Logic for Computer Science

07 – Functions

Wouter Swierstra

University of Utrecht

Last time

Proof strategies

Today

Functions

Functions

Given two sets A and B, we can form a new set $A \longrightarrow B$ consisting of **functions** from A to B.

Example

- Suppose I'm teaching a class with 5 students

S = {Alice, Bob, Carroll, David, Eve }.

- At the end of the class, I need to assign marks from 1 to 10 to each student.
- More precisely, this determines a **function**

$$marks : S \longrightarrow \{1 \ldots 10\}$$

More generally, we write $f : A \longrightarrow B$ to mean $f \in A \longrightarrow B$.

Functions

We write `marks(x) = y` when a student `x` is assigned the mark `y` by the `marks` function.

Crucially, each student is assigned a **single** grade.

This rules out situations such as:

```
marks(Alice) = 7
marks(Alice) = 10
```

Furthermore, the `marks` function should assign a mark to **every** student. That is, for each student `s` in S, there is a mark `m` in `{1..10}` such that

```
marks(s) = m
```

A function $A \to B$ must map **every** element $a \in A$ to a **single** element $b \in B$.

Careful!

This is not to say that no two students can have the same grade:

```
marks(Bob)     = 8
marks(Carroll) = 8
```

But the marks function should not associate two different grades with a single student.

Careful!

This is not to say that no two students can have the same grade:

```
marks(Bob)     = 8
marks(Carroll) = 8
```

But the marks function should not associate two different grades with a single student.

Similarly, not all grades need to have a student associated with it.

For example, all students might receive a passing mark.

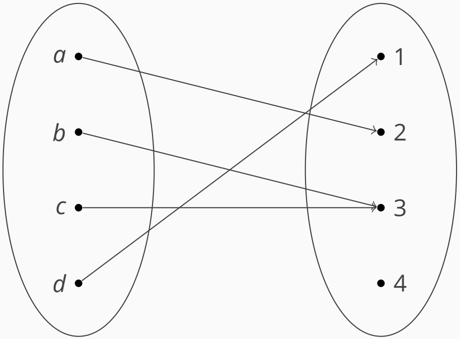
Familiar examples

Functions pop up all over the place:

- `sin` and `cos` are functions $\mathbb{R} \to \mathbb{R}$;
- you might define a method `sort` mapping an array of integers to an array of integers;
- A function `legal : Int × Int × Board` $\to$ `Bool` that checks if placing a new token at position `(x,y)` is a legal move on a Reversi board `b`.
- ...

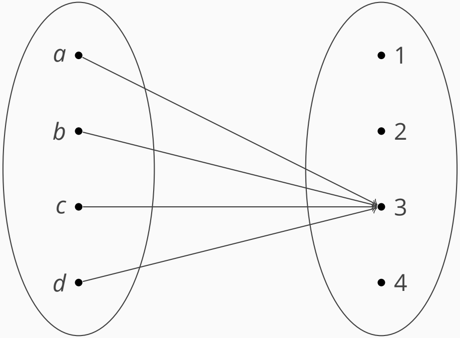
Functions are one of the most important building blocks in Computer Science!

Visualizing functions

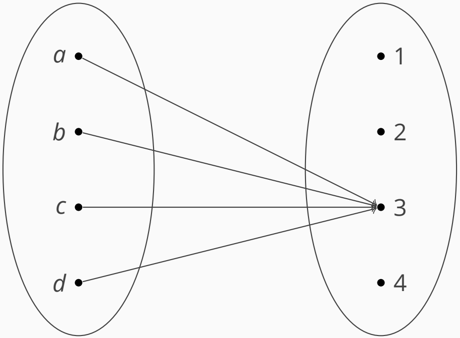


Here we can visualise a function from $f : \{a, b, c, d\} \longrightarrow \{1, 2, 3, 4\}$

Is this a function?

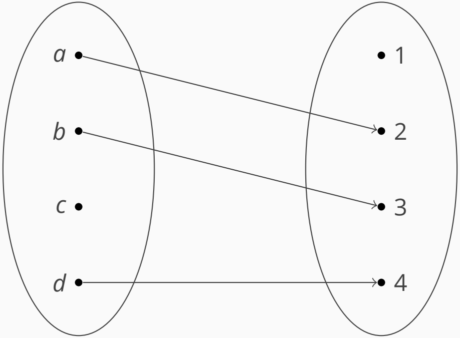


Is this a function?

