CSE 547T Automata Theory January 24, 2017

Solutions for Homework 1

1.

Claim 1. *The language L cannot be written as $L_1 \cdot L_2$ unless one of L_1 or L_2 is $\{\varepsilon\}$.*

Proof. Let L_1 and L_2 be languages whose concatenation $L_{12} = L_1 \cdot L_2$ is exactly the language L of strings of the form yy, $y \in \{0,1\}^*$. Both of L_1 and L_2 must contain ε; otherwise, L_{12} could not contain ε, which is certainly in L ($\varepsilon = \varepsilon \cdot \varepsilon$). It follows that neither L_1 nor L_2 contains "0"; otherwise, we would have $0 \cdot \varepsilon = 0$ or $\varepsilon \cdot 0 = 0$ in L_{12}, even though "0" is not in L. Similarly, neither L_1 nor L_2 can contain "1".

We now observe the following key fact: exactly one of L_1 or L_2 must contain both "00" and "11". Neither of these strings can be constructed from smaller substrings, since L_1 and L_2 lack both "0" and "1". Moreover, if one language had "00" and the other had "11", L_{12} would contain either "0011" or "1100", neither of which is in L.

Suppose "00" and "11" are in L_1. Let $z \in L_2$ be any nonempty string; then both $00z$ and $11z$ are in L_{12}. Because we are claiming $L = L_{12}$, there must be strings y and w for which $00z = yy$ and $11z = ww$. Now $|y| = |w|$, and both are suffixes of z. Hence, $y = w$. But this is impossible because y must begin with "00", while w must begin with "11". Conclude that the string z cannot exist, and so L_2 cannot contain a nonempty string! The opposite case, when L_2 contains "00" and "11", similarly results in the conclusion that L_1 can have no nonempty string.

Conclude that either L_1 or L_2 is $\{\varepsilon\}$. □

2. We first argue that any language accepted by a DFA is also accepted by an all-NFA. As for an ordinary NFA, a DFA is just a special case of an all-NFA. In particular, let DFA $M = (Q, \Sigma, q_0, A, \delta_M)$. Then the corresponding all-NFA N is defined to be $(Q, \Sigma, q_0, A, \delta_N)$, where $\delta_N(q,a) = \{\delta_M(q,a)\}$ as for an ordinary NFA. By construction, N deterministically chooses its next states in the same way M does, so like M, it is always in at most a single state at once. Hence, N accepts precisely when M does, and so $L(N) = L(M)$.

We now make the converse argument, that any language accepted by an all-NFA is also accepted by a DFA. Let $N = (Q, \Sigma, q_0, A, \delta_N)$ be an all-NFA. We use a modified form of the subset construction to convert N to a DFA $M = (Q', \Sigma, q'_0, A', \delta_M)$.

There is one subtlety that we need to take care of before we convert N to a DFA. An NFA (and hence an all-NFA) can *crash* reading string x if the path chosen for x reaches a state from which no transition is labeled with the next character of x. A path that leads to a crash certainly does not accept x, so the all-NFA would not accept x. However, the subset DFA behaves identically on strings that reach a given state set ψ after reading x *without* crashing, and on strings for which some trajectories crash and others reach set ψ. Hence, the regular subset construction cannot distinguish these two cases.

To fix this, we first augment N as follows. Add a new, non-accepting "crash state" q_c to N, which has self-loops on all $a \in \Sigma$. For each $q \in Q$ and $a \in \Sigma$ for which $\delta(q,a) = \emptyset$, redefine $\delta(q,a) = \{q_c\}$. The modified N is equivalent to the original except that it never crashes.

After fixing N as above, the only modification relative to the construction described in class is that

$$A' = \{\langle\psi\rangle \in Q' \mid \psi \subseteq A\},$$

that is, a state of the DFA is accepting iff the corresponding set of states in the NFA is a subset of A (rather than just having a nonempty intersection with A, as in an ordinary NFA). Note that N itself cannot wind up in the empty set of states because it cannot crash, so state $\langle\emptyset\rangle$ of the DFA is not reachable.

Since we've defined δ_M^* the same as in the subset construction for ordinary NFAs, we have as for that construction that

$$\delta_M^*(\langle\psi\rangle, x) = \langle\delta_N^*(\psi, x)\rangle.$$

Now M accepts x iff $\delta_M^*(\langle\{q_0\}\rangle, x) \in A'$. By the above equivalence and the definition of A', this occurs iff $\delta_N^*(\{q_0\}, x) \subseteq A$. Conclude that M accepts x precisely when N does, and so $L(M) = L(N)$.

