CS172 Midterm 1 Solutions

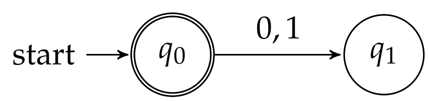
Fall 2013

Recall we are always working on the alphabet $\Sigma = \{0, 1\}$.

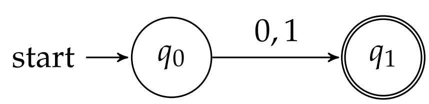
1. (a) True. L can be written as $L = L_1^R \cap L_2^c$. Since the set of regular languages is under reversal, complement and intersection, L_1^R and L_2^c are both regular, and hence so is L.

   (b) False, even for $k = 1$. There are infinite number of regular languages containing only 1 string, but there are only finite number of DFAs with at most 2 states, so some of these languages cannot be recognized by this kind of DFAs.

   (c) False. A counterexample:
   Let N be



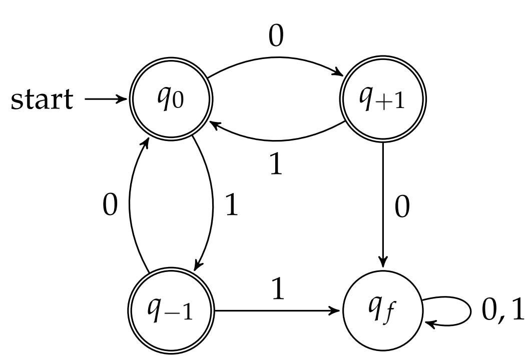
   Then N' will be:



   We have $L(N) = \{\epsilon\}$, $L(N') = \{0, 1\} \neq L(N)^c$.

   (d) False. A counterexample:
   Let $L_j = \{0^j 1^j\}$ for any $j \geq 1$. Each L_j is regular, since it is finite. But their union $\cup_{j=1}^{\infty} L_j = \{0^j 1^j | j \geq 1\}$ is not regular.

2. (a) The DFA is as follows:



When the number of 0's seen so far $-$ the number of 1's seen so far $= 0, +1$ or -1, this DFA is in state q_0, q_{+1} or q_{-1}, all of which are accepting. But as soon as it sees two more 0's than 1's or two more 1's than 0's, it will move to state q_f and never leave it, and this DFA will reject the input string.

(b) Such DFA does not exist. Here we give two proofs.

Proof 1: Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA with 3 states that accepts L, where $Q = \{q_0, q_1, q_2\}$. Since $\epsilon \in L$, we must have $q_0 \in F$. Then, consider $\delta(q_0, 0)$. It cannot be q_0, because otherwise M would accept $00 \notin L$. Without loss of generality, we assume $\delta(q_0, 0) = q_1$. Since $0 \in L$, we get $q_1 \in F$. Similarly, $\delta(q_0, 1) \neq q_0$, because otherwise the DFA would accept $11 \notin L$. Now,

i. If $\delta(q_0, 1) = q_1$, then M will end up in the same state after reading 01 and 11, but $01 \in L$ and $11 \notin L$, which is a contradiction;

ii. otherwise, $\delta(q_0, 1) = q_2$. Since $1 \in L$, we have $q_2 \in F$. Then we get $F = Q$, and hence M accepts any string, which is also a contradiction. $\square$

Proof 2: We will use the Myhill-Nerode theorem. Consider the four strings $x_1 = 0$, $x_2 = 1$, $x_3 = 01$, $x_4 = 11$. We claim that they are pairwise distinguishable by L:

i. Let $z_1 = \epsilon$, then $x_i z_1 \in L$, for $i = 1, 2, 3$, but $x_4 z_1 \notin L$. So $x_i \not\sim_L x_4$, for $i = 1, 2, 3$;

ii. Let $z_2 = 0$, then $x_1 z_2 = 00 \notin L$, $x_2 z_2 = 10 \in L$, $x_3 z_2 = 010 \in L$. So $x_1 \not\sim_L x_j$, for $j = 2, 3$;

iii. Let $z_3 = 1$, then $x_2 z_3 = 11 \notin L$, $x_3 z_3 = 011 \in L$. So $x_2 \not\sim_L x_3$.

