Problem 1

We say that a state $q \in Q$ of a DFA $D = (Q, \Sigma, \delta, q_0, F)$ is reachable if there exists a string $w \in \Sigma^*$ such that $q = \delta(q_0, w)$. Otherwise, we say that $q$ is unreachable. Devise an algorithm to find out the reachable states of a DFA. The input of your algorithm is a DFA $D = (Q, \Sigma, \delta, q_0, F)$ and the output is the set of reachable states $R \subseteq Q$ of $D$. Your algorithm should use only the five parts of $D$.

Solution:
An efficient algorithm for finding out the reachable states of a DFA $D = (Q, \Sigma, \delta, q_0, F)$ is as follows:

```plaintext
let $S_1 \leftarrow \emptyset$
let $S_2 \leftarrow \{q_0\}$
let $i \leftarrow 0$
while $i \leq |Q|$ do
    let $S_3 \leftarrow \emptyset$
    for each $s \in S_2$ do
        for each $x \in \Sigma$ do
            $q \leftarrow \delta(s, x)$
            if $q \notin S_1$ then
                $S_3 \leftarrow S_3 \cup \{q\}$
            endif
        endfor
    endfor
    $S_2 \leftarrow S_3$
    $S_1 \leftarrow S_1 \cup S_3$
    $i \leftarrow i + 1$
endwhile
```

What is the logic behind this algorithm? Why does it work? Why does it always terminate? Why is it “efficient”? Find out all that during your recitation class!
Problem 2

Recall that, given an NFA $M = (Q, \Sigma, \delta, q_0, F)$, the subset construction builds a DFA $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ such that the number of states in $Q_D$ is exactly $2^{|Q|}$, where $|Q|$ is the cardinality of $Q$. In general, several states of the DFA $D$ resulting from the subset construction are unreachable, which means that they can be removed from the DFA along with the transition from and to them without modifying the language accepted by the DFA. So, it is natural to wonder if there is a “better” way of building the DFA corresponding to an NFA. What do you think? If you think so, devise such a better construction and apply it to the NFA $M = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{s_0, s_1, s_2, s_3\}$, $\Sigma = \{0, 1\}$, $q_0 = s_0$, $F = \{s_2\}$, and the transition function $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ is given by

$$
\begin{array}{c|cc}
\delta & 0 & 1 \\
\hline
s_0 & \{s_1\} & \emptyset \\
s_1 & \{s_2\} & \{s_3\} \\
s_2 & \emptyset & \emptyset \\
s_3 & \emptyset & \{s_1\}
\end{array}
$$

Solution:

An interesting idea is to modify the subset construction so that we build $Q_D$ incrementally. To illustrate our idea, we apply it to the NFA $M$ above. We start with $Q_D = \{\{s_0\}\}$, where $\{s_0\}$ is the initial state of $D$. Then, we compute $\delta(s_0, 0) = \{s_1\}$ and $\delta(s_0, 1) = \emptyset$. This computation tells us that $Q_D$ should contain two more states: $\{s_1\}$ and $\emptyset$, as they are reachable from $s_0$ using a string of length 1. Now, we must find out the states that can be reached from both $\{s_1\}$ and $\emptyset$. To do that, we compute $\bigcup_{p \in \{s_1\}} \delta(p, 0)$, $\bigcup_{p \in \{s_1\}} \delta(p, 1)$, $\bigcup_{p \in \emptyset} \delta(p, 0)$, and $\bigcup_{p \in \emptyset} \delta(p, 1)$. This means that $Q_D$ should also contain the states $\{s_2\}$ and $\{s_3\}$, as they are reachable from $s_0$ using a string of length 2. Now, we must find out the states that can be reached from both $\{s_2\}$ and $\{s_3\}$. To do that, we compute $\bigcup_{p \in \{s_2\}} \delta(p, 0)$, $\bigcup_{p \in \{s_2\}} \delta(p, 1)$, $\bigcup_{p \in \{s_3\}} \delta(p, 0)$, and $\bigcup_{p \in \{s_3\}} \delta(p, 1)$. This time, we have not found any state that we had not found before, so we can stop and state that $Q_D = \{\{s_0\}, \{s_1\}, \{s_2\}, \{s_3\}, \emptyset\}$ (why?). From now on, we define $F_D$ and $\delta_D$ as in the subset construction: $F_D = \{\{s_2\}\}$ and $\delta_D : Q_D \times \Sigma \rightarrow Q_D$, where

$$
\begin{array}{c|cc}
\delta_D & 0 & 1 \\
\hline
\{s_0\} & \{s_1\} & \emptyset \\
\{s_1\} & \{s_2\} & \{s_3\} \\
\{s_2\} & \emptyset & \emptyset \\
\{s_3\} & \emptyset & \{s_1\} \\
\emptyset & \emptyset & \emptyset
\end{array}
$$

