Problem 1

Show that the language $L$ such that

$$L = \{ x \in \{a, b\}^* \mid x = x^r \} = \{ \epsilon, 0, 1, 00, 11, 000, 010, 101, 111, 0000, 0110, \ldots \}$$

is not a regular language. Recall that $x^r$ is the reverse of a string $x$.

Solution:

We show that $L$ is not regular by using the contrapositive form of the Pumping Lemma. Given any nonnegative integer $p$, consider any integer $n$ such that $n > p$, and let $w$ be the string $0^n 10^n$. Note that $w \in L$ and $|w| \geq p$. For all strings $x$, $y$ and $z$ from $\{a, b\}^*$ such that $xyz = w$, $|y| > 0$, and $|xy| \leq p$, we want to show that $xy^i z \not\in L$, for some $i \geq 0$. We now claim that $i = 2$ does the job. Since $|xy| \leq p$ and $p < n$, we must have $x = 0^k$, $y = 0^l$, and $z = 0^m 10^n$, with $k + l + m = n$. Since $|y| > 0$ and $l = |y|$, we must have $0 < l \leq p$. So, $xy^2 z = 0^k 0^{2l} 0^m 10^n = 0^{k+l+m+l} 10^n = 0^{n+l} 10^n$. Since $n > p$ and $0 < l \leq p$, we must have that $n + l > n$, which means that $xy^2 z = 0^{n+l} 10^n \not\in L$, as the $(n + 1)$-th symbol of $xy^2 z$ is 0 and the $(n + 1)$-th symbol of $(xy^2 z)^r$ is 1. So, according to the contrapositive of the Pumping Lemma, the language $L$ is not regular.
Problem 2

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. Then, show that $L(D)$ is nonempty if and only if $L(D)$ contains a string whose length is less than $|Q|$, where $|Q|$ is the number of states of $D$.

Solution:

If $L(D)$ contains a string $w$ such that $|w| < |Q|$ then $L(D)$ is obviously nonempty. So, the “if” part of our claim is trivially true. Let us now show that the “only if” part is also true. If $L(D)$ is nonempty then $L(D)$ contains at least one string from $\Sigma^*$. We want to show that at least one string in $L(D)$ has length smaller than $|Q|$. Let $w$ be the shortest string of $L(D)$. If $|w|$ is smaller than $|Q|$ then we are done. So, assume that $|w| \geq |Q|$, i.e., assume that $L(D)$ is nonempty but it does not contain any string whose length is smaller than $|Q|$. Since $L(D)$ is regular, we can invoke the Pumping Lemma to write $w = xyz$ for some $x$, $y$, and $z$ in $\Sigma^*$, with $y \neq \epsilon$ and $|xy| \leq |Q|$, so that $xy^iz \in L(D)$, for all nonnegative integers $i$. This means that $xy^0z \in L(D)$. Since $xy^0z = xz$ and $|y| > 0$ and $|w| = |xyz| \geq |Q|$, we have that $|xy^0z| = |xz| < |xyz| = |w|$. Since $|w| \geq |Q|$ and $|xy| \leq |Q|$, we must also have that (1) either $|xz|$ is smaller than $|Q|$, or (2) $|xz|$ is equal to or greater than $|Q|$, but smaller than $|w| = |xyz|$. If (1) holds then it contradicts our assumption that $L(D)$ has no string with length smaller than $|Q|$. If (2) holds then it also contradicts our assumption that $w$ is the shortest string of $L(D)$ whose length is equal to or greater than $|Q|$. So, $L(D)$ must have some string whose length is smaller than $|Q|$. 


**Problem 3**

Give an algorithm for deciding if two given deterministic finite automata \( D_1 \) and \( D_2 \) recognize the same language. The input for your algorithm is the two quintuples \( D_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1) \) and \( D_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2) \) corresponding to the two automata, and the output of your algorithm is either “yes” or “no”. The output is “yes” if \( L(D_1) = L(D_2) \) and the output is “no” if \( L(D_1) \neq L(D_2) \).

**Solution:**

Our algorithm makes use of the properties of regular languages and of our answer for Problem 2. First, we build a new DFA \( D \) (or NFA or \( \epsilon \)-NFA) from \( D_1 \) and \( D_2 \) to recognize the language \( L = (L(D_1) \cap (\Sigma^* - L(D_2))) \cup ((\Sigma^* - L(D_1)) \cap L_2) \). We do that by using the constructions for union, complementation, and intersection that we saw in class and in recitations. Why \( L \)? By examining \( L \), you can conclude that \( L(D_1) = L(D_2) \) if and only if \( L = \emptyset \). You should prove this fact to convince yourself that it is true. Now, what is the point in building \( D \) such that \( L(D) = L \)? According to our answer to Problem 2, \( L(D) = \emptyset \) if and only if \( L(D) \) does not contain a string whose length is smaller than \( |Q| \), where \( |Q| \) is the number of states of \( D \). Hence, after building \( D \), our algorithm tests \( D \) against all strings of length less than \( |Q| \). If \( D \) does not accept any of them, we output “yes”. Otherwise, we output “no”. **How can we carry out the test of \( D \) against all strings of length less than \( |Q| \) in an efficient way? Do you have a better idea for this algorithm? There are some alternative answers. Think about one of them.**