So the index of L is at least 4. By the Myhill-Nerode theorem, any DFA recognizing L must have at least four states. $\square$

3. Here we give two proofs.

Proof 1(Easy): Since L is regular, let p be the constant promised by the pumping lemma. Then, since L is infinite, it must contain a string w such that $|w| > p$. By the pumping lemma, there exist strings x, y and z such that $s = xyz$, $|y| > 0$, $|xy| \leq p$, and $xy^iz \in L, \forall i \geq 0$. Now let us define $L_1 = \{xy^iz | i \text{ is even}\} \subseteq L$. Then, L_1 is infinite, since $|y| > 0$. Also, L_1 is regular, since $x(yy)^*z$ is a regular expression for L_1. Now, since both L and L_1 are regular, we get $L \setminus L_1 = L \cap L_1^c$ is also regular (by the closure of regular languages under complement and intersection). Moreover, $L \setminus L_1 \supseteq \{xy^iz | i \text{ is odd}\}$, and hence $L \setminus L_1$ is also infinite. $\square$

Proof 2 (With explicit construction of DFA): Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts L. Since L is regular, let p be the constant promised by the pumping lemma. Then, since L is infinite, it must contain a string w such that $|w| > p$. Then, by the proof of the pumping lemma, there exists a state $q \in Q$ such that M has visited q at least twice when processing w. In fact, the proof also implies that there exist an infinite sequence of strings $v_1, v_2, \dots$ such that $v_i \in L$ and M visits q exactly i times when processing v_i. Now we define

$$L_1 = \{w | w \in L, \; M \text{ visits } q \text{ an odd number of times when processing } w\}. \quad (1)$$

Then L_1 is infinite (since it contains $v_1, v_3, v_5, \dots$). Moreover,

$$L \setminus L_1 = \{w | w \in L, \; M \text{ visits } q \text{ an even number of times when processing } w\} \quad (2)$$

is also infinite (since it contains $v_2, v_4, v_6, \dots$). It remains to show that both L_1 and $L \setminus L_1$ are regular. We prove this by explicitly constructing the DFAs for them. Consider $M' = (Q', \Sigma, \delta', q_0', F')$, where $Q' = Q \times \{0,1\}$, $q_0' = (q_0, 0)$, $F' = F \times \{1\}$, and

$$\delta'((p,i),a) = \begin{cases} (\delta(p,a), i), & \text{if } \delta(p,a) \neq q; \\ (\delta(p,a), i \oplus 1), & \text{otherwise.} \end{cases} \quad (3)$$

Namely, the first coordinate simulates the computation of M, and the second coordinate records the parity of the number of times M has visited q. Every time M visits q, the second bit gets flipped. It is easy to see that M' accepts exactly L_1. Furthermore, if we change F' into $F \times \{0\}$, then M' will accept $L \setminus L_1$ instead. □

4. Proof by contradiction. Suppose L is regular. Then let p be the constant promised by the pumping lemma. Consider $s = 0^{2p}1^{2p}0^{2p}$. Since $s = (0^p1^p0^p) \circ (0^p1^p0^p)$, $s \in L$. By the pumping lemma, there exist strings x, y, z such that $s = xyz$, $|y| > 0$, $|xy| \leq p$, and
$$xy^iz \in L, \; \forall i \geq 0. \quad (4)$$
Now since $|xy| \leq p$ and $|y| > 0$, we have $y = 0^l$ for some $0 < l \leq p$. Now consider $xy^3z = 0^{2p+2l}1^{2p}0^{2p}$. We claim that there is no string b such that $xy^3z = b \circ b$. Suppose otherwise, then the bits in the odd positions of xy^3z indicate $b = 0^{p+l}1^p0^p$, but the bits in the even positions of xy^3z indicate $b = 0^p1^p0^{p+l}$, which is a contradiction. So $xy^3z \notin L$, which contradicts (4). Thus, L is not regular.

5. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing L. We will construct an NFA $N = (Q', \Sigma, \delta', q_0', F')$ recognizing $L^{1/2}$. The basic idea is that we replace the states in Q by the tuples in $Q \times Q$, where the first coordinate goes forward and the second coordinate goes backward. Formally, define $f: Q \to Q$ and $g: Q \to Q$ as follows

$$\begin{aligned} f(q) &= \{p \in Q | \exists a \in \Sigma, \; s.t. \; p = \delta(q,a)\}, \\ g(q) &= \{p \in Q | \exists a \in \Sigma, \; s.t. \; q = \delta(p,a)\}. \end{aligned} \quad (5)$$

(Namely, $f(q)$ consists of the states that can be reached from q in one step, while $g(q)$ consists of the states that can reach q in one step.) Then, let q_0' be a special state, and let
$$Q' = \{q_0'\} \cup (Q \times Q);$$

$$F' = \{(p,q) \in Q \times Q | p = q \text{ or } q \in f(p)\};$$
$$\delta'((p,q),a) = \{(\delta(p,a),r) | r \in g(q)\}, \ \forall p,q \in Q;$$
$$\delta'(q_0', \epsilon) = \{(q_0, r) | r \in F\};$$
$$\delta'(s,b) = \emptyset, \ \text{for other } (s,b) \in Q \times \Sigma_\epsilon.$$

Claim 1: If $x \in L$, then $x^{1/2} \in L(N)$.
Proof: Suppose $x = x_1 x_2 \ldots x_n$ where $x_j \in \Sigma$. Let $r_0 = q_0, r_1, \ldots, r_n$ be the sequence of state M has gone through when processing x. Since $x \in L$, we have $r_n \in F$. Now, consider the following computation of N on $x^{1/2}$: starting from q_0', it jumps to (q_0, r_n) via ϵ-transition. Then, on reading x_1, it moves to (r_1, r_{n-1}). This is valid, since $r_1 = \delta(q_0, x_1)$ and $r_{n-1} \in g(r_n)$. Then, on reading x_2, it moves to (r_2, r_{n-2}), and this is valid too. Continue this procedure, until we have consumed all of $x^{1/2}$. Now N is in the state $(r_{\lfloor n/2 \rfloor}, r_{\lfloor n/2 \rfloor})$ if n is even, or $(r_{\lfloor n/2 \rfloor}, r_{\lfloor n/2 \rfloor + 1})$ if n is odd. Either way, it is in F' by our definition. Hence, $x^{1/2}$ is accepted by N. □

Claim 2: If $y \in L(N)$, then $y \in L^{1/2}$.
Proof: Suppose $y = y_1 y_2 \ldots y_m$ where $y_j \in \Sigma$. By our definition, $y \in L(N)$ implies that there exist (r_0, s_0), (r_1, s_1), ..., $(r_m, s_m) \in Q \times Q$, such that

- $r_0 = q_0$, $s_0 \in F$;
- $r_j = \delta(r_{j-1}, y_j)$, $\forall 1 \leq j \leq m$;
- $s_j \in g(s_{j-1})$, i.e. $\exists z_j \in \Sigma$, s.t. $s_{j-1} = \delta(s_j, z_j)$, $\forall 1 \leq j \leq m$.
- Either $r_m = s_m$, or $s_m \in f(r_m)$, i.e. $\exists z \in \Sigma$, s.t. $s_m = \delta(r_m, z)$.

Now consider the behavior of M on the string $y' = y_1 y_2 \ldots y_m z_m z_{m-1} \ldots z_1$ if $r_m = s_m$, or $y' = y_1 y_2 \ldots y_m z z_m z_{m-1} \ldots z_1$ otherwise (where the z_j's and z are defined as above). We can see that M would gone through the sequence of states: q_0, r_1, r_2, ..., $r_m = s_m$, s_{m-1}, s_{m-2}, ..., $s_0 \in F$ or q_0, r_1, r_2, ..., r_m, s_m, s_{m-1}, s_{m-2}, ..., $s_0 \in F$. Either way, it ends up in an accept state. So $y' \in L$, and $y = (y')^{1/2} \in L^{1/2}$. □

Remark: You do not need to give this formal proof in the exam. A high-level explanation should suffice.