Lesson 10: Universal Turing machine and Halting problem

Theme: Universal Turing machine and Halting problem

The string representation of a Turing machine. Recall that a Turing machine is defined as a system $\mathcal{M} = \langle \Sigma, \Gamma, Q, q_0, q_{\text{acc}}, q_{\text{rej}}, \delta \rangle$, where we can assume that $\Sigma = \{0, 1\}$ and $\Gamma = \{\prec, 0, 1, \sqcup\}$. Without loss of generality, we can also assume that $Q = \{0, 1, \ldots, n\}$ for some positive integer $n$ with 0 being the initial state.

We note the following.

- Each state $i \in Q$ is written as a string in its binary form.
- Each transition $(i, a) \rightarrow (j, b, \alpha) \in \delta$ can be written as string over the symbols $0, 1, (, )$, $\prec, \sqcup, L, R, S$, where the symbol $\sqcup$ represents $\sqcup$, and $L, R, S$ represent Left, Right, Stay, respectively.

So, the whole system $\mathcal{M} = \langle \Sigma, \Gamma, Q, 0, q_{\text{acc}}, q_{\text{rej}}, \delta \rangle$ can be written as a string:

$$[\Sigma] \# [\Gamma] \# [Q] \# [0] \# [q_{\text{acc}}] \# [q_{\text{rej}}] \# [\delta]$$

where $[\cdot]$ denotes the string representing the component $\cdot$ and $\#$ the symbol separating two consecutive components.

This shows that every Turing machine (whose tape alphabet is $\Gamma = \{\prec, 0, 1, \sqcup\}$) can be described as a string over a fixed set of the symbols, i.e., $0, 1, (, )$, $\prec, \sqcup, L, R, S, \#$. All these symbols can be further encoded into strings over 0 and 1 to obtain a binary string, which we denote by $[\mathcal{M}]$. That is, $[\mathcal{M}]$ is the binary string representing the Turing machine $\mathcal{M}$. Sometimes, we will also say $[\mathcal{M}]$ is the string description of $\mathcal{M}$, or the description of $\mathcal{M}$, for short.

Universal Turing machine (UTM). A universal Turing machine (UTM) is a Turing machine $\mathcal{U}$ that gets as input a description of a Turing machine $[\mathcal{M}]$ and a word $w$. On such input, it simulates $\mathcal{M}$ on $w$. (Some textbooks use the phrase “it runs $\mathcal{M}$ on $w$” for “it simulates $\mathcal{M}$ on $w$.”)

Halting problem. We define the following languages:

$$\text{HALT} := \{[\mathcal{M}]w \mid \mathcal{M} \text{ accepts } w \text{ where } w \in \{0, 1\}^*\}.$$  
$$\text{HALT}_0 := \{[\mathcal{M}] \mid \mathcal{M} \text{ accepts } [\mathcal{M}]\}.$$  
$$\text{HALT}'_0 := \{[\mathcal{M}] \mid \mathcal{M} \text{ does not accept } [\mathcal{M}]\}.$$  

**Theorem 10.1** $\text{HALT}'_0$ is undecidable.

**Corollary 10.2** $\text{HALT}_0$ and $\text{HALT}$ are undecidable.

**Proposition 10.3** The language $\text{HALT}_0$ and $\text{HALT}$ are recognizable (recursively enumerable).

Recall that if both $L$ and its complement $\overline{L} = \Sigma^* - L$ are recognizable, then both are decidable. Then, the following corollary follows immediately from above.

**Corollary 10.4** The language $\overline{\text{HALT}}$ is not recognizable (recursively enumerable).

*Obviously, since we consider only Turing machines with $\Sigma = \{0, 1\}$ and $\Gamma = \{\prec, 0, 1, \sqcup\}$, it is not necessary to include them in $[\mathcal{M}]$. But for the sake of consistency in our notation, we simply include them.*