TheoryPrep Readings

May 2020

Selected from Mathematics for Computer Science (Lehman, Leighton, Meyer 2017) and Building Blocks for Computer Science (Fleck 2013)

4.3 Functions

4.3.1 Domains and Images

A *function* assigns an element of one set, called the *domain*, to an element of another set, called the *codomain*. The notation

$$f : A \to B$$

indicates that f is a function with domain A and codomain B. The familiar notation "$f(a) = b$" indicates that f assigns the element $b \in B$ to a. Here b would be called the *value* of f at *argument* a.

Functions are often defined by formulas, as in:

$$f_1(x) ::= \frac{1}{x^2}$$

where x is a real-valued variable, or

$$f_2(y, z) ::= y10yz$$

where y and z range over binary strings, or

$$f_3(x, n) ::= \text{ the length } n \text{ sequence } \underbrace{(x, \ldots, x)}_{n\ x\text{'s}}$$

where n ranges over the nonnegative integers.

A function with a finite domain could be specified by a table that shows the value of the function at each element of the domain. For example, a function $f_4(P, Q)$ where P and Q are propositional variables is specified by:

P	Q	$f_4(P, Q)$
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

Notice that f_4 could also have been described by a formula:

$$f_4(P, Q) ::= [P \text{ IMPLIES } Q].$$

A function might also be defined by a procedure for computing its value at any element of its domain, or by some other kind of specification. For example, define

$f_5(y)$ to be the length of a left to right search of the bits in the binary string y until a `1` appears, so

$$\begin{aligned} f_5(0010) &= 3, \\ f_5(100) &= 1, \\ f_5(0000) &\text{ is undefined.} \end{aligned}$$

Notice that f_5 does not assign a value to any string of just `0`'s. This illustrates an important fact about functions: they need not assign a value to every element in the domain. In fact this came up in our first example $f_1(x) = 1/x^2$, which does not assign a value to 0. So in general, functions may be *partial functions*, meaning that there may be domain elements for which the function is not defined. The set of domain elements for which a function is defined is called the *support* of the function. If a function assigns a value to every element of its domain, that is, its support equals its domain, it is called a *total function*.

It's often useful to find the set of values a function takes when applied to the elements in *a set* of arguments. So if $f : A \to B$, and S is a subset of A, we define $f(S)$ to be the set of all the values that f takes when it is applied to elements of S. That is,

$$f(S) ::= \{b \in B \mid f(s) = b \text{ for some } s \in S\}.$$

For example, if we let $[r, s]$ denote set of numbers in the interval from r to s on the real line, then $f_1([1, 2]) = [1/4, 1]$.

For another example, let's take the "search for a `1`" function f_5. If we let X be the set of binary words which start with an even number of `0`'s followed by a `1`, then $f_5(X)$ would be the odd nonnegative integers.

Applying f to a set S of arguments is referred to as "applying f *pointwise* to S", and the set $f(S)$ is referred to as the *image* of S under f.[4] The set of values that arise from applying f to all possible arguments is called the *range* of f. That is,

$$\text{range}(f) ::= f(\text{domain}(f)).$$

Some authors refer to the codomain as the range of a function, but they shouldn't. The distinction between the range and codomain will be important later in Sections 4.5 when we relate sizes of sets to properties of functions between them.

[4] There is a picky distinction between the function f which applies to elements of A and the function which applies f pointwise to subsets of A, because the domain of f is A, while the domain of pointwise-f is $\text{pow}(A)$. It is usually clear from context whether f or pointwise-f is meant, so there is no harm in overloading the symbol f in this way.

4.3.2 Function Composition

Doing things step by step is a universal idea. Taking a walk is a literal example, but so is cooking from a recipe, executing a computer program, evaluating a formula, and recovering from substance abuse.

Abstractly, taking a step amounts to applying a function, and going step by step corresponds to applying functions one after the other. This is captured by the operation of *composing* functions. Composing the functions f and g means that first f is applied to some argument, x, to produce $f(x)$, and then g is applied to that result to produce $g(f(x))$.

Definition 4.3.1. For functions $f : A \to B$ and $g : B \to C$, the *composition*, $g \circ f$, of g with f is defined to be the function from A to C defined by the rule:

$$(g \circ f)(x) ::= g(f(x)),$$

for all $x \in A$.

