Uncountability

CSE 311 Autumn 24
Lecture 28

Miscellaneous Announcements

CC29 (from Friday's lecture) will be due Monday at 11:59 PM.

CC30 re-asks you the survey questions from CC0 over again (along with some extra ones),

Please fill out course evals---they really help us update the course.

Punchline for Today's Lecture

There are more functions than there are computer programs.

So for some functions there just isn't a computer program that computes it.

Outline

Some definitions – what do we mean by “more”?

How many programs are there?

Proving there are more functions.

Sizes of sets

How do we know two sets are the same size?

Easy. Count the number of elements in both.

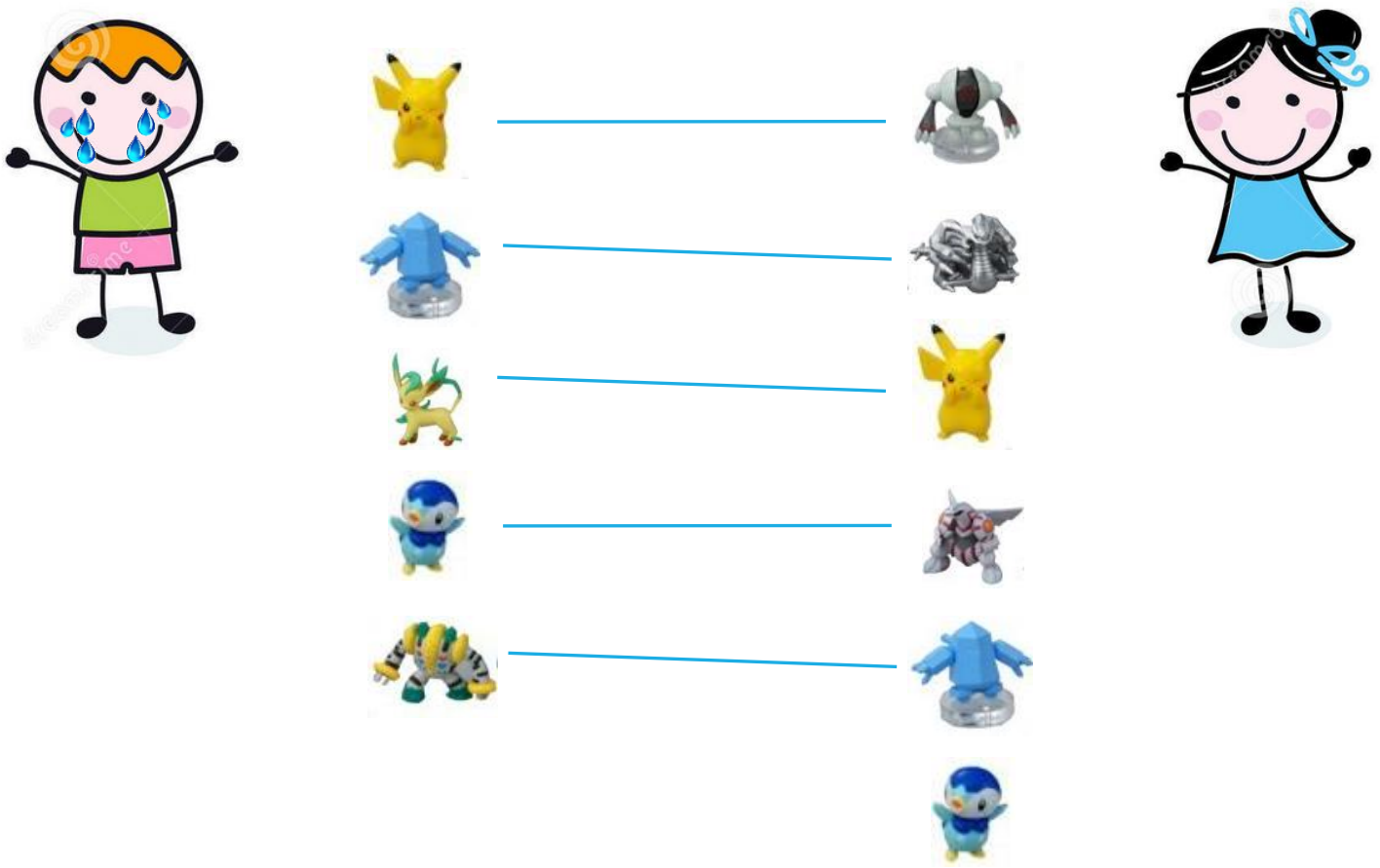
That works great for finite sets, but ∞ isn't really a number we get to count to…

More Practical

What does it mean that two sets have the same size?

