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 $M = \langle \Sigma, \Gamma, Q, q_0, q_{\text{acc}}, q_{\text{rej}}, \delta \rangle$, where we can assume that $\Sigma = \{0, 1\}$ and $\Gamma = \{<, 0, 1, \sqcup\}$. 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 $0, 1, (, )$, $<, \sqcup, L, R, S$, where the symbol $\sqcup$ represents $\sqcup$, and $L, R, S$ represent Left, Right, Stay, respectively.

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

$$\lfloor \Sigma \rfloor \ # \ \lfloor \Gamma \rfloor \ # \ \lfloor Q \rfloor \ # \ \lfloor 0 \rfloor \ # \ \lfloor q_{\text{acc}} \rfloor \ # \ \lfloor q_{\text{rej}} \rfloor \ # \ \lfloor \delta \rfloor$$

where $\lfloor \cdot \rfloor$ denotes the string representing the component $\cdot$ and $\#$ the symbol separating two consecutive components.\(^1\)

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

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

Halting problem. We define the following two languages:

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

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

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

Proposition 10.3 For every language $L \subseteq \Sigma^*$, if both $L$ and its complement $\overline{L} = \Sigma^* - L$ are recognizable (recursively enumerable), then both are decidable.

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

---

\(^1\)Obviously, since we consider only Turing machines with $\Sigma = \{0, 1\}$ and $\Gamma = \{<, 0, 1, \sqcup\}$, it is not necessary to include them in $\lfloor M \rfloor$. But for the sake of consistency in our notation, we simply include them.