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

1. In class, we proved the following theorem:

**Theorem.** If $M$ is an NFA accepting a language $L$, then there is an NFA $M'$ accepting the language $L^*$.

This shows that the set of regular languages is closed under the Kleene star operation. Consider the following “attempt” at constructing an NFA $M'$ from $M$ to prove this theorem.

Let $M = (\Sigma, Q_0, q_0, F_0, \delta_0)$ be a description of an NFA with the usual components (alphabet, states, start state, accept states, and transition function, respectively). Define $M' = (\Sigma, Q_1, q_1, F_1, \delta_1)$ as follows:

- $Q_1 = Q_0$: The states of $M'$ are the same as $M$.
- $q_1 = q_0$: The start state of $M'$ is the same as the start state of $M$.
- $F_1 = F_0 \cup \{q_1\}$: keep all the accepting states of $M$, and also make the start state an accepting state.
- Define $\delta_1 : Q_1 \times (\Sigma \cup \{\varepsilon\}) \to P(Q_1)$ as follows. For each state $q \in Q_1$ and character $a \in \Sigma \cup \{\varepsilon\}$,

  $$\delta_1(q, a) = \begin{cases} 
  \delta_0(q, a) & \text{if } q \text{ is not an accepting state or } a \neq \varepsilon \\
  \delta_0(q, a) \cup \{q_1\} & \text{if } q \text{ is an accepting state and } a = \varepsilon.
  \end{cases}$$

That is, $\delta_1$ is the same as $\delta_0$ except we add $\varepsilon$ transitions from the final states of $M'$ to the start state of $M'$.

Show that this construction does not prove the above theorem. That is, give a language $L$ with an NFA $M$ accepting it such that the NFA $M'$ obtained by the above construction does not accept $L^*$.

2. Give regular expressions for the following languages over the alphabet $\Sigma = \{0, 1\}$.
(a) \( \{ x \mid \text{every odd character of } x \text{ is a } 1 \} \).
(b) \( \{ x \mid x \text{ does not contain the substring } 10 \} \).
(c) (Extra Practice) \( \{ x \mid x \text{ contains at least three } 1 \text{s} \} \).
(d) (Extra Practice) \( \{ 0, 1 \}^* \setminus \{ \varepsilon \} \).

3. For any string \( w = w_1 w_2 \cdots w_n \) the reverse of \( w \), denoted \( w^R \), is (unsurprisingly) the string \( w_n w_{n-1} \cdots w_2 w_1 \). Given a language \( A \), let \( A^R = \{ w^R : w \in A \} \). Show that if \( A \) is regular then so is \( A^R \).

4. In this question we will continue the development of the “distinguishability” relation, from Assignment 2. If \( L \) is a language and \( x, y \) are strings recall that \( x \) and \( y \) are distinguishable by \( L \) if there is a string \( z \) such that \( xz \in L, yz \notin L \).

\[
\begin{array}{c}
\text{start} \\
q_\varepsilon \\
\downarrow a \\
q_a \\
\downarrow b \\
q_\varepsilon a \\
\downarrow a \\
q_{aa} \\
\uparrow b \\
\downarrow b \\
q_{aaa} \\
\downarrow a, b \\
q_{aaab} \\
\end{array}
\]

Figure 1: The DFA \( \mathcal{M} \)

(a) Consider the DFA \( \mathcal{M} \) drawn above, which accepts the language

\[ L = \{ x \mid x \text{ contains the substring } aaab \} \, . \]

For any state \( q \) in the automata, let \( S(q) \) be the set of strings defined by

\[ S(q) = \{ x \mid \text{the computation of } \mathcal{M} \text{ on } x \text{ ends at } q \} \, . \]

For example, \( S(q_{aaab}) = L \), and a regular expression for \( S(q_a) \) is \( (b^* a)(b \cup ab)b^* a^* \).

Give regular expressions for \( S(q_\varepsilon) \) and for \( S(q_{aaa}) \).

(b) Let \( x, y \in S(q_a) \) be any two strings that end up at the state \( q_a \) when run on \( M \). Show that \( x \equiv_L y \).

(c) (Extra Practice). Let \( L \) be a regular language and let \( M \) be a DFA computing \( L \). Let \( q \) be any state in \( M \). Show that, for any \( x, y \in S(q) \), \( x \equiv_L y \). (Observe that this implies the number of equivalence classes of \( L \) is at most the number of states in the smallest DFA computing \( L! \)).