Mathematics for Informatics

Carlos Areces and Patrick Blackburn

areces@loria.fr blackbur@loria.fr

http://www.loria.fr/~areces http://www.loria.fr/~blacbur

INRIA Lorraine
Nancy, France

2007/2008

Data types in $\mathscr{S}$

We saw that the only data type i $\mathscr{S}$ are the natural numbers

But we can simulate other types. For example, we represented the type Bool using 1 (for true) and 0 (for false).

Today we will codify,

- pairs of natural numbers
- finite sequences of natural numbers

Codifying pairs

We define the following primitive recursive function:

$$\langle x, y \rangle = 2^x(2 \cdot y + 1) \dot{-} 1$$

Note that $2^x(2 \cdot y + 1) \neq 0$.

Proposition

there is a unique solution (x, y) to the equation $\langle x, y \rangle = z$.

Proof.

- x is the maximum number such that $2^x | (z + 1)$
- $y = ((z + 1)/2^x - 1)$ div 2

□

Proyection functions for pairs

The proyections for the pair $z = \langle x, y \rangle$ are

- $l(z) = x$
- $r(z) = y$

Proposition

Proyections are primitive recursive functions.

Proof.

As $x, y < z + 1$ we have that

- $l(z) = \min_{x \leq z} ((\exists y)_{\leq z}\ z = \langle x, y \rangle)$
- $r(z) = \min_{y \leq z} ((\exists x)_{\leq z}\ z = \langle x, y \rangle)$

$\square$

Proyection functions for pairs

The proyections for the pair $z = \langle x, y \rangle$ are

- $l(z) = x$
- $r(z) = y$

Proposition

Proyections are primitive recursive functions.

Proof.

As $x, y < z + 1$ we have that

- $l(z) = \min_{x \leq z} ((\exists y)_{\leq z}\ z = \langle x, y \rangle)$
- $r(z) = \min_{y \leq z} ((\exists x)_{\leq z}\ z = \langle x, y \rangle)$

$\square$

For example,

- $\langle 2, 5 \rangle = 2^2(2 \cdot 5 + 1) \dot{-} 1 = 43$
- $l(43) = 2$
- $r(43) = 5$

Codifying sequences

The Gödel number for the sequence

$$a_1, \ldots, a_n$$

is the number

$$[a_1, \ldots, a_n] = \prod_{i=1}^{n} p_i^{a_i}.$$

For example, the Gödel number of the sequence

$$1, 3, 3, 2, 2$$

is

$$[1, 3, 3, 2, 2] = 2^1 \cdot 3^3 \cdot 5^3 \cdot 7^2 \cdot 11^2 = 40020750$$

Properties of the codification of sequences

Theorem

If $[a_1, \ldots, a_n] = [b_1, \ldots, b_n]$ *then* $a_i = b_i$ *for each* $i \in \{1, \ldots, n\}$.

Proof.

Because of the unique factorization into primes. $\square$

Properties of the codification of sequences

Theorem

If $[a_1, \ldots, a_n] = [b_1, \ldots, b_n]$ *then* $a_i = b_i$ *for each* $i \in \{1, \ldots, n\}$.

Proof.

Because of the unique factorization into primes. $\square$

Observe that

$$[a_1, \ldots, a_n] = [a_1, \ldots, a_n, 0] = [a_1, \ldots, a_n, 0, 0] = \ldots$$

but

$$[a_1, \ldots, a_n] \neq [0, a_1, \ldots, a_n]$$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \ldots, a_n]$ are

- $x[i] = a_i$
- $|x| = $ lenght of x

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x} (x[i] \neq 0 \land (\forall j)_{\leq x}(j \leq i \lor x[i] = 0))$

$\square$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \ldots, a_n]$ are

- $x[i] = a_i$
- $|x| =$ lenght of x

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x} (x[i] \neq 0 \wedge (\forall j)_{\leq x}(j \leq i \vee x[i] = 0))$

