Chapter 1

Writing Proofs

Proof serves many purposes simultaneously. In being exposed to the scrutiny and judgment of a new audience, the proof is subject to a constant process of criticism and revalidation. Errors, ambiguities, and misunderstandings are cleared up by constant exposure. Proof is respectability. Proof is the seal of authority.

Proof, in its best instances, increases understanding by revealing the heart of the matter. Proof suggests new mathematics. The novice who studies proofs gets closer to the creation of new mathematics. Proof is mathematical power, the electric voltage of the subject which vitalizes the static assertions of the theorems.

Finally, proof is ritual, and a celebration of the power of pure reason. Such an exercise in reassurance may be necessary in view of all the messes that clear thinking clearly gets us into.

Philip J. Davis and Reuben Hersh

One of the goals for this course is to improve your ability both to discover and to express (in oral or written form) proofs of mathematical assertions. This section presents some guidelines that will be useful in each of these tasks, especially in preparing written proofs. These notes on writing proofs were originally produced by M. Stob. This version has been revised somewhat by R. Pruim

1.1 Proofs in Mathematics

I have made such wonderful discoveries that I am myself lost in astonishment; out of nothing I have created a new and another world.

John Bolyai

Mathematicians prove their assertions. This distinguishes mathematics from all other sciences and, indeed, all other intellectual pursuits. Indeed,
one definition of mathematics is that it is “proving statements about abstract objects.” You probably first met this conception of mathematics in your secondary school geometry course. While Euclid wasn’t the first person to prove mathematical propositions, his treatment of geometry was the first systematization of a large body of mathematics and has served as a model of mathematical thought for 2200 years. To understand the role of proof in mathematics, we can do no better than to start with Euclid. The basic ingredients of Euclidean geometry are three.

**Primitives terms.** Point, line, plane are all primitive terms in modern treatments of Euclidean geometry. We usually call these undefined terms and do not attempt to give definitions for them. (Euclid did give definitions for all his terms; we discuss the role of definition in mathematics later.) Of course a logical system must start somewhere; it is impossible to define all terms without falling prey to either circularity or infinite regress. David Hilbert, in his *Foundations of Geometry*, simply starts as follows

> Let us consider three distinct systems of things. The things composing the first system, we will call *points* and designate them by the letters $A$, $B$, $C$, …; those of the second, we will call *straight lines*, and designate them by the letters $a$, $b$, $c$, …; and those of the third system, we will call *planes* and designate them by the Greek letters $\alpha$, $\beta$, $\gamma$, ….

Hilbert says nothing about what the “things” are.

**Axioms.** An axiom is a proposition about the objects in question which we do not attempt to prove but rather which we accept as given. One of Euclid’s axioms, for example, was “It shall be possible to draw a straight line joining any two points.” Aristotle describes the role of axioms.

> It is not everything that can be proved, otherwise the chain of proof would be endless. You must begin somewhere, and you start with things admitted but undemonstrable. These are first principles common to all sciences which are called axioms or common opinions.

Euclid and Aristotle thought of axioms as propositions which were “obviously” true. But it is not necessary to think of them as either true or false. Rather, they are propositions which we agree, perhaps for the sake of argument, to accept as given.

**Rules of inference.** Axioms and previously proved theorems are combined to prove new theorems using the laws of logic. The nature of such rules of inference is best illustrated by an example.
All men are mortal
\[ \text{Socrates is a man} \]
\[ \text{Socrates is mortal} \]

In this example, the axioms (called premises or hypotheses) are written above the line and the theorem (called the conclusion) is written below the line. The whole argument is called a deduction. This particular argument is an example of a rule of inference which is now usually called Universal Instantiation. Two important features of this argument characterize a rule of inference. First, the relationship of the conclusion to the hypotheses is such that we cannot fail to accept the truth of the conclusion if we accept the truth of the hypotheses. Of course we do not have to accept the truth of the hypotheses and so are not compelled to believe the conclusion. But a rule of inference must necessarily produce true conclusions from true hypotheses. Second, this relationship does not depend on the concepts mentioned (humanity, mortality, Socratiety) but only on the form of the propositions. Hilbert said it this way: “One must be able at any time to relace ‘points, lines, and planes’ with ‘tables, chairs, and beer mugs.’” The next nonsensical argument is as valid as the previous one.

All beer mugs are chairs
\[ \text{Socrates is a beer mug} \]
\[ \text{Socrates is a chair} \]

While neither hypothesis is true and certainly the conclusion is false (perhaps your favorite chair is named Socrates but the Socrates I am thinking of was not a chair), this argument too is a perfectly acceptable example of universal instantiation.

This then is how mathematics is created. Starting from axioms which are propositions about certain objects which may be undefined, the rules of inference are used to prove new theorems. Anyone accepting the truth of the axioms, must accept the truth of all the theorems.

Exercises

1. Morris Kline says that

   Mathematics is a body of knowledge. But it contains no truths.

   In a similar vein, Hilbert said

   It is not truth but only certainty that is at stake.

   What do they mean?
1.2 Written Proofs

*I like being a writer; it’s the paperwork I dislike.*

PETER DE VRIES

This chapter is specifically concerned with written proofs. While a proof might be thought of as an abstract object existing only in our minds, the fact is that mathematics advances only in so far as proofs are communicated. And writing remains the principal means of such communication. So to be a mathematician, you need to learn how to prove things but also to write those proofs clearly and correctly. Learning to write proofs also makes reading other people’s proofs easier.

What does a written proof look like? Like any other piece of prose, a written proof is organized into sentences and paragraphs. Like all other correct writing, those sentences are written following all the standard conventions of the language in which they are written. Because of the precision of the thoughts that a proof must convey, it is especially important that the prose be clear and correct. In order to help the reader follow the argument being presented, it is helpful to use a style that emphasizes the organization and flow of the argument.

It is useful to think of a proof as consisting of three sorts of sentences. First, some sentences express the result of a deduction. That is they result from other sentences by applying one of the rules of inference as described in the last section. These sentences can be recognized by the fact that they start with words like “therefore,” “thus,” “so,” or “from this it follows that.” Of course such a sentence will not normally be the first sentence of the proof and such a sentence normally depends on one or more earlier sentences of the proof.

A second sort of sentence is used to state a given fact. These sentences normally are restatements of hypotheses of the theorem being proved or recall facts that the reader should know already. An example of the latter type is the sentence “Recall that \( \sqrt{2} \) is an irrational number.” Again, these sentences should contain some sort of cue that they are stating hypotheses or previously proved facts rather than asserting something new.

Finally, a third sort of sentence appearing in a proof is what might be called “glue” or the prose that holds the other sentences together. These sentences are used for purposes such as to inform the reader of the structure of the argument (“Next we will show that \( G \) is abelian.”) or to establish notation (“Let \( X = Y \oplus Z \).”) This glue may also include longer passages that outline an entire argument in advance, summarize it at the conclusion, provide motivation for the methods being employed, or describe an example. It is a common error of beginning proof-writers to use too few of these sentences so that the structure of the argument is not clear.

This description of what a proof should look like is at odds with some of the ideas of proof that students take with them from high-school math-
emematics and perhaps even calculus. Some students think that a proof, like the solution to any homework problem, should be a series of equations with few, if any, words. Now an equation is a perfectly acceptable sentence. The equation “$x + 2 = 3$” is a sentence and could, perhaps, appear in a proof. Mathematical notation serves as abbreviation for words and phrases which would be tedious to write many times. However this equation almost certainly should not appear as a sentence of a proof, for it is not clear whether it expresses a hypothesis or a conclusion. Probably what is meant is “Suppose that $x$ is a real number such that $x + 2 = 3$.” or “Therefore, it follows from the last equation that $x + 2 = 3$.” Equations must be tied together with words.

A second misconception of proof is a side-effect of studying geometry in high school. High school geometry courses often teach that a proof should be arranged in two columns; in one column we write assertions and in the other we write the reason that the corresponding assertion is true. Look in a mathematics journal and you will see that no mathematician writes proofs in this manner. The main reason for teaching “two-column” proofs is that they sometimes help beginning proof writers concentrate on the logical structure of the argument without having to attend to the details of grammatical writing. A two-column proof reinforces the notion that each assertion in a proof should serve a definite purpose and must be justified. For realistic theorems however (unlike the baby theorems of high school geometry), the two-column proof becomes cumbersome and hard to read. The two-column proof, among other defects, is missing the “glue” described above. These informative sentences are crucial to helping a reader understand a complicated proof. We want the reader to understand not only the justification of each step but also the logical structure of the argument as a whole. Nevertheless, a two-column proof is sometimes a good place to start on scratch paper.

1.3 Proverbs

Here I have written out for you sayings
full of knowledge and wise advice

Proverbs 22:20

Proof-writing is, to some extent at least, a creative activity. There is no step-by-step recipe that, if followed carefully, is guaranteed to produce a proof. Proof-writing is different in this way than differentiation in calculus. Machines can differentiate as well as you can, but have not made much progress in producing proofs. However, experienced proof-writers do have certain principles by which their search for a proof is at least implicitly guided. In these notes we try to summarize at least some of these principles in the form of proverbs. As a vehicle for introducing the first few of these proverbs, we consider the proof of the following (easy) theorem.
Theorem 1.1 The square of an even natural number is even.

How does one get started writing a proof? The following proverb suggests the answer.

Proverb 1.2 The form of the theorem suggests the outline of the proof.

In fact

Proverb 1.3 You can write the first line of a proof, even if you don’t understand the theorem.

In a later section, we make a more detailed study of theorem forms. However Theorem 1.1 has a common and easily recognizable form. Here is an abstract version of that form.

Theorem 1.4 If object $x$ has property $P$ then $x$ has property $Q$.

or more simply

Theorem 1.5 If $x$ is a $P$ then $x$ is a $Q$.

It will become clear to you in your study of abstract mathematics that many theorems have exactly this form, though you might have to rewrite the theorem a bit to see this clearly. Here are some examples from all sorts of different mathematical specialities.

Theorem 1.6 Every differentiable function is continuous.

Theorem 1.7 Every semistable elliptic curve is modular.

Theorem 1.8 The lattice of computably enumerable sets is distributive.

Theorem 1.9 If $I$ is a prime ideal of a ring $R$, then $R/I$ is an integral domain.

The first of these theorems is one you know and love from your calculus class. The second is the famous theorem of Wiles proved in 1993 (it implies Fermat’s last theorem). The third comes from a field of mathematical logic known as recursion theory. The last theorem is one that you would meet in an abstract algebra class. Notice that Theorem 1.9 is really about two objects and both properties $P$ and $Q$ express relationships between these two objects. Note too that while Theorem 1.8 appears to be about only one object (the lattice of recursively enumerable sets), the other three theorems are about an unspecified number of objects. In fact, our Theorem 1.1 is about infinitely many objects.

So if Proverb 1.2 is correct, the outline of the proof Theorem 1.4 should suggest itself. In fact, by Proverb 1.3, we should be at least able to write
the first sentence of the proof. Indeed, here is a reasonable way to begin theorems such as Theorem 1.4.

Proof (of Theorem 1.4). Suppose \( x \) has property \( P \). We must show that \( x \) has property \( Q \).

The second sentence might be a bit pedantic, but the first clearly sets the stage. Thus, our proof of Theorem 1.1 begins as follows.

Proof (of Theorem 1.1). Suppose that \( n \) is an even natural number . . .

Actually, we can also predict the last line of theorems with this form. We would expect the proof of Theorem 1.1 to look like this.

Proof (of Theorem 1.1). Suppose that \( n \) is an even natural number. . . . Thus \( n^2 \) is even.

So what remains in the proof of the theorem is to fill in the . . . . We need to reason from the fact that \( x \) has property \( P \) to the fact that \( x \) has property \( Q \). In our particular example, and in many similar cases, the next proverb supplies the key.

