As usual, the questions below marked “Extra Practice” are for your own interest, and will not be marked.

1. Use the pumping lemma to show that the language \( L = \{xxx \mid x \in \{0, 1\}^*\} \) is not regular.

   **Solution.** Assume that \( L \) is regular. The pumping lemma states that there is a \( p \) such that for all strings \( s \in L \) there is a partition of \( s \) into strings \( s = uvw \) such that
   
   - \( uv^iw \in L \) for all \( i \geq 0 \).
   - \( |uv| \leq p \).
   - \( |v| > 0 \).

   Consider the string \( s = 0^p1^p10^p1 \), and write \( s = uvw \) as guaranteed by the pumping lemma. Since \( |uv| \leq p \) it follows that \( uv \) is a substring of \( 0^p \) by the definition of \( s \), and so we can write \( v = 0^k \) for some positive integer \( k \). The pumping lemma implies that the string \( uw = 0^{p-k}1^p10^p1 \in L \), which is clearly a contradiction since \( k > 0 \).

2. (a) Show that the language

   \[
   L_1 = \{1^ky \mid y \text{ contains at least } k \text{ 1s for } k \geq 1\}
   \]

   is regular.

   **Solution.** We give an equivalent way of writing down \( L_1 \) which will make the definition of an NFA much more clear. Consider the set

   \[
   L = \{1y \mid y \text{ contains at least one 1} \}.
   \]

   We claim that \( L = L_1 \). Clearly \( L \subseteq L_1 \), so, we prove that \( L_1 \subseteq L \). Consider any \( x \in L_1 \), and write \( x = 1^ky \) where \( y \) has at least \( k \) 1s. Let \( z = 1^{k-1}y \), and observe \( x = 1^ky = 1z; \) since \( y \) has at least one 1 it follows that \( z \) has at least one 1, and so \( x \in L \) and thus \( L_1 \subseteq L \).

   Here is an NFA for the language \( L = L_1 \).
(b) (Extra Practice) Show that the language

$$L_2 = \{1^k y \mid y \text{ contains at most } k \text{ 1s for } k \geq 1\}$$

is not regular.

3. Prove that the following languages over \{0, 1\} are not regular. You may use the pumping lemma and the fact that regular languages are closed under union, intersection, and complement.

(a) \{0^n1^m0^n \mid m, n \geq 0\}.

**Solution.** Let

$$L = \{0^n1^m0^n \mid n \geq 0\} = \{0^n1^m0^n \mid m, n \geq 0\} \cap 0^*10^*.$$

Clearly \(0^*10^*\) is regular, so assume by way of contradiction that \(\{0^n1^m0^n \mid m, n \geq 0\}\) is regular. This implies that \(L\) is regular since regular sets are closed under intersection.

By the pumping lemma there is a positive integer \(p\) so that the pumping conditions hold for every string in \(L\); so, consider the string \(s = 0^p10^p\), and apply the pumping lemma to obtain a partition \(s = uvw\) satisfying the pumping property. Since \(|uv| \leq p\) it follows (just like in the solution for Problem 1) that \(v = 0^k\) for some positive integer \(k\). The pumping lemma implies that the string \(uvw'\) is in \(L\), but this string is clearly not a palindrome. Thus \(L\) is not regular, and so is \(\{0^n1^m0^n \mid m, n \geq 0\}\).

(b) \(\{w \mid w \in \{0, 1\}^* \text{ is not a palindrome}\}\).

**Solution.** We show that the set of palindromes \(P = \{w \mid w = w^R\}\) is not regular; since regular sets are closed under complementation it follows that the above language is also not regular. Just as in the previous solution, assume that \(P\) is regular and consider the string \(s = 0^p10^p\), which is clearly a palindrome. Apply the pumping lemma to write \(s = uvw\) and, exactly as we argued in the previous solution, it must be that \(v = 0^k\) for some positive integer \(k\). The pumping lemma implies that the string \(uvw^2w = 0^{p+k}10^p\) is in \(P\), but this string is clearly not a palindrome. Thus \(P\) is not regular, and neither is the above set.

(c) (Extra Practice) \(\{0^n1^n \mid m \neq n\}\).

4. On this assignment, we complete our study of the “indistinguishability” relation. Let \(\Sigma\) be an alphabet and let \(L \subseteq \Sigma^*\) be a language. Recall that two strings \(x, y\) are distinguishable by \(L\) if there is a string \(z\) such that \(xz \in L, yz \notin L\). If \(x\) and \(y\) are indistinguishable by \(L\) we write \(x \equiv_L y\). On Assignment 2, we showed that for any \(L, \equiv_L\) is an equivalence relation.
Recall that if $M$ is any DFA and $q$ is a state in $M$, we let

$$S(q) := \{x \mid \text{the computation of } M \text{ on } x \text{ ends at } q\}.$$

On Assignment 3, we showed the following claim.

**Claim.** If $M$ is any DFA computing $L$, and $q$ is any state in $M$, then $x \equiv_L y$ for all $x, y \in S(q)$.

(a) Let $L$ be any language, and let $k$ be the number of equivalence classes of $\equiv_L$. Using the above Claim, prove that $k$ is at most the number of states in the smallest DFA computing $L$.

**Solution.** Let $E_1, E_2, \ldots, E_k$ be the equivalence classes of $\equiv_L$, and choose

$$x_1 \in E_1, x_2 \in E_2, \ldots, x_k \in E_k$$

be arbitrarily in each equivalence class; note that $x_i \not\equiv_L x_j$ for each $i \neq j$. Let $M$ be the smallest DFA computing $L$, and let $Q$ be the states of $M$. For each $i = 1, 2, \ldots, k$ let $q_i \in Q$ be the state in $M$ at which the machine will halt when executed on the string $x_i$. We claim that these states are all distinct; that is, $q_i \neq q_j$ for each $i \neq j$.

Suppose that $q_i = q_j$ by way of contradiction. Then, the computation of $M$ on the strings $x_i$ and $x_j$ will end at the state $q = q_i = q_j$, and so $x_i, x_j \in S(q)$. But the Claim implies $x_i \equiv_L x_j$, but this is a contradiction! Thus $q_i \neq q_j$ for all $i \neq j$, and thus $Q$ contains $k$ distinct states. It follows that $|Q| \geq k$.

(b) (Extra Practice) Let $E_1, E_2, \ldots, E_k$ be the equivalence classes of $\equiv_L$. Prove that there exists a DFA computing $L$ with $k$ states. (**Hint:** What should the set of states be? What is the start state?)