, specify what this means precisely). For example, let us write  $\text{cat} \sqsubseteq \text{cave}$  to indicate that  $\text{cat}$  is alphabetically earlier than  $\text{cave}$ .

Then the list  $[\text{aardvark}, \text{beaver}, \text{civit}, \text{wombat}]$  is sorted.

---

*A not very good algorithm for sorting*

There are many algorithms for sorting lists that depend on storing a list in a particular way in a computer. They are typically discussed in a data structures course. In this subsection, we introduce an easily understood, but not very efficient method. The idea is to find an algorithm (call it *sort*) that takes a list  $L$  consisting of items that have some meaningful comparison  $\sqsubseteq$ , though it might actually be  $\leq$  for natural numbers, or  $|$  for natural numbers. The point is that it needs to be reflexive and transitive. Usually, it is also assumed to be anti-symmetric, but that is not actually needed.

Just to make the discussion easier, let us say a collection **has a comparison** if there is a reflexive, transitive relation  $\sqsubseteq$  on  $X$ .

The idea of the algorithm is to find where to insert a new item into a list when the new item might need to go somewhere other than at the head of the list. The insertion should not change anything else about the given list. But everything earlier in the list should be smaller than  $x$ .

With this “helper” insertion operation, sorting a given list becomes a simple recursion.

---

**ALGORITHM 12:** Insertion into a list
 

---

Suppose  $X$  is a collection that has a comparison  $\sqsubseteq$ .

For a list  $L$  in  $\text{List}[X]$ , and element  $x \in X$ , define  $\text{insert}(x, L)$  by the following:

$$\begin{aligned} \text{insert}(x, []) &= [x] \\ \text{insert}(x, y : K) &= \begin{cases} x : y : K & \text{if } x \sqsubseteq y \\ y : \text{insert}(x, K) & \text{otherwise} \end{cases} \end{aligned}$$


---

This algorithm ensures three things that can be proved in an exercise. To make the discussion easier, let  $L' = \text{insert}(x, L)$ . First, there is an  $i < \text{len}(L')$  so that  $L'_i = x$ , and for every  $k < i$ ,  $L'_k \neq x$ . This ensures that  $x$  appears for the first time in  $L'$  as item  $i$ . Second,  $L = L'[0 : i] \uplus L'[i + 1 : \text{len} L']$ . Third,  $L'_k \sqsubseteq x$  for every  $k < i$ .




---

**EXERCISES:**


---

78. Formulate the claims in the previous paragraph as a lemma.
79. By induction on lists, the lemma you formulated above.
- 

With this insertion algorithm, sorting is easy.

---

**ALGORITHM 13:** Insertion Sort
 

---

Suppose  $X$  is a collection that has a comparison  $\sqsubseteq$ .

For a list  $L$  in  $\text{List}[X]$ , define  $\text{sort}(L)$  by recursion as follows.

$$\begin{aligned} \text{sort}([]) &= [] \\ \text{sort}(x : K) &= \text{insert}(x, \text{sort}(K)) \end{aligned}$$


---

---

**THEOREM 2:** Insert sort is correct
 

---

Let  $X$  be a set with a comparison. For any list  $L$  in  $\text{List}[X]$ .

Then

- $\text{sort}(L)$  is sorted;
- $\text{len}(\text{sort}(L)) = \text{len}(L)$ ;
- For each  $x \in X$ ,  $\text{count}(x, L) = \text{count}(x, \text{sort}(L))$ .

*Proof.* The proof is by induction on the list  $L$ .

**Basis** All three claims are trivial for  $[]$  because  $\text{sort}([]) = []$ .

**Inductive hypothesis** Suppose all three claims are true for some list  $K$ , and let  $K' = \text{sort}(K)$ .

**Inductive step** Consider some  $x \in X$ , and the list  $L' = \text{sort}(x : K)$ . So  $L' = \text{insert}(x, \text{sort}(K))$ .

By the inductive hypothesis,  $K'$  is sorted, and by the sorting algorithm,  $L' = \text{insert}(x, K')$ . Let  $i$  be the first index where  $x$  appears in  $L'$ . According to the lemma from exercise 78, for each  $k < i$ ,  $L'_k = K'_k$ ,  $L'_i = x$  and for each  $k > i$ ,  $L'_{k+1} = K'_i$ . By cases, for every  $j \leq k < \text{len}(L')$ ,  $L'_j \sqsubseteq L'_k$ .

By the inductive hypothesis,  $\text{len}(K') = \text{len}(K)$ , and by the lemma from exercise 78,  $\text{len}(\text{insert}(x, K')) = \text{len}(K') \frown$ . So  $\text{len}(L') = \text{len}(x : K)$ . Finally, for  $y \in x : L$ , either  $y = x$ , or  $y \neq x$ . In the former case,  $\text{count}(y, L') = \text{count}(x, K') + 1 = \text{count}(x, K) + 1 = \text{count}(x, x : K)$ .

In the latter case,  $\text{count}(y, L') = \text{count}(x, K') = \text{count}(x, K) = \text{count}(x, x : K)$ .

□

---

### *Lazy lists, streams, sequences and coinduction*

Lists, as we have agreed, are finite. This is “enforced” by the axiom of induction which says that lists can only be built from  $[]$  by prepending items using  $:$ . But, let’s be honest, it could be useful to be able to talk about the “infinite list”  $[0, 1, 2, 3, \dots]$  (the “list” of natural numbers),  $[2, 3, 5, 7, \dots]$  (the “list” of primes),  $[0, 1, 1, 2, 3, 5, 8, \dots]$  (the “list” of Fibonacci numbers), and so on.

Mathematicians for a long time have taken an obvious point of view that an infinite list is essentially a sequence, which can be modelled as a function  $\sigma: \mathbb{N} \rightarrow X$ . This is likely the way you

encountered sequences in your calculus courses. For example, the sequence  $(1 + 1)^1, (1 + \frac{1}{2})^2, (1 + \frac{1}{3})^3, (1 + \frac{1}{4})^4, \dots$ , is a sequence that converges to the natural base  $e$ .

Some people even take that as the definition of  $e$ .

More recently, with insights from computer scientists, mathematicians have developed ways to deal with possibly infinite lists in a uniform way that makes it clearer that (finite) lists and (infinite) sequences are fundamentally similar.

Of course, *possibly* infinite lists have been used in mathematics long before mechanical computing became available. But computer scientists helped focus attention on a key feature that fits possibly infinite sequences into a larger scheme of definitions, showing that they are characterized by a sort of “opposite” of induction, called “coinduction.” The objects we get are often called, at least in computer science, **lazy lists**.

Laziness can be contrasted with *eagerness*. In a computational setting, suppose you ask a data structure for some information (say, you ask a list for its 100<sup>th</sup> item). An eager data structure will respond by giving you the entire contents of the structure, even if that is much more than you requested. So, an eager list will respond with the whole list, even if it has, say, a trillion items.

A lazy structure, on the other hand, will respond by evaluating only as much of its contents as needed to satisfy the request. In particular, a lazy list will unpack the first 100 items, and stop, handing over only the 100<sup>th</sup> item. It will not continue to unpack the rest of the list.

A university library is typically organized as an eager structure. Suppose you wish to check out a specific book. You walk into the stacks, where the entire collection is on display, and find your book. From the library’s point of view, it gives the patron access to the whole collection all at once. It is *eager* to share its contents.

Some library collections (such as special collections of rare books, or inter-library loans) operate as lazy structures. To check a particular book out, you fill out a request, and the library retrieves just that book and nothing else. It fulfills your request *lazily* by giving you only what you requested and nothing more.

Of course, in practice, most data structures use some combination of eager and lazy strategies. Even a library might have general access stacks and special retrieval for reserved books. But the distinction is useful. Eagerness, it turns out, is associated with induction. Laziness requires a different mode of reasoning, which is, in some sense, opposite of induction. We refer to this as **coinduction**.

The relation between induction and coinduction hinges on a more precise understanding of the roles played by basic vocabularies.

Our basic vocabulary of  $\mathbb{N}$  (0 and  $\hookrightarrow$ ) establishes a way to conduct

pattern matching. That is, a natural number either matches the pattern 0 (it is 0), or it matches the pattern  $(-)^{\frown}$  (it is a successor of another natural number).

The first two postulates of natural numbers (Postulates 1 and 2 on page 17) insist on that matching is unambiguous: no number can match both patterns 0 and  $n^{\text{xt}}$ , and if a number matches the pattern by  $n^{\frown}$ , then  $n$  is unique).

The Axiom of Induction (Postulate 3) insists that this pattern matching applies “eagerly.” There are no extras that are unnecessary for the pattern matching to work.

As discussed briefly in Chapter 1, we can summarize the scheme of patterns for natural numbers informally by

$$n \quad \leftrightarrow \quad 0 \quad | \quad n^{\frown}$$

where  $n$  is a placeholder for a natural number in the pattern, and the vertical bar separates the two patterns: 0 and  $n^{\frown}$ . Think of this as specifying three features of natural numbers:

- Each natural number matches exactly one of the patterns.
- Anything that matches a pattern is a natural number. Precisely, 0 is a natural number, and if  $m$  matches the pattern  $n^{\frown}$  for some  $n \in \mathbb{N}$ , then  $m$  is a natural number.
- If two natural numbers are distinct, they must distinguishable using patterns.

The last feature requires some more explanation. Suppose  $m$  is a natural number matching the pattern  $(-)^{\frown}$ , so  $m$  is a successor. Then the predecessor, in turn, is a natural number, so it must match either pattern 0 or  $(-)^{\frown}$ . So  $m$  matches either  $0^{\frown}$  or  $(-)^{\frown\text{frown}}$ . Iterating this,  $m$  might match  $(-)^{\frown\text{frown}\text{frown}}$ ,  $(-)^{\frown\text{frown}\text{frown}\text{frown}}$  and so on. If  $m$  differs from  $n$ , then at some point,  $m$  and  $n$  will disagree about a pattern  $(-)^{\frown\cdots\text{frown}}$ . Conversely, if  $m$  and  $n$  agree about which patterns they match, then they must be equal.

Consider the picture of Figure 1.7 in Chapter 1. Let’s refer to the collection of elements (the stepping stones) in that picture as  $\mathbb{N}^{\infty}$ . So  $\mathbb{N}^{\infty}$  consists of all the standard natural numbers plus  $\star$ . Evidently,  $\mathbb{N}^{\infty}$  follows exactly the same pattern matching rule as  $\mathbb{N}$ .

- Each element of  $\mathbb{N}^{\infty}$  (including  $\star$ ) matches exactly one of the patterns.
- Anything that matches a pattern is an element of  $\mathbb{N}^{\infty}$ . Precisely, 0 is in the picture, and if  $k = j^{\frown}$  for some  $j$  in the picture, then  $k$  is in the picture.

Rules like this can be made formal. In fact, several contemporary programming languages such as Agda use notation very similar to this to define inductive data types.

- If two elements of  $\mathbb{N}^*$  are distinct, they must differ by pattern matching.

For example, the element labelled  $\star$  is distinct from the element labelled 2 because  $\star$  matches the pattern  $(-)\curvearrowright\curvearrowright\curvearrowright$ , but 2 fails. In other words,  $(-)\curvearrowright\curvearrowright\curvearrowright$  is enough of a pattern to tell  $\star$  from 2. On the other hand, that pattern is not enough to tell  $\star$  from 3, or 4, and so on. Likewise,  $(-)\curvearrowright\curvearrowright\curvearrowright\curvearrowright$  distinguishes  $\star$  from 3, but not from 4. So,  $\mathbb{N}^\infty$ , as depicted in Figure 1.7, agrees with the pattern matching regime just as well as the actual natural numbers: any two distinct elements differ by some pattern.

But now consider the elements labelled  $\star_1$  and  $\star_2$  in Figure 5.1. They are distinct (according to the picture). But every pattern that  $\star_1$  matches — say  $(-)\curvearrowright\curvearrowright\curvearrowright$ , because  $\star_1$  is a successor of a successor of a successor — is also matched by  $\star_2$ . In other words, pattern matching cannot tell the two apart. So Figure 5.1 fails to follow the pattern matching rule.

It is not hard to see (if you want to be thorough, check by induction) that the only patterns that can be built from the rule are  $0\curvearrowright\dots\curvearrowright$  — meaning the element matches exactly one “true” natural number, or  $(-)\curvearrowright\dots\curvearrowright$  — meaning an item has at least a certain number of predecessors. From this, we see that  $\mathbb{N}$  and  $\mathbb{N}^\infty$  are actually the only two structures that follow the rule. The pattern matching itself does not force us to pick between  $\mathbb{N}$  or  $\mathbb{N}^\infty$ . We still need another principle to decide which one we intend.

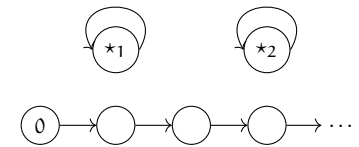


Figure 5.1: A model of the natural numbers?

---

#### INDUCTIVE VERSUS CO-INDUCTIVE NATURAL NUMBERS

---

For the pattern matching rule

$$n \quad \leftrightarrow \quad 0 \mid n\curvearrowright,$$

- the natural numbers  $n \in \mathbb{N}$  constitute the smallest possible structure satisfying the pattern matching rule —  $\mathbb{N}$  is said to be *defined inductively* from the rule;
- the “lazy” natural numbers  $n \in \mathbb{N}^\infty$  constitute the largest possible structure satisfying the rule —  $\mathbb{N}^\infty$  is said to be *defined coinductively* from the rule.

---

Similarly, for a collection  $A$ , the collection  $\text{List}[A]$  — consisting of lists with items in  $A$ , is defined *inductively* by the pattern matching rule

$$L \quad \longleftrightarrow \quad [] \mid a:L$$

where  $a$  ranges over elements of  $A$ . The inductive aspect of this means that only those things that are required by the rule to be lists are lists. For example,  $[]$  belongs to  $\text{List}[\mathbb{N}]$ , as does  $2 : []$ , and so on. But an “infinite” list does not because such things are not required.

But consider  $\text{List}^\infty[A]$ , defined coinductively by the same rule. Then in  $\text{List}^\infty[\mathbb{N}]$ , the “lazy” list  $0 : 1 : 2 \dots$  that goes on infinitely makes sense. Indeed, all such lazy lists belong to  $\text{List}^\infty[\mathbb{N}]$ , as do all “standard” finite lists.

Coinductive (lazy) lists and natural numbers can be used to define operations. For example, the length of a lazy list is defined by

$$\begin{aligned}\text{len}^\infty([]) &= 0 \\ \text{len}^\infty(a : L) &= \text{len}(L)^\frown\end{aligned}$$

This looks identical to the definition of  $\text{len}$ , but the argument  $L$  may be infinite. So the apparent recursion may not terminate. That is, the “length” of an infinite list is a successor of a successor of  $\dots$ . This cannot be distinguished from  $\infty$ , so it is  $\infty$ . In other words,  $\text{len}$  for lazy lists takes possible values in  $\mathbb{N}^\infty$  rather than in  $\mathbb{N}$ . Makes sense.

### Streams

In some situations, we may want to consider only definitely infinite lists, that is to explicitly exclude  $[]$ , or any other finite list. The result are called **streams**. For example,  $[0, 0, 0, \dots]$  is a stream of natural numbers.

Streams are characterized as *always* having a head and tail, in which the tail is another stream. So we can describe them by a simpler pattern matching rule.

---

#### STREAMS OF $A$ s

---

For a collection  $A$ , define the collection  $\text{Stream}[A]$  — elements of which are called **streams** coinductively by the rule

$$\sigma \quad \longleftrightarrow \quad a : \sigma.$$

Thus, the streams are all those things that match the pattern of  $a : \sigma$  where  $a$  belongs to  $A$  and  $\sigma$  is also a stream.

---

So, streams are data that produce elements of  $A$  *ad infinitum*. We have operations  $\text{hd}$  and  $\text{tl}$  (for head and tail) on streams that extract the initial element, and the rest of the stream.

$$\text{hd}(a : \sigma) = a$$

$$\text{tl}(a : \sigma) = \sigma$$

And for any natural number  $n \in \mathbb{N}$  and any  $\sigma \in \text{Stream}[A]$ , we can extract the  $n^{\text{th}}$  item by

$$\begin{aligned}\sigma_0 &= \text{hd}(\sigma) \\ \sigma_{n\sim} &= \text{tl}(\sigma)_n\end{aligned}$$

In general, induction permits us to define algorithms that *operate* on data. We have seen many examples. Coinduction permits us to define “coalgorithms” that *produce* data. For example, we can define, for  $a \in A$ , the constant stream

$$\text{const}(a) = a : \text{const}(a).$$

For natural numbers, we can define the streams  $\uparrow(n) = [n, n\sim, n\sim\sim, \dots]$  by

$$\uparrow(n) = n : \uparrow(n\sim)$$

So,

$$\uparrow(0) = 0 : \uparrow(1) = 0 : 1 : \uparrow(2) = 0 : 1 : 2 : \uparrow(3) = \dots$$

For example, to evaluate  $(\uparrow 5)_3$  we can proceed as

$$\begin{aligned}(\uparrow 5)_3 &= (5 : \uparrow 6)_{0\sim\sim\sim} \\ &= (\uparrow 6)_{0\sim\sim} \\ &= (6 : \uparrow 7)_{0\sim\sim} \\ &= (\uparrow 7)_{0\sim} \\ &= (7 : \uparrow 8)_{0\sim} \\ &= (\uparrow 8)_0 \\ &= (8 : \uparrow 9)_0 \\ &= 8\end{aligned}$$

### *Crossing the streams*

Pop culture aside, it is perfectly harmless to combine streams. There is no danger of total protonic reversal (I think, mostly). In fact, it can be useful and fun. Here is an example.



---

**ALGORITHM 14:** Adding streams of numbers
 

---

*For streams of natural numbers  $\sigma$  and  $\tau$ , define  $\sigma + \tau$  by*

$$(m : \sigma) + (n : \tau) = (m + n) : (\sigma + \tau).$$

*More generally, the same idea applies for any binary operation.*

---

This is not, technically, an algorithm, but a coalgorithm.

It should be clear that  $+$  and the constant stream  $0^*$  makes  $\text{Stream}[\mathbb{N}]$  another monoid. We leave that as an exercise.

Using the addition of streams, we can very efficiently (that is, efficient in terms of writing things down) define things like the Fibonacci stream coinductively as the stream satisfying

$$\text{fib} = 0 : 1 : (\text{fib} + \text{tl}(\text{fib})).$$

Then  $\text{fib}_0 = 0$ ,  $\text{fib}_1 = 1$ , and  $\text{fib}_{n+2} = \text{fib}_n + \text{fib}_{n+1}$ . Thus, this defines the stream corresponding to the Fibonacci sequence.

This illustrates one way that very recent developments in theoretical computer science shed some light on long-standing interests in mathematics.

If you encounter the programming language Haskell, or one of many similar languages, you will find that lazy lists and streams are fundamental. Indeed, Haskell is built on the idea of lazy evaluation. Lazy lists are built into the language.


**EXERCISES:**


---

80. Suppose we have a stream of natural numbers  $\sigma$  that satisfies the equation  $\text{tl}(\sigma) = \sigma + \sigma$ . Suppose moreover,  $\text{hd}(\sigma) = 1$ . Describe all the items on  $\sigma$ . Specifically, what is  $\sigma_n$ ?
-

*Part II*  
*Structure*

THE MATHEMATICAL UNIVERSE consists of various types of mathematical objects: numbers, functions, graphs, lists, matrices, groups, vector spaces, linear operators, and so on. We need ways to talk about these things, in isolation and in relation to one another.

An important aspect of structure is how we can put things together. That is, how we can *compose* pieces into larger things. We devote the next chapter to the general idea of composition. Sequential composition, putting one thing after another, is analogous to connecting pieces of equipment. Parallel composition, putting two (or more) things next to each other, is analogous to operating two pieces of equipment independently of each other. The content of next chapter is actually that concrete. We mainly look at how things fit together by these two forms of composition.

Of all the various types of objects, some seem to be fundamental in that others can be built from them. Following the discussion of composition, we introduce three fundamental types of objects: sets, functions, and relations. These, most mathematicians seem to agree, can be used as the fundamental building blocks of mathematics.

Suppose you are asked to solve the equation

$$0 = 2x^3 - x^2 + 6x - 3$$

for  $x$ . You would have a right to ask whether  $x$  is meant to be an integer (in which case there are no solutions), a real number (in which case there is one solution) or a complex number (in which case there are three solutions). So “solve for  $x$ ” does not really mean anything until you know what type of solution is being sought. This is no different than in ordinary conversation. You can not answer “What would you like?” unless you know what the question is about. Are you being asked what to have for dinner, what to get you for your birthday, what to do on vacation, or something else?

The development of programming languages has made clear the importance of data types. In most contemporary programming languages, each datum has an associated type. For example, a program might involve integer data separate from character data and separate from floating point data (roughly, floating point data approximate real number data in a way that is computationally tractable). Other data types might include things like matrices, arrays, lists, and much more.

In mathematics, *natural numbers*, *real numbers*, *integer polynomials*, *complex matrices*, *continuous functions on the reals* and so on are essentially the counterparts of data types. They help us to organize the things in much the same way that data types do in java or C++.

Though mathematicians tend to use the idea of types more informally than do computer scientists, one of the important lessons learned from computer science is that closer attention to type information helps clarify how things are related.

A *set* is, in the formulation of Cantor, *jedes Viele, welches sich als Eines denken lasst* “any multiplicity which can be comprehended as one.” For example, several playing cards taken together form a single deck of cards; the deck is a multiplicity of cards comprehended as one thing. The several students taking Discrete Math right now can be comprehended as one class. The infinitely many natural numbers can be regarded as single thing, the *set of natural numbers*. Thus a set is essentially a collection of elements.

In “typed” way of thinking, the natural numbers constitute a *type* of mathematical data, distinct from, say, the type of real numbers, the type of  $n \times n$  matrices, the type of Python programs, etc. The important point is, for example, that you know what it means to add two natural numbers, or prove something about them by simple arithmetic induction. On the other hand, adding two Python programs makes no sense. Proving something about real numbers by simple arithmetic induction makes no sense (but other “inductive” techniques do). A “typed” way of thinking, the emphasis is put on what you can do to

a datum. A type is essentially a specification of how to build data, what can be done to data of that type, and how to tell when data are equal. We hinted at this in Part I, when we developed the vocabulary and postulates for natural numbers. In effect, we presented the natural numbers as a type.

A *function* is a correlation of the members of one set with members of another set. Functions can be thought of and used in many, many ways. Here are some examples.

- Polynomial functions such as  $f(x) = x^3 + 2x^2 - x + 1$  are a particular kind of functions. They can be used in a wide variety of modeling problems in their own right, but also as approximations of more complicated phenomena.
- Definite integration takes a given function  $f$  and produces a second function  $\int_0^x f(t) dt$ . Integration is itself a “higher-order” function. Because it transforms one function into another.
- Cryptography is based almost entirely on the problem of designing functions with special properties. A crypto-system is based on two functions  $E$  (standing for “encrypt”) and  $D$  (“decrypt”). The function  $E$  will take a message  $m$  and an encryption key  $k_e$ , and produce an encrypted message  $E(m, k_e)$ . The function  $D$  will take an encrypted message  $c$  and a decryption key  $k_d$  and produce a clear message  $D(c, k_d)$ . The system of these two functions is *correct* if it is the case that whenever  $k_e$  and  $k_d$  are correctly paired,  $D(E(m, k_e), k_d) = m$ . In words, encrypting a message and then decrypting it with the matched key restores the original message. A correct system may still not be safe to use. To be minimally cryptographically safe, it must also be difficult computationally to determine  $m$  from  $E(m, k_e)$  and  $k_e$ , and must also be difficult to determine  $k_d$  from  $k_e$ .
- In programming, it is possible to implement a given process in many different ways. For example, one programmer may use a **while** loop, whereas another uses recursion. To understand how to compare two implementations, one needs to know that they both may implement the same function. Software engineers refer to “functional specifications” when they consider this.

A *relation* is more or less what is sounds like. For example,  $\leq$  is a relation on natural numbers. It is either true ( $4 \leq 7$ ) or not ( $4 \not\leq 2$ ) for any two natural numbers. Relations do not have to be defined *on* data of one type. For example, “solves” might be a relation between real numbers and polynomials: 2 solves  $0 = x^2 + x - 6$ , but 4 does not. In everyday life, we commonly think in terms of relations such as “Sam is a friend of Frodo,” “Frodo is shorter than Gandalf,” and so on.

Sets, functions and relations constitute a type-oriented framework in which virtually all of mathematics can be built. So an understanding of sets, functions and relations is key to a rigorous approach to most other parts of mathematics as well as of computing.

# 6

## Composition

*A mathematician, like a painter or poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with ideas.*

— Godfrey Harold Hardy

COMPOSITION arises naturally in mathematics and physics, system biology, programming, software design, web application design, electronics engineering, mechanical engineering, operations management, and almost any other discipline in which analysis or design involves putting things together. The basic mathematics of composition is simple and very widely used, yet giving rise to a sophisticated branch of mathematics called **category theory**.

Think about connecting your laptop to a monitor via an HDMI cable. With the connection made, it makes sense to think of the ensemble (laptop, cable, and monitor) as comprising a single new device. There are only specific ways to connect a laptop to a monitor. One cannot connect the monitor to an HDMI cable and try to connect that to the laptop's power input. Things just won't fit together if the connectors do not match. On the other hand, if we replace a 3 foot HDMI cable with a 6 foot cable, the assembled system will not change operationally (if we ignore some physical limitations such as impedance and noise).

Connecting two devices via a matching cable is called **sequential composition**. A bit of thinking about how sequential composition ought to behave leads to some very simple principles, which seem almost trivial and obvious, and which will be familiar to you in another form. But by writing these principles down carefully, we gain the advantage of being able to recognize sequential composition in a wide range of situations that have nothing to do with hooking electronic devices together.

**Parallel composition** refers to the idea of assembling two components to operate side-by-side. There can be different kinds of parallel composition depending on whether the two components interact.

---

### CHAPTER GOALS

---

In this chapter, we discuss two basic ways to compose devices: sequentially and in parallel.

---

Almost always, mathematicians drop the word “sequential” and just speak about composition. We will follow suit later.

In this chapter, we concentrate mainly on parallel *non-interacting* composition for two reasons. First, we think this is the most intuitive way to put things together. Two lamps can operate independently. For practical purposes, we can think of them as not interacting with each other (again, ignoring physical subtleties such as the heat of one lamp radiating to warm the other one slightly). Second, parallel non-interacting composition is critical for almost all areas mathematics, whereas parallel interacting composition tends to be special for a particular application.

The main task for this chapter is to explain the informal ideas of sequential and parallel composition, and to make them precise.

### Combinational Boolean Circuits

In any textbook on electronics engineering, you will find diagrams similar to Figure 6.1. This particular diagram shows circuit with three inputs:  $a$ ,  $b$ , and  $c_{in}$  and two outputs:  $s$  and  $c_{out}$ . The shapes  $\square$ ,  $\curlywedge$ , and  $\vee$  represent ‘and’, ‘xor’, and ‘or’ gates, respectively. The lines connecting them represent wires that can transmit single-bit values (0 or 1). The circuit represented here implements adding the single-bit values  $a, b$ , and  $c_{in}$ , resulting single-bit values  $s$  and  $c_{out}$  so that  $a + b + c_{in} = 2c_{out} + s$ . Stringing several of these basic circuits together results in circuit that adds two large numbers.

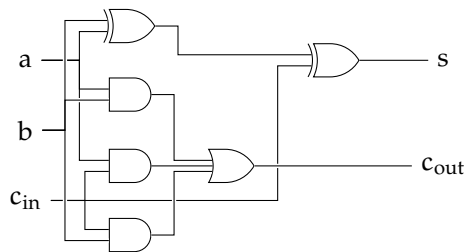


Figure 6.1: A fuller adder circuit

A full adder illustrates sequential composition. For example, the output of the top left ‘xor’ gate feeds into one of the inputs of the top right ‘xor’ gate. It also illustrates parallel composition. For example, the three ‘and’ gates operate in independently parallel.

Though the details of how to read such circuit diagrams is a later topic, the idea is that we can compose larger circuits from simpler ones. The mathematics of how composition works is quite general, and quite simple. Circuits provide one example, but there are many others that we will encounter later.

## Components and interfaces

Suppose a type of *component* comes with a specification of what kind of connection it needs for input and what kind it needs for output. A schematic picture of what components might look like is shown in Figure 6.

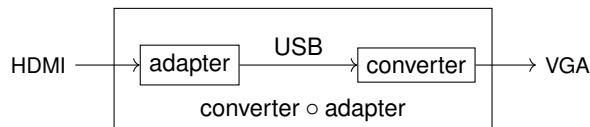
In reality, a component might need multiple inputs and multiple outputs. We will consider that situation when parallel composition comes into the picture. For now, it is simpler to think about single input, single output.

To make this precise, we suppose there is a collection of connector types that might consist of things like HDMI, USB, VGA, and so on. A type of component must have a specified connector type for input and a specified connector type for output.

Let us write  $C: \mathcal{X} \rightarrow \mathcal{Y}$  to mean that  $C$  is a type of component with input  $\mathcal{X}$  and output  $\mathcal{Y}$ . For example,  $\text{adapter}: \text{HDMI} \rightarrow \text{USB}$  is how we can write about the first component in Figure 6. Sometimes, it is more convenient to put the name of the component on top of the arrow instead of in front. So for example,  $C: \mathcal{X} \rightarrow \mathcal{Y}$  could also be written  $\mathcal{X} \xrightarrow{C} \mathcal{Y}$ .

## Sequential Composition

The two components depicted in Figure 6 can be connected in the obvious way to form a new component as in Figure 6.3. This is possible only because the output interface of *adapter* exactly matches the input interface for *converter*. We usually write  $\text{converter} \circ \text{adapter}$  (read “converter after adapter”) for sequential composition.



To deal with components and composition more generally, let us drop the references to HDMI, USB, adapters, and so on, and just deal with abstract connector types and components. So for this chapter,  $W, X, Y$  and so on will denote connector types and  $C, D, E$  and so on will denote component types.

Suppose we have three composable components  $C: W \rightarrow X$ ,  $D: X \rightarrow Y$  and  $E: Y \rightarrow Z$ . These can be combined sequentially as  $E \circ (D \circ C)$  and as  $(E \circ D) \circ C$ , pictured in Figure 6.4. The parentheses indicate the order of composition, corresponding to how the sub-assemblies in the figure were put together.

Other terms we could be using instead of *component*: *device*, *part*, *gadget*, *thingamabob*, etc. But *component* is useful because it has the same Latin root as *compose* — “place with”. *Component* means “thing that can be placed with other things.”

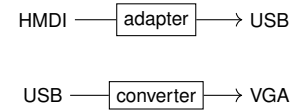


Figure 6.2: Two components with different interfaces.

Our example of electronic connectors is slightly misleading. HDMI, USB, etc., are actually bi-directional. So it is not really sensible to distinguish between input and output. But the two ends of an HDMI cable are shaped differently, so that it is impossible to connect them in the wrong way. We can use “input” and “output” as neutral terms to distinguish one end from the other.

The notation  $D \circ C$  is called *applicative notation*. In some situations it is also convenient to write  $C;D$  instead (“ $C$  before  $D$ ”, or “ $C$  then  $D$ ”). This is called *diagrammatic notation*.

Figure 6.3: Sequentially composed components.

The two orders of assembly result in the composite device because it doesn't matter whether  $Y$  is plugged together then  $X$ , or  $X$  is plugged together then  $Y$ . Once the whole thing is assembled, it would look like Figure 6.5. So a natural principle of sequential composition is the order of assembly should not matter. Putting it in technical language, composition should be associative.

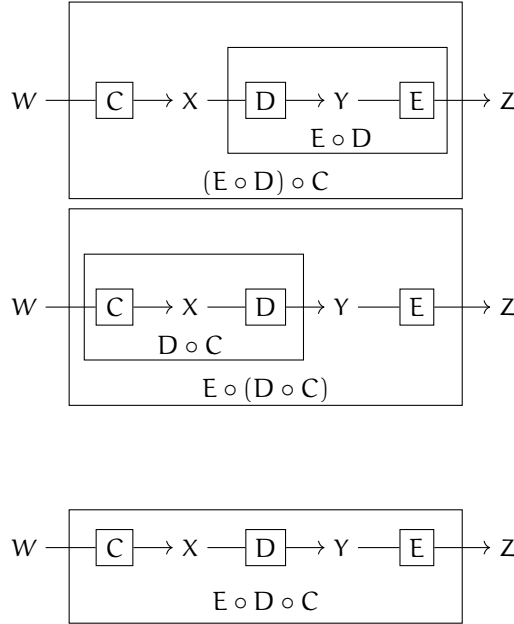


Figure 6.4: Two ways to construct  $E \circ D \circ C$ .

Figure 6.5: Sequential composition does not depend on grouping, so internal “boxing” is not needed.

For any connector type  $X$ , we also will suppose there is a *pass-through* component  $X \xrightarrow{\text{id}_X} X$ , depicted as in Figure 6.6. The purpose of pass-through components will be clearer once we look at parallel composition, but we can already understand what “pass through” ought to mean.

For any component  $C: X \rightarrow Y$  it ought to be the case that passing through  $Y$  after  $C$  should be the indistinguishable from  $C$  by itself. Likewise,  $C$  after passing through  $X$  ought to be the indistinguishable from  $C$ . This is summarized in Figure 6.7.

One useful way to think of pass-through components (we will start calling them *identities*) is that they allow us to extend a cable without changing anything operationally. In the diagrams of Figure 6.7,  $\text{id}_X$  and  $\text{id}_Y$  have the effect of extending the length of the input or output wire without changing how the whole assembly works.

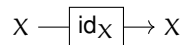


Figure 6.6: An identity or “pass-through” component.

Sequential composition is summarized by saying



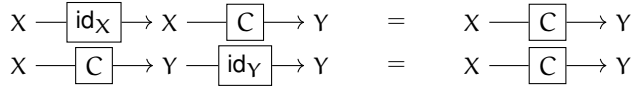


Figure 6.7: Identity components are “pass through” components

- composition is *typed* — components can be connected only if the input/output connectors match;
- composition is associative when it is permitted by matching types;
- the pass-through components  $\text{id}_X$  are *identities* for composition.

This leads to a formal definition.

---

#### DEFINITION 15: Categories

---

A *category*  $\mathcal{C}$  consists of a collection  $\text{Obj}_{\mathcal{C}}$ , whose elements are called *objects*, and for each  $X$  and  $Y$  in  $\text{Obj}_{\mathcal{C}}$ , a collection  $\mathcal{C}(X, Y)$ , called the *morphisms from  $X$  to  $Y$* . To simplify notation, we write  $f: X \rightarrow Y$  to indicate that  $f$  belongs to  $\mathcal{C}(X, Y)$ , and refer to  $X$  as *domain of  $f$*  and  $Y$  as the *codomain of  $f$* . Moreover, these data support sequential composition in the sense that the following hold.

- For any  $f: X \rightarrow Y$  and  $g: Y \rightarrow Z$ , there is a composite  $g \circ f: X \rightarrow Z$ , called “ $g$  after  $f$ .”
- For any  $X$  in  $\text{Obj}_{\mathcal{C}}$ , there is a designated “pass-through”  $\text{id}_X: X \rightarrow X$  called the *identity for  $X$* .
- For any  $f: W \rightarrow X$ ,  $g: X \rightarrow Y$ , and  $h: Y \rightarrow Z$ , it is the case that

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

- For any  $f: X \rightarrow Y$ , it is the case that

$$\text{id}_Y \circ f = f = f \circ \text{id}_X.$$

In this chapter,  $\text{Obj}_{\mathcal{C}}$  corresponds to the collection of connector types. Categories were first formulated in a mathematical setting where it made more sense to call them *objects*. We stick with tradition and write  $\text{Obj}_{\mathcal{C}}$  for the collection of objects (in our setting, the collection of connector types). Likewise,  $\mathcal{C}(X, Y)$  corresponds to the collection of components with  $X$  input and  $Y$  output. In the original mathematical setting, these are called *morphisms*;  $X$  is called the domain;  $Y$ , the codomain.

The last two conditions merely state that  $\text{id}_X$  really behaves as a pass-through and that sequential composition really behaves associatively.

---

#### Relation between categories and monoids

Suppose  $\mathcal{C}$  is a category. For each object  $X$ , the collection  $\mathcal{C}(X, X)$  constitutes all the morphisms with matching domain and codomain  $X$ . So any two of these can be composed, and  $\text{id}_X$  is the identity for composition inside  $\mathcal{C}(X, X)$ . In other words,  $(\mathcal{C}(X, X), \circ, \text{id}_X)$  is a monoid.

Conversely, every monoid corresponds to a category with a single object (a single connector type). For example, consider the additive

monoid  $\mathbb{N}$ . This corresponds to a category with exactly one object (say we call the object  $\bullet$ ). Then  $\mathcal{N}(\bullet, \bullet) = \mathbb{N}$ . Composition is addition:  $m \circ n = m + n$ , and 0 is the identity morphism for composition:  $\text{id}_\bullet = 0$ .

This demonstrates that categories, in a sense, are “generalized monoids.” A monoid is essentially just a special kind of category where any two morphisms can be composed.

### *Isomorphisms: Lossless conversion*

Suppose our category has a morphism  $X \xrightarrow{C} Y$  that corresponds to a converter of  $X$  inputs to  $Y$  outputs. Think of the power brick for your laptop. It takes a 110V alternating current input (“house current”) and transforms it (probably) to 12V direct current. But it also gives off a lot of heat. The heat is a loss. But suppose we could design a “lossless” converter. How would that look in a category? There is a very nice answer by looking at what it takes to convert output back into input.

In the foregoing exercises, you considered morphisms  $f: X \rightarrow Y$  and  $g: Y \rightarrow X$  satisfying  $g \circ f = \text{id}_X$ . Think of  $f$  and  $g$  as converters. Then the equation says that using  $f$  to convert from  $X$  to  $Y$  and then using  $g$  to convert from  $Y$  to  $X$  results in no change in  $X$ . So  $f$  could not have lost anything. Otherwise,  $g$  would be unable to recover everything. Thus  $f$  is a kind of lossless converter.

But usually, we think of “lossless” in a stronger sense, that  $f$  also does not add anything. Strictly speaking the name “lossless” is misleading, but bear with it.

Now we can ask for  $f$  to be lossless in the stronger sense that it neither loses nor adds anything in the process of converting from  $X$  to  $Y$ .

---

#### DEFINITION 16: Isomorphisms

---

Two objects  $X$  and  $Y$  in a category are **isomorphic** if there are morphisms  $f: X \rightarrow Y$  and  $g: Y \rightarrow X$  satisfying

$$g \circ f = \text{id}_X \quad \text{and} \quad f \circ g = \text{id}_Y.$$

The morphisms  $f$  and  $g$  are called **isomorphisms**. And  $g$  is an **inverse** of  $f$ .

---

The definition is symmetric. So if  $g$  is an inverse of  $f$ , then  $f$  is an inverse of  $g$ .

Isomorphisms correspond to strongly lossless converters. We can interchange  $X$  for  $Y$  without worrying that we somehow tampered

with the information carried by either. So if some other gadget has  $Y$  as its domain, we can simply turn it into a gadget that has  $X$  as its domain without loss of functionality.

Suppose you learn that  $f$  is an isomorphism (so really you learn that  $f$  has an inverse). Perhaps  $f$  has two inverses  $g$  and  $h$ . This means that  $g \circ f = \text{id}_X = h \circ f$  and  $f \circ g = \text{id}_Y = f \circ h$ . But then

$$g = \text{id}_X \circ g = (h \circ f) \circ g = h \circ (f \circ g) = h \circ \text{id}_Y = h.$$

If  $f$  has an inverse, it has only one. So it makes sense to speak about *the* inverse of  $f$  when it has an inverse. A common notation for the inverse of an isomorphism is  $f^{-1}$ .

We will encounter some isomorphisms in the next section. The important thing to remember for now is that they represent a lossless conversion in either direction between two objects.




