Problem 1

Part (a). Consider the DFA $M_1 = (Q, \text{start}, F, \Sigma, \delta)$ defined by $Q = \{q_0, q_1, q_2, q_3\}$, $\text{start} = q_0$, $F = \{q_0, q_2\}$, $\Sigma = \{0, 1\}$ and $\delta$ given by the following table. Give the state diagram for this machine.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_1$</td>
<td>$q_1$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_0$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$q_3$</td>
<td>$q_3$</td>
</tr>
</tbody>
</table>

Solution:
There are four states and the start state is $q_0$ and accepting states are $q_0$ and $q_2$. The following figure depicts the state diagram for this machine.

Part (b). Give the formal description (i.e., the 5-tuple $(Q, \text{start}, F, \Sigma, \delta)$) of the DFA $M_2$ represented by the state diagram in Figure 1.

Solution:
The formal description of the DFA $M_2$ is as follows:
$Q = \{q_0, q_1, q_2\}$, $\text{start} = q_0$, $F = \emptyset$, $\Sigma = \{a, b\}$ and $\delta$ given by the following table:

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_2$</td>
<td>$q_0$</td>
</tr>
</tbody>
</table>

Notice there are no accepting states in the state diagram; hence we have an empty set $\emptyset$ for the set of the accepting states, $F$. 

Part (c). For the DFA $M_1$ given in part (a), and for each of the following input strings $w$, give the computation of $M_1$ on $w$, and say whether the computation is accepting or not. (Note: The computation is the sequence of the states the DFA goes through on the given input.)

Solutions:

1. $w = 1011001$ → computation: $q_0, q_1, q_2, q_3, q_3, q_3, q_3$; not accepting
2. $w = 01100101$ → computation: $q_0, q_1, q_2, q_3, q_3, q_3, q_3, q_3$; not accepting
3. $w = 00101011$ → computation: $q_0, q_1, q_2, q_0, q_1, q_2, q_3$; not accepting

Problem 2

Give the state diagram of DFAs recognizing the following languages. In all cases the alphabet is $\Sigma = \{a, b\}$.

Solutions:

1. $\{w \mid w \text{ contains at most } 3 \text{ } a\text{'s}\}$

   The machine does not accept a string that contains more than 3 $a$’s. Hence, it needs to be able to count the number of $a$’s up to three of them, and after the input exceeds the number, it should go to a non-accepting state no matter what the input symbols are. Since the number of $b$’s doesn’t matter, the machine can just loop on $b$ in the same state in each state.
2. \{w | w \text{ has odd length and has exactly 1 } b\}

The machine does not accept a string that contains more than one \(b\), hence any string with more than one \(b\) should go to a non-accepting state. Since the machine accepts odd length string with exactly one \(b\), the number of \(a\)'s in an accepted string should be even. There are two possibilities for this: one path is where the input \(b\) is in between an even number of \(a\)'s, and the other path is where the input \(b\) is in between an odd number of \(a\)'s. The state diagram depicts these two paths and the non-accepting path as well.

![State Diagram](diagram1.png)

3. \{w | w \text{ doesn’t contain the substring } aab\}

The machine accepts strings that don’t contain the substring \(aab\); hence if the input contains the substring \(aab\), then it should go to a non-accepting state. Otherwise the machine should accept. This is similar to a machine that accepts the strings that contain the substring \(aab\) except that the accepting states are reversed.

![State Diagram](diagram2.png)

**Problem 3**

Show that the difference of regular languages is regular – that is, given an automaton \(M_1\) accepting \(L_1\) and another automaton \(M_2\) accepting \(L_2\), build an automaton \(M\) that accepts \(L_1 - L_2\). (Remember the difference of two sets \(L_1 - L_2\) is the set of all strings that belongs to \(L_1\) but not to \(L_2\).

