Homework 1

Jan 26, 2017

(804501476)

Problem 1.0:

Proof. Let A and B be regular then this implies that $\exists$ DFAs
$M_1 = (Q_1, \Sigma_1, \delta_1, q_0, F)$
$M_2 = (Q_2, \Sigma_2, \delta_2, q_0^{'}, F^{'})$
which recognize A and B respectively.
We construct PDA $N = (Q, \Sigma, \delta, q_{start}, F, \Gamma)$:
1) $Q = q_{start} \cup Q_1 \cup Q_2$
2) $\Sigma = \Sigma_1 \cup \Sigma_2$
3) $\Gamma = \{\$, u \in \Sigma_1\}$
4) $\delta^{'}(r \in Q, a \in \Sigma, b) = \begin{cases} \{q_0, (\varepsilon, \varepsilon \to \$)\}, \text{ if } r = q_{start} \\ \{\delta_1(r, a), (a, \varepsilon \to a)\}, \text{ if } a \in \Sigma_1 \\ \{q_0^{'}, (a, b \to \varepsilon)\}, \text{if } r \in F, \ a \in \Sigma_2 \text{ and } b \in \Sigma_1 \\ \{\delta_2(r, a), (a, b \to \varepsilon)\}, \text{ if } a \in \Sigma_2 \text{ and } b \in \Sigma_1 \\ \{q_{end}, (\varepsilon, \$ \to \varepsilon)\}, \text{ if } r \in F^{'} \end{cases}$
5) q_{start}
6) $F^{''} = q_{end}$

Claim: N accepts $A \nabla B \iff M_1$ accepts A and M_2 accepts B
($\Leftarrow$) Let $x \in A$ and $y \in B$ s.t $|x| = |y| = n$. Now M_1 recognizes x as follows: $\exists$ states $a_1, a_2, \ldots, a_n$ s.t $\delta_1(a_i, x_i) = \{a_j \in Q_1\}$, $a_1 = q_0$, and $a_n \in F$. M_2 recognizes y as follows: $\exists$ states $b_1, b_2, \ldots, b_n$ s.t $\delta_2(b_i, y_i) = \{b_j \in Q_2\}$, $b_1 = q_0^{'}$, and $b_n \in F$. Now the states $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ is a concatenation of paths from M_1 to M_2 machine N has start state q_{start} which makes an epsilon transition to $q_0 = a_1$ and will only get to q_{end} if it has gone through $q_0^{'} \in F^{'}$. This machine begins with an empty stack and pushes every character from x into the stack and will only accept if y has an equal amount of chararacters since $|x| = |y| = n$ we accept.
($\Rightarrow$) Let $z = z_1 z_2 \ldots z_{2n} \in A \nabla B$. Now machine N recognizes z as follows $\exists$ states $r_0 r_1 r_2 \ldots r_{2n+1}$ and strings $s_0 s_1 \ldots s_{n+1} \in \Gamma^*$ s.t $\delta(r_i, s_i, a) = (r_{i+1}, b)$ s.t $s_i = at$ and $s_{i+1} = bt$ for $a, b \in \Gamma$ and $t \in \Gamma^*$. We also have that $r_0 = q_{start}$, $s_0 = \$$, $r_{2n+1} = q_{end}$. Now the states $r_0 r_1 r_2 \ldots r_{2n+1}$ correspond to the states $q_{new}, a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n, q_{end}$. The states $a_1, a_2, \ldots, a_n$ are a computational path on machine M_1 and the states $b_1, b_2, \ldots, b_n$ are a computational path on machine M_2.
We have proved equivalency between PDA's and CFG $\implies A \nabla B$ is a CFL. $\square$

Problem 2.0:

a.
L regular $\implies \exists$ DFA $M = (Q, \Sigma, \delta, q_o, F)$.
We construct $G_L = (V, \Sigma, R, S)$:
1) $V = Q$
2) $\Sigma = \Sigma$
3) $S = q_0$
4) $R = \begin{cases} q^{'} \to aq^{''}, & \text{if } \delta(q^{'}, a) = q^{''} \\ q^{'} \to \varepsilon, & \text{if } \delta(q^{'}, a) = q^{''} \text{ } and \text{ } q^{''} \in F \end{cases}$

Claim: G_L generates L

Proof.
($\Rightarrow$) By construction of G_L every $w \in G_L$ creates a computation path on $M \implies w \in L$
($\Leftarrow$) $\forall$ $w = w_1 w_2 \ldots W_n \in L$ then $\exists$ a computation path on M, $q_0 q_1 q_2 \ldots q_{n+1}$. By construction of G_L, in deriving w, the start symbol of w's derivation is q_0 and we continue to derive using δ to pick the next variable as such

$$q_0 \to w_1 q_1 \to w_1 w_2 q_2 \to \cdots \to w_n q_{n+1} \tag{2.1}$$

The set of variables produces is the same computational path on M, so w is generated by G_L. □

b. $G = (V, \Sigma, R, S)$ be a regular grammar this implies for every variable $A \in V$ we have the following set of rules.
1) $A \to aB$
2) $A \to a$
3) $A \to \varepsilon$
where a is a terminal symbol and B, $a \in \Sigma$ ad $B \in V$.
We construct NFA $N = (Q, \Sigma, \delta, q_0, F)$ as follows:
1) $Q = \{q_i \mid q_i \in V\}$
2) $\Sigma = \Sigma$
3) $q_0 = S$
4) $F = q_{end}$
5) $\delta(q_i, a \in \Sigma) = \begin{cases} \delta(q_i, a) = q_j, & \text{if } q_i \to aq_j \\ q_{end}, & \text{if } q_i \to \varepsilon \mid a \end{cases}$

