7 Infinite Sets

This chapter is about infinite sets and some challenges in proving things about them.

Wait a minute! Why bring up infinity in a Mathematics for *Computer Science* text? After all, any data set in a computer is limited by the size of the computer's memory, and there is a bound on the possible size of computer memory, for the simple reason that the universe is (or at least appears to be) bounded. So why not stick with *finite* sets of some large, but bounded, size? This is a good question, but let's see if we can persuade you that dealing with infinite sets is inevitable.

You may not have noticed, but up to now you've already accepted the routine use of the integers, the rationals and irrationals, and sequences of them—infinite sets, all. Further, do you really want Physics or the other sciences to give up the real numbers on the grounds that only a bounded number of bounded measurements can be made in a bounded universe? It's pretty convincing—and a lot simpler—to ignore such big and uncertain bounds (the universe seems to be getting bigger all the time) and accept theories using real numbers.

Likewise in computer science, it's implausible to think that writing a program to add nonnegative integers with up to as many digits as, say, the stars in the sky—billions of galaxies each with billions of stars—would be different from writing a program that would add *any* two integers, no matter how many digits they had. The same is true in designing a compiler: it's neither useful nor sensible to make use of the fact that in a bounded universe, only a bounded number of programs will ever be compiled.

Infinite sets also provide a nice setting to practice proof methods, because it's harder to sneak in unjustified steps under the guise of intuition. And there has been a truly astonishing outcome of studying infinite sets. Their study led to the discovery of fundamental, logical limits on what computers can possibly do. For example, in Section 7.2, we'll use reasoning developed for infinite sets to prove that it's impossible to have a perfect type-checker for a programming language.

So in this chapter, we ask you to bite the bullet and start learning to cope with infinity.

7.1 Infinite Cardinality

In the late nineteenth century, the mathematician Georg Cantor was studying the convergence of Fourier series and found some series that he wanted to say converged "most of the time," even though there were an infinite number of points where they didn't converge. As a result, Cantor needed a way to compare the size of infinite sets. To get a grip on this, he got the idea of extending the Mapping Rule Theorem 4.5.4 to infinite sets: he regarded two infinite sets as having the "same size" when there was a bijection between them. Likewise, an infinite set A should be considered "as big as" a set B when A surj B. So we could consider A to be "strictly smaller" than B, which we abbreviate as A strict B, when A is *not* "as big as" B:

Definition 7.1.1. $A \text{ strict } B \quad \text{iff} \quad \text{NOT}(A \text{ surj } B).$

On finite sets, this strict relation really does mean "strictly smaller." This follows immediately from the Mapping Rule Theorem 4.5.4.

Corollary 7.1.2. *For finite sets A, B,*

$$A \text{ strict } B \quad \textit{iff} \quad |A| < |B|.$$

Proof.

$$\begin{aligned} A \text{ strict } B \quad & \text{iff} \quad \text{NOT}(A \text{ surj } B) && \text{(Def 7.1.1)} \\ & \text{iff} \quad \text{NOT}(|A| \geq |B|) && \text{(Theorem 4.5.4.(4.5))} \\ & \text{iff} \quad |A| < |B|. \end{aligned}$$

■

Cantor got diverted from his study of Fourier series by his effort to develop a theory of infinite sizes based on these ideas. His theory ultimately had profound consequences for the foundations of mathematics and computer science. But Cantor made a lot of enemies in his own time because of his work: the general mathematical community doubted the relevance of what they called "Cantor's paradise" of unheard-of infinite sizes.

A nice technical feature of Cantor's idea is that it avoids the need for a definition of what the "size" of an infinite set might be—all it does is compare "sizes."

Warning: We haven't, and won't, define what the "size" of an infinite set is. The definition of infinite "sizes" requires the definition of some infinite sets called *ordinals* with special well-ordering properties. The theory of ordinals requires getting deeper into technical set theory than we want to go, and we can get by just fine without defining infinite sizes. All we need are the “as big as” and “same size” relations, surj and bij, between sets.

But there’s something else to watch out for: we’ve referred to surj as an “as big as” relation and bij as a “same size” relation on sets. Of course, most of the “as big as” and “same size” properties of surj and bij on finite sets do carry over to infinite sets, but *some important ones don’t*—as we’re about to show. So you have to be careful: don’t assume that surj has any particular “as big as” property on *infinite* sets until it’s been proved.

Let’s begin with some familiar properties of the “as big as” and “same size” relations on finite sets that do carry over exactly to infinite sets:

Lemma 7.1.3. *For any sets, A, B, C,*

1. A surj B *iff* B inj A.
2. *If* A surj B *and* B surj C*, then* A surj C.
3. *If* A bij B *and* B bij C*, then* A bij C.
4. A bij B *iff* B bij A.

Part 1. follows from the fact that R has the [≤ 1 out, ≥ 1 in] surjective function property iff R^{-1} has the [≥ 1 out, ≤ 1 in] total, injective property. Part 2. follows from the fact that compositions of surjections are surjections. Parts 3. and 4. follow from the first two parts because R is a bijection iff R and R^{-1} are surjective functions. We’ll leave verification of these facts to Problem 4.22.

Another familiar property of finite sets carries over to infinite sets, but this time some real ingenuity is needed to prove it:

Theorem 7.1.4. *[Schröder-Bernstein] For any sets A, B, if A surj B and B surj A, then A bij B.*

That is, the Schröder-Bernstein Theorem says that if A is at least as big as B and conversely, B is at least as big as A, then A is the same size as B. Phrased this way, you might be tempted to take this theorem for granted, but that would be a mistake. For infinite sets A and B, the Schröder-Bernstein Theorem is actually pretty technical. Just because there is a surjective function $f : A \to B$—which need not be a bijection—and a surjective function $g : B \to A$—which also need not be a bijection—it’s not at all clear that there must be a bijection $e : A \to B$. The idea is to construct e from parts of both f and g. We’ll leave the actual construction to Problem 7.11.

Another familiar set property is that for any two sets, either the first is at least as big as the second, or vice-versa. For finite sets this follows trivially from the Mapping Rule. It's actually still true for infinite sets, but assuming it was obvious would be mistaken again.

Theorem 7.1.5. *For* all *sets* A, B,

$$A \text{ surj } B \quad \text{OR} \quad B \text{ surj } A.$$

Theorem 7.1.5 lets us prove that another basic property of finite sets carries over to infinite ones:

Lemma 7.1.6.

$$A \text{ strict } B \;\text{ AND }\; B \text{ strict } C \tag{7.1}$$

implies

$$A \text{ strict } C$$

for all sets A, B, C.

Proof. (of Lemma 7.1.6)

Suppose 7.1 holds, and assume for the sake of contradiction that NOT(A strict C), which means that A surj C. Now since B strict C, Theorem 7.1.5 lets us conclude that C surj B. So we have

$$A \text{ surj } C \;\text{ AND }\; C \text{ surj } B,$$

and Lemma 7.1.3.2 lets us conclude that A surj B, contradicting the fact that A strict B. ■

We're omitting a proof of Theorem 7.1.5 because proving it involves technical set theory—typically the theory of ordinals again—that we're not going to get into. But since proving Lemma 7.1.6 is the only use we'll make of Theorem 7.1.5, we hope you won't feel cheated not to see a proof.

7.1.1 Infinity is different

A basic property of finite sets that does *not* carry over to infinite sets is that adding something new makes a set bigger. That is, if A is a finite set and $b \notin A$, then $|A \cup \{b\}| = |A| + 1$, and so A and $A \cup \{b\}$ are not the same size. But if A is infinite, then these two sets *are* the same size!

Lemma 7.1.7. *Let A be a set and $b \notin A$. Then A is infinite iff A* bij *$A \cup \{b\}$.*

Proof. Since A is *not* the same size as $A \cup \{b\}$ when A is finite, we only have to show that $A \cup \{b\}$ *is* the same size as A when A is infinite.