$\square$

For example,

- $[1, 3, 3, 2, 2][2] = 4$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \ldots, a_n]$ are

- $x[i] = a_i$
- $|x| =$ lenght of x

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x} (x[i] \neq 0 \wedge (\forall j)_{\leq x}(j \leq i \vee x[i] = 0))$

□

For example,

- $[1, 3, 3, 2, 2][2] = 4$
- $[1, 3, 3, 2, 2][6] = 0$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \ldots, a_n]$ are

- $x[i] = a_i$
- $|x| = \text{lenght of } x$

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x}\,(x[i] \neq 0 \wedge (\forall j)_{\leq x}(j \leq i \vee x[i] = 0))$

$\square$

For example,

- $[1, 3, 3, 2, 2][2] = 4$
- $[1, 3, 3, 2, 2][6] = 0$
- $|[1, 3, 3, 2, 2]| = 5$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \dots, a_n]$ are

- $x[i] = a_i$
- $|x| =$ lenght of x

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x} (x[i] \neq 0 \wedge (\forall j)_{\leq x}(j \leq i \vee x[i] = 0))$

$\square$

For example,

- $[1, 3, 3, 2, 2][2] = 4$
- $[1, 3, 3, 2, 2][6] = 0$
- $|[1, 3, 3, 2, 2]| = 5$
- $|[1, 3, 3, 2, 2, 0]| = |[1, 3, 3, 2, 2, 0, 0]| = 5$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \ldots, a_n]$ are

- $x[i] = a_i$
- $|x| =$ lenght of x

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x} (x[i] \neq 0 \wedge (\forall j)_{\leq x}(j \leq i \vee x[i] = 0))$

$\square$

For example,

- $[1, 3, 3, 2, 2][2] = 4 = 40020750[2]$
- $[1, 3, 3, 2, 2][6] = 0 = 40020750[6]$
- $|[1, 3, 3, 2, 2]| = 5 = |40020750|$
- $|[1, 3, 3, 2, 2, 0]| = |[1, 3, 3, 2, 2, 0, 0]| = 5 = |40020750|$

Proyectors for sequences

The proyector functions for the sequence $x = [a_1, \ldots, a_n]$ are

- $x[i] = a_i$
- $|x| =$ lenght of x

Proposition

The proyector functions for sequences are primitive recursive.

Proof.

- $x[i] = \min_{t \leq x}(\neg p_i^{t+1} | x)$
- $|x| = \min_{i \leq x} \left(x[i] \neq 0 \wedge (\forall j)_{\leq x}(j \leq i \vee x[i] = 0)\right)$

$\square$

For example,

- $[1, 3, 3, 2, 2][2] = 4 = 40020750[2]$
- $[1, 3, 3, 2, 2][6] = 0 = 40020750[6]$
- $|[1, 3, 3, 2, 2]| = 5 = |40020750|$
- $|[1, 3, 3, 2, 2, 0]| = |[1, 3, 3, 2, 2, 0, 0]| = 5 = |40020750|$
- $x[0] = 0$ for any x
- $0[i] = 0$ for any i

Summing up

Theorem (Codifying pairs)

- $l(\langle x, y \rangle) = x, r(\langle x, y \rangle) = y$
- $z = \langle l(z), r(z) \rangle$
- $l(z), r(z) \leq z$
- *the codification and proyectors for pairs are p.r.*

Summing up

Theorem (Codifying pairs)

- $l(\langle x, y\rangle) = x, r(\langle x, y\rangle) = y$
- $z = \langle l(z), r(z)\rangle$
- $l(z), r(z) \leq z$
- *the codification and proyectors for pairs are p.r.*

Theorem (Codifying sequences)

- $[a_1, \ldots, a_n][i] = \begin{cases} a_i & \text{if } 1 \leq i \leq n \\ 0 & \text{otherwise} \end{cases}$
- *If* $n \geq |x|$ *then* $[x[1], \ldots, x[n]] = x$
- *the codification and proyectors for sequences are p.r.*

