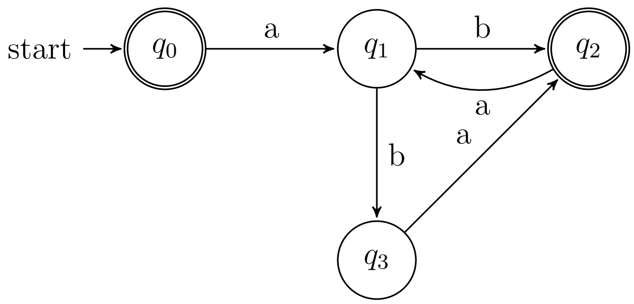
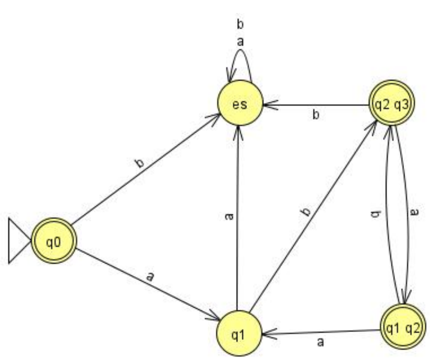
MATH 4LT/6LT3 Assignment #2 Solutions
Due: Friday, 3 October by 11:59pm.

1. Consider the following NFA:



Convert this NFA into an equivalent DFA using the procedure provided in the proof of Theorem 1.3.20. **Note**: You may disregard any states in the construction that cannot be reached from the initial state. So your solution should have far fewer than 16 states.

Solution: The following is a diagram for the DFA produced from this NFA using the construction in the proof of Theorem 1.3.20. Note that only 5 of the 16 states are included, since the remaining ones do not play a role in the processing of a string. The 5 states are labelled according to the subset of $\{q0, q1, q2, q3\}$ they correspond to. The label es is for the empty set.



2. Show that each of the following languages is not regular:

(a) $L_0 = \{0^n 1^n 2^n \mid n \geq 0\}$.

Solution: It suffices to find an infinite set of strings that is pairwise separable with respect to L_0. The following set works (but there are many others):

$$Z_0 = \{0^n \mid n \geq 0\}.$$

To see this, let $0 \leq i < j$ and let $w = 1^i 2^i$. Then $0^i w \in L_0$ but $0^j w$ is not.

(b) $L_1 = \{www \mid w \in \{a,b\}^*\}$.

Solution: Let

$$Z_1 = \{a^n b \mid n \geq 0\}.$$

Z_1 is pairwise separable with respect to L_1 since if $0 \leq i < j$, we have, with $v = a^i b a^i b$, $a^i b v \in L_1$ but $a^j b v$ is not. Since $i \neq j$, then there is no way to break $a^j v$ into three identical substrings, since such a substring must end in b and contain exactly one b.

(c) $L_2 = \{a^m b^n \mid m \neq n\}$.

Solution: There are a couple of ways to solve this. One is to use some of the closure properties of regular languages and the fact that the language $L = \{a^n b^n \mid n \geq 0\}$ is not regular (shown in class). Note that $L = L(a^* b^*) \cap \overline{L_2}$. Since the collection of regular languages over $\{a,b\}$ is closed under intersection and complementation, then if L_2 is regular, so is $\overline{L_2}$ and hence so is $L(a^* b^*) \cap \overline{L_2} = L$, a contradiction.

Alternatively, it can be seen that the set $\{a^i \mid i \geq 0\}$ is pairwise separable with respect to L_2.

(d) $L_3 = \{x = y + z \mid |x| = |y| + |z| \text{ for } x, y, z \in \{1\}^*\}$. L_3 is a language over the alphabet $\{1, =, +\}$.

Solution: Let

$$Z_3 = \{1^n = \mid \ n > 1\}.$$

Z_3 is pairwise separable with respect to L_3 since if $2 \leq i < j$, we have, with $v = 1^{i-1} + 1$, $1^i = v \in L_3$ but $1^j = v$ is not.

3. Let Σ be an alphabet and $w \in \Sigma^*$. If $w = a_1 a_2 \dots a_k$ for some $k \geq 0$ and $a_i \in \Sigma$, define the reverse of w to be the string $w^r = a_k a_{k-1} \dots a_2 a_1$. Show that if L is a regular language, then so is $L^r = \{w^r \mid w \in L\}$.

   Solution: One way to prove this is to take a DFA M that accepts L and modify it so that the resulting NFA M^r accepts L^r. To modify M, "reverse" its transitions, i.e., if $\delta(q, a) = q'$ is a transition of M, then include q in the set $\delta^r(q', a)$, where δ^r is the transition function of M^r. Further, declare the initial state of M to be the only final state of M^r and add a new initial state R to M^r. For each final state q of M, add q to the transition $\delta^r(R, \epsilon)$.