Function composition is familiar as a basic concept from elementary calculus, and it plays an equally basic role in discrete mathematics.

4.4 Binary Relations

Binary relations define relations between two objects. For example, "less-than" on the real numbers relates every real number a to a real number b, precisely when $a < b$. Similarly, the subset relation relates a set A to another set B precisely when $A \subseteq B$. A function $f : A \to B$ is a special case of binary relation in which an element $a \in A$ is related to an element $b \in B$ precisely when $b = f(a)$.

In this section we'll define some basic vocabulary and properties of binary relations.

Definition 4.4.1. A *binary relation* R consists of a set A, called the *domain* of R, a set B called the *codomain* of R, and a subset of $A \times B$ called the *graph of* R.

A relation whose domain is A and codomain is B is said to be "between A and B", or "from A to B." As with functions, we write $R : A \to B$ to indicate that R is a relation from A to B. When the domain and codomain are the same set A we simply say the relation is "on A." It's common to use "$a\ R\ b$" to mean that the pair (a, b) is in the graph of R.[5]

[5]Writing the relation or operator symbol between its arguments is called *infix notation*. Infix expressions like "$m < n$" or "$m + n$" are the usual notation used for things like the less-then relation or the addition operation rather than prefix notation like "$< (m, n)$" or "$+(m, n)$."

Notice that Definition 4.4.1 is exactly the same as the definition in Section 4.3 of a *function*, except that it doesn't require the functional condition that, for each domain element a, there is *at most* one pair in the graph whose first coordinate is a. As we said, a function is a special case of a binary relation.

The "in-charge of" relation *Chrg* for MIT in Spring '10 subjects and instructors is a handy example of a binary relation. Its domain Fac is the names of all the MIT faculty and instructional staff, and its codomain is the set SubNums of subject numbers in the Fall '09–Spring '10 MIT subject listing. The graph of *Chrg* contains precisely the pairs of the form

$$(\langle\text{instructor-name}\rangle\,,\langle\text{subject-num}\rangle)$$

such that the faculty member named ⟨instructor-name⟩ is in charge of the subject with number ⟨subject-num⟩ that was offered in Spring '10. So graph(*Chrg*) contains pairs like

```
(T. Eng,       6.UAT)
(G. Freeman,   6.011)
(G. Freeman,   6.UAT)
(G. Freeman,   6.881)
(G. Freeman,   6.882)
(J. Guttag,    6.00)
(A. R. Meyer,  6.042)
(A. R. Meyer,  18.062)
(A. R. Meyer,  6.844)
(T. Leighton,  6.042)
(T. Leighton,  18.062)
    ⋮
```

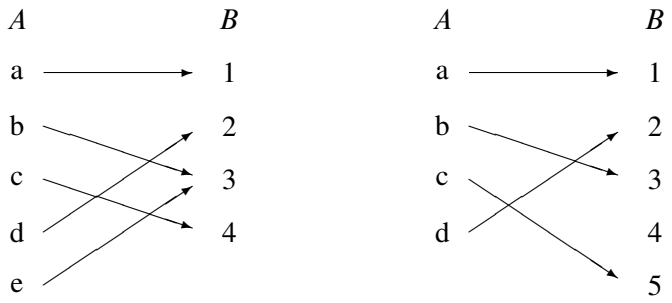
(4.4)

Some subjects in the codomain SubNums do not appear among this list of pairs—that is, they are not in range(*Chrg*). These are the Fall term-only subjects. Similarly, there are instructors in the domain Fac who do not appear in the list because they are not in charge of any Spring term subjects.

4.4.1 Relation Diagrams

Some standard properties of a relation can be visualized in terms of a diagram. The diagram for a binary relation R has points corresponding to the elements of the domain appearing in one column (a very long column if domain(R) is infinite). All the elements of the codomain appear in another column which we'll usually picture as being to the right of the domain column. There is an arrow going from a point a in the left-hand, domain column to a point b in the right-hand, codomain column,

precisely when the corresponding elements are related by R. For example, here are diagrams for two functions:



Being a function is certainly an important property of a binary relation. What it means is that every point in the domain column has *at most one arrow coming out of it*. So we can describe being a function as the "≤ 1 arrow out" property. There are four more standard properties of relations that come up all the time. Here are all five properties defined in terms of arrows:

Definition 4.4.2. A binary relation R is:

- a *function* when it has the [≤ 1 arrow **out**] property.
- *surjective* when it has the [≥ 1 arrows **in**] property. That is, every point in the right-hand, codomain column has at least one arrow pointing to it.
- *total* when it has the [≥ 1 arrows **out**] property.
- *injective* when it has the [≤ 1 arrow **in**] property.
- *bijective* when it has both the [$= 1$ arrow **out**] and the [$= 1$ arrow **in**] property.