3. Let $M = (Q, \Sigma, q_0, A, \delta)$ be a DFA accepting language L. Define the ε-NFA $M^R = (Q^R, \Sigma, q_0^R, A^R, \delta^R)$ as follows:

   - $Q^R = Q \cup \{q_s\}$;
   - $q_0^R = q_s$;
   - $A^R = \{q_0\}$;
   - δ^R is defined as follows:
     - for every $q \in Q$ and $a \in \Sigma$,
       $$\delta^R(q, a) = \{p \mid \delta(p, a) = q\}.$$
     - $\delta^R(q_s, \varepsilon) = A$.

   Intuitively, M^R reverses every transition edge in M; accepting paths in M^R start at some state of A and end at q_0.

   Claim 2. *For any $q, p \in Q$,*
   $$\delta^{R*}(q, x) = \{p \mid \delta^*(p, x^R) = q\}.$$

   Proof. Proceed by induction on $|x|$. In the base case, $|x| = 0$, so $x = \varepsilon$. We have that $\delta^{R*}(q, \varepsilon) = \{q\}$. Observe that $\delta^*(q, \varepsilon) = q$, and $\delta^*(p, \varepsilon) \neq q$ for $p \neq q$; hence the claim holds.

   Now consider the general case of $x = by$, $b \in \Sigma$, $y \in \Sigma^*$. There are no ε-transitions in M^R between two states of Q, so we treat it as an ordinary NFA for purposes of this proof. We first observe that

   $$\begin{aligned}\delta^{R*}(q, by) &= \delta^{R*}(\delta^R(q, b), y)\\ &= \bigcup_{r \in \delta^R(q,b)} \delta^{R*}(r, y).\end{aligned}$$

   Applying our inductive hypothesis, we have that

   $$\delta^{R*}(r, y) = \{s \mid \delta^*(s, y^R) = r\},$$

   and so

   $$\begin{aligned}\delta^{R*}(q, by) &= \bigcup_{r \in \delta^R(q,b)} \{s \mid \delta^*(s, y^R) = r\}\\ &= \bigcup_{\{r \mid \delta(r,b) = q\}} \{s \mid \delta^*(s, y^R) = r\}\\ &= \{s \mid \delta^*(s, y^R b) = q\},\end{aligned}$$

which is what we want. □

Note that the claim holds equally well if we interchange x and x^R, since $(x^R)^R = x$. We now complete our proof of correctness by establishing that

Claim 3. *$x \in L(M)$ iff $x^R \in L(M^R)$.*

Proof. ($\rightarrow$) Suppose $x \in L(M)$. Then for some $q \in A$, $\delta^*(q_0, x) = q$. By the previous claim, it follows that $q_0 \in \delta^{R*}(q, x^R)$. Conclude that, on reading x^R, the machine M^R could follow the path $q_s \rightarrow q \rightarrow^* q_0$; since $q_0 \in A^R$, M^R accepts x^R.

($\leftarrow$) Suppose $x^R \in L(M^R)$. Then we have that $q_0 \in \delta^{R*}(q_s, x^R)$. Now the first step of any accepting computation for x^R must move from q_s to some state $q \in A$ without consuming any input, and so we have $q_0 \in \delta^{R*}(q, x^R)$. By the previous claim, it follows that $\delta^*(q_0, x) = q$, and so M accepts x. □

4. (a) Let $M = (Q, \Sigma, q_0, A, \delta)$ be a DFA accepting L. Define the NFA $M_c = (Q, \Sigma, q_0, A, \delta_c)$ that is identical to M except for a new transition function defined as follows: for every $a \in \Sigma$,

$$\delta_c(q, a) = \{p \mid \exists b \in \Sigma \text{ s.t. } \delta(q, b) = p\}.$$

In other words, on any input symbol, M_c moves from q to the set of all states reachable in one step from q in M (on every input symbol).

We first establish that for $x \in \Sigma^*$,

Claim 4.

$$\delta_c^*(q, x) = \{p \mid \exists y \in \Sigma^* \text{ s.t. } |x| = |y| \text{ and } \delta^*(q, y) = p\}.$$

Proof. We proceed by induction on $|x|$. In the base case, $x = \varepsilon$. We have that $\delta_c^*(q, \varepsilon) = \{q\}$, and indeed, for every string y of length 0 (i.e. for ε), y takes M from q to q.

