1. A language over an alphabet of size 3, happens to be recognized by an NFA, $N$, that has 11 states and 5 $\varepsilon$-transitions. Suppose we apply the subset construction to mechanically convert $N$ into an equivalent DFA $D$. What is the maximum possible number of states that $D$ could have?

**Solution:** $D$ could have up to $2^{11} = 2048$ states. The alphabet size and the number of $\varepsilon$-transitions are immaterial.

2. Draw an NFA that recognizes the language \{a^i b^j c^k : i \geq 0, j \geq 1, k \geq 2\}. Keep it simple!

**Solution:**

```
q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{c} q_2 \xrightarrow{c} q_3
```
3. Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be two DFAs over the same alphabet $\Sigma$. Write a formal description of a DFA that recognizes the language $L(M_1) \cap L(M_2)$. No proof of correctness required.

Solution: The following DFA, $M$, recognizes $L(M_1) \cap L(M_2)$.

$$M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2),$$

where $\delta$ is given by

$$\delta((q, r), a) = (\delta_1(q, a), \delta_2(r, a)), \quad \forall q \in Q_1, r \in Q_2, a \in \Sigma.$$

4. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and let $x = a_1 a_2 \ldots a_n$ be a string with each $a_i \in \Sigma$. Write a formal mathematical definition of what we mean when we say “$M$ accepts $x$.”

Solution: “$M$ accepts $x$” means that there exists a sequence $r_0, r_1, \ldots, r_n$ with each $r_i \in Q$ such that

- $r_0 = q_0$,
- $\delta(r_i, a_{i+1}) = r_{i+1}$, $\forall i$ with $0 \leq i < n$, and
- $r_n \in F$. 
5. Write a regular expression for the language \( \{ x \in \{ R, G, B \}^* : x \text{ has an odd number of } R \text{'s} \} \).
You do not have to give a proof of correctness.

**Solution:** The intuition is that any string in the given language consists of
- an initial block of symbols ending in the string's first \( R \), followed by
- zero or more blocks, each containing two \( R \)'s and ending in the second \( R \), followed by
- a final block of symbols containing no \( R \)'s.

Thus, the following regular expression generates the language:
\[
(G \cup B)^* R ((G \cup B)^* R (G \cup B)^* R)^* (G \cup B)^* .
\]

6. Consider the language \( L = 0^* 1^* \cap ((0 \cup 1)(0 \cup 1))^* \).

6.1. Why is the above expression for \( L \) not a regular expression?

**Solution:** Because it contains an intersection operator \( \cap \), whereas regular expressions may only contain union, concatenation and Kleene star operators.

6.2. Write a regular expression for \( L \).

**Solution:** Observe that \( L \) consists of all even-length strings of the form \( 0^i 1^j \), where \( i \) and \( j \) are non-negative integers. In such a string, either \( i \) and \( j \) are both even or they are both odd. Therefore,
\[
L = (00)^* (11)^* \cup 0(00)^* 1(11)^* .
\]
7. For a string \( x \in \{0, 1\}^* \), let \( N_0(x) \) and \( N_1(x) \) denote the number of 0’s and 1’s (respectively) in \( x \). Of the following two languages, at least one is regular. Prove this.

\[
\begin{align*}
L_1 &= \{ x \in \{0, 1\}^* : |N_0(x) - N_1(x)| = 5 \}, \\
L_2 &= \{ x \in \{0, 1\}^* : |N_0(x) - N_1(x)| \text{ is divisible by 5} \}.
\end{align*}
\]

Obviously you should start by picking one of the two languages. Remember, you don’t need to prove anything about the other language.

[10 points]

Solution: The language \( L_2 \) is regular, because it is recognized by the following DFA.

![Diagram of a DFA]

The language \( L_1 \) is not regular. You did not have to prove this, but you should be able to do so by using the pumping lemma to try to pump the string \( 0^{p+5}1^p \in L_1 \), where \( p \) is the hypothetical pumping length of \( L_1 \).