Codifying programs of $\mathscr{S}$

Remember that the instructions of $\mathscr{S}$ are:

1. $V \leftarrow V + 1$
2. $V \leftarrow V - 1$
3. IF $V \neq 0$ GOTO L'

Codifying programs of $\mathscr{S}$

Remember that the instructions of $\mathscr{S}$ are:

1. $V \leftarrow V + 1$
2. $V \leftarrow V - 1$
3. IF $V \neq 0$ GOTO L'

For convenience we are going to add a fourth instruction

4. $V \leftarrow V$: it does nothing

Observe that for any instruction

- it can be labeled or not by L
- it mentions exactly one variable V
- the IF construction always mentions a label L'

Codifying variables and labes in $\mathscr{S}$

Let's order the variables:

$$Y, X_1, Z_1, X_2, Z_2, X_3, Z_3, \ldots$$

Let's order the labels:

$$A, B, C, D, \ldots, Z, AA, AB, AC, \ldots, AZ, BA, BB, \ldots, BZ, \ldots$$

We write $\#(V)$ for the position that a variable V occupies in the list. Idem for $\#(L)$ with the label L.

For example,

- $\#(Y) = 1$
- $\#(X_2) = 4$
- $\#(A) = 1$
- $\#(C) = 3$

Codifying the instructions of $\mathscr{S}$

We codifyin the instruction I as

$$\#(I) = \langle a, \langle b, c \rangle \rangle$$

where

1. if I has a label L, then $a = \#(L)$; otherwise $a = 0$
2. if the variable mentioned in I is V then $c = \#(V) - 1$
3. if the instruction I is
   3.1 $V \leftarrow V$ then $b = 0$
   3.2 $V \leftarrow V + 1$ then $b = 1$
   3.3 $V \leftarrow V - 1$ then $b = 2$
   3.4 IF $V \neq 0$ GOTO L' then $b = \#(L') + 2$

Codifying the instructions of $\mathscr{S}$

We codifyin the instruction I as

$$\#(I) = \langle a, \langle b, c\rangle\rangle$$

where

1. if I has a label L, then $a = \#(L)$; otherwise $a = 0$
2. if the variable mentioned in I is V then $c = \#(V) - 1$
3. if the instruction I is
   3.1 $V \leftarrow V$ then $b = 0$
   3.2 $V \leftarrow V + 1$ then $b = 1$
   3.3 $V \leftarrow V - 1$ then $b = 2$
   3.4 IF $V \neq 0$ GOTO L' then $b = \#(L') + 2$

For example,

- $\#(X \leftarrow X + 1) = \langle 0, \langle 1, 1\rangle\rangle = \langle 0, 5\rangle = 10$
- $\#([A] \quad X \leftarrow X + 1) = \langle 1, \langle 1, 1\rangle\rangle = \langle 1, 5\rangle = 21$
- $\#(\text{IF } X \neq 0 \text{ GOTO } A) = \langle 0, \langle 3, 1\rangle\rangle = \langle 0, 23\rangle = 46$
- $\#(Y \leftarrow Y) = \langle 0, \langle 0, 0\rangle\rangle = \langle 0, 0\rangle = 0$

Any number x represent a unique instruction I.

Codifying the programs of $\mathscr{S}$

A program P is a (finite) list of instructions $I_1, \ldots, I_k$

We codify the program P as

$$\#(P) = [\#(I_1), \ldots, \#(I_k)] - 1$$

For exampel, for the program P

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \end{array}$$

we have

$$\#(P) = [\#(I_1), \#(I_2)] = [21, 46] = 2^{21} \cdot 3^{46} - 1$$

Ambiguities

We say that P

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \end{array}$$

has number $[21, 46]$.

Ambiguities

We say that P

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \end{array}$$

has number $[21, 46]$. But

$$[21, 46] = [21, 46, 0]$$

The same number could represent more than a program!