From here on, we'll stop mentioning the arrows in these properties and for example, just write [≤ 1 in] instead of [≤ 1 arrows in].

So in the diagrams above, the relation on the left has the [$= 1$ out] and [≥ 1 in] properties, which means it is a total, surjective function. But it does not have the [≤ 1 in] property because element 3 has two arrows going into it; it is not injective.

The relation on the right has the [$= 1$ out] and [≤ 1 in] properties, which means it is a total, injective function. But it does not have the [≥ 1 in] property because element 4 has no arrow going into it; it is not surjective.

The arrows in a diagram for R correspond, of course, exactly to the pairs in the graph of R. Notice that the arrows alone are not enough to determine, for example,

Strengthening Induction

The Math for CS book defines injectivity, surjectivity, and bijectivity for relations. While we thought it was good to see the basics of relations, since they capture an even more general structure than functions (which can help in cases with more complex connections such as the example in the reading with instructors for a set of classes), we (and 121/124) focus on functions. Additionally, the definitions for injectivity, surjectivity, and bijectivity given in the Math for CS book (ie the ≥ 1 arrows properties) are a little bit hand-wavy and non-standard. Thus, we give the common definition for injectivity, surjectivity, and bijectivity of functions for you to use in this course. These are taken from the videos for the course.

Definition 0.1 *A total function f from set A to set B, written as $f : A \to B$ assigns each element of A to exactly one element of B.*

Note that when we say function in this course, unless we explicitly say "partial function," we mean total function.

Definition 0.2 *A function $f : A \to B$ is surjective if $\forall b \in B \exists a \in A. f(a) = b$.*

Definition 0.3 *A function $f : A \to B$ is injective if $\forall a \neq a' \in A. f(a) \neq f(a')$.*

Definition 0.4 *A function f is bijective if it is both injective and surjective.*

The definitions for other properties, such as the image or preimage, are given in the text for relations and therefore hold for functions as well (since every function is a relation).

if R has the [$\geq$ 1 out], total, property. If all we knew were the arrows, we wouldn't know about any points in the domain column that had no arrows out. In other words, graph(R) alone does not determine whether R is total: we also need to know what domain(R) is.

Example 4.4.3. The function defined by the formula $1/x^2$ has the [$\geq$ 1 out] property if its domain is $\mathbb{R}^+$, but not if its domain is some set of real numbers including 0. It has the [$=$ 1 in] and [$=$ 1 out] property if its domain and codomain are both $\mathbb{R}^+$, but it has neither the [$\leq$ 1 in] nor the [$\geq$ 1 out] property if its domain and codomain are both $\mathbb{R}$.

4.4.2 Relational Images

The idea of the image of a set under a function extends directly to relations.

Definition 4.4.4. The *image* of a set Y under a relation R written $R(Y)$, is the set of elements of the codomain B of R that are related to some element in Y. In terms of the relation diagram, $R(Y)$ is the set of points with an arrow coming in that starts from some point in Y. The *range* range(R) of R is the image $R(A)$ of the domain A of R. That is, range(R) is the set of all points in the codomain with an arrow coming in.

For example, the set of subject numbers that Meyer is in charge of in Spring '10 is exactly *Chrg*(A. Meyer). To figure out what this is, we look for all the arrows in the *Chrg* diagram that start at "A. Meyer," and see which subject-numbers are at the other end of these arrows. Looking at the list (4.4) of pairs in graph(*Chrg*), we see that these subject-numbers are {6.042, 18.062, 6.844}. Similarly, to find the subject numbers that either Freeman or Eng are in charge of, we can collect all the arrows that start at either "G. Freeman," or "T. Eng" and, again, see which subject-numbers are at the other end of these arrows. This is *Chrg*({G. Freeman, T. Eng}). Looking again at the list (4.4), we see that

$$\mathit{Chrg}(\{\text{G. Freeman, T. Eng}\}) = \{6.011, 6.881, 6.882, 6.\text{UAT}\}$$

Finally, Fac is the set of all in-charge instructors, so *Chrg*(Fac) is the set of all the subjects listed for Spring '10.