Can you tell why this construction works? For our example, we found all states of $Q_D$ in three iterations of our construction “algorithm”. Can you tell how many iterations are needed in general?
Problem 3

Let \( L \) be a language over an alphabet \( \Sigma \). Prove that if there is a DFA \( D = (Q, \Sigma, \delta, q_0, F) \) such that \( L(D) = L \), then there must be an NFA \( M = (Q, \Sigma, \delta_M, \{q_0\}, F) \) such that \( L(M) = L \).

Solution:

Our proof is constructive, i.e., we build an NFA \( M \) out of the given DFA \( D \) and then we show that this NFA does the job! Note that the statement of the problem already tells us that the set of states, the initial states, and the set of final states of \( M \) are the same as the ones of the DFA \( D \). So, we are left with the task of specifying \( \delta_M \). Since we are free to specify \( \delta_M \) as we want, we let \( \delta_M(q, a) = \{ \delta(q, a) \} \), for all \( q \in Q \) and \( a \in \Sigma \). That is, the state diagram of our NFA \( M \) is the same as the state diagram of the DFA \( D \). To prove that \( L(M) = L = L(D) \), we actually show that \( \bar{\delta}(q_0, w) \cap F \neq \emptyset \) if and only if \( \delta(q_0, w) \in F \). So, let us use induction on the length \( |w| \) of \( w \) to show that \( \delta_M(q_0, w) = \{ \bar{\delta}(q_0, w) \} \), for every \( w \in \Sigma^* \).

Base Case: Assume that \( |w| = 0 \). So, \( w = \varepsilon \). Since \( \bar{\delta}(q_0, \varepsilon) = q_0 \) and \( \bar{\delta}_M(q_0, \varepsilon) = \{q_0\} \), we have that \( \delta_M(q_0, w) = \{ \bar{\delta}(q_0, w) \} \), and therefore our claim holds for \( |w| = 0 \).

Induction Hypothesis: Let \( w \in \Sigma^k \) be any string over \( \Sigma \) such that \( |w| = k \), for some integer \( k \) with \( k \geq 0 \). Then we assume that \( \bar{\delta}_M(q_0, w) = \{ \bar{\delta}(q_0, w) \} \).

Inductive Step: Consider the case in which \( |w| = k + 1 \), i.e., \( w \in \Sigma^{k+1} \). Since \( k + 1 > 0 \), we can write \( w \) as \( xa \), for some \( x \in \Sigma^k \) and \( a \in \Sigma \). So, \( \bar{\delta}(q_0, w) = \bar{\delta}(q_0, xa) = \bar{\delta}(\bar{\delta}(q_0, x), a) \) and \( \bar{\delta}_M(q_0, w) = \bar{\delta}_M(q_0, xa) \). Let \( q = \delta(q_0, x) \in Q \). From the induction hypothesis, \( \bar{\delta}_M(q_0, x) = \{ \bar{\delta}(q_0, x) \} = \{q\} \). So, we have that \( \bar{\delta}(q_0, w) = \bar{\delta}(q_0, xa) = \bar{\delta}(\bar{\delta}(q_0, x), a) = \delta(q, a) \) and \( \bar{\delta}_M(q_0, w) = \bar{\delta}_M(q_0, xa) = \bigcup_{p \in \bar{\delta}_M(q_0, x)} \delta_M(p, a) = \bigcup_{p \in \{q\}} \delta_M(p, a) = \delta_M(q, a) \). By definition of \( \delta_M \), we have \( \delta_M(q, a) = \{ \delta(q, a) \} \). Since \( \bar{\delta}_M(q_0, w) = \delta_M(q, a) \) and \( \bar{\delta}(q_0, w) = \delta(q, a) \), we can conclude that \( \bar{\delta}_M(q_0, w) = \{ \bar{\delta}(q_0, w) \} \). So, our claim also holds for \( w \in \Sigma^{k+1} \). From the first principle of induction, our claim is true for every \( w \in \Sigma^* \).
Problem 4

Let $L$ be a language over an alphabet $\Sigma$. Prove that if there is an NFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = L$, then there must be a DFA $D = (Q_D, \Sigma, \delta_D, q_{D0}, F_D)$ such that $L(D) = L$.

