Theory of Computation — CSE 105

Half-Language

Solution for Problem 1.42

Idea: Sine L is regular, Let $M = (Q, \Sigma, \delta, q_0, A)$ be a DFA recognizing L. The idea for recognizing $\frac{1}{2}L$ is the following: We are given a string x and we need to check if there is a string y of equal length as x such that $xy \in L$. We want to make transitions (according to DFA M) on x while at the same time keeping track of all possible 'backward' steps we could take from one of accepting states of M. Thus, if we run the 'forward' and the 'backward' transitions in synchrony, these transitions can meet at the same state if there is an equal length string y such that $xy \in L$. We use the cartesian product construction to carry out the 'parallel' simulation required to implement our strategy.

Construction: We will now design a machine $\frac{1}{2}M = (Q', \Sigma, \delta', s_0, A')$ where $Q' = Q \times Q \cup \{s_0\}$, s_0 is a new start state distinct from the states in Q, and $A' = \{(p, p) | p \in Q\}$. δ' is defined as follows:

We create transitions from the start state s_0 to every pair of states such that the first state is the start state q_0 of M and the second state in the pair is an accepting state of M.

For every $p \in A$, $\delta'(s_0, \epsilon) = (q_0, p)$.

For every pair of states $((p, q), (p', q'))$, and input symbol $\sigma \in \Sigma$, we create a transition from (p, q) to (p', q') on the symbol σ if the following condition holds:

$\delta(p, \sigma) = p'$ and $\exists \sigma' \in \Sigma$ such that $\delta(q', \sigma') = q$.

In other words, we create a transition between a pair of states on an input symbol if there is a forward transition between the corresponding first states of the pair on the input symbol and there is a 'backward' transition between the second states on some input symbol.

This completes the description of the machine $\frac{1}{2}M$ for recognizing $\frac{1}{2}L$.

Justification: We now want to argue that the NFA $\frac{1}{2}M$ does indeed recognize $\frac{1}{2}L$.

Consider any string $x \in \frac{1}{2}L$. Since $x \in \frac{1}{2}L$, there exists a y of equal length as x such that $xy \in L$. Let $xy = u_1 \cdots u_k u_{k+1} \cdots u_{2k}$ where $u_i \in \Sigma$. Let $q_0, q_1, q_k, q_{k+1}, \ldots, q_{2k}$ be the sequence of states traversed by the machine M when the input xy is presented. q_{2k} must be an accepting state of M. Based on our construction of $\frac{1}{2}M$, it is clear that $\frac{1}{2}M$ when presented with x can traverse thru the following sequence of states: $s_0, (q_0, q_{2k}), (q_1, q_{2k-1}), \ldots, (q_{k-1}, q_{k+1}), (q_k, q_k)$. Thus x will be accepted by $\frac{1}{2}M$.

In the other direction, consider any string x accepted by $\frac{1}{2}M$. Let $s_0, (q_0, q_{2k}), (q_1, q_{2k-1}), \ldots, (q_{k-1}, q_{k+1}), (q_k, q_k)$ be an accepting sequence of states traversed by x in $\frac{1}{2}M$. By our construction, q_{2k} must be an accepting state. Also input x will traverse thru the states $q_0, q_1, \ldots q_k$ in M. Let y be the string that causes M to traverse through the states $q_k, q_{k+1}, \ldots q_{2k}$. Such a sequence must exists by our construction and moreover the length of y is the same as that of x. Thus the sequence of states $q_0, q_1, \ldots, q_k, q_{k+1}, \ldots q_{2k}$ is an accepting sequence of states for xy in M. Thus $xy \in L$.