   If a string w is accepted by M, then after starting in its initial state, M will end up in one of its accepting states, q. By reversing this path of transitions in M, it can be seen that it will be an accepting path for w^r in the NFA M^r, after first making an ϵ transition from the state R to the state q.

   On the other hand, if M^r accepts some word w, then the path of transitions from the initial state of M^r to its only accepting state must first pass through one of the accepting states q of M, and then follow a chain of δ^r transitions to end up at the initial state of M. By reversing this chain of transitions, we obtain one that starts at the beginning of w^r, in the initial state of M, and ends up in the state q, one of the final states of M. Thus, w^r is accepted by M. This establishes that for any word w, M accepts w if and only if M^r accepts w^r, showing that L^r is also regular.

4. Exercise 1.9.7.

   Solution: Let L be a finite language over the alphabet Σ. We will prove by induction on the size of L that it is regular. If $|L| = 0$ then L is the empty language, and it is easy to come up with a DFA M with $L(M) = \emptyset$ (any DFA that has no accepting states will work).

   Next, suppose that $|L| = 1$, say $L = \{w\}$, where $w = a_1 a_2 \cdots a_k$ for some $k \geq 0$ and $a_i \in \Sigma$. We will show by induction on k that L is regular. When $k = 0$, $w = \epsilon$ and it is not hard to produce a DFA that only accepts this string (a 2 state DFA can be found with this property). Similarly, if $k = 1$, then there is a 3 state DFA that only accepts the string a_1. Using that the concatenation of two regular languages is regular and that both of the languages $\{a_1 a_2 \cdots a_k\}$ (by the induction hypothesis) and $\{a_{k+1}\}$ are regular, it follows that $\{a_1 a_2 \cdots a_k a_{k+1}\}$ is as well.

Now, suppose that $n \geq 1$ and that the claim holds for all languages of size at most n. Assume that $|L| = n + 1$ and let w be any member of L. Let $L_0 = L \setminus \{w\}$ and let $L_1 = \{w\}$. By induction, L_0 and L_1 are regular and so their union, L, is also regular.

An alternate solution of this is to first show that for each string $w = a_1 a_2 \ldots a_k$, there is a $k+1$-state NFA N_w with $L(N_w) = \{w\}$. With the set of states of N_w equal to $\{0, 1, \ldots, k\}$, 0 will be the start state, and k will be the only accept state. For each $i < k$, there is one transition from state i to $i + 1$, labelled by the symbol a_{i+1}. If $L = \{w_1, w_2, \ldots, w_n\}$, then we can set N to be the NFA that is the union of the NFAs N_{w_1} through to N_{w_n}. Here, we assume that the set of states of these NFAs are disjoint (by suitably renaming them), the set of starts states consists of the start states of each of the NFAs and the set of accept states consists of the set of accept states of the NFAs. Then $L(N) = L$.

5. Exercise 1.9.21.

Solution: For B a subset of $\mathbb{N}$, let $L_B = \{a^n \mid n \in B\}$. Let S be ultimately periodic, with n and p given as in the definition. Let $S_0 = S \cap \{k \in \mathbb{N} \mid k < n\}$ and $S_1 = S \setminus S_0$. So S_0 is a finite set containing all numbers from S that are less than n and S_1 contains the other members of S. Since S_0 is finite, then so is L_{S_0} and hence is regular (by the previous homework question). Since $L_S = L_{S_0} \cup L_{S_1}$ then to show that L_S is regular, it is enough to show that L_{S_1} is.

For each $0 \leq j < p$, let $P_j = \{m \in S_1 \mid m \equiv j \bmod p\}$. Clearly S_1 is the finite union of all of the nonempty P_j. So, it suffices to show that each nonempty L_{P_j} is regular. If P_j is nonempty, let m_j be the smallest member of P_j. Then by the ultimately periodic property, $P_j = \{m_j + kp \mid k \in \mathbb{N}\}$. Then $L_{P_j} = \{a^{m_j}\} \cdot \{a\}^*$. Since both $\{a^{m_j}\}$ and $\{a\}^*$ are regular, then so is L_{P_j}, as required. Thus L_{S_1} and L_S are regular.

For the converse, Let $M = (Q, \Sigma, s, T, \delta)$ be a DFA with $\Sigma = \{a\}$ and let $L = L(M)$. Let $S = \{n \in \mathbb{N} \mid a^n \in L\}$. Define the infinite sequence of members of Q, q_n, inductively by: $q_0 = s$ and for $n \geq 0$, $q_{n+1} = \delta(q_n, a)$. It can be seen that for any $n \geq 0$, after processing the string a^n the DFA M will be in state $q_n \in Q$.