Inverse Relations and Images

Definition 4.4.5. The *inverse*, R^{-1} of a relation $R : A \to B$ is the relation from B to A defined by the rule

$$b\ R^{-1}\ a \quad \text{IFF} \quad a\ R\ b.$$

In other words, R^{-1} is the relation you get by reversing the direction of the arrows in the diagram of R.

Definition 4.4.6. The *inverse image* of a set $X \subseteq B$ under the relation R is defined to be $R^{-1}(X)$, namely, the set of elements in A connected by an arrow to some element in B. The *support* support(R) is defined to $R^{-1}(B)$, namely, the set of domain elements with at least one arrow out. The support of R is also called the *domain of definition* of R.

Continuing with the in-charge example above, the set of instructors in charge of 6.UAT in Spring '10 is exactly the inverse image of {6.UAT} under the *Chrg* relation. From the list (4.4), we see that Eng and Freeman are both in charge of 6.UAT, that is,

$$\{\text{T. Eng, D. Freeman}\} \subseteq \mathit{Chrg}^{-1}(\{\text{6.UAT}\}).$$

We can't assert equality here because there may be additional pairs further down the list showing that additional instructors are co-incharge of 6.UAT.

Now let Intro be the set of introductory course 6 subject numbers. These are the subject numbers that start with "6.0." So the set of names of the instructors who were in-charge of introductory course 6 subjects in Spring '10, is $\mathit{Chrg}^{-1}(\text{Intro})$. From the part of the *Chrg* list shown in (4.4), we see that Meyer, Leighton, Freeman, and Guttag were among the instructors in charge of introductory subjects in Spring '10. That is,

$$\{\text{Meyer, Leighton, Freeman, Guttag}\} \subseteq \mathit{Chrg}^{-1}(\text{Intro}).$$

Finally, $\mathit{Chrg}^{-1}(\text{SubNums})$ is the set of all instructors who were in charge of a subject listed for Spring '10.

4.5 Finite Cardinality

A finite set is one that has only a finite number of elements. This number of elements is the "size" or *cardinality* of the set:

Definition 4.5.1. If A is a finite set, the *cardinality* $|A|$ of A is the number of elements in A.

A finite set may have no elements (the empty set), or one element, or two elements,..., so the cardinality of finite sets is always a nonnegative integer.

Now suppose $R : A \to B$ is a function. This means that every element of A contributes at most one arrow to the diagram for R, so the number of arrows is at most the number of elements in A. That is, if R is a function, then

$$|A| \geq \#\text{arrows}.$$

If R is also surjective, then every element of B has an arrow into it, so there must be at least as many arrows in the diagram as the size of B. That is,

$$\#\text{arrows} \geq |B|.$$

Combining these inequalities implies that if R is a surjective function, then $|A| \geq |B|$.

In short, if we write A surj B to mean that there is a surjective function from A to B, then we've just proved a lemma: if A surj B for finite sets A, B, then $|A| \geq |B|$. The following definition and lemma lists this statement and three similar rules relating domain and codomain size to relational properties.

Definition 4.5.2. Let A, B be (not necessarily finite) sets. Then

1. A surj B iff there is a surjective *function* from A to B.
2. A inj B iff there is an injective *total* relation from A to B.
3. A bij B iff there is a bijection from A to B.

Lemma 4.5.3. *For finite sets A, B:*

1. *If A surj B, then $|A| \geq |B|$.*
2. *If A inj B, then $|A| \leq |B|$.*
3. *If A bij B, then $|A| = |B|$.*

Proof. We've already given an "arrow" proof of implication 1. Implication 2. follows immediately from the fact that if R has the [≤ 1 out], function property, and the [≥ 1 in], surjective property, then R^{-1} is total and injective, so A surj B iff B inj A. Finally, since a bijection is both a surjective function and a total injective relation, implication 3. is an immediate consequence of the first two. ■

Lemma 4.5.3.1. has a converse: if the size of a finite set A is greater than or equal to the size of another finite set B then it's always possible to define a surjective

function from A to B. In fact, the surjection can be a total function. To see how this works, suppose for example that

$$A = \{a_0, a_1, a_2, a_3, a_4, a_5\}$$
$$B = \{b_0, b_1, b_2, b_3\}.$$

Then define a total function $f : A \to B$ by the rules

$$f(a_0) ::= b_0,\ \ f(a_1) ::= b_1,\ \ f(a_2) ::= b_2,\ \ f(a_3) = f(a_4) = f(a_5) ::= b_3.$$

More concisely,

$$f(a_i) ::= b_{\min(i,3)},$$

for $0 \le i \le 5$. Since $5 \ge 3$, this f is a surjection.