More Practical

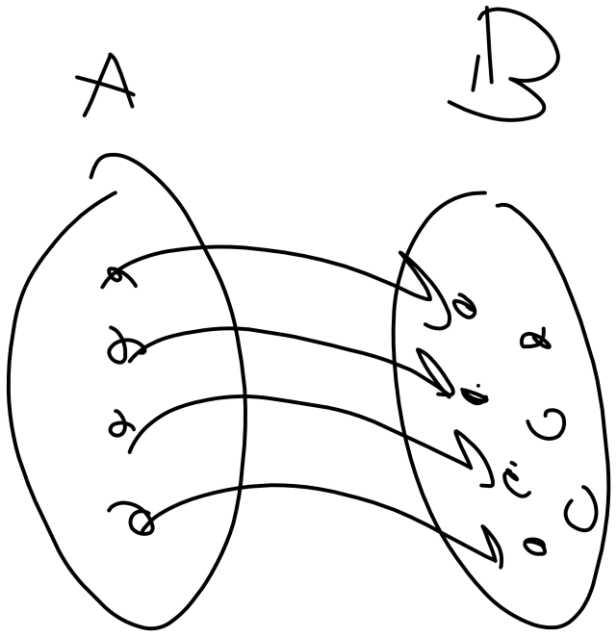
What does it mean that two sets have the same size?



Two Requirements for a Bijection

A function $f: A \to B$ maps every element of A to one element of B

A is the "domain", B is the "co-domain"



One-to-one (aka injection)

A function f is one-to-one iff

$$\forall a \forall b (f(a) = f(b) \to a = b)$$

That is, every output has at most one possible input.

Two Requirements for a Bijection

A function $f: A \to B$ maps every element of A to one element of B

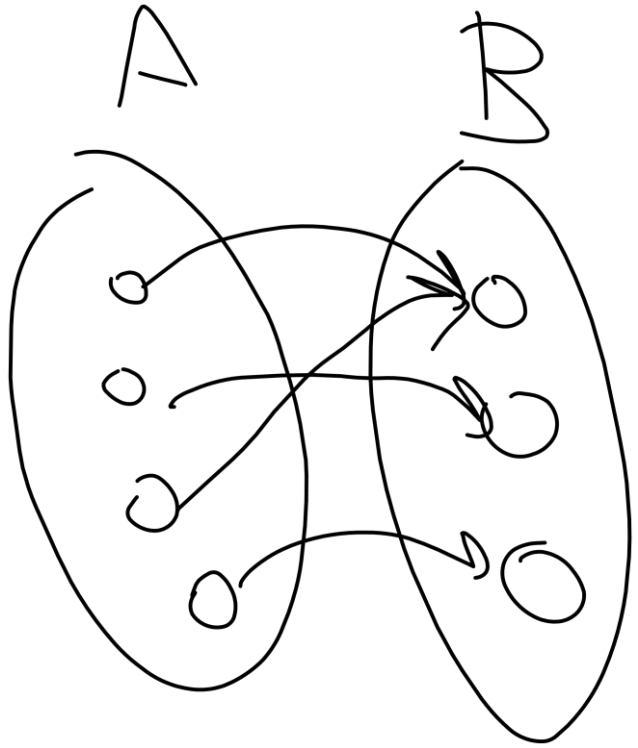
A is the "domain", B is the "co-domain"

Onto (aka surjection)

A function $f: A \to B$ is onto iff

$$\forall b \in B \exists a \in A(b = f(a))$$

Every output has at least one input that maps to it.