That is, we have to find a bijection between $A \cup \{b\}$ and A when A is infinite. Here's how: since A is infinite, it certainly has at least one element; call it a_0. But since A is infinite, it has at least two elements, and one of them must not equal to a_0; call this new element a_1. But since A is infinite, it has at least three elements, one of which must not equal both a_0 and a_1; call this new element a_2. Continuing in this way, we conclude that there is an infinite sequence $a_0, a_1, a_2, \dots, a_n, \dots$ of different elements of A. Now it's easy to define a bijection $e : A \cup \{b\} \to A$:

$$\begin{aligned} e(b) &::= a_0, \\ e(a_n) &::= a_{n+1} && \text{for } n \in \mathbb{N}, \\ e(a) &::= a && \text{for } a \in A - \{b, a_0, a_1, \dots\}. \end{aligned}$$

■

7.1.2 Countable Sets

A set, C, is *countable* iff its elements can be listed in order, that is, the elements in C are precisely the elements in the sequence

$$c_0, c_1, \dots, c_n, \dots .$$

Assuming no repeats in the list, saying that C can be listed in this way is formally the same as saying that the function, $f : \mathbb{N} \to C$ defined by the rule that $f(i) ::= c_i$, is a bijection.

Definition 7.1.8. A set, C, is *countably infinite* iff $\mathbb{N}$ bij C. A set is *countable* iff it is finite or countably infinite.

We can also make an infinite list using just a finite set of elements if we allow repeats. For example, we can list the elements in the three-element set $\{2, 4, 6\}$ as

$$2, 4, 6, 6, 6, \dots .$$

This simple observation leads to an alternative characterization of countable sets that does not make separate cases of finite and infinite sets. Namely, a set C is countable iff there is a list

$$c_0, c_1, \dots, c_n, \dots$$

of the elements of C, possibly with repeats.

Lemma 7.1.9. *A set, C, is countable iff* $\mathbb{N}$ surj C. *In fact, a nonempty set C is countable iff there is a* total *surjective function $g : \mathbb{N} \to C$.*

The proof is left to Problem 7.12.

The most fundamental countably infinite set is the set, $\mathbb{N}$, itself. But the set, $\mathbb{Z}$, of *all* integers is also countably infinite, because the integers can be listed in the order:

$$0, -1, 1, -2, 2, -3, 3, \ldots. \tag{7.2}$$

In this case, there is a simple formula for the nth element of the list (7.2). That is, the bijection $f : \mathbb{N} \to \mathbb{Z}$ such that $f(n)$ is the nth element of the list can be defined as:

$$f(n) ::= \begin{cases} n/2 & \text{if } n \text{ is even,} \\ -(n+1)/2 & \text{if } n \text{ is odd.} \end{cases}$$

There is also a simple way to list all *pairs* of nonnegative integers, which shows that $(\mathbb{N} \times \mathbb{N})$ is also countably infinite (Problem 7.16). From this, it's a small step to reach the conclusion that the set, $\mathbb{Q}^{\geq 0}$, of nonnegative rational numbers is countable. This may be a surprise—after all, the rationals densely fill up the space between integers, and for any two, there's another in between. So it might seem as though you couldn't write out all the rationals in a list, but Problem 7.10 illustrates how to do it. More generally, it is easy to show that countable sets are closed under unions and products (Problems 7.1 and 7.16) which implies the countability of a bunch of familiar sets:

Corollary 7.1.10. *The following sets are countably infinite:*

$$\mathbb{Z}^+, \mathbb{Z}, \mathbb{N} \times \mathbb{N}, \mathbb{Q}^+, \mathbb{Z} \times \mathbb{Z}, \mathbb{Q}.$$

A small modification of the proof of Lemma 7.1.7 shows that countably infinite sets are the "smallest" infinite sets, or more precisely that if A is an infinite set, and B is countable, then A surj B (see Problem 7.9).

Also, since adding one new element to an infinite set doesn't change its size, you can add any *finite* number of elements without changing the size by simply adding one element after another. Something even stronger is true: you can add a *countably* infinite number of new elements to an infinite set and still wind up with just a set of the same size (Problem 7.13).

By the way, it's a common mistake to think that, because you can add any finite number of elements to an infinite set and have a bijection with the original set, that you can also throw in infinitely many new elements. In general it isn't true that just because it's OK to do something any finite number of times, it also OK to do it an infinite number of times. For example, starting from 3, you can increment by 1 any finite number of times, and the result will be some integer greater than or equal to 3. But if you increment an infinite number of times, you don't get an integer at all.

7.1.3 Power sets are strictly bigger

Cantor's astonishing discovery was that *not all infinite sets are the same size*. In particular, he proved that for any set, A, the power set, $\text{pow}(A)$, is "strictly bigger" than A. That is,

Theorem 7.1.11. *[Cantor] For any set, A,*

$$A \text{ strict pow}(A).$$

Proof. To show that A is strictly smaller than $\text{pow}(A)$, we have to show that if g is a function from A to $\text{pow}(A)$, then g is *not* a surjection. To do this, we'll simply find a subset, $A_g \subseteq A$ that is not in the range of g. The idea is, for any element $a \in A$, to look at the set $g(a) \subseteq A$ and ask whether or not a happens to be in $g(a)$. First, define

$$A_g ::= \{a \in A \mid a \notin g(a)\}.$$

A_g is now a well-defined subset of A, which means it is a member of $\text{pow}(A)$. But A_g can't be in the range of g, because if it were, we would have

$$A_g = g(a_0)$$

for some $a_0 \in A$, so by definition of A_g,

$$a \in g(a_0) \quad \text{iff} \quad a \in A_g \quad \text{iff} \quad a \notin g(a)$$

for all $a \in A$. Now letting $a = a_0$ yields the contradiction

$$a_0 \in g(a_0) \quad \text{iff} \quad a_0 \notin g(a_0).$$

So g is not a surjection, because there is an element in the power set of A, specifically the set A_g, that is not in the range of g. ■

Cantor's Theorem immediately implies:

Corollary 7.1.12. $\text{pow}(\mathbb{N})$ *is uncountable.*

The bijection between subsets of an n-element set and the length n bit-strings, $\{0, 1\}^n$, used to prove Theorem 4.5.5, carries over to a bijection between subsets of a countably infinite set and the infinite bit-strings, $\{0, 1\}^\omega$. That is,

$$\text{pow}(\mathbb{N}) \text{ bij } \{0, 1\}^\omega.$$

This immediately implies

Corollary 7.1.13. $\{0, 1\}^\omega$ *is uncountable.*

More Countable and Uncountable Sets

Once we have a few sets we know are countable or uncountable, we can get lots more examples using Lemma 7.1.3. In particular, we can appeal to the following immediate corollary of the Lemma:

Corollary 7.1.14.

(a) If U is an uncountable set and A surj *U, then A is uncountable.*

(b) If C is a countable set and C surj *A, then A is countable.*

For example, now that we know that the set $\{0,1\}^{\omega}$ of infinite bit strings is uncountable, it's a small step to conclude that

Corollary 7.1.15. *The set $\mathbb{R}$ of real numbers is uncountable.*

To prove this, think about the infinite decimal expansion of a real number:

$$\begin{aligned}
\sqrt{2} &= 1.4142\ldots,\\
5 &= 5.000\ldots,\\
1/10 &= 0.1000\ldots,\\
1/3 &= 0.333\ldots,\\
1/9 &= 0.111\ldots,\\
4\,\frac{1}{99} &= 4.010101\ldots.
\end{aligned}$$

Let's map any real number r to the infinite bit string $b(r)$ equal to the sequence of bits in the decimal expansion of r, starting at the decimal point. If the decimal expansion of r happens to contain a digit other than 0 or 1, leave $b(r)$ undefined. For example,

$$\begin{aligned}
b(5) &= 000\ldots,\\
b(1/10) &= 1000\ldots,\\
b(1/9) &= 111\ldots,\\
b(4\,\frac{1}{99}) &= 010101\ldots\\
b(\sqrt{2}), b(1/3) &\text{ are undefined.}
\end{aligned}$$

Now b is a function from real numbers to infinite bit strings.[1] It is not a total function, but it clearly is a surjection. This shows that

$$\mathbb{R} \text{ surj } \{0, 1\}^\omega,$$

and the uncountability of the reals now follows by Corollary 7.1.14.(a).