In the general case, let $x = ya$. We have that

$$\begin{aligned} \delta_c^*(q, ya) &= \delta_c(\delta_c^*(q, y), a) \\ &= \delta_c(\psi(q, y), a) \end{aligned}$$

where $\psi(q, y)$ is the set of all p for which some $z \in \Sigma^*$ exists, $|z| = |y|$, for which $\delta^*(q, z) = p$. This second step follows by the inductive hypothesis. Now

$$\begin{aligned} \delta_c(\psi(q, y), a) &= \bigcup_{r \in \psi(q,y)} \delta_c(r, a) \\ &= \bigcup_{r \in \psi(q,y)} \{s \mid \exists b \in \Sigma \text{ s.t. } \delta(r, b) = s\}. \end{aligned}$$

Observe that, if some string z of length $|y|$ takes M from q to r, and some symbol b takes M from r to s, then the string zb takes M from q to s. Conclude that

$$\delta_c^*(q, ya) = \{s \mid \exists zb \in \Sigma^* \text{ s.t. } |zb| = |ya| \text{ and } \delta^*(q, zb) = s\},$$

which is what we want. □

We now complete our proof of correctness by establishing that

Claim 5. *$x \in L(M_c)$ iff there exists $y \in L(M)$ with $|y| = |x|$.*

Proof. By definition of an NFA, $x \in L(M_c)$ iff, for some $q \in A$, $q \in \delta_c^*(q_0, x)$. By our previous claim, this holds iff there exists $y \in \Sigma^*$ with $|y| = |x|$ for which $\delta^*(q_0, y) = q$. But this is true iff $y \in L(M)$, which proves our claim. ☐

(b) Let $M = (Q, \Sigma, q_0, A, \delta)$. We construct an ε-NFA $M_{1/2}$ from M in stages as follows.

  i. For each $q \in Q$, create a new machine M_q that is identical to M except that its start state is q. Clearly, M_q accepts x iff $\delta^*(q, x) \in A$.

  ii. For each M_q, construct a machine M_{qc} using the construction of part (a). By that construction, M_{qc} accepts x iff there exists y, $|x| = |y|$, such that $y \in L(M_q)$, that is, if $\delta^*(q, y) \in A$.

  iii. For each $q \in Q$, create a machine M'_q that is identical to M except that it has the single accepting state q. Observe that $x \in L(M'_q)$ iff $\delta^*(q_0, x) = q$.

  iv. Use the suggested Cartesian product construction to construct, for each $q \in Q$, a machine $M_{q/2}$ that accepts x iff both M_{qc} and M'_q accept x. By construction of these two component machines, $M_{q/2}$ accepts x iff $\delta^*(q_0, x) = q$, and there exists a string y, $|y| = |x|$, for which $\delta^*(q, y) \in A$. In other words, $M_{q/2}$ accepts x iff for some y of equal length, $xy \in L(M)$ *and* the "middle state" of the accepting computation is q.

  v. Finally, use the standard union construction from class to create a ε-NFA $M_{1/2}$ that accepts $\bigcup_{q \in Q} L(M_{q/2})$, which is exactly $L(M)_{1/2}$.

5. In this problem we use the notation $\ell(M)$ to denote the set of live states of M.

   Let M be a DFA accepting an infinite language L. Since M has only finitely many accepting states but accepts infinitely many strings, there exists some accepting state q^* of M such that, for two strings $x, y \in L$ with $|x| < |y|$, $\delta^*(q_0, x) = \delta^*(q_0, y) = q^*$.

   Let M_x be a simple linear DFA that accepts the language $\{x\}$. M_x contains $|x| + 1$ states q_i, such that $\delta(q_i, x[i]) = q_{i+1}$, plus one dead state q_d that is the target of all transitions not defined above. The state $q_{|x|}$ is the only accepting state of M_x.

   Now consider the DFA M' obtained by taking the cross product of M with M_x and assigning the accepting states so that M' accepts $L(M) \cup L(M_x)$. (That is, state (q, q') of M' is accepting if either q is accepting in M or q' is accepting in M_x.) We claim that M' has strictly more live states than does M. Indeed, every live state q of M corresponds to at least one live state (q, q') of M', where q' is some state of M_x. Hence, $|\ell(M')| \geq |\ell(M)|$. But for state q^* of M, it must be that *both* $(q^*, q_{|x|})$ and (q^*, q_d) are live (indeed, accepting!) in M', since M accepts both x and a longer string y at q^*. Conclude that $|\ell(M')| > |\ell(M)|$, as claimed. Moreover, since $x \in L$, $L(M') = L(M)$.

   (*Note*: this argument is actually stronger than the original claim, since it shows that we can always build a machine to accept $L(M)$ with strictly more *accepting* states. Every accepting state is live, but the converse is not true.)

   The argument does not work if L is finite because the initial pigeonhole argument for the existence of q^* fails. Hence, we may not be able to find a string x to split out in the above construction.