# Undecidability

## What is undecidability?

There is a specific problem that is unsolvable by an algorithm.

In one type of unsolvable problem, you are given a computer program and a precise specification of what the program is supposed to do. So you need to verify that the program performs as specified.

The undecidability of a specific language highlights the problem of determining whether a TM accepts a given input string.

Let $A_{TM} = \{< M,\ w > \ | M \, is \, a \, TM \, and \, M \, accepts \, w\}$

## The diagonalization method

The proof of undecidability is based on a technique called diagonalization, discovered by George Cantor. He observed that two finite sets have the same size if the elements of one set can be paired with the elements of the other set.

$$f \, is \, injective \, if \, f(a) \neq \ f(b) \, whenever \, a \neq b$$
$$f \, is \, surjective \, if \, f: A \rightarrow B \, and \, for \, every \, b \in B, \ there \, is \, an \, a \in A \, so \, that \, f(a) = b$$
$$f \, is \, injective \, and \, surjective \Rightarrow \ f \, bijective \, / \, correspondence$$

In a correspondence, every element of set A maps to a unique element of set B.

**Example:**

let N be {1, 2, 3 …}, the set of natural numbers<br>
let ε be the set of even natural numbers {2, 4, 6 …}

Using Cantor’s method the sets have the same size because of the mapping function from N to ε: $f(n) = 2n$.

| $n$ | $f(n)$ |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| $\vdots$ | $\vdots$ |

We can pair the elements inside the two sets, therefore the sets are of the same size.

# Q is countable

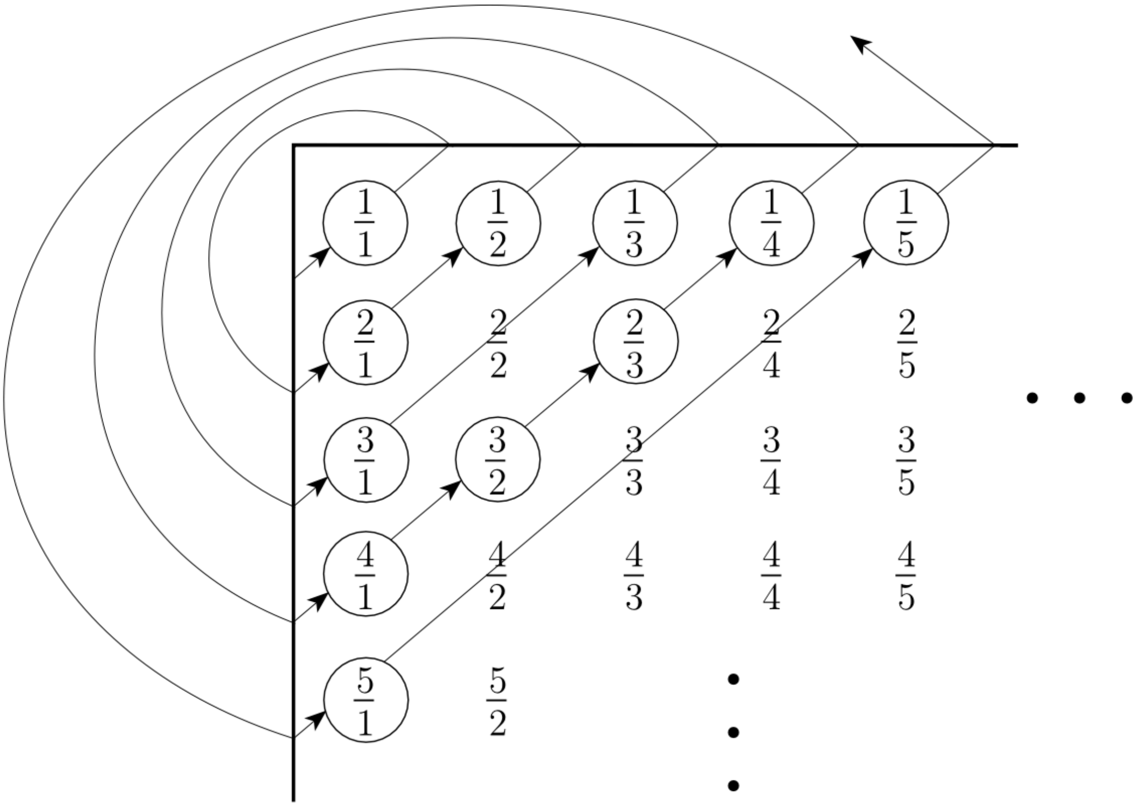
**A set is countable if either it is finite or has the same size as N.**

let $Q = \{\frac{m}{n},\ m,\ n \in N\}$, be the set of positive rational numbers.

At first it might seem that it is much bigger than N, thus uncountable, but using Cantor’s Method we can prove that they are equal, thus countable.

We create a matrix where $M[i, j] \ = \frac{i}{j}$.

We map the values from N to Q starting from the first diagonal, then the second, … and so on. We must make sure that we skip repetitive values.