Ambiguities

We say that P

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \end{array}$$

has number $[21, 46]$. But

$$[21, 46] = [21, 46, 0]$$

The same number could represent more than a program!
Luckily, the program $[21, 46, 0]$ is

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \\ & Y \leftarrow Y \end{array}$$

and it is equivalent to P.

Ambiguities

We say that P

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \end{array}$$

has number $[21, 46]$. But

$$[21, 46] = [21, 46, 0]$$

The same number could represent more than a program!
Luckily, the program $[21, 46, 0]$ is

$$\begin{array}{ll} [A] & X \leftarrow X + 1 \\ & \text{IF } X \neq 0 \text{ GOTO } A \\ & Y \leftarrow Y \end{array}$$

and it is equivalent to P.

In any case, we eliminate this ambiguity stipulating that

the last instruction in a program cannot be $Y \leftarrow Y$

Under this condition, each number represent a unique program.

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0, 1]$ *is not enumerable.*

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0, 1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$r_1 = \quad 0, \quad r_{11} \quad r_{12} \quad r_{13} \quad r_{14} \quad \dots$$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0,1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{llllll} r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \ldots \\ r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \ldots \end{array}$$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0, 1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{llllll} r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\ r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\ r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \dots \end{array}$$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0, 1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$
\begin{array}{llllll}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\
r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \dots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & r_{44} & \dots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \dots \\
\vdots & & & & & & \ddots
\end{array}
$$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0,1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{rlllll}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\
r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \dots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & r_{44} & \dots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \dots \\
\vdots & & & & & & \ddots
\end{array}$$

Define the following number x

$$x = \quad 0, \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad \dots$$

with $x_i = (r_{ii} + 2) \mod 10$. $\square$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0, 1]$ is not enumerable.

Proof.

Suppose that it is. Then I can enumerate it:

$$
\begin{array}{rllllll}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\
r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \dots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & r_{44} & \dots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \dots \\
\vdots & & & & & & \ddots
\end{array}
$$

Define the following number x

$$x = \quad 0, \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad \dots$$

with $x_i = (r_{ii} + 2) \mod 10$. $\square$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0,1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{rcccccc}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \ldots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \ldots \\
r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \ldots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & r_{44} & \ldots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \ldots \\
\vdots & & & & & & \ddots
\end{array}$$

Define the following number x

$$x = \quad 0, \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad \ldots$$

with $x_i = (r_{ii} + 2) \mod 10$. $\square$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0,1]$ is not enumerable.

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{rllllll}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\
r_3 = & 0, & r_{31} & r_{32} & \color{red}{r_{33}} & r_{34} & \dots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & r_{44} & \dots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \dots \\
\vdots & & & & & & \ddots
\end{array}$$

Define the following number x

$$x = \quad 0, \quad x_1 \quad x_2 \quad \color{red}{x_3} \quad x_4 \quad \dots$$

with $x_i = (r_{ii} + 2) \mod 10$. $\square$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0, 1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{rlllll}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\
r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \dots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & \color{red}{r_{44}} & \dots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \dots \\
\vdots & & & & & & \ddots
\end{array}$$

Define the following number x

$$x = \quad 0, \quad x_1 \quad x_2 \quad x_3 \quad \color{red}{x_4} \quad \dots$$

with $x_i = (r_{ii} + 2) \mod 10$. $\square$

Enumerable and not enumerable sets

The natural numbers are enumerable.

Theorem (Cantor)

The set of real numbers in $[0,1]$ *is not enumerable.*

Proof.