Proof. ($\Rightarrow$) By construction of N every $w \in L$ which N accepts is a valid derivation from a regular grammar G.
($\Leftarrow$) $\forall w = w_1 w_2 \ldots w_n \in G$ $\exists$ a derivation

$$S \to w_1 V_1 \to w_2 V_2 \to \cdots \to w_n V_n \tag{2.2}$$

Problem 2.0 continued on next page. . .

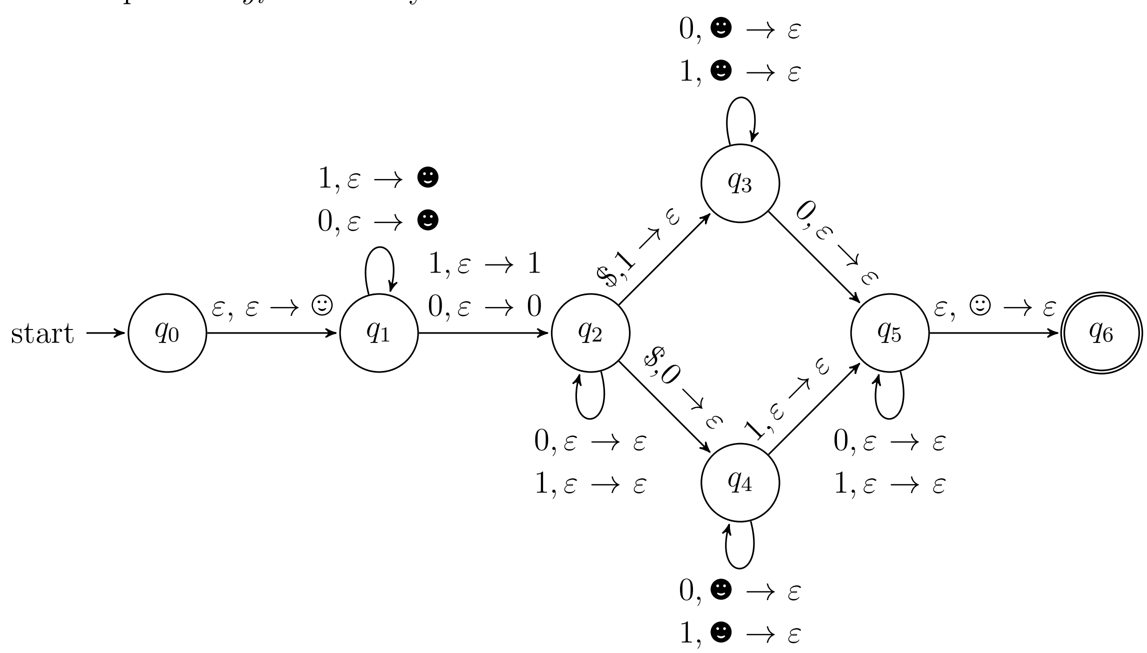
By construction of N $S = q_o$ and the variables V_i correspond to states in N s.t. $\delta(V_j, a) = V_i$ so w induces a computational path on M and a terminal rule such as $v_i \rightarrow a \mid \varepsilon$ is a transitiion onto an accepting state of N. Therefore w is accepted $\implies w \in L$. $\square$

Problem 3.0:

a) For x and y to be different they have to be different in at least one place.

$$\{0 \cup 1\}^a x_i \{0 \cup 1\}^b \$ \{0 \cup 1\}^a y_i \{0 \cup 1\}^c \tag{3.3}$$

We construct a non-deterministic PDA M that reads x and pushes ☻ onto the stack as a counter for the index before position i. We non-determinitically guess where i is and what symbol x_i is and push it onto the stack. We continue to read x without performing any action until we reach \$. We transition according the the top of the stack, x_i, and pop. As we read y we pop the ☻, this will index to position y_i and if they are different then there is transition if not the machine dies.



b) For x and y to be different they have to be different in at least one place

$$\{0 \cup 1\}^a x_i \{0 \cup 1\}^b \{0 \cup 1\}^a y_i \{0 \cup 1\}^b \tag{3.4}$$
$$\{0 \cup 1\}^a x_i \{0 \cup 1\}^a \{0 \cup 1\}^b y_i \{0 \cup 1\}^b \tag{3.5}$$

Let $X = \{\{0 \cup 1\}^a x_i \{0 \cup 1\}^a\}$ and $Y = \{\{0 \cup 1\}^b y_i \{0 \cup 1\}^b\}$. We can construct a CFG $G = (R, \{X, Y\}, \{0, 1\}, S)$ for L_2 . With the following rules R:

1) $S \rightarrow XY | YX$
2) $X \rightarrow 0X0 | 1X1 | 0X1 | 1X0 | 1$
3) $Y \rightarrow 0X0 | 1X1 | 0X1 | 1X0 | 0$