For another example, let's prove

Corollary 7.1.16. *The set $(\mathbb{Z}^+)^*$ of all finite sequences of positive integers is countable.*

To prove this, think about the prime factorization of a nonnegative integer:

$$\begin{aligned} 20 &= 2^2 \cdot 3^0 \cdot 5^1 \cdot 7^0 \cdot 11^0 \cdot 13^0 \cdots, \\ 6615 &= 2^0 \cdot 3^3 \cdot 5^1 \cdot 7^2 \cdot 11^0 \cdot 13^0 \cdots. \end{aligned}$$

Let's map any nonnegative integer n to the finite sequence $e(n)$ of nonzero exponents in its prime factorization. For example,

$$\begin{aligned} e(20) &= (2, 1), \\ e(6615) &= (3, 1, 2), \\ e(5^{13} \cdot 11^9 \cdot 47^{817} \cdot 103^{44}) &= (13, 9, 817, 44), \\ e(1) &= \lambda, && \text{(the empty string)} \\ e(0) &\text{ is undefined.} \end{aligned}$$

Now e is a function from $\mathbb{N}$ to $(\mathbb{Z}^+)^*$. It is defined on all positive integers, and it clearly is a surjection. This shows that

$$\mathbb{N} \text{ surj } (\mathbb{Z}^+)^*,$$

and the countability of the finite strings of positive integers now follows by Corollary 7.1.14.(b).

[1] Some rational numbers can be expanded in two ways—as an infinite sequence ending in all 0's or as an infinite sequence ending in all 9's. For example,

$$\begin{aligned} 5 &= 5.000\cdots = 4.999\ldots, \\ \frac{1}{10} &= 0.1000\cdots = 0.0999\ldots. \end{aligned}$$

In such cases, define $b(r)$ to be the sequence that ends with all 0's.

Larger Infinities

There are lots of different sizes of infinite sets. For example, starting with the infinite set, $\mathbb{N}$, of nonnegative integers, we can build the infinite sequence of sets

$$\mathbb{N} \text{ strict } \text{pow}(\mathbb{N}) \text{ strict } \text{pow}(\text{pow}(\mathbb{N})) \text{ strict } \text{pow}(\text{pow}(\text{pow}(\mathbb{N}))) \text{ strict } \ldots.$$

By Cantor’s Theorem 7.1.11, each of these sets is strictly bigger than all the preceding ones. But that’s not all: the union of all the sets in the sequence is strictly bigger than each set in the sequence (see Problem 7.23). In this way you can keep going indefinitely, building “bigger” infinities all the way.

7.1.4 Diagonal Argument

Theorem 7.1.11 and similar proofs are collectively known as “diagonal arguments” because of a more intuitive version of the proof described in terms of on an infinite square array. Namely, suppose there was a bijection between $\mathbb{N}$ and $\{0, 1\}^\omega$. If such a relation existed, we would be able to display it as a list of the infinite bit strings in some countable order or another. Once we’d found a viable way to organize this list, any given string in $\{0, 1\}^\omega$ would appear in a finite number of steps, just as any integer you can name will show up a finite number of steps from 0. This hypothetical list would look something like the one below, extending to infinity both vertically and horizontally:

$$\boxed{\begin{array}{cccccccccc}
A_0 & = & 1 & 0 & 0 & 0 & 1 & 1 & \cdots \\
A_1 & = & 0 & 1 & 1 & 1 & 0 & 1 & \cdots \\
A_2 & = & 1 & 1 & 1 & 1 & 1 & 1 & \cdots \\
A_3 & = & 0 & 1 & 0 & 0 & 1 & 0 & \cdots \\
A_4 & = & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\
A_5 & = & 1 & 0 & 0 & 1 & 1 & 1 & \cdots \\
\vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}}$$

But now we can exhibit a sequence that’s missing from our allegedly complete list of all the sequences. Look at the diagonal in our sample list:

$$\boxed{\begin{array}{cccccccccc}
A_0 & = & \mathbf{1} & 0 & 0 & 0 & 1 & 1 & \cdots \\
A_1 & = & 0 & \mathbf{1} & 1 & 1 & 0 & 1 & \cdots \\
A_2 & = & 1 & 1 & \mathbf{1} & 1 & 1 & 1 & \cdots \\
A_3 & = & 0 & 1 & 0 & \mathbf{0} & 1 & 0 & \cdots \\
A_4 & = & 0 & 0 & 1 & 0 & \mathbf{0} & 0 & \cdots \\
A_5 & = & 1 & 0 & 0 & 1 & 1 & \mathbf{1} & \cdots \\
\vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}}$$

Here is why the diagonal argument has its name: we can form a sequence D consisting of the bits on the diagonal.

$$D = \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad \cdots,$$

Then, we can form another sequence by switching the **1**'s and **0**'s along the diagonal. Call this sequence C:

$$C = \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad \cdots.$$

Now if nth term of A_n is **1** then the nth term of C is **0**, and *vice versa*, which guarantees that C differs from A_n. In other words, C has at least one bit different from *every* sequence on our list. So C is an element of $\{0, 1\}^\omega$ that does not appear in our list—our list can't be complete!

This diagonal sequence C corresponds to the set $\{a \in A \mid a \notin g(a)\}$ in the proof of Theorem 7.1.11. Both are defined in terms of a countable subset of the uncountable infinity in a way that excludes them from that subset, thereby proving that no countable subset can be as big as the uncountable set.

7.2 The Halting Problem

Although towers of larger and larger infinite sets are at best a romantic concern for a computer scientist, the *reasoning* that leads to these conclusions plays a critical role in the theory of computation. Diagonal arguments are used to show that lots of problems can't be solved by computation, and there is no getting around it.

This story begins with a reminder that having procedures operate on programs is a basic part of computer science technology. For example, *compilation* refers to taking any given program text written in some "high level" programming language like Java, C++, Python, ..., and then generating a program of low-level instructions that does the same thing but is targeted to run well on available hardware. Similarly, *interpreters* or *virtual machines* are procedures that take a program text designed to be run on one kind of computer and simulate it on another kind of computer. Routine features of compilers involve "type-checking" programs to ensure that certain kinds of run-time errors won't happen, and "optimizing" the generated programs so they run faster or use less memory.

The fundamental thing that just can't be done by computation is a *perfect* job of type-checking, optimizing, or any kind of analysis of the overall run time behavior of programs. In this section, we'll illustrate this with a basic example known as the *Halting Problem*. The general Halting Problem for some programming language is, given an arbitrary program, to determine whether the program will run forever if it is not interrupted. If the program does not run forever, it is said to halt. Real programs may halt in many ways, for example, by returning some final value, aborting with some kind of error, or by awaiting user input. But it's easy to detect when any given program will halt: just run it on a virtual machine and wait till it stops. The problem comes when the given program does *not* halt—you may wind up waiting indefinitely without realizing that the wait is fruitless. So how could you detect that the program does *not* halt? We will use a diagonal argument to prove that if an analysis program tries to recognize the non-halting programs, it is bound to give wrong answers, or no answers, for an infinite number of the programs it is supposed to be able to analyze!

To be precise about this, let's call a programming procedure—written in your favorite programming language—a *string procedure* when it is applicable to strings over a standard alphabet—say, the 256 character ASCII alphabet. As a simple example, you might think about how to write a string procedure that halts precisely when it is applied to a *double letter* ASCII string, namely, a string in which every character occurs twice in a row. For example, `aaCC33`, and `zz++ccBB` are double letter strings, but `aa;bb`, `b33`, and `AAAAA` are not.

We'll call a set of strings *recognizable* if there is a string procedure that halts when it is applied to any string in that set and does not halt when applied to any string not in the set. For example, we've just agreed that the set of double letter strings is recognizable.