Proverb 1.10 Use the definition.

In some cases it will be obvious that an \( x \) with property \( P \) also has property \( Q \). However, if it is that obvious, you probably will not be asked to supply a proof in the first place. So the first step is usually to determine what it means to have property \( P \). All mathematical terms, except the primitive ones, have precise definitions. In our case, the following definition is the important one.

Definition 1.11 A natural number \( n \) is even if there is a natural number \( k \) such that \( n = 2k \).

Using the definition gives us the second line of our proof. It also gives us a hint as to what the next to last line of the proof might be.

Proof (of Theorem 1.1). Suppose that \( n \) is an even natural number. Then there is a natural number \( k \) such that \( n = 2k \). . . . Thus there is a natural number \( l \) such that \( n^2 = 2l \). Thus \( n^2 \) is even.

The only work in this theorem is filling in the remaining gap. And this work is simply a computation. Thus the complete proof is

Proof (of Theorem 1.1). Suppose that \( n \) is an even natural number. Then there is a natural number \( k \) such that \( n = 2k \). So \( n^2 = (2k)^2 = 4k^2 = 2(2k^2) \). Thus there is a natural number \( l \) (namely \( 2k^2 \)) such that \( n^2 = 2l \). Thus \( n^2 \) is even.

Notice that the end of each of these proofs above is marked by the symbol \( \square \). This or some other symbol is often used in books and articles to mark the ends of proofs. Especially in older writing, the letters QED (which stand
for a latin phrase meaning ‘that which was to be proven has been shown’) are often used. Even when using such a marker, the last sentence or two of a proof should also indicate somehow that the conclusion of the argument has been reached, but the additional marker is also helpful if, for example, the reader wants to skip over the proof on a first reading or get an estimate on its length.

Here is one more (easy) theorem of this type will serve to reinforce the point of this section.

**Theorem 1.12** The base angles of an isosceles triangle are congruent.

Recasting this theorem in the form of Theorem 1.4, we might write this theorem as

**Theorem 1.13** If a triangle is isosceles, then the base angles of that triangle are congruent.

A diagram, Figure 1.3, serves to illustrate the statement of the theorem.

![Diagram of an isosceles triangle](image)

**Proof.** Suppose that \( \triangle ABC \) is isosceles. That is, suppose that \( AB \) and \( BC \) are the same length. (Note the clever use of the definition of isosceles triangle.) We must show that the base angles \( \alpha \) and \( \gamma \) are equal. To do that notice that triangles \( \triangle ABC \) and \( \triangle CBA \) are congruent by side-angle-side. Furthermore, \( \alpha \) and \( \gamma \) are corresponding angles of these two triangles. Thus \( \gamma \) and \( \alpha \) are congruent (equal) since corresponding parts of congruent triangles are congruent. \( \square \)

**Exercises**

1. Write good first and last lines to the proofs of
   (a) Theorem 1.6
   (b) Theorem 1.7
   (c) Theorem 1.8
   (d) Theorem 1.9.

2. (a) Write a good definition of “odd natural number.”
(b) Prove that the square of an odd number is odd.

3. Prove the following theorem

**Theorem 1.14** If a triangle has two congruent angles, then the triangle is isosceles.

(This theorem is the converse of Theorem 1.12. It is always difficult to know what you can assume in geometry, but the proof of this Theorem is similar to that of Theorem 1.12 — it only uses simple facts about congruent triangles.)

### 1.4 Contraposition and Contradiction

“There is no use trying,” said Alice; “one can’t believe impossible things.”

“I dare say you haven’t had much practice,” said the Queen, “When I was your age, I always did it for half an hour a day. Why, sometimes I’ve believed as many as six impossible things before breakfast.”

---

**Lewis Carroll**

In the last section, we saw how to produce a reasonable outline of a proof of a theorem with the following form.

**Theorem 1.15** For all \( x \), if \( x \) has property \( P \) then \( x \) has property \( Q \).

However,

**Proverb 1.16** There is more than one way to prove a theorem.

In this section, we look at two alternative outlines of the proof of Theorem 1.15. The first of these is called “proof by contraposition.” It depends on the following lemma.

**Lemma 1.17** Suppose that \( \alpha \) and \( \beta \) are propositions. Then the statement

\[
\text{If } \alpha \text{ then } \beta
\]

is logically equivalent to the statement

\[
\text{If } \beta \text{ is not true then } \alpha \text{ is not true.}
\]
The statement 1.2 is called the contrapositive of the statement 1.1. Here logically equivalent means that 1.1 is true just in case 1.2 is true. So to prove a statement of form 1.1, it is sufficient to prove 1.2. This fact gives us the first of two alternate outlines for a proof of Theorem 1.15.

Proof (of Theorem 1.15). We prove the theorem by contraposition. Suppose that \( x \) does not have property \( Q \). \( \ldots \) Then \( x \) does not have property \( P \). \( \square \)

As a specific example of a proof with this outline, consider the following theorem.

**Theorem 1.18** Suppose that \( n \) is a natural number such that \( n^2 \) is odd. Then \( n \) is odd.

*Proof.* We prove the theorem by contraposition. Suppose that \( n \) is not odd. That is suppose that \( n \) is even. Then \( n^2 \) is even (by Theorem 1.1). Thus \( n^2 \) is not odd. \( \square \)

How does one decide whether to prove a theorem directly or by contraposition? Often, the form of the properties \( P \) and \( Q \) give us a clue. If \( Q \) is a negative sort of property, (such as \( x \) is not divisible by 3), it may very well be easier to work with the hypothesis that \( x \) does not have property \( Q \). Of course we could always try to prove the theorem both ways and see which one works.

Another alternate way to prove a Theorem is to give a proof by contradiction. In a proof by contradiction, we suppose that the theorem is false and derive some sort of false statement from that assumption. If mathematics is consistent and if our reasoning is sound, this means that our assumption that the theorem is false is in error. So the outline of a proof by contradiction of Theorem 1.15 is as follows.

Proof (of of Thm 1.15). We prove the theorem by contradiction. So suppose that \( x \) has property \( P \) but that \( x \) does not have property \( Q \). \( \ldots \) Then \( 1=2 \) (or any other obviously false statement). Thus our assumption is in error and it must be the case that if \( x \) has property \( P \), \( x \) also has property \( Q \). \( \square \)

The most famous theorem which is usually proved by contradiction is the following.

**Theorem 1.19** \( \sqrt{2} \) is irrational.

We first rephrase the theorem so that it has the form of Theorem 1.15.

**Theorem 1.20** If \( x \) is any real number such that \( x^2 = 2 \), then \( x \) is not rational.
This form is logically superior to that of Theorem 1.19 since it doesn’t assume the existence or uniqueness of a number $\sqrt{2}$. This form also suggests either contraposition or contradiction since the property “is not rational” is not as easy to work with as the property “is rational.”

**Proof.** We proof Theorem 1.20 by contradiction. So suppose that $x$ is a number such that $x^2 = 2$ and $x$ is rational. Then there are natural numbers $p$ and $q$ such that $p$ and $q$ have no common divisors and $x = \frac{p}{q}$. Then

$$2 = x^2 = \left(\frac{p}{q}\right)^2$$

Therefore $p^2 = 2q^2$. This implies that $p^2$ is even and so $p$ is even (See exercise 1 below.) Since $p$ is even, $p = 2k$ for some natural number $k$. Thus $2q^2 = p^2 = 4k^2$ or $q^2 = 2k^2$. But this implies that $q^2$ is even and so that $q$ is even. Thus $p$ and $q$ are both even but this cannot be true since we assumed that $p$ and $q$ had no common divisors. □

Students sometimes confuse proofs by contradiction and contraposition. The proofs look similar in the beginning since each begins by assuming that $x$ does not have property $Q$. But a proof by contradiction also assume that $x$ has property $P$ and proves the negation of some true proposition while a proof by contraposition does not make this assumption and simply proves that $x$ does not have property $P$.

**Exercises**

1. Prove that if $p$ is a natural number such that $p^2$ is even then $p$ is even.

2. Prove that $\sqrt{3}$ is irrational.

3. Prove that $\sqrt{6}$ is irrational.

4. Write the contrapositive of the following theorems.
   (a) Theorem 1.6
   (b) Theorem 1.7
   (c) Theorem 1.8
   (d) Theorem 1.9.

5. Write the assumption made in a proof by contradiction of the following theorems.
   (a) Theorem 1.6
   (b) Theorem 1.7
   (c) Theorem 1.8
   (d) Theorem 1.9.
1.5 Ten Rules

When I read some of the rules for speaking and writing the English language correctly . . . I think any fool can make a rule and every fool will mind it.

HENRY THOREAU

As in most writing, there are certain mistakes that occur over and over again in mathematical writing. As well, there are certain conventions that mathematicians adhere to. In this section, we present ten rules to follow in writing proofs. If you follow these rules, you will go a long way towards making your writing clear and correct. Most of these rules come in one form or another from the wonderful book, Mathematical Writing, by Donald Knuth.

**Rule 1** Use the present tense, first person plural, active voice.

Bad: It will now be shown that . . .  
Good: We show . . .

**Rule 2** Choose the right technical term.

For example, not all formulas are equations.

**Rule 3** Don’t start a sentence with a symbol.

Bad: $x^n - a$ has $n$ distinct roots.  
Good: The polynomial $x^n - a$ has $n$ distinct roots.

**Rule 4** Respect the equal sign.

Bad: $x^2 = 4 = |x| = 2.$  
Good: If $x^2 = 4$, then $|x| = 2.$

**Rule 5** Normally, if “if”, then “then”.

Bad: If $x$ is positive, $x > 0$.  
Good: If $x$ is positive, then $x > 0$.

**Rule 6** Don’t omit “that”.

Bad: Assume $x$ is positive.  
Good: Assume that $x$ is positive.
Rule 7 Identify the type of variables.

Bad: For all $x, y$, $|x + y| \leq |x| + |y|$.
Good: For all real numbers $x, y$, we have $|x + y| \leq |x| + |y|$.

There is an exception to this rule. In many situations, the notation used implies the type of a variable. For example, it is usually understood that $n$ denotes a natural number. In Real Analysis, $x$ almost always denotes a real number. In these cases, the type can be suppressed in the interest of saving ink and avoiding the distraction of extra words.

Rule 8 Use “that” and “which” correctly.

Bad: The least integer which is greater than $\sqrt{27}$ . . .
Good: The least integer that is greater than $\sqrt{27}$ . . .

The distinction is this: ‘that’ introduces a modifying clause that adds information necessary to specify the object modified; ‘which’ introduces a modifying clause that adds additional information about an already specified object.

Rule 9 A variable used as an appositive need not be set off by commas.

Bad: Consider the group, $G$, that . . .
Good: Consider the group $G$ that . . .

Rule 10 Don’t use symbols such as $\exists, \forall, \lor, >$ in text; replace them by words. Symbols may, of course, be used in formulas.

Bad: Let $S$ be the set of numbers $< 1$.
Good: Let $S$ be the set of numbers less than 1.

1.6 Forms of Theorems

*And if you go in, should you turn left or right . . .
or right-and-three-quarters? Or maybe not quite?
*Or go around back and sneak in from behind?
*Simple it’s not, I’m afraid you will find,
*for a mind-maker-upper to make up his mind.*
Dr. Seuss

Not all theorems have the form of Theorem 1.4. In this section we look at a few of the more common forms.

1.7 Biconditional Theorems

A biconditional theorem is one with form

**Theorem 1.21** \( P \) is true if and only if \( Q \) is true.

The proof of Theorem 1.21, as you would suspect, has two parts.

