Theory of Computation

Spring 2024, Homework # 1 Reference Solutions

---

1. Let $\equiv_L$ be the equivalence relation induced by L. According to Myhill-Nerode theorem, L is regular iff $\equiv_L$ is of finite index (see p. 95 of Chapter 1 lecture notes). Consider the following sequence of strings: $ab, ab^2, ab^3, \ldots$. Pick any two strings ab^i and ab^j from the sequence where $i, j \in \mathbb{N}$ and $i < j$. Since $ab^i c^i \in L$ and $ab^j c^i \notin L$, ab^i and ab^j must be in different equivalence classes of $\equiv_L$. In other words, $\equiv_L$ must have a unique equivalence class for every string in the above-mentioned sequence. As there are infinitely many strings in the sequence, $\equiv_L$ has infinitely many equivalence classes. Therefore L is not regular.

2. (a) **Correct**. Let $\Gamma = \{c\}$ and h a morphism from Σ to Γ^* where $h(a) = h(b) = c$ and $h(\sigma) = \epsilon$ for all $\sigma \in \Sigma \backslash \{a, b\}$. Let $L_h = h(A)$. Then $L_h \subseteq \Gamma^*$ and for every $u \in \Gamma^*$, $u \in L_h$ if and only if $u = h(w)$ for some $w \in A$. According to the definition of h, $h(w) = c^{\#_a(w)+\#_b(w)}$ for every $w \in \Sigma^*$. Therefore $L_h = \{c^n \mid \exists w \in A, \#_a(w) + \#_b(w) = n\}$. Since regular languages are closed under morphism, L_h is regular if A is regular.

   (b) **Incorrect**. Let $L_1 = a^*$ and $L_2 = b^*$. Then $L = \{a^n 0 b^n | n \in \mathbb{N}\}$. Let h be a morphism from $\{a, b, 0\}^*$ to $\{a, b\}^*$ where $h(a) = a$, $h(b) = b$, and $h(0) = \epsilon$. Then $L' = h(L) = \{a^n b^n | n \in \mathbb{N}\}$. Since L' is not regular, L is not regular (since regular languages are closed under morphism).

   (c) **Correct**. Let $L'_1 = \{x \mid ax \in L\}$ and $L'_2 = \{x \mid xb \in L\}$. Clearly $L' = L'_1 \cup L'_2$. If we can prove that L'_1 and L'_2 are both regular, we can then prove that L' is regular since regular languages are closed under union.
   Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA that recognized L (D must exists since L is regular). So $L(D) = L$. Construct an NFA $N'_1 = (Q \cup q'_0, \Sigma, \delta'_1, q'_0, F)$ where

$$\delta'_1(q, w) = \begin{cases} \delta(q, w) & \text{if } q \in Q \\ \delta(q_0, a) & \text{if } q = q'_0 \text{ and } w = \epsilon \\ \emptyset & \text{if } q = q'_0 \text{ and } w \neq \epsilon \end{cases}$$

   Note that $Q \cap \{q'_0\} = \emptyset$. Then $L(N'_1) = \{x \mid ax \in L(D)\} = L'_1$. Alternatively, since D is a DFA, $\delta(q_0, a)$ is unique. One can use it to construct a DFA $D'_1 = (Q, \Sigma, \delta_1, \delta(q_0, a), F)$, where $L(D'_1)$ is also L'_1. This proves that L'_1 is regular.
   As for L'_2, construct a DFA $D'_2 = (Q, \Sigma, \delta, q_0, F')$ where $F' = \{q \mid \delta(q, b) \in F\}$. Then we have $L(D'_2) = \{x \mid xb \in L(D)\} = L'_2$, which proves that L'_2 is regular.