Let ASCII* be the set of (finite) strings of ASCII characters. There is no harm in assuming that every program can be written using only the ASCII characters; they usually are. When a string $s \in \text{ASCII}^*$ is actually the ASCII description of some string procedure, we'll refer to that string procedure as P_s. You can think of P_s as the result of compiling s.[2] It's technically helpful to treat *every* ASCII string as a program for a string procedure. So when a string $s \in \text{ASCII}^*$ doesn't parse as a proper string procedure, we'll define P_s to be some default string procedure—say one that never halts on any input.

Focusing just on string procedures, the general Halting Problem is to decide, given strings s and t, whether or not the procedure P_s halts when applied to t. We'll show that the general problem can't be solved by showing that a special case can't be solved, namely, whether or not P_s applied to s halts. So, let's define

[2] The string, $s \in \text{ASCII}^*$, and the procedure, P_s, have to be distinguished to avoid a type error: you can't apply a string to string. For example, let s be the string that you wrote as your program to recognize the double letter strings. Applying s to a string argument, say `aabbccdd`, should throw a type exception; what you need to do is compile s to the procedure P_s and then apply P_s to `aabbccdd`.

Definition 7.2.1.

$$\text{No-halt} ::= \{s \in \text{ASCII}^* \mid P_s \text{ applied to } s \text{ does not halt}\}. \qquad (7.3)$$

We're going to prove

Theorem 7.2.2. *No-halt is not recognizable.*

We'll use an argument just like Cantor's in the proof of Theorem 7.1.11.

Proof. For any string $s \in \text{ASCII}^*$, let $f(s)$ be the set of strings recognized by P_s:

$$f(s) ::= \{t \in \text{ASCII}^* \mid P_s \text{ halts when applied to } t\}.$$

By convention, we associated a string procedure, P_s, with every string, $s \in \text{ASCII}^*$, which makes f a total function, and by definition,

$$s \in \text{No-halt IFF } s \notin f(s), \qquad (7.4)$$

for all strings, $s \in \text{ASCII}^*$.

Now suppose to the contrary that No-halt was recognizable. This means there is some procedure P_{s_0} that recognizes No-halt, which is the same as saying that

$$\text{No-halt} = f(s_0).$$

Combined with (7.4), we get

$$s \in f(s_0) \quad \text{iff} \quad s \notin f(s) \qquad (7.5)$$

for all $s \in \text{ASCII}^*$. Now letting $s = s_0$ in (7.5) yields the immediate contradiction

$$s_0 \in f(s_0) \quad \text{iff} \quad s_0 \notin f(s_0).$$

This contradiction implies that No-halt cannot be recognized by any string procedure. ■

So that does it: it's logically impossible for programs in any particular language to solve just this special case of the general Halting Problem for programs in that language. And having proved that it's impossible to have a procedure that figures out whether an arbitrary program halts, it's easy to show that it's impossible to have a procedure that is a perfect recognizer for *any* overall run time property.[3]

[3] The weasel word "overall" creeps in here to rule out some run time properties that are easy to recognize because they depend only on part of the run time behavior. For example, the set of programs that halt after executing at most 100 instructions is recognizable.

For example, most compilers do "static" type-checking at compile time to ensure that programs won't make run-time type errors. A program that type-checks is guaranteed not to cause a run-time type-error. But since it's impossible to recognize perfectly when programs won't cause type-errors, it follows that the type-checker must be rejecting programs that really wouldn't cause a type-error. The conclusion is that no type-checker is perfect—you can always do better!

It's a different story if we think about the *practical* possibility of writing programming analyzers. The fact that it's logically impossible to analyze perfectly arbitrary programs does not mean that you can't do a very good job analyzing interesting programs that come up in practice. In fact, these "interesting" programs are commonly *intended* to be analyzable in order to confirm that they do what they're supposed to do.

In the end, it's not clear how much of a hurdle this theoretical limitation implies in practice. But the theory does provide some perspective on claims about general analysis methods for programs. The theory tells us that people who make such claims either

- are exaggerating the power (if any) of their methods, perhaps to make a sale or get a grant, or
- are trying to keep things simple by not going into technical limitations they're aware of, or
- perhaps most commonly, are so excited about some useful practical successes of their methods that they haven't bothered to think about the limitations which must be there.

So from now on, if you hear people making claims about having general program analysis/verification/optimization methods, you'll know they can't be telling the whole story.

One more important point: there's no hope of getting around this by switching programming languages. Our proof covered programs written in some given programming language like Java, for example, and concluded that no Java program can perfectly analyze all Java programs. Could there be a C++ analysis procedure that successfully takes on all Java programs? After all, C++ does allow more intimate manipulation of computer memory than Java does. But there is no loophole here: it's possible to write a virtual machine for C++ in Java, so if there were a C++ procedure that analyzed Java programs, the Java virtual machine would be able to do it too, and that's impossible. These logical limitations on the power of computation apply no matter what kinds of programs or computers you use.

7.3 The Logic of Sets

7.3.1 Russell's Paradox

Reasoning naively about sets turns out to be risky. In fact, one of the earliest attempts to come up with precise axioms for sets in the late nineteenth century by the logician Gotlob Frege, was shot down by a three line argument known as *Russell's Paradox*[4] which reasons in nearly the same way as the proof of Cantor's Theorem 7.1.11. This was an astonishing blow to efforts to provide an axiomatic foundation for mathematics:

Russell's Paradox

Let S be a variable ranging over all sets, and define

$$W ::= \{S \mid S \not\in S\}.$$

So by definition,

$$S \in W \text{ iff } S \not\in S,$$

for every set S. In particular, we can let S be W, and obtain the contradictory result that

$$W \in W \text{ iff } W \not\in W.$$

The simplest reasoning about sets crashes mathematics! Russell and his colleague Whitehead spent years trying to develop a set theory that was not contradictory, but would still do the job of serving as a solid logical foundation for all of mathematics.

Actually, a way out of the paradox was clear to Russell and others at the time: *it's unjustified to assume that W is a set*. The step in the proof where we let S be W has no justification, because S ranges over sets, and W might not be a set. In fact, the paradox implies that W had better not be a set!

[4] Bertrand Russell was a mathematician/logician at Cambridge University at the turn of the Twentieth Century. He reported that when he felt too old to do mathematics, he began to study and write about philosophy, and when he was no longer smart enough to do philosophy, he began writing about politics. He was jailed as a conscientious objector during World War I. For his extensive philosophical and political writing, he won a Nobel Prize for Literature.

But denying that W is a set means we must *reject* the very natural axiom that every mathematically well-defined collection of sets is actually a set. The problem faced by Frege, Russell and their fellow logicians was how to specify *which* well-defined collections are sets. Russell and his Cambridge University colleague Whitehead immediately went to work on this problem. They spent a dozen years developing a huge new axiom system in an even huger monograph called *Principia Mathematica*, but for all intents and purposes, their approach failed. It was so cumbersome no one ever used it, and it was subsumed by a much simpler, and now widely accepted, axiomatization of set theory by the logicians Zermelo and Fraenkel.

7.3.2 The ZFC Axioms for Sets

A *formula of set theory*[5] is a predicate formula that only uses the predicates "$x = y$" and "$x \in y$." The domain of discourse is the collection of sets, and "$x \in y$" is interpreted to mean that x and y are variables that range over sets, and x is one of the elements in y.

It's generally agreed that, using some simple logical deduction rules, essentially all of mathematics can be derived from some formulas of set theory called the Axioms of *Zermelo-Fraenkel Set Theory* with Choice (ZFC).

For example, since x is a subset of y iff every element of x is also an element of y, here's how we can express x being a subset of y with a formula of set theory:

$$(x \subseteq y) ::= \forall z.\,(z \in x \text{ IMPLIES } z \in y). \tag{7.6}$$

Now we can express formulas of set theory using "$x \subseteq y$" as an abbreviation for formula (7.6).

