1. Using the method from class, give an NFA that recognizes $L_1 \cup L_2$, where $L_1$ and $L_2$ are the languages from Exercise 5 of Homework 1.

Solution:
2. Using the method from class, give an NFA that recognizes $L_1 \circ L_2$, where $L_1$ and $L_2$ are the languages from Exercise 5 of Homework 1.

Solution:

3. Using the method from class, give an NFA that recognizes $L_1^*$, where $L_1$ is the language from Exercise 5 of Homework 1.

Solution:

4. Give regular expressions for the following languages:

   (a) $\{w \in \{0,1\}^* | w$ starts with a 0 and has odd length $\}$. 
Solution: 0((0 ∪ 1)(0 ∪ 1))∗
(b) \{w ∈ \{0, 1\}∗|w contains 011 as a substring\}.
Solution: (0 ∪ 1)∗011(0 ∪ 1)∗
(c) \{w ∈ \{0, 1\}∗|w does not contain 011 as a substring\}.
[There is a simple regular expression for this language, but it requires
some thinking to find it.]
Solution: 1∗(0 ∪ 01)∗
(d) \{w ∈ \{0, 1\}∗|one of the last three symbols in w is a 1\}.
Solution: (0 ∪ 1)∗1(0 ∪ ε)(0 ∪ ε)

5. Convert the regular expression ((a ∪ b)c)∗ into an NFA using the method
from class (which is the same as the method from the book and is different
from the method in JFLAP).

Solution:
First we have the following NFAs for a, b, and c.

```
a: --→ a
    □
```

```
b: --→ b
    □
```

```
c: --→ c
    □
```

Then, we obtain an NFA for a ∪ b
```
    □
  ε --→ a
    □
```

```
    □
  ε --→ b
    □
```

and the following for (a ∪ b)c.
Finally, we obtain the NFA below for \(((a \cup b)c)^*\).

6. Convert the DFA $M_1$ in Exercise 1.1 of the textbook into a regular expression using the method from class (which is the same as the method in the book, and is not the same as the method in JFLAP).

**Solution:**

Turning this into a GNFA, we get
Eliminating state $q_1$ gives

and the regular expression is $b^*a((a \cup b)(bb^*a \cup a))^*$.

7. **Problem 1.31** Show that if $A$ is regular so is $A^R$.

**Proof:** Since $A$ is regular, there is some NFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes $A$. We construct an NFA $M'$ such that $L(M') = A^R$. The idea is that the start state of $M'$ is the accept state of $M$, the accept state of $M'$ is the start state of $M$, and for every transition in $M$, there is a transition with the same label in $M'$, but going in the reverse direction. The problem with this is that $M$ may have more than one accept state. Since $M'$ can have only one start state, we add a new start state to $M'$ and $\varepsilon$ transitions from the new start state to all the accept states of $M$. The formal construction is:
\[ M' = (Q \cup \{q_s\}, \Sigma, \delta', q_s, \{q_0\}) \]

where \( q_s \) is a new state and

\[
\delta'(q, a) = \begin{cases} 
\{ p : q \in \delta(p, a) \} & \text{if } q \neq q_s \\
F & \text{if } q = q_s, a = \varepsilon \\
\emptyset & \text{if } q = q_s, a \neq \varepsilon
\end{cases}
\]

8. (a) Let \( M \) be the NFA given in the solution to Problem 5(b) on Homework 1. Give an NFA \( N \) with three states and no \( \varepsilon \)-transitions that recognizes the same language.

Solution:

\[ 0 \quad 1 \quad 2 \]

(b) Generalize what you did in Part (a) of this problem by proving the following theorem:

**Theorem:** If \( M \) is an NFA, then there is an NFA \( N \) with the following properties

1. \( N \) has the same number of states as \( M \).
2. \( N \) has no \( \varepsilon \)-transitions.
3. \( L(N) = L(M) \).

**Solution:** We obtain \( N \) from \( M \) by doing three things:

1. We remove all \( \varepsilon \)-transitions.
2. If in \( M \) there is a series of \( \varepsilon \)-transitions that leads from a state \( p \) to an accept state \( q \), then in \( N \), \( p \) is also an accept state.
3. If in \( M \) there is a series of \( \varepsilon \)-transitions from a state \( p \) to a state \( q \), and \( M \) can go from \( q \) to a state \( r \) reading a symbol \( a \), then \( N \) can go directly from \( p \) to \( r \) reading \( a \).

Formally, let \( M = (Q, \Sigma, \delta, q_0, F) \). We define \( N = (Q, \Sigma, \delta', q_0, F') \).

If \( q \) is a state in \( Q \), we use the notation \( \mathcal{E}_M(q) \) for the set of states that can be reached from \( q \) using 0 or more \( \varepsilon \)-transitions in \( M \). The definition of \( N \) is now given by

\[ F' = \{ q \in Q | \mathcal{E}_M(q) \cap F \neq \emptyset \} \]
and

\[ \delta'(q, a) = \begin{cases} \emptyset & \text{if } a = \varepsilon \\ \bigcup \{ \delta(p, a) \mid p \in E_M(q) \} & \text{if } a \neq \varepsilon \end{cases} \]