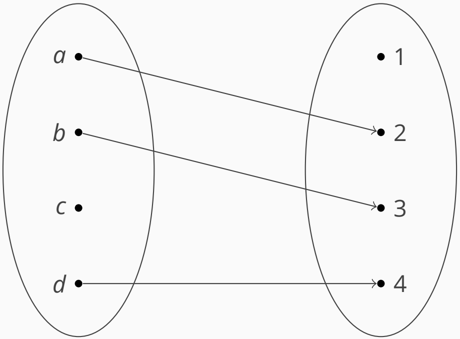


Yes! All inputs have exactly one output associated with them.

Is this a function?



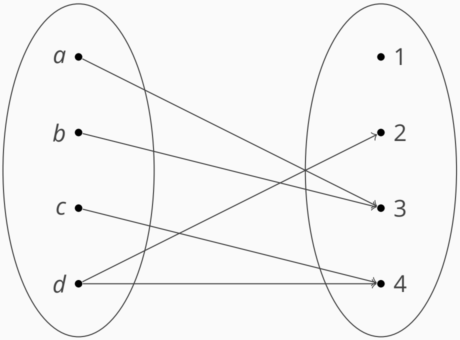
Is this a function?



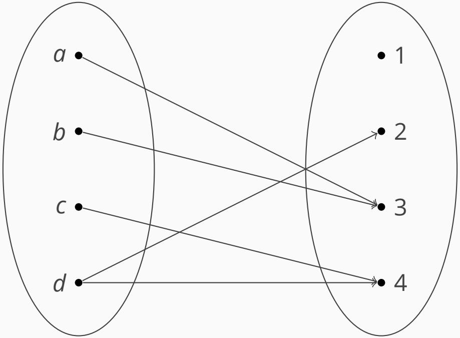
No! There is no output associated with the input c.

We sometimes refer to such ‘incomplete’ functions as *partial functions*.

Is this a function?



Is this a function?



No! There is more than one output associated with the input d.

This is an example of a *relation* – which we'll cover in the next lecture.

Terminology

Given a function $f : A \longrightarrow B$ we introduce the following terminology:

- We call the set A the **domain** of the function;
- The set B is the **codomain** of the function;
- If a function takes more than one argument, `f : A × B × C → D` we refer to the number of arguments as the **arity**.
- A function with two arguments is sometimes called a **binary function**; often we use *infix* notation, writing `x + y` rather than `+(x,y)`.
- The **range** of f is the subset of B that f can produce:

$$\text{range}(f) = \{f(a) : a \in A\}$$

Question: Give an example of a function whose range and codomain are different?

Associativity and precedence

We repeat this construction more than once.

A set of built from $(A \longrightarrow (B \longrightarrow C))$ is a function that, given an $a \in A$, returns a new function $B \longrightarrow C$.

The function arrow associates to the right and has lower precedence than cartesian product.

$A \times B \longrightarrow C \longrightarrow D$

Should be bracketed as:

$(A \times B) \longrightarrow (C \longrightarrow D)$

You'll learn more about programming with such functions next year, when you take the course on *Functional Programming*.

Image and pre-image

Besides applying a function to a single element, we can consider what happens when we apply a function to **all** the elements of a set.

Given a function $f : A \to B$

- The **image** of a subset $S \subseteq A$ under f is the subset of B defined by:

$$f(S) = \{f(a) \ : \ a \in S\}$$

- The **preimage** of a subset $S \subseteq B$, denoted by f^{-1}, is defined by

$$f^{-1}(S) = \{a \ : \ a \in A \wedge f(a) \in S\}$$

Example: square root

Consider the square root function on real numbers:

sqrt : $\mathbb{R} \to \mathbb{R}$

Question: What is the image sqrt on the set {1,4,9,16}?

Question: And what is the preimage of sqrt on the set {1,4,9,16}?

Graphs

Given a function $f : A \longrightarrow B$ we can define the following subset of $A \times B$:

$$G = \{(a, f(a)) \ : \ a \in A\}$$

This is sometimes called the **graph** of a function.

