1. Give a DFA which accepts the following language.
\[ \{ w \mid w = a_1 b_1 a_2 b_2 \ldots a_n b_n, \text{for some } n, \text{where } a_i, b_i \in \{0, 1\} \text{ and } a_1 a_2 \ldots a_n > b_1 b_2 \ldots b_n \text{ (interpreted as binary numbers)} \} \]
In other words, DFA can decide if \( a_1 a_2 \ldots a_n > b_1 b_2 \ldots b_n \), if the inputs are given in a specific format.

2. For a DFA \( A = (Q, \Sigma, \delta, q_0, F) \), let \( \hat{\delta} \) be as defined in class. Show that \( \hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y) \), for all strings \( x, y \) over \( \Sigma^* \), and all states \( q \in Q \).

3. Give an NFA which accepts the set of strings which end in \( bba \). Use your NFA to construct a DFA which accepts the same language, using the method of converting NFA’s to DFA’s done in class (do not construct a DFA directly, but only via the method discussed in class).

4. For the NFA with \( \epsilon \)-transitions as given in Figure 1:
   (a) give the transition table
   (b) find \( E_{\text{close}}(q) \), for each state \( q \).
   (c) find \( \hat{\delta}(q_2, a) \) and \( \hat{\delta}(q_2, b) \).
   (d) find a DFA which is equivalent to the given automata (you need not go through the formal method discussed in class).

   In the figure, in the transitions, \( e \) denotes \( \epsilon \).

5. Prove or disprove the following:
   (a) \( L((R+S)^*) = L((R^*S^*)^*) \), for all regular expressions \( R \) and \( S \).
   (b) \( L(S(R+S)^*S) = L((SR^*S)^+) \), for all regular expressions \( R \) and \( S \).

6. Use the method discussed in class to give a regular expression for the language accepted by the DFA \( (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_2\}) \), where \( \delta \) is defined as follows.
   \( \delta(q_1, 1) = q_1 \), \( \delta(q_1, 0) = q_2 \), \( \delta(q_2, 1) = q_2 \), \( \delta(q_2, 0) = q_1 \).

7. Consider the DFA given in Figure 2. Give the minimal DFA which accepts the same language as accepted by the DFA in figure 2.
Figure 1: NFA for Q4
Figure 2: DFA for Q7