**Proof (of Theorem 1.21).** First, suppose that \( P \) is true. ... Then \( Q \) is true.

Next, suppose that \( Q \) is true. ... Then \( P \) is true. \( \square \)

Often, one of the two directions of a biconditional is best proved by contraposition.

1.8 Existence Theorems

An existence theorem is one with the form

**Theorem 1.22** There is an \( x \) with property \( P \).

Here is a specific example.

**Theorem 1.23** There is a real number \( x \) such that \( x^2 + 6x - 17 = 0 \).

The most common form of proof of a theorem with the form of Theorem 1.22 is

**Proof.** Let \( x \) be defined (constructed, given) by ... Then \( x \) has property \( P \) because ... \( \square \)

For example, here is a complete proof of Theorem 1.23.

**Proof.** Let \( x \) be given by

\[
x = -3 + \sqrt{26}.
\]

Then \( x^2 + 6x - 17 = (-3 + \sqrt{26})^2 + 6(-3 + \sqrt{26}) - 17 = 9 - 6\sqrt{26} + 26 - 18 + 6\sqrt{26} - 17 = 0. \)

It is sometimes possible to prove Theorem 1.22 by contradiction. Such a proof would look like

**Proof (of Theorem 1.22).** We prove the theorem by contradiction. So suppose that no such \( x \) exists. Thus no \( x \) has property \( P \). ... Then 0=1. Therefore our assumption that no such \( x \) exists must be in error so indeed such an \( x \) does exist. \( \square \)

Such a proof by contradiction is rather peculiar since we now know that \( x \) exists but the proof does not give us any way to find \( x \).
1.9 Uniqueness

A uniqueness theorem is one with form

**Theorem 1.24** There is a unique $x$ with property $P$.

This theorem actually says two things; there is an $x$ with property $P$ but there is only one $x$ with that property. Thus, the proof has two parts; an existence part (and we have already discussed that) and a uniqueness part. The proof of Theorem 1.24 therefore looks like this.

**Proof.** **Existence.** Define $x$ by ... Then $x$ has property $P$ because ...

**Uniqueness.** Suppose that $x$ and $y$ have property $P$. ... Then $x = y$.

Often, the uniqueness portion of Theorem 1.24 is proved by contradiction. Then the second part of the proof looks like

**Proof.** **Uniqueness.** We prove uniqueness by contradiction. So suppose $x$ and $y$ have property $P$ and $x \neq y$. ... Then $0=1$. Therefore our assumption that $x \neq y$ must be in error.

The next theorem is a typical (though simple) example of a uniqueness theorem. Recall that an additive identity is a number 0 with the property that $0 + x = x + 0 = x$ for all real numbers $x$.

**Theorem 1.25** The additive identity for the set of real numbers is unique.

**Proof.** Suppose that 0 and 0’ are numbers satisfying the defining condition for being an additive identity, namely that

$$0 + x = x + 0 = x \quad \text{for all } x \quad (1.3)$$

and

$$0’ + x = x + 0’ = x \quad \text{for all } x \quad (1.4)$$

Then by 1.3 (with $x = 0’$) we have that $0 + 0’ = 0’$ and by 1.4 (with $x = 0$) we have that $0 + 0’ = 0$. Thus $0 = 0’$.

1.10 Universal Statements

A universal statement is one of form

**Theorem 1.26** For all $x$, $x$ has property $R$.

In fact, our original theorem form, Theorem 1.4, can best be understood as a universal statement.

**Theorem 1.27** For all $x$, if $x$ has property $P$ then $x$ has property $Q$. 
Consider again our proof of Theorem 1.4. We write it somewhat differently.

**Proof.** Let \( x \) be an arbitrary object with property \( P \). Then \( \ldots \) Then \( x \) has property \( Q \). \( \square \)

The key feature of this proof that allows us to claim that the Theorem is true of all objects, is that we did not assume any special properties of \( x \) (other than \( P \)). That is, though we named \( x \), \( x \) was otherwise a “generic” object. You will recall this sort of reasoning from geometry. If you were asked to prove something for all triangles, you probably drew a triangle and labeled the vertices \( A, B, \) and \( C \), but realized that you were not allowed to assume anything special about the triangle (say, from the diagram). Your proof had to work for equilateral triangles as well as triangles that were not even isosceles. So the general proof of a universal statement like Theorem 1.26 goes as follows.

**Proof.** Let \( x \) be given. \( \ldots \) Then \( x \) has property \( R \). \( \square \)

Here, the \( \ldots \) is filled in by an argument which assumes nothing in particular about \( x \). As a simple concrete example, consider the following Theorem.

**Theorem 1.28** For all real numbers \( x \), \( x^2 + x + 1 \geq 0 \).

**Proof.** Let \( x \) be an arbitrary real number.

\[
x^2 + x + 1 = (x^2 + x + \frac{1}{4}) + \frac{3}{4} = (x + \frac{1}{2})^2 + \frac{3}{4}
\]

But \((x + 1/2)^2 \geq 0\) so that \((x + 1/2)^2 + 3/4 > 0\). Thus \(x^2 + x + 1 > 0\). \( \square \)

In this proof, we used no special property of \( x \) and so we were entitled to conclude that the property in question held for all \( x \).

To prove a theorem like Theorem 1.26 by contradiction, we have to realize that to deny that every \( x \) has property \( R \) is to assert that some \( x \) does not have property \( R \). So a proof by contradiction of Theorem 1.26 might look like this.

**Proof.** We prove the theorem by contradiction. So suppose that there is an \( x \) which does not have property \( R \). \( \ldots \) Then \( 0=1 \). Therefore, no such \( x \) exists, i.e., all \( x \) have property \( R \). \( \square \)

The middle part of this proof would presumably be filled in by an argument showing that \( x \) has some impossible property.
Chapter 2

Cardinality

In this chapter we want to investigate the sizes of sets, especially infinite sets. We will discover that infinite sets can come in different sizes and make an important distinction between the smallest infinite sets (called countably infinite sets) and larger infinite sets (called uncountable sets).

In Section 2.2 we will make precise what we mean by the size of a set and present some of the most important basic results on cardinality (the technical name for the size of a set). In section 2.3, we will illustrate Cantor’s zig-zag and diagonalization techniques, two important techniques for establishing the cardinality of a set. But first, we turn our attention to some motivating examples from everyday life.

2.1 Counting, Children, and Chairs

We will motivate the definitions of cardinality by considering some simpler situations that involve counting of everyday objects, like blocks and people and chairs. If you have ever seen a young child attempt to count, you know that this is something that must be learned, and that young children must pass through several phases as they learn to count. In each phase, some new skill must be learned or some error corrected.

The first step is to memorize a list of words: one, two, three, four . . . \(^1\) Even once the list can be repeated, in order and without error, the child cannot really count anything until she understands how to associate these words with the objects being counted.

And so the child is taught to say these words while pointing to objects or pictures in a book. This phase of associating the number-words with objects develops slowly, and at first the child is likely to make one or both of two errors that lead to a miscount. The first error is to skip some of the objects and leads to an undercount. The second error is to say more than

\(^1\) Later, for counting larger sets, it will be necessary to grasp the pattern by which new names for numbers are generated, but this generally comes after a child has learned to count with the memorized list of numbers to, say, twenty.
one number while pointing to the same object and leads to an overcount. So a typical young child presented with 7 objects might obtain a “count” of ten by randomly pointing at objects and saying numbers until they reach ten (a natural stopping point). Some objects will have been pointed at more than once, of course. Perhaps others were never pointed at.

Counting has only been mastered when the child understands that each object must be associated with exactly one number from the list (no skips, no double counts). The last number spoken is then called the size of the collection of objects.\(^2\)

So we see that counting is the association of elements in two sets (in our example above a set of number-words and a set of objects) such that each element of one set is paired with exactly one element of the other set. We could visualize counting three blocks as follows:

\[
\begin{align*}
1 & \leftrightarrow \text{A} \\
2 & \leftrightarrow \text{F} \\
3 & \leftrightarrow \text{D}
\end{align*}
\]

Now suppose you are having a dinner party. At some point shortly before dinner you begin to fear that you do not have the correct number of chairs at the table. There are two ways to check. One is to count the chairs and the people and see if the numbers match. But there is another (albeit potentially more embarrassing) method. Simply invite everyone to the table. If every person has a chair and every chair has one person sitting on it, then the number of chairs and people is the same, even though we don’t know just what that number is without some additional work. On the other hand, if there is an unoccupied chair, or if there is a person left standing . . .

The point here is that even though we do not know names for “infinite numbers” (although they do exist) and do not have time to count elements in infinite sets (or very large finite sets) by pointing to them and reciting a memorized list of words, we can still compare two sets to see if they are the same size by pairing up elements in one set with elements in the other. Provided we do so while avoiding the errors of skipping elements or double-matching elements, we will know our two sets have the same size. We will formalize this in more mathematical language in the next section.

\(^2\)A good indication that all this is in place is when a child says something like “one, two, three; three blocks”. The repetition of “three” indicating that the size of the collection of blocks has been associated with the last number used to count them. Thus “three” is actually being used to in two related but somewhat different ways – no wonder it takes some practice to learn to count.
2.2 Basic Cardinality Results

Mathematically, the “pairing up” of elements between two sets is done by means of a function $f$ from $A$ to $B$ (written $f : A \to B$), which we can think of informally as an assignment of an element in $B$ to each element of $A$.

We want our function to avoid the errors of skipping or double-counting, so we introduce the following definitions:

**Definition 2.1 (one-to-one)** A function $f : A \to B$ is one-to-one if for any $a_1, a_2 \in A$, if $a_1 \neq a_2$, then $f(a_1) \neq f(a_2)$.

**Definition 2.2 (onto)** A function $f : A \to B$ is onto if for any $b \in B$, there is some $a \in A$ such that $f(a) = b$.

Notice that a one-to-one function is a function that does not double-count and an onto function is one that does not skip. One-to-one functions are sometimes called injective functions or injections. Onto functions are sometimes called surjective functions or surjections. A function that is both one-to-one and onto is called a bijection.

With all this background, it is pretty clear what it means to say that two sets have the same size:

**Definition 2.3 (Same-sized sets)** Two sets $A$ and $B$ have the same cardinality (written $|A| = |B|$) if there is a function $f : A \to B$ such that

- $f$ is one-to-one, and
- $f$ is onto.

Note that although the notation suggests that we have implicitly defined $|A|$ and $|B|$ (the sizes of the sets $A$ and $B$), we have not really done so. This can be done by specifying the list of infinite numbers (called cardinals) to be used once we finish with the the natural numbers, but it is not necessary for our purposes.\(^3\)

**Example 2.4** Let $E$ be the non-negative even integers ($E = \{0, 2, 4, 6, 8, \ldots \}$). Then $|N| = |E|$, since $f : n \mapsto 2n$ is one-to-one and onto.

Notice that the example above shows that infinite sets behave a bit differently from finite sets. The natural numbers include all of the evens, and much more besides; nevertheless, these two sets have the same size. In fact, this can be taken as the definition of an infinite set:

---

\(^3\)The study of cardinal numbers and cardinal arithmetic is part of a branch of logic known as set theory. It turns out that the properties of cardinals are closely related to properties of sets and in fact depend to some extent on the axioms one chooses to use for set theory.
Definition 2.5 (Finite, infinite) A set $A$ is infinite if it has a proper subset $B \subseteq A$ such that $|A| = |B|$. Otherwise, $A$ is finite.

If when we pair up the elements of $A$ with elements of $B$ we use up all of $A$ but perhaps have skipped over some elements of $B$, then $A$ cannot be larger than $B$:

Definition 2.6 (No bigger than) A set $A$ is no bigger than the set $B$ (written $|A| \leq |B|$), if there is a function $f : A \rightarrow B$, such that

- $f$ is one-to-one.