---

### EXERCISES:

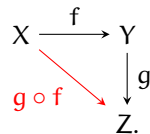
---

81. In a category, suppose  $f: X \rightarrow Y$  and  $g: Y \rightarrow X$  are morphisms so that  $X \xrightarrow{f} Y \xrightarrow{g} X$  equals  $X \xrightarrow{\text{id}_X} X$ .
- (a) For any other two morphisms  $h: X \rightarrow Z$  and  $k: X \rightarrow Z$ , draw  $k \circ g$  and  $h \circ g$ . Show that if  $k \circ g = h \circ g$ , then  $k = h$ .
  - (b) For any other two morphisms  $h: W \rightarrow X$  and  $k: W \rightarrow X$ , show that if  $f \circ k = f \circ h$ , then  $k = h$ .
82. Prove that a composition of isomorphisms is an isomorphism.
- 

### External Diagrams

Wiring diagrams are useful for understanding the details of composition. But they are awkward for reasoning about equations because we need to draw two (possibly complicated) diagrams just to state that they represent equal morphisms. That's fine for small examples, but is not helpful for more complicated situations involving many different morphisms.

When we are mainly concerned with how morphisms interact, an individual morphism can be depicted simply as  $X \xrightarrow{f} Y$ . Then a sequential composition can be depicted as



We do not really need to draw  $g \circ f$  as a separate arrow because the *path* from  $X$  to  $Y$  to  $Z$  is already an implicit part of the diagram. So the simpler diagram

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ & & \downarrow g \\ & & Z \end{array}$$

suffices to convey the same information, that  $f$  and  $g$  can be composed because the domain of  $g$  matches the codomain of  $f$ .

The identity morphisms for  $X$ ,  $Y$  and  $Z$  are also implicit. Had we drawn them, they would have appeared as “loops”.

The diagram

$$\begin{array}{ccc} W & \xrightarrow{f} & X \\ h \downarrow & & \downarrow g \\ Y & \xrightarrow{k} & Z \end{array}$$

depicts ten morphisms all together: four identities, the named morphisms  $f$ ,  $g$ ,  $h$ , and  $k$ , and the two composites  $g \circ f$  and  $k \circ h$ . We say that such a diagram **commutes** if  $g \circ f = k \circ h$ . More generally, to assert that an external diagram like this commutes means all the compositions depicted implicitly in the diagram that can be equal are equal. So a **commutative diagram** is a graphical depiction of equations. This takes some getting used to, but it is worth the practice.

Consider the Law of Associativity for composition. For composable morphisms, we can draw the basic situation as

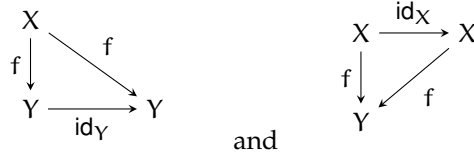
$$\begin{array}{ccc} W & \xrightarrow{f} & X \\ & & \downarrow g \\ & & Y \xrightarrow{h} Z \end{array}$$

Including the compositions  $g \circ f$  and  $h \circ g$ , we get

$$\begin{array}{ccccc} W & \xrightarrow{f} & X & & \\ & \searrow & \downarrow g & \searrow h \circ g & \\ & g \circ f & Y & \xrightarrow{h} & Z \end{array}$$

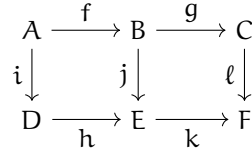
Associativity says that this diagram commutes. Likewise, the Identity

Laws say that the diagrams



both commute.

Commuting diagrams are especially efficient for reasoning about equations when diagrams are pasted together. Consider the following diagram.



Suppose the left square and the right square commute. That is,  $j \circ f = h \circ i$  and  $\ell \circ g = k \circ j$ . Then the outer rectangle (consisting of the four morphisms  $g \circ f$ ,  $\ell$ ,  $i$ , and  $k \circ h$ ) also commutes, as the following calculation shows.

$$\begin{aligned}
 \ell \circ (g \circ f) &= (\ell \circ g) \circ f && \text{— associativity} \\
 &= (k \circ j) \circ f && \text{— the right square commutes} \\
 &= k \circ (j \circ f) && \text{— associativity} \\
 &= k \circ (h \circ i) && \text{— the left square commutes} \\
 &= (k \circ h) \circ i && \text{— associativity}
 \end{aligned}$$

The calculation is tedious, but leads to a helpful generalization. We do not need to make these calculations every time we see a commuting diagram because the result of pasting any commuting diagrams together along their edges is always a commuting diagram. This makes it possible to reason very efficiently, and therefore more reliably, about larger systems of equations.

### Parallel composition

For two unrelated components  $C: W \rightarrow X$  and  $D: Y \rightarrow Z$ , their *parallel composition* ought to be a single component that consists of  $C$  and  $D$  operating independently. Figure 6 is the basic picture we are after.

But there is an obvious problem. In the definition of a *category*, we supposed that a component (morphism) has one input connector (domain) and one output connector (codomain). But parallel composition requires two inputs and two outputs. So a category that allows

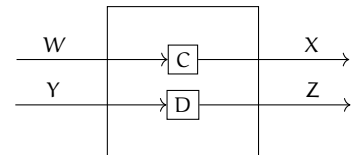


Figure 6.8: Two unrelated components composed in parallel

for parallel composition needs some added features. This leads to what are called **strict monoidal categories** (you might want to guess where “monoidal” comes from), and to an important generalization to **monoidal categories**. The difference between strict and non-strict is interesting in its own right, but wouldn’t make sense to discuss right now.

To support parallel composition, let us allow any two wires  $X$  and  $Y$  to be “bundled together” to form a new wire (sort of a “cable” consisting of  $X$  and  $Y$ ). A common notation for this is  $X \otimes Y$ . A reasonable assumption is that  $\otimes$  should be associative. Let us also suppose there is an “empty” wire. A common notation for this is  $\mathbb{I}$ . Bundling an empty cable to  $X$  should result in  $X$ . So the objects of our category (the wires) form a monoid  $(\mathbf{Obj}_C, \otimes, \mathbb{I})$ , where, informally,  $\otimes$  and  $\mathbb{I}$  specify how to bundle wires together.

The requirement of single input/single output for sequential composition does not need any change. If we want to think about a morphism with two inputs, we can just specify its domain to be  $X \otimes Y$ . So the definition of category has not changed. In order to (sequentially) compose two morphisms  $g$  following  $f$ , the codomain of  $f$  must exactly match the domain of  $g$ .

Evidently, the diagram in Figure 6 should now depict a component with domain  $W \otimes Y$  and codomain  $X \otimes Z$ . We will denote it by  $f \parallel g$ . Just like sequential composition, we expect parallel composition to be associative, as illustrated in Figure 6.9.

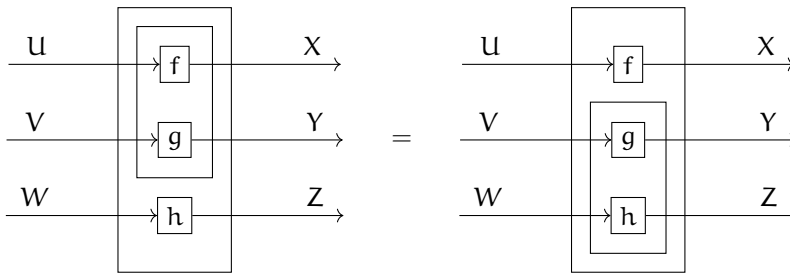


Figure 6.9: Parallel composition is associative. So

$$(f \parallel g) \parallel h = f \parallel (g \parallel h).$$

Because  $\mathbb{I} \otimes X = X$  and  $X = X \otimes \mathbb{I}$ , we can expect there to be a morphism  $e: \mathbb{I} \rightarrow \mathbb{I}$  that is meant to be the identity for parallel composition. In diagrams,  $e$  is best described as an empty diagram because parallel composition with it does nothing. To emphasize the “presence” of an empty diagram, we will sometimes use the symbol  $\vdash$ . Figure 6.10 shows how parallel composition with  $\vdash$  works.

The associativity of sequential composition really just says that the order in which we happen to compose parts sequentially does not

The name  $e$  stands for “empty.”

Stack an empty diagram on top of another diagram. Nothing changes.

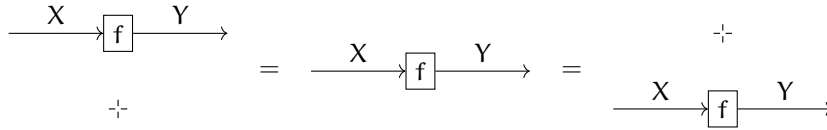


Figure 6.10: Parallel composition with an empty diagram does nothing. So

$$e \parallel f = f = f \parallel e.$$

influence how the resulting assembly works. Likewise, associativity of parallel composition says the order in which we compose parts in parallel does not matter. This should also be the case when we mix sequential and parallel composition. Figure 6.11 illustrates two different orders of assembly involving sequential and parallel composition.

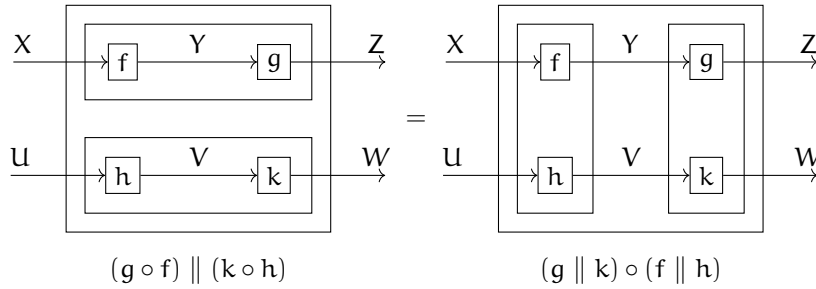


Figure 6.11: Parallel and sequential composition are exchangeable. So

$$(g \circ f) \parallel (h \circ k) = (g \parallel k) \circ (f \parallel h).$$

This is described by saying that parallel composition is *functorial*.

---

#### DEFINITION 17: Strict Monoidal Categories

---

A **strict monoidal category** is a category  $\mathcal{C}$  with the following additional structure:

- The objects of  $\mathcal{C}$  form a monoid  $(\mathbf{Obj}_{\mathcal{C}}, \otimes, \mathbb{I})$ .
- For any two morphisms  $f: W \rightarrow Y$  and  $g: X \rightarrow Z$ , there is a parallel composite component  $f \parallel g: W \otimes X \rightarrow Y \otimes Z$ .
- There is a morphism  $e: \mathbb{I} \rightarrow \mathbb{I}$ .
- Parallel composition is associative and  $e$  is the identity for  $\parallel$  (see Figures 6.9 and 6.10).
- Parallel composition of identities is an identity. That is,  $\text{id}_{X \otimes Y} = \text{id}_X \parallel \text{id}_Y$ .
- Sequential and parallel composition exchange:

$$(g \circ f) \parallel (h \circ k) = (g \parallel k) \circ (f \parallel h).$$

So far, nothing ensures that in a parallel composition, the two parts do not interact. Two parallel components could exchange information or noise, so that operating them separately would not be the same as operating them side-by-side. We need the mathematical equivalent of a shield between  $f$  and  $g$  to make sure that each one is completely isolated from the other in  $f \parallel g$ .

To formalize the shield, first suppose it is always possible to erase a wire. So for each  $X$ , there is a component  $er_X: X \rightarrow \mathbb{I}$ . In a diagram, I will write erasure as  $\xrightarrow{X} \blacklozenge$ . Now if  $f: X \rightarrow Y$  can truly operate on its own, shielded from its surroundings, then erasing the output of  $f$  ought to be the same as not including  $f$  in the first place. Figure 6.12 illustrates the idea.

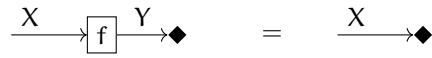


Figure 6.12: Erasing the output of  $f$  is the same as not using  $f$  at all. So,

$$er_Y \circ f = er_X.$$

Also suppose that any object can be forked by copying connectors. So for each  $X$ , there is a component  $cp_X: X \rightarrow X \otimes X$  that splits the wire  $X$ . In diagrams, I denote copying by  $\xrightarrow{X} \begin{smallmatrix} \nearrow \\ \searrow \end{smallmatrix}$ . Figure 6.13 illustrates how copying ought to work. The idea is that if a component  $f$  can operate shielded from its surrounding, then having one instance of  $f$  followed by copying its output should be the same as copying the input and then having two instances of  $f$  operate independently.

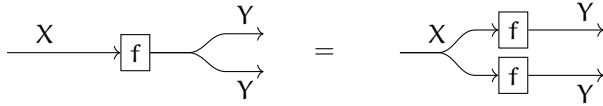


Figure 6.13: Splitting the codomain of  $f$  is the same as splitting the domain followed by  $f \parallel f$ . So,

$$cp_Y \circ f = (f \parallel f) \circ cp_X.$$

The equations illustrated in Figures 6.12 and 6.13 are summarized by saying that  $er$  and  $cp$  are *natural transformations*. This means, essentially, that they behave the same for every object.

Finally, three more principles are needed to capture the idea that components are shielded from one another. Figures 6.14, Figure 6.15, and Figure 6.16 illustrate them.



Figure 6.14: The only way to erase is to use  $er$ . So,

$$h = er_X$$

for any  $h: X \rightarrow \mathbb{I}$ . This is equivalent to requiring

$$er_{\mathbb{I}} = id_{\mathbb{I}}.$$



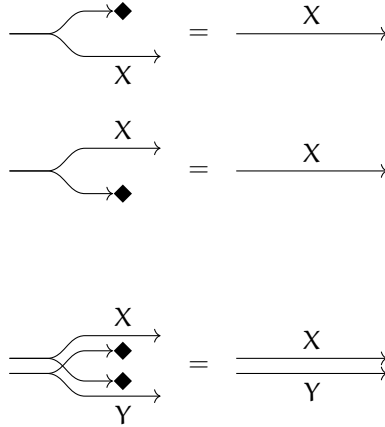


Figure 6.15: Splitting input, then ignoring one copy is the same as not splitting. That is,

$$(er_X \parallel id_X) \circ cp_X = id_X$$

and

$$(id_X \parallel er_X) \circ cp_X = id_X.$$

Figure 6.16: Splitting a parallel input, then ignoring one part of each copy is the same as doing nothing. That is,

$$(id_X \parallel er_Y \parallel er_Y \parallel id_Y) \circ cp_{X \otimes Y} = id_{X \otimes Y}.$$

---

#### DEFINITION 18: Cartesian Strict Monoidal Categories

---

A strict monoidal category is **cartesian** if each object  $X$  has an eraser  $er_X: X \rightarrow \mathbb{I}$  and a copier  $cp_X: X \rightarrow X \otimes X$  that together satisfy all the equations illustrated in Figures 6.12 through 6.16, that is:

- $er_Y \circ f = er_X$ .
  - $cp_Y \circ f = (f \parallel f) \circ cp_X$ .
  - $er_{\mathbb{I}} = id_{\mathbb{I}}$ .
  - $(er_X \parallel id_X) \circ cp_X = id_X$  and  $(id_X \parallel er_X) \circ cp_X = id_X$ .
  - $(id_X \parallel er_Y \parallel er_Y \parallel id_Y) \circ cp_{X \otimes Y} = id_{X \otimes Y}$ .
- 

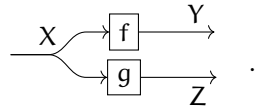


#### EXERCISES:

---

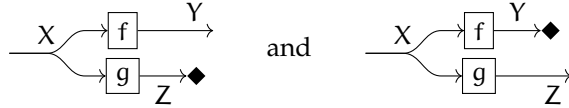
In these exercises, suppose  $\mathcal{C}$  is a cartesian strict monoidal category.

83. Imagine two morphisms  $f: X \rightarrow Y$  and  $g: X \rightarrow Z$ . Note that these have the same domain. Then with copying, we can assemble a single morphism



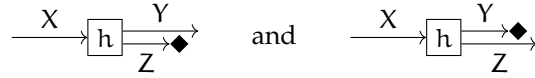
This is an interesting gadget. We can refer to it as  $\langle f, g \rangle$ . By “stubbing off” one or the other of the parts of the codomain, the result is either  $f$  or  $g$ . That is,

In electronics manufacturing, this is a common engineering choice. One device can perform either of two functions.



are exactly  $f$  and  $g$ , respectively.

Now suppose  $h: X \rightarrow Y \otimes Z$  is a morphism that functions the same way. That is,



are  $f$  and  $g$ , respectively. Show that  $h = \langle f, g \rangle$ .

84. Recall that isomorphisms are morphisms that represent “lossless” conversion. Find an isomorphism from  $X \otimes Y$  to  $Y \otimes X$ .

A monoidal category in which every  $X \otimes Y$  is naturally isomorphic to  $Y \otimes X$  is said to be **symmetric**. This exercise shows that any cartesian monoidal category is symmetric.

### Non-strictness

Categories typically arise, not really by design, but because we want to study some situation where “putting things together” makes sense. In “found” categories, we have to take things as they are. In any category, remember that an *isomorphism* is a “lossless converter.” It is a morphism  $f: X \rightarrow Y$  that has an inverse  $f^{-1}: Y \rightarrow X$  so that  $f^{-1} \circ f = \text{id}_X$  and  $f \circ f^{-1} = \text{id}_Y$ . In practice, isomorphisms are important because they describe how two objects are essentially the same. So, when we need to think about laws like associativity, it makes sense to relax the requirement for equality with isomorphisms.

Suppose  $\mathcal{C}$  is a category (not necessarily strictly monoidal). Instead of  $X \otimes (Y \otimes Z) = (X \otimes Y) \otimes Z$  for all objects, it might be the case that  $X \otimes (Y \otimes Z)$  is isomorphic to  $(X \otimes Y) \otimes Z$  for all objects. Then parallel composition still can make perfectly good sense, provided it cooperates with the isomorphisms.

---

### DEFINITION 19: Monoidal Categories

---

A **monoidal category** is a category  $\mathcal{C}$  with the following additional structure:

- The objects of  $\mathcal{C}$  have a binary operation  $\otimes$ , and an object  $\mathbb{I}$ .
- For any two morphisms  $f: W \rightarrow Y$  and  $g: X \rightarrow Z$ , there is a parallel composite component  $f \parallel g: W \otimes X \rightarrow Y \otimes Z$ .
- Sequential and parallel composition exchange:

$$(g \circ f) \parallel (h \circ k) = (g \parallel k) \circ (f \parallel h).$$

- *Parallel composition of identities is an identity. That is,  $\text{id}_{X \otimes Y} = \text{id}_X \parallel \text{id}_Y$ .*
- *There is a morphism  $e: \mathbb{I} \rightarrow \mathbb{I}$ .*
- *For all objects  $X, Y$  and  $Z$ , there is an isomorphism  $\alpha_{X,Y,Z}: (X \otimes Y) \otimes Z \rightarrow X \otimes (Y \otimes Z)$ .*
- *For all objects  $X$ , there is an isomorphism  $\lambda_X: X \otimes \mathbb{I} \rightarrow X$  and an isomorphism  $\rho_X: \mathbb{I} \otimes X \rightarrow X$ .*
- *Parallel composition cooperates with  $\alpha$ ,  $\lambda$ , and  $\rho$  in the sense that for any morphisms  $f: U \rightarrow X$ ,  $g: V \rightarrow Y$ , and  $h: W \rightarrow Z$ ,*

$$\begin{aligned} (f \parallel (g \parallel h)) \circ \alpha_{X,Y,Z} &= \alpha_{U,V,W} \circ ((f \parallel g) \parallel h), \\ (f \parallel e) \circ \rho_U &= \rho_X \circ f, \\ (e \parallel f) \circ \lambda_U &= \lambda_X \circ f. \end{aligned}$$

A monoidal category is **cartesian** if each object  $X$  has an eraser  $er_X: X \rightarrow \mathbb{I}$  and a copier  $cp_X: X \rightarrow X \otimes X$  that together satisfy the equations of Definition 18, although they have to be adapted to account for  $\alpha$ ,  $\lambda$ , and  $\rho$ . I leave that to you.

---

# 7

## *An Overview of Sets, Functions, Predicates, and Relations*

*The laws of nature are written by the hand of God in the language of mathematics.*

— Galileo Galilei (attributed)

THE FUNDAMENTAL BUILDING BLOCKS of contemporary mathematics are **sets**, **functions**, **predicates**, and **relations**. To understand mathematical reasoning (and its very close cousin, computation), we need to understand how these interact. Here is a very rough idea of what we mean.

- Sets correspond to types of mathematical data. For example, the natural numbers form a set that we already encountered and denoted by  $\mathbb{N}$ . Also, the real numbers form a set. The real number matrices form a set. The complex polynomials form a set. The letters of the Latin alphabet form a set.
- Functions correspond to transformations of data. For example, the trigonometric functions  $\sin$  and  $\cos$  transforms an angle  $\theta$  to the corresponding horizontal and vertical displacements  $\sin \theta$  and  $\cos \theta$ . The function  $+$  transforms a pair of natural numbers into a single natural number.
- Predicates are properties of elements of a set. For example, primality is a property of natural numbers. For each natural number  $n$ , either “ $n$  is prime” is true or not: 7 is prime, 8 is not prime. For a function on real numbers, that function is either continuous everywhere or it is not. So continuity is a predicate on the set of all real valued functions.
- Relations permit us to speak precisely about how data of possibly differing types may interact with one another. For example, “less

---

### CHAPTER GOALS

---

In this chapter, we introduce three interrelated foundational concepts: sets, functions, predicates, and relations. The main goal is to provide some basic ways to talk about this things, not to build a complete story.

---

Functions have many other characterizations that we discuss later.

than” is a relation on natural numbers; “solves” is a relation between real numbers and real number polynomial equations (as in “3 solves the equation  $0 = x^2 - x - 6$ ”); “overlaps” is a relation between discs in the plane (some discs overlap each other, others do not). Just like predicates, relations are either true or not for any particular data of the right kind. Relations can involve more than two data. For example, “gave” is what linguists call a *ditransitive verb* because it generally expects a subject, direct object and indirect object. In “Mary gave Joe the skunk eye,” “Mary” is the subject, “Joe” is the indirect object, and “the skunk eye” is the direct object. For our purposes, binary relations are mostly enough.

## Sets

A **set** is, in the formulation of Cantor, *jedes Viele, welches sich als Eines denken lässt* “any multiplicity which can be comprehended as one.” Cantor was getting at the idea that certain collections of objects can be understood as single things. For example, a collection of playing cards taken together can be understood to be a single deck of cards. When a *collection* of individuals can be understood as a single entity in its own right, it is a *set*.

Cantor makes a subtle point here: some collections may not be sets because it is not coherent to take the members of the collection together as a single thing. We will return to this point later.

---

### VOCABULARY 3: Basic Vocabulary of Sets

---

A **set** is an entity  $A$  “consisting” of **elements**. What that means is that for any other entity  $x$ , either  $x$  is an element of  $A$  or  $x$  is not an element of  $A$ . We write  $x \in A$  to indicate that  $x$  is an element of  $A$ , and  $x \notin A$  to indicate that  $x$  is not an element of  $A$ . For variety, all of the following phrases mean the same thing:

- $x$  is an **element of**  $A$ ,
  - $x$  **belongs to**  $A$ ,
  - $x$  is in  $A$ ,
  - $A$  **contains**  $x$ .
  - $x$  is a **member of**  $A$ ,
- 

This vocabulary comes with some obligations. When you write “ $x \in A$ ,” it must be the case that  $A$  denotes a set. For example, the phrase “ $5 \in \text{Julius Caesar}$ ” doesn’t make sense. Julius Caesar is not a set.

To describe a set, we have to provide a precise criterion for membership — an unambiguous way to determine what is in and what is not in. A collection for which the criterion of membership is vague is not a set. For example, “aging professors” does not constitute a set because “aging” is a vague term. On the other hand, “professors at least 45 years old” would constitute a set, assuming that precise age can be pinned down and assuming that professors are mathematical objects.

A set is completely determined by its elements. That is, to say that two sets are equal is to say they have exactly the same elements. This leads us to the following principle.

---

PRINCIPLE 1: Set extensionality

---

*Two sets  $X$  and  $Y$  are equal if and only if for all  $x$ ,*

$$x \in X \text{ if and only if } x \in Y.$$


---

This principle tells us that, in order to give a set, it is enough to list all its elements, if that is possible. For instance, consider the numbers 1, 2, and 7. There is a set whose elements are exactly these three numbers. We denote that set by  $\{1, 2, 7\}$ . These way of denoting sets by listing all its elements is useful, specially for small finite sets.

---

VOCABULARY 4: Notation for small sets

---

*To denote a set consisting of a small finite number of elements,  $x_1, x_2, \dots, x_n$ , we can list the elements between braces, like this*

$$\{x_1, x_2, \dots, x_n\}.$$


---

It is universally understood that, if braces are used instead of brackets or some other punctuation, then the list denotes a set. So for example,  $[3, 4, 5]$  is a *list*, but  $\{3, 4, 5\}$  is the set whose only elements are 3, 4, and 5.

This notation is also used informally to denote bigger sets. For example,  $\{0, 1, 2, \dots, 100\}$  is meant to be the set of natural numbers up to and including 100. For another example,  $\{1, 2, 4, 8, \dots, 2048\}$  denotes the first twelve powers of 2. You need to be careful using the notation this way, because you have to assume your reader will understand what you mean by “...”. Most of the time, it is obvious.

The term “extensionality” is meant to contrast “intensionality.” The latter refers to the *way* something is defined. For example, “spouse of the third president of the US” identifies a particular person; “Dolly Madison” also identifies a particular person. Intensionally, they are distinct because the former describes a person by virtue of standing in a particular relation to someone else, while the latter describes a person directly by naming her. But both descriptions actually refer to the same person. *Extensionally*, they are the same. The principle of set extensionality captures the idea that *how* a set is described is not part of its identity.

But it is a good idea to explain what you mean if there is a chance of misunderstanding. For example,  $\{1, 6, 28, 496, \dots, 2^{4422} \cdot (2^{4423} - 1)\}$  is a very obscure list. I intend it to be the list of the first 20 perfect numbers. But if I did not say that, you'd probably have a hard time figuring how what goes in the "...".

A **perfect** number is a positive natural number that is the sum of its proper factors. For example,  $6 = 1 + 2 + 3$  and  $28 = 1 + 2 + 4 + 7 + 14$ . Only fifty-one numbers so far are known to be perfect. The biggest one has over 80 million digits in base ten.

---

EXAMPLE 16:

---

A standard deck of poker cards can be described by

$$\begin{aligned} \text{Deck} = \{ & \text{A}\clubsuit, 2\clubsuit, 3\clubsuit, 4\clubsuit, 5\clubsuit, 6\clubsuit, 7\clubsuit, 8\clubsuit, 9\clubsuit, 10\clubsuit, \text{J}\clubsuit, \text{Q}\clubsuit, \text{K}\clubsuit, \\ & \text{A}\diamond, 2\diamond, 3\diamond, 4\diamond, 5\diamond, 6\diamond, 7\diamond, 8\diamond, 9\diamond, 10\diamond, \text{J}\diamond, \text{Q}\diamond, \text{K}\diamond, \\ & \text{A}\spadesuit, 2\spadesuit, 3\spadesuit, 4\spadesuit, 5\spadesuit, 6\spadesuit, 7\spadesuit, 8\spadesuit, 9\spadesuit, 10\spadesuit, \text{J}\spadesuit, \text{Q}\spadesuit, \text{K}\spadesuit, \\ & \text{A}\heartsuit, 2\heartsuit, 3\heartsuit, 4\heartsuit, 5\heartsuit, 6\heartsuit, 7\heartsuit, 8\heartsuit, 9\heartsuit, 10\heartsuit, \text{J}\heartsuit, \text{Q}\heartsuit, \text{K}\heartsuit\}. \end{aligned}$$

The elements are arranged here conveniently, but we could just as well have listed the cards in any shuffled order. The set — Deck itself — is the same regardless of how it is shuffled.

We can also describe two other sets: the set of ranks and the set of suits:

$$\begin{aligned} \text{Rank} &= \{\text{A}, 2, 3, 4, 5, 6, 7, 8, 9, 10, \text{J}, \text{Q}, \text{K}\} \\ \text{Suit} &= \{\clubsuit, \diamond, \spadesuit, \heartsuit\}. \end{aligned}$$

As you know, Deck consists of 52 cards: one card for each possible rank and each possible suit. So Deck is, in a sense, built from Rank and Suit by systematically pairing each rank with each suit. This example points to a very general and important idea: If A and B are sets (so they are collections that can be regarded as entities on their own), then the collection consisting of pairs from A and B is also a set. In this example, the deck of cards Deck is built from the set of suits and the set of ranks.

I am not using jokers in my deck.

---

We have already seen how the number 0 and the empty list [] are important artifacts of their respective theories. In the same way, we have to postulate the existence of an *empty set*, which will be very useful to develop set theory.

---

**PRINCIPLE 2:** Existence of an empty set
 

---

*There is a set, called **empty set** and denoted  $\emptyset$ , that does not contain any element.*

---

The principle of extensionality would tell us then that *the empty set is unique*, because every set is determined by its elements, and two empty sets would have exactly the same elements (none) and therefore be equal.

We will introduce ways to construct sets as we need them. For now, we just list some common sets that are needed throughout mathematics.

---

**SOME IMPORTANT SETS**


---

The following sets are denoted by the special symbols:

$\mathbb{N}$  = the set of natural numbers: 0, 1, 2, ...

$\mathbb{N}^+$  = the set of positive natural numbers: 1, 2, 3, ...

$\mathbb{Z}$  = the set of integers: ..., -2, -1, 0, 1, 2, ...

$\mathbb{Q}$  = the set of rational numbers:  $\frac{1}{2}$ ,  $\frac{24}{23}$ , and so on

$\mathbb{R}$  = the set of real numbers

$\mathbb{C}$  = the set of complex numbers

$\emptyset$  = the empty set, having no elements

These symbols are very common. You will be understood by any mathematician if you use them. In addition to these common sets, we define some other sets that are especially useful, but do not have common symbols.

$\mathbf{1} = \{\bullet\}$  — a set containing exactly one element

$\mathbb{B} = \{0, 1\}$  — a set containing exactly two elements

alpha =  $\{\mathbf{a}, \dots, \mathbf{z}\}$

Alpha =  $\{\mathbf{A}, \dots, \mathbf{Z}\}$

digit =  $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}\}$

In the sets alpha, Alpha and digit the elements are intended to be just symbols. For example, *digit 0* is not the same as the *number 0*. The digit **0** does not mean anything. It is just a character that we can distinguish from the characters **1**, **2**, and so on. Likewise, the letters

In this text, we will only rarely refer to complex numbers. Nevertheless, you ought to know the standard symbol.

In handwritten work, a digit can be written with an underline 0, or in quotes "0" to distinguish it from the corresponding number.



of the alphabet are merely characters. So  $x$  is just a letter. That is not the same as a variable  $x$ .

---

### Set comparison

Compare the sets  $\{1, 2, 3, 4, 5\}$  and  $\{1, 5, 4, 2, 3\}$ . Are they distinct? We can easily see that these sets have exactly the same elements, so they are equal. Likewise, the set Deck that we defined earlier is the same set no matter how we shuffle the cards.

But, what is the relation between  $X = \{1, 2, 3\}$  and  $Y = \{1, 3, 5, 4, 2\}$ ? Not every element of the second set is in the first set, for instance  $5 \in Y$  but  $5 \notin X$ . So, this means that  $X \neq Y$ . But every element of  $X$  is in  $Y$ , or as we will say,  $X$  is a *subset* of  $Y$ .

Remember, the idea is that a set is just a collection of things. So elements are either in or out. Nothing more.

---

#### DEFINITION 20: Subsets

---

For sets  $X$  and  $Y$ ,  $X$  is a **subset** of  $Y$  provided that every element of  $X$  is an element of  $Y$ . We write this as  $X \subseteq Y$ . If  $X$  is not a subset of  $Y$ , we write  $X \not\subseteq Y$ . If  $X \subseteq Y$  but  $X \neq Y$ , then  $X$  is called a **proper subset** of  $Y$ , and we write  $X \subsetneq Y$ .

For variety, all of the following phrases and notations mean the same as  $X \subseteq Y$ :

- $X$  is included in  $Y$ ,
  - $Y$  is a superset of  $X$ ,
  - $Y$  includes  $X$ ,
  - $Y \supseteq X$ .
- 

Subset inclusion gives an intuitive meaning to sentences such as “all cats are mammals.” Think of “cats” as denoting the set of all actual cats, and “mammals” as the set of all actual mammals. Then “all cats are mammals” asserts that the set “cats” is a subset of the set “mammals.” To write this using our new notation, it is helpful to remove the quote marks to distinguish the *word* “cats” from the set of cats. So let us write  $Cats$  and  $Mammals$  for the sets. Now “all cats are mammals” asserts that

$$Cats \subseteq Mammals.$$

Like  $\in$ , it is important to understand how  $\subseteq$  is used. It only makes sense to write “ $X \subseteq Y$ ” when both  $X$  and  $Y$  are sets. For example  $\mathbb{N}^+ \subseteq Deck$  makes sense because both  $\mathbb{N}^+$  and  $Deck$  are sets — it is not true, but it makes sense. So never, ever, ever write “something  $\subseteq$  something else” unless both somethings are sets.

Some people write  $X \subset Y$  to mean  $X$  is a subset of  $Y$ . But other people write  $X \subset Y$  to mean  $X$  is a *proper* subset of  $Y$ . That is confusing. It is better to avoid the symbol  $\subset$  completely. So  $\subseteq$  means subset;  $\subsetneq$  means proper subset; and  $\not\subseteq$  means not subset.

More properly,  $Cats$  and  $Mammals$  denote sets of mathematical objects that we are using to model *actual* cats and mammals. As any cat will tell you, cats are not mathematical objects.

---

EXAMPLE 17:

---

Here are some examples and counter-examples of the subset relation.

- $\{1, 2, 3\} \subseteq \{0, 1, 2, 3\}$ .
  - $\{\{1\}\} \not\subseteq \{0, 1\}$  because the only *element* of the first set is  $\{1\}$ , and  $\{1\} \notin \{0, 1\}$ .
  - $\emptyset \subseteq X$  for any set  $X$  because every element of  $\emptyset$  (there are none) is an element of  $X$ .
  - $\{1, 2, 3\} \not\subseteq \{0, 2, 3\}$  because  $1 \in \{1, 2, 3\}$  but  $1 \notin \{0, 2, 3\}$ .
  - $\{1, 2, 3\} \subseteq \{2, 3, 1, 5\}$ .
  - $\{\spadesuit\} \subseteq \text{Suit}$ .
  - $\text{Julius Caesar} \subseteq \mathbb{N}$  is neither true nor false. It is complete nonsense because Julius Caesar is not a set.
- 

Clearly, all cats are cats. In fact, any sentence “all  $X$ s are  $X$ s” is true as long as “ $X$ ” denotes a set. This is common sense, but also follows directly from the definition of  $\subseteq$ , since every element of  $X$  is an element of  $X$ , no matter what set  $X$  is.

Now, let’s reflect on the following statements: All cats are mammals. Also, all mammals are animals. Therefore, all cats are animals. Clearly, this is a general principle. If “all  $X$ s are  $Y$ s” is true and “all  $Y$ s are  $Z$ s” is true, so is “all  $X$ s are  $Z$ s.” Again this is common sense, but also is built into the technical definition of  $\subseteq$ .

This bit of simple reasoning is one of a collection of simple logical arguments that Aristotle called a **syllogism** (συλλογισμός).

---

EXAMPLE 18:

---

What about this argument?

- All natural born comedians are humans.
- All mathematicians are natural born comedians.
- Therefore, all mathematicians are humans.

The argument is completely correct, although the conclusion is doubtful. Clearly, something went wrong in the first two assertions.

---

To summarize this small bit of common sense reasoning, compare  $\subseteq$  to the standard order of the natural numbers  $\leq$ , and the divisibility order of the natural numbers  $|$ , as well as to prefix order of lists

$\subseteq$ . These three relations are partial orders, since they are reflexive, transitive, and antisymmetric relations. We can also see that this is true for  $\subseteq$ .

Remember:  $m \mid n$  means  $m$  divides  $n$ . And for lists,  $L \subseteq L$  means that  $L + M = N$  for some list  $M$ .

*Reflexivity* For any set  $X$ , it is the case that  $X \subseteq X$ .

*Transitivity* For any sets  $X$ ,  $Y$  and  $Z$ , if  $X \subseteq Y$  and  $Y \subseteq Z$ , then  $X \subseteq Z$ .

*Antisymmetry* For any sets  $X$  and  $Y$ , if  $X \subseteq Y$  and  $Y \subseteq X$ , then  $X = Y$ .

Antisymmetry is an immediate consequence of the definition of  $\subseteq$  and the principle of extensionality. Indeed, if  $X \subseteq Y$  and  $Y \subseteq X$ , we are saying that every element of  $X$  is an element of  $Y$  and every element of  $Y$  is an element of  $X$ . That is,  $X$  and  $Y$  have the same elements, which by extensionality, implies that  $X$  and  $Y$  are the same set.

---

EXAMPLE 19:

---

Let  $A = \{1, 2, 4\}$ . Let  $B$  be the set of numbers that solve  $0 = x^3 - 7x^2 + 14x - 8$ . Since  $0 = 4^3 - 7 \cdot 4^2 + 14 \cdot 4 - 8$ , it is the case that  $4 \in B$ . Similarly,  $1 \in B$  and  $2 \in B$ . In other words, every element of  $A$  is an element of  $B$ . So  $A$  is a subset of  $B$ . As you know, a cubic polynomial  $p$  has at most three distinct roots (solutions of  $0 = p(x)$ ). So if  $n \in B$ , then  $n$  must equal 1, 2 or 4 — it can not be any other number. So,  $B$  is a subset of  $A$ . We conclude that  $A = B$ .

---




---

EXERCISES:

---

85. For each of the following pairs of sets, determine whether or not the first is a subset of the second. Explain each answer in one sentence.

- (a)  $\{0, 1\}$  and  $\{1, 0\}$
  - (b)  $\{a, b, c, d\}$  and  $\{a, b, d, e, c\}$
  - (c)  $\emptyset$  and  $\{\emptyset\}$
  - (d)  $\{0, 3, 6, 10\}$  and  $\{10, 9, 8, 7, 5, 4, 2, 1, 0\}$
  - (e)  $\{1, 2, 1\}$  and  $\{1, 2, 3\}$
-

## Functions

Each card in the standard poker deck has a rank and a suit: the *rank* of  $5\heartsuit$  is 5, the *suit* of  $Q\spadesuit$  is  $\spadesuit$ , and so on. Think of *rank* as an attribute of a card, and *suit* as another attribute. If  $c$  is a card, we can write  $\text{rank}(c)$  for that card's rank, and  $\text{suit}(c)$  for its suit, reading  $\text{rank}(c)$  as "the rank of  $c$ ," and similarly for  $\text{suit}(c)$ .

Putting things a bit differently, rank acts like an element of the set Rank that depends on a card. So  $\text{rank}(c) \in \text{Rank}$  for each  $c \in \text{Deck}$ .

More familiar mathematical functions behave the same way. We say "the square of  $x$ " to describe  $x^2$ , because "square of" is an attribute of real numbers — each real number has one.

These simple examples lead us to introduce a general way to talk sensibly about this kind of attribution.

---

### VOCABULARY 5: Basic Vocabulary of Functions

---

*For a set  $A$  and a set  $B$ , a **function from  $A$  to  $B$**  is an entity  $f$  with the feature that each element  $a \in A$  determines an element  $f(a) \in B$ , spoken as "f of a," or sometimes as "f evaluated at a."*

---

In some situations it is convenient (or just traditional) to omit the parentheses. For example, in trigonometry,  $\sin \theta$  is the traditional way to write  $\sin(\theta)$ . Also, for some special purposes,  $f_a$  is an alternative notation for  $f(a)$ .

