Advanced Topics in Computability

1 Diagonalisation

Definition: Countable

A set S is countable if there is a one-to-one correspondence between S and the set of natural numbers $\mathbb{N}$

2 Cantor's Proof

Proposition: The set of reals in the interval (0,1) is uncountable

Proof: A real number A in (0,1) is an (infinite) decimal expansion: $A = 0.a_1a_2a_3...$

Assume, for the sake of contradiction, there is a one-to-one correspondence between the real interval (0,1) and $\mathbb{N}$, i.e. all the reals in (0,1) can be ordered in a sequence

$$A_1, A_2, A_3...$$

We will construct a real number which is not in the sequence

3 Cantor's diagonal argument

Denote $A_i = 0.a^i_1 a^i_2 a^i_3....$ and put the sequence in the following rectangular table

$$\begin{array}{lllllllll}
A_1 = & 0 & . & a^1_1 & a^1_2 & a^1_3 & \dots & \dots & \cdots \\
A_2 = & 0 & \cdot & a^2_1 & a^2_2 & a^2_3 & \dots & \dots & \cdots \\
A_3 = & 0 & \cdot & a^3_1 & a^3_2 & a^3_3 & \dots & \dots & \cdots \\
 & \vdots & & & & \ddots & & & \\
A_i = & 0 & . & a^i_2 & a^i_3 & \dots & a^i_i & \cdots & \cdots \\
\vdots & & & & & & \ddots & & \\
\vdots & & & & & & & \ddots & \\
 & & & & & & & & \ddots
\end{array}$$

Construct a new number $B = 0.b_1b_2b_3...$ by taking

$$b_i = \begin{cases} a^i_i + 1 & \text{if } a^i_i < 9 \\ 0 & \text{if } a^i_i = 9 \end{cases}$$

Now, B is a real number in (0,1) which is not in the table above, as $b_i \neq a^i_i$ for every i

4 Halting problem by diagonalisation

The set of all strings over a finite alphabet is countable - order them by length first and order the ones of the same length in lexicographic order

$$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, \dots$$

Therefore, the set of all Turing machines is countable, too. Put all TMs vs all inputs in an infinite table.

$$\begin{array}{c|cccccccc}
HALT\,(M,w) & w_0 & w_1 & \cdots & w_i & \cdots & w_j & \cdots \\
\hline
M_0 & h_{00} & h_{01} & & & & & \\
M_1 & h_{10} & h_{11} & & & & & \\
\vdots & & & \ddots & & & \vdots & \\
M_i & & & \cdots & h_{ii} & \cdots & h_{ij} = \begin{cases} 1 & M_i \text{ halts on } w_j \\ 0 & \text{otherwise} \end{cases} & \cdots \\
\vdots & & & & & \ddots & \vdots &
\end{array}$$

With the help of HALT machine, we created a TM M tat everywhere disagrees with the diagonal

5 The class of Nice machines

A set of Turing machines $\mathcal{N}$ has a Universal machine $U_{\mathcal{N}}(i,w)$ if

1. For every machines $N \in \mathcal{N}$, there is a number n such that $N(w) = U_{\mathcal{N}}(n,w)$ or all inputs w
2. For every number n, the machine $U_{\mathcal{N}}(n,.) \in \mathcal{N}$

Definition: Nice machines

The class of "nice" machines $\mathcal{N}$ is the set of all TMs that terminate on every input

Proposition: The class of "nice" machines $\mathcal{N}$ does not have a universal machine

Proof: Assume that there is a universal function $U_{\mathcal{N}}(i,w)$. Diagonalise: consider the machine M defined by

$$M(w_i) = \neg U_{\mathcal{N}}(i,w_i)$$

for all i

M itself is a nice machine, so there much be a number n such that $M(w) = U_{\mathcal{N}}(n,w)$ for all inputs w. In particular, for $w = w_n$ we would have that

$$M(w_n) = U_{\mathcal{N}}(n,w_n)$$

However, but by the construction of M we have that

$$M(w_n) = \neg U_{\mathcal{N}}(n,w_n)$$

which is a contradiction

6 Self-Reference

We want a program (Turing machine) that ignores the input and produced its own source code (description) as output.

Definition: Quine

A program that generates a copy of its own source code as its complete output

7 Solution by mutual recursion

A quine that consists of two parts: A followed by B. A prints out B in a straightforward way, and then B prints out A using the output that has just been produced by A.