   (d) **Correct**. Let $L' = L \cap a^* = \{a^p \mid p \text{ is prime }\}$. Since regular languages are closed under intersection, L is regular iff L' is regular. Assume that L' is regular. Let p be the pumping length. Choose $s = a^{p'}$, where p' is a prime number and $p' > p + 1$. By the pumping lemma, there is a partition $s = xyz$ such that $|y| > 0$, $|xy| \leq p$, and $xy^i z \in L'$ for $i \geq 0$. Consider $s' = xy^{p'-|y|}z$. Note that $p' - |y| > 0$ since $|y| \leq |xy| \leq p < p' - 1$. $|s'| = |x| + |z| + (p' - |y|)|y| = (p' - |y|) + (p' - |y|) * |y| = (p' - |y|)(1 + |y|)$ is not a prime number. So $s' \notin L'$. This is a contradition. Therefore L' is not regular. Hence L is not regular.

3. (a) Add a new initial state and two ϵ-transitions from the new initial state to the initial states of A and B. Let $A = (Q_A, \Sigma, \delta_A, q_{A0}, F_A)$ and $B = (Q_B, \Sigma, \delta_B, q_{B0}, F_B)$. Without loss of generality, assume $Q_A \cap Q_B = \emptyset$. Construct $C = (Q_C, \Sigma, \delta_C, q_{C0}, F_C)$ as following:

    - $Q_C = Q_A \cup Q_B \cup \{q_{C0}\}$. Note that $q_{C0} \notin Q_A \cup Q_B$.
    - $\delta_C(q_i, s_i) = \begin{cases} \delta_A(q_i, s_i) & \text{if } q_i \in Q_A \\ \delta_B(q_i, s_i) & \text{if } q_i \in Q_B \\ \{q_{A0}, q_{B0}\} & \text{if } q_i = q_{C0} \text{ and } s_i = \epsilon \\ \emptyset & \text{if } q_i = q_{C0} \text{ and } s_i \neq \epsilon \end{cases}$
    - $F_C = F_A \cup F_B$

    To prove that $L(C) = L(A) \cup L(B)$, we need to show that (i) $L(C) \subseteq (L(A) \cup L(B))$ and (ii) $L(A) \cup L(B) \subseteq L(C)$. For any $s \in L(C)$, let $r = r_0, r_1, r2, \ldots$ be the accepting run of C on s. Based on the way C is constructed, we have $r_0 = q_{C0}$ and $r_1 \in \{q_{A0}, q_{B0}\}$. If $r_1 = q_{A0}$, then $r_i \in Q_A \forall i \geq 2$. Since $inf(r) \cap F_C \neq \emptyset$ and $inf(r) \cap Q_B = \emptyset$, $inf(r) \cap F_A \neq \emptyset$. Hence r is an accepting run of A on s. That is, $s \in L(A)$. Similarly, if $r_1 = q_{B0}$, then r is an acepting run of B on s and $s \in L(B)$. Therefore $L(C) \subseteq L(A) \cup L(B)$. The proof for $L(A) \cup L(B) \subseteq L(C)$ is let as an exercise.

    Note: If you choose not to use ϵ-transitions and instead directly combine q_{A0} and q_{B0} into the new state q_{C0}, you need to also handle any transitions in A and B that might be linking back to their respective initial states. You need to avoid the case where C starts off with A then at some point returns to q_{C0}, which is also q_{B0}, and starts running in B. In this case an acccepting run could potentially contains states from both A and B, which means C might accept strings not in $L(A) \cup L(B)$.

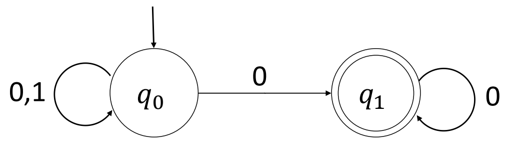
   (b) As suggested in the hint, C needs to accept the runs where F_A and F_B were visited infinitely many times but not necessarily simultaneously. So the idea is to construct C in such a way that it records the visits to F_A and F_B in "pairs" (you can think of it as a partial run $r_p = r_A, r_1, \ldots, r_B$ where $r_A \in F_A$ and $r_B \in F_B$) and only accept the runs where the pairs were visited infinitely many times. Note the usual steps for dealing with intersection problems by constructing C so that A and B run synchronously still need to be followed. Construct $C = (Q_C, \Sigma, \delta_C, q_{C0}, F_C)$ as following:

    - $Q_C = Q_A \times Q_B \times \{1, 2\}$
    - $q_{C0} = (q_{A0}, q_{B0}, 1)$
    - $\delta_C((q_a, q_b, j), s_i) = \begin{cases} (\delta_A(q_a, s_i), \delta_B(q_b, s_i), 1) & \text{if } j = 1 \text{ and } q_a \notin F_A \\ (\delta_A(q_a, s_i), \delta_B(q_b, s_i), 2) & \text{if } j = 1 \text{ and } q_a \in F_A \\ (\delta_A(q_a, s_i), \delta_B(q_b, s_i), 2) & \text{if } j = 2 \text{ and } q_b \notin F_B \\ (\delta_A(q_a, s_i), \delta_B(q_b, s_i), 1) & \text{if } j = 2 \text{ and } q_b \in F_B \end{cases}$
    - $F_C = Q_A \times F_B \times \{2\}$

    To prove that $L(C) = L(A) \cup L(B)$, we need to show that (i) $L(C) \subseteq (L(A) \cup L(B))$ and (ii) $L(A) \cup L(B) \subseteq L(C)$. For any $s \in L(C)$, let $r = r_0, r_1, \ldots =$

$(r_0^A, r_0^B, j_0), (r_1^A, r_1^B, j_1), \ldots$ be the accepting run of C on s, then $inf(r) \cap F_C \neq \emptyset$. Let $r^A = r_0^A, r_1^A, \ldots$ and $r^B = r_0^B, r_1^B, \ldots$. Since $F_C = Q_A \times \mathbf{F_B} \times \{2\}$, $r^B \cap F_B \neq \emptyset$. So r^B is an accepting run of B on s and $s \in L(B)$. Observe that each time an accepting state ($(q_a, q_b, \mathbf{2})$ where $q_b \in F_B$) is visited, the next state must be $(q'_a, q'_b, \mathbf{1})$ (per the fourth line of the transition function). Then before reaching the next accepting state, the second line of the transition function must be executed, which means a state $(q''_a, q''_b, 1)$ where $q''_a \in F_A$ must be visited. Therefore $inf(r^A) \cap F_A \neq \emptyset$ and $s \in L(A)$. Since $s \in L(A) \cap L(B)$, $L(C) \subseteq L(A) \cap L(B)$. The proof of $L(A) \cap L(B) \subseteq L(C)$ is left as an exercies.

(c) (i) The following nondeterminstic IIFA accepts L.



(ii) Let D be a deterministic IIFA that accepts $L(M)$. Since $t_1 = 10^\omega \in L(M)$, D must have an accepting run on t_1. Note the run must be unique since D is deterministic. We can find a string $p_1 = 10^{n_1}$ where $n_1 \geq 1$ so that D entered an accepting state after reading p_1. Since $t_2 = 10^{n_1}100^\omega \in L(M)$, D must also have an unique run on t_2, which extends the aforementioned run on p_1, that is accepting. Therefore we can find a string $p_2 = p_1 10^{n_2} = 10^{n_1}10^{n_2}$ where $n_2 \geq 1$ so that D enters an accepting state after reading p_2. Note that the accepting state D enters after reading p_1 need **not** be the same as the one D enters after reading p_2. Repeating the same argument infinitely many time, we can construct a string $t = 10^{n_1}10^{n_2}10^{n_3}\ldots$ with infinite length where p_i is a prefix of t $\forall i = 1, 2, \ldots$. Consider the unique run r of D on t. Since D visits an accepting state after reading p_1, p_2, p_3, $\ldots$, r contains infinitely many occurrences of accepting states. Since the set of accepting states F contains only finite number of states, some of which must be visited infinitely many times, i.e., $inf(r) \cap F \neq \emptyset$. This means r is an accepting run and therefore $t \in L(D)$. However, since t has infinitely many 1's, $t \notin L(M)$. A contradiction.