The notation chosen above suggests that “same size as” and “no bigger than” behave in nice ways. The next two theorems demonstrate that this is the case:

Theorem 2.7 (Properties of =) “Same size as” is an equivalence relation. That is,

1. “Same size” is reflexive: $|A| = |A|$.
2. “Same size” is symmetric: If $|A| = |B|$, then $|B| = |A|$.
3. “Same size” is transitive: If $|A| = |B|$ and $|B| = |C|$, then $|A| = |C|$.

Theorem 2.8 (Properties of ≤)

1. “No bigger than” is reflexive: $|A| \leq |A|$.
2. “No bigger than” is transitive: If $|A| \leq |B|$ and $|B| \leq |C|$, then $|A| \leq |C|$.
3. If $A \subseteq B$, then $|A| \leq |B|$.
4. $|B| \leq |A|$ if and only if there is a function $f : A \rightarrow B$ that is onto.
5. If $|A| \leq |B|$ and $|B| \leq |A|$, then $|A| = |B|$.

Thus the use of $=$ and $\leq$ is justified in our notation. This also suggests some extensions to the notation:

- $|B| \geq |A|$ means $|A| \leq |B|$,
- $|A| \neq |B|$ means it is not the case that $|A| = |B|$,
- $|A| < |B|$ means $|A| \leq |B|$ but $|A| \neq |B|$.

Definition 2.9 (Countable) If $|A| = |\mathbb{N}|$, then we say that $A$ is countably infinite. A countable set is any set that is either finite or countably infinite. In other words, a countable set is the same size as some subset of $\mathbb{N}$.
Most of the important infinite sets we will encounter in this class will be countably infinite. The following theorem establishes some nice properties of countable sets.

**Theorem 2.10 (Properties of countable sets)**

1. *The following sets are all countable: \( \mathbb{N} \), \( \mathbb{Z} \) (the integers), \( \mathbb{Q} \) (the rationals), the set of even integers.*

2. *The following sets are uncountable: \( \mathbb{R} \) (the real numbers), \([0, 1]\) (the reals in the interval from 0 to 1).*

3. *A finite union of countable sets is countable:*
   
   If \( A \) and \( B \) are countable, then \( A \cup B \) is also countable.

4. *The cross product of countable sets is countable:*
   
   Let \( A \) and \( B \) be countable, then \( A \times B \) is countable.

5. *A countable union of countable sets is countable:*
   
   Suppose that for each natural number \( n \), \( A_n \) is countable. Then \( A = \bigcup_{n=0}^{\infty} A_n \) is also countable.

6. *The set of all finite sequences from a countable set is countable:*
   
   Let \( A^{\leq n} \) denote the set of all sequences of \( n \) items from \( A \). (For example, \((1, 4, 3, 0) \in \mathbb{N}^{\leq 4}\).) Let \( A^* = \bigcup_{n=0}^{\infty} A^{\leq n} \). Then \( A^{\leq n} \) and \( A^* \) are countable.

7. *Every infinite set has a countably infinite subset.*

Finally, Cantor’s diagonalization argument can be used to establish the following general fact:

**Theorem 2.11 (Power set)** Let \( \mathcal{P}(A) \) denote the power set of \( A \) (i.e., the set of all subsets of \( A \)). Then \( |A| < |\mathcal{P}(A)| \).

Among other things, this shows that there is no largest size of set.

You are ased to prove 3 of Theorem 2.10 in the exercises. The proofs the rest of the this theorem and of Theorem 2.11 will have to wait until we have seen Cantor’s two powerful techniques of zig-zag and diagonalization.

**Exercises**

1. Prove Theorem 2.7.

2. Prove Theorem 2.8.

3. Prove part 3 of Theorem 2.10.

4. Show that \( \mathbb{Z} \) is countable.

5. Assuming that \( \mathbb{R} \) is uncountable and \( \mathbb{Q} \) is countable, determine whether the set of irrationals is countable or uncountable. Justify your claim.
2.3 Cantor’s Two Ideas

Zig-Zag

Several of the countability results of the previous section can be proven using a zig-zag argument that goes back to Cantor. We will use this method here to prove part 4 of Theorem 2.10.

Proof (of Theorem 2.10, part 4). Let $A$ and $B$ be countably infinite. We must show that $A \times B$ is countably infinite. (Strictly speaking, we need to deal with the cases where one or both of the sets are finite, too, but we will only do the case where both are infinite here.)

Since $A$ and $B$ are countably infinite, there are one-to-one, onto functions $f : \mathbb{N} \to A$ and $g : \mathbb{N} \to B$. So $A \times B = \{(f(i), g(j))|i, j \in \mathbb{N}\}$. The key idea of Cantor is to arrange the elements of $A \times B$ in a rectangular grid filling one quadrant of the plane:

As the picture indicates, we can enumerate $A \times B$ beginning in the upper left-hand corner and following the arrows. In fact, by slightly modifying the zig-zag argument (travel each diagonal from top to bottom instead of zig-zagging), it is not too hard to give an exact formula for a one-to-one, onto function mapping $A \times B$ to $\mathbb{N}$ (or vice versa).

It is worthwhile to give another proof of this.
Proof (of Theorem 2.10, part 4). This time we will make use of part 5 of Theorem 2.8. For this we need to exhibit one-to-one functions $\alpha : A \times B \to \mathbb{N}$ and $\beta : \mathbb{N} \to A \times B$. The following two functions can easily be shown to be one-to-one (for the first we use the fact that prime factorizations are unique):

$$
\alpha : (f(i), g(j)) \mapsto 2^i3^j \\
\beta : n \mapsto (f(n), g(0))
$$

Diagonalization

Cantor’s diagonalization idea is even cleverer than the previous idea. We will use it here to prove Theorem 2.11.

Proof (of Theorem 2.11). Let $A$ be any set, we need to show that $|A| < |\mathcal{P}(A)|$. First notice that clearly $|A| \leq |\mathcal{P}(A)|$, since

$$
x \mapsto \{x\}
$$

is one-to-one.

The heart of the matter is to show that there is no function $f : A \to \mathcal{P}(A)$ that is onto. We will do this using the method of “defeating an arbitrary example”. For any function $f : A \to \mathcal{P}(A)$, we will describe a method to show that it is not onto. That is, for any such function $f$, we must find some subset $S_f$ of $A$ that gets “missed” by the function $f$. One such set is

$$
S_f = \{a \in A | a \notin f(a)\}.
$$

Since for every $a$, $a \in S_f \iff a \notin f(a)$, we see that $S_f \neq f(a)$; that is, $S_f$ is missed by the function $f$. Since we have not assumed anything special about $f$, this shows that no function $f$ can be onto. Therefore $|A| \neq |\mathcal{P}(A)|$. □

Exercises

1. Prove Theorem 2.10.

2. Show that the set of irrational numbers is uncountable.

3. Which of the following are countable, which are uncountable?

   (a) The set of all functions from $\mathbb{N}$ to $\mathbb{N}$.

   (b) The set of all one-to-one functions from $\mathbb{N}$ to $\mathbb{N}$.

   (c) The set of all functions from $\mathbb{N}$ to $\mathbb{N}$ that are eventually 0.

   (d) The set of all functions from $\mathbb{N}$ to $\mathbb{N}$ that are eventually constant.
Chapter 3

Set Theory

As a branch of mathematics, set theory is less than one hundred years old, yet it occupies a unique and critical position. Set-theoretic principles and methods pervade mathematics. Set-theoretic results have shaken the worlds of analysis, algebra, and topology. Simple questions about sets have split the mathematical community into hostile camps, and the romance of its infinite sets have charmed and challenged philosophers as nothing else in mathematics.

Jim Henle

Set theory and set-theoretic notation were born out of the nineteenth century struggle in mathematics to give a clear account of the real number system. In this chapter we will investigate an axiom system known as ZF (for Zermelo-Fraenkel).

An investigation of set theory begins by asking the question: What is a set? In school you were probably taught that a set is a collection of objects. While this intuition is important, this approach won’t get us very far toward our goal of a rigorous foundation for set theory. It is not very precise. (After all, what is a collection? For that matter, what is an object?) Furthermore, this description makes the (for us) unnecessary distinction between sets and objects.

Instead, we will take the approach of stipulating how sets behave. The fundamental property of any set has to do with what “objects” are “in” it. So it seems natural that our language will need to have some way of talking about membership of one set in another set (our objects will all be sets themselves). Let $\mathcal{L} = \{\in\}$ be the language with one binary predicate (the intended meaning of which is to indicate membership of one set in another). It turns out that this simple language will suffice to express axioms rich enough to do an amazing amount mathematics not only about sets but also about many other familiar mathematical objects, like numbers, functions, etc.
3.1 Naive Set Theory

Our first attempt at specifying how sets behave, naive set theory had a certain elegance about it. It was based on only two fundamental principals (the axiom of extension and the axiom scheme of comprehension), both of which seemed intuitively obvious, or at least reasonable assumptions based on the way we naively think about sets.

Our study of naive set theory revealed two things:

1. Naive set theory seemed to be useful for doing mathematics.
   We were able to define many useful mathematical objects like intersections, unions, ordered pairs, functions, relations, etc. and prove things about them.

2. Naive set theory is inconsistent.
   We used a form of Russel’s paradox to show that naive set theory was able to prove the sentence

\[ \forall x (P(x) \not\subseteq x) \]

is both true and false.

We would like to remedy this by revising our axioms in such a way that the resulting theory of sets is still useful for doing mathematics, but no longer able to prove contradictory statements.

3.2 A Second Attempt at Set Theory

We won’t actually quite succeed in meeting the goals listed above. In fact, it is inherently impossible to do so. But we will come close. We will introduce a new set of axioms called ZF that will not longer be susceptible to the Russell attack on its consistency. Unfortunately, we won’t know with certainty that there is not some other inconsistency that no one has yet detected.

We also won’t take time to fully develop “all of mathematics” in ZF, but we will give some indication that this might be possible by doing the following:

- We will show that ZF \vdash \text{Con}(PA).

That is, assuming the axioms of Zermelo-Fraenkel set theory, we can build a model for the axioms of Peano Arithmetic. Although we will not do it here, a similar thing can be done to construct models for the integers, the rationals and the reals. In fact, one can attempt in this way to “do all of ordinary mathematics” (like calculus) and see how much truth there is to the slogan “All mathematics is set theory.”
• We will do some infinite arithmetic.

Actually there are two types of infinite arithmetic, ordinal and cardinal. Hopefully, we will have a chance to talk a little about each one.

But before we can make progress on these goals, we need to take a closer look at the axioms of ZF.

### 3.3 The Axioms of ZF

Our first axiom will formalize our statement that a set is determined by what is “in” it. It is identical to the Axiom of Extensionality from naive set theory.

**Axiom (Extensionality):** Membership determines the set.

\[ \forall a \forall b [\forall z (z \in a \leftrightarrow z \in b) \rightarrow a = b]. \]

In particular, this says that it doesn’t matter how we describe a set, how we denote a set, or how we construct a set, only what ends up belonging to the set (as determined by the relation \( \in \)). Same members, same set. Different members, different sets.

Our second axiom provides us with our first example of a set.

**Axiom (Empty Set):** There is a set with no members.

\[ \exists a \forall x. x \notin a \]

Notice that since we won’t have the powerful comprehension scheme of naive set theory, we need to have a separate axiom to build this set. (How could one show that it does not follow from extensionality alone that there is an empty set? Must there be a set at all?)

Also notice that we have introduced an abbreviation here. Whenever we use the “phrase” \( y \notin x \), officially we mean \( \neg y \in x \). In fact, we will use many abbreviations in our study of set theory. This will make our expressions much easier to read and understand. But in principal, every such statement with abbreviations could be written down as a first order wff in the language \( \mathcal{L} = \{ \in \} \). In fact, for any relation defined by a formula \( \varphi \), we will allow ourselves the convenience of introducing an abbreviation.