---

### EXAMPLE 20:

---

The trigonometric functions  $\sin$  and  $\cos$  are indeed functions from  $\mathbb{R}$  to  $\mathbb{R}$  because, every  $\theta \in \mathbb{R}$ , determines  $\sin \theta \in \mathbb{R}$ , and similarly for  $\cos$ . On the other hand,  $\tan$  is *not* a function from  $\mathbb{R}$  to  $\mathbb{R}$  because, for example, there is no such thing as  $\tan(\frac{\pi}{2})$ . To deal with such things,  $\tan$  is sometimes said to be *partial* function. We won't deal with partial functions as a distinct concept in this text. But because partial functions "exist in nature", we will need to make sense of them. We will do that later using other simpler ideas.

You will not have any trouble thinking of many other natural examples of functions from  $\mathbb{R}$  to  $\mathbb{R}$ . For example, any polynomial expression like  $x^2 + 4x - 3$  describes a function.

---

Does  $\frac{1}{x}$  describe a function from  $\mathbb{R}$  to  $\mathbb{R}$ ?

## Rules

To define a function, we must specify a set  $A$ , a set  $B$ , and for each  $x \in A$ , an element  $f(x) \in B$ . If we can come up with a rule for determining  $f(x)$ , we are in business.

For example, in order to define a function  $f$  we can consider  $A = B = \mathbb{N}$  and the rule “for every number  $x$ , raise 2 to the number  $x$  and subtract 1.” We can write this simply as  $f(x) = 2^x \div 1$ . What matters in a definition like this is that the expression on the right must clearly describe a single element of the codomain that may depend on an element  $x$  belonging to domain.

If we were in the business of designing a programming language, we would need to be completely explicit about what is allowed in a rule and what is not allowed. But mathematicians are not typically concerned only with computation. So it is better not to be too fussy in laying down rules for “rules.” The definition of a function  $f$  must be completely clear about which element of the codomain  $f(x)$  is for each  $x$  in the domain. Nevertheless how that clarity is achieved can be left open.

Two important ways to define a function correspond to sequential composition and identity for sequential composition.

---

**DEFINITION 21:** Function composition and identities

---

*Suppose we are given two functions  $f$  from  $A$  to  $B$  and  $g$  from  $B$  to  $C$ . Then we can define a new function  $g \circ f$  from  $A$  to  $C$  by the rule*

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

*For a set  $A$ , we can define a function from  $A$  to  $A$  by the trivial rule*

$$\text{id}_A(x) = x.$$


---

---

**EXAMPLE 21:**

---

Let  $f$  and  $g$  be the functions from  $\mathbb{R}$  to  $\mathbb{R}$  defined by

$$f(x) = x + 1 \quad \text{and} \quad g(x) = x^2.$$

Then  $g \circ f$  is the function  $(g \circ f)(x) = g(x + 1) = (x + 1)^2$ . And likewise,  $(f \circ g)(x) = f(g(x)) = f(x^2) = x^2 + 1$ . In particular,  $(g \circ f)(2) = 9$  and  $(f \circ g)(2) = 5$ .

---

Consider functions  $h$  and  $k$  from  $\mathbb{N}$  to  $\mathbb{N}$  defined by the rules

$$h(n) = (n + 1)^2 \quad \text{and} \quad k(n) = n^2 + 2n + 1.$$

Are these functions equal? The steps of calculation are quite different. In the first, one squares the successor of the input. In the

Is there a difference between defining a function from  $\mathbb{N}$  to  $\mathbb{N}$  by the rule

$$f(x) = 2^x \div 1$$

versus

$$f(n) = 2^n \div 1?$$

I am recycling the notation  $\circ$  and  $\text{id}$  from Chapter 6. You can guess why.

Is the composition of functions commutative?

second, one squares the input, doubles it, adds those two results and then adds one to the sum. So  $h$  and  $k$  present very different calculations. Nevertheless, they always arrive at the same results:  $h(n) = k(n)$  is true for all  $n$ .

So even though the *rules* defining  $h$  and  $k$  are different, the functions themselves behave the same. This leads to a general criterion for equality of functions.

---

PRINCIPLE 3: Function Extensionality

---

For functions  $f$  and  $g$  from set  $A$  to set  $B$ ,

$$f = g \quad \text{if and only if} \quad f(x) = g(x) \text{ for all } x \in A.$$

**Important:** Equality of functions only makes sense when the two functions share the same input set and the same output set.

---

The recycled notation for sequential composition from Chapter 6 is not an accident. The next exercises show why.

---



EXERCISES:

---

86. Suppose  $A, B, C$  and  $D$  are sets, and that  $f$  is a function from  $A$  to  $B$ ,  $g$  is a function from  $B$  to  $C$  and  $h$  is a function from  $C$  to  $D$ .  
Then  $g \circ f$  is a function from  $A$  to  $C$ . So  $h \circ (g \circ f)$  is a function from  $A$  to  $D$ . Similarly,  $(h \circ g) \circ f$  is a function from  $A$  to  $D$ . Prove that  $h \circ (g \circ f) = (h \circ g) \circ f$ .
87. Suppose  $A$  and  $B$  are sets, and that  $f$  is a function from  $A$  to  $B$ .  
Since  $\text{id}_A$  is a function from  $A$  to  $A$ , it is also the case that  $f \circ \text{id}_A$  is a function from  $A$  to  $B$ . Prove that  $f \circ \text{id}_A = f$ . Likewise, prove that  $\text{id}_B \circ f = f$ .
- 

These exercises tell us that sets and functions form a category.

---

DEFINITION 22: The category of sets and functions

---

Let **Set** denote the category in which the objects are sets and for each set  $A$  and  $B$ ,  $\mathbf{Set}(A, B)$  is the collection of all functions from  $A$  to  $B$ . So it is reasonable to use the notation for categories:

- To indicate that  $f$  is a function from  $A$  to  $B$ , write  $A \xrightarrow{f} B$  or  $f: A \rightarrow B$ .
  - For a function  $A \xrightarrow{f} B$ , the set  $A$  is the domain and  $B$  is the codomain.
-

You might wonder whether sets and functions also support a parallel composition. Actually, the category **Set** has finite products. So it is possible to deal with parallel composition.

### *Thunks*

In software design, it is common to define a function that takes no arguments. These special functions are called **thunks**. Thunks are used commonly in graphical user interface designs when, for example, a value needs to be stored for later use in a menu and button system. For example, clicking a certain button may trigger evaluation of a thunk, thus giving the GUI access to a stored value.

Mathematically, a function  $p: \mathbf{1} \rightarrow X$  plays the role of a thunk, where  $\mathbf{1} = \{\bullet\}$  (or some other agreed-upon singleton set). Because input to  $p$  can only be one value ( $\bullet$ ), it does not really depend on anything external. The notation  $p(\bullet)$  simply signals that we wish to evaluate  $p$  to get its “stored” value. In many programming languages, this is written  $p()$  to emphasize that the result does not depend on any actual input.

Each thunk  $p: \mathbf{1} \rightarrow A$  determines an element  $p() \in A$ . Conversely, for any  $a \in A$ , the rule  $\bullet \mapsto a$  determines a thunk.

We denote the thunk determined by  $a$  by  $\hat{a}$ . That is,  $\hat{a}(\bullet) = a$ . And for any thunk  $\widehat{p(\bullet)} = p$ .

### *Predicates and Relations*

In mathematics, we make claims like the following:

- “5 is a multiple of 3” — a claim that is not true.
- “Some prime numbers are even” — a claim that is true.
- “Addition on the natural numbers is associative” — a claim that is true.
- “Every positive natural number has a unique prime factorization” — another claim that is true.
- “5 solves the equation  $x^2 - 6x + 8 = 0$ ” — another claim that is not true.
- “sin is prime” — probably nonsense. Unless there is some other meaning of the word “prime,” functions like sin are just not the sorts of things that can be prime or not. But “sin is an odd function” is a true statement.
- “less 7 than is 5” — what? That’s not grammatical.

All of these except the last follow the grammar of English similar to many other languages, where a simple declarative sentence (the sort of sentence that asserts something to be true) takes the form NP VP, where NP and VP stand for “noun phrase” and “verb phrase.” The noun phrase at the beginning of a sentence is the *subject*. The verb phrase is the *predicate*. For example, “the balloon” is the subject and “is red” is the predicate in the sentence “The balloon is red.” In a statement such as “5 is prime,” “5” is the subject and “is prime” is the predicate.

We do not emphasize grammar in this text, but it is helpful to keep in mind that mathematics is *communicated* using a language (in our case, English). So being aware of how mathematical ideas are expressed grammatically can be useful.

Predicates are usually informally *typed*. For example, “sleeps” is an intransitive verb that stands as a predicate in “John sleeps.” But this predicate only makes sense when it is applied to animate things (humans and animals), or perhaps used analogically to electronics (your computer sleeps).

Most mathematical predicates are likewise typed. The sentence “5 is prime” is intelligible because “is prime” is a predicate on natural numbers. The sentence “The unit circle is prime” or “The function  $\cos$  is prime” are neither false nor true. They are just nonsense, unless we define a new meaning for these expressions.

So we will suppose that a predicate in mathematics comes with a set that is the type of data being predicated.

Sentences that start “There are ...” or “There is ...” are slightly different, and do play a role in mathematics. For example, “There are infinitely many primes” is a true claim. Sentences like this comply with the grammatical rule of NP VP, but the NP “There” is just a place holder. The predicate “are ...” or “is ...” carries all the information about what claim is being made.

Natural language likes a metaphor. So “democracy sleeps” is not a crazy thing to say, even though it can not be understood literally. On the other hand, “5 sleeps” doesn’t make any sense to me at all. A number simply is not the sort of thing that could sleep — I don’t even know how to make sense of it as a metaphor.

---

## VOCABULARY 6: Predicates

---

For a set  $A$ , a **predicate on  $A$**  is an entity  $P$  so that for any element of  $A$ , either  $P$  holds or  $P$  does not hold.

---

Following English grammar, if  $P$  is an predicate on  $A$  and  $x \in A$ , then

$$x P$$

constitutes a declarative statement. For example, “is prime” is a predicate on  $\mathbb{N}$  (or we could say, is a natural number predicate). So “0 is prime,” “1 is prime,” “2 is prime” and so on, are all statements. Only one of the three statements is true, but they are all statements.

For a given set  $A$ , any subset  $B \subseteq A$  determines a predicate “ $\in B$ ” on  $A$ . For every element  $x$  of  $A$ , it is completely determined whether  $x \in B$  or  $x \notin B$ . Take, for instance, the relation “is a cat” on the set of all mammals. For every mammal  $x$ , we can decide whether  $x$  is a cat or not. Formally, “is a cat” is the predicate on the set Mammals given



by the subset Cats, namely, “is a cat” is the predicate “ $\in \text{Cats}$ ” on Mammals.

It is only natural to think that, for any predicate  $P$  on a set  $A$ , we can form the subset of  $A$  consisting of the elements of  $A$  that satisfy  $P$ . As a consequence, for any given set  $A$ , predicates on  $A$  and subsets of  $A$  are interchangeable. This is another set theoretical principle called *specification* or *separation*.

---

**PRINCIPLE 4:** Specification or Subset Separation

---

*Given a set  $A$  and a predicate  $P$  on  $A$ , there is a set consisting of exactly those elements of  $A$  for which  $P$  is true. This is typically written as*

$$\{x \in A \mid x P\}.$$

Remember: This means that for each  $x \in A$ , either  $x P$  is true or  $x P$  is false.

---

This simple and intuitive principle has very interesting and unexpected consequences.

---

**THEOREM 3:** Russell’s Paradox

---

*There is no set containing all sets.*

*Proof.* Suppose that there is a set  $V$  whose elements are all sets and consider the following predicate  $P$  on  $V$ :

“ $x$  is a set that doesn’t belong to itself.”

In symbols, the predicate is written as  $x \notin x$ . Thus, by the Separation Principle, there is a set  $K = \{x \in V : x \notin x\}$  of all the elements of  $V$  that do not belong to themselves. Since the elements of  $V$  are all the sets,  $K$  is the set of all the sets that do not belong to themselves.

Now, we could ask, does  $K$  belong to itself? If  $K$  belongs to itself, then  $K$  should satisfy the predicate  $P$ , that is, we should have that  $K \notin K$ , which is absurd. Therefore, we deduce that  $K$  doesn’t belong to itself. But this means that  $K \notin K$ , and therefore is a set satisfying the predicate  $P$  and hence it should be an element of  $K$ , that is  $K \in K$ , which is in contradiction with the fact that  $K \notin K$ .

Hence, the assumption that there is a set  $V$  containing all sets leads to a contradiction, and therefore we conclude that such a set doesn’t exist. □

---

In the same way that two sets are equal if and only if they have the same elements (by the extensionality principle), two predicates on a

set  $A$  are equal if and only if they are satisfied by the same elements of  $A$ , that is, if and only if they determine the same subset of  $A$ .

---

EXAMPLE 22:

---

Because “is prime” is a predicate on natural numbers, the set

$$\text{Prime} = \{n \in \mathbb{N} \mid n \text{ is prime}\}$$

consists of all the prime numbers. Thus “ $k \in \text{Prime}$ ” and “ $k$  is prime” mean exactly the same thing.

---



---

EXAMPLE 23: Prefix sets of  $\mathbb{N}$

---

Each  $n \in \mathbb{N}$  determines a predicate “ $< n$ ” on natural numbers  $m < n$  is either true or false for each  $m$ . So  $n$  determines a set  $\{m \in \mathbb{N} \mid m < n\}$ . For example,  $\{m \in \mathbb{N} \mid m < 5\} = \{0, 1, 2, 3, 4\}$ .

Notice that the set defined by  $n$  has exactly  $n$  elements. This is a handy idea. If we need a finite set of particular size  $n$ , we can just use this one. Even though this is such a useful idea, mathematicians have not agreed on a standard notation for it. In the rest of this text, I will write

$$\underline{n} = \{m \in \mathbb{N} \mid m < n\}.$$

More concretely,  $\underline{0} = \emptyset$ ,  $\underline{1} = \{0\}$ ,  $\underline{2} = \{0, 1\}$ , and so on.

---

Now consider a verb such as “equals” or a coordinated phrase “less than or equal to” (strictly speaking the latter is not a verb phrase, but it functions as one in mathematics). These require an additional datum (an object). So it makes sense to assert “ $2 + 3$  equals 5” because “equals 5” constitutes a predicate on natural numbers, and “ $2 + 3$ ” denotes a natural number.

The verb “equals” is said to be **transitive**, in contrast to “sleeps,” which is **intransitive**. A transitive verb requires an **object** to form a complete predicate. For example, “5” is the object of “equals” in “equals 5.” English also has **ditransitive** verbs like “gives” that require a subject, a direct object and an indirect object as in “Susan gives James a book.” There is no need to deal with ditransitives explicitly here because they can be reduced to simpler cases with other mechanisms that we need anyway. We will return to that later.

A *binary relation* specifies the meanings of words and phrases like

The mathematician and physicist John von Neumann went a step farther and proposed to identify natural numbers with these sets. That is, 0 is  $\emptyset$ , 1 is  $\{0\} = \{\emptyset\}$ , 2 is  $\{0, 1\} = \{\emptyset, \{\emptyset\}\}$ , 3 is  $\{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}$ , and so on.

To deal with ditransitive verbs, we could talk about relations that are *ternary* instead of binary. That turns out to be unnecessary, so from now on the term “relation” will always mean “*binary* relation.”

“equals” or “is less than or equal to” or “solves” that behave like transitive verbs. Binary relations, just like predicates, are typed.

---

#### VOCABULARY 7: Binary Relations

---

For sets  $A$  and  $B$ , a **binary relation from  $A$  to  $B$**  is an entity  $R$  that determines a predicate on  $A$  for every element of  $B$ . That is, for each  $y \in B$ ,  $R$  determines an  $A$ -predicate “ $R y$ .” We write  $x R y$  if  $x (R y)$  is true and  $x \not R y$  if  $x (R y)$  is false.

Informally, think of  $A$  is the *subject* and  $B$ , the *object* of  $R$ .

To make things more concise, we sometimes say  $R$  is an  $(A, B)$ -**relation**. Also, a **binary relation on  $A$**  is a binary relation from  $A$  to  $A$ .

---



---

#### EXAMPLE 24:

---

“Is less than or equal to” is a binary relation *on*  $\mathbb{N}$ , in symbols, “ $m \leq n$ ”. For any  $n \in \mathbb{N}$ , “is less than or equal to  $n$ ” is a predicate on natural numbers, because for any  $n$  and  $m$ , “ $m$  is less than or equal to  $n$ ” is either true or false.

Analogously, “is a divisor” is also a binary relation on  $\mathbb{N}$ , in symbols “ $m \mid n$ ”, for the same reason. That is, for any  $n \in \mathbb{N}$ , “is a divisor of  $n$ ” is a predicate on natural numbers, since for every  $m \in \mathbb{N}$ , “ $m$  is a divisor of  $n$ ” is either true or false.

---

For each set  $A$ , we can define two important binary relations on  $A$ : the *diagonal* and the *total* relations on  $A$ . Notice that, “equals  $\frac{628}{200}$ ” and “equals  $\pi$ ” are predicates on the real numbers. And so “ $\frac{314}{100}$  equals  $\frac{628}{200}$ ” and “ $\frac{314}{100}$  equals  $\pi$ ” are statements. One is false; one is true. There are various ways to denote “equality restricted to a set  $A$ .” One common bit of notation is to write it as  $=_A$  to distinguish the *relation* on  $A$  from general equality. As for the total relation on  $A$ , this is the binary relation according to which every element of  $A$  is related to every other element of  $A$ .

---

**DEFINITION 23:** Total and diagonal relations
 

---

Given a set  $A$ , the **diagonal** of  $A$  is the binary relation  $\Delta_A$  on  $A$  determined by

$$x \Delta_A y \text{ if and only if } x \text{ and } y \text{ are the same element.}$$

The **total** relation on  $A$  is the binary relation  $\nabla_A$  determined by

$$x \nabla_A y \text{ if and only if } x, y \in A.$$


---

We can define a composition operation on relations.

---

**DEFINITION 24:** Relational composition
 

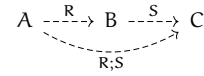
---

Suppose  $A$ ,  $B$  and  $C$  are sets, and  $R$  is a relation from  $A$  to  $B$ , and  $S$  is a relation from  $B$  to  $C$ . Then we may define a new relation  $R;S$  from  $A$  to  $C$  by

$$x R;S z \text{ if and only if there is some } y \in B \text{ such that } x R y \text{ and } y S z.$$


---

The terminology “diagonal” comes from one way to represent relations. In that representation,  $\Delta_A$  is represented as a diagonal matrix. We will look at matrix representation of relations later. We also denote  $\Delta_A$  by  $=_A$ .




---

**EXAMPLE 25:**


---

Suppose that  $A = \{a, b, c\}$ ,  $B = \{1, 2, 3\}$ ,  $R$  is the relation from  $A$  to  $B$  determined by

$$a R 1, \quad a R 2, \quad b R 2, \quad b R 3, \quad c R 2,$$

and  $S$  is the relation from  $B$  to  $A$  determined by

$$1 S b, \quad 2 S c, \quad 3 S a.$$

Then,  $R;S$  is a binary relation on  $A$ , while  $S;R$  is a binary relation on  $B$ . We can see that, for example,  $a R;S b$ , because  $a R 1$  and  $1 S b$ , and  $1 S;R 2$ , since  $1 S b$  and  $b R 2$ . On the other hand, it is not true that  $a R;S a$ , because there is no element  $y \in B$  so that  $a R y$  and  $y S a$ . Analogously, it is not true that  $1 S;R 1$ , because there is no element  $y \in A$  so that  $1 S y$  and  $y R 1$ .

---

---

EXAMPLE 26:

Suppose  $H$  is a set corresponding to humans, and parent-of is a relation on  $H$  corresponding to parentage:  $x$  parent-of  $y$  means that  $x$  is a parent of  $y$ . Then parent-of ; parent-of is the grandparent relation. That is,

$$x \text{ parent-of ; parent-of } z$$

means that  $x$  is the parent of someone who is the parent of  $z$ . In other words,  $x$  is a parent of a parent of  $z$ .

---



---

EXAMPLE 27:

Suppose we have defined a set  $\mathbb{R}[x]$  that consists of all polynomial functions having real coefficients. So  $p = x^2 - 3x + 2$  is in  $\mathbb{R}[x]$ , and so on. Then we can define an relation between  $\mathbb{R}$  and  $\mathbb{R}[x]$  named solves so that

$$a \text{ solves } p \quad \text{if and only if} \quad p(a) = 0.$$


---

The relation solves from  $\mathbb{R}$  to  $\mathbb{R}[x]$  determines the relation from  $\mathbb{R}[x]$  to  $\mathbb{R}$  using the passive voice: that is “ $a$  solves  $p$ ” if and only if “ $p$  is solved by  $a$ .” In a similar manner, the parent-of relation on  $H$  also allows us to define a child-of relation.

---



---

DEFINITION 25: Converse relations

Given a relation  $R$  from  $A$  to  $B$ , the *converse* of  $R$  is the relation  $R^\circ$  from  $B$  to  $A$  satisfying

$$y R^\circ x \quad \text{if and only if} \quad x R y.$$


---

Now the child-of relation is simply the converse of parent-of, that is,  $\text{child-of} = \text{parent-of}^\circ$ . Notice also that for any given binary relation  $R$  from a set  $A$  to another set  $B$ , then  $R^{\circ\circ} = R$ .

---

EXAMPLE 28:

The converse of “less than” is “greater than” because  $n$  is greater than  $m$  if and only if  $m$  is less than  $n$ . It is common to use reversed symbols to denote converses. That is, we use the symbol  $\geq$  for  $\leq^\circ$ , the symbol  $>$  for  $<^\circ$ , the symbol  $\succ$  for  $\prec^\circ$ , and so on.

---

The grandparent relation is defined by parent-of ; parent-of. With converses and composition, we can also encode other familial relations. For example, siblings are people who share a parent. So,

$$x \text{ parent-of}^\circ ; \text{parent-of } y$$

means precisely that a parent of  $x$  is also parent of  $y$ . That is,  $x$  and  $y$  are siblings if  $x$  is a child of a parent of  $y$ .

Cousins are two people whose parents are siblings. I’ll leave it to you to sort that one out.

The correspondence between a relation  $R$  and its converse  $R^\circ$  is, frequently, the same as the correspondence between active and passive voice. For example, “Milton eats bamboo” and “Bamboo is eaten by Milton” express the same relation between an animal (Milton) and a food (bamboo). So “is eaten by” is the converse relation of “eats.” This is only roughly true in the simplest situations. Consider the two sentences “Mathematicians tell the best jokes” versus “The best jokes are told by mathematicians.” They do not quite mean the same thing to me.

This definition has a bug. According to it, anyone who has a parent (anyone but Adam or Eve, I suppose), is a “self-sibling.”



---

EXERCISES:

88. How is “cousin-of” defined in terms of parent-of.
  89. For any relation  $R$  from  $A$  to  $B$  and any relation  $S$  from  $B$  to  $C$ , show that  $(R ; S)^\circ = S^\circ ; R^\circ$ .
  90. Show that sets with binary relations forms a category by showing that relational composition behaves as a form of sequential composition, and that the relations  $\Delta_A$  are identities for composition. That is, show that
    - (a)  $R ; (S ; T) = (R ; S) ; T$  for any relation  $R$  from  $A$  to  $B$ ,  $S$  from  $B$  to  $C$ , and  $T$  from  $C$  to  $D$ ;
    - (b)  $\Delta_A ; R = R = R ; \Delta_B$  for any relation  $R$  from  $A$  to  $B$ .
- 

The facts you just checked show that sets together with binary relations between them constitute another category, different from **Set**.

---

DEFINITION 26: The category of sets and relations

---

Let **Rel** denote the category in which the objects are sets and for each pair of sets  $A$  and  $B$ ,  $\mathbf{Rel}(A, B)$  is the collection of all relations from  $A$  to  $B$ . To distinguish relations from functions, we use an alternative arrow and slightly different terminology.

- To indicate that  $R$  is a relation between  $A$  and  $B$ , write  $R: A \dashrightarrow B$  or  $A \overset{R}{\dashrightarrow} B$ .
  - For a relation  $A \overset{R}{\dashrightarrow} B$ , the set  $A$  is the source and  $B$  is the target.
- 

### Chickens and eggs

Though we have discussed the concepts of sets, functions, predicates, and relations in this chapter, you may have noticed that it looks like we needed sets first, just to make sense of domain and codomain of functions, and to make sense of types of predicates and relations. In fact, with a bit more development, we can interpret relations as being special kinds of sets and functions as being special kinds of relations. In that sense, we only need sets. Most mathematicians over more than the last century and a half have been taught very explicitly to think this way.

If our main concern is to limit the number of different kinds of things in mathematics, then a sets-first view is pretty reasonable. We get an attractive economy of ideas by supposing that everything is built from sets.

But it turns out that, with a bit more effort, we could also have taken a relations first view, defining functions and sets as special kinds of relations. Or we could have taken functions to be basic, defining relations and sets in terms of functions.

Historically, the sets-first view has been dominant (I think for justifiable reasons). But in this chapter by putting sets, functions, predicates, and relations on more an even footing, we can see their distinct roles better. In later chapters, we will have chances to concentrate on one or the other.

# 8

## *Injectons and Surjections*

THE SET  $\{N, S, E, W\}$  CORRESPONDS to the cardinal compass directions. The set  $\{\mathbf{top}, \mathbf{bottom}, \mathbf{left}, \mathbf{right}\}$  corresponds to the edges of a page (or display). Any time you consult a map, you use the convention that the top of the map corresponds to N, the left edge to E, and so on. The edges of a page are *not* compass directions. But we have agreed to use them as if they were.

This correspondence is, mathematically, a function:

$$\begin{aligned} d(N) &= \mathbf{top} \\ d(S) &= \mathbf{bottom} \\ d(E) &= \mathbf{left} \\ d(W) &= \mathbf{right} \end{aligned}$$

The set  $\mathbf{alphabet} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$  is not a subset of  $\mathbb{N}$ . But we can pretend it is. For example, we can number the letters of the alphabet:  $\mathbf{a}$  is letter 0,  $\mathbf{b}$  is letter 1,  $\mathbf{c}$  is letter 2 and  $\mathbf{d}$  is letter 3, and so on. This assignment of numbers to letters is a *ord* from  $\mathbf{alphabet}$  to  $\mathbb{N}$ :  $\text{ord}(\mathbf{a}) = 0$ ,  $\text{ord}(\mathbf{b}) = 1$  and so on. So even though  $\mathbf{alphabet}$  is not a subset of  $\mathbb{N}$ , the set

$$\{\text{ord}(\mathbf{a}), \text{ord}(\mathbf{b}), \text{ord}(\mathbf{c}), \dots, \text{ord}(\mathbf{z})\}$$

is. In a sense, *ord* describes how an “image of  $\mathbf{alphabet}$ ” appears in  $\mathbb{N}$ . This is useful because it allows us to say what alphabetical order means. For example,  $\text{ord}(\mathbf{c}) \leq \text{ord}(\mathbf{f})$  is the same as saying  $2 \leq 5$ . But it encodes what we want:  $\mathbf{c}$  occurs before  $\mathbf{f}$  in the alphabet.

This example suggests that certain functions can be employed to represent how one set  $X$  may be embedded in another set  $Y$  by “renaming” the elements of  $X$ . What matters is that this renaming does not assign the same name to different elements of  $X$ . In the alphabet example, it would not be helpful to have defined  $\text{ord}(\mathbf{a}) = 0$  and  $\text{ord}(\mathbf{b}) = 0$ .

“ord” stands for *ordinal*.



## Injections and Surjections

---

### DEFINITION 27: Monomorphisms

---

A **monomorphism** is a function  $m: X \rightarrow Y$  so that for any two functions  $f, g: W \rightarrow X$ , if  $m \circ f = m \circ g$  then  $f = g$ . We say that  $m$  cancels on the left.

We will sometimes use a special arrow and write  $m: X \hookrightarrow Y$  to indicate that  $m$  is a monomorphism.

---

The definition of monomorphism only involves composition. So *monomorphism* is a concept that makes sense in any category.

Suppose  $m: X \hookrightarrow Y$  is a monomorphism. Consider  $y \in Y$ . It is certainly possible for  $m(x) = y$  to be true for some  $x$ . But if  $m(x) = y$  and  $m(x') = y$ , then  $m \circ \hat{x} = m \circ \hat{x}'$ . So  $x = x'$ . In other words,  $m(x) = y$  is true for at most one  $x \in X$ . Let us make that condition into another definition.

---

### DEFINITION 28: Injective functions

---

An **injection** is a function  $h: X \rightarrow Y$  so that for any  $y \in Y$ , there is at most one element  $x \in X$  satisfying  $h(x) = y$ . An injection is also said to be **one-to-one**, or **injective**.

---

In the paragraph above, we argued that a monomorphism in **Set** is necessarily also an injection. The next exercises address the converse.

---



Unlike the definition of monomorphism, the definition of injective functions involves elements of sets. So it is special to **Set** and related categories.

The defining condition for an injective function can be stated another way. Suppose  $h$  is injective, and consider  $x$  and  $x'$  so that  $h(x) = h(x')$ . Since  $h(x')$  is an element of  $Y$ , there is at most one element of  $X$  satisfying  $h(x) = h(x')$ . So  $x = x'$ . That is, injectivity is equivalent to saying that  $h(x) = h(x')$  implies  $x = x'$ .



### EXERCISES:

---

91. Write out your own proof that if  $m: X \rightarrow Y$  is a monomorphism, then it is a one-to-one.
  92. Show that if  $m: X \rightarrow Y$  is one-to-one, then it is a monomorphism.
- 

These observations are useful enough to warrant declaring a theorem that we can cite in the future.

---

THEOREM 4: Monomorphisms in **Set** are injections

---

*A function is a monomorphism if and only if it is an injection.*

*Proof.* The proof is in the previous paragraphs and Exercise 93.  $\square$

---



---

EXAMPLE 29:

---

The doubling function  $d: \mathbb{N} \rightarrow \mathbb{N}$ , defined by  $d(n) = 2 \cdot n$  is one-to-one. Namely, suppose  $d(m) = d(n)$ . Then  $2 \cdot m = 2 \cdot n$ . But multiplication by 2 is cancellative. So  $m = n$ . This shows  $d$  is one-to-one, and so we know it is also monomorphism.

The successor function  $\text{suc}: \mathbb{N} \rightarrow \mathbb{N}$  defined by  $\text{suc}(n) = n^{\frown}$  is a monomorphism because this was one of our postulates characterizing the natural numbers:  $m^{\frown} = n^{\frown}$  implies  $m = n$ .

---

Now we know that a function  $m$  is *cancellable on the left* if and only if it is one-to-one. This has the practical meaning that  $m$  produces a copy of  $X$  in  $Y$ . A special case of this happens when  $X \subseteq Y$ . Then the inclusion function  $\text{incl}_{X,Y}: X \rightarrow Y$  defined by

$$\text{incl}_{X,Y}(x) = x$$

is certainly a monomorphism. It tells us explicitly how  $X$  is sitting inside  $Y$ .

---



EXERCISES:

---

93. Show that the squaring function  $\text{sqr}: \mathbb{N} \rightarrow \mathbb{N}$  given by  $n \mapsto n^2$  is a monomorphism. [Hint: Suppose  $\text{sqr}(m) = \text{sqr}(n)$ . Without loss of generality, also suppose  $m \leq n$ , so that  $m + d = n$  for some natural number  $d$ . Now show that  $d = 0$ , hence  $m = n$ .]
  94. Show that the squaring function  $\text{sqr}': \mathbb{Z} \rightarrow \mathbb{Z}$  defined on all integers is not a monomorphism.
- 

So what does cancellability on the right mean?

---

DEFINITION 29: Epimorphisms

---

A function  $e: X \rightarrow Y$  is an **epimorphism** if it cancels on the right — that is, for any  $f: Y \rightarrow Z$  and  $g: Y \rightarrow Z$ , if  $f \circ e = g \circ e$ , then  $f = g$ .

We sometimes use the special arrow  $e: Y \twoheadrightarrow X$  to indicate that  $e$  is an epimorphism.

---



Like monomorphisms, epimorphisms are defined in a way that makes sense in any category.

Without examples, it is not immediately clear that there are any interesting epimorphisms. Obviously, every identity function is an epimorphism:  $f \circ \text{id}_X = g \circ \text{id}_X$  implies  $f = g$ . But what about other examples? It will be easier to find such examples if we can find a condition involving elements, similar to injectivity, that is equivalent to being an epimorphism.

---



---

DEFINITION 30: Surjective functions

---

A **surjection** is a function  $h: X \rightarrow Y$  so that for any  $y \in Y$ , there is at least one element  $x \in X$  satisfying  $h(x) = y$ . A surjection is also said to be **onto**, or **surjective**.

---



It is really important that you compare this definition carefully to the definition of injectivity. They are very closely related.

So, for a function  $h: X \rightarrow Y$ , *injectivity* means that every  $y \in Y$  has at most one  $x \in X$  satisfying  $f(x) = y$ ; *surjectivity* means that every  $y \in Y$  has at least one  $x \in X$  satisfying  $f(x) = y$ .

---



---

EXAMPLE 30:

---

The function  $\text{cube}: \mathbb{R} \rightarrow \mathbb{R}$  defined by  $\text{cube}(x) = x^3$  is surjective because every real number  $y$  has a cube root  $\sqrt[3]{y}$ , and  $\text{cube}(\sqrt[3]{y}) = y$ . So there is at least one possible input that produces  $y$ .

The function  $\text{sqr}: \mathbb{R} \rightarrow \mathbb{R}$  defined by squaring instead of cubing is not surjective. For example,  $-1$  is not equal to any real number squared. So  $-1$  does not have at least one  $x$  satisfying  $\text{sqr}(x) = -1$ . One way to look at the invention of complex numbers is that they extend the real numbers to a bigger system of numbers in which  $\text{sqr}$  actually is surjective.

---

This leads to an obvious possibility. Perhaps epimorphisms and surjections are the same, just as monomorphisms and injections are.

---

**THEOREM 5:** The epimorphisms in **Set** are the surjections.

---

*A function is a epimorphism if and only if it is a surjection.*

*Proof.* Suppose  $h: X \rightarrow Y$  is a surjection. Consider two functions

$$Y \begin{matrix} \xrightarrow{f} \\ \xrightarrow{g} \end{matrix} Z$$

so that  $f \circ h = g \circ h$ . We would like to show that  $f = g$ . But by function extensionality (Principle 3), that is the same as showing that  $f(y) = g(y)$  for every  $y \in Y$ . Now consider some  $y \in Y$ . Because  $h$  is a surjection, there is at least one  $x \in X$  satisfying  $h(x) = y$ . Hence  $f(y) = f(h(x))$  and  $g(y) = g(h(x))$ . We assumed that  $f \circ h = g \circ h$ . So  $f(y) = g(y)$ . Thus we have shown that if  $h$  is a surjection, it is an epimorphism.

To prove the converse is a bit trickier. We will prove the contrapositive statement — if  $h: X \rightarrow Y$  is *not* a surjection, then it is *not* an epimorphism. Suppose  $h$  is not a surjection. That is, there is some  $y_0 \in Y$  so that  $f(x) \neq y_0$  for every  $x \in X$ . Now consider the set  $\underline{2} = \{0, 1\}$ . Define  $f: Y \rightarrow \underline{2}$  to be the constant function  $f(y) = 1$ . Define  $g: Y \rightarrow \underline{2}$  by

$$g(y) = \begin{cases} 0 & y = y_0 \\ 1 & \text{otherwise.} \end{cases}$$

Clearly  $f \neq g$ , but both  $f \circ h(x) = 1$  and  $g \circ h(x) = 1$  for every  $x \in X$ . So  $h$  is not an epimorphism.  $\square$

---

You have been using a particular surjection for many years, whenever you have read or written a numeral in base ten notation. A **numeral** consists of a list of digits, as in **20671**. But you understand that numeral to represent a certain natural number. Every numeral corresponds to a number. That is, there is a function from numerals to numbers. And this function has better be a surjection. Otherwise, there would be natural numbers that could not be written down as numerals.

Consider the set  $\text{Digit} = \{0, 1, 2, \dots, 9\}$ . This consists of the usual *characters* we use for digits. There is an obvious one-to-one function  $v: \text{Digit} \rightarrow \mathbb{N}$  that assigns the actual numerical value corresponding to each digit. So  $v(0) = 0$ ,  $v(1) = 0^\wedge$ , and so on.

A **base ten numeral** is essentially just a list of digits. So **572** is not a natural number, but is the list of digits **[5, 7, 2]**. To interpret this as a number, we need a function sending numerals to numbers.

I am avoid negative numbers and decimal points.

The name  $v$  stands for “value”.

We can make this precise by defining a function  $\text{val}: \text{List}[\text{Digit}] \rightarrow \mathbb{N}$  that supplies the numerical value corresponding to a numeral. This is a situation where it is more convenient to suppose that a list of digits (a numeral) is built using  $\frown$ , so that **572**, represented by the list  $[5, 7, 2]$  is built up as  $[5, 7] \frown 2$ .

---

ALGORITHM 15: The value of a base ten numeral

---

Now define  $\text{val}: \text{List}[\text{Digit}] \rightarrow \mathbb{N}$  by recursion on lists:

$$\begin{aligned}\text{val}([\ ])&= 0 \\ \text{val}(K \frown d)&= 10 \cdot \text{val}(K) + v(d)\end{aligned}$$


---

Now we compute an example:

$$\begin{aligned}\text{val}([5, 7, 2]) &= 10 \cdot \text{val}([5, 7]) + v(2) \\ &= 10 \cdot (10 \cdot \text{val}([5]) + v(7)) + v(2) \\ &= 10 \cdot (10 \cdot (\text{val}([\ ]) + v(5)) + v(7)) + v(2) \\ &= 10 \cdot (10 \cdot (0 + v(5)) + v(7)) + v(2) \\ &= 10 \cdot (10 \cdot (0 + 5) + 7) + 2 \\ &= 572\end{aligned}$$

The function  $\text{val}$  is surjective. That is, for any actual natural number  $n$ , there is a numeral  $N \in \text{List}[\text{Digit}]$  so that  $n = \text{val}(N)$ . The proof of surjectivity involves the division and remainder functions from Chapter 3.

Notice that  $\text{val}$  is not an injection. For example, **025** and **25** are assigned the same value. This corresponds to what you already know: leading zeroes do not affect the value of a numeral.



#### EXERCISES:

---

95. Write out the calculation of  $\text{val}(\mathbf{3921})$  following the example in the previous paragraph.
  96. Show that the successor function on the natural numbers is not an epimorphism.
  97. Show that the addition function  $\text{add}: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  is an epimorphism.
  98. Consider some function  $f: \mathbb{N} \rightarrow \mathbb{N}^{\mathbb{N}}$ . Show that  $f$  is not an epimorphism. Hint: Find an element  $d \in \mathbb{N}^{\mathbb{N}}$  (namely, a function from  $\mathbb{N}$  to  $\mathbb{N}$ ) so that  $f(n) \neq d$  for every  $n \in \mathbb{N}$ .
-

## Bijections

When you were a child, you played matching games. For example, maybe you were shown pictures of an orange, a lemon and an apple on one side of the page, and the words “red”, “yellow” and “orange” on the other. Then you drew a line from a picture to a word. You were constructing a “matching” between the pictures and the words.

---

### DEFINITION 31: Matching

---

For two sets  $A$  and  $B$ , a **matching** consists of two functions  $f: A \rightarrow B$  and  $g: B \rightarrow A$  so that

$$f(x) = y \quad \text{if and only if} \quad x = g(y).$$


---

Suppose  $f: A \rightarrow B$  and  $g: B \rightarrow A$  constitute a matching. Because  $f(x) = f(x)$  is always true,  $x = g(f(x))$  holds for all  $x \in X$ . In other words,  $g \circ f = \text{id}_X$ . Likewise,  $f \circ g = \text{id}_Y$ . So apparently, matchings are isomorphisms (see Definition 16 for a reminder).

