CS 381: Diagonalization and the Halting Problem

Church-Turing thesis Our intuitive notion of algorithms is equivalent to Turing machine algorithms. Another way to think of it is that any real-world computation can be translated into an equivalent computation done by a Turing machine. Turing discussed the notion of "effective calculability", which contains a set of properties (with different variants suggested over the years) that essentially define computation as the work that a human computing agent could do mechanically (mechanical calculability).

So far we discussed recognizable and decidable languages, and saw examples of decidable languages, like recognizing finite sequences of zeroes whose length is a power of 2. Next we will show that not all languages are decidable.

1 Undecidability

Consider the following language (also known as the *Halting Problem*):

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

Theorem 1.1. *A_{TM} is undecidable.*

Observation 1. *A_{TM} is Turing-recognizable. Why? Here is a machine to recognize A_{TM}:*

> Turing machine U: *On input $\langle M, w\rangle$, where M is a TM and w is a string:*
>
> 1. Simulate the machine M on input w.
> 2. If M reaches the accept state on w, then accept. If M reaches the reject state, reject.

But if M goes on forever, machine U also does. So U is not a decider for A_{TM}.

We will introduce some useful concepts first.

Definition 1 (Bijective functions (aka Correspondences))**.** *A function $f : A \to B$ is injective (or 1-1) if for all $a \neq b$, $f(a) \neq f(b)$. The function f is surjective (or onto) if for all $b \in B$, there is $a \in A$ such that $f(a) = b$. If f satisfies both conditions, it is called a bijection or correspondence.*

Definition 2 (Same size)**.** *Two sets A and B have the same size if there is a correspondence $f : A \to B$.*

For finite sets this is trivial. What about infinite sets?

Problem 1. *Do the set of natural numbers $\mathbb{N}$ and the set of even numbers $2\mathbb{N}$ have the same size?*

Proof. Yes. Define $f : \mathbb{N} \to 2\mathbb{N}$ by $f(n) = 2n$. □

Definition 3 (Countable set)**.** *A set A is countable if it is finite or has the same size as $\mathbb{N}$.*

Problem 2. *Is the set of positive rational numbers $Q = \{\frac{a}{b} \mid a, b \in \mathbb{N}^*\}$ countable?*

Proof. If true, we have to find a correspondence between $\mathbb{N}$ and Q. Consider the following way of pairing $\mathbb{N}$ and Q. An idea would be to go through Q row by row. But the rows are infinite, so we would never return from the first row. Instead we traverse Q diagonally as in Figure 1, mapping each number in the list with the next natural number and skipping repeated elements (e.g. $2/2$). So there is a correspondence between $\mathbb{N}$ and Q as required. □

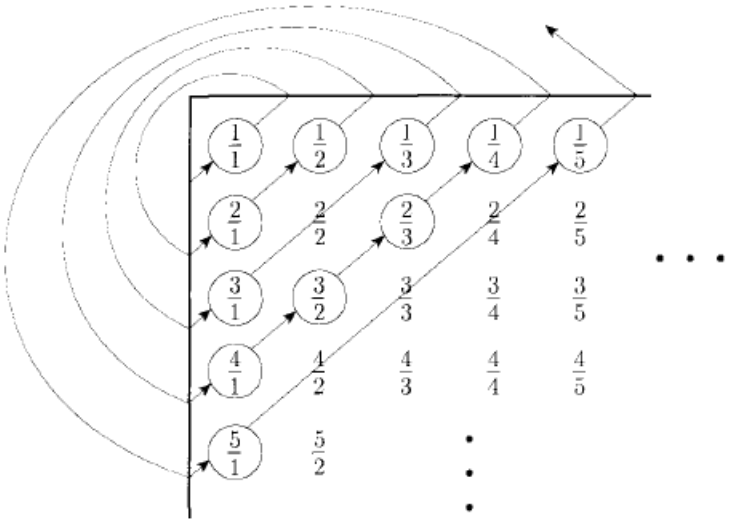

Figure 1: Enumerating the set of positive rational numbers

Problem 3. *Is $\Re$ countable?*

Proof. Suppose $\Re$ is countable. Then there is a correspondence $f : \mathbb{N} \to \Re$. Let $s_i = f(i)$ for all $i \in \mathbb{N}$. We construct now a number that is not on the list $s_1, s_2, \ldots$ as follows. Let $s^* = 0.d_1d_2\ldots$ such that for each index $i \in \mathbb{N}$, if the ith digit (after the decimal point) of s_i is

- Equal to 1, set $d_i = 2$.
- Else, set $d_i = 1$.

Then $s^* \neq f(i)$ for all $i \in \mathbb{N}$, so s^* is not on our list. But f was supposed to be a correspondence. We reached a contradiction, so f cannot exist. □

Proof of Theorem 1.1. Suppose towards a contradiction that A_{TM} is decidable. Let H be a decider for A_{TM}. Then given input $\langle M, w\rangle$, H accepts if M accepts w and rejects if H does not accept w;

so H always halts. Now construct a new machine D that runs H as a subroutine and behaves in an opposite way.

That is, on input $\langle M \rangle$, where M is a TM, D does the following:

- Call H (as a subroutine) on input $(\langle M, \langle M \rangle)$.
- Do the opposite of what H does: if H accepts, *reject*, and if H rejects, *accept*.

What does D do when given its own description, $\langle D \rangle$, as input?

- D calls H on input $(\langle D, \langle D \rangle)$.
- Then it flips the answer of H: if D accepts $\langle D \rangle$, then it rejects, and if D rejects, then it accepts.

So does D accept or reject the input $\langle D \rangle$? This is a contradiction, so the assumption must have been false and H cannot exist. □

Note 1. *We didn't explicitly say where diagonalization was used, but it's built in the proof. Suppose we have some table with the result of the execution of a machine on the description of another machine (see Figure 2).*

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1	*accept*		*accept*		
M_2	*accept*	*accept*	*accept*	*accept*	
M_3					$\cdots$
M_4	*accept*	*accept*			
$\vdots$			$\vdots$		

Figure 2: Each entry (i, j) is *accept* if M_i accepts input $\langle M_j \rangle$. Missing entries if M_i loops on $\langle M_j \rangle$.

If we construct the table for D, then it is different (on the diagonal) for every machine in the list:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1	*accept*	*reject*	*accept*	*reject*	
M_2	*accept*	*accept*	*accept*	*accept*	
M_3	*reject*	*reject*	*reject*	*reject*	$\cdots$
M_4	*accept*	*accept*	*reject*	*reject*	
$\vdots$			$\vdots$		

Figure 3: Table for machine H. The missing entries have been filled in with reject since H is a decider.

Finally, consider the table for D.

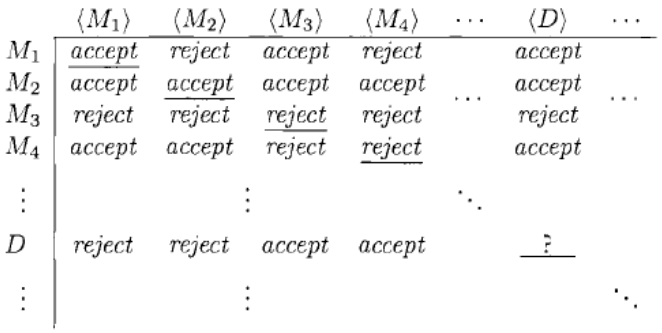

Figure 4: Table for machine D. There is a problem at the entry $(D, \langle D \rangle)$.

Note 2. *Rice's theorem says that in fact every non-trivial property of the language of a TM is undecidable.*