Notes on coding of Turing machines

We want to show that there is a fixed finite alphabet Σ_{TM} so that every TM M can be represented uniquely as a string in Σ_{TM}^*. We will write $\langle M \rangle$ to denote this string.

Recall that a TM M is specified by a 7-tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r \rangle$, where Q is the set of *states*, Σ is the set of *input symbols,* $\Gamma \supseteq \Sigma$ is the set of *tape symbols,* q_0 is the *initial state,* q_a is the *accepting state* and q_r is the rejecting states.

Suppose $Q = \{q_0, \dots, q_k\}$ – we will assume that q_0 is the initial state, $q_a = q_{k-1}$ and $q_r = q_k$. Suppose $\Sigma = \{a_0, \dots, a_\ell\}$. Since $\Sigma \subseteq \Gamma$, we can assume $\Gamma = \{a_0, \dots, a_\ell, a_{\ell+1}, \dots, a_m\}$ so to specify Γ we only need to specify $a_{\ell+1}, \dots, a_m$. Also, we assume that $a_m = \sqcup$.

The transition function δ can be represented as a set of 5-tuples, e.g., if $\delta(q, a) = (p, b, R)$ then we have the 5-tuple $\langle q, a, p, b, R \rangle$.

Referring to the example TM from Lecture 10, we specified it as

$$M = \langle \{q_0, q_1, q_2, q_3, q_a, q_r\}, \{a, b\}, \{a, b, a', b', \sqcup\}, \delta, q_0, q_a, q_r \rangle$$

Using the conventions above, represent M by

$$M = \langle \{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a_0, a_1\}, \{a_3, a_4, a_5\}, \delta, q_0, q_4, q_5 \rangle$$

This is a finite string, but it is not over a (fixed) finite alphabet. The problem is that there is no bound on the number of states or symbols that a TM could use. So we need a different way to represent them – the easiest way is to use the binary notation for the subscript. So for the machine above we would have

$$M = \langle \{0,1,10,11,100,101\}, \{0,1\}, \{11,100,101\}, \delta, 0,100,101 \rangle$$

Using this approach, we see that any TM can be represented using a string over the alphabet whose symbols are $\{, \}, \langle, \rangle, 0, 1, L, R$ and , (comma)

Later on we will see how we can represent strings in Σ^* for any finite Σ as strings in $\{0,1\}^*$, but for now we won't worry about this.

Using ideas similar to the TM simulation of an ideal computer, we can see that there is a *universal Turing machine U* which is given $\langle M \rangle$ and u as input, can simulate the execution of $\langle M \rangle$ on u.