Bijection

One-to-one (aka injection)

A function f is one-to-one iff
$\forall a \forall b(f(a) = f(b) \rightarrow a = b)$

Onto (aka surjection)

A function $f: A \rightarrow B$ is onto iff
$\forall b \in B \exists a \in A(b = f(a))$

Bijection

A function $f: A \rightarrow B$ is a bijection iff
f is one-to-one and onto

A bijection maps every element of the domain to **exactly** one element of the co-domain, and every element of the domain to **exactly** one element of the domain.

Definition

Two sets A, B have the same size (same cardinality) if and only if there is a bijection $f: A \to B$

This matches our intuition on finite sets.

But it also works for infinite sets!

Let's see just how infinite these sets are.

Some infinite sets

Two sets A, B have the same size (same cardinality) if and only if there is a bijection $f: A \to B$

Let's compare the sizes of: $\mathbb{N}$, $\mathbb{Z}$, $\{x : x \text{ is an even integer}\}$

Pollev.com/robbie

They're all the same size.

$\mathbb{Z}$ and even integers?

$f(x) = 2x$ Is it a bijection?

$f(x) = f(y) \rightarrow 2x = 2y \rightarrow x = y$;

If z is even then $z = 2k$ for some integer k and $f(k) = z$.

$\mathbb{N}$ and $\mathbb{Z}$

$$g(x) = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ -\frac{x+1}{2} & \text{if } x \text{ is odd} \end{cases}$$

They're all the same size…

$\mathbb{N}$ and even integers?

$g(f(x))$ will work nicely. You can also build one explicitly.

Good exercise: show that if f and g are bijections then $f \circ g$ is also a bijection.

Countable

Countable

The set A is countable iff there is an injection from A to $\mathbb{N}$, Equivalently, A is countable iff it is finite or there is a bijection from A to $\mathbb{N}$

$\mathbb{N}, \mathbb{Z}, \{x: x \text{ is an even integer}\}$ are all countable.

To build a bijection from A to $\mathbb{N}$, just list all the elements!

Let’s Try one that’s a little harder

What about $\mathbb{Q}$. There’s gotta be more of those right?

It’s pretty intuitive to think there are more rationals than integers.

The rationals are **dense**.

Between every two rationals, there’s another rational number.

Or said in more intimidating fashion: between every two rationals there are infinitely many others!

The set of positive rational numbers

1/1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	…
2/1	2/2	2/3	2/4	2/5	2/6	2/7	2/8	…
3/1	3/2	3/3	3/4	3/5	3/6	3/7	3/8	…
4/1	4/2	4/3	4/4	4/5	4/6	4/7	4/8	…
5/1	5/2	5/3	5/4	5/5	5/6	5/7	…	
6/1	6/2	6/3	6/4	6/5	6/6	…		
7/1	7/2	7/3	7/4	7/5	….			
…	…	…	…	…				

In bijection with the natural numbers

Order the rationals by their denominator (increasing), breaking ties by numerator.

$1/1, 1/2, 1/3, 2/3, 1/4, 3/4, 1/5, 2/5, 3/5, 4/5, 1/6, \ldots$

$f(x) =$ the x^{th} number in that list (indexed from 0)

That's a bijection from $\mathbb{N}$ to $\mathbb{Q}^+$ (it's not a nice clean formula, but it's definitely a function)

In Bijection with the natural numbers

How do we get all of $\mathbb{Q}$?

We already know how to "get twice as many" – map the even naturals to positives, and the odds to negatives. Like when we were mapping $\mathbb{N}$ to $\mathbb{Z}$.

Fun fact:

The "order via diagonals" technique is closely related to "dovetailing" a super-useful technique in compuatability theory (take 431 to learn more)

Uncountable

Alright. There are clever ways to build bijections.

Is there anything that's bigger than $\mathbb{N}$?

And…like…how would we prove it?

A proof idea

A set is countable iff it can be listed (a list is a bijection with $\mathbb{N}$).

We'll take advantage of that to find an uncountable set.

Claim $\mathbb{R}$ is uncountable.

Actually, it's easier if we show $[0,1)$ is uncountable (i.e. real numbers between 0 and 1).

What do real numbers look like

0. 3 3 3 3 3 3 3 3…
0. 2 7 2 7 2 8 5 4…
0. 1 4 1 5 9 2 6 5…
0. 2 2 2 2 2 2 2 2…
0. 1 2 3 4 5 6 7 8…
0. 9 8 7 6 5 4 3 2…
0. 8 2 7 6 4 5 7 4…
0. 5 9 4 2 7 5 1 7…

A string of digits!