**Examples.**

1. The binary relation \( a \subseteq b \) is defined by the wff (with 2 free variables) \( \varphi(a, b) = \forall x [x \in a \rightarrow x \in b] \).

2. The 3-ary relation \( c = a \cap b \) is defined by the wff (with 3 free variables) \( \varphi(a, b, c) = \forall x [x \in c \leftrightarrow [x \in a \land x \in b]] \).
This example deserves a bit more discussion. Typically, we think of intersection (\(\cap\)) as being an operation that builds a new set from two sets, rather than as a relation among three sets. Once we have shown that for any \(a\) and \(b\) there is a set \(c\) such that \(c = a \cap b\) (this will require some more axioms) and that it must be unique (that much we can already do from Extensionality), then we will be free to treat \(\cap\) as a operator (or function) in this way. But officially, when we make some claim \(\Psi(a \cap b)\) we really mean

\[
\exists c \left[ c = a \cap b \land \Psi(c) \right],
\]

which is of course just an abbreviation for

\[
\exists c \left[ \forall x \left[ x \in c \leftrightarrow [x \in a \land x \in b] \land \Psi(c) \right] \right].
\]

You can begin to see why abbreviations are going to be necessary to do anything complicated.

3. \(a = \emptyset\) is an abbreviation for \(\forall x \ x \notin a\), so the Axiom of Extensionality can be rewritten as

\[
\exists a \ a = \emptyset.
\]

**Exercise 3.1.** Write down a wff that defines each of the following relations: \(c = a \cup b\), \(c = \{a, b\}\), \(c = a \setminus b\).\(^1\) \(\triangleright\)

**Exercise 3.2.** There is another kind of abbreviation that will be useful to us. Write wffs that are abbreviated by \(\exists x \in a \varphi\), \(\forall x \in a \varphi\), \(\exists!x \varphi\), and \(\exists!x \in a \varphi\). (\(\exists!x\) is intended to mean ‘there is a unique \(x\) such that’). \(\triangleright\)

We can’t do much with just these first two axioms. In fact,

**Exercise 3.3.** Show that there is a model for Extensionality and Empty Set that has only one element in its universe. \(\triangleright\)

What we need is some way to generate more sets. The next few axioms of \(ZF\) will provide us with ways to build new sets from old sets. We know from the paradoxes we have already discussed that we will need to be at least a little bit careful as we do this. In general, the guiding principal is not to allow sets that are “too big” (like the set of all sets or some other “bad” thing). The four most important of these “set-building axioms” are

---

\(^1\) \(a \setminus b\) is the set of all elements of \(a\) that are not elements of \(b\). In set theory, we often use the symbol \(\setminus\) instead of the usual subtraction symbol since everything is a set and we want to distinguish between, for example, \(5 - 3\) (which equals 2) and \(5 \setminus 3\) (which equals \(\{3, 4\}\)).
Axiom (Pairing): If $a$ and $b$ are sets, so is $\{a, b\}$.

\[ \forall a \forall b \exists c \forall x \ [x \in c \iff x = a \lor x = b] \]

Axiom (Union): If $a$ is a set, then $\cup a$ is a set.

\[ \forall a \exists c \forall x \ [x \in c \iff \exists b [b \in a \land x \in b]] \]

Axiom (Powerset): If $a$ is a set, so is the set of all subsets of $a$.

\[ \forall a \exists b \forall x \ [x \in b \iff x \subseteq a] \]

Axiom (Separation): If $a$ is a set and $\varphi$ is a wff, then $\{x \in a \mid \varphi(x)\}$ is a set.

\[ \forall a \exists b \forall x \ [x \in b \iff [x \in a \land \varphi(x)]] \]

Exercise 3.4. We used an abbreviation ($\subseteq$) in the Powerset Axiom. Rewrite that axiom as an $\mathcal{L}$-wff without any abbreviations.

The Pairing and Powerset Axioms are fairly straightforward. The Union Axiom deserves a little explanation. By $\cup a$ we mean the set $\cup a = \{x \mid \exists b (b \in a \land x \in b)\}$. In this notation, the more familiar $A \cup B$ is $\cup\{A, B\}$. Since by Pairing, $\{A, B\}$ is a set if $A$ and $B$ are sets, we see that ZF allows us to construct $A \cup B$ whenever $A$ and $B$ are sets.

The Separation Axiom is really an axiom scheme. By this we mean that it is a description of countably many axioms, one for each possible wff $\varphi$.\(^2\) Separation allows us to form “definable” subsets. That is, we can select out from any set all of the members of that set which satisfy some property that can be expressed with a wff. This is weaker than saying that any subset can be formed, since there may be subsets that cannot be defined in this manner. For doing everyday mathematics, however, this is usually sufficient, since usually it is not difficult to express the kinds of subsets we need in this manner. And Separation is much weaker than Comprehension, which allowed us to make any definable set, even if it wasn’t a subset of some other set. This is the sense in which have tried to avoid sets that are “too large”.

\(^2\)We have been a bit imprecise about what kind of wff $\varphi$ may be. Clearly $\varphi$ will usually have $x$ free. It may also have $a$ free, but $c$ should not appear free in $\varphi$. Also, $\varphi$ may have additional free variables, in which case we need to preface the axiom with universal quantifiers over all the additional variables – what we have denoted as $\forall \exists$ in the past. This is called the universal closure of a wff with free variables. The separation scheme actually includes all universal closures of the wffs just described, but we will suppress the listing of the free variables and their universal quantifiers to keep the notation manageable.
Note that once we have Separation, then weaker versions of the Pairing, Union, and Powerset Axioms suffice:

\[ \forall x \forall y \exists z \left[ x \in z \land y \in z \right] \]

\[ \forall x \exists z \forall y \forall u \left[ y \in x \land u \in y \rightarrow u \in z \right] \]

\[ \forall x \exists y \forall u \left[ u \subseteq x \rightarrow u \in z \right] \]

If one’s goal is to study models of set theory, then these weaker versions are somewhat easier to establish for a given model. If we are only interested in consequences of ZF, then it doesn’t matter which version we assume, since the stronger versions follow from the weaker versions and Separation.

The Empty Set Axiom is also unnecessary provided we know that there is some set (like the one guaranteed by the Infinity Axiom discussed below), since we can now define the empty set as

\[ \{ x \in a \mid x \neq x \} \]

for any set \( a \).

The remaining axioms of ZF are Infinity, Regularity, and Replacement. The Infinity Axiom is like the Empty Set Axiom in that it guarantees the existence of one particular set, rather than giving a general way for building new sets from old.

**Axiom (Infinity):** There is an infinite set.

\[ \exists a \left[ \emptyset \in a \land \forall x \left[ x \in a \rightarrow x \cup \{ x \} \in a \right] \right] \]

Without this axiom, it is possible to have a model in which every set is finite (although the model itself must be infinite).

**Exercise 3.5.** Show that any model of ZF \(-\text{Inf}\), the axioms of ZF without the Infinity Axiom, must be infinite. Hint: How large is a powerset? Show that there must be sets of infinitely many different sizes. \( \Box \)

With this axiom, there must be a set that contains \( \emptyset, \emptyset \cup \{ \emptyset \} = \{ \emptyset \}, \{ \emptyset \} \cup \{ \{ \emptyset \} \} = \{ \emptyset, \{ \emptyset \} \}, \ldots \) This axiom will be important in our definition of a model for PA, and in fact, we will interpret the successor function of PA using \( S(x) = x \cup \{ x \} \).

**Axiom (Regularity):**

\[ \forall a \left[ a \neq \emptyset \rightarrow \exists b \left[ b \in a \land b \cap a = \emptyset \right] \right] \]

The Axiom of Regularity is less intuitive than some of the others, and in fact, much can be done without it, but is useful to avoid certain pathological sets and relationships between sets. \( \text{ZF}^{-} \) is sometimes used to denote ZF with the Regularity Axiom deleted.
Lemma 3.1 There is no set $a$ such that $a \in a$.

Proof. First note that $\{a\}$ exists since it is the same as $\{a, a\}$, which exists by Pairing. Now apply Regularity to $\{a\}$. Since $\{a\}$ is non-empty and $a$ is the only element in $\{a\}$, it must be that $a \cap \{a\} = \emptyset$. Since clearly $a \in \{a\}$, this means that $a \notin a$, else the intersection would be non-empty. \qed

Exercise 3.6. Show that if $a \in b$, then $b \notin a$. Hint: Apply Regularity to $\{a, b\}$.

We will defer discussion of Replacement until we need it.

3.4 The axioms of PA

Peano Arithmetic is an axiomatization of basic arithmetic on the natural numbers (non-negative integers). Peano Arithmetic works in the language $\mathcal{L} = \{0, 1, S, +, \times\}$, where 0 and 1 are constants, $S$ is a unary function, and $+$ and $\times$ are binary functions. $S$ stands for successor, and the intended meaning is that $S(x)$ should be the “next” natural number after $x$. The goal in choosing these axioms is to choose statements that are “obviously true” about arithmetic on natural numbers, powerful enough to form the basis of our reasoning about and with the natural numbers, yet as simple and few as possible. Over time, the axioms proposed by Giuseppe Peano, an Italian mathematician, have become the standard choice.

Peano Arithmetic has seven regular axioms and one axiom scheme. The axiom scheme is intended to capture how induction works.

1. $\forall x \forall y \left[ S(x) = S(y) \rightarrow x = y \right]$
2. $\forall x \left[ S(x) \neq 0 \right]$
3. $S(0) = 1$
4. $\forall x \left[ x + 0 = x \right]$
5. $\forall x \forall y \left[ x + S(y) = S(x + y) \right]$
6. $\forall x \left[ x \times 0 = 0 \right]$
7. $\forall x \forall y \left[ x \times S(y)) = (x \times y) + x \right]$
8. For any wff $\varphi$ with free variable $x$ (and possibly other free variables, too), the universal closure of

$$\left[ \varphi(0) \land \forall x \left( \varphi(x) \rightarrow \varphi(S(x)) \right) \right] \rightarrow \forall x \varphi(x)$$

is an axiom. This axiom is called induction on the wff $\varphi$. 

You might have expected some other familiar statements to be in this list, things like $\forall x \forall y \ x + y = y + x$, and other sentences saying that addition and multiplication have the usual associative, commutative and distributive properties. It turns out that all these (and much more) are consequences of $\text{PA}$, so for simplicity’s sake, we leave them out of the axioms and instead make them theorems (statements we prove with the axioms as premises). Here are two examples. The parenthesized $\text{PA}$ before each statement indicates that these are consequences of the Peano axioms.

**Theorem 3.2** ($\text{PA}$) $\forall x \ S(x) = x + 1$

*Proof.* Let $x$ be arbitrary (i.e., an arbitrary natural number). By (3) (and the indiscernibility of identicals), $x + 1 = x + S(0)$. By (5), $x + S(0) = S(x + 0)$. By (4), $S(x + 0) = S(x)$. So by transitivity and symmetry of = (or by several uses of indiscernibility of identicals, if you want to go all the way back to first principles), $S(x) = x + 1$. Since $x$ was arbitrary, this is true for all $x$. □

Theorem 3.2 may leave you wondering why one bothers to introduce $S$ into the language at all, since everything can be expressed using $+$. Indeed, *Language, Proof and Logic* does not introduce $S$ into the language. (See page 456 for the axioms there.) Nevertheless, there are reasons to do so. In particular, there is a subtle distinction between adding 1 (successor) and adding an arbitrary number. The successor is the foundation on which all the other addition is built, as you can see from the axioms. Furthermore, what $S(x)$ is depends in a much more significant way on $x$ than on 1, so it proper to think of it as a unary operator. This corresponds in some sense the difference between $x+1$ and $x++$ in the C programming language. In any case, we will need to think carefully about the successor when we build our model of $\text{PA}$ in $\text{ZF}$, so the distinction is a useful one for us.