We're *not* going to be studying the axioms of ZFC in this text, but we thought you might like to see them—and while you're at it, get some practice reading quantified formulas:

Extensionality. Two sets are equal if they have the same members.

$$(\forall z.\ z \in x \text{ IFF } z \in y) \text{ IMPLIES } x = y.$$

Pairing. For any two sets x and y, there is a set, $\{x, y\}$, with x and y as its only elements:

$$\forall x, y.\ \exists u.\ \forall z.\ [z \in u \text{ IFF } (z = x \text{ OR } z = y)]$$

[5] Technically this is called a *first-order predicate formula* of set theory

Union. The union, u, of a collection, z, of sets is also a set:

$$\forall z.\, \exists u.\, \forall x.\, (\exists y.\ x \in y \text{ AND } y \in z) \text{ IFF } x \in u.$$

Infinity. There is an infinite set. Specifically, there is a nonempty set, x, such that for any set $y \in x$, the set $\{y\}$ is also a member of x.

Subset. Given any set, x, and any definable property of sets, there is a set containing precisely those elements $y \in x$ that have the property.

$$\forall x.\, \exists z.\, \forall y.\, y \in z \text{ IFF } [y \in x \text{ AND } \phi(y)]$$

where $\phi(y)$ is any assertion about y definable in the notation of set theory.

Power Set. All the subsets of a set form another set:

$$\forall x.\, \exists p.\, \forall u.\, u \subseteq x \text{ IFF } u \in p.$$

Replacement. Suppose a formula, ϕ, of set theory defines the graph of a function, that is,

$$\forall x, y, z.\, [\phi(x, y) \text{ AND } \phi(x, z)] \text{ IMPLIES } y = z.$$

Then the image of any set, s, under that function is also a set, t. Namely,

$$\forall s\, \exists t\, \forall y.\, [\exists x.\, \phi(x, y) \text{ IFF } y \in t].$$

Foundation. There cannot be an infinite sequence

$$\cdots \in x_n \in \cdots \in x_1 \in x_0$$

of sets each of which is a member of the previous one. This is equivalent to saying every nonempty set has a “member-minimal” element. Namely, define

$$\text{member-minimal}(m, x) ::= [m \in x \text{ AND } \forall y \in x.\, y \notin m].$$

Then the foundation axiom is

$$\forall x.\, x \neq \emptyset \text{ IMPLIES } \exists m.\, \text{member-minimal}(m, x).$$

Choice. Given a set, s, whose members are nonempty sets no two of which have any element in common, then there is a set, c, consisting of exactly one element from each set in s. The formula is given in Problem 7.28.

7.3.3 Avoiding Russell's Paradox

These modern ZFC axioms for set theory are much simpler than the system Russell and Whitehead first came up with to avoid paradox. In fact, the ZFC axioms are as simple and intuitive as Frege's original axioms, with one technical addition: the Foundation axiom. Foundation captures the intuitive idea that sets must be built up from "simpler" sets in certain standard ways. And in particular, Foundation implies that no set is ever a member of itself. So the modern resolution of Russell's paradox goes as follows: since $S \not\in S$ for all sets S, it follows that W, defined above, contains every set. This means W can't be a set—or it would be a member of itself.

7.4 Does All This Really Work?

So this is where mainstream mathematics stands today: there is a handful of ZFC axioms from which virtually everything else in mathematics can be logically derived. This sounds like a rosy situation, but there are several dark clouds, suggesting that the essence of truth in mathematics is not completely resolved.

- The ZFC axioms weren't etched in stone by God. Instead, they were mostly made up by Zermelo, who may have been a brilliant logician, but was also a fallible human being—probably some days he forgot his house keys. So maybe Zermelo, just like Frege, didn't get his axioms right and will be shot down by some successor to Russell who will use his axioms to prove a proposition P and its negation $\overline{P}$. Then math as we understand it would be broken—this may sound crazy, but it has happened before.

  In fact, while there is broad agreement that the ZFC axioms are capable of proving all of standard mathematics, the axioms have some further consequences that sound paradoxical. For example, the Banach-Tarski Theorem says that, as a consequence of the *axiom of choice*, a solid ball can be divided into six pieces and then the pieces can be rigidly rearranged to give *two* solid balls of the same size as the original!

- Some basic questions about the nature of sets remain unresolved. For example, Cantor raised the question whether there is a set whose size is strictly between the smallest infinite set, $\mathbb{N}$ (see Problem 7.9), and the strictly larger set, $\text{pow}(\mathbb{N})$? Cantor guessed not:

Cantor’s *Contiuum Hypothesis*: There is no set, A, such that

$$\mathbb{N} \text{ strict } A \text{ strict } \text{pow}(\mathbb{N}).$$

The Continuum Hypothesis remains an open problem a century later. Its difficulty arises from one of the deepest results in modern Set Theory—discovered in part by Gödel in the 1930’s and Paul Cohen in the 1960’s—namely, the ZFC axioms are not sufficient to settle the Continuum Hypothesis: there are two collections of sets, each obeying the laws of ZFC, and in one collection the Continuum Hypothesis is true, and in the other it is false. Until a mathematician with a deep understanding of sets can extend ZFC with persuasive new axioms, the Continuum Hypothesis will remain undecided.

- But even if we use more or different axioms about sets, there are some unavoidable problems. In the 1930’s, Gödel proved that, assuming that an axiom system like ZFC is consistent—meaning you can’t prove both P and $\overline{P}$ for any proposition, P—then the very proposition that the system is consistent (which is not too hard to express as a logical formula) cannot be proved in the system. In other words, no consistent system is strong enough to verify itself.

7.4.1 Large Infinities in Computer Science

If the romance of different-size infinities and continuum hypotheses doesn’t appeal to you, not knowing about them is not going to limit you as a computer scientist. These abstract issues about infinite sets rarely come up in mainstream mathematics, and they don’t come up at all in computer science, where the focus is generally on “countable,” and often just finite, sets. In practice, only logicians and set theorists have to worry about collections that are “too big” to be sets. That’s part of the reason that the 19th century mathematical community made jokes about “Cantor’s paradise” of obscure infinities. But the challenge of reasoning correctly about this far-out stuff led directly to the profound discoveries about the logical limits of computation described in Section 7.2, and that really is something every computer scientist should understand.

Problems for Section 7.1

Practice Problems

Problem 7.1.
Prove that if A and B are countable sets, then so is $A \cup B$.

Problem 7.2.
Show that the set $\{0,1\}^*$ of finite binary strings is countable.

Problem 7.3.
Describe an example of two **un**countable sets A and B such that there is no bijection between A and B.

Problem 7.4.
Prove that if there is a total injective ($[\geq 1$ out, ≤ 1 in$]$) relation from $S \to \mathbb{N}$, then S is countable.

Problem 7.5.
For each of the following sets, indicate whether it is finite, countably infinite, or uncountable.

1. The set of solutions to the equation $x^3 - x = -0.1$.
2. The set of natural numbers $\mathbb{N}$.
3. The set of rational numbers $\mathbb{Q}$.
4. The set of real numbers $\mathbb{R}$.
5. The set of integers $\mathbb{Z}$.
6. The set of complex numbers $\mathbb{C}$.
7. The set of words in the English language no more than 20 characters long.
8. The powerset of the set of all possible bijections from $\{1, 2, \ldots, 10\}$ to itself.
9. An infinite set S with the property that there exists a total surjective function $f : \mathbb{N} \to S$.
10. A set $A \cup B$ where A is countable and B is uncountable.

Problem 7.6.
Circle the correct completions (there may be more than one)
A strict $\mathbb{N}$ IFF ...

- $|A|$ is undefined.
- A is countably infinite.
- A is uncountable.
- A is finite.
- $\mathbb{N}$ surj A.
- $\forall n \in \mathbb{N}, |A| \leq n$.
- $\forall n \in \mathbb{N}, |A| \geq n$.
- $\exists n \in \mathbb{N}. |A| \leq n$.
- $\exists n \in \mathbb{N}. |A| < n$.

Problem 7.7.

Let A to be some infinite set and B to be some countable set. We know from Lemma 7.1.7 that

$$A \text{ bij } (A \cup \{b_0\})$$

for any element $b_0 \in B$. An easy induction implies that

$$A \text{ bij } (A \cup \{b_0, b_1, \dots, b_n\}) \tag{7.7}$$

for any finite subset $\{b_0, b_1, \dots, b_n\} \subset B$.