Well not a “string” An infinitely long sequence of digits is more accurate.

Uncountable

Suppose, for the sake of contradiction, that $[0,1)$ is countable.

Then there is a bijection $f: \mathbb{N} \to [0,1)$.

Use that bijection to make the following table…

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3	3	3	3	…
$f(1)$	0.	2	7	2	7	2	8	5	4	…
$f(2)$	0.	1	4	1	5	9	2	6	5	…
$f(3)$	0.	2	2	2	2	2	2	2	2	…
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
…	…	…	…	…	…	…	…	…	…	…

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3				
$f(1)$	0.	2	7	2	7	2				
$f(2)$	0.	1	4	1	5	9				
$f(3)$	0.	2	2	2	2	2				
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
…	…	…	…	…	…	…	…	…	…	…

Goal: find a real number between 0 and 1 that isn't on our table.
(contradiction to bijection)

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3				
$f(1)$	0.	2	7	2	7	2				
$f(2)$	0.	1	4	1	5	9				
$f(3)$	0.	2	2	2	2	2				
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
…	…	…	…	…	…	…	…	…	…	…

How do we find a number that's not in our list?

Well let's make sure whatever our number is, it's not $f(0)$

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3				
$f(1)$	0.	2	7	2	7	2				
$f(2)$	0.	1	4	1	5	9				
$f(3)$	0.	2	2	2	2	2				
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
				…	…	…	…	…	…	…

Well let's make sure whatever our number is, it's not $f(0)$

Set the 0 column to not 3, say…7.

0.7

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3				
$f(1)$	0.	2	7	2	7	2				
$f(2)$	0.	1	4	1	5	9				
$f(3)$	0.	2	2	2	2	2				
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
				…	…	…	…	…	…	…

Well let's make sure whatever our number is, it's not $f(1)$

Set the 1 column to not 7, say…3.

0.73

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3				
$f(1)$	0.	2	7	2	7	2				
$f(2)$	0.	1	4	1	5	9				
$f(3)$	0.	2	2	2	2	2				
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
				…	…	…	…	…	…	…

Well let's make sure whatever our number is, it's not $f(2)$

Set the 2 column to not 1, say…7.

0.737

Proof that [0,1) is not countable

Suppose, for the sake of contradiction, that there is a list of them:

Number	Digits after decimal	0	1	2	3	4	5	6	7	…
$f(0)$	0.	3	3	3	3	3				
$f(1)$	0.	2	7	2	7	2				
$f(2)$	0.	1	4	1	5	9				
$f(3)$	0.	2	2	2	2	2				
$f(4)$	0.	1	2	3	4	5	6	7	8	…
$f(5)$	0.	9	8	7	6	5	4	3	2	…
$f(6)$	0.	8	2	7	6	4	5	7	4	…
$f(7)$	0.	5	9	4	2	7	5	1	7	…
				…	…	…	…	…	…	…

Flipping Rule: let's set the i^{th} column to:
7 if $f(i)$'s i^{th} column is not 7
3 if $f(i)'s\ i^{th}$ column is 7.

0.73777733…

Wrapping Up

0.73777733…

What is it?

It's a real number between 0 and 1(!!!)

Is the number on the list? Well it's not $f(\mathbf{0})$, they differ in column $\mathbf{0}$.

It's not $f(\mathbf{1})$, they differ in column $\mathbf{1}$.

It's not $f(i)$, they differ in column i.

But…f was a bijection. That's a contradiction!

Diagonalization

This proof technique is called **diagonalization**

Often “Cantor’s Diagonalization” (after Cantor, who developed it).

Takeaway 1

There are differing levels of infinity.

Some infinite sets are equal in size.

Other infinite sets are bigger than others.

If this is mind-bending you're in good company.

Cantor's contemporaries accused him of being a "scientific charlatan" and a "corruptor of youth"

But Cantor was right – and his ideas eventually were recognized as correct.

Let's Do Another!

Let $B = \{0,1\}$. Call a function $g: \mathbb{N} \to B$ a "binary valued function"

Intuitively, g would be something like
public boolean g(BigInteger input){ }

If we could write that g in Java.