Question: What function has the following graph?

{ (0,0), (1,2), (2,4), (3,6), ...}

Graphs

Given a function f : A $\longrightarrow$ B we can define the following subset of A × B:

$$G = \{(a, f(a)) \ : \ a \in A\}$$

This is sometimes called the **graph** of a function.

Question: What function has the following graph?

{ (0,0), (1,2), (2,4), (3,6), ...}

For example, one choice might be:

```
double(x) = x + x
```

Special functions

- On any set A, we can define the **identity function** $\mathrm{id} : A \longrightarrow A$ as follows:

$$\mathrm{id}(x) = x$$

- For any subset S of A, we can define the **characteristic function**, typically denoted by

  $\chi : A \longrightarrow \{0,1\}$

  that returns true precisely when its argument is in S:

$$\chi(a) = \begin{cases} 1 & \text{when } a \in S \\ 0 & \text{when } a \notin S \end{cases}$$

Properties of functions: injective

A function $f : A \longrightarrow B$ is called **injective** or **one-to-one**

if for all $a \in A$ and $a' \in A$, whenever $f(a) = f(a')$ then $a = a'$

In other words, no two different elements of A are mapped to the same element of B.

Properties of functions: injective

A function $f : A \longrightarrow B$ is called **injective** or **one-to-one**

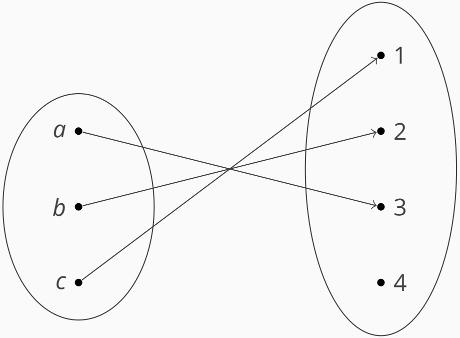
if for all $a \in A$ and $a' \in A$, whenever $f(a) = f(a')$ then $a = a'$

In other words, no two different elements of A are mapped to the same element of B.

Even more precisely, f is injective if

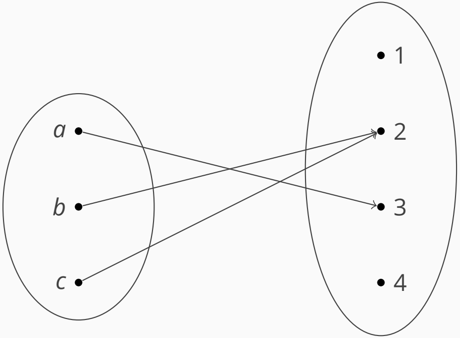
$$\forall a \in A \, \forall a' \in A \quad f(a) = f(a') \quad \Rightarrow \quad a = a'$$

Injective functions



This function is injective – each output has at most one input associated with it.

Not an injective function



This function is **not** injective – the both b and c are mapped to 2.

Examples

Examples and non-examples of injective functions:

- square : $\mathbb{N} \to \mathbb{N}$ is injective
- length : String $\to \mathbb{N}$ is **not** injective.

Question: Why not?

Examples

Examples and non-examples of injective functions:

- square : $\mathbb{N} \to \mathbb{N}$ is injective
- length : String $\to \mathbb{N}$ is **not** injective.

Question: Why not?

We have length(“a”) = length(“b”) but “a” and “b” are different!

Properties of functions: surjective

A function $f : A \rightarrow B$ is called **surjective** or **onto** if for all elements $b \in B$, there is an $a \in A$ such that $f(a) = b$.

In other words, each element of B has at least one $a \in A$ that is mapped to it by f.

Properties of functions: surjective

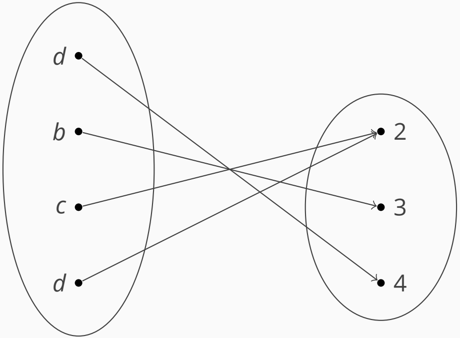
A function $f : A \to B$ is called **surjective** or **onto** if for all elements $b \in B$, there is an $a \in A$ such that $f(a) = b$.