Solution:

To prove that $L(D) = L = L(N)$, it suffices to show that $\delta_D(q_{D0}, w) = \delta(q_0, w)$, for all $w \in \Sigma^*$. Note that both $\delta_D$ and $\delta$ return a set of states from $Q$, but $\delta_D$ “sees” this set as one state of $Q_D = \mathcal{P}(Q)$, and $\delta$ “sees” this set as a subset of states of $Q$. Why does $\delta_D(q_{D0}, w) = \delta(q_0, w)$, for all $w \in \Sigma^*$, guarantee that $L(D) = L = L(N)$? Because this condition implies that $\delta_D(q_{D0}, w) \in F_D$ if and only if $\delta(q_0, w) \cap F \neq \emptyset$, as $F_D$ is defined by the subset construction as the set of all subsets of $Q$ that contains at least one state from $F$. Now, we are ready to prove our claim. By now, you should guess that our proof is an induction on the length $|w|$ of $w$.

Base Case: Assume that $|w| = 0$. So, $w = \epsilon$. By definition of $\tilde{\delta}_D$ and $\tilde{\delta}$, we have that $\tilde{\delta}_D(q_{D0}, \epsilon) = q_{D0}$ and $\tilde{\delta}(q_0, \epsilon) = \{q_0\}$. From the subset construction, we know that $q_{D0} = \{q_0\}$. So, our claim holds for $|w| = 0$.

Induction Hypothesis: Let $w \in \Sigma^k$ be any string over $\Sigma$ such that $|w| = k$, for some integer $k$ with $k \geq 0$. Then, we assume that $\tilde{\delta}(q_0, w) = \tilde{\delta}_D(q_{D0}, w)$.

Inductive Step: Let $w \in \Sigma^{k+1}$ be any string over $\Sigma$ such that $|w| = k + 1$. Since $k + 1 > 0$, we can write $w = xa$, where $x \in \Sigma^k$ and $a \in \Sigma$. By the induction hypothesis, $\tilde{\delta}(q_0, x) = \tilde{\delta}_D(q_{D0}, x)$. From the definition of $\tilde{\delta}$, we have that

$$\tilde{\delta}(q_0, w) = \bigcup_{r \in \tilde{\delta}(q_0, x)} \delta(r, a) = \bigcup_{r \in \{p_1, p_2, \ldots, p_n\}} \delta(r, a),$$

where $\{p_1, p_2, \ldots, p_n\} = \delta(q_0, x)$. On the other hand, the subset construction tells us that

$$\tilde{\delta}_D(\{p_1, p_2, \ldots, p_n\}, a) = \bigcup_{r \in \{p_1, p_2, \ldots, p_n\}} \delta(r, a).$$

Since $\tilde{\delta}_D(q_{D0}, x) = \{p_1, p_2, \ldots, p_n\}$ and $\tilde{\delta}_D(q_{D0}, w) = \tilde{\delta}_D(q_{D0}, x, a)$, we have that

$$\tilde{\delta}_D(q_{D0}, w) = \tilde{\delta}_D(\{p_1, p_2, \ldots, p_n\}, a) = \bigcup_{r \in \{p_1, p_2, \ldots, p_n\}} \delta(r, a).$$

But, since $\tilde{\delta}(q_0, w) = \bigcup_{r \in \{p_1, p_2, \ldots, p_n\}} \delta(r, a)$, we can conclude that $\tilde{\delta}(q_0, w) = \tilde{\delta}_D(q_{D0}, w)$ for all $w \in \Sigma^{k+1}$. From the first principle of induction, our claim holds for all $w \in \Sigma^*$. 