How many possible $g: \mathbb{N} \to B$ are there?

Proof that ~~[0,1)~~ set of binary-valued functions is not countable

Suppose, for the sake of contradiction, that there is a list of them:

f bijection from $\mathbb{N}$ to function		Output on 0	Output on 1	Output on 2	Output on 3	Output on 4	Output on 5	Output on 6	Output on 7	…
$f(0)$		1	0	1	1	1	0	1	1	…
$f(1)$		0	1	1	0	0	1	0	0	…
$f(2)$		1	1	1	0	0	0	1	1	…
$f(3)$		0	0	0	0	0	0	0	0	…
$f(4)$		1	0	1	1	1	0	1	1	…
$f(5)$		0	0	0	1	0	1	1	1	…
$f(6)$		1	1	0	1	0	1	1	0	…
$f(7)$		0	2	0	1	1	0	1	0	…
…		…	…	…	…	…	…	…	…	…

Proof that ~~$[0,1)$~~ set of binary-valued functions is not countable

Suppose, for the sake of contradiction, that there is a list of them:

f bijection from $\mathbb{N}$ to function		Output on 0	Output on 1	Output on 2	Output on 3	Output on 4	Output on 5	Output on 6	Output on 7	…
$f(0)$		1	0	1	1					
$f(1)$		0	1	1	0					
$f(2)$		1	1	1	0					
$f(3)$		0	0	0	0					
$f(4)$		1	0	1	1	1	0	1	1	…
$f(5)$		0	0	0	1	0	1	1	1	…
$f(6)$		1	1	0	1	0	1	1	0	…
$f(7)$		0	2	0	1	1	0	1	0	…
…		…	…	…	…	…	…	…	…	…

Goal: find a function $g_{diag}: \mathbb{N} \to \{0,1\}$ that isn't on our table. (contradiction to bijection)

Proof that ~~[0,1)~~ set of binary-valued functions is not countable

Suppose, for the sake of contradiction, that there is a list of them:

f bijection from $\mathbb{N}$ to function		Output on 0	Output on 1	Output on 2	Output on 3	Output on 4	Output on 5	Output on 6	Output on 7	…
$f(0)$		1	0	1	1					
$f(1)$		0	1	1	0					
$f(2)$		1	1	1	0					
$f(3)$		0	0	0	0					
$f(4)$		1	0	1	1	1	0	1	1	…
$f(5)$		0	0	0	1	0	1	1	1	…
			1	0	1	0	1	1	0	…
			2	0	1	1	0	1	0	…
			…	…	…	…	…	…	…	…

How do we find a function not on our list?
Well to make sure it's not $f(0)$ (the function in the first row)
Have $g_{diag}(0) = 0$

$$g_{diag}(x) = \begin{cases} 0 \text{ if } x = 1 \\ \dots \\ \dots \end{cases}$$

Proof that ~~[0,1)~~ set of binary-valued functions is not countable

Suppose, for the sake of contradiction, that there is a list of them:

f bijection from $\mathbb{N}$ to function		Output on 0	Output on 1	Output on 2	Output on 3	Output on 4	Output on 5	Output on 6	Output on 7	…
$f(0)$		1	0	1	1					
$f(1)$		0	1	1	0					
$f(2)$		1	1	1	0					
$f(3)$		0	0	0	0					
$f(4)$		1	0	1	1	1	0	1	1	…
$f(5)$		0	0	0	1	0	1	1	1	…
			1	0	1	0	1	1	0	…
			2	0	1	1	0	1	0	…
			…	…	…	…	…	…	…	…

How do we find a function not on our list?
Well to make sure it's not $f(0)$ (the function in the first row)
Have $g_{diag}(0) = 0$

$$g_{diag}(x) = \begin{cases} 0 \text{ if } x = 1 \\ \dots \\ \dots \end{cases}$$