In other words, each element of B has at least one $a \in A$ that is mapped to it by f.

Even more precisely, f is injective if

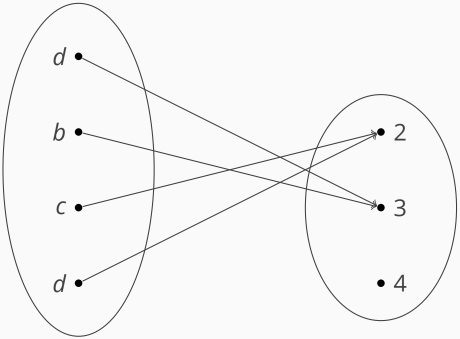
$$\forall\, b \in B\ \exists\, a \in A \quad f(a) = b$$

Surjective functions



This function is surjective – each number in the codomain has an incoming arrow (or more precisely, for each number *n* in the codomain there is a letter *c* such that $f(c) = n$).

Surjective functions



This function is not surjective – there is no letter mapped to 4.

Examples and non-examples of surjective functions

Examples:

- length : String $\to \mathbb{N}$ is surjective
- square : $\mathbb{N} \to \mathbb{N}$ is **not** surjective

Question: Why is square not surjective?

Examples and non-examples of surjective functions

Examples:

- length : String $\longrightarrow \mathbb{N}$ is surjective
- square : $\mathbb{N} \longrightarrow \mathbb{N}$ is **not** surjective

Question: Why is square not surjective?

There is no natural number *n* such that square(*n*) = 3.

Examples

Question: Which functions are injective? Which are surjective?

- $\text{sort} : \text{Array} \longrightarrow \text{Array}$
- $\text{isEven} : \mathbb{N} \longrightarrow \text{Bool}$
- $\text{halve} : \mathbb{R} \longrightarrow \mathbb{R}$
- $\text{square} : \mathbb{R} \longrightarrow \mathbb{R}$

Properties of functions: bijective

A function $f : A \rightarrow B$ that is both **injective** and **surjective** is called **bijective**.

Since f is surjective, every element b $\in$ B there is some element $a \in A$ such that $f(a) = b$.

Since f is injective, this element is **unique**.

This suggests that we can define a new function $f^{-1} : B \rightarrow A$, that *inverts* f.

That is, $f^{-1}(b) = a$ exactly when $f(a) = b$.

The function f^{-1} is also a bijection; inverting f twice yields our original f.

Bijections, injections, surjections: why care?

- If you're developing a cryptographic function `encode : String` $\longrightarrow$ `String` – you really want to be sure that there is an inverse function `decode : String` $\longrightarrow$ `String` – hence `encode` should be **bijective**.

Bijections, injections, surjections: why care?

- If you're developing a cryptographic function `encode : String` $\to$ `String` – you really want to be sure that there is an inverse function `decode : String` $\to$ `String` – hence `encode` should be **bijective**.

- If you're writing a function that lets you save the current state of a Reversi game, you want to make sure that every saved file corresponds to exactly one game state – the saving should be **injective**!

Bijections, injections, surjections: why care?

- If you're developing a cryptographic function `encode : String` $\rightarrow$ `String` – you really want to be sure that there is an inverse function `decode : String` $\rightarrow$ `String` – hence `encode` should be **bijective**.

- If you're writing a function that lets you save the current state of a Reversi game, you want to make sure that every saved file corresponds to exactly one game state – the saving should be **injective**!

- Suppose I want to number the elements of a set A. One way to do so is to define a function that maps each number $n \in \mathbb{N}$ to an element of A. But this function needs to be **surjective** – otherwise there might be elements of A that are not numbered.

Example

Question: Consider the function f : $\mathbb{Q} \rightarrow \mathbb{Q}$ defined as follows

$$f(x) = 2x + 3$$

Is it injective? Surjective? Bijective? If so, what is its inverse?

Example

Question: Consider the function f : $\mathbb{Q} \rightarrow \mathbb{Q}$ defined as follows

$$f(x) = 2x + 3$$

Is it injective? Surjective? Bijective? If so, what is its inverse?