Suppose that it is. Then I can enumerate it:

$$\begin{array}{llllllll}
r_1 = & 0, & r_{11} & r_{12} & r_{13} & r_{14} & \dots \\
r_2 = & 0, & r_{21} & r_{22} & r_{23} & r_{24} & \dots \\
r_3 = & 0, & r_{31} & r_{32} & r_{33} & r_{34} & \dots \\
r_3 = & 0, & r_{41} & r_{42} & r_{43} & r_{44} & \dots \\
\vdots & & & & & & \\
r_k = & 0, & r_{k1} & r_{k2} & r_{k3} & r_{k4} & \dots \\
\vdots & & & & & & \ddots
\end{array}$$

Define the following number x

$$x = \quad 0, \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad \dots$$

with $x_i = (r_{ii} + 2) \mod 10$. Then x is not a real. $\square$

There are non computable functions

There are as many total functions $f : \mathbb{N} \to \{0, \dots, 9\}$ as there are real numbers in $[0, 1]$.
We can codify the function f as

$$0, \quad f(0) \quad f(1) \quad f(2) \quad f(3) \quad \dots$$

- every function can be represented as a real in $[0, 1]$
- every real in $[0, 1]$ represents a unique function

There are non computable functions

There are as many total functions $f : \mathbb{N} \to \{0, \ldots, 9\}$ as there are real numbers in $[0, 1]$.
We can codify the function f as

$$0, \quad f(0) \quad f(1) \quad f(2) \quad f(3) \quad \ldots$$

- every function can be represented as a real in $[0, 1]$
- every real in $[0, 1]$ represents a unique function

In general (talking informally),

- there are as many functions $\mathbb{N} \to \mathbb{N}$ as there are reals

There are non computable functions

There are as many total functions $f : \mathbb{N} \to \{0, \ldots, 9\}$ as there are real numbers in $[0, 1]$.
We can codify the function f as

$$0, \quad f(0) \quad f(1) \quad f(2) \quad f(3) \quad \ldots$$

- every function can be represented as a real in $[0, 1]$
- every real in $[0, 1]$ represents a unique function

In general (talking informally),

- there are as many functions $\mathbb{N} \to \mathbb{N}$ as there are reals
- there are more functions $\mathbb{N} \to \mathbb{N}$ than natural numbers

There are non computable functions

There are as many total functions $f : \mathbb{N} \to \{0, \ldots, 9\}$ as there are real numbers in $[0, 1]$.
We can codify the function f as

$$0, \quad f(0) \quad f(1) \quad f(2) \quad f(3) \quad \ldots$$

- every function can be represented as a real in $[0, 1]$
- every real in $[0, 1]$ represents a unique function

In general (talking informally),

- there are as many functions $\mathbb{N} \to \mathbb{N}$ as there are reals
- there are more functions $\mathbb{N} \to \mathbb{N}$ than natural numbers
- there are as many programs as natural numbers

There are non computable functions

There are as many total functions $f : \mathbb{N} \to \{0, \ldots, 9\}$ as there are real numbers in $[0, 1]$.
We can codify the function f as

$$0, \quad f(0) \quad f(1) \quad f(2) \quad f(3) \quad \ldots$$

- every function can be represented as a real in $[0, 1]$
- every real in $[0, 1]$ represents a unique function

In general (talking informally),

- there are as many functions $\mathbb{N} \to \mathbb{N}$ as there are reals
- there are more functions $\mathbb{N} \to \mathbb{N}$ than natural numbers
- there are as many programs as natural numbers
- there are as many partially computable functions as natural numbers

There are non computable functions

There are as many total functions $f : \mathbb{N} \to \{0, \ldots, 9\}$ as there are real numbers in $[0, 1]$.
We can codify the function f as

$$0, \quad f(0) \quad f(1) \quad f(2) \quad f(3) \quad \ldots$$

- every function can be represented as a real in $[0, 1]$
- every real in $[0, 1]$ represents a unique function

In general (talking informally),

- there are as many functions $\mathbb{N} \to \mathbb{N}$ as there are reals
- there are more functions $\mathbb{N} \to \mathbb{N}$ than natural numbers
- there are as many programs as natural numbers
- there are as many partially computable functions as natural numbers
- there should be functions $\mathbb{N} \to \mathbb{N}$ which are not computable