So we have figured out that if A and B are finite sets, then $|A| \ge |B|$ *if and only if* A surj B. All told, this argument wraps up the proof of a theorem that summarizes the whole finite cardinality story:

Theorem 4.5.4. *[Mapping Rules] For* finite *sets* A, B,

$$|A| \ge |B| \quad \textit{iff} \quad A \text{ surj } B, \tag{4.5}$$
$$|A| \le |B| \quad \textit{iff} \quad A \text{ inj } B, \tag{4.6}$$
$$|A| = |B| \quad \textit{iff} \quad A \text{ bij } B, \tag{4.7}$$

4.5.1 How Many Subsets of a Finite Set?

As an application of the bijection mapping rule (4.7), we can give an easy proof of:

Theorem 4.5.5. *There are* 2^n *subsets of an* n*-element set. That is,*

$$|A| = n \quad \textit{implies} \quad |\operatorname{pow}(A)| = 2^n.$$

For example, the three-element set $\{a_1, a_2, a_3\}$ has eight different subsets:

$$\begin{array}{cccc} \emptyset & \{a_1\} & \{a_2\} & \{a_1, a_2\} \\ \{a_3\} & \{a_1, a_3\} & \{a_2, a_3\} & \{a_1, a_2, a_3\} \end{array}$$

Theorem 4.5.5 follows from the fact that there is a simple bijection from subsets of A to $\{0,1\}^n$, the n-bit sequences. Namely, let $a_1, a_2, \ldots, a_n$ be the elements of A. The bijection maps each subset of $S \subseteq A$ to the bit sequence $(b_1, \ldots, b_n)$ defined by the rule that

$$b_i = 1 \quad \text{iff} \quad a_i \in S.$$

For example, if $n = 10$, then the subset $\{a_2, a_3, a_5, a_7, a_{10}\}$ maps to a 10-bit sequence as follows:

$$\begin{array}{rccccccccccc} \text{subset:} & \{ & & a_2, & a_3, & & a_5, & & a_7, & & & a_{10} & \} \\ \text{sequence:} & (& 0, & 1, & 1, & 0, & 1, & 0, & 1, & 0, & 0, & 1 &) \end{array}$$

Now by bijection case of the Mapping Rules 4.5.4.(4.7),

$$|\operatorname{pow}(A)| = |\{0,1\}^n|.$$

But every computer scientist knows[6] that there are 2^n n-bit sequences! So we've proved Theorem 4.5.5!

Problems for Section 4.1

Practice Problems

Problem 4.1.
For any set A, let $\operatorname{pow}(A)$ be its *power set*, the set of all its subsets; note that A is itself a member of $\operatorname{pow}(A)$. Let $\emptyset$ denote the empty set.

(a) The elements of $\operatorname{pow}(\{1,2\})$ are:

(b) The elements of $\operatorname{pow}(\{\emptyset, \{\emptyset\}\})$ are:

(c) How many elements are there in $\operatorname{pow}(\{1,2,\ldots,8\})$?

Problem 4.2.
Express each of the following assertions about sets by a formula of set theory.[7] Expressions may use abbreviations introduced earlier (so it is now legal to use "=" because we just defined it).

(a) $x = \emptyset$.

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

(c) $x \subseteq y$. (x is a subset of y that might equal y.)

[6] In case you're someone who doesn't know how many n-bit sequences there are, you'll find the 2^n explained in Section 15.2.2.

[7] See Section 8.3.2.

8 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. These are all infinite sets. 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 8.2, 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.