Yes. Yes. Yes. Its inverse is:

$$f^{-1}(x) = (x - 3)/2$$

Doing these proofs is too much like hard work. Aren't there more general results we can use?

Example

Question: Consider the function f : $\mathbb{Q} \rightarrow \mathbb{Q}$ defined as follows

$$f(x) = 2x + 3$$

Is it injective? Surjective? Bijective? If so, what is its inverse?

Yes. Yes. Yes. Its inverse is:

$$f^{-1}(x) = (x - 3)/2$$

Doing these proofs is too much like hard work. Aren't there more general results we can use?

The function f is *composed* of smaller pieces. In what follows, we'll describe *function composition* and how it preserves the key properties of functions.

Function composition

Given two functions $f : A \longrightarrow B$ and $g : B \longrightarrow C$, we can **compose** them to define a new function

$$g \circ f : A \longrightarrow C$$

$$(g \circ f)(a) = g(f(a))$$

Note the order! The composition $g \circ f$ applies g *after* f.

Function composition: preserves properties

- If both $f : A \longrightarrow B$ and $g : B \longrightarrow C$ are both injective, then so is $(g \circ f)$;
- If both $f : A \longrightarrow B$ and $g : B \longrightarrow C$ are both surjective, then so is $(g \circ f)$;
- If both $f : A \longrightarrow B$ and $g : B \longrightarrow C$ are both bijective, then so is $(g \circ f)$;

Question: How do you prove these properties?

Function composition: consequences

To show a 'complicated' function such as

$f(x) = 2x + 3$

is bijective, it suffices to show that:

$g(x) = 2x$

$h(x) = x + 3$

are both bijective and for all x, $(h \circ g)(x) = f(x)$.

Function composition: consequences

To show a 'complicated' function such as

$f(x) = 2x + 3$

is bijective, it suffices to show that:

$g(x) = 2x$

$h(x) = x + 3$

are both bijective and for all x, $(h \circ g)(x) = f(x)$.

Reasoning about functions is inherently *compositional* – the behaviour of a compound function is entirely determined by the behaviour of its parts.

Function composition: identity

Note that the identity function is the *unit of composition operator*.

That is, forall $f : A \longrightarrow B$ and $idB : B \longrightarrow B$ and $idA : A \longrightarrow A$, we have:

$$idB \circ f = f = f \circ idA$$

The identity behaves like 0 and addition, or 1 and multiplication, or “” and string concatenation,...

Partial functions and restrictions

If $A \subseteq A'$ and $f : A' \longrightarrow B$, we can define a function $A \longrightarrow B$ as follows:

$$f|_A(x) = f(x)$$

That is, we *restrict* f to only work on the subset A of A'.

Partial functions

The functions $f : A \longrightarrow B$ we have seen so far are sometimes referred to as **total** functions, that is, they assign a value in B to every element of A.

In practice, many functions we work with in Computer Science are not total, but can fail for many different reasons:

- division by zero;
- invalid or unexpected inputs;
- square roots of negative numbers;
- non-terminating loops;
- corrupted data;
- ...

Partial functions

A partial function $f : A \longrightarrow B$ is only defined for some subset $A' \subseteq A$.

On other values, it will not produce an output in B.

Examples include:

- $inv(x) = 1/x$
- $toInt : String \longrightarrow Int32$
- $first : List \longrightarrow Int$

These functions may all fail on some inputs.

Case study: animations

Suppose we have a colour display with dimensions 1680×1050. Each pixel is colored with an RGB value between 0 and 255.

Question: How can you model the current screen?

Question: What about animations, built up from n different frames?

Case study: animations

Suppose we have a colour display with dimensions 1680×1050. Each pixel is colored with an RGB value between 0 and 255.

Question: How can you model the current screen?

Question: What about animations, built up from n different frames?

Define $RGB = \{0..255\} \times \{0..255\} \times \{0..255\}$

We can model the current screen contents as:

$S = \{0..1680\} \times \{0..1050\} \longrightarrow RGB$

Animations then correspond to:

$\{0, 1, 2, ..., n\} \longrightarrow S$

That is, for any given point in time, describe the contents of the screen.

Cardinality

For any finite set A, we define the **cardinality** of A as its number of elements, usually written $|A|$.

For example, the cardinality of the set {Alice, Bob, Carroll, David, Eve } is 5.