Conversely, suppose  $g \circ f = \text{id}_X$ . Consider some  $x \in X$  and  $y \in Y$ . If  $f(x) = y$ , then  $y = g(f(x)) = g(y)$ . In other words, if  $g \circ f = \text{id}_X$ , then  $f(x) = y$  implies  $x = g(y)$ . Likewise, if  $f \circ g = \text{id}_Y$ , then  $x = g(y)$  implies  $f(x) = y$ .

Putting these observations together,  $f$  and  $g$  constitute a matching if and only if they are inverse isomorphisms for each other. Those matching games you played when you were young were teaching you how isomorphisms work in sets.

---

### DEFINITION 32: Bijections

---

A function  $f: X \rightarrow Y$  is a **bijection** provided it is both an injection and a surjection. That is, for each  $y \in Y$ , there is exactly one  $x$  so that  $f(x) = y$ .

A bijection is also called a **one-to-one, onto function**.

---



### EXERCISES:

---

99. Show that, in the category **Set**, any isomorphism is a bijection.
- 

The converse is also true, but we do not yet have the machinery to prove it yet.

## 9

# The Size of Sets

*The cardinality of a set is what remains, if we make an abstraction with regard to the individual characters of its elements.*

— Georg Cantor

## The Size of Sets

It seems reasonable that if we want to know how many members there are in a given collection, all we have to do is to *count* them. But, what does it mean “to count” an infinite set? When we count a finite collection of things, we go through each one of those things as we recite a well-established sequence of words, which in English starts with “one, two, three, four, five, . . .” If we think about it, all we are doing is assigning to each member of our collection exactly one element of that sequence. By the end of that process, if successful, we would have gone through all the members exactly once, and we’ll say that the number of elements is “seventeen,” or “one thousand twenty four,” or whatever was the last element of our sequence of numbers that we used.

But, why does this method work? Well, first notice that no two elements of the sequence of numbers are the same. Moreover, if  $x$  is the last element of the sequence that we use, then every member of the collection is assigned one element in the sequence up to  $x$ , in a way so that no two members of the collection are assigned to the same element in the sequence. That is, we establish a matching between the members of the collection and the elements of the sequence up to  $x$ . Finally, we use a certain mental association, learnt in school, of each one of the elements of the sequence with a particular notion of the “quantity” that it represents. So, we conclude that the number of members in our collection is *as numerous* as the element  $x$  is intended to represent.

There are, therefore, two distinct parts in the process of counting the members of a collection  $A$ : one purely mechanical, which consists

---

### CHAPTER GOALS

---

We answer the question “when do two sets have the same number of elements?” We call two sets with the same number of elements *equipollent*. We prove that there are fundamentally two different kind of sets: those that are equivalent to some of its proper parts and those that are not, which gives a clearcut between infinite and finite sets.

Moreover, we show that not two every infinite sets are equipollent, that is, there are different types of infinities, some of them larger than others! In particular, we show that  $\mathbb{N}$ ,  $\mathbb{Z}$ , and  $\mathbb{Q}$  are equipollent, while  $\mathbb{N}$  and  $\mathbb{R}$  are not.

Finally, we compute the size of the powerset of any given finite set.

---

of establishing a certain matching between  $A$  and a well-known set  $\{1, 2, \dots, n\}$ , for some particular  $n$ , and another one purely mental, of recognizing the quantity that the element  $n$  (the last element of the sequence used) represents.

We saw in the last chapter the right tool to formalize the first part of this process, namely, bijections. Thus, two sets  $A$  and  $B$  will have the same number of elements if there is a bijection between them, that is, a matching of every element of  $A$  with exactly one element of  $B$  in a way that every element of  $B$  is also matched to exactly one element of  $A$ .

---

**DEFINITION 33:** Equipollent Sets

---

Two sets  $A$  and  $B$  are **equipollent** or **equinumerous**, and we denote it by  $A \simeq B$ , if there is a matching between them, that is, if there is a bijection  $f: A \rightarrow B$ .

---

We can immediately see that this produces a classification of all sets according to their sizes.

---

**LEMMA 12:** Equipollence is an equivalence relation.

---

The following properties hold for any given sets  $A$ ,  $B$ , and  $C$ :

1.  $A \simeq A$ ,
2.  $A \simeq B$  and  $B \simeq C$  imply that  $A \simeq C$ ,
3.  $A \simeq B$  implies that  $B \simeq A$ .

*Proof.* These properties are an immediate consequence of the fact that there  $\text{id}_A: A \rightarrow A$  is a bijection, the composition of bijections is a bijection, and that if  $f: A \rightarrow B$  is a bijection, its inverse  $f^{-1}: B \rightarrow A$  is also a bijection. □

---

Thus, we have already established (or perhaps more precisely, recognized) that two sets have the same number of elements if they are equipollent. But still, we would like to answer the question of how many elements a set has, that is, what is the *cardinality* of a set. Thus, the cardinality of a set  $A$  should be a particular set, denoted  $|A|$ , that is “representative” among all the sets equipollent to  $A$  and that gives a precise idea of the amount of elements  $A$  has.

For finite sets, we have a standard solution to this problem: the natural numbers. It turns out that, for technical reasons, instead of



counting from “one, two, three, ...,” it’s better to start with “zero, one, two, ...” Earlier, we defined  $\underline{n}$  to be the set of natural numbers  $\{k \in \mathbb{N} \mid k < n\}$ . Thus, the cardinality of a finite set will be  $n$  if there is a bijection between  $A$  and  $\underline{n}$ . Let’s formalize all this. For each natural number  $n \in \mathbb{N}$ , we observe that  $\underline{n}$  has  $n$  elements. We use this to give a precise definition of a finite set.

---

LEMMA 13:

---

*If  $n, m$  are different natural numbers, then  $\underline{n} \not\simeq \underline{m}$ .*

*Proof.* Notice that, without loss of generality, we can assume that  $n < m$ . We can prove then, by induction on  $n$ , that for every natural number  $m$ , if  $n < m$ , there is no injective function  $f: \underline{m} \rightarrow \underline{n}$ .

For  $n = 0$ , this is trivial, because there is no function from a nonempty set to  $\underline{0} = \emptyset$ . Suppose now that for a certain  $k$ , we have that for all  $m > k$  there is no injective map from  $\underline{m} \rightarrow \underline{k}$ . Assume then that  $p > k^\frown = k + 1$ , let  $f: \underline{p} \rightarrow \underline{k^\frown}$  be an arbitrary function, and let’s see that  $f$  cannot be injective. Notice that  $\underline{k^\frown} = \{0, 1, \dots, k\}$ . Let  $S = \{x \in \underline{p} : f(x) = k\}$  and let’s distinguish three cases. Clearly, if  $S$  has two different elements, then  $f$  is not injective. If  $S = \emptyset$ , then we can define a function  $g: \underline{p} \rightarrow \underline{k}$  given by  $g(x) = f(x)$ . Notice that  $p > k^\frown > k$ , and therefore  $g$  cannot be injective by the inductive hypothesis, whence we deduce that  $f$  is not injective either. Finally, if  $S = \{x\}$  for some element  $x \in \underline{p}$ , we can first notice that since  $p > k^\frown$ , the number  $p$  must be the successor of a natural number  $m$ , and  $\underline{p} = \{0, 1, \dots, m\}$ . We can find a bijective map  $h: \underline{p} \setminus \{x\} \rightarrow \underline{m}$  and composing it with  $f$  we obtain a map  $f \circ h: \underline{m} \rightarrow \underline{k}$ . Given that  $m + 1 = p > k^\frown = k + 1$ , we also have that  $m > k$ , and by the inductive hypothesis,  $f \circ h$  cannot be injective, and therefore  $f$  cannot be injective, since  $h$  is a bijection.  $\square$

---



---

DEFINITION 34: Finite and Infinite Sets

---

*A set  $A$  is **finite** if there is a natural number  $n \in \mathbb{N}$  such that  $A \simeq \underline{n}$ . We say that  $n$  is the **cardinality** of  $A$ , and denote it by  $|A| = n$ . A set  $A$  is **infinite** if it is not finite.*

---

Notice that, by the previous lemma, the cardinality of a finite set is well defined, that is, it is a unique natural number. Indeed, if  $A \simeq \underline{n}$  and  $A \simeq \underline{m}$  for natural numbers  $n$  and  $m$ , then  $\underline{n} \simeq \underline{m}$ , and by the previous lemma,  $n = m$ .

We are very familiar with finite sets, and many properties seem very natural to us. For example, a finite set is never equipollent with any of its proper parts. That is, if  $A$  is finite set and  $B \subsetneq A$ , then  $A \not\approx B$ . This is not true for infinite sets.

One famous example was Galilei's argument that the set of all natural numbers is "just as numerous" (in his terminology, translated to English) as the set of all perfect square numbers, because we can match every natural number  $n$  with its square  $n^2$ . In our terminology, we would say that the set  $\mathbb{N}$  and the set  $S = \{x \in \mathbb{N} : x = n^2, \text{ for some } n \in \mathbb{N}\}$  are equipollent, since the function  $f: \mathbb{N} \rightarrow S$  given by  $f(n) = n^2$  is a bijection.

$$\begin{array}{ccc} \mathbb{N} & \xrightarrow{f} & S \\ 0 & \longmapsto & 0 \\ 1 & \longmapsto & 1 \\ 2 & \longmapsto & 4 \\ 3 & \longmapsto & 9 \\ \vdots & & \vdots \end{array}$$

This was very surprising to Galilei because, on the one hand,  $S \subsetneq \mathbb{N}$  suggests that  $\mathbb{N}$  has more elements than  $S$ , that is,  $\mathbb{N}$  is "more numerous" than  $S$ , but on the other, the correspondence  $f$  shows that  $\mathbb{N}$  and  $S$  are "equally numerous." Galilei interpreted this as a paradox, that is, a contradiction, and simply deduced that

[...] the attributes "equal," "greater," and "less," are not applicable to infinite, but only to finite, quantities.

We should point out that this "paradox" is an apparent contradiction with our intuitions, but not a contradiction of the theory itself. It shows a behavior of infinite sets that goes against our experience about quantities, but mathematically there is nothing wrong. At this point, we can either join Galilei and reject the notion of cardinality for infinite sets, or we could follow the definitions that we gave to their ultimate consequences and accept that infinite sets are not going to behave always like finite sets. After all, our daily experience with quantities is just about finite sets, and this behavior of infinite sets could just be a feature that seems strange because unfamiliar.

This was precisely the approach that is followed in mathematics nowadays. Then, for instance, we accept that infinite sets can be equipollent to proper subsets. Actually, we can give many examples. We already mentioned that  $\mathbb{N}$  is equipollent to the set of perfect squares. Another example easier to prove is that  $\mathbb{N}$  is equivalent to the set  $2\mathbb{N} = \{x \in \mathbb{N} : x = 2 \cdot n, \text{ for some } n \in \mathbb{N}\}$  of all even numbers. For instance, the function  $f: \mathbb{N} \rightarrow 2\mathbb{N}$  given by  $f(n) = 2 \cdot n$

is a bijection. Indeed,  $f$  is injective since  $f(m) = f(n)$  is equivalent to  $2 \cdot m = 2 \cdot n$ , which implies that  $m = n$ , by cancellativity, since  $2 \neq 0$ . On the other hand, every element  $x \in 2\mathbb{N}$  is of the form  $x = 2 \cdot n$ , that is, there is some  $n \in \mathbb{N}$  such that  $x = 2 \cdot n = f(n)$ , and therefore  $f$  is also surjective.

What about other sets of numbers? It turns out that we can prove that many different set of numbers are equipollent to  $\mathbb{N}$  (and then, equipollent to each other). It is not difficult to see that the set  $\mathbb{Z}$  of the integer numbers is equipollent to  $\mathbb{N}$ . All we need is to show a bijection. For example, we define a function  $f: \mathbb{N} \rightarrow \mathbb{Z}$  by counting the integer numbers starting with 0, then  $-1$ , then 1, then  $-2$ , then 2, and so on. That is,  $f(0) = 0$ ,  $f(1) = -1$ ,  $f(2) = 1$ ,  $f(3) = -2$ ,  $f(4) = 2$ , ... It would be more compelling if we had an explicit formula for  $f(n)$ . This is it:

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even,} \\ -\frac{n+1}{2} & \text{if } n \text{ is odd.} \end{cases}$$

It is a simple computation to show that the map  $g: \mathbb{Z} \rightarrow \mathbb{N}$  given by

$$g(z) = \begin{cases} 2 \cdot n & \text{if } n \text{ is nonnegative,} \\ -2 \cdot n - 1 & \text{if } n \text{ is negative} \end{cases}$$

is the inverse of  $f$ , what shows that  $f$  is a bijection and hence  $\mathbb{N} \simeq \mathbb{Z}$ .

This fact might not be that surprising, since we can understand that  $\mathbb{Z}$  results somehow by duplicating the number of elements of  $\mathbb{N}$  (by adding one negative version of  $n$  for every natural number  $n$ ), and we have already seen that  $2\mathbb{N}$  (which results from  $\mathbb{N}$  by deleting half it elements) is equipollent with  $\mathbb{N}$ . So, it seems that duplicating the number of elements of an infinite set is not going to change its *cardinality*. But, surely, the set of all rational numbers  $\mathbb{Q}$  should be larger, right? Well, not really, it can be shown that  $\mathbb{Q}$  is equipollent to  $\mathbb{N}$ . Indeed, we can use the positive numbers to count the positive rational numbers,  $\mathbb{Q}^+$ . Just consider the following two tables:

$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	...	1	2	4	7	...
$\frac{2}{1}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{4}$	...	3	5	8	12	...
$\frac{3}{1}$	$\frac{3}{2}$	$\frac{3}{3}$	$\frac{3}{4}$	...	6	9	13	18	...
$\frac{4}{1}$	$\frac{4}{2}$	$\frac{4}{3}$	$\frac{4}{4}$	...	10	14	19	25	...
$\frac{5}{1}$	$\frac{5}{2}$	$\frac{5}{3}$	$\frac{5}{4}$	...	15	20	26	33	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

The rule to generate the first table seems clear, we position all rational numbers on the table according to their numerator and their denominator. But, let's note that each rational number will appear several times in the first table. The second table requires

some explanation. We position each positive natural number on the table filling the diagonals  $\swarrow$ , from left to right and top to bottom-left. And we can use these two tables to establish a bijection between the positive natural numbers and the representations of the rational numbers. This is not a bijection between  $\mathbb{N}^+$  and  $\mathbb{Q}^+$ , but it can be used to define such a bijection  $h_+ : \mathbb{N}^+ \rightarrow \mathbb{Q}^+$ . Then, we can use this bijection to define a bijection  $h : \mathbb{Z} \rightarrow \mathbb{Q}$  by

$$h(z) = \begin{cases} h_+(z) & \text{if } z > 0, \\ 0 & \text{if } z = 0, \\ -h_+(-z) & \text{if } z < 0. \end{cases}$$

That is,  $\mathbb{Z} \simeq \mathbb{Q}$ , and since  $\mathbb{N} \simeq \mathbb{Z}$ , we also have that  $\mathbb{Z} \simeq \mathbb{Q}$ .

Now, if we let ourselves be guided by this examples, it could seem that all infinite sets of numbers must be equipollent, and this is far from being true! Indeed, Cantor prove that the set  $\mathbb{R}$  of all real numbers is not equipollent with  $\mathbb{N}$ !

---

**THEOREM 6:** (Cantor)

---

*There is no surjective function from  $\mathbb{N}$  to  $\mathbb{R}$ .*

*Proof.* Suppose that  $f : \mathbb{N} \rightarrow \mathbb{R}$  is a function and let's prove that there is a number  $x \in \mathbb{R}$  that is not in the image of  $\mathbb{N}$ , that is,  $x \neq f(n)$  for every  $n \in \mathbb{N}$ . We will use the decimal expression of the real numbers.

$$\begin{aligned} f(0) &= a_0.a_{00}a_{01}a_{02}a_{03}a_{04}\dots \\ f(1) &= a_1.a_{10}a_{11}a_{12}a_{13}a_{14}\dots \\ f(2) &= a_2.a_{20}a_{21}a_{22}a_{23}a_{24}\dots \\ f(3) &= a_3.a_{30}a_{31}a_{32}a_{33}a_{34}\dots \\ &\vdots \end{aligned}$$

Let's construct the number  $x = 0.x_0x_1x_2x_3\dots$ , where the  $n$ th decimal digit of the decimal expression of  $x$  is

$$x_n = \begin{cases} 0 & \text{if } a_{nn} \neq 0, \\ 1 & \text{if } a_{nn} = 0. \end{cases}$$

Therefore,  $x$  cannot be in the list. Notice that if  $x = f(n)$ , then the  $n$ th decimal digit of  $x$  is  $x_n \neq a_{nn}$ , which is the decimal digit of  $f(n)$ , which is a contradiction. Therefore, there is no  $n \in \mathbb{N}$  such that  $f(n) = x$ .

Therefore, it doesn't matter which function  $f : \mathbb{N} \rightarrow \mathbb{R}$  we have, it cannot be surjective.  $\square$

---

Cantor's Theorem shows that, despite  $\mathbb{N}$  and  $\mathbb{R}$  being both infinite, they are not equipollent. Actually, since  $\mathbb{R}$  contains  $\mathbb{N}$ , we can say that the cardinality of  $\mathbb{R}$  is actually larger than the cardinality of  $\mathbb{N}$ . That is, there are infinities that are larger than other infinities. The cardinality of  $\mathbb{N}$  is denoted by  $\aleph_0$  ( $\aleph$  is the first letter of the Hebrew alphabet), while the cardinality of  $\mathbb{R}$  is denoted by  $\mathfrak{c}$ , since the set of real numbers is often called the **continuum**.

It is a very natural question then whether there are other infinities even larger than that of the continuum. And the answer is astonishing: there are an unlimited amount of infinities, each larger than the previous one! To prove this fact, we will use the notion of subset. Recall that a set  $B$  is a *subset* of a given set  $A$  if every element of  $B$  is an element of  $A$ , and we denote this fact by  $B \subseteq A$ . One of the principles of Set Theory is that we can collect all the subsets of any particular given set and form a new set with them.

---

**PRINCIPLE 5:** Existence of the Powersets

---

*For every set  $A$ , there is a set denoted  $\mathcal{P}(A)$  whose elements are the subsets of  $A$ , that is, for every set  $Z$ ,*

$$Z \in \mathcal{P}(A) \iff Z \subseteq A.$$


---

---

**EXAMPLE 31:**

---

1. The empty set  $\emptyset$  only has one subset, which is itself. Therefore,  $\mathcal{P}(\emptyset) = \{\emptyset\}$ .
2. The set  $\{1\}$  containing just one element, the number 1, has two subsets, namely  $\emptyset$  and  $\{1\}$ . Therefore,  $\mathcal{P}(\{1\}) = \{\emptyset, \{1\}\}$ .
3. The set  $A = \{1, 2\}$  has four subsets:  $\emptyset, \{1\}, \{2\}, \{1, 2\} = A$ . Therefore, its powerset is  $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, A\}$ .
4. The set  $A = \{1, 2, 3\}$  has eight subsets. Its powerset is

$$\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, A\}.$$


---

As we can see in the previous examples, the powerset of a set always has more elements than  $A$ . Actually, just with these few examples we could postulate that if a finite set  $A$  has cardinality  $n$ , then its powerset has cardinality  $2^n$ . This phenomenon continues happening with infinite sets too.

---

**THEOREM 7: (Cantor)**


---

*Given any set  $A$ , we always have that there is no surjection  $A \rightarrow \mathcal{P}(A)$ . In particular,  $A \not\cong \mathcal{P}(A)$ .*

*Proof.* Suppose that  $f: A \rightarrow \mathcal{P}(A)$  is an arbitrary function. We will show that there is a set  $K \in \mathcal{P}(A)$  that is not the image of any element of  $A$  by  $f$ , and therefore  $f$  is not a surjection. Notice that, given any element  $a \in A$ , we have that  $f(a) \in \mathcal{P}(A)$ , that is,  $f(a) \subseteq A$ . Hence,  $f(a)$ , as a subset of  $A$ , will only contain elements of  $A$  but maybe not all of them. In particular,  $f(a)$  might contain or not the element  $a \in A$ . Consider the set

$$K = \{a \in A : a \notin f(a)\}.$$

Obviously,  $K \subseteq A$  by construction, that is,  $K \in \mathcal{P}(A)$ , and moreover, for every element  $a \in A$ , we have that

$$a \in K \iff a \notin f(a). \quad (9.1)$$

Suppose that  $k \in A$  is an element such that  $f(k) = K$ . Applying (9.1), we obtain that

$$k \in K \iff k \notin f(k) = K,$$

which is a contradiction. Therefore, there is no element  $k \in A$  such that  $f(k) = K$ , and hence,  $f$  is not a surjection.  $\square$

---

In this way, the sets  $\mathbb{N}$ ,  $\mathcal{P}(\mathbb{N})$ ,  $\mathcal{P}(\mathcal{P}(\mathbb{N}))$ ,  $\mathcal{P}(\mathcal{P}(\mathcal{P}(\mathbb{N})))$ ,  $\dots$  will all be infinite, and each one of them with a larger cardinality than the previous one.

Where does the sets  $\mathbb{R}$  and  $\mathbb{C}$  of real and complex numbers fit in this chain of infinite sets? Well, it can be proven that  $\mathbb{R} \simeq \mathcal{P}(\mathbb{N}) \simeq \mathbb{C}$ .

Cantor investigated the cardinalities of the subsets of  $\mathbb{R}$  and showed many interesting results. For example, a set of numbers that is of crucial interest in Algebra is the set of all the so-called **algebraic numbers**. These are the real numbers that are the roots of polynomials with integer coefficients. Thus, for example, every rational number  $\frac{p}{q}$  is an algebraic number, since  $\frac{p}{q}$  is a solution of the equation  $qx - p = 0$  and  $qx - p$  is a polynomial with integer coefficients. But also many irrational numbers, like  $\sqrt{2}$ , are algebraic. In particular,  $\sqrt{2}$  is a solution of the equation  $x^2 - 2 = 0$ , which has integer coefficients. The real numbers that are not algebraic are called **transcendental**. Liouville proved the existence of transcendental numbers in 1844. After him, other mathematicians proved the existence of other transcendental numbers. Of particular importance in

Mathematics are  $\pi$  and Euler's number  $e$ , which were proved to be transcendental too. Cantor proved that the cardinality of the set of algebraic numbers is  $\aleph_0$ . And since the cardinality of the continuum is much larger than  $\aleph_0$ , we deduce that, not only transcendental numbers exist, but actually *most of real numbers are transcendental!*

To end this discussion, let's mention that in his investigations, Cantor only found sets of numbers that have cardinality  $\aleph_0$  or cardinality  $c$ , so he made one of the most famous conjectures in Mathematics.

---

#### CONJECTURE 1: Cantor's Continuum Hypothesis

---

*If  $A$  is an infinite set of real numbers, then  $|A| = \aleph_0$  or  $|A| = c$ .*

---

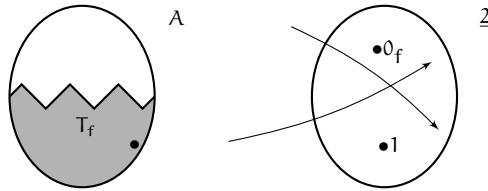
The answer to whether this conjecture is true or not is one of the most fascinating in Mathematics: Whatever you want! Indeed, it turns out that neither Cantor's Hypothesis nor its negation can be proved in the standard Set Theory, and therefore there is no problem in assuming that it is true or it is false. Of course, we cannot assume both at the same time. If we assume one or the other, we will obtain *different mathematics*, as both Cantor's hypothesis and its negation have their corresponding ramifications. But both theories will be equally valid, from the formal point of view.

#### *Characteristic Functions and the Size of the Powersets*

Recall that  $\underline{2} = \{0, 1\}$  and let's denote by  $\underline{2}^A$  the set of all the functions  $A \rightarrow \underline{2}$ . Thus, if  $f \in \underline{2}^A$ , we have that  $f: A \rightarrow \{0, 1\}$  and we can define the subset

$$T_f = \{a \in A : f(a) = 1\}.$$

That is, we collect in  $T_f$  all the elements of  $A$  with image 1.



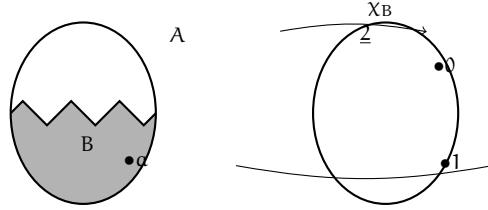
Obviously,  $T_f \subset A$ , that is,  $T_f \in \mathcal{P}(A)$ . Therefore, we can define a function  $T: \underline{2}^A \rightarrow \mathcal{P}(A)$  by  $T(f) = T_f = \{a \in A : f(a) = 1\}$ .

We will see that this function  $T$  is actually a bijection, and in order to do that, we will construct the inverse of  $T$ .

Given a subset  $B \subseteq A$ , we can define the function  $\chi_B: A \rightarrow \{0, 1\}$  given by the following rule:

$$\chi_B(a) = \begin{cases} 1 & \text{if } a \in B, \\ 0 & \text{if } a \notin B. \end{cases}$$

Obviously, this is well defined, because for ever element  $a \in A$ , either  $a \in B$  or  $a \notin B$ . This function  $\chi_B: A \rightarrow \underline{2}$  is called the **characteristic function** of  $B$  with respect to  $A$ .



We can use that to define a function  $\chi: \mathcal{P}(A) \rightarrow \underline{2}^A$  given by  $\chi(B) = \chi_B$ , that is,  $\chi$  assigns to every subset  $B$  of  $A$  its characteristic function. Now, it is very easy to see that  $\chi$  is the inverse of  $T$ .

---

**PROPOSITION 22:**

---

*For every given set  $A$ , and with the notation of the previous paragraphs, the pair  $(T, \chi)$  is a matching between  $\underline{2}^A$  and  $\mathcal{P}(A)$ , that is, the map  $T: \underline{2}^A \rightarrow \mathcal{P}(A)$  is bijective and  $\chi: \mathcal{P}(A) \rightarrow \underline{2}^A$  is its inverse.*

*Proof.* Let's first prove that  $\chi \circ T = \text{id}_{\underline{2}^A}$ .

$$\begin{array}{ccccc} \underline{2}^A & \xrightarrow{T} & \mathcal{P}(A) & \xrightarrow{\chi} & \underline{2}^A \\ & & \searrow & \nearrow & \\ & & \chi \circ T = \text{id}_{\underline{2}^A} & & \end{array}$$

Consider  $f \in \underline{2}^A$  an arbitrary element, that is, a function  $f: A \rightarrow \{0, 1\}$ . We have to check that  $(\chi \circ T)(f) = f$ . First, notice that

$$(\chi \circ T)(f) = \chi(T(f)) = \chi(T_f) = \chi_{T_f}: A \rightarrow \{0, 1\}.$$

Therefore, all we have to prove is that both  $\chi_{T_f}$  and  $f$  act in the same way on the elements of  $A$ . Let  $a \in A$  be an arbitrary element. By using the corresponding definitions, we have

$$\chi_{T_f}(a) = 1 \iff a \in T_f \iff f(a) = 1,$$

as we wanted to show.



For the other direction, we need to show that  $T \circ \chi = \text{id}_{\mathcal{P}(A)}$ .

$$\begin{array}{ccccc} \mathcal{P}(A) & \xrightarrow{\chi} & \underline{2}^A & \xrightarrow{T} & \mathcal{P}(A) \\ & \searrow & \text{---} & \nearrow & \\ & & T \circ \chi = \text{id}_{\mathcal{P}(A)} & & \end{array}$$

Let  $B \in \mathcal{P}(A)$  be an arbitrary subset of  $A$  and let's prove that  $(T \circ \chi)(B) = B$ . Notice that

$$(T \circ \chi)(B) = T(\chi(B)) = T(\chi_B) = T_{\chi_B} \in \mathcal{P}(A).$$

Thus, we have to show that both  $T_{\chi_B}$  and  $B$  contain exactly the same elements of  $A$ . Let  $a \in A$  be an arbitrary element. We have that

$$a \in T_{\chi_B} \iff \chi_B(a) = 1 \iff a \in B,$$

which is precisely what we wanted to prove.  $\square$

We can consider then that the characteristic function  $\chi_B$  of a subset  $B$  of a set  $A$  is a way of encoding the subset  $B$  as a function, assigning 1 to the elements that belong to  $B$  and 0 to the elements that are in  $A \setminus B$ , that is, in  $A$  but not in  $B$ . This can be used to count the elements of  $\mathcal{P}(A)$ , since it will have the same number of elements as  $\underline{2}^A$ .

#### COROLLARY 1:

*Given any set  $A$ , we have that  $\mathcal{P}(A) \simeq \underline{2}^A$ .*

In the case that  $A$  is finite, we can explicitly calculate the cardinality of  $\underline{2}^A$ . Actually, given two finite sets  $X$  and  $Y$ , we can easily calculate the cardinality of  $Y^X = \{f : f : X \rightarrow Y\}$ , the set of all functions from  $X$  to  $Y$ .

#### PROPOSITION 23:

*If  $X$  and  $Y$  are finite sets of cardinalities  $|X| = n$  and  $|Y| = m$ , then the set  $Y^X$  of all functions from  $X \rightarrow Y$  has cardinality  $|Y^X| = m^n$ .*

*Proof.* Suppose that  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_m\}$ . Each function  $f : X \rightarrow Y$  is completely determined if we know the images  $f(x_i)$ , for all  $1 \leq i \leq n$ . Notice that for  $x_1 \in X$ , we have  $m$  different possible ways to define  $f(x_1)$ , namely,  $f(x_1) = y_1, f(x_1) = y_2, \dots, f(x_1) = y_m$ . The same is true for every other  $x_i \in X$ . Notice also that all these choices are independent of each other. Thus, there are  $\underbrace{m \cdot \dots \cdot m}_n = m^n$  different ways of defining a function  $f : X \rightarrow Y$ .  $\square$

---

EXAMPLE 32:

Consider the sets  $X = \{\diamond, \heartsuit, \spadesuit\}$  and  $Y = \{\$, \text{X}\}$ . Clearly,  $|X| = 3$  and  $|Y| = 2$ , and by the previous proposition,  $|X^Y| = |X|^{|Y|} = 3^2 = 9$ . That is, there are 9 different functions  $Y \rightarrow X$ . Here there is a table collecting all of them.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$
$\$$	$\diamond$	$\diamond$	$\diamond$	$\heartsuit$	$\heartsuit$	$\heartsuit$	$\spadesuit$	$\spadesuit$	$\spadesuit$
$\text{X}$	$\diamond$	$\heartsuit$	$\spadesuit$	$\diamond$	$\heartsuit$	$\spadesuit$	$\diamond$	$\heartsuit$	$\spadesuit$

Thus, for instance, according to this table,  $f_4: Y \rightarrow X$  is the function determined by  $f_4(\$) = \heartsuit$  and  $f_4(\text{X}) = \diamond$ .

---

This last proposition justifies the notation  $X^Y$  for the set of all functions from  $Y$  to  $X$ , and also, the fact that  $\mathcal{P}(A) \simeq \underline{2}^A$  justifies the fact that  $\mathcal{P}(A)$  is called the *powerset* of  $A$ . Taking together Corollary 1 and Proposition 23, we deduce the cardinality of the powersets.

---

COROLLARY 2:

Given a finite set  $A$  of cardinality  $|A| = n$ , its powerset has cardinality

$$|\mathcal{P}(A)| = 2^n.$$


---

### *Embeddings, size comparison, and the Cantor-Bernstein Theorem*

The previous paragraphs concentrate on *equipollency* — when are two sets the same size. But clearly, we can also compare sets. Evidently, if  $A \subseteq B$ , then  $A$  is no bigger than  $B$ , though it might be smaller. More generally, if there is an injective function from  $A$  to  $B$ , then  $A$  is no bigger. We can make this a definition

---

DEFINITION 35:

*Embedding of sets* For sets  $A$  and  $B$ , we sometimes refer to an injective function  $f: A \rightarrow B$  as an **embedding** of  $A$  in  $A$ . Also,  $A$  **embeds in**  $B$  means there is some embedding of  $A$  in  $B$ .

Following the notation for equipollency, we write  $A \lesssim B$  when  $A$  embeds in  $B$ .

---

---

EXAMPLE 33:

---

The set  $\mathbb{N}$  embeds in the set  $\mathbb{N} \times \mathbb{N}$  via the so-called diagonal function  $\delta: \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$  given by  $\delta(n) = (n, n)$ . Evidently, this same idea works for any set  $A$ . So  $\mathbb{N} \lesssim \mathbb{N} \times \mathbb{N}$ . The function  $(m, n) \mapsto 2^m \cdot 3^n$  shows that there is an embedding from  $\mathbb{N} \times \mathbb{N}$  to  $\mathbb{N}$ , as well. Hence  $\mathbb{N} \times \mathbb{N} \lesssim \mathbb{N}$ .

The set  $\mathbb{N}$  embeds in the set  $\mathbb{Q}^+$  of non-negative rational numbers by  $f: \mathbb{N} \rightarrow \mathbb{Q}$  given by  $f(n) = \frac{n}{1}$ . Also,  $\mathbb{Q}^+$  embeds in  $\mathbb{N} \times \mathbb{N}$ , for any rational number can be written uniquely as a simplified fraction  $\frac{n}{d}$  where  $n$  and  $d$  have no factors in common.

For any set  $A$ ,  $A \lesssim \mathcal{P}(\mathbb{N})$  by the **singleton function**  $x \mapsto \{x\}$ .

---



---

PROPOSITION 24: Embedding is a preorder relation

---

For any sets  $A$ ,  $B$  and  $C$ ,

- if  $A \lesssim B$  and  $B \lesssim C$ , then  $A \lesssim C$ ; and
- $A \lesssim A$ .

*Proof.* A composition of two injective functions is injective. And the identity on  $A$  is an injective function. □

---

Clearly,  $\lesssim$  is not anti-symmetric. For example,  $\mathbb{N}$  and  $\mathbb{N}^+$  are not equal, but they embed in each other. In fact,  $\mathbb{N} \simeq \mathbb{N}^+$  because there is a suitable bijection. This raised a question for Cantor. Suppose  $f: A \rightarrow B$  and  $g: B \rightarrow A$  are both injections, but we do not know anything about how they are related. Cantor concluded that it must be the case that  $A$  and  $B$  are actually equipollent, and stated this without a proof.

Shortly after that, Richard Dedekind proved it as a consequence of a stronger principle of set theory.

Sometime later, Bernstein (at the age of 19) independently proved it and included the proof in his doctoral thesis. The result is also often attributed to Schröder, who first published an incorrect proof, and shortly later found a correction.

In principle, the theorem ought to be associated with Dedekind, but traditionally, mathematicians tend to name it for Cantor and Bernstein (and sometimes Schröder).

---

**THEOREM 8:** The Cantor-Bernstein (and really, -Dedekind Theorem)
 

---

For sets  $A$  and  $B$ , if  $A \preceq B$  and  $B \preceq A$ , then  $A \simeq B$ .

*Proof.* Suppose  $f: A \rightarrow B$  and  $g: B \rightarrow A$  are both injective functions. The goal is to prove that  $A \simeq B$ .

The proof is simplified by first proving a general lemma about bijections.

**Claim 1:** Suppose

- (i)  $A = \bigcup_{i \in I} A_i$  and  $B = \bigcup_{i \in I} B_i$ ,
- (ii) for each  $i, j \in I$ , if  $i \neq j$ , then  $A_i \cap A_j = \emptyset$  and  $B_i \cap B_j = \emptyset$ , and
- (iii) for each  $i \in I$ ,  $A_i \simeq B_i$ .

Then  $A \simeq B$ .

*Proof of claim 1:* Suppose  $A$  and  $B$  satisfy the hypothesis, and for each  $i \in I$ ,  $h_i: A_i \rightarrow B_i$  is a bijection witnessing that  $A_i \simeq B_i$ . Because of (ii) and (iii), each  $x \in A$  belongs to exactly one  $A_i$ . So, we can define a function  $k: A \rightarrow I$  that so that  $x \in A_{k(x)}$  holds for every  $x \in A$ . In plain English,  $k$  assigns to each  $x \in A$ , the subset  $A_i$  to which  $x$  belongs. Now define  $h: A \rightarrow B$  by

$$h(x) = h_{k(x)}(x).$$

OK. That is not exactly *plain* English, but it is precise.

The function  $h$  is well-defined, because  $x \in A_{k(x)}$ , so  $x$  is in the domain of  $h_{k(x)}$ . For any  $y \in B$ ,  $y \in B_i$  for some  $i$  that is uniquely determined by  $y$ . Since  $h_i$  is a surjection to  $B_i$ , there is some  $x \in A_i$  for which  $h_i(x) = y$ . Hence  $h(x) = y$ . This proves that  $h$  is a surjection. To see that  $h$  is also an injection, consider elements  $x_0, x_1 \in A$  satisfying  $h(x_0) = h(x_1)$ . Let  $i = k(x_0)$ , and  $j = k(x_1)$ . Then  $h_i(x_0) = h_j(x_1)$ . But if  $i \neq j$ , then  $B_i$  is disjoint from  $B_j$ . So  $h_i(x_0) = h_j(x_1)$  is impossible. So  $i = j$ , and since  $h_i$  is an injection,  $x_0 = x_1$ . This proves that  $h$  is also an injection., completing the proof of claim 1.

The plan is to define sets  $A_i \subseteq A$  and  $B_i \subseteq B$ , specifically for  $I = \mathbb{N} \cup \{*\}$ . That is, we split  $A$  and  $B$  into subsets  $A_n$  for each natural number  $n$  plus “extras”  $A_*$  and  $B_*$ , so that (spelling out the conditions needed to use the claim in this case)

- (i)  $A = A_* \cup \bigcup_{n \in \mathbb{N}} A_n$  and  $B = B_* \cup \bigcup_{n \in \mathbb{N}} B_n$ ,
- (ii) for each  $m \in \mathbb{N}$ ;
  - (a) for each  $n \in \mathbb{N}$ , if  $m \neq n$ , then  $A_m \cap A_n = \emptyset$  and  $B_m \cap B_n = \emptyset$ , and

(b)  $A_m \cap A_* = \emptyset$  and  $B_m \cap B_* = \emptyset$ ;

(iii) and

(a) for each  $m \in \mathbb{N}$ ,  $A_m \simeq B_m$ , and

(b)  $A_* \simeq B_*$ .

Recall that any function  $h: X \rightarrow Y$  determines a function  $h^+: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$  define by  $h^+(X_0) = \{y \in Y \mid y = h(x) \text{ for some } x \in X_0\}$ . In general,  $h^+$  does not necessarily preserve intersections, but if  $h$  is an injection, it does. That is,  $h^+(X_0 \cap X_1) = h^+(X_0) \cap h^+(X_1)$ . Also,  $h^+$  preserves unions, in particular,  $\emptyset: h^+(\emptyset) = \emptyset$ .

To split  $A$  and  $B$ , we first define subsets  $C_n \subseteq A$  and  $D_n \subseteq B$  as illustrated in Figure 9.1. By recursion, define

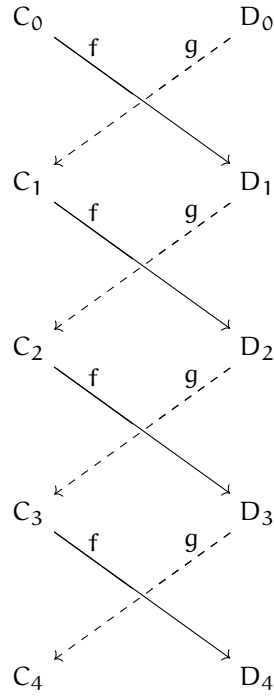


Figure 9.1: Splitting  $A$  and  $B$  into disjoint sets. Solid arrows indicate restriction of  $f$  to bijections between corresponding subsets. Dashed arrows indicate restriction of  $g$  to a bijection between corresponding sets.

$$C_0 = A \setminus g^+(B)$$

$$D_0 = B \setminus f^+(A)$$

$$C_{k+1} = g^+(B_k)$$

$$D_{k+1} = f^+(A_k)$$

This ensures that for all  $n$ ,  $C_n \simeq D_{n+1}$  and  $C_{n+1} \simeq D_n$ . Now for each natural number  $n$ , define  $A_n$  and  $B_n$ , and define  $A_*$  and  $B_*$  by

$$A_n = C_{2n} \cup C_{2n+1}$$

$$\begin{aligned}
B_n &= D_{2n+1} \cup D_{2n} \\
A_* &= A \setminus \bigcup_{n \in \mathbb{N}} A_n \\
B_* &= B \setminus \bigcup_{n \in \mathbb{N}} B_n
\end{aligned}$$

It remains to check that these data satisfy the the conditions (i), (ii), and (iii) from above.

