1. Give a one sentence proof that any NFA can be converted into an equivalent NFA that has no \( \varepsilon \)-transitions. Feel free to use results proved in class.  

**Solution:** As proved in class, any NFA can be converted into an equivalent DFA, which is, of course, automatically an NFA without \( \varepsilon \)-transitions.

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

**Solution:**

```
q_0 \rightarrow a \rightarrow q_1 \rightarrow a \rightarrow q_2 \rightarrow b \rightarrow q_3 \rightarrow \varepsilon \rightarrow q_4
```
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. [10 points]

**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. Write a regular expression for the language $\{x \in \{R, G, B\}^* : x$ has an odd number of $R$'s$\}$. You do not have to give a proof of correctness. [10 points]

**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:

$$ (G \cup B)^* R ((G \cup B)^* R (G \cup B)^* R)^* (G \cup B)^*. $$
5. In class, we went through a proof that a DFA $M = (\{q_1, \ldots, q_n\}, \Sigma, \delta, q_1, F)$ can be converted into an equivalent regular expression. During this proof we defined some sets $R^k_{ij}$ and proved, using induction, that they were all regular.

5.1. Define the sets $R^k_{ij}$. An informal definition will suffice.

**Solution:** $R^k_{ij}$ is the set of all strings in $\Sigma^*$ that take $M$ from state $q_i$ to state $q_j$ without ever passing through a state numbered higher than $q_k$. Here, “passing through” means entering and then leaving.

5.2. Write down a system of equations which inductively, and completely, specifies every set $R^k_{ij}$. Make sure your equations cover all the base cases and all the induction steps. You do not need to prove that your equations are correct.

**Solution:** Here are the desired equations:

\[ R^0_{ij} = \{ a \in \Sigma : \delta(q_i, a) = q_j \}, \quad \text{for } i, j \in \{1, \ldots, n\} \text{ with } i \neq j, \]
\[ R^0_{ii} = \{ \varepsilon \} \cup \{ a \in \Sigma : \delta(q_i, a) = q_i \}, \quad \text{for } i \in \{1, \ldots, n\}, \]
\[ R^k_{ij} = R^{k-1}_{ij} \cup R^{k-1}_{ik} \left( R^{k-1}_{kk} \right)^* R^{k-1}_{kj}, \quad \text{for } i, j, k \in \{1, \ldots, n\}. \]
6. 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, exactly one is regular. Specify which language is regular and prove that it is so. You do not have to give a proof that the other language is not regular.

$$L_1 = \{ x \in \{0, 1\}^* : |N_0(x) - N_1(x)| = 5 \},$$
$$L_2 = \{ x \in \{0, 1\}^* : |N_0(x) - N_1(x)| \text{ is divisible by 5} \}.$$

[15 points]

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

![DFA Diagram]

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 $L_1$. 