**Theorem 3.3** ($\text{PA}$) $\forall x \ x + 1 = 1 + x$

*Proof.* This proof is more involved, since it requires the use of induction. Our wff $\varphi(x)$ will be the following:

$$x + 1 = 1 + x$$

We must show that $\varphi(0)$ holds (base case) and that for any $x$, $\varphi(x) \rightarrow \varphi(x + 1)$ (inductive step). (Now that we know that $S(x) = x + 1$, we will freely choose which one we use in a particular situation.) We begin showing the base case, $0 + 1 = 1 + 0$. By axiom (3), $0 + 1 = 1$. By axiom (4), $1 + 0 = 1$. So $0 + 1 = 1 + 0$.

Now we do the inductive step. Let $x$ be any number such that $\varphi(x)$. We must show $\varphi(x + 1)$, i.e., that $(x + 1) + 1 = 1 + (x + 1)$. By the inductive
hypothesis \((\varphi(x))\), we know that \(x + 1 = 1 + x\), so
\[
(x + 1) + 1 = (1 + x) + 1 \text{ by inductive hypothesis}
\]
\[
= 1 + (x + 1) \text{ by axiom (5)}
\]
\]

The other familiar properties of arithmetic follow by similar (but longer) sorts of proofs. Each one will use induction. This is because Peano’s axiom scheme gives inductive (or recursive) definitions of addition and multiplication.

One last note on PA. We have constant symbols for 0 and for 1, but not for the other natural numbers. If necessary, we can consider 2, 3, etc. to be abbreviations for \(S(S(0))\), \(S(S(S(0)))\), etc. In this way, we are able to talk about any of our old favorite natural numbers we like even though they do not have constant symbol names.

### 3.5 A Model for PA: The Universe

We want to show that in ZF we can construct a model of PA. This will show us that if ZF is consistent, then so is PA, since it has a model. (Of course, if ZF is not consistent, then it can prove that there is a model for PA and it can prove there is no such model.) We take this as evidence of two things: ZF is useful for doing mathematics, and PA is a reasonable axiom set for arithmetic.

Let’s call the model we are going to construct \(\mathcal{M}\). We are part of the way there already. We will let \(0^\mathcal{M}\) be 0, and \(S^\mathcal{M}\) be defined by \(S^\mathcal{M}(x) = x \cup \{x\}\). But we are getting a little bit ahead of ourselves. We haven’t even said what the universe of our model is supposed to be. Furthermore, we need to say something about functions, since \(S^\mathcal{M}\) is supposed to be a function defined on the universe. We will deal with the universe in this section and the functions (including \(+^\mathcal{M}\) and \(*^\mathcal{M}\) as well as \(S^\mathcal{M}\)) in the next section.

We want our universe to be exactly \(\omega = \{0, S(0), S(S(0)), \ldots\}\). (In set theory this set is usually denoted by \(\omega\) rather than \(\mathbb{N}\), although it is essentially the same object.) So we are building the “standard model” of PA, which until now we have just been assuming existed. All we need to do is show that ZF implies that the \(\omega\) exists. The Infinity Axiom almost says this, but we need to combine it with Separation to get just the set we want.

Let \(\varphi(z)\) be the wff
\[
\varphi(z) = \emptyset \in z \land \forall x \ [x \in z \rightarrow x \cup \{x\} \in z] .
\]
Then the Infinity Axiom is just \(\exists z \varphi(z)\). For some such \(z\), we use Separation to build the set
\[
\omega = \{x \in z \mid \forall u \ [u = x \lor u \in x] \rightarrow \{u = \emptyset \lor \exists v \ S(v) = u\}\} .
\]
The intuition for this definition is that we want to include in $\omega$ only those things which are built up from $\emptyset$ using successor. We want to remove from $z$ any other types of sets. We will say more about how to formalize $S(x) = x \cup \{x\}$ as a function from $\omega$ to $\omega$ in the next section. For now, let’s plow on and show that $S$ has the desired properties.

**Lemma 3.4 (ZF)** The following statements are true about $S$ and $\omega$:

1. $\forall x \ 0 \neq S(x)$;
2. $\forall x \forall y \ [S(x) = S(y) \rightarrow x = y]$;
3. $\forall x \in \omega \ S(x) \in \omega$;
4. $\forall y \ [\varphi(0) \land \forall x \ [\varphi(x) \rightarrow \varphi(S(x))] \rightarrow \forall x \in \omega \ \varphi(x)]$.

**Exercise 3.7.** Prove Lemma 3.4. Hint: For (2) use Exercise 6. For (4) apply Regularity to $X = \{x \in \omega \mid \neg \varphi(x)\}$ to show that $X$ must be empty. 

By Lemma 3.4, once we have shown that $S$ is a function on $\omega$, our model $M$ will satisfy axioms 1, 2 and Induction of $\text{PA}$. Statement (4) also justifies our use of informal induction on $\omega$, which we can use to prove the following useful facts about the way $\in$ behaves on $\omega$.

**Lemma 3.5 (ZF)** Properties of $\in$ on $\omega$:

1. $\forall x \in \omega \ \forall y \in \omega \ [x \in y \rightarrow y \not\subseteq x]$.
2. $\forall x \in \omega \exists y \ [y \in x \rightarrow y \subseteq x]$;
3. $\forall x \in \omega \ \forall y \in \omega \ x \in y \iff x \subseteq y$
4. $\forall x \in \omega \ \forall y \in \omega \ \forall z \in \omega \ [x \in y \land y \in z \rightarrow x \in z]$;
5. $\forall x \in \omega \ \forall y \in \omega \ [x = y \lor x \in y \lor y \in x]$.

**Proof.** (1): This follows immediately from Exercise 6.

(2): Induct on $x$. If $x$ is 0, then the statement is vacuously true. If $x = k \cup \{k\}$ and the statement is true for $k$ in place of $x$, then $y \in x = k \cup \{k\}$ implies that either $y \in k$ so by induction $y \subseteq k \subseteq x$, or else $y = k \subseteq x$.

(3): This is proved by induction on $y$.

- **Base Case:** $y = 0$.
  It is vacuously true if $y = 0$, since there are no $x$ such that $x \in 0$ and there are no $x$ such that $x \subseteq 0$.

- **Inductive Step:** Suppose that $y = S(z) = z \cup \{z\}$ for some $z$ and that $x \in z \iff x \subseteq z$.

  First, we show that if $x \in y$, then $x \subseteq y$. Since $x \in y = z \cup \{z\}$, there are two cases to consider:

  1. $x \in z$.
  2. $x = z$.

  In both cases, $x \subseteq z$ and hence $x \subseteq y$. 

  Next, we show that if $x \subseteq y$, then $x \in y$.

  1. If $x \subseteq z$,
  2. If $x = z$.

  In both cases, $x \in y$.
Case 1: $x = z$. In this case, $x = z \subseteq z \cup \{z\} = y$, so $x \subseteq y$.

Case 2: $x \in z$.

In this case, by the inductive hypothesis, $x \subseteq z \subseteq z \cup \{z\} = y$.

So in either case $x \subseteq y$.

On the other hand, suppose that $x \subseteq y = z \cup \{z\}$. Again there are two cases to consider.

- $x \subseteq z$
  - If $x \subseteq z$, then either $x \subseteq z$ or $x = z$. If $x \subseteq z$, then $x \in z$ (by the inductive hypothesis). Either way, $x \in z \cup \{z\} = y$.
  - $z \in x$
    - In this case, by the inductive hypothesis, $z \subseteq x$. So $z \subseteq x \subseteq z \cup \{z\}$, which is a contradiction, since $z$ is missing only one element from $z \cup \{z\}$. So this case doesn’t happen.

(4): If $x \in y$ and $y \in z$, then by (2) $x \subseteq y$ and $y \subseteq z$, so $x \subseteq z$, hence by (2) again, $x \in z$.

(5): First notice that by (3), (5) is equivalent to $\forall x \in \omega \forall y \in \omega \ [x \subseteq y \lor y \subseteq x]$. Now induct on $x$.

- Base case: If $x = 0$ the result is obvious since $0 \subseteq y$ for any $y$.

- Inductive step: $x = k \cup \{k\}$, and for all $y \in \omega$, $y \subseteq k$, $k \subseteq y$ or $y = k$.

  Let’s look at each case.

  - If $y \subseteq k$, then $y \subseteq k \subseteq x$, so $y \subseteq x$.
  - If $y = k$, then $y \subseteq x$.
  - If $k \subseteq y$, then (by (3)) $k \in y$, so $\{k\} \subseteq y$, so $x = k \cup \{k\} \subseteq y$.

\[\square\]

Notice that (1), (3) and (4) imply that $\in$ is a strict linear order on $\omega$.

You may remember that when we introduced PA, we mentioned that one can define $x < y$ by $\exists z \ x + S(z) = y$. Of course, once we have defined addition on $\omega$, we could do the same thing here. It turns out that both orders are the same, that is

$$\forall x \in \omega \forall y \in \omega \ [x \in y \iff \exists z \in \omega \ x + S(z) = y]$$

We will write $x < y$ if $x \in y$, and $x \leq y$ if $x \in y \lor x = y$.

### 3.6 A Model for PA: The Functions

In the last section we postponed the question of what a function is. Now we need to answer it. So what is a function? Well, in set theory, everything
is a set, so a function must be some sort of set. And sets are determined by their members, so what we are really asking is what belongs to (the set representing) a function.

Let’s suppose $f : A \rightarrow B$ is a function. What set should it be? Typically, we think of the function $f$ as a rule telling us how to assign to each element $a \in A$ an unique element $b \in B$. In set theory, we will assume that that rule is expressed as a list (i.e., a set) of all such ordered pairs $a$ and $b$.

But what is an ordered pair? Once again, it must be some set (everything is a set). Let’s denote an ordered pair by $h\ a;b\ i$. The key property of an ordered pair is that $h\ a;b\ i = h\ c;d\ i$ if and only if $a = c$ and $b = d$. The Pairing Axiom lets us build sets like $f\ a;b\ g\ f\ a;g\$, but this set does not distinguish the order of $a$ and $b$ and is the same as $f\ b;a\ g\$. After a little experimenting, we find that there is a reasonable set to call $h\ a;b\ i$ and that the axioms of ZF imply that this set exists whenever $a$ and $b$ are sets.

Exercise 3.8. Here is a list of possibilities for $h\ a;b\ i$. Only one of them has the desired properties. Find it and prove that it works. For the others, show why they fail to work:

\[
\{a,b\} \quad \{a,\{b\}\} \quad \{(a),\{b\}\} \quad \{(a),\{a,\{b\}\}\} \quad \{a,\{a,\{b\}\}\}
\]

Exercise 3.9. Prove that if $A$ and $B$ are sets, then $A \times B$ is a set. Hint: Find a set big enough to contain $A \times B$ and then use Separation to get exactly $A \times B$. You will need the answer to Exercise 8 to do this.

A function from $A$ to $B$ is now just a set with the following properties:

- $f \subseteq A \times B$,
- $\forall a \in A \exists b \in B \ (a,b) \in f$,
- $\forall a\forall y\forall z \ ((a,y) \in f \land (a,z) \in f) \rightarrow y = z$.

Notice that the last two properties can be combined into the following wff (with abbreviations):

- $\forall a \in A \exists! b \in B \ (a,b) \in f$.