Evidently, (i) and (ii)(b) hold by construction of  $A_*$  and  $B_*$ . To prove (ii)(a) and (iii)(a), we prove another claim.

**Claim 2:** For every  $p, q \in \mathbb{N}$ , if  $p < q$ , then  $C_p \cap C_q = \emptyset$  and  $D_p \cap D_q = \emptyset$ .

*Proof of Claim 2:* By induction on  $p$ , we prove thst for every  $q \in \mathbb{N}$ , if  $p < q$ , then  $C_p \cap C_q = \emptyset$  amd  $D_p \cap D_q = \emptyset$ .

The basis is simple, since  $C_0$  is by definition disjoint from  $g^+(B)$ , which contains  $C_q$  for any positive  $q$ . Likewise for  $D_0$  and  $D_q$ . For the inductive hypothesis, suppose that for some  $k$ ,  $C_k \cap C_j = \emptyset$  and  $D_k \cap D_j = \emptyset$  for every  $j > k$ . For the inductive step, suppose  $k^\wedge < q$ . Then  $k < q \dot{-} 1$ . So, by the inductive hypothesis,  $C_k \cap C_{q \dot{-} 1} = \emptyset$  and  $D_k \cap D_{q \dot{-} 1} = \emptyset$ . But

$$\begin{aligned}
C_{k^\wedge} \cap C_q &= g^+(D_k) \cap g^+(D_{q \dot{-} 1}) \\
&= g^+(D_k \cap D_{q \dot{-} 1}) \\
&= g^+(\emptyset) \\
&= \emptyset.
\end{aligned}$$

This completes the proof of claim 2.

By claim 2,  $C_{2n} \cap C_{2n+1} = \emptyset$ , and  $D_{2n+1} \cap D_{2n} = \emptyset$ . That is,  $A_n = C_{2n} \cup C_{2n+1}$  and  $B_n = D_{2n+1} \cup D_{2n}$  satisfy the conditions of claim 1, so for each  $n \in \mathbb{N}$ ,  $A_n \simeq B_n$ . That is, (iii)(a) holds.

For  $m \neq n$ ,

$$\begin{aligned}
A_m \cap A_n &= (C_{2m} \cup C_{2m+1}) \cap (C_{2n} \cup C_{2n+1}) \\
&= (C_{2m} \cap C_{2n}) \cup (C_{2m} \cap C_{2n+1}) \\
&\quad \cup (C_{2m+1} \cap C_{2n}) \cup (C_{2m+1} \cap C_{2n+1}).
\end{aligned}$$

Each of the intersections in this is empty by claim 2. So (ii)(a) holds.

It remains to confirm (iii)(b), namely  $A_* \simeq B_*$ .

In particular, we claim that  $f \upharpoonright A_*$  is a bijection between  $A_*$  and  $B_*$ . If  $x \in A_*$ , then  $f(x)$  cannot belong to any  $B_n$ , for  $B_0$  is disjoint from the image of  $f$  by definition, and  $f(x) \in B_{k^\wedge}$  implies  $x \in A_k$  by definition. So,  $x \in A_*$  implies  $f(x) \in B_*$ . Conversely, if  $x \in A_k$ , then  $f(x) \in B_{k^\wedge}$ . So,  $f \upharpoonright A_*$  is a surjection to  $B_*$ . Since  $f$  is injective, so i,  $f \upharpoonright A_*$  is. This proves (iii)(b).

Putting these facts together, the sets  $A_n$ ,  $A_*$ ,  $B_n$ , and  $B_*$  satisfy the conditions of claim 1, so  $A \simeq B$ .  $\square$

---



---

EXAMPLE 34:

---

The open interval  $(0, 1) = \{x \in \mathbb{R} \mid 0 < x < 1\}$  obviously embeds in  $[0, 1] = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$  as a subset. But also, the function  $g: [0, 1] \rightarrow (0, 1)$  defined by  $f(x) = \frac{1+x}{3}$  embeds  $[0, 1]$  in  $(0, 1)$ . Hence the two sets are equipollent in spite of the endpoints 0 and 1 seeming to be “extra” elements of  $[0, 1]$ .

Notice that since we are only concerned with equipollency, the specific bijection from  $(0, 1)$  and  $[0, 1]$  is not important. In the example, the bijection  $h: (0, 1) \rightarrow [0, 1]$  turns out (you can prove this for yourself) that

$$\begin{aligned} h\left(\frac{3^{n+1}-1}{2 \cdot 3^{n+1}}\right) &= \frac{3^n-1}{2 \cdot 3^n} \\ h\left(\frac{3^{n+1}+1}{2 \cdot 3^{n+1}}\right) &= \frac{3^n+1}{2 \cdot 3^n} \\ h(x) &= x \quad \text{all other } x \end{aligned}$$

This function is quite useless for any other mathematical purpose. Nevertheless, it does prove that the two sets  $(0, 1)$  and  $[0, 1]$  are equipollent.

---

The point of this example is partly that we may not really need (or even want) to know details of a particular bijection. The Cantor-Bernstein Theorem provides a condition for existence. That is enough to prove two sets are equipollent.



EXERCISES:

---

Let  $\text{Seq}[\mathbb{N}]$  denote the set of all sequences of natural numbers.

100. Prove that  $\text{Seq}[\mathbb{N}] \simeq (\text{Seq}[\mathbb{N}])^2$  by defining an injection from  $\text{Seq}[\mathbb{N}]$  to  $(\text{Seq}[\mathbb{N}])^2$ , and an injection from  $(\text{Seq}[\mathbb{N}])^2$  to  $\text{Seq}[\mathbb{N}]$ .
  101. Prove that  $\mathbb{R} \simeq \text{Seq}[\mathbb{N}]$ .
  102. Prove that  $\mathbb{R} \simeq \mathbb{R}^2$ .
- 

A sequence of As is a function  $\mathbb{N} \rightarrow A$ . So, alternatively,  $\text{Seq}[A]$  is defined by  $A^{\mathbb{N}}$  – the set of all functions from  $\mathbb{N}$  to  $A$ .

## Operations on Sets

In this chapter, we will assume that we have a set  $U$  (as large as we need it to be) and  $A, B, C$ , etc. will denote subsets of  $U$ . We call  $U$  the **universe of discourse** or just the **universe**, for short. We analyze in this chapter the ways  $A$  and  $B$  can be combined to form new subsets of  $U$ . The result of this analysis will show that  $\mathcal{P}(U)$  is what is known as a **Boolean algebra**. This is intimately connected to classical logic.

### *The Finitary Structure of Powersets*

The **intersection** of two sets  $A$  and  $B$ , denoted by  $A \cap B$ , is the subset consisting of those elements that  $A$  and  $B$  have in common. Formally, we would write it as follows:

$$A \cap B = \{x \in U : x \in A \text{ and } x \in B\}.$$

Likewise, we might consider merging  $A$  and  $B$  into a single set. This is called the **union** and is denoted by  $A \cup B$ .

$$A \cup B = \{x \in U : x \in A \text{ or } x \in B\}.$$

We can characterize  $\cap$  and  $\cup$  in an instructive way to highlight their similarity. It is easy to check that  $C \subseteq A \cap B$  holds if and only if both  $C \subseteq A$  and  $C \subseteq B$ . Moreover,  $A \cap B$  is the only subset of  $U$  with this property. Reversing the inclusions,  $A \cup B \subseteq C$  if and only if  $A \subseteq C$  and  $B \subseteq C$ .

We already can see the connection between these two operations of intersection and union with logic. In particular, the intersection is defined in terms of the connective ‘and’, while the union is defined in terms of the connective ‘or.’ Therefore, intersections and unions will behave like conjunctions and disjunctions, respectively.

There are other logic connectives that are very important: ‘implies’ and ‘not’. Let’s study first implication of two sets  $A$  and  $B$ , which is defined as follows:

$$A \Rightarrow B = \{x \in U : \text{if } x \in A \text{ then } x \in B\}.$$



This operation is related to intersection and can be characterized in the following way:  $A \Rightarrow B$  is a largest set  $C$  so that  $A \cap C \subseteq B$ . For this reason  $A \Rightarrow B$  is also called the *residual* of the intersection. Let's prove this assertion:

First, we need to show that the intersection of  $A$  and  $A \Rightarrow B$  is included in  $B$ , that is,  $A \cap (A \Rightarrow B) \subseteq B$ . Suppose then that  $x \in A \cap (A \Rightarrow B)$  and let's show that  $x \in B$ . Since  $x$  belongs to the intersection of  $A$  and  $A \Rightarrow B$ , then  $x \in A$  and  $x \in A \Rightarrow B$ . But,  $x \in A \Rightarrow B$  means that if  $x \in A$  then  $x \in B$ . But we also know that  $x \in A$ . Therefore, we deduce that  $x \in B$ , as we wanted.

For the other direction, suppose that  $C$  is a set such that  $A \cap C \subseteq B$ . We need to show that  $C \subseteq A \Rightarrow B$ . Suppose then that  $x \in C$  and prove that  $x \in A \Rightarrow B$ . In order to do that, suppose that  $x \in A$ . Then, we have that  $x \in A$  and  $x \in C$ , whence we obtain that  $x \in A \cap C$ . But, since we assumed that  $A \cap C \subseteq B$ , we also have that  $x \in B$ . That is, we just showed that if  $x \in A$  then  $x \in B$ , what by definition means that  $x \in A \Rightarrow B$ .

Flipping things the other way, there is also a smallest set  $C$  so that  $A \subseteq B \cup C$ . This is called the **set difference** and is denoted by  $A \setminus B$ . This is characterized by "but not". So  $x \in A \setminus B$  if and only if  $x \in A$  but not  $x \in B$ . That is,

$$A \setminus B = \{x \in U : x \in A \text{ but } x \notin B\}.$$

These operations on subsets of  $U$  are closely related to the logic of propositions. Imagine that  $U$  consists of a "universe of possible worlds." Then subsets of  $U$  are collections of worlds where certain things are true. For example, perhaps  $P \subseteq U$  is the set of worlds in which pigs fly;  $K \subseteq U$  is the set of worlds in which kittens smoke cigars. So  $P \cap K$  is the set of worlds in which pigs fly *and* kittens smoke cigars. Likewise,  $P \cup K$  is the set of worlds in which *either* pigs fly *or* kittens smoke cigars or both. And  $P \Rightarrow K$  is the set of worlds in which if pigs fly, *then* kittens smoke cigars. Finally,  $P \setminus K$  is the set of worlds in which pigs fly, but kittens don't smoke cigars.

Understanding the interaction of  $\cap$ ,  $\cup$ , and  $\Rightarrow$  is closely related to the logic of "and", "or," and "implies." If we add a sentence "False" that is never true and another one "True" that is always true, then the logic of "and", "or," and "implies" form what is called a *Heyting algebra*. The other operation  $\setminus$  is less commonly singled out in logic, but is the dual of  $\Rightarrow$ . It turns out, however, that  $\setminus$  is used very commonly when dealing with sets *per se*.

In fact, "implies" interacts with "False" in a useful way. It turns out that "P implies False" is essentially the same as saying "P is not true". And if "P is not true" is not true, then "P" must be true. This observation indicates that the Heyting algebra of subsets is actually

what is called a **Boolean algebra**.

The operations of intersection, union, residual and difference, as well as the specific subsets  $\emptyset$  and  $U$ , all can be characterized in terms of how they relate to other subsets.

---

CHARACTERIZING FINITARY OPERATIONS ON SUBSETS.

---

In the following, let  $U$  be a set, and  $A, B, C \subseteq U$  be subsets.

- $C \subseteq A \cap B$  if and only if  $C \subseteq A$  and  $C \subseteq B$ .
- $A \cup B \subseteq C$  if and only if  $A \subseteq C$  and  $B \subseteq C$ .
- $C \subseteq A \Rightarrow B$  if and only if  $C \cap A \subseteq B$ .
- $A \setminus B \subseteq C$  if and only if  $A \subseteq B \cup C$ .
- $C \subseteq U$
- $\emptyset \subseteq C$

These properties describe nearly all the properties of these operations we need.

---

Additionally, “implies” and “false” interact in a stronger way, as do “but not” and “true”. In particular,  $\mathcal{P}(U)$  is a *Boolean algebra*. Let us abbreviate  $A \Rightarrow \emptyset$  by writing  $A^*$  (read this informally as “not  $A$ ”). In terms of its elements,  $A^*$  turns out to be

$$A^* = U \setminus A = \{x \in U : x \notin A\},$$

which we leave as an exercise. Then the Law of Double Negation which characterizes Boolean algebras, asserts that double negations do not change anything:  $A^{**} = A$ .

---

LEMMA 14: The Law of Double Negation

---

For any given set  $A \subseteq U$ ,  $A^{**} = A$ .

*Proof.* Calculating the members of  $A^*$ , it is easy to check that for every element  $x \in X$ , either  $x \in A$  or  $x \in A^*$ , but not both. So  $x \in A^{**}$  if and only if  $x \notin A^*$  if and only if  $x \in A$ . □

---

One can define the term “Boolean algebra” to mean “Heyting algebra that satisfies the Law of Double Negation.” So  $\mathcal{P}(X)$  is indeed a Boolean algebra. This means that  $\cap, \cup, \Rightarrow, \setminus, *, \emptyset$  and  $U$  satisfy various laws. We spell the most important laws out here.

---

LAWS OF FINITARY SET OPERATIONS

---

For any set  $U$  and any  $A, B, C \in \mathcal{P}(U)$ :

**Semilattice Laws**

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

**Commutativity**       $A \cap B = B \cap A$   
 $A \cup B = B \cup A$

**Idempotency**       $A \cap A = A$   
 $A \cup A = A$

**Lattice Laws**

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

**Ordering**       $A = B \cap A$  if and only if  $A \cup B = B$  if and only if  $A \subseteq B$

**Bounded Lattice Laws**

**Identity**       $A \cap U = A$   
 $A \cup \emptyset = A$

**Heyting Algebra Law**

**Residuation**       $C \cap A \subseteq B$  if and only if  $C \subseteq A \Rightarrow B$

**co-Heyting Algebra Law**

**Co-Residuation**       $A \setminus B \subseteq C$  if and only if  $A \subseteq B \cup C$

**Boolean Algebra Law**

**Double Negation**       $A^{**} = A$

**Distributive Lattice Laws**

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

**Other Useful Laws**

**De Morgan's Laws**       $(A \cap B)^* = A^* \cup B^*$   
 $(A \cup B)^* = A^* \cap B^*$

---

Several remarks are in order.

- The Semilattice Laws describe how  $\cap$  and  $\cup$  behave without any interaction between the two. In fact, any binary operation that is

associative, commutative and idempotent is called a **semilattice operation**.

- The Lattice Laws describe how  $\cap$  and  $\cup$  interact. The two Absorption Laws together are equivalent to the Ordering Law. For suppose  $A \cup (B \cap A) = A$  holds for all  $A$  and  $B$ . Now suppose  $X = X \cap Y$ . Then  $Y \cup X = Y \cup (X \cap Y) = Y$ . Conversely, suppose  $B \cap A = A$  implies  $B \cup A = B$  for all  $A$  and  $B$ . Then  $(X \cap Y) = (X \cap Y) \cap X$ . So  $X \cup (X \cap Y) = X$ .
- Some of the remaining laws are stated in terms of  $\subseteq$  instead of equality. Notice that  $A \subseteq B$  is equivalent to  $A = B \cap A$  and also equivalent to  $B = A \cup B$ , because of the Ordering Laws. So, we could state any law involving  $\subseteq$  in terms of equality. Though that is possible, it is not particularly helpful.
- The Bounded Lattice Laws indicate that  $U$  is the unit element for  $\cap$  and  $\emptyset$  is the unit element for  $\cup$ . It follows that  $\emptyset$  is the smallest element of  $\mathcal{P}(U)$  and  $U$  is the largest.
- The Residuation and Co-residuation Laws show that  $A \Rightarrow B$  and  $A \setminus B$  are defined as *duals* of one another.
- In the Double Negation Law recall that  $A^*$  is defined to be  $A \Rightarrow \emptyset$ . Since  $A^* \subseteq A \Rightarrow \emptyset$ , it follows from Residuation that  $A \cap A^* \subseteq \emptyset$ , which by Commutativity is equivalent to  $A^* \cap A \subseteq \emptyset$ , and also equivalent to  $A \subseteq A^* \Rightarrow \emptyset = A^{**}$ , again by Residuation. So in fact, in order to have  $A = A^{**}$ , we only need to insist that  $A^{**} \subseteq A$ .
- With respect to Distributivity, it is worth noticing that in any lattice the inclusions  $A \cap B \subseteq A \cap (B \cup C)$  and  $A \cap C \subseteq A \cap (B \cup C)$  hold. So again, the other inclusions are the ones that are special.
- Similarly, the de Morgan laws are “half-true” in any lattice where  $*$  is defined:  $A^* \cup B^* \subseteq (A \cap B)^*$  and  $(A \cup B)^* \subseteq A^* \cap B^*$ .
- The Heyting (or co-Heyting) Law implies Distributivity.

Let’s prove the distributivity laws. For the first one, we will use the residuation law and the characterization of unions of page 168. Suppose that we have arbitrary  $A, B, C, D \subseteq U$ . Then, we have the following equivalences:

$$\begin{aligned}
 A \cap (B \cup C) \subseteq D &\iff B \cup C \subseteq A \Rightarrow D && \text{— residuation} \\
 &\iff B \subseteq A \Rightarrow D \text{ and } C \subseteq A \Rightarrow D && \text{— char. union} \\
 &\iff A \cap B \subseteq D \text{ and } A \cap C \subseteq D && \text{— residuation} \\
 &\iff (A \cap B) \cup (A \cap C) \subseteq D && \text{— char. union}
 \end{aligned}$$

Since  $D$  is an arbitrary set, we deduce  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ .

For the second distributivity law, we will follow a series of equalities using the absorption laws and the first distributivity law. (We will also use the commutativity of the operations without mentioning it.)

$$\begin{aligned}
 (A \cup B) \cap (A \cup C) &= ((A \cup B) \cap A) \cup ((A \cup B) \cap C) && \text{— first distr. law} \\
 &= A \cup ((A \cup B) \cap C) && \text{— absorption} \\
 &= A \cup ((A \cap C) \cup (B \cap C)) && \text{— first distr. law} \\
 &= (A \cup (A \cap C)) \cup (B \cap C) && \text{— associativity} \\
 &= A \cup (B \cap C) && \text{— absorption}
 \end{aligned}$$



### EXERCISES:

Assume that  $U$  is a set, and all other sets are subsets of  $U$ .

103. Prove, using only the Semilattice Laws, the Lattice Laws and the Bounded Lattice Laws, that  $A \cap \emptyset = \emptyset$ . Likewise, show that  $A \cup U = U$ .
104. We have derived the second distributivity law from the first distributivity law (using the Semilattice and Lattice Laws). Actually, both distributivity laws are equivalent. Prove, using only the Semilattice and Lattice Laws, that if  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$  holds for all  $A, B, C$ , then so does  $(A \cap B) \cup (A \cap C) = A \cap (B \cup C)$ .
105. Prove, using only the Semilattice, Lattice and co-Heyting Algebra Law, that the second Distributivity Law holds.
106. Prove, using only the Semilattice, Lattice, Heyting Algebra and Boolean Algebra Laws, that the two de Morgan's Laws hold.
107. Prove that  $A \Rightarrow B = A^* \cup B$  using any method.
108. Prove that  $A \Rightarrow \emptyset = U \setminus A$  using any method.

### *Cartesian Products and Binary Relations*

Given two elements  $x$  and  $y$ , we can form the set  $\{x, y\}$ , which is the set containing only  $x$  and  $y$  as elements. But, as we know, none of them is the first element nor the second one, because the set is completely determined by its elements, independently from the order in which we denote the set. That is,

$$\{x, y\} = \{y, x\}.$$

But, for many reasons, there is the need for a construction of an entity  $\langle x, y \rangle$  containing  $x$  as “its first component” and  $y$  as “its second component.” In principle, this does not need to be a set, *per se*. But if one is committed to the idea that all things are constructed as sets, there are many different ways to construct a set that corresponds to an ordered pair. Probably, the most famous one is Kuratowski’s **ordered pair**, given by

$$\langle x, y \rangle = \{\{x\}, \{x, y\}\}.$$

But, the important property of the ordered pair, independently of how we define it, is that given elements  $x, y, a, b$ , we have that

$$\langle x, y \rangle = \langle a, b \rangle \iff x = a \text{ and } y = b.$$

Our development of lists provides another, obvious way to construct ordered pairs.

$$\langle x, y \rangle = [x, y].$$

No matter how we think of constructing  $\langle x, y \rangle$ , given two sets  $X$  and  $Y$ , we can collect all the ordered pairs  $\langle x, y \rangle$ , in which  $x \in X$  and  $y \in Y$  in a single set.

---

**DEFINITION 36:** Cartesian Product

---

*The Cartesian product of two sets  $X$  and  $Y$  is the set*

$$X \times Y = \{\langle x, y \rangle : x \in X \text{ and } y \in Y\}.$$


---

---

**EXAMPLE 35:**

---

Consider the sets  $\underline{3} = \{0, 1, 2\}$  and  $A = \{a, b\}$ . We can construct with them two different Cartesian products:

$$\begin{aligned} \underline{3} \times A &= \{\langle 0, a \rangle, \langle 0, b \rangle, \langle 1, a \rangle, \langle 1, b \rangle, \langle 2, a \rangle, \langle 2, b \rangle\}, \\ A \times \underline{3} &= \{\langle a, 0 \rangle, \langle b, 0 \rangle, \langle a, 1 \rangle, \langle b, 1 \rangle, \langle a, 2 \rangle, \langle b, 2 \rangle\}. \end{aligned}$$

Therefore, in general  $X \times Y \neq Y \times X$ .

---

---

**EXAMPLE 36:**


---

Recall from Example 16 of Chapter 7 the sets  $\text{Suit} = \{\clubsuit, \diamondsuit, \spadesuit, \heartsuit\}$  and  $\text{Rank} = \{A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K\}$ . We can recover the set Deck as the Cartesian product

$$\text{Deck} = \text{Rank} \times \text{Suit}.$$


---

Cartesian products are very useful. In particular, we can use them to encode binary relations between sets. Given two sets  $A$  and  $B$  and a binary relation  $R: A \dashrightarrow B$ . We can encode the relation  $R$  as a subset  $\theta_R \subset A \times B$  in the following way:

$$\theta_R = \{\langle a, b \rangle \in A \times B : a R b\}.$$

That is, we collect all the pairs  $\langle a, b \rangle$  with  $a \in A$  and  $b \in B$  such that  $a R b$  holds. Thus, for all  $a \in A$  and  $b \in B$ , we have

$$\langle a, b \rangle \in \theta_R \iff a R b.$$

Reciprocally, given a subset  $\theta \subset A \times B$ , we can define a binary relation  $R_\theta: A \dashrightarrow B$  by the rule

$$a R_\theta b \iff \langle a, b \rangle \in \theta.$$

In that sense, relations from a set  $A$  to a set  $B$  can be identified with the subsets of  $A \times B$ . Mathematicians do this out of habit. Thus, given a subset  $\theta \subseteq A \times B$ , and elements  $a \in A$  and  $b \in B$ , we may denote  $\langle a, b \rangle \in \theta$  by  $a \theta b$  without further remark.

---

**EXAMPLE 37:**


---

Recall that for any given set  $A$ , we have two relations  $A \dashrightarrow A$ : the diagonal relation  $\Delta_A$  and the total relation  $\nabla_A$ . They are given by the rules  $x \Delta_A y$  if and only if  $x = y$ , and  $x \nabla_A y$ , for all  $x, y \in A$ . We can view these relations as the following subsets of  $A \times A$ .

$$\Delta_A = \{\langle x, y \rangle \in A \times A : x = y\} = \{\langle x, x \rangle : x \in A\} \subseteq A \times A$$

$$\nabla_A = \{\langle x, y \rangle \in A \times A : x, y \in A\} = A \times A.$$

Thus, if  $A = \{0, 1, 2\}$ , we have  $\Delta_A = \{\langle 0, 0 \rangle, \langle 1, 1 \rangle, \langle 2, 2 \rangle\}$  and  $\nabla_A = \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 2, 0 \rangle, \langle 2, 1 \rangle, \langle 2, 2 \rangle\}$ .

---

---

**EXAMPLE 38:**


---

Consider the standard order of the natural number  $\leq$  restricted to the set  $A = \{0, 1, 2, 3\}$ . This relation, is

$$R_{\leq} = \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 0, 3 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 2 \rangle, \langle 2, 3 \rangle\}.$$

Notice that, indeed, given any numbers  $a, b \in A$ ,

$$\langle a, b \rangle \in R_{\leq} \iff a \leq b.$$


---

### *Binary Relations and Quantification*

We might want to define sets such as the set of perfect squares. One way would be to pick out those natural numbers of the form  $n^2$  for some  $n$ . Informally, we would like to define

$$\{n \in \mathbb{N} \mid m^2 = n \text{ for some } m \in \mathbb{N}\}.$$

We make this precise by introducing a formal interpretation for “for some”. Mathematicians use the symbol  $\exists$  for this purpose. The dual concept is “for all” (written  $\forall$ ), which we discuss later.

Looking at the informal definition of the set of perfect squares, we see that it involves a subset of  $\mathbb{N} \times \mathbb{N}$ . Namely,  $E' = \{(m, n) \in \mathbb{N} \times \mathbb{N} \mid m^2 = n\}$ , which according to the previous section corresponds to a relation from  $\mathbb{N}$  to  $\mathbb{N}$ . Now the perfect squares are those  $n$ ’s for which  $(m, n) \in E'$  for some  $m$ . So the perfect squares are defined by virtue of being related to something else – in this case to a natural number that is the square root of the number in question.

Consider a set  $S$  modeling the students at Chapman and a set  $M$  modeling majors the university offers, then “majors in” is a relation between elements of  $S$  and elements of  $M$ . Perhaps “**Jethro** majors in **Phrenology**” is true, while “**Aczel** majors in **Flim-Flam Studies**” is not. We can model “majors in” as a subset of  $S \times M$ : the pair  $(s, m)$  is in the set if  $s$  majors in  $m$ .

For a more mathematical example, “is less than” is a relation between natural numbers and natural numbers. So “5 is less than 6” is true while “9 is less than 6” is not. We may think of  $<$  as a subset of  $\mathbb{N} \times \mathbb{N}$ .

Notice that neither “majors in” nor  $<$  are obviously modeled by a function because, in the first example, some students may not have a declared major and some students may be double majors. In the second example, any number is related to infinitely many other numbers. So we need the new idea of *relation*.



There are several equivalent ways to model the general notion. But the following is usually taken to be the official version.

---

DEFINITION 37:

---

A **relation between  $X$  and  $Y$**  is a subset  $R \subseteq X \times Y$ . We can write  $x R y$  instead of  $(x, y) \in R$  to mimic familiar examples. A **relation on  $X$**  is a relation between  $X$  and  $X$ .

---

Here are three common equivalent formulations.

- $R \subseteq X \times Y$  determines a characteristic function  $\text{ch}_R: X \times Y \rightarrow \mathbf{2}$  so that  $R = \{(x, y) \in X \times Y \mid \text{ch}_R(x, y) = \text{TRUE}\}$ .
- A characteristic function  $k: X \times Y \rightarrow \mathbf{2}$  determines a curried function  $\lambda k: X \rightarrow \mathbf{2}^Y$  so that  $\lambda k(x)(y) = k(x, y)$ .
- The correspondence between  $\mathbf{2}^Y$  and  $\mathcal{P}(Y)$  means that a function  $f: X \rightarrow \mathbf{2}^Y$  determines a function from  $X$  to  $\mathcal{P}(Y)$  by the rule  $x \mapsto \{y \in Y \mid f(x)(y) = \text{TRUE}\}$ .
- A function  $F: X \rightarrow \mathcal{P}(Y)$  determines a subset  $\{(x, y) \in X \times Y \mid y \in F(x)\}$ .

The ability to move between these can be helpful. So it is worth practicing.



#### EXERCISES:

---

109. Describe  $E' = \{(m, n) \in \mathbb{N} \times \mathbb{N} \mid m^2 = n\}$  as an equalizer. That is, define two functions  $f$  and  $g$  from  $\mathbb{N} \times \mathbb{N}$  so that  $E' = \{p \in \mathbb{N} \times \mathbb{N} \mid f(p) = g(p)\}$ . [Hint: Using projections on  $p \in \mathbb{N} \times \mathbb{N}$ , we can always regard  $p$  as a pair  $(\text{pr}_{\mathbb{N}, \mathbb{N}}(p), \text{pr}'_{\mathbb{N}, \mathbb{N}}(p))$ .]
110. Let  $r: \mathbb{N} \times \mathbb{N} \rightarrow \mathbf{2}$  be the function defined by

$$r(m, n) = \begin{cases} \text{TRUE} & \text{if } m = n^2 \\ \text{FALSE} & \text{otherwise.} \end{cases}$$

Consider the relation  $R = r^{-1}(\text{TRUE})$ . Explain your answers for the following.

- Is it the case that  $3 R 2$ ?
- Is it the case that  $9 R 3$ ?
- Give an example for which  $\text{ch}_R(m, n) = \text{TRUE}$ . Give an example for which  $\text{ch}_R(m, n) = \text{FALSE}$ .

- (d) Calculate  $\lambda r(m)$  for all values of  $m$  less than 10.  
 (e) What is  $\lambda r(100)$ ?

---

We will discuss the structure of relations in more detail in later.

### *Quantifiers and Completeness*

Consider again how we might try to define the set of perfect square natural numbers. We would be right to define this set by

$$\{n \in \mathbb{N} \mid \text{for some } m \in \mathbb{N}, m^2 = n\}.$$

To make sense of this, we need to understand what “for some  $m \in \mathbb{N}$ , ...” means formally. Just as we abbreviated “and” with symbol  $\wedge$  and “or” with  $\vee$ , we will abbreviate “for some  $m \in \mathbb{N}$ , ...” with  $\exists m \in \mathbb{N}, \dots$

---

#### LEMMA 15: Existential and Universal Quantification

---

For sets  $U$  and  $W$  and relation  $R \subseteq U \times W$ , there is a subset of  $U$ , denoted by  $\{x \in U \mid \exists w \in W. x R w\}$  so that for all sets  $C \subseteq U$ ,

$$\{x \in U \mid \exists w \in W. x R w\} \subseteq C \text{ if and only if } R \subseteq C \times W.$$

Dually, there is a subset of  $U$ , denoted by  $\{x \in U \mid \forall w \in W. x R w\}$  so that for all  $C \subseteq U$ ,

$$C \subseteq \{x \in U \mid \forall w \in W. x R w\} \text{ if and only if } C \times W \subseteq R.$$

*Proof.* The relation  $R$  determines a function  $r$  from  $U$  to  $\mathcal{P}(W)$  by  $x \mapsto \{w \in W \mid x R w\}$ . Specifically,  $\text{ch}_R: U \times W \rightarrow \mathbf{2}$  is the characteristic map for  $R$ , and  $\text{ch}_R^\dagger: U \rightarrow \mathbf{2}^W$  determines  $r$  by  $w \in r(x)$  if and only if  $\text{ch}_R^\dagger(w)(x) = \text{TRUE}$ .

Now define

$$\begin{aligned} \{x \in U \mid \exists w \in W. x R w\} &= r^-(\mathcal{P}(W) \setminus \{\emptyset\}) \\ \{x \in U \mid \forall w \in W. x R w\} &= r^-(\{W\}) \end{aligned}$$

Proving that these sets satisfy the desired conditions is technical, but routine.

Concretely,  $\{x \in U \mid \exists w \in W. x R w\}$  consists of those  $x \in U$  so that  $x R w$  for some  $w \in W$ ;  $\{x \in U \mid \forall w \in W. x R w\}$  consists of those  $x \in U$  so that  $x R w$  for all  $w \in W$ . This justifies our notation:  $\exists w \in W. x R w$  is read as “there exists  $w \in W$  satisfying  $x R w$ ,”  $\forall w \in W. x R w$  is read as “for all  $w \in W$ ,  $x R w$ .”  $\square$

---

---

**EXAMPLE 39:**


---

Expressions using  $\exists$  and  $\forall$  can be nested to define complicated sets. Suppose we wish to define the set of all functions  $\mathbb{R} \rightarrow \mathbb{R}$  that are continuous at  $a$ . So we are interested in a subset of  $\mathbb{R}^{\mathbb{R}}$ .

The usual formal definition of “continuity at  $a$ ” is complicated. It says that for every positive real number  $\epsilon$  there is a positive real number  $\delta$  so that for every input  $x$  that is close to  $a$  — namely, a distance less than  $\delta$  from  $a$  — the result  $f(x)$  is close to  $f(a)$  — namely, at a distance less than  $\epsilon$  from  $f(a)$ .

First, consider how we define the set of positive real numbers. We can notice that a real number is non-negative if it is at least as big as some squared number. So, define

$$\mathbb{R}^+ = \{x \in \mathbb{R} \mid \exists r \in \mathbb{R}, r^2 < x\}.$$

Now saying  $\epsilon$  is a positive real number is the same as saying that  $\epsilon \in \mathbb{R}^+$ .

Using these, we can define the set of real numbers that are “close” to  $a$ . For  $a \in \mathbb{R}$  and  $\epsilon \in \mathbb{R}^+$ , define

$$B(a, \epsilon) = \{x \in \mathbb{R} \mid (x - a)^2 < \epsilon^2\}.$$

This consists of those real numbers that are less than  $\epsilon$  distant from  $a$ .

Next, we can define  $C_0(a)$  to be the set of functions that are continuous at  $a$  by

$$C_0(a) = \{f \in \mathbb{R}^{\mathbb{R}} \mid \forall \epsilon \in \mathbb{R}^+ \exists \delta \in \mathbb{R}^+, B(a, \delta) \subseteq f^{-1}(B(f(a), \epsilon))\}.$$

So a function  $f$  belongs to this set if and only if for every  $\epsilon > 0$ , there is a  $\delta > 0$  so that for all  $x \in \mathbb{R}$ , if  $(x - a)^2 < \delta^2$  then  $(f(a) - f(x))^2 < \epsilon^2$  — or saying it more plainly, for any  $x$  that is close to  $a$ ,  $f(x)$  is close to  $f(a)$ . The reader who is familiar with calculus will recognize that this is the precise definition of *continuity at  $a$* .

---




---

**EXERCISES:**


---

111. Using the definition of continuity at  $a$  in the previous example, define the set of functions from  $\mathbb{R}$  to  $\mathbb{R}$  that are continuous everywhere.
- 

We can use  $\exists$  and  $\forall$  to generalize union and intersection to arbitrary collections of subsets of  $U$ .

---

**DEFINITION 38:** Unions
 

---

Let  $A: I \rightarrow \mathcal{P}(U)$  be a function into the powerset of  $U$ . We write  $A_i$  instead of  $A(i)$  to emphasize that each  $A_i$  is a subset of  $U$ . Then define

$$\bigcup_{i \in I} A_i = \{x \in U \mid \exists i \in I. x \in A_i\}.$$


---

According to Lemma 15,  $\bigcup_{i \in I} A_i$  is again a subset of  $U$ . For any  $C \subseteq U$ , it is the case that  $A_i \subseteq C$  for all  $i \in I$  if and only if  $\bigcup_{i \in I} A_i \subseteq C$ . So this generalizes  $\cup$  to the union of a family of subsets of  $U$ , rather than just two subsets. In this sense,  $\mathcal{P}(U)$  is **complete**. That is, the union of any family of subsets of  $U$  exists. This justifies saying that  $\mathcal{P}(U)$  is a **complete** Boolean algebra.




---

**EXERCISES:**


---

In the following, consider a family  $A: I \rightarrow \mathcal{P}(U)$ .

- 112. Show that  $\bigcup_{i \in I} A_i \subseteq C$  if and only if  $A_k \subseteq C$  holds every  $k \in I$ .
  - 113. Define  $\bigcap_{i \in I} A_i$  in analogy with  $\bigcup_{i \in I} A_i$ .
  - 114. For  $I = \emptyset$ , what is  $\bigcup_{i \in I} A_i$ ?
  - 115. For  $I = \emptyset$ , what is  $\bigcap_{i \in I} A_i$ ?
- 

### *Singletons and Atomicity*

The complete Boolean algebra  $\mathcal{P}(U)$  has another useful feature. It is **atomic**. This means, roughly, that all subsets are built from the simplest ones.

An **atom** of a Boolean algebra is an element with the property that it is strictly above the least element and yet there is nothing strictly between it and the least element (in this setting, the least element is  $\emptyset$ ). A singleton subset  $\{x\} \subseteq U$  is an atom of  $\mathcal{P}(U)$  because it is not empty and there are no other subsets lying between  $\emptyset$  and  $\{x\}$ . It will be helpful to know that  $x \mapsto \{x\}$  constitutes a function

---

LEMMA 16: Singleton subsets exist.

---

For any set  $U$ , the rule  $x \mapsto \{x\}$  determines a function  $\text{sg}: U \rightarrow \mathcal{P}(U)$ .

*Proof.* Let  $\Delta \subseteq U \times U$  denote the *diagonal relation* on  $U$ . That is,  $x \Delta y$  if and only if  $x = y$ . So  $\text{ch}_\Delta: U \times U \rightarrow \mathbf{2}$  is the characteristic function of  $\Delta$ . That is,  $\text{ch}_\Delta(x, y) = \text{TRUE}$  when  $x = y$  and  $\text{ch}_\Delta(x, y) = \text{FALSE}$  when  $x \neq y$ .

Let  $\text{sg}: U \rightarrow \mathcal{P}(U)$  be the unique function for which  $\exists_U \circ (\text{sg} \times \text{id}_U) = \text{ch}_\Delta$ . In other words  $\exists_U(\text{sg}(x), y) = \text{TRUE}$  if and only if  $x = y$ . Since  $\exists_U(\text{sg}(x), y) = \text{TRUE}$  if and only if  $y \in \text{sg}(x)$ , it is the case that  $y \in \text{sg}(x)$  if and only if  $x = y$ . So  $\text{sg}(x) = \{x\}$ . In other words,  $\text{sg}$  is defined by the desired rule.  $\square$

---

Now every subset of  $U$  is obtained as a union of singletons:  $A = \bigcup_{x \in A} \{x\}$ . For a complete Boolean algebra, this is what is meant by saying that  $\mathcal{P}(U)$  is a complete **atomic** Boolean algebra. [Atomicity is more subtle for Boolean algebras that are not complete.] Although we do not investigate this here, any complete atomic Boolean algebra has the same structure as  $\mathcal{P}(U)$  for some  $U$ . If we let  $\text{Atoms}(B)$  be the set of atoms of a Boolean algebra, then a complete atomic Boolean algebra  $B$  has the same structure as  $\mathcal{P}(\text{Atoms}(B))$ . And of course,  $\text{Atoms}(\mathcal{P}(U))$  is the set of singletons from  $U$ .

The complete Boolean structure (but not the atoms) of  $\mathcal{P}(U)$  is preserved by inverse images.

