Chapter 17

Countability

This chapter covers infinite sets and countability.

17.1 The rationals and the reals

You’re familiar with three basic sets of numbers: the integers, the rationals, and the reals. The integers are obviously discrete, in that there’s a big gap between successive pairs of integers.

To a first approximation, the rational numbers and the real numbers seem pretty similar. The rationals are dense in the reals: if I pick any real number $x$ and a distance $\delta$, there is always a rational number within distance $\delta$ of $x$. Between any two real numbers, there is always a rational number.

We know that the reals and the rationals are different sets, because we’ve shown that a few special numbers are not rational, e.g. \( \pi \) and $\sqrt{2}$. However, these irrational numbers seem like isolated cases. In fact, this intuition is entirely wrong: the vast majority of real numbers are irrational and the rationals are quite a small subset of the reals.
17.2 Completeness

One big difference between the two sets is that the reals have a so-called “completeness” property. It states that any subset of the reals with an upper bound has a smallest upper bound. (And similarly for lower bounds.) So if I have a sequence of reals that converges, the limit it converges to is also a real number. This isn’t true for the rationals. We can make a series of rational numbers that converge \( \pi \) (for example) such as

\[
\]

But there is no rational number equal to \( \pi \).

In fact, the reals are set up precisely to make completeness work. One way to construct the reals is to construct all convergent sequences of rationals and add new points to represent the limits of these sequences. Most of the machinery of calculus depends on the existence of these extra points.

17.3 Cardinality

We know how to calculate and compare the sizes of finite sets. To extend this idea to infinite sets, we use bijections functions to compare the sizes of sets:

**Definition:** Two sets \( A \) and \( B \) have the same cardinality (\(|A| = |B|\)) if and only if there is a bijection from \( A \) to \( B \).

We’ve seen that there is a bijection between two finite sets exactly when the sets have the same number of elements. So this definition of cardinality matches our normal notion of how big a finite set is. But, since we don’t have any numbers of infinite size, working with bijections extends better to infinite sets.

The integers and the natural numbers have the same cardinality, because we can construct a bijection between them. Consider the function \( f : \mathbb{N} \rightarrow \mathbb{Z} \) where \( f(n) = \frac{n}{2} \) when \( n \) is even and \( f(n) = \frac{-(n+1)}{2} \) when \( n \) is odd. \( f \) maps
the even natural numbers bijectively onto the non-negative integers. It maps the odd natural numbers bijectively onto the negative integers.

Because the integers are so important, there’s a special name for sets that have the same cardinality as the integers:

An infinite set $A$ is countably infinite if there is a bijection from $\mathbb{N}$ (or equivalently $\mathbb{Z}$) onto $A$.

The term countable is used to cover both finite sets and sets that are countably infinite.

### 17.4 More countably infinite sets

Now, let’s look at $\mathbb{N}^2$, the set of pairs of natural numbers. This set is countably infinite, because we can construct a bijection $f : \mathbb{N}^2 \rightarrow \mathbb{N}$. To do this, we’ll organize the set of pairs of natural numbers by size, where the size of a pair $(p, q)$ is the sum $p + q$ of its coordinates. The pairs of size $k$ are mapped onto an interval of length $k + 1$ starting at $\frac{k(k+1)}{2}$. You can verify that these intervals cover the natural numbers exactly, without overlapping. Within each interval, we map pairs to integer positions in order, based on the size of the first coordinate $p$. This map is a bijection from the pairs of natural numbers to the natural numbers.

Suppose that we have a finite set $A$ of $n$ characters. Then the set $A^*$ contains all finite sequences of characters from $A$. For example, $A$ might be the set of ASCII characters. In that case, $A^*$ contains character strings of various finite lengths, such as “a,” “This,” and “transitivity.” $A^*$ is countably infinite. Again, we organize these strings by size, but this time “size” is the length of the strings. The $n^k$ strings of length $k$ are mapped into an interval of the integers, starting right after the interval for the strings of length $k - 1$. If you work out the coordinates properly, this also ends up as a bijection.

These constructions can be extended to show the countability of a variety of similar sets, e.g. positive integers, triples of natural numbers, or 7-tuples of integers.
17.5 Cantor Schroeder Bernstein Theorem

For certain countably infinite sets, it’s awkward to directly build a bijection to the integers or natural numbers. Fortunately, a more general technique is available. Remember that for finite sets, we could build a one-to-one function from $A$ to $B$ if and only if $|A| \leq |B|$. Using this idea, we can define a partial order on sets, finite or infinite:

Definition: $|A| \leq |B|$ if and only if there is a one-to-one function from $A$ to $B$.