Proof that ~~[0,1)~~ set of binary-valued functions is not countable

Suppose, for the sake of contradiction, that there is a list of them:

f bijection from $\mathbb{N}$ to function		Output on 0	Output on 1	Output on 2	Output on 3	Output on 4	Output on 5	Output on 6	Output on 7	…
$f(0)$		1	0	1	1					
$f(1)$		0	1	1	0					
$f(2)$		1	1	1	0					
$f(3)$		0	0	0	0					
$f(4)$		1	0	1	1	1	0	1	1	…
$f(5)$		0	0	0	1	0	1	1	1	…
			1	0	1	0	1	1	0	…
			2	0	1	1	0	1	0	…
			…	…	…	…	…	…	…	…

How do we find a function not on our list?
Well to make sure it's not $f(i)$ (the function in the i^{th} row)
Have $g_{diag}(i) = 1 - f(i)(i)$

$$g_{diag}(x) = \begin{cases} 0 \text{ if } x = 1 \\ \dots \\ \dots \end{cases}$$

Proof that ~~[0,1)~~ set of binary-valued functions is not countable

Suppose, for the sake of contradiction, that there is a list of them:

f bijection from $\mathbb{N}$ to function		Output on 0	Output on 1	Output on 2	Output on 3	Output on 4	Output on 5	Output on 6	Output on 7	…
$f(0)$		1	0	1	1					
$f(1)$		0	1	1	0					
$f(2)$		1	1	1	0					
$f(3)$		0	0	0	0					
$f(4)$		1	0	1	1	1	0	1	1	…
$f(5)$		0	0	0	1	0	1	1	1	…
						0	1	1	0	…
						1	0	1	0	…
						…	…	…	…	…

How do we find a function not on our list?
Well to make sure it's not $f(i)$ (the function in the i^{th} row)
Have $g_{diag}(i) = 1 - f(i)(i)$

$$g_{diag}(x) = \begin{cases} 1 \text{ if } f(x) \text{ outputs } 0 \text{ on input } x \\ 0 \text{ if } f(x) \text{outputs } 0 \text{ on input } x \end{cases}$$

Wrapping up the proof

Wrapping up the proof.

Observe that g_{diag} is a fully-defined function, and that it has $\mathbb{N}$ as its domain and $\{0,1\}$ as its codomain. It therefore should be in the co-domain of f. But it cannot be on the list, as $g(i)$ is different from the function in the i^{th} row on input i for all i.

This contradicts f being onto! So we have that the set of binary-valued functions (with $\mathbb{N}$ as their domains) is uncountable.

Our Second big takeaway

How many Java methods can we write:

public boolean g(int input) ?

Can you list them?
Yeah!! Put them in **lexicographic** order
i.e. in increasing order of length, with ties broken by alphabetical order.

Wait…that means the number of such Java programs is countable.
And…the number of functions we're supposed to write is uncountable.

Our Second big takeaway

There are more functions $g: \mathbb{N} \to B$ than there are Java programs to compute them.

Some function must be **uncomputable**.

That is there is no piece of code which tells you the output of the function when you give it the appropriate input.

Not just Java

This isn't just about java programs. (all we used about java was that its programs are strings)…that's…well every programming language.

There are functions that simply cannot be computed.

Doesn't matter how clever you are. How fancy your new programming language is. Just doesn't work.*

*there's a difference between `int` and $\mathbb{N}$ here, for the proof to work you really need all integers to be valid inputs, not just integers in a certain range.

Does this matter?

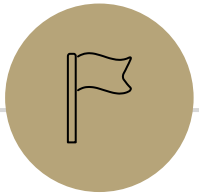
It's even worse than that – almost all functions are not computable.

So…how come this has never happened to you?

This might not be meaningful yet. Almost all functions are also inexpressible in a finite amount of English (English is a language too!)

You've probably never decided to write a program that computes a function you couldn't describe in English…

Are there any problems anyone is **interested** in solving that aren't computable?



The Halting Problem

A Practical Uncomputable Problem

Ever pressed the run button on your code and have it take a long time?

Like an infinitely long time?

What didn't your compiler…like, tell you **not** to push the button yet.

It tells you when your code doesn't compile before it runs it…why doesn't it check for infinite loops?

The Halting Problem

The Halting Problem

Given: source code for a program P and x an input we could give to P
Return: True if P will halt on x, False if it runs forever (e.g. goes in an infinite loop or infinitely recurses)

This would be super useful to solve!

We can't solve it…we'll find out why on Friday.