Students sometimes think that (7.7) shows that A bij $(A \cup B)$. Now it's true that A bij $(A \cup B)$ for all such A and B for any countable set B (Problem 7.13), but the facts above do not prove it.

To explain this, let's say that a predicate $P(C)$ is *finitely discontinuous* when $P(A \cup F)$ is true for every *finite* subset $F \subset B$, but $P(A \cup B)$ is false. The hole in the claim that (7.7) implies A bij $(A \cup B)$ is the assumption (without proof) that the predicate

$$P_0(C) ::= [A \text{ bij } C]$$

is not finitely discontinuous. This assumption about P_0 is correct, but it's not completely obvious and takes some proving.

To illustrate this point, let A be the nonnegative integers and B be the nonnegative rational numbers, and remember that both A and B are countably infinite. Some of the predicates $P(C)$ below are finitely **d**iscontinuous and some are **n**ot. Indicate which is which.

1. C is finite.

2. C is countable.

3. C is uncountable.

4. C contains only finitely many non-integers.

5. C contains the rational number 2/3.

6. There is a maximum non-integer in C.

7. There is an $\epsilon > 0$ such that any two elements of C are ϵ apart.

8. C is countable.

9. C is uncountable.

10. C has no infinite decreasing sequence $c_0 > c_1 > \cdots$.

11. Every nonempty subset of C has a minimum element.

12. C has a maximum element.

13. C has a minimum element.

Class Problems

Problem 7.8.
Show that the set $\mathbb{N}^*$ of finite sequences of nonnegative integers is countable.

Problem 7.9. **(a)** Several students felt the proof of Lemma 7.1.7 was worrisome, if not circular. What do you think?

(b) Use the proof of Lemma 7.1.7 to show that if A is an infinite set, then A surj $\mathbb{N}$, that is, every infinite set is "as big as" the set of nonnegative integers.

Problem 7.10.
The rational numbers fill the space between integers, so a first thought is that there must be more of them than the integers, but it's not true. In this problem you'll show that there are the same number of positive rationals as positive integers. That is, the positive rationals are countable.

(a) Define a bijection between the set, $\mathbb{Z}^+$, of positive integers, and the set, $(\mathbb{Z}^+ \times \mathbb{Z}^+)$, of all pairs of positive integers:

$$\begin{array}{l}
(1,1),(1,2),(1,3),(1,4),(1,5),\ldots \\
(2,1),(2,2),(2,3),(2,4),(2,5),\ldots \\
(3,1),(3,2),(3,3),(3,4),(3,5),\ldots \\
(4,1),(4,2),(4,3),(4,4),(4,5),\ldots \\
(5,1),(5,2),(5,3),(5,4),(5,5),\ldots \\
\qquad\vdots
\end{array}$$

(b) Conclude that the set, $\mathbb{Q}^+$, of all positive rational numbers is countable.

Problem 7.11.
This problem provides a proof of the [Schröder-Bernstein] Theorem:

$$\text{If } A \text{ surj } B \text{ and } B \text{ surj } A, \text{ then } A \text{ bij } B. \tag{7.8}$$

(a) It is OK to assume that A and B are disjoint. Why?

(b) Explain why there are total injective functions $f : A \to B$, and $g : B \to A$.

Picturing the diagrams for f and g, there is *exactly one* arrow *out* of each element —a left-to-right f-arrow if the element is in A and a right-to-left g-arrow if the element is in B. This is because f and g are total functions. Also, there is *at most one* arrow *into* any element, because f and g are injections.

So starting at any element, there is a unique and unending path of arrows going forwards. There is also a unique path of arrows going backwards, which might be unending, or might end at an element that has no arrow into it. These paths are completely separate: if two ran into each other, there would be two arrows into the element where they ran together.

This divides all the elements into separate paths of four kinds:

i. paths that are infinite in both directions,

ii. paths that are infinite going forwards starting from some element of A.

iii. paths that are infinite going forwards starting from some element of B.

iv. paths that are unending but finite.

(c) What do the paths of the last type (iv) look like?

(d) Show that for each type of path, either

- the f-arrows define a bijection between the A and B elements on the path, or
- the g-arrows define a bijection between B and A elements on the path, or
- both sets of arrows define bijections.

For which kinds of paths do both sets of arrows define bijections?

(e) Explain how to piece these bijections together to prove that A and B are the same size.

Problem 7.12. **(a)** Prove that if a nonempty set, C, is countable, then there is a *total* surjective function $f : \mathbb{N} \to C$.

(b) Conversely, suppose that $\mathbb{N}$ surj D, that is, there is a not necessarily total surjective function $f : \mathbb{N} D$. Prove that D is countable.

Homework Problems

Problem 7.13.
Prove that if A is an infinite set and B is a countably infinite set that has no elements in common with A, then

$$A \text{ bij } (A \cup B).$$

Reminder: You may assume any of the results from class, MITx, or the text as long as you state them explicitly.

Problem 7.14.
In this problem you will prove a fact that may surprise you—or make you even more convinced that set theory is nonsense: the half-open unit interval is actually the "*same size*" as the nonnegative quadrant of the real plane![6] Namely, there is a bijection from $(0, 1]$ to $[0, \infty) \times [0, \infty)$.

(a) Describe a bijection from $(0, 1]$ to $[0, \infty)$.

Hint: $1/x$ almost works.

(b) An infinite sequence of the decimal digits $\{\texttt{0}, \texttt{1}, \dots, \texttt{9}\}$ will be called *long* if it does not end with all 0's. An equivalent way to say this is that a long sequence

[6] The half-open unit interval, $(0, 1]$, is $\{r \in \mathbb{R} \mid 0 < r \leq 1\}$. Similarly, $[0, \infty) ::= \{r \in \mathbb{R} \mid r \geq 0\}$.

is one that has infinitely many occurrences of nonzero digits. Let L be the set of all such long sequences. Describe a bijection from L to the half-open real interval $(0, 1]$.

Hint: Put a decimal point at the beginning of the sequence.

(c) Describe a surjective function from L to L^2 that involves alternating digits from two long sequences. *Hint:* The surjection need not be total.

(d) Prove the following lemma and use it to conclude that there is a bijection from L^2 to $(0, 1]^2$.

Lemma 7.4.1. *Let A and B be nonempty sets. If there is a bijection from A to B, then there is also a bijection from $A \times A$ to $B \times B$.*

(e) Conclude from the previous parts that there is a surjection from $(0, 1]$ to $(0, 1]^2$. Then appeal to the Schröder-Bernstein Theorem to show that there is actually a bijection from $(0, 1]$ to $(0, 1]^2$.

(f) Complete the proof that there is a bijection from $(0, 1]$ to $[0, \infty)^2$.

Exam Problems

Problem 7.15.
Prove that if $A_0, A_1, \ldots, A_n, \ldots$ is an infinite sequence of countable sets, then so is

$$\bigcup_{n=0}^{\infty} A_n$$

Problem 7.16.
Let A and B be countably infinite sets:

$$\begin{aligned} A &= \{a_0, a_1, a_2, a_3, \ldots\} \\ B &= \{b_0, b_1, b_2, b_3, \ldots\} \end{aligned}$$

Show that their product, $A \times B$, is also a countable set by showing how to list the elements of $A \times B$. You need only show enough of the initial terms in your sequence to make the pattern clear—a half dozen or so terms usually suffice.

Problem 7.17. (a) Prove that if A and B are countable sets, then so is $A \cup B$.

(b) Prove that if C is a countable set and D is infinite, then there is a bijection between D and $C \cup D$.

Problem 7.18.

Let $\{0,1\}^*$ be the set of finite binary sequences, $\{0,1\}^\omega$ be the set of infinite binary sequences, and F be the set of sequences in $\{0,1\}^\omega$ that contain only a **f**inite number of occurrences of 1’s.

(a) Describe a simple surjective function from $\{0,1\}^*$ to F.