Now we introduce a number of abbreviations for functions. $f : A \rightarrow B$ is an abbreviation for “$f$ is a function from $A$ to $B$” (i.e., for the conjunction of the three wffs in the definition above); “$f$ is a function” abbreviates $\exists! A \exists B \ f : A \rightarrow B; f(x) = y$ abbreviates $(x,y) \in f$; range($f$) = $\{y \mid \exists x \ f(x) = y\}$; and $f \upharpoonright D = \{\langle x,y\rangle \in f \mid x \in D\}$.

Exercise 3.10. Write wffs (you may use other abbreviations) that define “$f$ is one-to-one”, “$f$ is onto $B$”, and “$f$ is a function from $A$ to $B$ and $g = f^{-1}$.”
These abbreviations will allow us to use our usual notation for functions when it is convenient to do so. There are times, however, when knowing that \( f \) is really just a set with certain properties is also handy.

The following properties of functions are easy to prove in ZF:

**Lemma 3.6 (Function Lemma)**

1. If \( f \) is a function, then \( \text{range}(f) \) is a set.
2. If \( f \) is a function and \( X \subseteq f \), then \( f \) is a function.
3. If \( f \) is a function and \( D \) is a set, then \( f \upharpoonright D \) is a function.
4. If \( f : A \to B \) is one-to-one and \( g = f^{-1} \), then \( g \) is a one-to-one function.

**Exercise 3.11.** Prove Lemma 3.6. Hint: For (2), remember that “\( f \) is a function” means \( f : A \to B \) for some sets \( A \) and \( B \). The trick is to show that the appropriate sets \( A \) and \( B \) exist. Use Union and Separation for this. For (3) and (4), use (2).

Now we are ready to finish our model \( M \) for PA. First let’s deal with successor:

**Lemma 3.7** There is a function \( f : \omega \to \omega \) such that for every \( x \in \omega \), \( f(x) = S(x) \).

**Proof.** Let \( f = \{ (x, y) \in \omega \times \omega : S(x) = y \} \).

Note that by Lemma 3.4 if we let \( S^M \) be the function from the previous lemma, then axioms 1, 2 and the Induction Scheme of PA are satisfied by our model.

**Lemma 3.8** There are functions \( \alpha : \omega \times \omega \to \omega \) and \( \mu : \omega \times \omega \to \omega \) such that

1. For all \( n \in \omega \), \( \alpha(\langle n, 0 \rangle) = n \).
2. For all \( n, k \in \omega \), \( \alpha(\langle n, S(k) \rangle) = S(\alpha(\langle n, k \rangle)) \).
3. For all \( n \in \omega \), \( \mu(\langle n, 0 \rangle) = 0 \).
4. For all \( n, k \in \omega \), \( \mu(\langle n, S(k) \rangle) = \alpha(\langle \mu(\langle n, k \rangle), n \rangle) \).

Note that once we have proven the lemma we will let \( +^M = \alpha \) and \( \ast^M = \mu \).

**Proof.** We will only prove the result for \( \alpha \), the proof for \( \mu \) is similar.

It turns out that this result is trickier than it might first appear. In fact, we will need to introduce the axiom scheme of Replacement in order to
accomplish it. Here is the basic idea of the proof. Suppose we want to show that such a set/function $\alpha$ exists. We would like to build it up in stages. For example, we know how to add when the second addend is 0: $m + 0 = m$. So let’s let

$$A_0 = \{m, n, r\} \in \omega \times \omega \times \omega \mid n = 0 \land m = r.$$ 

$A_0$ exists by Separation.

Now we want to let $A_1$ be the part of $\alpha$ that tells us what to do when we add 1:

$$A_1 = \{m, S(n), S(r)\} \in \omega \times \omega \times \omega \mid \langle m, n, r \rangle \in A_0.$$ 

And of course, we want to have

$$A_{i+1} = A_{S(i)} = \{m, S(n), S(r)\} \in \omega \times \omega \times \omega \mid \langle m, n, r \rangle \in A_i.$$ 

Finally we let $\alpha = \bigcup_{i=0}^{\infty} A_i$. In this way we build up $\alpha$ stage by stage.

So where is the rub? We need to justify the existence of all the $A_i$s and $\alpha = \bigcup_{i=0}^{\infty} A_i$. For the latter, we need a set $A = \{A_i \mid i \in \omega\}$. Then we can use the Union Axiom to define $\alpha = \bigcup A$. (Remember, that is the way we formalize $\bigcup_{i=0}^{\infty} A_i$.)

The Axiom of Replacement allows us to build just this sort of set. Notice the form of this set: for each $i \in \omega$, we want to put $A_i$ into $A$. That is we want to replace each $i$ with $A_i$. More generally, Replacement allow us to build sets of the form

$$\{F(x) \mid x \in A\}$$

where $A$ is a set and $F$ behaves like a function (for each $x \in A$ there must be exactly one $F(x)$) but is defined in terms of wffs rather than sets (since we want to use it to prove the existence of the sets involved).

Why should we allow such an axiom in ZF? The intuition is that if we map each element $x$ of the set $A$ to $F(x)$, then the resulting collection of all these images under $F$ is no larger than $A$ was, and no more complicated, so it should be a set, too.

Here is the formal axiom (scheme):

**Axiom (Replacement):** For each wff $\varphi$ with free variables $x$ and $\vec{z}$, we have the axiom

$$\forall a \forall \vec{z} \ [\forall x \in a \exists! y \varphi(x, y, \vec{z}) \rightarrow \exists b \forall x \in a \exists y \in b \ P(x, y, \vec{z})]$$

By combining this with separation, we can change the conclusion to be

$$\forall y \ [y \in b \leftrightarrow P(x, y, \vec{z})] .$$

That is, if we can prove that for each $x$ in some set $A$ there is a unique $y$ such that $\varphi(x, y)$, then we may justify the formation of

$$\{y \mid x \in A \land \varphi(x, y)\} .$$
As with our axiom schemes of Comprehension and Separation, the $\bar{z}$ can be thought of as allowing parametrization.

Now we have just what we need. We can use Replacement (and Separation) to define the $A_i$'s.

By Extension, each $A_i$ will be a unique set, so we can apply Replacement (and Separation) to build $A = A_i \mid i \in \omega$.

Finally, we let $\alpha = \bigcup A = \bigcup_{i \in \omega} A_i$.

Putting everything together, we get

\textbf{Theorem 3.9} $\text{ZF} \vdash \text{Con}(\text{PA})$. That is, assuming the axioms of Zermelo-Fraenkel set theory, we can show that there is a model for PA, and hence that PA cannot prove a contradiction. \qed

Note that we are implicitly using Soundness and Completeness here to say that there is such a connection between proof and truth.
Chapter 4

Computability Theory

These brief notes are meant to accompany the exposition of Turing machines in *Introduction to Automata Theory, Languages, and Computation* by Hopcroft and Ullman.

4.1 Informal Computability

Let’s begin by giving some definitions that will allow us to talk about an informal notion of computability.

**Definition 4.1 (Computable function)** A function $f : A \subseteq \Sigma^* \rightarrow \Sigma^*$ is a computable function with domain $A$ if there is a computation that for every input $x \in A$ outputs $f(x)$.

**Definition 4.2 (Computable set)** A set $A \subseteq \Sigma^*$ is computable (also called decidable) if there is a computation that on every input $x \in \Sigma^*$ correctly answers the question “is $x \in A$?”

**Definition 4.3 (Computably enumerable set)** A set $A \subseteq \Sigma^*$ is computably enumerable if there is a computation that successively lists all and only elements of $A$.

The idea behind this last definition is that by cycling through the various inputs $x$ and computing $f(x)$, we get a “listing” of all member of $A$ (perhaps with repetitions). Thus $f$ enumerates $A$.

*Examples.* Let $\Sigma$ be the set of symbols used in FOL.

1. $\text{SENTENCES} = \{ S \mid S \text{ is a sentence in FOL} \}$ is computable.
2. $\text{TAUT} = \{ S \mid S \text{ is a tautology in FOL} \}$ is computable.
3. $\text{CONSEQUENCES}_T = \{ S \mid T \vdash S \}$ is computably enumerable.
4.1.1 Numerical Computation

Fix some encoding of natural numbers as strings in $\Sigma^*$ for some finite alphabet $\Sigma$. (Examples: unary encoding, binary encoding, decimal encoding.) Let $\overline{n}$ be the encoding of $n$ in this scheme. With such an encoding scheme in hand, we can define computable functions and sets of natural numbers.

**Definition 4.4 (Computable function)** A function $f : \mathbb{N} \to \mathbb{N}$ is a computable function if there is a computation that for every input $\overline{n}$ outputs $\overline{f(n)}$.

**Definition 4.5 (Computable set)** A set $A \subseteq \mathbb{N}$ is computable (also called decidable) if there is a computation that on every input $\overline{n}$ correctly answers the question “is $n \in A$?”

**Definition 4.6 (Computably enumerable set)** A set $A \subseteq \mathbb{N}$ is computably enumerable if there is a computation that successively lists (the codes of) all and only elements of $A$.

Note that the same idea of encodings can be used to define sets and functions on other types of mathematical objects in terms of strings in some $\Sigma^*$. In particular, it is often handy to define functions with more than one input. There are many ways to do so, here is one easy way: We will encode input pairs $(x, y)$ with the string $x \# y$. That is, we will place $\#$ between the to inputs. This can be extended to arbitrarily many inputs in the obvious way. (We are assuming here that $\#$ is a symbol not already in $\Sigma$.)

**Examples.**

1. $\text{PRIMES} = \{n \mid n \text{ is a prime number}\}$ is a computable set.

2. Most of the commonly used functions in mathematics are computable functions. (Addition, multiplication, squaring, logarithms (rounded), etc.)

### 4.2 Formal Computability

Our definitions above are imprecise because all of them rely on the undefined notion of *computation*. In order to make the definitions above precise, we must give a precise definition of computation. Notice that computations must be able to do one of two things: In the definition of computable function the computation must produce “outputs” for given “inputs”; in the definition of computable set, the computation must make “decisions” (answer yes or no) for each given input.

Our (first) formal definition of computation will be “what a 1-way 1-tape Turing machine computes.” We will refer to this notion of computation as TM-computation. (See Hopcroft/Ullman for details about our definition of Turing machine.)
Definition 4.7 (Turing-computable function) A function \( f : A \subseteq \Sigma^* \to \Sigma^* \) is Turing-computable if there is a Turing machine \( M \) such that for every input \( x \in A \), \( M \) halts with \( f(x) \) on its tape and all other tape cells blank.

For Turing machine decisions, we have several roughly equivalent ways to represent a decision. Here is our official way: We will denote some of the states of the Turing machine as “accepting” states and the rest as rejecting states and use the state in which the machine halts to indicate the decision.

Definition 4.8 (Turing-computable set) A set \( A \subseteq \Sigma^* \) is Turing-computable if there is a Turing Machine that on every input \( x \in \Sigma^* \) halts in an accepting state if \( x \in A \) and in a rejecting state if \( x \notin A \).

Definition 4.9 (Computably enumerable set) A set \( A \subseteq \Sigma^* \) is computably enumerable if there is a Turing-computable function \( f \) such that \( A = \text{range}(f) \), that is \( x \in A \iff \exists z f(z) = x \).

The idea behind this last definition is that by cycling through the various inputs \( z \) and computing \( f(z) \), we get a “listing” of all members of \( A \) (perhaps with repetitions). Thus \( f \) “enumerates” \( A \). See the exercises for another description of computably enumerable sets.

We will need to demonstrate that this definition of computation corresponds well with our intuitive notions about computation and that it is not too sensitive to its particulars. The following theorems are a beginning in this direction.

Theorem 4.10 The following functions are Turing-computable:

1. \( f(n) = 0 \).
2. \( f(n) = n + 1 \).
3. \( f(n) = 2n \).
4. \( f(m, n) = m + n \).