---

LEMMA 17: Inverse image preserves Boolean operations

---

For any function  $f: X \rightarrow Y$ , any  $B: I \rightarrow \mathcal{P}(Y)$ , it is the case that  $f^{-1}(\bigcup_{i \in I} B_i) = \bigcup_{i \in I} f^{-1}(B_i)$  and  $f^{-1}(\bigcap_{i \in I} B_i) = \bigcap_{i \in I} f^{-1}(B_i)$ . Also, for any  $B \subseteq Y$ ,  $f^{-1}(Y \setminus B) = X \setminus f^{-1}(B)$ .

*Proof.* For any  $x \in X$ , it is the case that  $x \in f^{-1}(\bigcup_{i \in I} B_i)$  if and only if  $f(x) \in \bigcup_{i \in I} B_i$  if and only if  $f(x) \in B_k$  for some  $k \in I$  if and only if  $x \in f^{-1}(B_k)$  for some  $k \in I$  if and only if  $x \in \bigcup_{i \in I} f^{-1}(B_i)$ . The proof for  $\bigcap$  is similar with “for some ...” replaced by “for all ...”.

For any element  $x \in X$ , it is the case that  $x \in f^{-1}(Y \setminus B)$  if and only if  $f(x) \in Y \setminus B$  if and only if  $f(x) \notin B$  (because  $f(x)$  is guaranteed to belong to  $Y$ ) if and only if  $x \notin f^{-1}(B)$  if and only if  $x \in X \setminus f^{-1}(B)$  (because we assumed  $x \in X$ ).  $\square$

---



## EXERCISES:

116. Write out  $\mathcal{P}(\{a, b, c\})$
117. Write out  $\mathcal{P}(\emptyset)$
118. Is it the case that  $\emptyset \in \mathcal{P}(A)$  for any set  $A$ ? Explain.
119. Write out  $\mathcal{P}(2 \times 2)$  and  $\mathcal{P}(\mathcal{P}(2))$ . Pay attention to writing them in a systematic way, so that it is clear you have actually listed everything.
120. I claim that  $\mathcal{P}(\emptyset)$  is a terminal set (meaning that for any set  $A$  there is precisely one function from  $A$  to  $\mathcal{P}(\emptyset)$ ). Justify the claim.

### Images

We have noted that  $\mathcal{P}(U)$  is complete and atomic. Looked at the right way, this characterizes  $\mathcal{P}(U)$  in another important universal way. For a function  $f: A \rightarrow B$  the function  $f^-: \mathcal{P}(B) \rightarrow \mathcal{P}(A)$  has what is known as an *left adjoint*.

---

#### DEFINITION 39: Forward image

---

For  $f: X \rightarrow Y$ , define  $f^+: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$  by the rule  $A \mapsto \{y \in Y \mid \exists x \in A. f(x) = y\}$ . The subset  $f^+(A) \subseteq Y$  is called the **forward image of  $A$  with respect to  $f$** .

---

An important property of  $f^+$  is the following relation to  $f^-$ .

---

#### LEMMA 18: Forward image is adjoint to inverse image

---

Consider any function  $f: X \rightarrow Y$ . For any  $A \subseteq X$  and  $B \subseteq Y$ ,  $f^+(A) \subseteq B$  if and only if  $A \subseteq f^-(B)$ .

*Proof.* Suppose  $f^+(A) \subseteq B$ . For  $x \in A$ ,  $f(x) \in f^+(A)$ . So  $f(x) \in B$ . By definition, this means  $x \in f^-(B)$ . This shows that  $A \subseteq f^-(B)$ . Conversely, suppose  $A \subseteq f^-(B)$ . For  $y \in f^+(A)$ , there is some  $x \in A$  so that  $f(x) = y$ . So there is some  $x \in f^-(B)$  so that  $f(x) = y$ . Thus  $y \in B$ . This shows that  $f^+(A) \subseteq B$ . □

---

There is also a right adjoint of  $f^-$ , but as it is less commonly used, we do not investigate it here.

The important features of  $f^+$  are easily checked. First,  $f^+$  preserves atoms. That is,  $f^+(\{x\}) = \{f(x)\}$ . Second,  $f^+$  preserves all unions. So  $f^+(\bigcup_{i \in I} A_i) = \bigcup_{i \in I} f^+(A_i)$ . But  $f^+$  does not necessarily preserve intersections. This suggests that, with respect to  $f^+$ , the important features of  $\mathcal{P}(-)$  are the atoms and  $\bigcup$ . Recall that  $\mathcal{P}(U)$  is *generated by atoms*. That is, every subset of  $U$  is a union of singletons.

Now we can say that  $\mathcal{P}(U)$  is the *free* complete join semilattice generated by  $U$ . Notice that  $\bigcup$  is a function from sets of subsets of  $U$  to subsets of  $U$ . That is, it is a function  $\bigcup: \mathcal{P}(\mathcal{P}(U)) \rightarrow \mathcal{P}(U)$ . It satisfies two important equations:  $\bigcup\{A\} = A$  for any  $A \in \mathcal{P}(U)$ , and  $\bigcup(\bigcup^+(\mathcal{A})) = \bigcup(\bigcup(\mathcal{A}))$  for any  $\mathcal{A} \in \mathcal{P}(\mathcal{P}(U))$ . Suppose  $S$  is a set and  $J: \mathcal{P}(S) \rightarrow S$  is a function that behaves similarly. That is,  $J(\{x\}) = x$  for all  $x \in S$  and  $J(J^+(\mathcal{B})) = J(\bigcup(\mathcal{B}))$  for any  $\mathcal{B} \in \mathcal{P}(S)$ . We call  $(S, J)$  a **complete join semilattice**. Such a structure comes naturally with a partial order:  $a \leq b$  iff  $J(\{a, b\}) = b$ . In that partial order, every subset  $B$  has an upper bound:  $J(\{b, J(B)\}) = J(B)$  for each  $b \in B$ . And  $J(B)$  is the smallest element with that property: if  $J(\{b, a\}) = a$  for all  $b \in B$ , then  $J(\{J(B), a\}) = a$ .

---

LEMMA 19: Powersets are free complete join semilattices

---

For any complete join lattice  $(S, J)$  and any function  $f: U \rightarrow S$ , there is exactly one function  $f^\dagger: \mathcal{P}(U) \rightarrow S$  so that (i)  $f(x) = f^\dagger(\{x\})$  for all  $x \in U$  and (ii)  $f^\dagger(\bigcup \mathcal{A}) = J(f^\dagger^+(\mathcal{A}))$  for all  $\mathcal{A} \in \mathcal{P}(\mathcal{P}(U))$ .

*Proof.* On any  $A \subseteq U$ , define  $f^\dagger(A) = J(f^+(A))$ . It is easy to check that  $f^\dagger(\{x\}) = J(\{f(x)\}) = f(x)$ . The other condition is also easily checked. Now suppose  $g: \mathcal{P}(U) \rightarrow S$  satisfies (i')  $f(x) = g(\{x\})$  for all  $x \in U$  and (ii')  $g(\bigcup \mathcal{A}) = J(g^+(\mathcal{A}))$  for all  $\mathcal{A} \in \mathcal{P}(\mathcal{P}(U))$ . Consider some  $A \in \mathcal{P}(U)$ . Then we can prove (in an exercise) that  $f^\dagger(A) = g(A)$ . □

---



#### EXERCISES:

---

121. Define a function  $f: X \rightarrow Y$  and two subsets  $A, B \subseteq X$  so that  $f^+(A) \cap f^+(B) \neq f^+(A \cap B)$ . Try to find the smallest example you can.
122. Prove that for any function  $f: X \rightarrow Y$  and any  $A \subseteq X$ , the inclusion  $A \subseteq f^-(f^+(A))$  holds.

123. Prove that for any function  $f: X \rightarrow Y$  and any  $B \subseteq Y$ , the inclusion  $f^+(f^-(B)) \subseteq B$  holds.
124. Prove that for any function  $f: Y \rightarrow X$ , it the case that  $f$  is a one-to-one function if and only if  $f^-(f^+(A)) \subseteq A$  for all  $A \subseteq X$ .
125. Prove that for any function  $f: Y \rightarrow X$ , it the case that  $f$  is an onto function if and only if  $B \subseteq f^+(f^-(B))$  for all  $B \subseteq Y$ .
126. Finish the proof of Lemma [19](#).
-



## *Classifications on Sets*

In this chapter we will discuss the problem of classification in Set Theory. That is, we will formalize, using set-theoretical techniques, our notion of dividing a collection of objects into classes. There are at least two ways of understanding the process of classification of objects of a certain collection:

1. Using a criterion: we could say that two objects are of the same kind if they share a certain property.
2. Partitioning the collection: we could distribute the objects in several piles.

It turns out that both notions are very much related. Indeed, the end result will be in both cases the same: if we have a criterion to determine when two objects are of the same kind, we can partition the collection into piles of objects of the same kind; reciprocally, if we have a collection distributed into several piles, we could just determine that two objects are of the same kind if they are in the same pile.

Formally, these two notions will give rise to different set-theoretical concepts: *equivalent relations* and *partitions*. But, we prove that they are actually interchangeable.

### *Equivalence Relations*

Given a collection of objects that we intend to classify according to a certain criterion, we will typically say that two objects are *of the same kind* if both of them have a particular property in common. We would like to analyze the very basic properties that that this notion satisfies. Some of them will look very trivial, but it will be useful to make them explicit in order to capture the right notion when we start our formalization.

1. Every object will be of the same kind as itself. There is not much to say about this property. It is just true that one and the same object cannot be nonidentical to itself and therefore, every object shares all its properties with itself always.
2. If an object is of the same kind of another, this latter will be of the same kind of the former. This is true because sharing a property is something that occurs (or not) simultaneously by both objects.
3. If two objects are of the same kind of a third one, then these two objects are of the same kind themselves. Again, this is true because of the very notion of “having a property in common.”

These three properties can be formalized in the notion of an *equivalence relation*, which is a particular kind of binary relation on a set. Recall that we can always formalize a binary relation  $R: A \dashrightarrow B$  as a subset  $\theta_R \subseteq A \times B$ , in such a way that for all  $a \in A$  and  $b \in B$ , we have that

$$\langle a, b \rangle \in \theta_R \iff a R b.$$

In that way, there is no danger in identifying  $R$  and  $\theta_R$  and use the same letter for both of them.

---

#### DEFINITION 40: Equivalence Relation

---

An **equivalence relation** on a set  $A$  is a binary relation  $\theta: A \dashrightarrow A$  satisfying the following properties for all  $a, b, c \in A$ :

1. Reflexivity:  $a \theta a$ ;
  2. Symmetry: if  $a \theta b$  then  $b \theta a$ ;
  3. Transitivity: if  $a \theta b$  and  $b \theta c$ , then  $a \theta c$ .
- 

---

#### EXAMPLE 40:

---

In every set  $A$ , the diagonal and total relations,  $\Delta_A$  and  $\nabla_A$ , respectively, are equivalence relations. In the case of the diagonal  $\Delta_A$ , this is because  $a \Delta_A b$  if and only if  $a = b$ , and it is obvious that  $a = a$ , if  $a = b$  then  $b = a$ , and that if  $a = b$  and  $b = c$ , then  $a = c$ . In the case of the total relation  $\nabla_A$ , this is true because all elements are related with each other, and therefore the three conditions are trivially satisfied.

---

---

**EXAMPLE 41:**


---

Let's define the relation  $\theta$  on the set  $\mathbb{N}$  of natural numbers by the following property: for all  $a, b \in \mathbb{N}$

$$a \theta b \iff \text{either } a \text{ and } b \text{ are both even or they are both odd.}$$

Thus, for instance  $0 \theta 2$  holds because both 0 and 2 are even, but  $0 \theta 1$  doesn't hold, because 0 is even but 1 is odd.

The relation  $\theta$  thus defines is an equivalence relation. Reflexivity is trivial, because every number  $a$  is either even or odd, and therefore  $a \theta a$ . If  $a \theta b$  then either  $a$  and  $b$  are both even, and therefore also  $b \theta a$ , or they are both odd, and then  $b \theta a$ . Either way, if  $a \theta b$  then  $b \theta a$ , proving the symmetry of  $\theta$ . Finally, if  $a \theta b$  and  $b \theta c$ , then we can distinguish two cases: if  $a$  is even, then  $b$  is also even, because  $a \theta b$ , and hence  $c$  is also even, because  $b \theta c$ ; on the other hand, if  $a$  is odd, then  $b$  and  $c$  will be also odd, respectively, because  $a \theta b$  and  $b \theta c$ , respectively. In any case, either  $a$  and  $c$  are both even or they are both odd. That is,  $a \theta c$ . This argument shows the transitivity of  $\theta$ .

---



---

**EXAMPLE 42:**


---

In the set  $A = \{1, 2, 3\}$ , consider the relation

$$\theta = \{\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle\}.$$

This relation is reflexive because  $\langle a, a \rangle \in \theta$  for  $a = 1, 2, 3$ , and therefore  $a \theta a$ . The relation is symmetric because, if  $\langle a, b \rangle \in \theta$  then also  $\langle b, a \rangle \in \theta$ . This is trivial if  $a = b$ . The other cases are when  $a = 1$  and  $b = 2$ , that is, the pair  $\langle 1, 2 \rangle$ , and reverse one, namely, the pair  $\langle 2, 1 \rangle$ . Finally, the transitivity of  $\theta$  can also be proven by a case analysis.

---

---

EXAMPLE 43:

Consider the relation  $\theta$  on the set  $\mathbb{R}$  of the real numbers given by

$$a \theta b \iff a - b \text{ is an integer.}$$

First, for all real number  $a$ , we have that  $a - a = 0$ , which is an integer, showing that  $a \theta a$ . If we suppose that  $a \theta b$ , we would have that  $a - b$  is an integer. Thus,  $b - a = -(a - b)$  would also be an integer, whence we obtain that  $b \theta a$ . This is the symmetry of  $\theta$ . Finally, if  $a \theta b$  and  $b \theta c$ , we would have that  $a - b$  and  $b - c$  are both integers. Since the sum of an integer is an integer again, we would have that  $a - c = (a - b) + (b - c)$  is also an integer, showing that  $a \theta c$ . This is the transitivity of  $\theta$ .

---

Given an equivalence relation  $\theta$  on a set  $A$  and given an element  $a \in A$ , we could consider the set of all the elements of  $A$  that are related to  $a$  by  $\theta$ , that is, the set of all the elements  $x \in A$  that are *equivalent* according to  $\theta$  with  $a$ . We call this set the *equivalence class* of  $a$ .

---

DEFINITION 41:

Given a set  $A$ , an equivalence relation  $\theta$  on  $A$ , and an element  $a \in A$ , the *equivalence class of  $a$  modulo  $\theta$*  is the set

$$[a]_{\theta} = \{x \in A : x \theta a\} \subseteq A.$$

We call  $a$  a **representative** of the equivalence class  $[a]_{\theta}$ .

---



---

EXAMPLE 44:

Recall the equivalence relation  $\theta$  on  $\mathbb{N}$  defined in Example 41 given by  $a \theta b$  if and only if  $a$  and  $b$  have the same parity. It's not difficult to see that, according to this relation, the number 0 is equivalent to all the even numbers and to no odd number, while the number 1 is equivalent to all the odd numbers, but to no even number. So we would have the following equivalence classes:

$$\begin{aligned} [0]_{\theta} &= \{x \in \mathbb{N} : x \theta 0\} = \{x \in \mathbb{N} : x \text{ is even}\}, \\ [1]_{\theta} &= \{x \in \mathbb{N} : x \theta 1\} = \{x \in \mathbb{N} : x \text{ is odd}\}. \end{aligned}$$

It is also easy to check that  $[0]_\theta = [2]_\theta = [4]_\theta$ , etc. and  $[1]_\theta = [3]_\theta = [5]_\theta$ , etc.

EXAMPLE 45:

Recall the equivalence relation  $\theta$  on the set  $A = \{1, 2, 3\}$  defined in Example 42. Since there are only three elements in  $A$ , we can compute the corresponding equivalence classes manually. Notice that  $1 \theta 1$  and  $2 \theta 1$  but  $3 \theta 1$  doesn't hold, thus  $[1]_\theta = \{1, 2\}$ . Analogously,  $[2]_\theta = \{1, 2\}$ , since  $1 \theta 2$  and  $2 \theta 2$ , but  $3 \theta 2$  doesn't hold. Finally, the only  $x \in A$  such that  $x \theta 3$  is  $x = 3$ , so  $[3]_\theta = \{3\}$ . We only have two different equivalence classes:

$$[1]_\theta = [2]_\theta \quad \text{and} \quad [3]_\theta.$$

We have two things to say about these examples. First, is that for every  $a \in A$ , we always have that  $a \in [a]_\theta$ , and this is because  $a \theta a$ . The second thing that we notice is that if we have  $b \in [a]_\theta$ , then this means that  $b \theta a$ , which means that  $a$  and  $b$  are equivalent modulo  $\theta$ . So,  $a$  and  $b$  should determine the same equivalence class. We will write this as a lemma.

LEMMA 20:

*Given a set  $A$  and an equivalence relation  $\theta$  on  $A$ , for all  $a, b \in A$ , we have that*

$$b \in [a]_\theta \iff [a]_\theta = [b]_\theta.$$

*Proof.* ( $\Rightarrow$ ) Suppose that  $b \in [a]_\theta$  and let's prove that  $[a]_\theta = [b]_\theta$ , by showing the double inclusion  $[a]_\theta \subseteq [b]_\theta$  and  $[b]_\theta \subseteq [a]_\theta$ . But first notice that  $b \in [a]_\theta$  means that  $b \theta a$  and, by symmetry,  $a \theta b$ . Now, if  $x \in [a]_\theta$ , then  $x \theta a$ , and since we also have  $a \theta b$ , we deduce that  $x \theta b$ , by transitivity. Therefore,  $x \in [b]_\theta$ . For the other inclusion, if we take  $x \in [b]_\theta$ , we have that  $x \theta b$ , and since we also have that  $b \theta a$ , we deduce by transitivity that  $x \theta a$ , that is,  $x \in [a]_\theta$ .

( $\Leftarrow$ ) Suppose now that  $[a]_\theta = [b]_\theta$ . Notice that, as we mentioned before,  $b \in [b]_\theta$ , by the reflexivity of  $\theta$ , whence we deduce that  $b \in [a]_\theta$ , since we are assuming that  $[b]_\theta$  and  $[a]_\theta$  are the same set. □

Recall that, in Definition 41 we call  $a$  a *representative* of the equivalence class  $[a]_\theta$ , not *the representative* of the class  $[a]_\theta$ . And this is because of the preceding lemma, now we know that every element of an equivalence class is a representative of this class.

## Partitions

The second way in which we can classify a collection of objects is by dividing the collection into several piles. As we did in the previous section, let's try to analyze this concept to extract its fundamental properties.

1. No pile should be empty. Indeed, it would be useless to have an empty pile. A pile doesn't help to classify the objects if it doesn't contain any object.
2. Two piles cannot share any object. This is also clear, because a given object should be in one pile or another, but not in two piles simultaneously.
3. Each object should be in a pile. Maybe there is an object by itself in a pile, this is not a problem, but the partition should be total, in that, we have to distribute all the objects into piles.

Now, if we have our collection as a set  $A$ , then the piles would be subsets  $B \subseteq A$ , that is,  $B \in \mathcal{P}(A)$ . And the collection of all the piles will be a set of subsets of  $A$ , that is, a set  $P \subseteq \mathcal{P}(A)$ .

---

### DEFINITION 42:

---

A **partition** of a set  $A$  is a set  $P \subseteq \mathcal{P}(A)$  such that

1.  $\emptyset \notin P$ ;
2. if  $X, Y \in P$  and  $X \neq Y$ , then  $X \cap Y = \emptyset$ ;
3. for all  $a \in A$ , there is some  $X \in P$  such that  $a \in X$ .

We call the elements of  $P$  the **parts** of the partition.

---

---

EXAMPLE 46:

---

We can partition the set  $\mathbb{N}$  of the natural numbers in even numbers and odd numbers. That is, we can consider the sets  $E = \{x \in \mathbb{N} : x \text{ is even}\}$  and  $O = \{x \in \mathbb{N} : x \text{ is odd}\}$ . Both sets are nonempty,  $E \cap O = \emptyset$ , and every natural number  $n$  is either even, and then  $x \in E$ , or odd, and  $x \in O$ . This shows that  $P = \{E, O\}$  is a partition of the natural numbers.

---

Partitions are a source of equivalence relations.

---



---

PROPOSITION 25:

---

*Every partition  $P$  of a set  $A$  induces an equivalence relation  $\theta_P$  on  $A$  given by*

$$a \theta_P b \iff \text{there is some } X \in P \text{ so that } a, b \in X.$$

*That is, two elements are related by  $\theta_P$  if and only if they belong to the same part of  $P$ . Moreover, the equivalence classes of  $\theta_P$  are the parts of  $P$ .*

*Proof.* We need to show that  $\theta_P$  is reflexive, symmetric, and transitive. Given any  $a \in A$ , there is some part  $X \in P$  so that  $a \in X$ , by the definition of a partition. This implies that  $a \theta_P a$ . Now, if  $a \theta_P b$ , then there is some part  $X \in P$  so that  $a, b \in X$ . This also witnesses the fact that  $b \theta_P a$ , showing the symmetry of  $\theta_P$ . Finally, if  $a \theta_P b$  and  $b \theta_P c$ , then there are parts  $X, Y \in P$  so that  $a, b \in X$  and  $b, c \in Y$ . Notice that  $b$  is an element common to  $X$  and  $Y$ , that is,  $b \in X \cap Y$ . Therefore,  $X$  and  $Y$  cannot be different, since by the definition of a partition, different parts should have empty intersection. Hence,  $X = Y$ , whence we obtain that  $a, c \in X$ , which shows that  $a \theta_P c$ , finishing the proof of the transitivity of  $\theta_P$ .

Now, suppose that  $X \in P$ . Then, it contains an element  $a \in X$ , because it cannot be empty. Let's see that  $[a]_{\theta_P} = X$ . Indeed, if  $b \in X$  then by definition we have that  $b \theta_P a$ , what implies that  $b \in [a]_{\theta_P}$ ; and if  $b \in [a]_{\theta_P}$ , then  $b \theta_P a$ , what means that there is some part  $Y \in P$  such that  $a, b \in Y$ . Since we already know that  $a \in X$  and different partitions are disjoint, we deduce that  $Y = X$  and hence  $b \in X$ . That is, we have shown that  $X = [a]_{\theta_P}$ , and therefore every part of  $P$  is an equivalence class of  $\theta_P$ . The other direction is also easy to prove since, given an equivalence class  $[a]_{\theta_P}$ , we can consider the  $X \in P$  so that  $a \in X$ , and show that  $X = [a]_{\theta_P}$ , and therefore every equivalence class is also a part of  $P$ .  $\square$

---

As we mentioned previously, both methods—equivalence relations and partitions—should be equivalent to each other. We have already seen that every partition induces an equivalence relation. Let’s see now how every equivalence relation induces a partition.

---

PROPOSITION 26:

---

*Every equivalence relation  $\theta$  on a set  $A$  induces a partition  $P_\theta$  of  $A$  given by*

$$P_\theta = \{[a]_\theta : a \in A\}.$$

*That is, the parts of  $P_\theta$  are the equivalence classes of  $\theta$ .*

*Proof.* We have to show that  $P_\theta = \{[a]_\theta : a \in A\}$  doesn’t contain the empty set, its elements are disjoint, and every element of  $A$  belongs to some element of  $P_\theta$ . All these properties are quite straightforward. Indeed,  $P_\theta$  doesn’t contain the empty set because the elements of  $P_\theta$  are equivalence classes and we have seen that every equivalence class  $[a]_\theta$  contains at least one element, namely,  $a \in [a]_\theta$ . By the way, this also shows that every element of  $A$  belongs to a part of  $P_\theta$ . Finally, if we have that  $[a]_\theta \cap [b]_\theta \neq \emptyset$ , this means that both classes share some element  $c \in [a]_\theta$  and  $c \in [b]_\theta$ . By Lemma 20, we deduce that  $[a]_\theta = [c]_\theta = [b]_\theta$ . This shows that if two equivalence classes are different, they will have empty intersection, what finishes our proof.  $\square$

---



---

EXAMPLE 47:

---

In Example 45, we computed the equivalence classes of the equivalence relation on  $A = \{1, 2, 3\}$  defined in Example 42. These are  $[1]_\theta = [2]_\theta = \{1, 2\}$  and  $[3]_\theta = \{3\}$ . Therefore, the partition induced by  $\theta$  is

$$P_\theta = \{\{1, 2\}, \{3\}\}.$$


---

We have seen then that equivalence relations and partitions induce each other. Let’s finish now to tie these relationship between both concepts.



---

**THEOREM 9:** Equivalence relations and partitions encode the same information

---

*Given an arbitrary set  $A$ , the equivalence relations on  $A$  and the partitions of  $A$  are in a bijective correspondence. More specifically:*

1. *If  $\theta$  is an equivalence relation on  $A$ ,  $P_\theta$  its induced partition, and  $\theta_{P_\theta}$  the corresponding induced equivalence relation, we have that  $\theta = \theta_{P_\theta}$ .*
2. *If  $P$  is a partition of  $A$ ,  $\theta_P$  its induced equivalence relation, and  $P_{\theta_P}$  the corresponding induced partition, we have that  $P = P_{\theta_P}$ .*

*Proof.* 1. Let  $\theta$ ,  $P_\theta$ , and  $\theta_{P_\theta}$  as in the statement. Notice that, by the corresponding definitions, for all  $a, b \in A$ , we have that

$$a \theta_{P_\theta} b \iff \text{for some } X \in P_\theta \text{ we have } a, b \in X.$$

But, since  $P_\theta = \{[x]_\theta : x \in A\}$ , this means that

$$\begin{aligned} a \theta_{P_\theta} b &\iff \text{for some } x \in A \text{ we have } a, b \in [x]_\theta \\ &\iff \text{for some } x \in A \text{ we have } a \theta x \text{ and } b \theta x \\ &\iff a \theta b. \end{aligned}$$

There,  $\theta = \theta_{P_\theta}$ .

2. Let  $P$ ,  $\theta_P$ , and  $P_{\theta_P}$  as in the statement. By definition, the parts of  $P_{\theta_P}$  are the equivalence classes of  $\theta_P$ , which we showed in Proposition 25 to be the parts of  $P$ . Therefore,  $P = P_{\theta_P}$ .  $\square$
- 

Last theorem says then that there is a correspondence between equivalence relations on a set  $A$  and partitions of  $A$ . Let's see a few examples.

---

**EXAMPLE 48:**

---

The diagonal relation  $\Delta_A$  corresponds to the partition with most elements  $P_{\Delta_A} = \{\{a\} : a \in A\}$ , in which every equivalence class contains exactly one element. Indeed,  $x \in [a]_{\Delta_A}$  if and only if  $x \Delta_A a$ , that is, if and only if  $x = a$ .

On the other extreme, the total equivalence relation  $\nabla_A$  corresponds to the partition  $P_{\nabla_A} = \{A\}$  in which there is only one equivalence class. This is true because for each  $a, b \in A$ , we have that  $a \nabla_A b$ , and therefore  $[a]_{\nabla_A} = [b]_{\nabla_A}$ , that is, there is only one class, which contains all the elements of  $A$ .

---

---

**EXAMPLE 49:**

---

The equivalence relation  $\theta$  on  $\mathbb{N}$  that classifies the natural numbers according to their parity (see Example 41) corresponds to the partition  $P_\theta = \{E, O\}$  of Example 46. Indeed, we calculated the equivalence classes of this relation in Example 44.

---

*Quotients*

One of the most useful tools in Mathematics is the notion of a *quotient* of a structure. Quotients are present in all branches of Mathematics: in Linear Algebra, we have the quotient of a vector space by a subspace, in Group Theory we have the quotient of a group by a normal subgroup, in Ring Theory we have the quotient of a ring by an ideal, etc. In all these cases, making a quotient consists on finding an appropriate way of classifying the elements of the structure, that is, finding a well-behaved equivalence relation. The point is to see each one of the classes of equivalence, not as a collection of elements, but as a whole, as an element in its own right. The good behavior of the equivalence relation is what will allow us to give a structure to the set of classes.

---

**DEFINITION 43:**

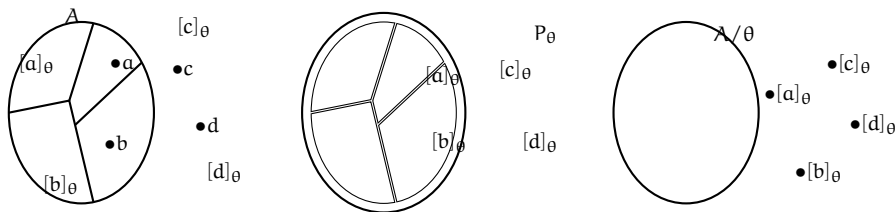
---

If  $A$  is a set and  $\theta$  is an equivalence relation, the **quotient** of  $A$  by  $\theta$  is the set

$$A/\theta = \{[a]_\theta : a \in A\}.$$


---

Notice that there is no formal distinction between the partition  $P_\theta$  induced by  $\theta$  (see Proposition 26) and the quotient  $A/\theta$ . Both are just the set of the equivalence classes of  $\theta$ . The distinction is conceptual: when we write  $P_\theta$  we are somehow looking at it as a set of subsets of  $A$ , while when we write  $A/\theta$  we want to emphasize the fact that this is a new set whose elements happen to be the equivalence classes of  $\theta$ .



In this situation we can define a function  $\pi_\theta: A \rightarrow A/\theta$  called the **canonical projection** of  $A$  onto  $A/\theta$  as follows: for every  $a \in A$ , we define  $\pi_\theta(a) = [a]_\theta$ . That is, to each element of  $A$ , the canonical projection assigns its equivalence class modulo  $\theta$ . It is easy to see that this map is a surjection, because every element of  $A/\theta$  is the image of some element of  $A$ . Indeed, every element of  $A/\theta$  is an equivalence class  $[x]_\theta$  for some  $x \in A$ , and therefore  $\pi_\theta(x) = [x]_\theta$ .

We will end this chapter with one of the most important theorems regarding the structure of sets and functions, which will have its corresponding analog in Linear Algebra, Ring Theory, Group Theory, etc.

---

**THEOREM 10:** First Isomorphism Theorem for Sets

---

Any given function  $f: A \rightarrow B$  decomposes as  $f = \iota \circ \tilde{f} \circ \pi$  for functions

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \pi \downarrow & & \uparrow \iota \\ C & \xrightarrow{\tilde{f}} & D \end{array}$$

so that  $\iota$  is an injection,  $\tilde{f}$  is a bijection, and  $\pi$  is a surjection.

*Proof.* First of all, consider the relation  $\theta_f$  on  $A$  given by  $a \theta_f b$  if and only if  $f(a) = f(b)$ . That is, two elements of  $A$  are related if and only if they have the same image by  $f$ . Obviously, this is a criterion of the form “two elements are related if and only if they share a property,” which implies that  $\theta_f$  is an equivalent relation. In what follows, we will denote it simply by  $\theta$ . Consider  $C = A/\theta$  and  $\pi = \pi_\theta: A \rightarrow A/\theta$  the canonical projection of  $A$  modulo  $\theta_f$ .

Consider also the set  $D = \{f(x) : x \in A\} \subset B$  of all the elements of  $B$  that are the image of some element of  $A$ . Obviously, this is a subset of  $B$  (called the *image* of  $A$  by  $f$ ), and the map  $\iota: D \rightarrow B$  given by  $\iota(b) = b$  is an injection.

Finally, we have to define the bijection  $\tilde{f}: A/\theta \rightarrow D$ . Notice that the elements of  $A/\theta$  are equivalence classes of the form  $[a]_\theta$  for some element  $a \in A$ . Thus, we can define the following function

$$\tilde{f}([a]_\theta) = f(a).$$

First of all, notice that this function uses a representative of the equivalence class  $[a]_\theta$ , namely, the element  $a$ , to define its image. Thus, we have to check that taking a different representative wouldn't give any problem. Thus, suppose that  $b$  is another representative of  $[a]_\theta$ . This means that  $a \theta b$ , and therefore  $f(a) = f(b)$ . So, we can

take indeed any representative we want, the function is indeed well defined.

In order to prove that  $\tilde{f}$  is a surjection, consider any element  $b \in B$ . By the definition of  $B$ , we have that  $b = f(x)$  for some  $x \in A$ . Therefore, we would have that  $\tilde{f}([x]_\theta) = f(x) = b$ . That is, we have found an element of  $A/\theta$ , namely  $[x]_\theta$ , so that its image by  $\tilde{f}$  is precisely the element  $b$ . Hence,  $\tilde{f}$  is a surjection.

Finally, we only need to check that  $\tilde{f}$  is also an injection. Suppose then that  $[a]_\theta$  and  $[b]_\theta$  are two elements of  $A/\theta$  so that  $\tilde{f}([a]_\theta) = \tilde{f}([b]_\theta)$ . By definition of  $\tilde{f}$ , we obtain that  $f(a) = f(b)$ , and therefor,  $a \theta b$ , that is,  $[a]_\theta = [b]_\theta$ . That is, we have shown that if two elements of  $A/\theta$  have the same image by  $\tilde{f}$ , then they are the same, as we needed to prove.  $\square$

---

## The Integers and Rationals

*Plus ça change, plus c'est la même chose.*

*[The more things change the more they stay the same.]*

— Jean-Baptiste Alphonse Karr

INTEGERS AND RATIONAL NUMBERS extend  $\mathbb{N}$  to allow for exact subtraction and division. By “exact” we mean subtraction and division as you know them, rather than the approximations  $m \dot{-} n$  and  $m // n$ . that we looked at in Chapter 3. Remember that in the natural numbers, subtraction can be handled approximately by the monus operation  $m \dot{-} n$ , and division by the quotient operation  $m // n$ . *Exact* subtraction should satisfy  $m - n = p$  if and only if  $m = n + p$ , whereas monus only satisfies  $m \dot{-} n \leq p$  if and only if  $m \leq n + p$ . Likewise, *exact* division should satisfy  $m \cdot n = p$  if and only if  $m = p/n$  for a non-zero  $n$ , whereas natural number quotient only satisfies  $m \cdot n \leq p$  if and only if  $m \leq p // n$ .

### Integer Numbers

To support exact subtraction, you know what is needed. The natural numbers must be extended with **negative numbers**, resulting in the **integers**. And likewise, to support exact division, the integers must be extended with **fractions**, resulting in the **rational numbers**. In this chapter, we explore these situations from two angles. First, what should count as extensions? Second, are the desired extensions even possible?

The first question is something like a software specification. We need to understand what we are looking for. The second question is about implementation: is there an actual way to build something that meets the specification?

The main technical ideas for finding these implementations are quite general. The ideas are relatively simple, but need to be put together in a chain that can be a bit confusing at first. It is worth

going through in detail, partly just to understand the integers and rational numbers better, but also partly to understand the general process.

Imagine you have a little DIY workshop in your garage. In it, you keep various tools and hardware. You have a little jar containing one inch brass machine screws, another containing  $\frac{3}{4}$  inch steel wood screws, another containing one inch zinc finish nails, and so on. So if you need a screw for some reason, you can just grab one from the right jar. You will not care exactly which screw you pick because all of the screws in a jar are the same. On the other hand, they are not *equal*. Somehow there is a difference between saying two screws are the same (both are the same length and gauge, have the same head shape, are made of the same metal, and so on), and saying they are the equal.

You can think of many other situations in which “same” and “equal” mean different things. So it is not a big surprise that this kind of distinction shows up in mathematics. It is also not a big surprise that, in the hands of mathematicians, the idea is turned to technical advantage.

The integers are intended to extend the natural numbers to support subtraction. Remember that the behavior of addition on the natural numbers is partly summarized in the fact that  $(\mathbb{N}, +, 0)$  is a monoid. So we seek a new monoid  $(\mathbb{Z}, +, 0)$  that includes  $(\mathbb{N}, +, 0)$  so that  $\mathbb{Z}$  has another operation  $-$  satisfying  $a = b + c$  if and only if  $a - b = c$  for every  $a, b, c \in \mathbb{Z}$ .

The idea of inclusion seems simple enough, but turns out to cause technical complications. By relaxing it very slightly, we get a more useable specification. So we will not require  $\mathbb{N}$  to be included directly in  $\mathbb{Z}$ , but will only require that every natural number corresponds to an integer. To make this clear, we will need to distinguish between addition of natural numbers and addition of integers. For this, we will write  $+_{\mathbb{Z}}$  for addition of integers.

The fact that addition is commutative means that we only need one new operation. If addition were not commutative, we would have to think about the subtraction “from the left” and “from the right”.

---

#### VOCABULARY 8: Basic Vocabulary of Integers

---

*For every natural number  $n$ , there is an integer  ${}^+n$ . So for example, 5 is a natural number, and  ${}^+5$  is the corresponding integer.*

*For any two integers  $a$  and  $b$ , there are integers called their sum and difference, written as  $a +_{\mathbb{Z}} b$  and  $a -_{\mathbb{Z}} b$ .*

---

For example,  ${}^+5 +_{\mathbb{Z}} {}^+7$  is an integer, and so is  ${}^+(5 + 7)$ . The first expression is meant to denote the sum of the two integers corresponding to 5 and 7. The second expression is meant to denote the

integer corresponding to  $5 + 7$ . But the vocabulary does not tell us how  ${}^+n$  is related to  $n$ , nor does it tell us how  ${}^+m +_{\mathbb{Z}} {}^+n$  is related to  $m + n$ . Of course, we intend that  ${}^+5$  and  $5$  should denote the same value. The only distinction is whether we are discussing integers or natural numbers. Also, adding two non-negative integers should be the same as adding two natural numbers.

---

POSTULATE 7: Natural numbers correspond to integers

---

*For natural numbers  $m$  and  $n$ , if  ${}^+m = {}^+n$  then  $m = n$ . Also  ${}^+m +_{\mathbb{Z}} {}^+n = {}^+(m + n)$ .*

---

The second part of this simply states that if we wish to add two natural numbers as integers, we can either add take them to be integers and add that way, or we can add them as natural numbers and take the result to be an integer.

Now we need to establish that addition works as expected and that subtraction is its inverse.

---

POSTULATE 8: The integers form a monoid with  ${}^+0$  as the identity

---

*For integers  $a$ ,  $b$  and  $c$ ,*

$$a +_{\mathbb{Z}} {}^+0 = a$$

*and*

$$a +_{\mathbb{Z}} (b +_{\mathbb{Z}} c) = (a +_{\mathbb{Z}} b) +_{\mathbb{Z}} c.$$


---

The next postulate stipulates that subtraction is inverse of addition.

---

POSTULATE 9: Addition and subtraction are inverses

---

*For any integers  $a$ ,  $b$  and  $c$ ,*

$$a = b +_{\mathbb{Z}} c \quad \text{if and only if} \quad a -_{\mathbb{Z}} b = c$$


---

For example, from this we see that  ${}^+5 -_{\mathbb{Z}} {}^+7$  is an integer  $c$  (we know it actually corresponds to negative 2) and that  ${}^+5 = {}^+7 +_{\mathbb{Z}} c$ .

These postulates summarize that the integers extend the natural numbers so that addition and subtraction are defined as they should be. But to reason generally about integers, we need an axiom specifying that there are no extras.

---

**POSTULATE 10:** Axiom of Induction of Integers
 

---

*No integers can be eliminated without violating Vocabulary 8.*

---

This postulate tells us that an integer can only be formed as  $+m$  for some natural number  $m$ , or by adding existing integers  $a +_{\mathbb{Z}} b$ , or by subtracting existing integers  $a -_{\mathbb{Z}} b$ . So, proofs about all integers can be organized inductively. Suppose we wish to prove some property  $P(a)$  is true for all integers  $a$ . If  $P(+m)$  holds for every natural number  $m$ , and for any integers  $b$  and  $c$ , if  $P(b)$  and  $P(c)$  hold, then so do  $P(b +_{\mathbb{Z}} c)$  and  $P(b -_{\mathbb{Z}} b)$ , then by Postulate 10, we may conclude that  $P(a)$  holds for all integers  $a$ .