It is the case that if $|A| \leq |B|$ and $|B| \leq |A|$, then $|A| = |B|$. That is, if you can build one-to-one functions in both directions, a bijection does exist. This result, called the Cantor Schroeder Bernstein Theorem, isn’t obvious when the sets are infinite. You do have enough background to understand its proof (which you can look up), but it’s somewhat messy.

To see this theorem in action, consider the rational numbers. Rational numbers are almost the same as fractions, which are basically pairs of integers, which we know to be countably infinite. But this isn’t quite right: each rational number is represented by many fractions.

So, let’s use Cantor Schroeder Bernstein: to show that the positive rational numbers are the same size as the natural numbers show that $|\mathbb{N}| \leq |\mathbb{Q}^+|$ and $|\mathbb{Q}^+| \leq |\mathbb{N}|$. It’s easy to make a one-to-one function from the natural numbers to the positive rational numbers: just map each natural number $n$ to $n + 1$ (because $\mathbb{Q}^+$ doesn’t include zero). So $|\mathbb{N}| \leq |\mathbb{Q}^+|$. So now we just need a one-to-one function from the rationals to the integers, to show that $|\mathbb{Q}^+| \leq |\mathbb{N}|$.

To map the positive rational numbers to the natural numbers, first map each rational number to one representative fraction, e.g. the one in lowest terms. This isn’t a bijection, but it is one-to-one. Then use the method we saw above to map the fractions, which are just pairs of positive integers, to the natural numbers. We now have the required one-to-one function from the positive rationals to the natural numbers.

Again, this construction can be adapted to also handle negative rational numbers. So the set of rational numbers is countably infinite.
17.6 \( \mathcal{P}(\mathbb{N}) \) isn’t countable

Before looking at the real numbers, let’s first prove a closely-related result that’s less messy: \( \mathcal{P}(\mathbb{N}) \) isn’t countable. Recall that \( \mathcal{P}(\mathbb{N}) \) is the power set of the natural numbers i.e. the set containing all subsets of the natural numbers.

Suppose that \( A \) is a finite set \( \{a_0, a_1, a_2, \ldots, a_n\} \). We can represent a subset \( X \) of \( A \) as a bit vector \( \{b_0, b_1, b_2, \ldots, b_n\} \) where \( b_i \) is 1 if and only if \( a_i \) is in \( X \). For example, if \( A = \{7, 8, 9, 10, 11\} \), then the bit-vector 01100 would represent the subset \( \{8, 9\} \) and the bit-vector 10101 would represent the subset \( \{7, 9, 11\} \). Similarly, we can represent a subset of the natural numbers as an infinite-length bit vector.

We’ll use a procedure called “diagonalization” (due to Georg Cantor) to show that it’s impossible to build a bijection from the natural numbers to these infinite bit vectors representing the subsets of the natural numbers. Suppose that there were such a bijection. Then we could put all the bit vectors into a list, e.g. \( v_0 \) would be the first bit vector, \( v_1 \) the second, and so forth. Our list of bit vectors might look like this, where the \( k \)th column contains the value of the \( k \)th digit \( (b_k) \) for all the bit vectors:

<table>
<thead>
<tr>
<th></th>
<th>( b_0 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( b_4 )</th>
<th>( b_5 )</th>
<th>( b_6 )</th>
<th>( b_7 )</th>
<th>( b_8 )</th>
<th>( b_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_0 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>( v_1 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>( v_2 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>( v_3 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>( v_4 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>( v_5 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

This is supposed to be a complete list of all the bit vectors. But we can construct a bit vector \( x \) that’s not on the list. The value of \( x_k \), i.e. the \( k \)th bit in our new vector, will be 0 if the \( k \)th digit of \( v_k \) is 1, and 1 if the \( k \)th digit of \( v_k \) is 0. Notice that \( x \) is different from \( v_3 \) because the two vectors differ in the third position. It can’t be \( v_20 \) because the two vectors differ in the twentieth position. And, in general, \( x \) can’t equal \( v_k \) because the two vectors differ in the \( k \)th position. For the example above, the new vector not in the list would start out: 0 0 1 0 0 1 ...
So, it’s not possible to put these infinite bit vectors into a list indexed by the natural numbers, because we can always construct a new bit vector that’s not on the list. That is, there can’t be a one-to-one function from the infinite bit vectors to the natural numbers. So there can’t be a one-to-one function from the subsets of the natural numbers to the natural numbers. So \(\mathcal{P}(\mathbb{N})\) isn’t countable. That is, the subsets of the natural numbers are more numerous than the natural numbers themselves.