**Solution:**

Proof by construction – given the DFAs \(M_1\) and \(M_2\) that accept (recognize) \(L_1\) and \(L_2\), respectively, construct a DFA \(M\) that accepts (recognizes) \(L_1 - L_2\). Let \(M_1 = (Q_1, s_1, F_1, \Sigma, \delta_1)\) recognizes \(L_1\) and \(M_2 = (Q_2, s_2, F_2, \Sigma, \delta_2)\) recognizes \(L_2\). Then we can construct a DFA \(M = (Q, s, F, \Sigma, \delta)\) that recognizes \(L_1 - L_2\) as follows:
1. $Q = \{(r_1, r_2) | r_1 \in Q_1 \land r_2 \in Q_2\}$. This set is the Cartesian product of sets $Q_1$ and $Q_2$ and is written $Q_1 \times Q_2$. It is the set of all pairs of states, the first from $Q_1$ and the second from $Q_2$.

2. $s$, the start state, is the pair $(s_1, s_2)$.

3. $F$ is the set of pairs of states in which the first is an accept state of $M_1$ and the second is not accepting state of $M_2$. This is because to recognize the language $L_1 - L_2$, we need to accept the strings that are accepted by $M_1$ but not by $M_2$. We can write it as $F = \{(r_1, r_2) | r_1 \in F_1 \land r_2 \notin F_2\}$. This expression is the same as $F = F_1 \times (Q_2 - F_2)$.

4. $\Sigma$, the alphabet is the same as in $M_1$ and $M_2$. This is because we assume for simplicity that both machines $M_1$ and $M_2$ have the same input alphabet $\Sigma$.

5. $\delta$, the transition function, is defined as follows. For each $(r_1, r_2) \in Q$ and each $a \in \Sigma$, let $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$. Hence $\delta$ gets the state of $M$ (which is a pair of states from $M_1$ and $M_2$), together with an input symbol, and returns $M$'s next state.

This concludes the construction of the finite automaton (DFA) $M$ that recognizes the difference of $L_1$ and $L_2$.

**Problem 4**

This is similar to the first problem, but with nondeterministic finite automata (NFA) instead of deterministic ones.

**Part (a).** Consider the NFA $M_3 = (Q, \text{start}, F, \Sigma, \delta)$ defined by $Q = \{q_0, q_1, q_2, q_3\}$, $\text{start} = q_0$, $F = \{q_2\}$, $\Sigma = \{0, 1\}$ and $\delta$ given by the following table. Give the state diagram for this machine.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>${q_0}$</td>
<td>${q_0, q_1}$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>${q_2}$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>${q_0, q_3}$</td>
<td>$\emptyset$</td>
<td>${q_3}$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>${q_3}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

**Solution:**

![State Diagram](image)
Part (b). The following is the state diagram of an NFA, $M_4$. Give the formal
description of the machine $M_4$ represented by the diagram in Figure 2.

![Figure 2: NFA $M_4$]

**Solution:**
$M_4 = (Q, \text{start}, F, \Sigma, \delta)$ is defined by $Q = \{q_0, q_1, q_2\}$, $\text{start} = q_0$, $F = \{q_0\}$,
$\Sigma = \{0, 1\}$ and $\delta$ given by the following table.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>${q_1}$</td>
<td>${q_1, q_2}$</td>
<td>${q_2}$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>${q_0, q_1}$</td>
<td>${q_1}$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\emptyset$</td>
<td>${q_1, q_2}$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

Part (c). For the NFA $M_4$ given in part (b), give an accepting computation for the
input string $w = 0101100$. (Note: since the automaton is nondeterministic, there
might be more than a single computation, some accepting, some not. The automaton
accept the input string if any of the computations is accepting.) Remember that for
nondeterministic automata, a computation is defined as a sequence $r_0w_1r_1w_2r_2 \ldots r_n$,
that alternates states $r_i \in Q$ and symbols $w_i \in \Sigma \cup \{\epsilon\}$.

**Solution:**
$q_00q_11q_00q_01q_10q_10q_0$