(b) The set $\overline{F} ::= \{0,1\}^\omega - F$ consists of all the infinite binary sequences with *infinitely* many 1’s. Use the previous problem part to prove that $\overline{F}$ is uncountable.

Hint: We know that $\{0,1\}^*$ is countable and $\{0,1\}^\omega$ is not.

Problem 7.19.

Let $\{0,1\}^\omega$ be the set of infinite binary strings, and let $B \subset \{0,1\}^\omega$ be the set of infinite binary strings containing infinitely many occurrences of 1’s. Prove that B is uncountable. (We have already shown that $\{0,1\}^\omega$ is uncountable.)

Hint: Define a suitable function from $\{0,1\}^\omega$ to B.

Problem 7.20.

A real number is called *quadratic* when it is a root of a degree two polynomial with integer coefficients. Explain why there are only countably many quadratic reals.

Problem 7.21.

Describe which of the following sets have bijections between them:

$\mathbb{Z}$ (integers),
$\mathbb{C}$ (complex numbers),
$\text{pow}(\mathbb{Z})$ (all subsets of integers),
$\text{pow}(\text{pow}(\emptyset))$,
$\{0,1\}^\omega$ (infinite binary sequences)
$\text{pow}(\{\mathbf{T},\mathbf{F}\})$,
$\mathbb{R}$ (real numbers),
$\mathbb{Q}$ (rational numbers),
$\text{pow}(\emptyset)$,
$\{0,1\}^*$ (finite binary sequences),
$\{\mathbf{T},\mathbf{F}\}$ (truth values)
$\text{pow}(\{0,1\}^\omega)$

Problems for Section 7.2

Class Problems

Problem 7.22.
Let $\mathbb{N}^{\omega}$ be the set of infinite sequences of nonnegative integers. For example, some sequences of this kind are:

$$\begin{gathered}(0, 1, 2, 3, 4, \ldots),\\ (2, 3, 5, 7, 11, \ldots),\\ (3, 1, 4, 5, 9, \ldots).\end{gathered}$$

Prove that this set of sequences is uncountable.

Problem 7.23.
There are lots of different sizes of infinite sets. For example, starting with the infinite set, $\mathbb{N}$, of nonnegative integers, we can build the infinite sequence of sets

$$\mathbb{N} \text{ strict } \mathrm{pow}(\mathbb{N}) \text{ strict } \mathrm{pow}(\mathrm{pow}(\mathbb{N})) \text{ strict } \mathrm{pow}(\mathrm{pow}(\mathrm{pow}(\mathbb{N}))) \text{ strict } \ldots.$$

where each set is “strictly smaller” than the next one by Theorem 7.1.11. Let $\mathrm{pow}^n(\mathbb{N})$ be the nth set in the sequence, and

$$U ::= \bigcup_{n=0}^{\infty} \mathrm{pow}^n(\mathbb{N}).$$

(a) Prove that

$$U \text{ surj } \mathrm{pow}^n(\mathbb{N}), \tag{7.9}$$

for all $n > 0$.

(b) Prove that

$$\mathrm{pow}^n(\mathbb{N}) \text{ strict } U$$

for all $n \in \mathbb{N}$.

Now of course, we could take $U, \mathrm{pow}(U), \mathrm{pow}(\mathrm{pow}(U)), \ldots$ and keep on in this way building still bigger infinities indefinitely.

Problem 7.24.
The method used to prove Cantor’s Theorem that the power set is “bigger” than the

set, leads to many important results in logic and computer science. In this problem we'll apply that idea to describe a set of binary strings that can't be described by ordinary logical formulas. To be provocative, we could say that we will describe an undescribable set of strings!

The following logical formula illustrates how a formula can describe a set of strings. The formula

$$\text{NOT}[\exists y.\, \exists z.\, s = y1z], \qquad (\text{no-1s}(s))$$

where the variables range over the set, $\{0,1\}^*$, of finite binary strings, says that the binary string, s, does not contain a 1.

We'll call such a predicate formula, $G(s)$, about strings a *string formula*, and we'll use the notation strings(G) for the set of binary strings with the property described by G. That is,

$$\text{strings}(G) ::= \{s \in \{0,1\}^* \mid G(s)\}.$$

A set of binary strings is *describable* if it equals strings(G) for some string formula, G. So the set, 0^*, of finite strings of 0's is describable because it equals strings(no-1s).[7]

The idea of representing data in binary is a no-brainer for a computer scientist, so it won't be a stretch to agree that any string formula can be represented by a binary string. We'll use the notation G_x for the string formula with binary representation $x \in \{0,1\}^*$. The details of the representation don't matter, except that there ought to be a display procedure that can actually display G_x given x.

Standard binary representations of formulas are often based on character-by-character translation into binary, which means that only a sparse set of binary strings actually represent string formulas. It will be technically convenient to have *every* binary string represent some string formula. This is easy to do: tweak the display procedure so it displays some default formula, say no-1s, when it gets a binary string that isn't a standard representation of a string formula. With this tweak, *every* binary string, x, will now represent a string formula, G_x.

Now we have just the kind of situation where a Cantor-style diagonal argument can be applied, namely, we'll ask whether a string describes a property of *itself*! That may sound like a mind-bender, but all we're asking is whether $x \in \text{strings}(G_x)$.

For example, using character-by-character translations of formulas into binary, neither the string 0000 nor the string 10 would be the binary representation of a formula, so the display procedure applied to either of them would display no-1s.

[7] no-1s and similar formulas were examined in Problem 3.25, but it is not necessary to have done that problem to do this one.

That is, $G_{0000} = G_{10} =$ no-1s and so strings$(G_{0000}) =$ strings$(G_{10}) = 0^*$. This means that

$$0000 \in \text{strings}(G_{0000}) \quad \text{and} \quad 10 \notin \text{strings}(G_{10}).$$

Now we are in a position to give a precise mathematical description of an "undescribable" set of binary strings, namely, let

Theorem. *Define*

$$U ::= \{x \in \{0,1\}^* \mid x \notin \text{strings}(G_x)\}. \tag{7.10}$$

The set U is not describable.

Use reasoning similar to Cantor's Theorem 7.1.11 to prove this Theorem.

Homework Problems

Problem 7.25.
For any sets, A, and B, let $[A \to B]$ be the set of total functions from A to B. Prove that if A is not empty and B has more than one element, then NOT(A surj $[A \to B]$).

Hint: Suppose that σ is a function from A to $[A \to B]$ mapping each element $a \in A$ to a function $\sigma_a : A \to B$. Pick any two elements of B; call them 0 and 1. Then define

$$\text{diag}(a) ::= \begin{cases} 0 \text{ if } \sigma_a(a) = 1, \\ 1 \text{ otherwise.} \end{cases}$$

Exam Problems

Problem 7.26.
Let $\{1,2,3\}^\omega$ be the set of infinite sequences containing only the numbers 1, 2, and 3. For example, some sequences of this kind are:

$$(1,1,1,1...),$$
$$(2,2,2,2...),$$
$$(3,2,1,3...).$$

Prove that $\{1,2,3\}^\omega$ is uncountable.

Hint: One approach is to define a surjective function from $\{1,2,3\}^\omega$ to the power set pow$(\mathbb{N})$.

Problems for Section 7.3

Class Problems

Problem 7.27.
Forming a pair (a, b) of items a and b is a mathematical operation that we can safely take for granted. But when we're trying to show how all of mathematics can be reduced to set theory, we need a way to represent the pair (a, b) as a set.

(a) Explain why representing (a, b) by $\{a, b\}$ won't work.

(b) Explain why representing (a, b) by $\{a, \{b\}\}$ won't work either. *Hint:* What pair does $\{\{1\}, \{2\}\}$ represent?

(c) Define

$$\text{pair}(a, b) ::= \{a, \{a, b\}\}.$$

Explain why representing (a, b) as pair(a, b) uniquely determines a and b. *Hint:* Sets can't be indirect members of themselves: $a \in a$ never holds for any set a, and neither can $a \in b \in a$ hold for any b.