Since Q is finite there will be a smallest $n \in \mathbb{N}$ such that for some $p > 0$, $q_n = q_{n+p}$. Choose p to be the smallest positive integer with this property. If one considers the transition diagram of M as a directed graph, then there will be a path of length n going from s to q_n, traversing the distinct states q_1 up to q_n, followed by a loop of length p, traversing the states q_n up to q_{n+p-1} and then back to q_n. It follows that for any $m \geq n$, $q_m = q_{m+p}$.

Let $S_0 = \{k \in \mathbb{N} \mid k < n \text{ and } a^k \in L\}$ and let $S_1 = \{k \in \mathbb{N} \mid a^k \in L \text{ and } k \geq n\}$. Clearly $S = S_0 \cup S_1$. Furthermore, if $m \in S_1$, then $a^m \in L$, which implies that $q_m \in T$. Since $q_{m+p} = q_m$ in this case, it follows that $a_{m+p} \in L$ and hence that $m + p \in S_1$. This establishes that the set S is ultimately periodic.

The following questions are for students enrolled in MATH 6LT3. Students in MATH 4LT3 can treat them as bonus questions.

B1 For L_1 and L_2 languages over the alphabet Σ, define $L_1 \wr L_2$ to be the language

$$\{w \in \Sigma^* \mid w = a_1 b_1 a_2 b_2 \ldots a_k b_k \text{ for some } k \geq 0,\ a_1 a_2 \ldots a_k \in L_1 \text{ and } b_1 b_2 \ldots b_k \in L_2\}.$$

Prove that if L_1 and L_2 are regular then so is $L_1 \wr L_2$.

Solution: We first prove a special case of this, when one of L_1 or L_2 is Σ^*. Once we have shown that $L_1 \wr \Sigma^*$ and $\Sigma^* \wr L_2$ are regular, then since $L_1 \wr L_2 = L_1 \wr \Sigma^* \cap \Sigma^* \wr L_2$ it will follow that $L_1 \wr L_2$ is also regular.

The argument for showing that $\Sigma^* \wr L_2$ is regular is similar to that for showing that $L_1 \wr \Sigma^*$ is, and so we will only deal with the latter claim. (In fact, $\Sigma^* \wr L_2 = (L_2^r \wr \Sigma^*)^r$ and so we can use the result from question #3 to conclude that it is regular, if L_2 is.)

Suppose that L_1 is regular and let $M = (Q, \Sigma, s, T, \delta)$ be a DFA with $L_1 = L(M)$. We describe how to construct a DFA M' from M with $L(M') = L_1 \wr \Sigma^*$. The idea is that when M' is processing an input string, it ignores every other symbol. To accomplish this, M' will have a copy of each state of M and its transition function will rely on M's when processing the odd-numbered symbols of an input string. Define

$$M' = (Q \cup Q', \Sigma, s', T', \delta'),$$

where $Q' = \{q' \mid q \in Q\}$ is a disjoint copy of Q, $T' = \{t' \in Q' \mid t \in T\}$ and δ' is defined by: $\delta'(q, a) = q'$ and $\delta'(q', a) = \delta(q, a)$ for any $q \in Q$ and $a \in \Sigma$.

We claim that for $w = a_1 b_1 a_2 b_2 \dots a_k b_k$ for some $k \geq 0$ and a_i, $b_i \in \Sigma$, that $w \in L(M')$ if and only if $a_1 a_2 \dots a_k \in L(M)$. To show this we prove by induction on k, that if after processing $a_1 a_2 \dots a_k$, M is in state q, then after processing w, M' will be in state q', and conversely. The claim follows from this since the set of accept states of M' is T'.

When $k = 0$, this property holds since the initial state of M' is the copy of M's initial state. Suppose that the property holds for k and consider the string $v = a_1 b_1 a_2 b_2 \dots a_k b_k a_{k+1} b_{k+1}$. Let q be the state of M after processing $a_1 a_2 \dots a_k$. Then by induction the state of M' after processing $a_1 b_1 a_2 b_2 \dots a_k b_k$ is q'. Then the state of M after processing $a_1 a_2 \dots a_k a_{k+1}$ will be $\delta(q, a_{k+1})$, while the state of M' after processing $a_1 b_1 a_2 b_2 \dots a_k b_k a_{k+1} b_{k+1}$ will be $\delta'(\delta'(q', a_{k+1}), b_{k+1}))$. By the definition of δ', this state will be $\delta(q, a_{k+1})'$, as required. The converse can be proved similarly.

B2 Exercise 1.9.40.

Solution: A statement and proof of the Pumping Lemma can found in most standard references for the foundations of computing. For instance, a proof can be found in Sipser's book on this subject.