Cardinality

For any finite set A, we define the **cardinality** of A as its number of elements, usually written $|A|$.

For example, the cardinality of the set {Alice, Bob, Carroll, David, Eve } is 5.

This notion of cardinality, mapping sets to numbers, is fine for **finite** sets – but what is the cardinality of the set of all natural numbers?

We cannot easily compare infinite sets in this fashion.

Comparing the size of sets

Rather than talk about the exact number of elements in a set, we can compare two sets A and B by constructing functions between them.

If we can find a **bijection** between A and B, then we consider A and B to have the same size, written $A \simeq B$.

This notion works for both finite **and** infinite sets.

It generalizes our previous notion of cardinality, as for all *finite* sets A and B we have that $|A| = |B| \quad \Leftrightarrow \quad A \simeq B$

Examples

- The set of weekdays {Monday, Tuesday, ..., Sunday } $\simeq$ {0,..,6}
- Let $\mathbb{N}_{>0}$ be the set of all natural numbers strictly greater than 0. Then $\mathbb{N}_{>0} \simeq \mathbb{N}$
- Let E be the set of even numbers, {0,2,4,..}. Then $\mathbb{N} \simeq$ E. Why?
- **Theorem:** For any finite set A, there is no set B such that $A \subset B$ and $|A| = |B|$.

Examples

- The set of weekdays {Monday, Tuesday, ..., Sunday } $\simeq$ {0,..,6}
- Let $\mathbb{N}_{>0}$ be the set of all natural numbers strictly greater than 0. Then $\mathbb{N}_{>0} \simeq \mathbb{N}$
- Let *E* be the set of even numbers, {0,2,4,..}. Then $\mathbb{N} \simeq$ E. Why?
- **Theorem:** For any finite set *A*, there is no set *B* such that $A \subset B$ and $|A| = |B|$.

So do all infinite sets have the same cardinality?

Relative cardinalities

Before we can answer the question on the previous slide, we need to introduce several new concepts.

- $A \preceq B$ if there exists an *injective* function from A to B;
- $A \succeq B$ if there exists an *surjective* function from A to B;

Theorem For any sets A and B, $A \preceq B$ if and only if $B \succeq A$.

Theorem For any sets A and B, if $A \preceq B$ and $B \preceq A$ then $A \simeq B$.

The proofs are not entirely trivial... Remember, these statements say something about the existence of a function rather than comparing numbers!

Proof

Theorem For any sets A and B, $A \preceq B$ if and only if $B \succeq A$.

Proof

Suppose $A \preceq B$, that is we have an injection $f : A \longrightarrow B$.

We need to find a surjection $g : B \longrightarrow A$.

Theorem For any sets A and B, $A \preceq B$ if and only if $B \succeq A$.

Proof

Suppose $A \preceq B$, that is we have an injection $f : A \to B$.

We need to find a surjection $g : B \to A$.

Choose $a \in A$ be an arbitrary element of A.

We now construct $g(b)$ as follows:

- if there is an $x \in A$ such that $f(x) = b$ then choose $g(b) = x$. Note that because f is injective, this choice for x is unique.
- otherwise, $g(b) = a$.

This function is surjective because for each element $a \in A$, we have that $a = g(f(a))$.

Theorem For any sets A and B, $A \preceq B$ if and only if $B \succeq A$.

Proof

Suppose $B \succeq A$, that is we have a surjection $g : B \to A$.

We need to find an injection $f : A \to B$.

Question

How should we construct the desired f?

Proof

Theorem For any sets A and B, $A \preceq B$ if and only if $B \succeq A$.

Proof

Suppose $B \succeq A$, that is we have a surjection $g : B \to A$.

We need to find an injection $f : A \to B$.

Question

How should we construct the desired f?

Because g is surjective, we know that for each $a \in A$ we have at least one value $b \in B$ such that $g(b) = a$; hence we can define f to map each value $a \in A$ to such $b \in B$.

Schröder-Bernstein theorem

Theorem For any sets A and B, if $A \preceq B$ and $B \preceq A$ then $A \simeq B$.

The proof is a good example of something that seems obvious – but is surprisingly hard to construct the desired bijection.

Homework

Go through the proof in the book and identify the proof strategies being used.