Problem 7.28.
The axiom of choice says that if s is a set whose members are nonempty sets that are *pairwise disjoint* —that is no two sets in s have an element in common —then there is a set, c, consisting of exactly one element from each set in s.

In formal logic, we could describe s with the formula,

$$\begin{aligned}&\text{pairwise-disjoint}(s)\\ ::=&\forall x \in s.\, x \neq \emptyset \text{ AND}\\ &\forall x, y \in s.\, x \neq y \text{ IMPLIES } x \cap y = \emptyset.\end{aligned}$$

Similarly we could describe c with the formula

$$\text{choice-set}(c, s) ::= \quad \forall x \in s.\, \exists! z.\, z \in c \cap x.$$

Here “$\exists!\, z$.” is fairly standard notation for “there exists a *unique* z.”

Now we can give the formal definition:

Definition (Axiom of Choice)**.**

$$\forall s.\, \text{pairwise-disjoint}(s) \text{ IMPLIES } \exists c.\, \text{choice-set}(c, s).$$

The only issue here is that set theory is technically supposed to be expressed in terms of *pure* formulas in the language of sets, which means formula that uses only the membership relation, $\in$, propositional connectives, the two quantifies $\forall$ and $\exists$, and variables ranging over all sets. Verify that the axiom of choice can be expressed as a pure formula, by explaining how to replace all impure subformulas above with equivalent pure formulas.

For example, the formula $x = y$ could be replaced with the pure formula $\forall z.\, z \in x$ IFF $z \in y$.

Problem 7.29.
Let $R : A \to A$ be a binary relation on a set, A. If $a_1 \; R \; a_0$, we'll say that a_1 is "R-smaller" than a_0. R is called *well founded* when there is no infinite "R-decreasing" sequence:

$$\cdots R \; a_n \; R \cdots R \; a_1 \; R \; a_0, \tag{7.11}$$

of elements $a_i \in A$.

For example, if $A = \mathbb{N}$ and R is the $<$-relation, then R is well founded because if you keep counting down with nonnegative integers, you eventually get stuck at zero:

$$0 < \cdots < n - 1 < n.$$

But you can keep counting up forever, so the $>$-relation is not well founded:

$$\cdots > n > \cdots > 1 > 0.$$

Also, the $\leq$-relation on $\mathbb{N}$ is not well founded because a *constant* sequence of, say, 2's, gets $\leq$-smaller forever:

$$\cdots \leq 2 \leq \cdots \leq 2 \leq 2.$$

(a) If B is a subset of A, an element $b \in B$ is defined to be *R-minimal in B* iff there is no R-smaller element in B. Prove that $R : A \to A$ is well founded iff every nonempty subset of A has an R-minimal element.

A logic *formula of set theory* has only predicates of the form "$x \in y$" for variables x, y ranging over sets, along with quantifiers and propositional operations. For example,

$$\text{isempty}(x) ::= \forall w.\ \text{NOT}(w \in x)$$

is a formula of set theory that means that "x is empty."

(b) Write a formula, member-minimal(u, v), of set theory that means that u is $\in$-minimal in v.

(c) The Foundation axiom of set theory says that $\in$ is a well founded relation on sets. Express the Foundation axiom as a formula of set theory. You may use "member-minimal" and "isempty" in your formula as abbreviations for the formulas defined above.

(d) Explain why the Foundation axiom implies that no set is a member of itself.

Homework Problems

Problem 7.30. **(a)** Explain how to write a formula, $\text{Subset}_n(x, y_1, y_2, \ldots, y_n)$, of set theory [8] that means $x \subseteq \{y_1, y_2, \ldots, y_n\}$.

(b) Now use the formula Subset_n to write a formula, $\text{Atmost}_n(x)$, of set theory that means that x has at most n elements.

(c) Explain how to write a formula, Exactly_n, of set theory that means that x has exactly n elements. Your formula should only be about twice the length of the formula Atmost_n.

(d) The obvious way to write a formula, $D_n(y_1, \ldots, y_n)$, of set theory that means that $y_1, \ldots, y_n$ are distinct elements is to write an AND of subformulas "$y_i \neq y_j$" for $1 \leq i < j \leq n$. Since there are $n(n-1)/2$ such subformulas, this approach leads to a formula D_n whose length grows proportional to n^2. Describe how to write such a formula $D_n(y_1, \ldots, y_n)$ whose length only grows proportional to n.

Hint: Use Subset_n and Exactly_n.

Exam Problems

Problem 7.31. **(a)** Explain how to write a formula $\text{Members}(p, a, b)$ of set theory[9] that means $p = \{a, b\}$.

Hint: Say that everything in p is either a or b. It's OK to use subformulas of the form "$x = y$," since we can regard "$x = y$" as an abbreviation for a genuine set theory formula.

A *pair* (a, b) is simply a sequence of length two whose first item is a and whose second is b. Sequences are a basic mathematical data type we take for granted, but when we're trying to show how all of mathematics can be reduced to set theory, we need a way to represent the ordered pair (a, b) as a set. One way that will work[10]

[8] See Section 7.3.2.

[9] See Section 7.3.2.

[10] Some similar ways that don't work are described in problem 7.27.

is to represent (a, b) as

$$\text{pair}(a, b) ::= \{a, \{a, b\}\}.$$

(b) Explain how to write a formula Pair(p, a, b), of set theory [11] that means $p = \text{pair}(a, b)$.

Hint: Now it's OK to use subformulas of the form "Members(p, a, b)."

(c) Explain how to write a formula Second(p, b), of set theory that means p is a pair whose second item is b.

Problems for Section 7.4

Homework Problems

Problem 7.32.

For any set x, define next(x) to be the set consisting of all the elements of x, along with x itself:

$$\text{next}(x) ::= x \cup \{x\}.$$

So by definition,

$$x \in \text{next}(x) \text{ and } x \subset \text{next}(x). \tag{7.12}$$

Now we give a recursive definition of a collection, Ord, of sets called *ordinals* that provide a way to count infinite sets. Namely,

Definition.

$$\begin{gathered}\emptyset \in \text{Ord}, \\ \text{if } \nu \in \text{Ord}, \text{ then next}(\nu) \in \text{Ord}, \\ \text{if } S \subset \text{Ord}, \text{ then } \bigcup_{\nu \in S} \nu \in \text{Ord}.\end{gathered}$$

There is a method for proving things about ordinals that follows directly from the way they are defined. Namely, let $P(x)$ be some property of sets. The *Ordinal Induction Rule* says that to prove that $P(\nu)$ is true for all ordinals ν, you need only show two things

- If P holds for all the members of next(x), then it holds for next(x), and
- if P holds for all members of some set S, then it holds for their union.

[11] See Section 7.3.2.

That is:

Rule. ***Ordinal Induction***

$$\frac{\begin{array}{c}\forall x.\, (\forall y \in \mathit{next}(x).\, P(y)) \text{ IMPLIES } P(\mathit{next}(x)), \\ \forall S.\, (\forall x \in S.\, P(x)) \text{ IMPLIES } P(\bigcup_{x \in S} x)\end{array}}{\forall \nu \in \mathit{Ord}.\, P(\nu)}$$

The intuitive justification for the Ordinal Induction Rule is similar to the justification for strong induction. We will accept the soundness of the Ordinal Induction Rule as a basic axiom.

(a) A set x is *closed under membership* if every element of x is also a subset of x, that is

$$\forall y \in x.\, y \subset x.$$

Prove that every ordinal ν is closed under membership.

(b) A sequence

$$\cdots \in \nu_{n+1} \in \nu_n \in \cdots \in \nu_1 \in \nu_0 \tag{7.13}$$

of ordinals ν_i is called a *member-decreasing* sequence starting at ν_0. Use Ordinal Induction to prove that no ordinal starts an infinite member-decreasing sequence.[12]

[12] Do not assume the Foundation Axiom of ZFC (Section 7.3.2) which says that there isn't *any* set that starts an infinite member-decreasing sequence. Even in versions of set theory in which the Foundation Axiom does not hold, there cannot be any infinite member-decreasing sequence of ordinals.