8.1 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 8.1.1. A strict B iff NOT(A 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 8.1.2. *For finite sets A, B,*

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

Proof.

$$\begin{array}{lll} A \text{ strict } B & \text{iff} & \text{NOT}(A \text{ surj } B) \qquad\qquad (\text{Def } 8.1.1) \\ & \text{iff} & \text{NOT}(|A| \geq |B|) \qquad (\text{Theorem } 4.5.4.(4.5)) \\ & \text{iff} & |A| < |B|. \end{array}$$

■

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 need 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 proven.

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 8.1.3. *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 $[\leq 1 \text{ out}, \geq 1 \text{ in}]$ surjective function property iff R^{-1} has the $[\geq 1 \text{ out}, \leq 1 \text{ in}]$ 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 4.21.

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

Theorem 8.1.4. *[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 8.12.

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 is obvious would be mistaken again.

Theorem 8.1.5. *For* all *sets* A, B,

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

Theorem 8.1.5 lets us prove that another basic property of finite sets carries over to infinite ones:

Lemma 8.1.6.

$$A \text{ strict } B \quad \text{AND} \quad B \text{ strict } C \tag{8.1}$$

implies

$$A \text{ strict } C$$

for all sets A, B, C.

Proof. (of Lemma 8.1.6)

Suppose 8.1 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 8.1.5 lets us conclude that C surj B. So we have

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

and Lemma 8.1.3.2 lets us conclude that A surj B, contradicting the fact that A strict B. ■

We're omitting a proof of Theorem 8.1.5 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 8.1.6 is the only use we'll make of Theorem 8.1.5, we hope you won't feel cheated not to see a proof.

8.1.1 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 8.1.7. *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 $a_0, a_1, a_2, \ldots, a_n, \ldots$ of different elements of A. Now it's easy to define a bijection $e : A \cup \{b\} \to A$:

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

■

8.1.2 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

$$c_0, c_1, \ldots, c_n, \ldots.$$

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 8.1.8. A set C is *countably infinite* iff $\mathbb{N}$ bij C. A set is *countable* iff it is finite or countably infinite. A set is *uncountable* iff it is not countable.

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

$$2, 4, 6, 6, 6, \ldots.$$

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

$$c_0, c_1, \ldots, c_n, \ldots$$

of the elements of C, possibly with repeats.

Lemma 8.1.9. *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 8.13.

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:

$$0, -1, 1, -2, 2, -3, 3, \ldots. \tag{8.2}$$

In this case, there is a simple formula for the nth element of the list (8.2). 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 8.23). 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 8.11 illustrates how to do it. More generally, it is easy to show that countable sets are closed under unions and products (Problems 8.22 and 8.23) which implies the countability of a bunch of familiar sets:

Corollary 8.1.10. *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 8.1.7 shows that countably infinite sets are the "smallest" infinite sets. Namely,

Lemma 8.1.11. *If A is an infinite set, and B is countable, then A* surj *B.*

We leave the proof to Problem 8.10.

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 8.15).

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's 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.

8.1.3 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 8.1.12. *[Cantor] For any set A,*

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

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. Since any partial function with nonempty codomain can be extended to a total function with the same range (reader: ask yourself how), we can safely assume that g is total.

To show that g is not a surjection, 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 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 it differs at a from each set $g(a)$ in the range of g.

To spell this out more, suppose to the contrary that A_g was in the range of g, that is,

$$A_g = g(a_0)$$

for some $a_0 \in A$. Now 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 8.1.13. $\text{pow}(\mathbb{N})$ *is uncountable.*

Proof. By Lemma 8.1.9, U is uncountable iff $\mathbb{N}$ strict U. ■

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 8.1.14. *$\{0,1\}^\omega$ 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 8.1.3. In particular, we can appeal to the following immediate corollary of the Lemma:

Corollary 8.1.15.

(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 8.1.16. *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,

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

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 8.1.15.(a).

For another example, let's prove

Corollary 8.1.17. *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,

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

[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.

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 8.1.15.(b).

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 8.1.12, 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 8.14). In this way you can keep going indefinitely, building "bigger" infinities all the way.

8.1.4 Diagonal Argument

Theorem 8.1.12 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:

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

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:

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

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

$$D = \begin{array}{cccccccc} 1 & 1 & 1 & 0 & 0 & 1 & \cdots, \end{array}$$

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

$$C = \begin{array}{cccccccc} 0 & 0 & 0 & 1 & 1 & 0 & \cdots. \end{array}$$

Now if the 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 8.1.12. 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.

8.2 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