The vocabulary and postulates specify how we expect the integers to behave, but they do not say there actually is a structure satisfying all these requirements. So our next goal is to construct a *model of the integers*.

Since the idea of integers is to keep track of positive and negative, let us start by considering a very simplified accounting idea known as a **balance sheet**. In real accounting, a balance sheet shows much more information, but at least it includes a summary of assets and a summary of liabilities. These are the two “bottom lines” of a balance sheet. Simplifying to just this, let us say a balance sheet is a pair of natural numbers. For example,  $(8, 2)$  indicates 8 (florins, let’s say) in assets and 2 in liabilities. So this balance has a net positive value of 6.

Addition of balance sheets is easy. Define  $(m, n) \oplus (p, q) = (m + p, n + q)$ . That is, to add balance sheets we just combine assets and combine liabilities. Clearly,  $(m, n) \oplus (0, 0) = (m, n)$ .


**EXERCISES:**


---

127. Show that  $\oplus$  makes balance sheets into a monoid.
- 

Two balance sheets might be different, but still have the same net value. For example,  $(8, 2)$  and  $(10, 4)$  both have a net value of 6. For the purpose of net value, these are *the same* even though they are not equal, just as two distinct one inch wood screws are the same for the purpose of attaching pieces of wood.

A very simple test for sameness involves only natural number addition. Namely,  $(m, n)$  and  $(p, q)$  are the same if  $m + q = n + p$ .

To make the notation simpler, let’s agree to use uppercase latin letters  $A, B, C, \dots$  to indicate balance sheets. In many programming languages, a nice implementation would let us extract the asset part

The word “same” is a bit of a problem because it is easy to think “same” might mean “equal”. So in general, we typically use the word **equivalent** to suggest that things are *the same for our purpose*, but not necessarily equal. Notationally, we often write this as some modified version of an equal sign. For example, for balances, I write  $(m, n) \equiv (p, q)$  to mean  $(m, n)$  and  $(p, q)$  have the same net value. This is pronounced “ $(m, n)$  is equivalent to  $(p, q)$ .”



of a balance sheet  $A$  as  $A.as$  and the liability part as  $A.li$ . In other words,  $A$  is the pair  $(A.as, A.li)$ .




---

EXERCISES:

---

128. Show that  $\equiv$  is reflexive, meaning that  $A \equiv A$  for any balance sheet  $A$ .
  129. Show that  $\equiv$  is transitive, meaning that if  $A \equiv B$  and  $B \equiv C$  then  $A \equiv C$ .
  130. Show that  $\equiv$  is symmetric, meaning that if  $A \equiv B$ , then  $B \equiv A$ .
  131. Show that  $\equiv$  is a congruence for  $\oplus$ , meaning that if  $A \equiv B$  and  $C \equiv D$ , then  $A \oplus C \equiv B \oplus D$ .
- 

These exercises show that  $\equiv$  acts almost like equality for  $\oplus$ . Think about what “same” can mean in plain English. Whether we are talking about balance sheets, wood screws, or boxes of cereal, it should be true that “ $A$  is the same as  $A$ ” is always true. Likewise, if “ $A$  is the same as  $B$ ” and “ $B$  is the same as  $C$ ” are true, then “ $A$  is the same as  $C$ ” is true. And if “ $A$  is the same as  $B$ ” is true, then “ $B$  is the same as  $A$ ” is also true. Thus the results of exercises 128, 129, and 130 are really just confirming that  $\equiv$  provides a reasonable interpretation of sameness for balance sheets. Exercise 131 expresses something else. Informally, it tells us that  $\oplus$  agrees with our interpretation of  $\equiv$ . As far as  $\oplus$  is concerned, equivalent arguments result in equivalent results.

Generically, any relation that is reflexive, transitive and symmetric is called a **equivalence relation** (it encodes some concept of sameness). That is, an equivalence relation specifies which elements are, for present purposes, “interchangeable” with which elements. For an operation like  $\oplus$ , an equivalence relation may be a **congruence**, meaning that the operation does not distinguish among interchangeable elements.

We now also want an operation corresponding to subtraction. Define  $A \ominus (p, q) = A \oplus (q, p)$ . For example,  $(8, 2) \ominus (5, 3) = (8 + 3, 2 + 5) = (11, 7)$ . So a net value of 6 ‘minus’ a net value of 2 has a net value of 4. That seems promising.




---

EXERCISES:

---

132. Show that  $\equiv$  is also a congruence for  $\ominus$ .
133. Show that  $A \equiv B \oplus C$  if and only if  $A \ominus B \equiv C$ .

The last exercise shows that balance sheets behave exactly as we want integers to behave, except that for integers we want  $a = b +_{\mathbb{Z}} c$  if and only if  $a -_{\mathbb{Z}} b = c$ . For balance sheets we only have  $A \equiv B \oplus C$  if and only if  $A \ominus B \equiv C$ . Apparently, to get a suitable model of the integers from balance sheets, we need a way to replace  $\equiv$  with  $=$ .

There are two main approaches (I call them “minimalist” and “maximalist”) to replacing a congruence relation with equality. In the minimalist approach, we throw away most balance sheets, keeping as few as possible. So if  $A \equiv B$ , then we need to keep at most one of the two. We might not need either one if we agree to keep some other one that is equivalent to  $A$  and  $B$ . On the other hand, for any particular  $A$ , we had better keep at least one equivalent sheet.

Consider the following operation on balance sheets. For any balance sheet  $A = (m, n)$ , let  $\hat{A} = (m \dot{-} n, n \dot{-} m)$ .



#### EXERCISES:

134. Show that  $A \equiv \hat{A}$  for any balance sheet  $A$ .
135. Show that  $A \equiv B$  if and only if  $\hat{A} = \hat{B}$  for any balance sheets  $A$  and  $B$ .
136. Show that for  $\hat{A} = (p, q)$  either  $p = 0$  or  $q = 0$  for any balance sheet  $A$ .

From these exercises, the balance sheets that have either 0 assets or 0 liabilities are all we need. Every balance sheet is equivalent to one. If we interchange some  $A$  with  $\hat{A}$ , and  $B$  with  $\hat{B}$ , then  $A \equiv B$  is determined by  $\hat{A} = \hat{B}$ . This means we can model the integers as follows.

Say a balance sheet is **simple** if it has either 0 assets or 0 liabilities. Let  $Z$  denote the collection of all simple balance sheets, and write  $a, b, c, \dots$  for simple balance sheets. Define  $a +_{\mathbb{Z}} b = \widehat{a \oplus b}$ . Define  $a -_{\mathbb{Z}} b = \widehat{a \ominus b}$ . For a natural number  $m$ , define  ${}^+m$  to be the simple balance sheet  $(m, 0)$ . Now it is pretty routine (and kind of boring, really) to check that simple balance sheets do exactly what we want. They support the basic vocabulary of integers and satisfy all the postulates. In other words, we could define the integers to *be* simple balance sheets.

A *positive* integer  ${}^+5$  is just the simple balance sheet  $(5, 0)$  — 5 in assets and no liabilities; a *negative* integer  ${}^-5$  is just  $(0, 5)$  — no assets and 5 in liabilities. As a model of the integers, this is pretty reasonable.

This minimalist approach is very useful computationally because it gives us a concrete way to calculate. For example,  $+5 + -7$  is calculated by first getting  $(5, 0) \oplus (0, 7) = (5, 7)$ , and then simplifying to  $(0, 2)$ .

But the minimalist approach is not ideal for other reasons. First, the operation  $\hat{\Delta}$  is not unique. Any other operation  $\check{\Delta}$  that satisfies  $A \equiv \check{\Delta}$ ,  $\check{\Delta} = \check{\Delta}$ , and if  $A \equiv B$ , then  $\check{\Delta} = \check{\Delta}$  would work. As it happens, there are actually infinitely many such operations. So  $\hat{\Delta}$  is not special. Second, though this is not a problem for integers, in other situations where equivalence relations arise, there may not be anything analogous to  $\hat{\Delta}$  whatsoever.

With a maximalist approach, we don't try to keep barely enough balance sheets. Instead, we go out to our garage workshop and realize that we could keep balance sheets in jars (not really, but you get the point). Each jar contains balance sheets that are interchangeable with each other. For example, the jar containing  $(2, 4)$  also contains  $(1, 3)$ ,  $(3, 5)$  and so on because these are all equivalent. Now we can define operations on the jars: Suppose I want to "add" Jar1 to Jar2. Then

- Pull an item from Jar1. Call it A.
- Pull an item from Jar2. Call it B.
- Calculate  $A \oplus B$ .
- Then the result is the jar containing  $A \oplus B$ .

This procedure finds the same resulting jar no matter which A and B get pulled. This is because equivalence is a congruence for  $\oplus$ . It is like the way we treat a jar of screws — just pull any screw from the jar and build something with it. You won't be able to tell the difference between that object and one you could have built with a different screw. Of course, that depends on knowing that the contents of each jar is *homogeneous* — it contains only equivalent screws. This wouldn't work if a jar contains, say, a mix of one inch and  $\frac{1}{2}$  inch screws. Then pulling a screw from that jar would really matter. The same goes for jars of balance sheets.

Also, the last step of the procedure doesn't work properly unless each jar is *complete*, containing *all* equivalent balance sheets.

To make all this work, we need to figure out what the homogeneous, complete jars are.

---

**DEFINITION 44:** Equivalence classes (aka homogeneous, complete jars)

---

Suppose  $\equiv$  is an equivalence relation on a collection  $X$ . Also suppose  $J$  is a subcollection of  $X$ . Then

- $J$  is occupied if it contains something.
- $J$  is homogeneous if  $A \in J$  and  $B \in J$  implies  $A \equiv B$  for every  $A, B \in J$  — so  $J$  consists only of indistinguishable elements.
- $J$  is complete if  $A \in J$  and  $A \equiv B$  implies  $B \in J$  for every  $A, B \in X$  — so  $J$  contains all elements indistinguishable from any of its elements.
- $J$  is an **equivalence class** if it is occupied, homogeneous and complete.

Suppose  $B \in X$  and  $\equiv$  is an equivalence relation on  $X$ . Then  $[B]_{\equiv}$  denotes the equivalence class containing  $B$ . In other words, for  $A \in [B]_{\equiv}$  if and only if  $A \equiv B$ .

The collection consisting all equivalence classes is denoted by  $X / \equiv$ .

---

The case we are concerned with is  $X$  being  $\mathbb{N} \times \mathbb{N}$ , the collection of balance sheets.

So informally,  $X / \equiv$  consists of all the “jars” that we use to sort elements in  $X$  according to  $\equiv$ .

Rather than minimally keeping one special balance sheet (a simple one) for each equivalence class (each jar), the maximalist approach just keeps all the equivalence classes (keeps the jars themselves). For the situation we are interested in now, we define the integers to be  $(\mathbb{N} \times \mathbb{N}) / \equiv$ , the collection of equivalence classes balance sheets. For example, the integer  $-4$  would be the collection of all balance sheets having negative net value 4. It contains  $(0, 4)$ ,  $(1, 5)$ ,  $(2, 6)$  and so on. Arithmetic on the equivalence classes is definable by

$$\begin{aligned} [A]_{\equiv} +_{\mathbb{Z}} [B]_{\equiv} &= [A \oplus B]_{\equiv} \\ [A]_{\equiv} -_{\mathbb{Z}} [B]_{\equiv} &= [A \ominus B]_{\equiv} \\ +_m &= [(m, 0)]_{\equiv} \end{aligned}$$

These make sense because  $\equiv$  is a congruence for  $\oplus$  and  $\ominus$ . Even though  $A \oplus B$  depends on the particular choice of  $A$  and  $B$ ,  $[A \oplus B]_{\equiv}$  does not. Namely, if  $A \equiv A'$  and  $B \equiv B'$ , then  $A \oplus B \equiv A' \oplus B'$ . So  $[A \oplus B]_{\equiv} = [A' \oplus B']_{\equiv}$ .

Here is another way to think about this. The notation  $[A]_{\equiv}$  refers to the equivalence class containing  $A$ . So  $A$  is simply a way to locate that equivalence class — it is a *name* for the equivalence class. It is a bit like saying “Bob’s family” or “Kate’s family”. If Bob and Kate are siblings, then these are two names for the same family. If  $A \equiv A'$ , then  $[A]_{\equiv} = [A']_{\equiv}$ . So  $A$  and  $A'$  are two names for the same class. The congruence property means that the names for equivalence classes do not matter for an operation.

In the minimalist approach, we keep one **representative** for each equivalence class. In the maximalist approach, we keep the classes themselves. This leads to a definition for the integers.

---

DEFINITION 45: The integers (maximalist approach)

---

The *integers* are modeled by the collection  $(\mathbb{N} \times \mathbb{N})/\equiv$ , where  $\equiv$  is the equivalence relation on pairs defined by  $(m, n) \equiv (p, q)$  if and only if  $m + q = p + n$ . This model of the integers is denoted by  $\mathbb{Z}$ .

Arithmetic on  $\mathbb{Z}$  is given by  $+\mathbb{Z}$  and  $-\mathbb{Z}$ , with  $[(0, 0)]_{\equiv}$  being the identity for addition. Also, the operation that converts a natural number  $m$  to an integer is defined by  $^+m = [(m, 0)]_{\equiv}$ .

---

It is now routine to check our work. Equivalence classes of balance sheets *are* the integers. Addition and subtraction make sense. The natural numbers are included in the form  $[(m, 0)]_{\equiv}$ . Let us now go for broke, and figure out what to do about multiplication.




---

#### EXERCISES:

---

In these exercises, define  $\otimes$  to be the operation on balance sheets given by

$$(m, n) \otimes (p, q) = (m \cdot p + n \cdot q, m \cdot q + n \cdot p).$$

137. Prove that  $\otimes$  is associative, commutative and has  $^+1$  as its identity.
  138. Prove that  $\equiv$  is a congruence relation for  $\otimes$ .
  139. Prove that  $\otimes$  respects natural numbers in the sense that  $^+m \otimes ^+n = ^+(m \cdot n)$  for any natural numbers  $m$  and  $n$ .
  140. Prove that for any balance sheets,  $A \otimes (B \oplus C) \equiv (A \otimes B) \oplus (A \otimes C)$ .
- 

These exercises show that multiplication on integers is definable by  $[(m, n)]_{\equiv} \cdot [(p, q)]_{\equiv} = [(m, n) \otimes (p, q)]_{\equiv}$ .

### Rational numbers

Rational numbers embody the familiar idea of fractions, as in  $\frac{5}{7}$ . But as you know  $\frac{5}{7}$  and  $\frac{10}{14}$  denote the same number. Different *fractions* can denote the same *rational number*. This should look familiar. Just as different balance sheets can denote the same integer, different fractions can denote the same rational number. This hints at the

idea that we can define a system rational numbers in analogy with integers by stipulated an equivalence relation on fractions, and checking that it does what we want. The construction of rational numbers is so closely analogous to the construction of  $\mathbb{Z}$  that I can leave most of the details to you.

Let us say that a **fraction** is simply a pair consisting of an integer (the **numerator**) and a positive natural number (the **denominator**). Usually, we write fractions in the usual notation:  $\frac{+5}{7}$ . But really a fraction is simply an element of  $\mathbb{Z} \times \mathbb{N}^+$ . Notice that the numerator is meant to be an integer, so  $+5$  makes that clear.

In the following, let  $\square$  be the operation on fractions defined by

$$\frac{a}{m} \square \frac{b}{n} = \frac{a \cdot \mathbb{Z} b}{m \cdot n}.$$

Also, define the relation  $\simeq$  between fractions by  $\frac{a}{m} \simeq \frac{b}{n}$  if and only if  $a \cdot +n = b \cdot +m$ .




---

#### EXERCISES:

---

141. Prove that  $\simeq$  is a congruence with respect to  $\square$ .
  142. Prove that for any fraction  $\frac{a}{m}$  such that  $a \neq +0$ , there is fraction  $\frac{b}{n}$  satisfying  $\frac{a}{m} \square \frac{b}{n} \simeq \frac{+1}{1}$ .
  143. Define addition  $\boxplus$  on fractions, and show that  $\simeq$  is a congruence with respect to  $\boxplus$ .
  144. For fractions  $\frac{a}{m}, \frac{b}{n}$  where  $b \neq +0$ , define a fraction  $\frac{a}{m} \boxdot \frac{b}{n}$  satisfying  $\frac{a}{m} \simeq \frac{b}{n} \square \left( \frac{a}{m} \boxdot \frac{b}{n} \right)$ . Show that  $\simeq$  is a congruence with respect to  $\boxdot$ .
  145. Define subtraction  $\boxminus$  on fractions, and show that  $\simeq$  is a congruence with respect to  $\boxminus$ .
- 

These exercises, with a few others that are just as routine, show that we can define the rationals  $\mathbb{Q}$  to be  $(\mathbb{Z}\mathbb{Z} \times \mathbb{N}^+)/\simeq$ . Then addition, multiplication, subtraction and division can all be defined on  $\mathbb{Q}$  as expected by

$$\begin{aligned} \left[ \frac{a}{m} \right]_{\simeq} +_{\mathbb{Q}} \left[ \frac{b}{n} \right]_{\simeq} &= \left[ \frac{a}{m} \boxplus \frac{b}{n} \right]_{\simeq} \\ \left[ \frac{a}{m} \right]_{\simeq} \cdot_{\mathbb{Q}} \left[ \frac{b}{n} \right]_{\simeq} &= \left[ \frac{a}{m} \boxdot \frac{b}{n} \right]_{\simeq} \\ \left[ \frac{a}{m} \right]_{\simeq} -_{\mathbb{Q}} \left[ \frac{b}{n} \right]_{\simeq} &= \left[ \frac{a}{m} \boxminus \frac{b}{n} \right]_{\simeq} \\ \left[ \frac{a}{m} \right]_{\simeq} /_{\mathbb{Q}} \left[ \frac{b}{n} \right]_{\simeq} &= \left[ \frac{a}{m} \boxdiv \frac{b}{n} \right]_{\simeq} \end{aligned}$$

This captures a maximalist implementation of rational numbers. A minimalist implementation is also possible by defining **reduced** fractions as you learned the idea is grade school. For example, the reduced fraction corresponding to  $\frac{+24}{-15}$  is  $\frac{+8}{-5}$ . This idea is certainly familiar to you, so we do not need to walk through the details.

*Part III*  
*Applications*



## Minimum and Maximum

THE ORDERING OF NATURAL NUMBERS gives rise to two other operations: min and max. The minimum of two natural numbers  $m$  and  $n$  is, of course, the smaller of the two. It makes sense to say this because  $\leq$  is linear. That is, either  $m \leq n$  — in which case  $m$  is the minimum — or  $n < m$  — in which case  $n$  is the minimum. We write  $\min(m, n)$  for this. The maximum is written as  $\max(m, n)$ .

Both can be calculated using the arithmetic we already developed.

---

ALGORITHM 16: Minimum and Maximum

---

For natural numbers  $m$  and  $n$ ,  $\min(m, n)$  and  $\max(m, n)$  are calculated by

$$\begin{aligned}\min(m, n) &= n \dot{-} (n \dot{-} m) \\ \max(m, n) &= m + (n \dot{-} m)\end{aligned}$$

---

A proof that these algorithms actually implement minimum and maximum is an exercise.

---



EXERCISES:

---

146. Calculate explicitly  $\min(3, 6)$ .
  147. Calculate explicitly  $\max(4, 3)$ .
- 

The important property of min and max is summarized by the following lemma.

---

### CHAPTER GOALS

---

In this chapter, we define  $\min(m, n)$  and  $\max(m, n)$  for natural numbers and extend the laws of arithmetic to include them.

---

They really are algorithms because  $+$  and  $\dot{-}$  are defined by algorithms.

---

**LEMMA 21:** Characterizing min and max
 

---

For natural numbers  $m$ ,  $n$  and  $p$ ,

$$\begin{aligned} p \leq \min(m, n) &\iff p \leq m \text{ and } p \leq n \\ \max(m, n) \leq p &\iff m \leq p \text{ and } n \leq p. \end{aligned}$$

*Proof.* If  $m \leq n$ , then  $n \dot{-} m$  is the “true” difference. So  $m = n \dot{-} (n \dot{-} m)$ . If  $m > n$ , then  $n \dot{-} m = 0$ . So  $n \dot{-} (n \dot{-} m) = n$ . Either way,  $n \dot{-} (n \dot{-} m)$  is the minimum of the two.

The rest of the prove is an exercise. □

---

We summarize the lemma by saying that  $\min(m, n)$  is the *greatest lower bound* of  $m$  and  $n$ . Likewise,  $\max(m, n)$  is the *least upper bound*.

Some simple facts about min and max derive directly from these characterizations plus some facts about addition.

In particular, together with addition, min and max make the natural numbers into something called a *distributive lattice ordered monoid*. Basically, this means that addition together with min and max cooperate in specific ways that augment the basic laws of arithmetic. The most useful laws having to do with min and max follow.

---

**BASIC LAWS OF min AND max**


---

For any natural numbers,  $m$ ,  $n$  and  $p$ :

Like the basic laws of arithmetic presented in Chapter 2, there laws are organized here to emphasize similarities between min and max. Pay attention to that.

**Characterization via  $\leq$** 

$m \leq \min(n, p)$  if and only if  $m \leq n$  and  $m \leq p$   
 $\max(m, n) \leq p$  if and only if  $m \leq p$  and  $n \leq p$

**Associativity**       $\min(m, \min(n, p)) = \min(\min(m, n), p)$   
 $\max(m, \max(n, p)) = \max(\max(m, n), p)$

**Commutativity**       $\min(m, n) = \min(n, m)$   
 $\max(m, n) = \max(n, m)$

**Idempotency**       $\min(m, m) = m$   
 $\max(m, m) = m$

**Absorptivity**       $m = \min(m, \max(n, m))$   
 $m = \max(m, \min(n, m))$

**Distributivity**       $m + \min(n, p) = \min(m + n, m + p)$   
 $m + \max(n, p) = \max(m + n, m + p)$   
 $\max(m, \min(n, p)) = \min(\max(m, n), \max(m, p))$   
 $\min(m, \max(n, p)) = \max(\min(m, n), \min(m, p))$   
 $m \cdot \min(n, p) = \min(m \cdot n, m \cdot p)$   
 $m \cdot \max(n, p) = \max(m \cdot n, m \cdot p)$

**Modularity**       $m + n = \min(m, n) + \max(m, n)$

We will not prove most of these as they follow easily from arithmetic laws. Nevertheless, a sampling follows.

LEMMA 22: Addition distributes over min

$m + \min(n, p) = \min(m + n, m + p)$  for all natural numbers  $m, n, p$ .

*Proof.* Because  $\min(n, p) \leq n$  and addition is monotonic,  $m + \min(n, p) \leq m + n$ . Likewise  $m + \min(n, p) \leq m + p$ . So  $m + \min(n, p) \leq \min(m + n, m + p)$ .

So to complete the proof, we must show that  $\min(m + n, m + p) \leq m + \min(n, p)$ . Suppose  $k \leq \min(m + n, m + p)$ . Then  $k \leq m + n$  and  $k \leq m + p$ . If  $k \leq m$  then  $k \leq m + \min(n, p)$  obviously. Otherwise,  $m < k$  by Linearity. So  $m + d^\frown = k$  for some  $d$ . Hence  $m + d^\frown \leq m + n$  and  $m + d^\frown \leq m + p$ . Since  $\leq$  is order reflecting,  $d^\frown \leq \min(n, p)$ . Consequently,  $k = m + d^\frown \leq m + \min(n, p)$ . We have thus shown that  $k \leq \min(m + n, m + p)$  implies  $k \leq m + \min(n, p)$ . In particular, this applies to  $\min(m + n, m + p)$ .  $\square$

This is a good illustration of why antisymmetry of  $\leq$  is useful. We wish to prove  $m + \min(n, p) = \min(m + n, m + p)$ . Instead, we prove  $m + \min(n, p) \leq \min(m + n, m + p)$ , and separately,  $\min(m + n, m + p) \leq m + \min(n, p)$ . Then we let antisymmetry take us the rest of the way.



---

**EXERCISES:**

---

148. Calculate the following values. Show work.

(a)  $\min(5, \min(4, 6))$

(b)  $\min(5, \max(4, 6))$

(c)  $\min(340, \max(234, 340))$

(d)  $\min(5, \max(3, \min(\max(1, 2), 7)))$

(e)  $\min(5 + \max(4 + \min(3 + \max(7, 8), 3 + \min(7, 8)), 6), 7)$

149. Prove that min distributes over max.

150. Prove that multiplication distributes over min.

---

## Greatest Common Divisors and Least Common Multiples

In Chapter 13, the minimum of two natural numbers — written  $\min(m, n)$  — is characterized as the greatest lower bound of  $m$  and  $n$  with respect to the standard order  $\leq$ . Likewise,  $\max(m, n)$  is the least upper bound. That is, for any  $m, n$  and  $p$ , they satisfy

$$\begin{aligned} p \leq \min(m, n) &\iff p \leq m \text{ and } p \leq n \\ \max(m, n) \leq p &\iff m \leq p \text{ and } n \leq p \end{aligned}$$

This chapter concerns analogous operations with respect to divisibility. The idea is simple. If any two natural numbers have a greatest lower bound and a least upper bound with respect to the “natural” ordering of  $\leq$ , then perhaps they also have a greatest lower bound and a least upper bound with respect to divisibility.

In fact, they do. The greatest lower bound with respect to divisibility is called the *greatest common divisor* and is denoted  $\gcd(m, n)$ . The least upper bound with respect to divisibility is called the *least common multiple* and is denoted by  $\text{lcm}(m, n)$ . These satisfy precisely the analogous conditions as  $\max$  and  $\min$ :

$$\begin{aligned} p \mid \gcd(m, n) &\iff p \mid m \text{ and } p \mid n \\ \text{lcm}(m, n) \mid p &\iff m \mid p \text{ and } n \mid p \end{aligned}$$

Though it may not be immediately obvious that greatest common divisor or least common multiple for any two natural numbers exists, we have a good inkling that they do, at least for the following informal reason.

When you simplify a fraction, say  $\frac{24}{18}$ , you remove any common factors. For example,  $\frac{24}{18} = \frac{6}{6} \cdot \frac{4}{3}$ . So the simplified fraction is  $\frac{4}{3}$ . You could have factored out  $\frac{2}{2}$  or  $\frac{3}{3}$ , but  $\frac{6}{6}$  is in some sense the best you can do. It divides 24 and 18, and is divisible by any other common divisor. In short 6 is the greatest common divisor of 24 and 18. You

As you know, greatest common divisors and least common multiples play useful roles in manipulating fractions. For example, to add  $\frac{1}{6} + \frac{2}{15}$ , we first find that 30 is least common multiple of 6 and 15. So  $\frac{1}{6} = \frac{5}{30}$  and  $\frac{2}{15} = \frac{4}{30}$ . Thus  $\frac{1}{6} + \frac{2}{15} = \frac{9}{30}$ . The result is not reduced to simplest terms. Continuing, we note that 3 is the greatest common divisor of 9 and 30. So  $\frac{9}{30} = \frac{3 \cdot 3}{3 \cdot 10} = \frac{3}{10}$ .

have been finding greatest common divisors ever since you learned to simplify fractions. So now our task is simply to put that on a solid footing.

A proof that every two natural numbers have a greatest common divisor is quite old. Euclid included one proof in his geometry text, written in approximately twenty-two hundred years ago.

The proof given here is based Euclid's idea, but the details are closer to a proof due to the 18th century mathematician Bézout. This proof gives us more information about the greatest common divisor that will be useful later.

Bézout's original theorem is proved in terms of integers, not natural numbers, using negative numbers and subtraction. Our proof is longer and more complicated, essentially because it replaces subtraction with steps that involve cancellativity of addition.

As we know, zero is a bit strange with respect to division and multiplication. So let's deal with when (and if) it is possible that  $\gcd(m, n) = 0$ . Looking at the characterization in terms of divisibility, suppose  $\gcd(m, n) = 0$ . Then in particular,  $0 \mid m$  and  $0 \mid n$ . But this can only happen if  $m$  and  $n$  are both 0. So,  $\gcd(m, n) = 0$  implies  $m = 0$  and  $n = 0$ . Conversely, consider whether  $\gcd(0, 0)$  even makes sense. Whatever it is, it must be a divisor of 0. But every natural number is trivially a divisor of 0. So, we are really looking for the *greatest* natural number. This is a subtle point, but in this context "greatest" refers to divisibility, not  $\leq$ . We know there is no *largest* natural number with respect to  $\leq$ , but  $n \mid 0$  is true for all natural numbers  $n$ . So (somewhat counter-intuitively), 0 is the greatest natural number with respect to divisibility. And so  $\gcd(0, 0)$  must be 0.

Before moving to the proof of the theorem, we establish a helpful lemma that is related to Euclid's original proof.

---

LEMMA 23:

---

*For any natural numbers  $m$ ,  $n$  and  $p$ , if  $0 < m$ , then  $p$  is a common divisor of  $m$  and  $n$  if and only if  $p$  is a common divisor of  $(n \% m)$  and  $m$ .*

*Proof.* This follows immediately from the fact that for any  $m$ ,  $n$  and  $p$ , if  $p \mid m$  then it is the case that  $p \mid n$  if and only if  $p \mid (m + n)$ .  $\square$

---

To prove the main result of this section, we introduce a new definition (only needed in this section) and prove a useful lemma about it.

An advantage of this approach is that it more clearly highlights the fact that gcd really has nothing to do with negative numbers. It also suggests how one might design other, faster, algorithms for computing gcd. We will not explore this point, but if you study, for example, cryptography, you will likely encounter some of these faster algorithms.

In fact, Euclid did not employ division and remainders. Instead he used repeated subtraction of  $m$  until his algorithm reached  $n \% m$ .

---

**DEFINITION 46:** Bézout Pairs
 

---

For a pair of natural numbers  $(m, n)$ , say that a pair of natural numbers  $(a, b)$  is a **Bézout pair for**  $(m, n)$  if and only if  $(b \cdot n) \div (a \cdot m)$  is the greatest common divisor of  $m$  and  $n$ .

---

Notice that Bézout pairs depend on order. If  $(a, b)$  is a Bézout pair for  $(m, n)$ , it is almost never the case that  $(a, b)$ , or even  $(b, a)$  is a Bézout pair for the swapped pair  $(n, m)$ . Nevertheless, we will need to swap pairs  $(m, n)$  and still be guaranteed that existence of a Bézout pair.

---

**LEMMA 24:** Condition when Bézout pairs can be swapped
 

---

If  $(m, n)$  has a Bézout pair and  $m > 0$ , then  $(n, m)$  also has a Bézout pair.

*Proof.* Suppose  $m > 0$  and  $(a, b)$  is a Bézout pair for  $(n, m)$ . So,

$$g = b \cdot n \div a \cdot m \quad (14.1)$$

is the greatest common divisor of  $m$  and  $n$ . Notice that  $g$  must be positive, for otherwise  $m = n = 0$ . So it is also the case that  $a \cdot m < b \cdot n$ . Hence  $n > 0$  and

$$g + a \cdot m = b \cdot n. \quad (14.2)$$

Our goal is to find natural numbers  $a'$  and  $b'$  so that  $g + a' \cdot n = b' \cdot m$ .

We consider the two cases:  $a = 0$  and  $a > 0$ .

Suppose  $a = 0$ . Then  $0 < g = b' \cdot n$ . So  $n$  divides  $g$  and  $g$  divides  $n$ . Hence  $g = n$ ,  $b = 1$  and  $n \mid m$ . Since  $m < 0$ ,  $m = c \cdot n$  for some positive  $c$ . So,

$$\begin{aligned} g + (c \div 1) \cdot n &= g + (c \cdot n \div n) && \text{— Algebra} \\ &= g + (m \div n) && \text{— } c \cdot n = m \\ &= (g + m) \div n && \text{— Algebra } m \geq n \\ &= (n + m) \div n && \text{— } g = n \\ &= 1 \cdot m && \text{— Algebra} \end{aligned}$$

Consequently,  $(1, c \div 1)$  is a Bézout pair for  $(m, n)$ .

Since  $g$  divides both  $m$  and  $n$ , let  $c = m // g$  and  $d = n // g$ . So Equation 14.2 becomes

$$g + a \cdot c \cdot g = b \cdot d \cdot g$$

Hence  $1 + a \cdot c = b \cdot d$ , and  $0 < ac$ . Now,

$$\begin{aligned} 1 + (a \cdot c \div 1) \cdot b \cdot d &= 1 + a \cdot c \cdot b \cdot d \div bd \\ &= a \cdot c \cdot b \cdot d \div (b \cdot d \div 1) \\ &= a \cdot c \cdot b \cdot d \div a \cdot c \\ &= a \cdot c \cdot (b \cdot d \div 1) \end{aligned}$$

Hence,

$$\begin{aligned} g + (a \cdot b \cdot c \div b) \cdot n &= g + (a \cdot c \cdot b \div b) \cdot d \cdot g \\ &= g \cdot (1 + (a \cdot c \div 1) \cdot b \cdot d) \\ &= (a \cdot b \cdot d \div a) \cdot m \end{aligned}$$

Consequently,  $a' = a \cdot b \cdot c \div b$  and  $b' = a \cdot b \cdot d \div a$  constitute a Bezout pair for  $(n, m)$ .  $\square$

#### THEOREM 11: Bézout's Theorem

*For any natural numbers  $m$  and  $n$  for which  $m \leq n$  and  $0 < n$ , there is a Bézout pair for  $(m, n)$ .*

*That is, there are natural numbers  $a$  and  $b$  satisfying*

$$\gcd(m, n) = a \cdot m \div b \cdot n.$$

*Proof.* By strong induction on  $n$ , we show that for every  $m$ , if  $m \leq n$  then there is a Bézout pair for  $(m, n)$ .

*Strong inductive hypothesis* Suppose that for some positive  $k$  it is the case that for all pairs  $(i, j) \in \mathbb{N} \times \mathbb{N}^+$  so that  $i \leq j < k$ , there is a Bezout pair for  $(i, j)$ .

*Inductive step:* The goal is to find a Bézout pair for any pair  $(h, k)$  where  $h \leq k$ .

Suppose  $h \leq k$ . We consider three cases:  $h = 0$ ,  $h = k$ , and  $0 < h < k$ .

*Case  $h = 0$ :* Evidently  $k$  is the greatest common divisor of  $0$  and  $k$ .

And clearly  $k = 1 \cdot k \div 0 \cdot 0$ . So  $(0, 1)$  is a Bézout pair for  $(0, k)$ .

*Case  $0 < h \leq k$ :* Then  $k \% h < h$ . So by the strong inductive hypothesis, there is a Bézout pair  $(a', b')$  for  $(k \% h, h)$ . That is,  $g = b' \cdot h \div a' \cdot (k \% h)$  is the greatest common divisor of  $k \% h$  and  $h$ , and

$$g + a' \cdot (k \% h) = b' \cdot h. \quad (14.3)$$



By Lemma 23,  $g$  is also the greatest common divisor of  $h$  and  $k$ . So it remains to find  $a$  and  $b$  satisfying

$$g + a \cdot h = b \cdot k.$$

From Equation 14.3, by adding equals to each side and using the fact that  $k = h \cdot (k // h) + k \% h$ ,

$$g + a' \cdot (h \cdot (k // h) + k \% h) = b' \cdot h + a' \cdot h \cdot (k // h) \quad (14.4)$$

$$g + a' \cdot k = (b' + a' \cdot (k // h)) \cdot h. \quad (14.5)$$

So  $(a', b' + a' \cdot (k // h))$  is a Bezout pair for  $(k, h)$ . But  $k$  and  $h$  are in the wrong order. Because  $k$  is positive, Lemma 24 yields the desired pair for  $(h, k)$ .

□

---

Bézout's Theorem establishes that the greatest common divisor exists for any two natural numbers  $m \leq n$ . But because  $\gcd(m, n) = \gcd(n, m)$ , the order does not matter for existence of  $\gcd(m, n)$ , but only with respect to determining Bezout pairs.

An algorithm for calculating Bézout pairs, and therefore  $\gcd(m, n)$ , can be extracted from the proof. We can also extract a simpler algorithm for calculating  $\gcd(m, n)$  without finding actual Bézout pairs. This is the basis of the algorithm that Euclid provided.

Actually, the last half of the proof involved converting a Bézout pair for  $(n, k)$  into a Bézout triple for  $(k, n)$ . So in principle, we could adapt the proof to show directly that the order of arguments does not matter in any case.

---

#### ALGORITHM 17: Euclid's algorithm

---

*For natural numbers  $m$  and  $n$ ,  $\gcd(m, n)$  can be computed via the following:*

$$\begin{aligned} \gcd(0, n) &= n \\ \gcd(m, n) &= \gcd(n \% m, m) && \text{for any } m > 0 \end{aligned}$$


---

The least common multiple of  $m$  and  $n$  also exists and relates to  $\gcd(m, n)$  in the following way.

---

**THEOREM 12:** Least common multiples exist
 

---

*Any two natural numbers  $m$  and  $n$  have a least common multiple. Moreover, if we write  $\ell$  for the least common multiple of  $m$  and  $n$ , then  $m \cdot n = \gcd(m, n) \cdot \ell$ .*

*Proof.* If  $\gcd(m, n) = 0$ , then  $m = 0$  and  $n = 0$ . So 0 is their only common multiple, and  $m \cdot n = \gcd(m, n) \cdot 0$ .

Suppose that  $\gcd(m, n)$  is positive and that  $m \leq n$  (or swap  $m$  and  $n$  if not). So  $\gcd(m, n) \cdot p = m$  and  $\gcd(m, n) \cdot q = n$  for some natural numbers  $p$  and  $q$ .

Let  $s = \gcd(m, n) \cdot p \cdot q$ . Clearly,  $s$  is a multiple of  $m$  and is a multiple of  $n$ , and  $\gcd(m, n) \cdot s = m \cdot n$ . So  $s$  is a common multiple of  $m$  and  $n$ . To prove the result, we must show that  $s$  is the least. That is, if  $r$  is also a common multiple of  $m$  and  $n$ , then  $r$  is a multiple of  $s$ .

Suppose  $m \cdot p' = r$  and  $n \cdot q' = r$ . Then  $r = \gcd(m, n) \cdot p \cdot p' = \gcd(m, n) \cdot q \cdot q'$ . So  $p \cdot p' = q \cdot q'$ . Let  $t = p \cdot p'$ .

By Bezout's Theorem, find natural numbers  $a$  and  $b$  so that

$$\gcd(m, n) + a \cdot m = b \cdot n.$$

So

$$t \cdot \gcd(m, n) + t \cdot a \cdot m = t \cdot b \cdot n. \quad (14.6)$$

But  $t \cdot a \cdot m = s \cdot a \cdot q'$ , and  $t \cdot b \cdot n = s \cdot b \cdot p'$ . So Equation 14.6 can be written as

$$r + s \cdot a \cdot q' = s \cdot b \cdot p' \quad (14.7)$$

Since  $s \cdot a \cdot q'$  and  $s \cdot b \cdot p'$  are both multiples of  $s$ ,  $r$  is also a multiple of  $s$ .  $\square$

---

We can take the result of this theorem as a definition.