### 17.7 More uncountability results

This same diagonalization trick can be used to show that various other sets aren’t countable. For example, we can show that the real numbers aren’t countable. To show this, we show that even the numbers in the interval \([0, 1]\) aren’t countable. Suppose that the elements of \([0, 1]\) were countable. Then we could put these real numbers into a list \(a_1, a_2, \ldots\) and so forth. Let’s write out a table of the decimal expansions of the numbers on this list. Now, examine the digits along the diagonal of this table: \(a_{11}, a_{22}, \ldots\). Suppose we construct a new number \(b\) whose \(k\)th digit \(b_k\) is 4 when \(a_{kk}\) is 5, and 5 otherwise. Then \(b\) won’t match any of the numbers in our table, so our table wasn’t a complete list of all the numbers in \([0, 1]\). So, \([0, 1]\) is not countable, and therefore the reals can’t be countable.

Next, notice that an infinite bit vector is a function from the natural numbers to the set \(\{0, 1\}\). So we’ve shown that there are uncountably many functions from the natural numbers to \(\{0, 1\}\). So there must be uncountably many functions from the natural numbers to the natural numbers, or from the integers to the integers.

Another generalization involves noticing that our diagonalization proof doesn’t really depend on any special properties of the natural numbers. So it can be adapted to show that, if \(A\) is any set, \(\mathcal{P}(A)\) has a (strictly) larger cardinality than \(A\). So, not only is \(\mathcal{P}(\mathbb{N})\) larger than \(\mathcal{P}(\mathbb{N})\), but \(\mathcal{P}(\mathcal{P}(\mathbb{N}))\) is even larger. So there is a whole sequence of larger and larger infinite cardinalities.

So, in particular, \(\mathcal{P}(\mathbb{R})\) is larger than \(\mathbb{R}\). However, notice that \(\mathbb{R}^2\) has the same cardinality as \(\mathbb{R}\). Let’s consider a simpler version of this problem: \([0, 1)^2\) has the same cardinality as \([0, 1]\). Any element of \([0, 1)^2\) can be represented...
as two infinite sequences of decimal digits: \(0.a_1a_2a_3a_4\ldots\) and \(0.b_1b_2b_3b_4\ldots\).
We can map this to a single real number by interleaving the digits of the two numbers: \(0.a_1b_1a_2b_2a_3b_3\ldots\). This defines a bijection between the two sets. This method can be adapted to create a bijection between all of \(\mathbb{R}^2\) and \(\mathbb{R}\).

### 17.8 Uncomputability

We can pull some of these facts together into some interesting consequences for computer science. Notice that a formula for a function is just a finite string of characters. So the set of formulas is countable. But the set of functions, even from the integers to the integers, is uncountable. So there are more functions than formulas, i.e. some functions which have no finite formula.

Similarly, notice that a computer program is simply a finite string of ASCII characters. So there are only countably many computer programs. But there are uncountably many functions. So there are more functions than programs, i.e. there are functions which cannot be computed by any program.

A final problem is created by the fact that, although the code for a computer program is finite in length, the trace of the program’s execution may be infinite. Specifically, program traces fall into three categories

1. The program eventually halts, so the trace is finite.
2. The program loops, in the sense of returning back to a previous state.
3. The program keeps going forever, consuming more and more storage space.

The second type of behavior is like the decimal expansion of a rational number: repeating. This third type of behavior is similar to a real number’s decimal expansion: an infinite sequence which doesn’t repeat. As with numbers, the infinite non-repeating sequences create interesting complexity. In this case, the famous Halting Problem which shows that you can’t build a program that will decide whether other programs halt or not.
A final example, further afield but closely related mathematically, involves tilings of the plane. Periodic tilings, which are made up of many copies of a single repeated pattern, are like rational numbers or looping programs. Aperiodic tilings, whose existence was only proved in 1966, are like real numbers or the third type of program: they go on forever, but not by repeating a single pattern over and over. See the internet for pictures of aperiodic tilings, especially Penrose tilings. Aside from looking cool, they have applications in modelling “quasicrystals” e.g. you can create a crystal-like structures with a 5-fold symmetry.

17.9 Variation in notation

Authors differ as to whether the term countable includes finite sets or only countably infinite sets.