CMSC 141 Automata and Language Theory

Turing Machines

Mark Froilan B. Tandoc

November 19, 2014

Equinumerous Sets

Question

How can we say that a set is larger than another set?

Answer

Count the elements of the sets and the one with more elements is larger. (e.g. $\{a,b\}$, $\{1,0\}$)

Problem

What if the sets are infinite? How can we count them? Can we count infinite sets?

Diagonalization Method

- Proposed by Georg Cantor, a mathematician, in 1873.
- Two sets are equinumerous if there is a one-to-one pairing for all elements.
- We call the pairing as the *correspondence* of the sets.

Diagonalization Method

Example: Let $\mathbb{N}$ be the set of natural numbers $\{1, 2, 3, \dots\}$ and let $\mathbb{E}$ be the set of even natural numbers $\{2, 4, 6, \dots\}$. The correspondence function can be $f(n) = 2n$.

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

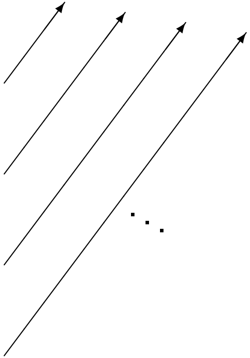
Definition

A set is *countable* if it is either finite or has the same size as $\mathbb{N}$

Diagonalization Method

How about the set of positive rational numbers
$\mathbb{Q} = \{\frac{m}{n} | m, n, \in \mathbb{N}\}$

$$
\begin{array}{|cccccc}
\hline
\frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \\
\frac{2}{1} & \frac{2}{2} & \frac{2}{3} & \frac{2}{4} & \frac{2}{5} & \\
\frac{3}{1} & \frac{3}{2} & \frac{3}{3} & \frac{3}{4} & \frac{3}{5} & \cdots \\
\frac{4}{1} & \frac{4}{2} & \frac{4}{3} & \frac{4}{4} & \frac{4}{5} & \\
\frac{5}{1} & \frac{5}{2} & \frac{5}{3} & \frac{5}{4} & \frac{5}{5} & \\
 & & \vdots & & &
\end{array}
$$



Diagonalization Method

- It appears that every infinite set can be shown to have the same size.
- Are there some set that is *uncountable*?
- An example is the set of real numbers $\mathbb{R}$

Diagonalization Method

Proof is by contradiction, assume we have a correspondence,

n	$f(n)$
1	3.14159...
2	1.23456...
3	0.50000...
4	0.54321...
$\vdots$	$\vdots$

then we find a real number x that is not paired with an element of $\mathbb{N}$

Diagonalization Method

- Let x be a number between 0 and 1.
- The objective is to ensure that $x \neq f(n)$ for any n.
  - To ensure that $x \neq f(1)$, we let the first digit after the decimal point different from the first fractional digit 1 of $f(1) = 3.14159\ldots$
  - To ensure that $x \neq f(2)$, we let the second digit after the decimal point different from the first fractional digit 3 of $f(1) = 1.23456\ldots$
  - and so on

Diagonalization Method

From the table:

n	$f(n)$
1	3.14159...
2	1.23456...
3	0.50000...
4	0.54321...
$\vdots$	$\vdots$

We can find $x = 0.2413...$

We are sure that x can never appear on the table because it will always differ on the n^{th} fractional digit.

Halting Problem

$$A_{TM} = \{< M, w > | \text{M is a TM and M halts on w}\}$$

$$H(< M, w >) = \begin{cases} \textit{accept} \text{ if M halts on w} \\ \textit{reject} \text{ if M does not halt on w} \end{cases}$$

Now we construct a new TM D with H as a subroutine with input $< M, < M >>$ and reverse the output

$$D(< M >) = \begin{cases} \textit{accept} \text{ if H does not accept } <\text{M},<\text{M}>> \\ \textit{reject} \text{ if H accepts } <\text{M},<\text{M}>> \end{cases}$$

Halting Problem

$$D(< M >) = \begin{cases} \textit{accept} \text{ if H does not accept <M,<M>>} \\ \textit{reject} \text{ if H accepts <M,<M>>} \end{cases}$$

In other words

$$D(< M >) = \begin{cases} \textit{accept} \text{ if M does not halt on <M>} \\ \textit{reject} \text{ if M halts on <M>} \end{cases}$$

However, if we run D with $< D >$ as input, we encounter a contradiction

$$D(< D >) = \begin{cases} \textit{accept} \text{ if D does not halt on <D>} \\ \textit{reject} \text{ if D halts on <D>} \end{cases}$$

Consider the table of results of running H

	$< M_1 >$	$< M_2 >$	$< M_3 >$	$< M_4 >$	$\dots$
M_1	*accept*	*reject*	*accept*	*reject*	
M_2	*accept*	*accept*	*accept*	*accept*	
M_3	*reject*	*reject*	*reject*	*reject*	
M_4	*accept*	*accept*	*reject*	*reject*	
$\vdots$		$\vdots$			

Since D is a TM, it should appear in our table.

Diagonalization on Halting Problem

Since D is a TM, it should appear in our table.

	$< M_1 >$	$< M_2 >$	$< M_3 >$	$< M_4 >$	$\dots$	$< D >$	$\dots$
M_1	*accept*	*reject*	*accept*	*reject*		*accept*	
M_2	*accept*	*accept*	*accept*	*accept*	$\dots$	*accept*	$\dots$
M_3	*reject*	*reject*	*reject*	*reject*		*reject*	
M_4	*accept*	*accept*	*reject*	*reject*		*accept*	
$\vdots$		$\vdots$			$\ddots$		
D	*reject*	*reject*	*accept*	*accept*		?	
$\vdots$		$\vdots$					$\ddots$

References

- Previous slides on CMSC 141
- M. Sipser. Introduction to the Theory of Computation. Thomson, 2007.
- J.E. Hopcroft, R. Motwani and J.D. Ullman. Introduction to Automata Theory, Languages and Computation. 2nd ed, Addison-Wesley, 2001.
- E.A. Albacea. Automata, Formal Languages and Computations, UPLB Foundation, Inc. 2005
- JFLAP, www.jflap.org
- Various online LaTeX and Beamer tutorials