---

**DEFINITION 47:** Least common multiple as an operation
 

---

*For any natural numbers  $m$  and  $n$ , let  $\text{lcm}(m, n)$  denote the least common multiple of  $m$  and  $n$ .*

---

Theorem 12 actually tells us more than mere existence of  $\text{lcm}(m, n)$ . In fact, it shows that  $\gcd(m, n) \cdot \text{lcm}(m, n) = m \cdot n$ . This is exactly analogous to one of the laws we established in Chapter 13:  $\min(m, n) + \max(m, n) = m + n$ . Though the proof of Theorem 12 is much more complicated, the two results are clearly similar. They

The proof of Theorem 12 can be used to extract an algorithm for computing  $\text{lcm}(m, n)$ . The simplest, but not most efficient, method is to take  $\text{lcm}(m, n) = (m \cdot n) // \gcd(m, n)$ .

are in fact closely related. Many of the laws we enumerated for  $\min$  and  $\max$  carry over to greatest common divisor and least common multiple. Compare the following table of laws to the table on page 208

---

#### BASIC LAWS OF $\gcd$ AND $\text{lcm}$

---

For any natural numbers  $m$ ,  $n$  and  $p$ :

**Characterization via  $|$**

$$m \mid \gcd(n, p) \text{ if and only if } m \mid n \text{ and } m \mid p$$

$$\text{lcm}(m, n) \mid p \text{ if and only if } m \mid p \text{ and } n \mid p$$

**Associativity**

$$\gcd(m, \gcd(n, p)) = \gcd(\gcd(m, n), p)$$

$$\text{lcm}(m, \text{lcm}(n, p)) = \text{lcm}(\text{lcm}(m, n), p)$$

**Commutativity**

$$\gcd(m, n) = \gcd(n, m)$$

$$\text{lcm}(m, n) = \text{lcm}(n, m)$$

**Idempotency**

$$\gcd(m, m) = m$$

$$\text{lcm}(m, m) = m$$

**Absorptivity**

$$m = \gcd(m, \text{lcm}(n, m))$$

$$m = \text{lcm}(m, \gcd(n, m))$$

**Distributivity**

$$m \cdot \gcd(n, p) = \gcd(m \cdot n, m \cdot p)$$

$$m \cdot \text{lcm}(n, p) = \text{lcm}(m \cdot n, m \cdot p)$$

$$\text{lcm}(m, \gcd(n, p)) = \gcd(\text{lcm}(m, n), \text{lcm}(m, p))$$

$$\gcd(m, \text{lcm}(n, p)) = \text{lcm}(\gcd(m, n), \gcd(m, p))$$

**Modularity**

$$m \cdot n = \gcd(m, n) \cdot \text{lcm}(m, n)$$


---

#### *Relative Primality*

One characterization of prime numbers is that  $p$  is prime if and only if  $p$  is greater than 1 and  $\gcd(p, n) = 1$  for all  $n$  lying strictly between 1 and  $p$ . For example, 5 is prime. It is easy to check that  $\gcd(5, 2) = \gcd(5, 3) = \gcd(5, 4) = 1$ . More generally, pairs of numbers for which  $\gcd(m, n) = 1$  play an important role in many parts of discrete mathematics. Hence the following definition.

---

**DEFINITION 48:** Relatively Prime Numbers
 

---

*Natural numbers  $m$  and  $n$  are said to be relatively prime if and only if  $\gcd(m, n) = 1$ .*

---

When you “reduce” a fraction, such as  $\frac{24}{15}$ , you find the greatest common divisor  $\gcd(24, 15)$  and factor it from the numerator and denominator. In this case, the result is  $\frac{3 \cdot 8}{3 \cdot 5} = \frac{8}{5}$ . The result has a numerator and denominator that are relatively prime. In fact, this is a useful definition of “reduced fraction”. I won’t pursue this idea here because fractions are not our concern for now. Nevertheless, a couple of useful lemmas regarding relative primality are in order. You can work out how these relate to fractions.

---

**LEMMA 25:** Factoring out a greatest common divisor results in relatively prime numbers.

---

*Suppose  $m$  and  $n$  are any two natural numbers. Choose  $p$  and  $q$  so that  $m = \gcd(m, n) \cdot p$  and  $n = \gcd(m, n) \cdot q$ . Then  $p$  and  $q$  are relatively prime.*

*Proof.* Exercise. □

---



---

**LEMMA 26:** Products of relative primes are relatively prime

---

*If  $\gcd(m, p) = 1$  and  $\gcd(n, p) = 1$ , then  $\gcd(m \cdot n, p) = 1$ .*

*Proof.* In case  $p = 0$ , this is trivial since  $\gcd(m, 0) = m$  and  $\gcd(n, 0) = n$ . Otherwise,  $\gcd(n, p) = \gcd(p \% n, n)$  and  $\gcd(m, p) = \gcd(p \% m, m)$ . Suppose  $\gcd(m \cdot n, p) > 1$ ,  $\gcd(m, p) = 1$  and  $\gcd(n, p) = 1$ . So  $\gcd(m \cdot n, p)$  divides  $p$ . Evidently, it can not divide  $m$  or  $n$ . So there is a non-zero remainder in both cases. That is,  $m = q \cdot \gcd(m \cdot n, p) + r$  and  $n = q' \cdot \gcd(m \cdot n, p) + r'$  for some  $q, r, q', r'$  where  $0 < r, r' < \gcd(m \cdot n, p)$ . □

---

---

LEMMA 27: A cancellation law for divisibility

---

If  $m$  and  $n$  are relatively prime and  $m \mid (n \cdot p)$ , then  $m \mid p$ .

*Proof.* Suppose  $\gcd(m, n) = 1$  and  $m$  divides  $n \cdot p$ . Either  $m \leq n$  or  $n < m$ . Suppose  $m \leq n$ , Bezout's Theorem yields natural numbers  $s$  and  $t$  so that  $1 + s \cdot m = t \cdot n$ . Notice that  $t$  can not be 0. So  $p + p \cdot s \cdot m = p \cdot t \cdot n$ . Since  $m$  divides the right side of this equation, it divide the left. Since  $m$  also divides  $p \cdot s \cdot m$ , it must divide  $p$ . Suppose  $n < m$ , then again Bezout's Theorem yields natural numbers  $s$  and  $t$  so that  $1 + s \cdot n = t \cdot m$ . In particular,  $t \neq 0$ . So  $m \mid t \cdot n \cdot p$  and  $t \cdot n \cdot p = p + s \cdot m \cdot p$ . Again  $m$  must also divide  $p$ .  $\square$

---




---

#### EXERCISES:

---

151. For each of the following pairs of numbers, calculate their greatest common divisor and indicate which pairs are relatively prime.
  152. Recall the Fibonacci numbers. Show that  $\text{fib}(n)$  and  $\text{fib}(n + 1)$  are always relatively prime.
  153. Show that for any natural numbers  $m$  and  $n$ ,  $m$  and  $m \cdot n + 1$  are always relatively prime.
- 

A natural next topic is primality. As you know, every positive natural number is composed of prime factors in essentially one way. For example, 30 is  $2 \cdot 3 \cdot 5$ , and there is no other decomposition of 30 except by re-ordering the factors. To make this general fact, known as the *Fundamental Theorem of Arithmetic*, precise, it will be helpful to have more mathematical equipment at our finger tips. Also, we now have some experience with proofs. It will help to study how proofs work.

## Prime Numbers

PRIME NUMBERS ARE BUILDING BLOCKS for the natural numbers. As such they are fundamental to many branches of discrete mathematics.

We all know that a prime number is a positive natural number that is not 1 and has no factors smaller than itself. One might ask why 1 is excluded. After all, 1 does not have any factors smaller than itself either. So at first blush, excluding 1 seems like an arbitrary decision. In fact, historically 1 has sometimes been regarded as a prime number. But that actually makes things more difficult. So now, everyone agrees that 1 should not be included. In the next paragraphs, we look at why this makes sense.

Recall that the product of a list of natural numbers is defined inductively by

- $\prod[] = 1$
- $\prod(n:L) = n \cdot \prod L$ .

We can use this to streamline the definition of primality.

---

DEFINITION 49: Prime Numbers

---

A **prime number** is a positive natural number  $p$  so that for any list  $L \in \text{List}[\mathbb{N}]$  of natural numbers, if  $p \mid \prod L$ , then  $p \mid n$  for some  $n \in L$ .

---

For example, 5 is prime. Suppose  $5 \mid \prod L$ . That means that  $5 \cdot d = \prod L$  for some  $d$ . But then at least one multiple of 5 appears somewhere on the list. And 6 is not prime because  $6 \mid \prod[2, 3]$ , but  $6 \nmid 2$  and  $6 \nmid 3$ . That is,  $[2, 3]$  is a list that has no multiples of 6, but has a multiple of 6 as its product.

With this definition, 1 is not prime for exactly the same reason that 6 is not. In both cases, all we need to show is the number divides

---

### CHAPTER GOALS

---

Define rigorously the prime numbers and prove the Fundamental Theorem of Arithmetic (each positive natural number has a unique prime factorization).

---

some product of natural numbers, but does not divide any of the factors of that product. For 1, just observe that 1 divides  $\prod L$ , but 1 does not divide any item on the list  $L$  because there are no items on the empty list.

Notice that for both of the counter-examples, 6 and 1, we were able to choose a product that does not just divide the number, but is equal to the number. This suggests that the definition of primality is related to a seemingly more special notion.

---

DEFINITION 50: Irreducible numbers

---

An **irreducible number** is a positive natural number  $p$  so that for any list  $L$  of natural numbers, if  $p = \prod L$ , then  $p \in L$ .

---

This definition is most likely what you were told is the definition of primality. That is, a prime number is usually defined to be any natural number greater than 1 that can not be factored into properly smaller numbers. It is not hard to see that this is precisely what irreducibility means.

The distinction between irreducibility and primality is an important one in other contexts. As it happens though, in the natural numbers primes and irreducibles are the same. So why bother to have two definitions?

First, in situations very closely related to the natural numbers, primes and irreducibles really are not the same. So it is helpful to be aware of the distinction now. Second, the proof that the two concepts are the same for natural numbers is instructive. To show that all primes are irreducible is fairly easy. The proof that all irreducibles are prime is more complicated.

---

LEMMA 28: All primes are irreducible

---

*Any prime number is also an irreducible number.*

*Proof.* Suppose  $p$  is prime. Consider a list  $L$  so that  $p = \prod L$ . Then  $p$  divides  $\prod L$ . So there is some  $n \in L$  so that  $p$  divides  $n$ . But  $n$  also divides  $\prod L$ . Because divisibility is anti-symmetric,  $p = n$ .  $\square$

---

To prove the converse, we need a fact that (in a different form) was known to Euclid.

---

**LEMMA 29:** Euclid's Lemma
 

---

*If  $p$  is irreducible, then for any  $m$  either  $m$  and  $p$  are relatively prime, or  $m$  is a multiple of  $p$ .*

*Proof.* Suppose  $p$  is irreducible. In particular, it is positive, so  $\gcd(p, m) \neq 0$ .

Suppose  $1 < \gcd(p, m) < p$ . Then for some  $q$ ,  $\gcd(p, m) \cdot q = p$ . But then  $1 < q < p$  as well. So  $p$  is not irreducible. This contradicts the assumption, so  $\gcd(p, m)$  must either equal 1 or be greater than or equal to  $p$ . But  $\gcd(p, m) \leq p$  always holds when  $p$  is positive.

□

---

**THEOREM 13:** Irreducibles are prime
 

---

*Every irreducible number is prime.*

*Proof.* Suppose  $p$  is irreducible. By induction on lists, we show that if  $p \mid \prod L$  holds for some list  $L$ , then  $p \mid n$  for some  $n$  on the list.

*Basis* The basis holds vacuously because  $\prod [] = 1 \neq p$ .

*Inductive Hypothesis* Suppose  $K$  is a list of natural numbers so that if  $p \mid \prod K$ , then  $p \mid n$  for some  $n$  on the list  $K$ .

*Inductive Step* The goal is to show that for any natural number  $m$ , if  $p \mid \prod (m : K)$ , then  $p \mid n$  for some  $n$  on the list  $m : K$ . That is, either  $p \mid m$  or  $p \mid n$  for some  $n$  on the list  $K$ .

By Euclid's Lemma, either  $m$  is a multiple of  $p$  or  $m$  and  $p$  are relatively prime. If it is the former, then  $p \mid m$ . If it is the latter, then  $p \mid \prod K$  by Lemma 27. So by the inductive hypothesis,  $p \mid n$  for some  $n$  on the list  $K$ .

□

---

Consequently, we can use the terms “irreducible” and “prime” interchangeably for natural numbers (also for integers when they come up). But bear in mind that the two concepts differ in other contexts.

Our next task is to prove two facts you have been told many times. First, the *Fundamental Theorem of Arithmetic* says that every positive natural number can be factored in essentially one way into prime numbers. Second, we show that there are infinitely many primes.



We split the proof of Fundamental Theorem into two parts. First, every positive natural number factors into irreducibles. Then any factorization into primes is unique except for the order in which the primes are listed.

---

LEMMA 30:

---

*For any positive natural number  $m$ , there is a list  $P$  consisting only of irreducible numbers so that  $m = \prod P$ .*

*Proof.* Here we use strong induction.

- [Strong Inductive Hypothesis] Assume that for some positive  $k$ , it is the case that for every  $0 < j < k$ , there is a list of irreducibles  $P$  so that  $j = \prod P$ .
- [Strong Inductive Step] There are three cases to consider. Either  $k = 1$ , or  $k = i \cdot j$  for some positive  $i$  and  $j$  strictly less than  $k$ , or neither of these holds.

Suppose  $k = 1$ . Then we let  $P = []$ . This is a (trivial) list of irreducibles whose product is  $k$ .

Suppose  $k = i \cdot j$  where  $i$  and  $j$  are both positive and strictly less than  $k$ . By the inductive hypothesis, there are lists of primes  $Q$  and  $R$  so that  $i = \prod Q$  and  $j = \prod R$ . Hence  $k = \prod Q \cdot \prod R = \prod (Q + R)$ .

Suppose  $k$  is neither equal to 1, nor equal to  $i \cdot j$  for any two natural numbers strictly less than  $k$ . Then  $k$  itself is irreducible. The proof of this is by induction on lists.

*Basis* Because  $k \neq \prod []$ , the basis is vacuously true. That is *if*  $k = \prod []$ , then  $k$  would appear on the list  $[]$ .

*Inductive hypothesis* Suppose that for a list  $K$ , if  $k = \prod K$ , then  $k$  appears on the list.

*Inductive step* The goal is to show that for any  $n$ , if  $k = \prod (n : K)$ , then  $k$  appears on the list  $n : K$ . That is, either  $k = n$  or  $k \in K$ .

By definition,  $\prod (n : K) = n \cdot \prod K$ . But the case of  $k$  under consideration is that  $k = i \cdot j$  implies either  $i \geq k$  or  $j \geq k$ .

Because  $k$  is positive, either  $k = n$  or  $k = \prod K$ . In the first case,  $k \in (n : K)$ . In the second case, by the inductive hypothesis for  $K$ ,  $k \in K$ , so  $k \in (n : K)$ .

So  $k$  is irreducible. And since  $k = \prod [k]$ , it is the product of a list of irreducibles.

These are the only possible cases. So the strong inductive step is proved. □

The list of irreducibles constructed in Lemma 30 is called a **prime factorization of  $m$**  (because we know irreducible and prime mean the same thing). Generally, a composite has more than one such factorization for trivial reasons. For example, the lists  $[2, 3]$  and  $[3, 2]$  both factor 6 into irreducibles. But the *only* way two such factorizations of the same number can differ is by the order in which the factors are listed. If we insist that our lists are sorted in increasing order, this ambiguity is avoided.

Recall from Chapter 5.5 that because  $\mathbb{N}$  is ordered by  $\leq$ , we can say what it means to have a sorted list of natural numbers. Namely, a list  $L$  of natural numbers is *sorted* if  $L_i \leq L_j$  whenever  $i \leq j < \text{len}(L)$ .

---

LEMMA 31: Sorted lists of primes are determined by their products

---

For any sorted lists of primes,  $P$  and  $Q$ , if  $\prod P = \prod Q$  then  $P = Q$ .

*Proof.* The proof is by list induction on  $P$ .

**Basis** For any list  $Q$  of irreducibles, if  $\prod[] = \prod Q$ , then  $Q$  must be the empty list as well. I leave the details of checking this to you.

**Inductive Hypothesis** Suppose that for some sorted list  $K$  consisting of primes it is the case that for every sorted list  $P$  of primes, if  $\prod K = \prod P$ , then  $K = P$ .

**Inductive Step** Consider some prime  $m$  so that  $m : K$  is sorted. The goal is to show that for any sorted list of primes  $Q$ , if  $\prod m : K = \prod Q$ , then  $m : K = Q$ . This is by list induction on  $Q$ .

**Basis** As before,  $\prod[] = 1$ . So the basis holds vacuously.

**Inductive Hypothesis** Suppose that for some sorted list  $J$ , it is the case that if  $\prod m : K = \prod J$ , then  $m : K = J$ .

**Inductive Step** Consider prime  $n$  so that  $n : J$  is sorted. If  $m = n$ , then  $\prod K = \prod J$ . So by the main inductive hypothesis,  $K = J$ . So  $m : K = n : J$ .

To complete the inductive step we show that  $m < n$  and  $n < m$  are both impossible. If  $m < n$ , then  $m$  is relatively prime to  $n$ . So  $m \mid \prod J$ , but because  $n : J$  is a sorted list of primes,  $m$  is also relatively prime to each item in  $J$ . This contradicts primality of  $m$ . Likewise,  $n < m$  contradicts primality of  $n$  for the same reason by swapping the roles of  $m$  and  $n$ , and of  $K$  and  $J$ .

So we have completed the inductive proof that for any sorted list of primes  $Q$ , if  $\prod m : K = \prod Q$ , then  $m : K = Q$ . This is the needed inductive step.

In other words, the factorization of a positive natural number into primes is unique up to the order in which we list the factors.

□

---

The preceding lemmas show that every positive  $m$  has a unique sorted prime factorization, typically called *the* prime factorization.

---



---

**THEOREM 14:** Fundamental Theorem of Arithmetic

---

*Every positive natural number has a unique sorted prime factorization.*

*Proof.* All that remains is to the remark that if  $P$  is a prime factorization of  $m$ , then  $P$  can be sorted into increasing order. The result has the same product  $P$  because of commutativity. □

---

For example, 24 is factored as  $[2, 2, 2, 3]$  and 800 is factored as  $[2, 2, 2, 2, 2, 5, 5]$ . We can get a more efficient representation by listing the number of times each prime is repeated. So we can represent 800 by  $[5, 0, 2]$ , signifying that  $800 = 2^5 3^0 5^2$ . Notice that in this notation, we need the middle 0 as a “place holder” to indicate that our number does not have any 3 factors. Also “trailing zeros” in this notation do not make a difference.  $[5, 0, 3, 0]$  also represents 800. The extra 0 at the end simply tells us that 800 is not divisible by the next prime (7). We will investigate this notation informally (without supplying proofs) after establishing that we have a plentiful supply of primes.

---



---

**THEOREM 15:** There are infinitely many primes

---

*For every natural number  $m$ , there is a prime  $p$  greater than  $m$ .*

*Proof.* We prove this by showing that no finite list of prime exhausts all of them.

Suppose  $L$  is a non-empty list consisting of primes. To show that  $L$  is missing something, consider the number  $m = 1 + \prod L$ . Since  $\prod L \geq 1$ ,  $m > 1$ . So  $m$  has a non-empty prime factorization, say  $M$ . Clearly  $M$  does not have any item in common with  $L$  (this could be proved explicitly by induction on  $L$ ). So any item on the list  $M$  is missing from  $L$ . Since  $M$  is not empty,  $L$  can not be an exhaustive list of all primes. □

---

---

**DEFINITION 51:**


---

We can enumerate the primes:  $2, 3, 5, 7, \dots$  in increasing order. For every natural number  $k$ , let  $p_k$  be the  $k^{\text{th}}$  prime. That is, the numbers  $p_k$  satisfy

$$p_0 = 2$$

$$p_{k+1} = \text{the smallest prime } q \text{ so that } p_k < q$$

Notice that this is well-defined because for any  $m$  there is a prime greater than  $m$ . We would not know this if we did not know there are infinitely many primes.

---




---

**EXERCISES:**


---

154. What is  $p_{10}$ ?
155. What is the prime factorization of 1440?
- 

Using  $p_k$ , we can succinctly represent any positive natural number by a list of exponents of primes. Namely, for a list  $R$  of natural numbers, define  $R_{\text{pe}}$  (for “prime exponent representation”) to be

$$R_{\text{pe}} := \prod_{i < \text{len} R} p_i^{R_i}.$$

For example,  $[1, 2, 0, 2]_{\text{pe}} = 2^1 3^2 5^0 7^2 = 882$ .

For a positive natural number  $n$ , let  $\text{PE}(n)$  denote the unique list so that (i)  $\text{PE}(n)_{\text{pe}} = n$  and (ii)  $\text{PE}(n)$  does not contain any trailing zeroes.

Define  $P \overline{+} Q$  for two lists of natural numbers by adding the items of the two lists itemwise. That is,

$$P \overline{+} [] = P$$

$$[] \overline{+} Q = Q$$

$$m : P \overline{+} n : Q = (m + n) : (P \overline{+} Q)$$

Then it is clear (we will not give a proof) that  $P_{\text{pe}} \cdot Q_{\text{pe}} = (P \overline{+} Q)_{\text{pe}}$ .

Define a relation  $P \preceq Q$  on lists of natural numbers by

- $[] \preceq Q$  always,
- $m : P \preceq []$  never, and
- $m : P \preceq n : Q$  if and only if  $m \leq n$  and  $P \preceq Q$ .

Then  $P_{pe} \mid Q_{pe}$  if and only if  $P \preceq Q$ . In other words, if we list the exponents of primes that constitute given numbers  $m$  and  $n$ , we can compare them for divisibility simply by comparing the exponents by  $\leq$ .

This leads to the relation between  $\min$  and  $\gcd$ . Namely, define an operation on lists of natural numbers by taking minima itemwise.

$$\begin{aligned}\overline{\min}(P, []) &= P \\ \overline{\min}([], Q) &= Q \\ \overline{\min}(m : P, n : Q) &= \min(m, n) : \overline{\min}(P, Q)\end{aligned}$$

Then it is also easy to check that  $\overline{\min}(P, Q)_{pe} = \gcd(P_{pe}, Q_{pe})$  for any two lists of natural numbers.

A similar definition of  $\overline{\max}$  will lead to  $\overline{\max}(P, Q)_{pe} = \text{lcm}(P_{pe}, Q_{pe})$  for any two lists of natural numbers as well.

Now the fact that  $m \cdot n = \gcd(m, n) \cdot \text{lcm}(m, n)$  becomes clearly a corollary of the fact that  $m + n = \min(m, n) + \max(m, n)$ . Namely, for any two lists of natural numbers,  $P$  and  $Q$ ,

$$P \overline{+} Q = \overline{\min}(P, Q) \overline{+} \overline{\max}(P, Q)$$

is easily proved by induction on lists. And the preceding paragraphs show that  $m \cdot n = \gcd(m, n) \cdot \text{lcm}(m, n)$  follows from this.

In principle,  $\gcd(m, n)$  could be calculated by first factoring  $m$  and  $n$  into primes, then using  $\overline{\min}$  on the resulting lists, then converting that back to a natural number. This method, however, is extremely inefficient because finding the prime factorization of a large number is difficult.

16

## *Counting*

*There'll be time enough for counting when the dealing's done.*

— Kenny Rogers

# A

## *Proofs*

*Trust in haste; regret in leisure.*

— appearing on a poster in the movie *Brazil*

Now that you have looked carefully at several proofs, and produced a few of your own, you might wonder what all the fuss is about.

### *The purpose of a proof*

Look again at the algorithm for addition and the assertion that  $0 + m = m$  for all natural numbers  $m$ . The assertion is not explicitly part of the algorithm. But you would be right to ask what the point of proving it is. We all know that  $0 + m = m$  already. So what is the benefit of writing down a proof?

### *Proof as a sanity check*

You can look at the Algorithm 1 as just spelling out some arbitrary instructions to be followed. The algorithm is purely mechanical. Though you may be convinced that it has a special interpretation (perhaps you think it describes actual addition), you certainly are not obliged to believe that. So how can you start to convince yourself at least that Algorithm 1 is doing something reasonable? How can you show it is not “buggy”?

One approach is to prove properties that you know the algorithm ought to have if it truly implements addition. For example, it had better be the case that  $0 + m = m$  for all  $m$ . Otherwise the algorithm would have an obvious bug. Likewise, if the algorithm really implements addition, it had better be true that  $m + n = n + m$  for any natural numbers  $m$  and  $n$ .

If you can convince yourself that Algorithm 1 produces results satisfying various laws (like identity and commutativity), you gain confidence that the algorithm is correct.

---

#### CHAPTER GOALS

---

This is an informal discussion of the general purposes and structure of proofs.

---

Engineers sometimes call a verification like this a **sanity check**. Having engineered an algorithm (like the one we claim implements addition), we then check that it behaves sanely. If it fails to be commutative, or associative, or it fails some other obvious test, we know the design is wrong.

Proofs often serve this sort of “debugging” function in mathematics. Though mathematicians do not typically use the word “debug”, they use proofs to make sure things are not going off the rails.

### *Proof as a discovery*

On another view, you could take Algorithm 1 to be *the* official definition of addition: to add two numbers, follow the algorithm. If  $m + n$  is intended to mean “start from  $m$  and count forward  $n$ ” then the algorithm seems to be a pretty faithful implementation of the idea. If you take that attitude, the algorithm is not just some arbitrary mechanical procedure. It *is* what we mean by addition.

On this view, a proof that addition is commutative is more like a **discovery** that the definition of addition forces it to have a useful property. So the proof conveys new (but in this example, pretty obvious) knowledge. Later in the text, you will see many proofs which are meant to convey something new. They are discoveries. For obvious reasons, mathematicians tend to pay closer attention to discoveries than to “mere” debugging checks. But both sorts of proofs are important.

The line between “proof as check” and “proof as discovery” is not firm. It is mainly a matter of perspective. In the text, we try to distinguish between what we think of as a sanity check and a new idea by the labels PROPOSITION and THEOREM. For us, a proposition is an assertion that needs a routine check, so it plays the role of sanity check. A theorem is an assertion of some new (one hopes, interesting) idea.

But take these labels with a grain of salt. There really can not be hard and fast rule for choosing between them because what seems like a routine fact to one person may be a surprise to someone else.

To add to the confusion, two other labels are commonly used. A LEMMA is an assertion that is mainly useful for proving other things. Think of a lemma as a kind of mathematical subroutine. Lemma 1 is an example. It says that  $m + n^{\wedge} = m^{\wedge} + n$ . That is not a particularly interesting fact on its own. But we used it in the proof of Proposition 3, and will reuse it later.

Finally, a COROLLARY is typically an assertion that follows almost directly from something that was just proved. The last chapter does not include anything we would want to call a theorem or a corollary.



We will encounter some of these in the next chapter.

Here is a summary of terminology we use for different kinds of assertions.

*Proposition* An assertion that should be checked, but is routine. In this text, propositions usually represent sanity checks.

*Lemma* An assertion that is not necessarily interesting on its own, but is useful for proving other things. In Chapter 2, Lemmas 1 and 2 are examples. A lemma is like a subroutine that helps organize things. Some lemmas turn out to be important because they can be re-used. In fact, there are quite a few lemmas in mathematics that become well known because they have a wide range of applications.

*Theorem* An assertion that provides a new idea. The new idea can be in the statement of the theorem itself, or in the proof.

*Corollary* An assertion, the proof of which follows easily from a theorem or proposition.

### *Structure of Proofs*

A proof typically follows the outline of nearly any expository essay you have ever written:

- I. Say what you plan on saying. [State a thesis.]
- II. Say it. [Develop the thesis.]
- III. Summarize what you have just said. [Recapitulate.]

There are some twists though. First, proofs tend to use a sort of hybrid language that involves English (or whatever language you are writing in) and mathematical notation that takes some getting used to. Second, the development part (II) frequently involves subclaims and proofs. So proofs are nested. Third, the development typically follows one of a handful of patterns (called “tactics” in this text) that are closely related to the form of the thesis. So for example, if the thesis is to prove “If  $n$  is even, then  $n^2$  is even,” then the development is likely to follow the form of the Direct Implication (Tactic 4). So it will read “Suppose  $n$  is even. Then there is . . .”

In English, sentences can be classified by their function into four basic kinds:

*Declarative* A sentence that makes a statement such as “My hovercraft is full of eels”, or “4 is a prime number”. A declarative does not need to be true, but it is the sort of sentence that can be judged to be true or false.

*Interrogative* A question such as “Do you prefer cheddar or emmentaler?”, or “Is cosine a continuous function?”.

*Imperative* A request or a command such as “Do not panic” or “Suppose  $n$  is a natural number.”

*Exclamation* An expression of surprise or other extreme emotion such as “My, what big numbers you have!”.

A mathematical proof consists mainly of declarative and imperative sentences. So we will concentrate on them. Do interrogatives and exclamations play a role in proofs? Not on your life!

Parts I and III are declarative. Usually the main thesis (I) is stated as the assertion to be proved. For example, in the proof of Proposition 3 (addition is commutative) we state the thesis as follows:

For all natural numbers  $m$  and  $n$ ,

$$m + n = n + m.$$

This is a grammatically correct, declarative sentence provided you understand that mathematical notation like “ $m + n = n + m$ ” can be read aloud like English.

The summary statement (Part III) is

Because the argument does not depend on any assumptions about  $n$ , it is valid for all  $n$ .

This example is a bit subtle because “it” refers to the previous argument that concludes: “ $m + n = n + m$  for all  $m$ .” So the summary statement really asserts that “ $m + n = n + m$  for all  $m$ ” is true for all  $n$ . This is a paraphrase of the original thesis, so we try not to bore the reader with stating it again verbatim.

The development of the thesis (Part II) comprises the main content of the proof just as in most good essays. The development, in turn, may consist of subproofs. The block on the next duplicates the entire proof of Proposition 3, marked up to show its structure. The proof consists of “nested” Thesis/Development/Summation structures.

Two imperative sentences appear in the proof: “Suppose  $n$  is a fixed natural number,” and “Assume that  $k + n = n + k$  for some  $k$ .” They are neither true nor false, because they are actually providing instructions to the reader. This is similar to your mom saying “eat your vegetables”. You, (in the case of a proof) the reader, are being told what to do. Imperatives play a critical role in a proof that we will dive into soon.

I. The goal is to prove that

$$m + n = n + m$$

for all natural numbers  $m$  and  $n$ .

II. Suppose  $n$  is a fixed natural number.

A. The goal is to show that for any natural number  $m$ ,  $m + n = n + m$ .

B. The proof is by induction on  $m$ .

**Basis** (The basis requires a subproof).

1. The goal is to show that  $0 + n = n + 0$ .
2. But  $0 + n = n = n + 0$  holds because of Proposition 2 and the definition of  $+$ .
3. (So  $0 + n = n + 0$ .)

**Inductive Hypothesis** Assume that  $k + n = n + k$  for some  $k$ .

**Inductive Step**

1. The goal is to show that  $k^{\wedge} + n = n + k^{\wedge}$ .
- 2.

$$\begin{aligned} k^{\wedge} + n &= k + n^{\wedge} && \text{--- Lemma 1} \\ &= (k + n)^{\wedge} && \text{--- Definition of } + \\ &= (n + k)^{\wedge} && \text{--- Inductive Hypothesis} \\ &= n + k^{\wedge} && \text{--- Definition of } + \end{aligned}$$

3. (So  $k^{\wedge} + n = n + k^{\wedge}$ .)

C. Therefore,  $m + n = n + m$  for all  $m$ .

III. Because the argument does not depend on any assumptions about  $n$ , it is valid for all  $n$ .

The proposition is the main goal.

By the tactic of Universal Generalization, change the goal

By the tactic of Simple Induction, introduce a goal for the basis, an inductive hypothesis, and a goal for the inductive step.

Use calculation to prove the basis

The summary (III) is omitted in the original proof because it is the result of the preceding calculation.

The inductive step also requires a subproof.

Use calculation

Summary omitted.

Summarize the conclusion of the induction.

Summarize the conclusion of the universal generalization.

---

The role of proof tactics is to guide the structure of an argument based on the goal and on what is available at some point in a proof. For example, according to the induction tactic, when the goal (the thesis) is an assertion for the form “for all natural numbers  $m, \dots$ ”, the proof can take the form of a simple arithmetic induction, consisting of a basis, inductive hypothesis, and inductive step.

### Context of a proof

As a practical matter, whether or not an argument is convincing depends partly on the background knowledge of the reader. A good

proof matches the background of the intended reader to the ideas presented in the proof. More emphatically, a proof's correctness can actually depend on that background. After all, there is no point in trying to convince a friend that "distraction" is associative if the friend does not know what distraction means.

There would also not be much point to give you a proof right now about distributive lattices because you don't know what those things are. On the other hand, distributive lattices are a pretty standard, important idea in the authors' research community (and by the end of the semester, you will know what they are). This suggests that proof writing, like all kinds of expository writing, should be guided by the principle

*Know your audience.*

A proof by induction may be convincing to you now, but that is because you have thought about how induction works. For someone without that background, an inductive proof would look like gibberish.

Some background knowledge is purely logical, and mostly summarized in proof tactics. Other background is mathematical (such as what you know about calculus, if we were trying to prove something new about, say, integration).

Informally, we can think of the mathematical background as a "context" in which a proof can be understood either to be valid or not. It is helpful to break that context into different kinds of information.

Part of our mathematical context consists of a kind of grammatical knowledge. For example, you know that the symbol " $\pi$ " stands for a concrete positive real number. So automatically, it makes sense that " $\frac{\pi}{2}$ " and  $\sqrt{\pi}$  also stand for positive real numbers. You do not need to know anything else about the actual number  $\pi$ , such as that it is a bit more than 3, or that it is defined as the length of the circumference of a circle with diameter of length one, to know that  $\pi$  can be halved, or that its square root makes sense.

In the first chapter, we agreed that " $0$ " stands for a natural number, and that " $\smile$ " is an operation taking a natural number to a natural number. Based on this basic vocabulary alone, not knowing anything about what  $0$  or  $\smile$  actually mean, you know that " $0\smile\smile\smile$ " also stands for a natural number. This does not mean you necessarily know that  $m\smile$  is intended to be the successor of  $m$ . This is no different than if you know that "civit" is some sort of animal and that "august" an adjective, then "an august civit" makes grammatical sense as a noun phrase.

Similarly, " $+$ " and " $\cdot$ " denote binary operations on natural num-

The Mock Turtle went on. "We had the best of educations . . . Reeling and Writhing, of course, to begin with, and then the different branches of Arithmetic — Ambition, Distraction, Uglification and Derision." — from *Alice's Adventures in Wonderland* by Lewis Carroll.

bers: if  $m$  and  $n$  are natural numbers, then so are  $m + n$  and  $m \cdot n$ . You do not need to know how addition or multiplication work to know that  $5 + 7$  and  $5 \cdot 7$  denote natural numbers. So the sentence “ $5 \cdot 7$  is a prime number” is intelligible (but false) because “is a prime number” is the sort of verb phrase that makes sense for natural numbers. In contrast, “ $5 \cdot 7$  smells of elderberries” is not even false. It is just nuts.

A completely formal development of a mathematical grammar isn’t needed for this text. But you should be aware that mathematical expressions ought to be intelligible as grammatically correct expressions. And correctness depends on how we have agreed things can be combined. A good test is that you ought to be able to read aloud any mathematical expression you write. For example, “ $5 + 4 = 9$ ” reads aloud as “five plus four equals nine.” But “ $4 \ 5 + = 9$ ” is grammatically wrong. Obviously, a really complicated expression might not be so *easy* to read aloud, but it ought to be pronounceable in principle.

Statements such as Postulates 1, 2 and 3, that assert something about how  $0$  and  $\cap$  interact, may also be part of a proof’s context. These are assertional, instead of grammatical, context. Moreover, once a proposition is proved, it becomes available as another part of the context. For example, now that you have understood the proof that addition is cancellative, you are free to use that fact any time it makes sense.

So context is dynamic. It changes depending on what you and your reader know.

Definitions such as Algorithms 1 and 2 supply both grammatical and assertional context. For example, that  $m + n$  is a natural number whenever  $m$  and  $n$  are natural numbers is a simple grammatical fact. The algorithm also asserts that  $m + 0 = m$  and  $m + k^\cap = (m + k)^\cap$  for all natural numbers  $m$  and  $k$ . These say something about how  $0$ ,  $\cap$ , and  $+$  interact.

Remember that, for our purpose, an algorithm is a certain kind of definition. It defines an operation by specifying how it is calculated.

## *Proof tactics*

Proof tactics are templates for how to justify a conclusion in a context. Many tactics are triggered exactly by the form of the conclusion. For example, when the goal is to prove that all natural numbers satisfy a certain property, then one option is to use induction. Other tactics are triggered by the form of some assertion in the context. For example, *modus ponens* is a tactic for proving an assertion  $T$  provided that “ $S$  implies  $T$ ” is part of the context of the proof. The proof proceeds by proving  $S$ , and then concluding  $T$ . It depends on “ $S$  implies  $T$ ” being part of the context.

There are tactics to reach conclusions of a particular form, and tac-

tics to use assertions of a particular form to reach other conclusions. Tactics 4 and 5 illustrate the distinction. The first is a tactic to prove an assertion of the form “if  $S$  then  $T$ ,” the second is to use such an assertion of that form to conclude something else.

To further illustrate this distinction between producing and using assertions of some form, the following two tactics are extremely simple, but haven’t been needed yet.

---

#### PROOF TACTIC 15: Conjunction Introduction

---

When the goal is to prove an assert of the form “ $S$  and  $T$ ”, do the following.

1. Prove  $S$ .
2. Separately, prove  $T$ .
3. Conclude “ $S$  and  $T$ ”.

The conclusion is justified by the plain meaning of the word “and”.

---



---

#### PROOF TACTIC 16: Conjunction Elimination

---

When the goal is to prove  $U$ , and an assertion “ $S$  and  $T$ ” is part of the context,

1. add  $S$  to the context,
2. add  $T$  to the context,
3. prove  $U$  using the new enlarged context.

This is also justified by the plain meaning of the word “and.” If “ $S$  and  $T$ ” is something you know, then you know  $S$  and you know  $T$ . So anything that can be proved ( $U$ ) using  $S$  and  $T$  as separate assumptions follows from the single assumption “ $S$  and  $T$ ”.

---

#### *Imperative sentences in a proof*

Remember that an *imperative* sentence is the grammatical kind of sentence that conveys a command or request: “Please pass the salt”, “Take your seats”, “Let us bow our heads”, “Do do that voodoo that you do so well”.

You have probably noticed in proofs so far that certain imperative sentences show up. “Suppose  $n$  is a natural number” is a typical example. It asks the reader to do something. But what? It asks the reader to change the grammatical context of the proof so that the symbol “ $n$ ” now stands for a natural number (the specific value of which is undetermined). Think of it as temporarily adding an entry to a dictionary stating “ $n$  is the name for some natural number”. The change of context lasts until the current subproof is done and  $n$  is no longer needed.

Likewise, in an inductive proof, the inductive hypothesis typically reads something like “Assume that  $k + n = n + k$  for some natural number  $k$ ”. This requests the reader to change contexts again so that  $k$  also names a natural number, and (whatever particular values  $n$  and  $k$  happen to take) they satisfy  $n + k = k + n$ . This change of context lasts until the end of the subproof. In this case, that is until the end of the inductive step.

The main imperative verbs that signal change of context in a proof are *suppose*, *assume* and *let*. For variety, sometimes people will write “Consider a prime number  $p$ ”. This is a paraphrase of “Suppose  $p$  is a prime number”.

The following advice is not uniformly part of mathematics. But we need all the help we can get, so we tend to use *suppose*, *assume* and *let* for distinct purposes.

- “Suppose” introduces a name for an arbitrary value of some kind. For example, “Suppose  $i$  is an integer” means that all we know is that  $i$  is now the name of an integer. This is a purely grammatical point: After the supposition,  $i$  can be used wherever an integer makes sense.
- “Assume” introduces a new assertion. For example, “Assume that  $k^2 > 5$ ” changes the context to include a new assertion.
- “Let” introduces a name for a particular value. For example, suppose  $k$  is a natural number, then “Let  $j = k^2$ ” means that  $j$  is now a name for the square of  $k$ .
- The formulation “Suppose  $k$  is a natural number. Assume that  $0 + k = k$ ” introduces a new identifier  $k$  and then asserts that  $k$  has a particular property. This is so common that we frequently shorten it to say “Assume that  $0 + k = k$  for some natural number  $k$ ”. In other words, this use of “assume” actually involves a supposition and an assumption.