$\frac{1}{1}$ $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$ $\frac{1}{5}$
$\frac{2}{1}$ $\frac{2}{2}$ $\frac{2}{3}$ $\frac{2}{4}$ $\frac{2}{5}$
$\frac{3}{1}$ $\frac{3}{2}$ $\frac{3}{3}$ $\frac{3}{4}$ $\frac{3}{5}$
$\frac{4}{1}$ $\frac{4}{2}$ $\frac{4}{3}$ $\frac{4}{4}$ $\frac{4}{5}$
$\frac{5}{1}$ $\frac{5}{2}$

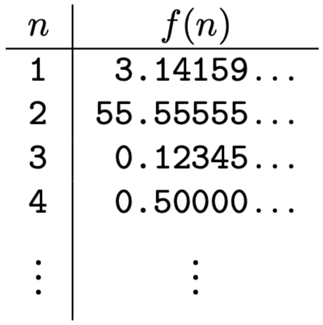
Using this proof, one might think that any two sets can be proven to have the same size, but that is not the case. Some sets are simply too big. Such sets are uncountable.

# R is uncountable

In order to show that R is uncountable, we show that there is no correspondence between N and R. The proof is by contradiction.

We suppose that a correspondence exists between N and R. We must find an $x \in R$ that is not paired with anything from N.

Our objective is to ensure that $x \neq f(n),\ \forall n \epsilon N$. We arbitrarily choose this initial mapping.



| $n$ | $f(n)$ |
|---|---|
| 1 | 3.14159... |
| 2 | 55.55555... |
| 3 | 0.12345... |
| 4 | 0.50000... |
| ⋮ | ⋮ |

We construct x by the following rules:

1. $x \epsilon (0, 1)$
2. $d(x, i) \neq d(f(i), i),\ i \in N, where$

   $d(x, i) \rightarrow returns\ the\ i - th\ decimal\ of\ x$

$let\ x = 0.4641...$

Continuing this way down the diagonal of the table for f we obtain all the digits of x. We know that x is not $f(n)\ \forall\ n \epsilon N$, because it differs from $f(n)$ in the n-th fractional digit.

So there is no correspondence between N and R. This proves that R is uncountable.

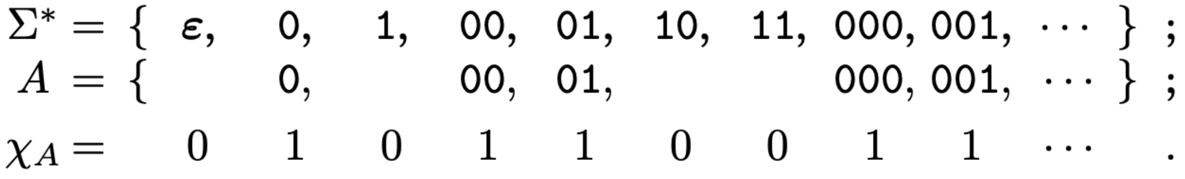
# Some languages are not Turing recognizable

The set of all TMs is countable because each TM M has an encoding into a string <M>.

The set of all infinite binary sequences is uncountable. Let B be the set of all infinite binary sequences. We can prove that B is uncountable with the same method used for the R set.

Let L be the set of all languages over alphabet Σ. We show that L is uncountable by giving a correspondence with B. Let $\Sigma^* = \{S1,\ S2,\ S3,\ ...\}$. Each language AϵL has a unique sequence in B. The i-th bit of that sequence is a 1 if $S_i \epsilon$ A, else is 0. This is called the characteristic sequence of A.

For example if A its the language of all strings that start with a 0 over alphabet {0, 1}, its characteristic sequence $\chi_A$ would be:



$$\begin{array}{rllllllllll} \Sigma^* = \{ & \varepsilon, & 0, & 1, & 00, & 01, & 10, & 11, & 000, & 001, & \cdots \} ; \\ A = \{ & & 0, & & 00, & 01, & & & 000, & 001, & \cdots \} ; \\ \chi_A = & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \cdots . \end{array}$$

The function $f: L \rightarrow B\ where\ f(A)$ equals the characteristic sequence of A, is bijective, hence is a correspondence. Therefore B is uncountable, L is uncountable as well.

Thus some languages are not Turing recognizable.

# $A_{TM}$ is undecidable

$$A_{TM} = \{< M, w > \mid M\ is\ a\ TM\ that\ accepts\ w\}$$

Suppose $A_{TM}$ is decidable. We will prove the contrary by contradiction. Define a new TM H, a decider, following the rule:

$$H\big(\langle M, w\rangle\big) = \begin{cases} accept & \text{if } M \text{ accepts } w \\ reject & \text{if } M \text{ does not accept } w. \end{cases}$$

Define another TM D, that has H as its subroutine. D takes another TM as input and it will return the opposite of what H returns.

$$D\big(\langle M\rangle\big) = \begin{cases} accept & \text{if } M \text{ does not accept } \langle M\rangle \\ reject & \text{if } M \text{ accepts } \langle M\rangle. \end{cases}$$

$$D\big(\langle D\rangle\big) = \begin{cases} accept & \text{if } D \text{ does not accept } \langle D\rangle \\ reject & \text{if } D \text{ accepts } \langle D\rangle. \end{cases}$$

Clear contradiction. Thus neither D nor H can exist.