The Halting Problem

The Halting Problem

We will show that **the problem of determining whether or not a given Turing machine accepts a given input is unsolvable**.

In other words, there is no Turing machine (and hence no algorithm) that decides

$$A_{\mathrm{TM}} = \{\langle M, w\rangle \mid M \text{ is a Turing machine and accepts } w\}.$$

(Later, we use this result to conclude that **the halting problem**, i.e., determining whether or not a given TM halts on a given input, is also unsolvable.)

The method of proof involves the technique of **diagonalization**, so we begin with this topic.

Diagonalization

A set is **countable** if either it is finite or it has the same size as $\mathcal{N}$; i.e., there is a **one-to-one mapping** from the set **onto** $\mathcal{N}$ (in other words, there is a **bijection** from the set to $\mathcal{N}$).

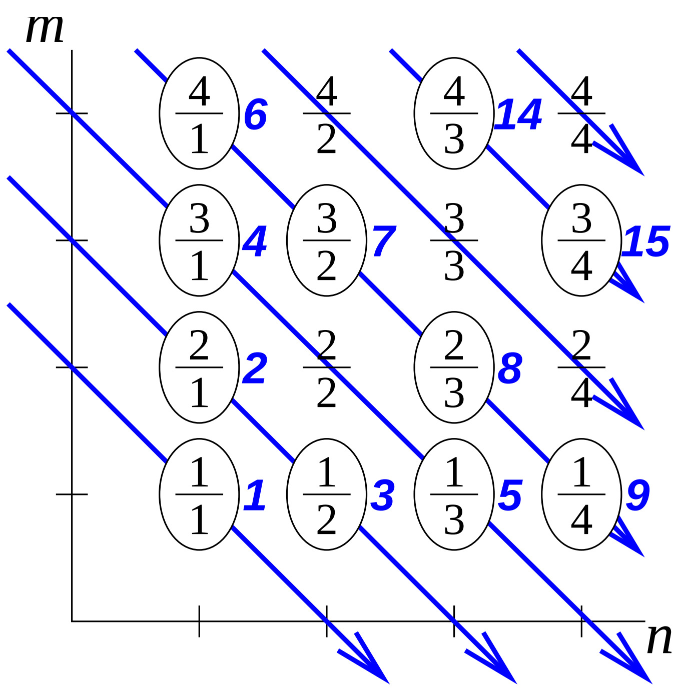
Fact. *Let $\mathcal{Q}$ be the set of all positive rational numbers and $\mathcal{R}$ the set of all positive real numbers. Then $\mathcal{Q}$ is countable while $\mathcal{R}$ is not.*

Proof For the former, each member of $\mathcal{Q}$ is expressed as a fraction $\frac{m}{n}$ such that $m, n \in \mathcal{N}$ and $\gcd(m, n) = 1$.

So we have only to come up with a bijection from $\mathcal{N}$ to the set $\{\frac{m}{n} \mid m, n \geq 1$ & $\gcd(m, n) = 1\}$.

$\mathcal{Q}$ is countable

For $p = 1, 2, \ldots$, visit the integral points on the line $m + n = p$ in the first quadrant of the xy-plane and count how many pairs (m, n) such that $\gcd(m, n) = 1$ have been seen.



$\mathcal{R}$ is not countable (proof by diagonalization)

For the latter, assume, for contradiction, that $\mathcal{R}$ is countable. Let f be a bijection from $\mathcal{R}$ to $\mathcal{N}$. For each $i \in \mathcal{N}$, let $r_i = f^{-1}(i)$. Define x to be the number between 0 and 1 defined as follows:

(*) **For every $i \in \mathcal{N}$, the ith digit of x after the decimal point in its decimal representation is that of r_i plus 1 (modulo 10).**

For example, if $r_1 = 3.\underline{1}4159, r_2 = 2.2\underline{3}606, r_3 = 1.73\underline{2}05, \ldots$, then $x = .243\ldots$,

This x is real. By assumption there is a unique k such that $x = f^{-1}(k)$. Then by definition

(!!) *the kth digit of x is the kth digit of x plus 1 modulo 10,*

which is a contradiction. So, $\mathcal{R}$ is not countable. ∎

An Immediate Application of Diagonalization

Corollary. *There is a language that is not Turing-recognizable.*

Proof The set of Turing machines is countable:

1. Fix an encoding scheme of Turing machines on an alphabet Σ.
2. **Go through all the strings in Σ**, e.g., in lexicographic order, and **assign numbers to all legal encodings** by counting **how many legal encodings have been seen so far.**

A language over Σ can be viewed as an infinite binary number $0.b_1 b_2 b_3 \cdots$, called the **characteristic sequence**, where for each $i \geq 1$, b_i corresponds to the membership of the ith string in the language. So the languages have the same cardinality as the set of binary reals between 0 and 1, which is uncountable. ∎

Theorem. *The language*

$$A_{\mathrm{TM}} = \{\langle M, w\rangle \mid M \text{ is a Turing machine and accepts } w\}$$

is not decidable.

Proof Assume that A_{TM} is decidable. Let T be a Turing machine that decides A_{TM}. Define D to be a machine that, on input w,

1. Checks whether w is a legal encoding of some Turing machine, say M. If not, D immediately rejects w.
2. Simulates T on $\langle M, \langle M\rangle\rangle$.
3. If T accepts, then D rejects; otherwise, it accepts.

Since T decides A_{TM} by assumption, it always halts; so does D.

Proof that A_{TM} is undecidable, cont'd

For every Turing machine M,

D **accepts** $\langle M \rangle \Leftrightarrow M$ **does not accept** $\langle M \rangle$

With $M = D$, we have

D **accepts** $\langle D \rangle \Leftrightarrow D$ **does not accept** $\langle D \rangle$**.**

This is a contradiction. ∎

Corollary. *$\overline{A_{\mathrm{TM}}}$ is not Turing-recognizable, and thus, not decidable.*

For this corollary we need the following fact.

Fact. *A language L is decidable if and only if both L and $\overline{L}$ are Turing-recognizable.*

Proof of Corollary A_{TM} is Turing-recognizable and is not decidable.
So, $\overline{A_{\mathrm{TM}}}$ is not Turing-recognizable ∎Corollary

Proof of Fact [$\Rightarrow$] Let L be decidable and let M be a Turing machine that decides L. By swapping q_{accept} and q_{reject} of M we get a Turing machine M' that decides $\overline{L}$. So both L and $\overline{L}$ are Turing-decidable, and thus, Turing-recognizable.

[$\Leftarrow$] Let L and $\overline{L}$ be recognized by TMs M_1 and M_2, respectively. Define a two-tape machine M that, on input x, does the following:

1. M copies x onto Tape 2.
2. M repeats the following until either M_1 or M_2 accepts:
   (a) M simulates one step of M_1 on Tape 1 then one step of M_2 on Tape 2.
3. M accepts x if M_1 has accepted and rejects if M_2 has accepted.

Then M decides L because for every x, at least one of the two machines halts on input x. ▮Fact