Definition 4.11 (Neat Computation) A Turing machine computation is neat if the following are true:

1. There is exactly one halting state, and the machine only halts in that state.
2. At the end of the computation, the tape head is returned to the left end of the tape.
3. The output (if any) begins in the leftmost tape cell.

Theorem 4.12 (Neat Computation Theorem) If \( f \) is a Turing computable function, then \( f \) is computed by a TM that computes neatly. If \( A \) is a Turing computable set, then \( A \) is computed by a TM that computes neatly.

Theorem 4.13 (Composition Theorem) If \( f \) and \( g \) are Turing-computable, then \( f \circ g \) is computable. (\( f \circ g(x) = g(f(x)) \).)
4.3 Notation

It is handy to introduce the following notation.

<table>
<thead>
<tr>
<th>notation</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>input tape</td>
<td>the tape that hold the input</td>
</tr>
<tr>
<td>output tape</td>
<td>the tape from which output is read (may be same as input tape)</td>
</tr>
<tr>
<td>work tape</td>
<td>a tape that is neither an input tape nor an output tape</td>
</tr>
<tr>
<td>$M(x) = y$</td>
<td>on input $x$, machine $M$ halts with $y$ on its output tape</td>
</tr>
<tr>
<td>$M(x) \downarrow$</td>
<td>on input $x$, machine $M$ halts</td>
</tr>
<tr>
<td>$M(x) \uparrow$</td>
<td>on input $x$, machine $M$ fails to halt</td>
</tr>
<tr>
<td>$M(x)$ accepts</td>
<td>on input $x$, machine $M$ halts in an accepting state</td>
</tr>
<tr>
<td>$M(x)$ rejects</td>
<td>on input $x$, machine $M$ halts in a rejecting state</td>
</tr>
</tbody>
</table>

4.4 Exercises

Exercise 4.12. Write 1-way infinite 1-tape Turing Machines to compute each of the following functions neatly.

1. $f(n) = 0$.
2. $f(n) = 2^n$.
3. $f(m, n) = n + m$

You may use either unary or binary encodings, but must use the same encoding for all three parts. If you use unary, use the code $\sim n \sim = 0^n + 1$ ($n + 1$ 0’s).

Exercise 4.13. More Challenging TM problems. Write 1-way infinite 1-tape Turing Machines to compute each of the following functions:

1. The conversion from a single binary input to its equivalent unary code.
2. The conversion from a single unary input to its equivalent binary code.

If you are unable to do this with a single tape, you can make this problem simpler by doing it with multiple tapes.


Exercise 4.15. Busy Beaver Problem. Let $b(n)$ denote the maximum number of ones that an $n$-state Turing machine with tape alphabet $\Gamma = \{1, \#\}$ can print on an initially blank tape and then halt. Extra credit is available for proving interesting facts about $b(n)$. For starters, you might like to determine $b(n)$ for some small values of $n$. 
4.5 Robustness of the Turing Machine Model

Chapter 7 of Hopcroft/Ullman shows examples of several ways that one might consider modifying the definition of Turing machine: 2-way infinite tapes, multiple tapes, 2-dimensional tapes, etc. Consult that text for details. The main point is that all these different variations compute the same functions. In fact, many formal models of computation have been proposed over the years, some of which look very different from Turing machines. All of them have been proven to be equivalent to or weaker than the Turing machine model. This has led most mathematicians to accept the following statement.

**Thesis 4.14 (Church’s Thesis)** A function is computable if and only if it is Turing-computable.

Church’s thesis cannot really be proven, since the notion of computable is not formalized to anything specific. This is why it is called a thesis rather than a theorem or conjecture. It could be essentially refuted by finding something that one can prove is not Turing-computable but which is nevertheless clearly computable (in some intuitive, informal sense). Similar statements can of course be made about computable sets and computably enumerable functions. The word “computable” is usually treated as a stand-in for any of the many equivalent formalizations, such as Turing-computable.

4.6 More Exercises

**Exercise 4.16.** There are many ways to define computable sets. Here is an example: Show that a set $S$ is computable if and only if there is a computable function $f$ such $x \in S \iff f(x) = 1$. Do this by describing at a high level the Turing machines involved.

You may use Church’s Thesis for the rest of these problems. This means that to show something is computable, you need only describe a high level computation (in any combination of programming language, psuedo-code and English).

**Exercise 4.17.** There are also many ways to define computably enumerable sets. Here is an example: Show that a set $S$ is computably enumerable if and only if there is a Turing machine $M$ such that $x \in S \iff M(x) \downarrow$.

**Exercise 4.18.** Here is yet another way to define computably enumerable sets: Show that a set $S$ is computably enumerable if and only if there is a two-tape Turing machine $M$ such that

- $M$ never moves left on its second tape and never overwrites a non-blank tape-cell on its second tape. [Such a tape is usually called an output tape.]
• When $M$ is run with blank input tape, then $M$ writes on its output tape all the elements of $S$ (in any order, repetitions allowed) separated by $\#$.

Note that $M$ might never halt.

Exercise 4.19. Show that if $A \subseteq \mathbb{N}$ is a computable set, then $\mathbb{N} - A$ is also a computable set.

Exercise 4.20. Show that the collection of computable sets is closed under intersection, union and complement.

Exercise 4.21. Show that there is a set $A$ that is not computable.

Exercise 4.22. Show that there is a set $A$ that is not computably enumerable.

Exercise 4.23. Let $A$ be computably enumerable. Show that if $\mathbb{N} - A$ is also a computably enumerable, then $A$ is computable.

Exercise 4.24. Show that the collection of computably enumerable sets is closed under intersection and union.

Exercise 4.25. Show that if the collection of computably enumerable sets is closed under complement, then every computably enumerable set is computable. (The fact that we have two different names is a hint that we will eventually show that there is a computably enumerable set that is not computable, hence that the computably enumerable sets are not closed under complement – stay tuned.)
Chapter 5

Computational Complexity

These brief notes are meant to accompany the exposition of Turing machines, computability and computational complexity in Introduction to Automata Theory, Languages, and Computation by Hopcroft and Ullman.

Earlier we saw that we could use a particular model of computation (Turing machines) to define computable sets and functions. Now we are interested in considering the computational complexity of sets of natural numbers. One can do this for functions as well, but we will restrict ourselves to sets here. Since most anything can be coded as a natural number, requiring our sets to be sets of natural numbers is not really much of a restriction.

5.1 Definitions

The classes of interest are the following:

Definition 5.1 1. Computable Sets.

A set $A$ is computable if there is a Turing machine $M$ such that for any input $x$,

(a) $M$ halts when run on input $x$ ($M(x) \downarrow$);

(b) $x \in A \iff M(x) \text{ accepts.}$\footnote{We defined acceptance in terms accepting and rejecting states. Acceptance can also be defined in other ways. For example, the Turing machine could have a special output tape on which it writes a 0 or a 1. This would be equivalent.}

2. Computably Enumerable Sets.

There are several equivalent definitions for computably enumerable sets. $A$ is computably enumerable if . . .

(a) there is a Turing machine $M$ such that $x \in A \iff M(x) \downarrow$
(b) there is a Turing machine $M$ such that if $x \in A$ then $M(x)$ accepts, and if $x \notin A$, then $M(x)$ may reject or may simply fail to halt.

(c) there is a Turing machine $M$ that if run with an empty input tape will write out all the elements of $A$ on an output tape.

(d) $A = \emptyset$ or there is a computable function $f$ such that $A = \{f(0), f(1), f(2), \ldots\}$. (Note that list enumeration need not be in order and may include repeats.)

(e) there is a computable set $C$ such that $x \in A \Leftrightarrow \exists y (x, y) \in C$

3. **P** – polynomial time

A set $A$ is said to be polynomial time computable (written $A \in \text{P}$) if there is a Turing machine $M$ and a polynomial $p$ such that for any input $x$

(a) $M$ halts when run on input $x$ ($M(x) \downarrow$) and this takes at most $p(|x|)$ steps, where $|x|$ is the length of $x$ (in binary);

(b) $x \in A \Leftrightarrow M(x)$ accepts.

4. **NP** – nondeterministic polynomial time

A set $A$ is said to be in the class $\text{NP}$ (written $A \in \text{NP}$) if there is a set $C \in \text{P}$ such that for all inputs $x$

- $x \in A \leftrightarrow \exists^p y (x, y) \in C$

that is, there is a polynomial $q$ such that

- $x \in A \leftrightarrow \exists y ((x, y) \in C \land |y| \leq q(|x|))$

The class $\text{NP}$ can also be defined in terms of Turing machines that are non-deterministic (they may have more than one possible transition for some situations).

### 5.2 Reductions

There are many ways in which algorithms for determining membership in one set might be used as sub-routines in an algorithm for determining membership in another. These lead to several different notions of reducibility between sets. We will study one of the most important of these, namely many-one reducibility. We will say that $A \leq_m B$ if there is a computable function $f$ such that

$$x \in A \leftrightarrow f(x) \in B.$$

If $f$ is computable in polynomial time, then we will write $A \leq^p_m B$.

**Theorem 5.2** $\text{P}$ and $\text{NP}$ are closed downward under $\leq^p_m$-reductions. That is,
1. If $A \leq_{m}^{P} B$ and $B \in \mathbf{P}$, then $A \in \mathbf{P}$.

2. If $A \leq_{m}^{P} B$ and $B \in \mathbf{NP}$, then $A \in \mathbf{NP}$.

Proof. We will prove only the claim for $\mathbf{NP}$. Let $f$ be a reduction $f : A \leq_{m}^{P} B$, and let $C \in \mathbf{P}$ be a witness for $B \in \mathbf{NP}$. Then

\[
\begin{align*}
  x \in A & \iff f(x) \in B \\
  & \iff \exists^{P} y (f(x), y) \in C \\
  & \iff \exists^{P} y (x, y) \in C^*
\end{align*}
\]

where $C^* = \{ (x, y) \mid (f(x), y) \in C \}$. Clearly $C^*$ is polynomial time computable (compute $f(x)$, then the pair $(f(x), y)$, then check (in polynomial time) that $(f(x), y) \in C$), so $A \in \mathbf{NP}$.

Among the many sets in $\mathbf{NP}$, there is one that is both related to logic and especially important. That set is

\[ \text{SAT} = \{ x \mid x \text{ codes a satisfiable quantifier-free formula } \varphi \} \]

SAT is important because it is one of the first examples of an NP-complete set.

Definition 5.3 Hardness and Completeness.

1. A set $L$ is $\mathbf{NP}$-hard if $L \in \mathbf{NP}$ and for any $A \in \mathbf{NP}$, $A \leq_{m}^{P} L$.

2. If $L \in \mathbf{NP}$ and $L$ is $\mathbf{NP}$-hard, we say that $L$ is $\mathbf{NP}$-complete (under many-one reductions).

Theorem 5.4 SAT is $\mathbf{NP}$-complete. That is, SAT $\in \mathbf{NP}$ and if $A \in \mathbf{NP}$ then $A \leq_{m}^{P} \text{SAT}$.

NP-complete sets (there are very many important problems that are $\mathbf{NP}$-complete) are the “hardest things in $\mathbf{NP}$”. In particular, if $L$ is $\mathbf{NP}$-complete and $L \in \mathbf{P}$, then $\mathbf{P} = \mathbf{NP}$, since $\mathbf{P}$ is closed under $\leq_{m}^{P}$-reductions. Since no one knows a polynomial time algorithm for any of these sets, and many of them have been studied extensively, most researchers believe that $\mathbf{P} \neq \mathbf{NP}$. In any case, if you are trying to write an algorithm for some problem and discover that it is $\mathbf{NP}$-complete, either you will be unable to find an algorithm that runs in polynomial time or you will be famous (at least, if you are the first to find a polynomial time algorithm for such a problem).