Terminology

- A set A is **finite** if there is a number n such that $A \simeq \{1, .., n\}$.
- A set A is **countably infinite** if $\mathbb{N} \simeq A$ – that is, if there is an bijection between $\mathbb{N}$ and A.
- A set that is finite or countably infinite is said to be **countable**.
- A set that is not countable is called **uncountable**.

What sets are uncountable?

Uncountable sets?

- Clearly, $\mathbb{N}$ is countable as the identity function is a bijection between $\mathbb{N}$ and itself.
- What about the set of all integers, $\mathbb{Z}$?

Uncountable sets?

- Clearly, $\mathbb{N}$ is countable as the identity function is a bijection between $\mathbb{N}$ and itself.
- What about the set of all integers, $\mathbb{Z}$?

We can find a bijection as follows:

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

$$\begin{array}{lccccc} \mathbb{N} = \{ & 0, & 1\,, & 2\,, & 3\,, & 4\ldots\} \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathbb{Z} = \{ & 0 & 1\,, & \text{-}1\,, & 2, & \text{-}2\ldots\} \end{array}$$

is a bijection!

Question

What is the inverse of this function?

Rationals

What about the rational numbers, $\mathbb{Q}$? Are these uncountable?

Rationals

What about the rational numbers, $\mathbb{Q}$? Are these uncountable?

$$
\begin{array}{rcccccccc}
\mathbb{N} = \{ & 0, & 1\,, & 2\,, & 3\,, & 4, & 5, & \ldots\} \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
\mathbb{N} \times \mathbb{N} = \{ & (0,0) & (1,0)\,, & (0,1)\,, & (0,2)\,, & (1,1), & (2,0) & \ldots\}
\end{array}
$$

Rationals

What about the rational numbers, $\mathbb{Q}$? Are these uncountable?

$$
\begin{array}{rccccccc}
\mathbb{N} = \{ & 0, & 1\,, & 2\,, & 3\,, & 4, & 5, & \ldots\} \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
\mathbb{N} \times \mathbb{N} = \{ & (0,0) & (1,0)\,, & (0,1)\,, & (0,2)\,, & (1,1), & (2,0) & \ldots\}
\end{array}
$$

A diagram helps:

$$
\begin{array}{cccccc}
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & \ldots \\
(1,0) & (1,1) & (1,2) & (1,3) & (1,4) & \ldots \\
(2,0) & (2,1) & (2,2) & (2,3) & (2,5) & \ldots \\
\ldots & & & & &
\end{array}
$$

The reals are uncountable

The set of real numbers between 0 and 1 are uncountable.

Proof: Suppose we have a bijective function that enumerates all the real numbers one by one:

$$
\begin{aligned}
0 &\mapsto 0.d_{00}\ d_{01}\ d_{02}\ d_{03}\ \ldots \\
1 &\mapsto 0.d_{10}\ d_{11}\ d_{12}\ d_{13}\ \ldots \\
2 &\mapsto 0.d_{20}\ d_{21}\ d_{22}\ d_{23}\ \ldots \\
3 &\mapsto 0.d_{30}\ d_{31}\ d_{32}\ d_{33}\ \ldots
\end{aligned}
$$

We can construct a real number $r = 0.r_0 r_1 r_2 r_3 \ldots$ as follows:

$$r_i = (d_{ii} + 1) \text{mod} 10$$

By construction, r cannot occur in the image of our 'bijection' – hence no such bijection can exist.

Other uncountable sets

Theorem For any set A, there is no surjection from A to $P(A)$.

(Proof left as one of the exercises)

Other uncountable sets

Theorem For any set A, there is no surjection from A to $P(A)$.

(Proof left as one of the exercises)

In particular, $\mathbb{N} \prec P(\mathbb{N})$

Why functions?

Using functions we can model quite complicated data structure *precisely* and *unambiguously* in a few lines.

Functions are *compositional* – we can build bigger functions from smaller pieces.

Yet easy to reason about mathematically.

Functions are one of the key concepts we'll learn in this course – and form an important foundation for much of Computer Science.

Today

- Proof strategies for quantifiers
- Functions and their properties

Next time

Generalizing functions to **relations**.

Material

- Modelling Computer Systems – Chapter 6 (excluding section 6.5)