CSCI 270 Lecture 29: The Limits of Computation

Suppose you are given some computer code, and the input which the code will receive. You want to know whether the code will halt, or enter an infinite loop. This problem is referred to as the **Halting Problem**.

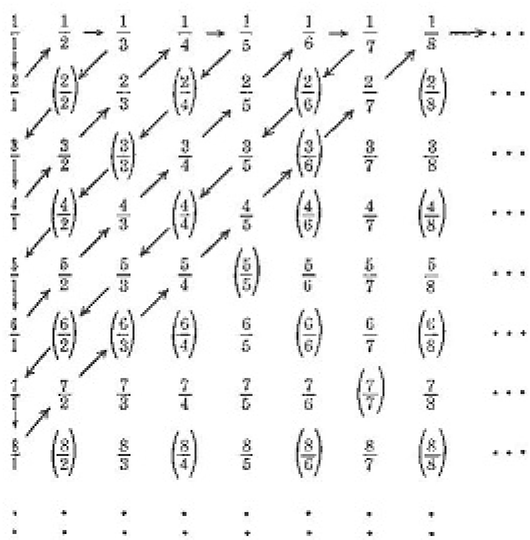
Countable versus Uncountable

Which set is larger: $\{0, 1, 2, 3, ...\}$, or $\{0, 2, 4, 6, ...\}$?

Two sets are the same size if there is a bijective function f which maps A to B.

That is, pair each number in A with a unique number in B such that every number in B is paired exactly once. Any set equal in size to $\{0, 1, 2, ...\}$ is called **countable** or **countably infinite**.

- Are the set of even integers countable?
- Are the set of positive rational numbers countable?



The set of real numbers are not countable.

Proof by contradiction: assume we have a bijective function f which maps A (the set of natural numbers) to B (the set of real numbers). Perhaps:

- $f(1) = \pi$
- $f(2) = 0.21345$
- $f(3) = 32.\overline{33}$ (repeating, of course), etc.

Now we will construct a real number $b : 0 < b < 1$ which does not appear in this table, using *diagonalization*.

If the ith decimal place of $f(i)$ is 1, then the ith decimal place of b will be 2, otherwise it will be 1. Given the above example function, $b = 0.221...$

- Does any input produce b as output?
- What does this have to do with Computer Science?
- Are the set of computer programs countable or not?
- Are the set of problems countable or not?

To answer the last question, lets make some simplifying assumptions. All problems require as input a single natural number (clearly there are other problems which don't meet this criteria). In addition, all problems will return a boolean value (true or false).

Undecidable Problems

We will assume that the Halting Problem can be solved by program $H(M, w)$. It takes as input computer code M, and input to that computer code w and returns True, meaning that M halts on input w, or False, meaning that it doesn't halt. H always halts (otherwise it doesn't solve the problem).

Using our solution to the Halting Problem, we can write a different program $B(M)$. It works as follows:

```
B(computer code M)
  If (H(M,M)==False) Then Return True
  Else loop forever
```

What happens when I run B on input $M = B$?

- If B(B) halts, then H(B,B) returns True, which means B(B) loops forever. Contradiction.
- If B(B) doesn't halt, then H(B,B) returns False, which means B(B) returns True. Contradiction.

B is the barber from the barber paradox. If you assume B exists, you get a paradox. The only possible conclusion is that B does not exist. We also showed $B \leq H$, so H doesn't exist either!

XKCD: