Problem 1: True or False?

1. If \( L_1 \) and \( L_2 \) are regular, then \( L_1 \cup L_2 \) is regular. \textbf{True}

2. If \( L_1 \) and \( L_2 \) are non-regular, then \( L_1 \cap L_2 \) is non-regular.
   \textbf{False}. Consider \( L_1 = \{0^n1^n : n \geq 0\} \) and \( L_2 = \{0^n1^n1^n : n \geq 0\} \).

3. If \( L_1 \) is regular and \( L_2 \) is non-regular, then \( L_1 \cup L_2 \) is non-regular.
   \textbf{False}. Consider \( L_1 = \Sigma^* \) and \( L_2 \) any non-regular language.

4. If \( L_1 \) is regular, \( L_2 \) is non-regular, and \( L_1 \cap L_2 \) is regular, than \( L_1 \cup L_2 \) is non-regular.
   \textbf{True}. Write \( L_2 = \{(L_1 \cup L_2) - L_1\} \cup (L_1 \cap L_2) \).

5. The following language is regular: The set of strings in \( \{0, 1\}^* \) having the property that the number of 0’s and the number of 1’s differ by no more than 2.
   \textbf{False}.

6. The following language is regular: The set of strings in \( \{0, 1\}^* \) having the property that in every prefix, the number of 0’s and the number of 1’s differ by no more than 2.
   \textbf{True}. A simple 5-state DFA accepts this language.

Problem 2: Regular Expressions. Write regular expressions for the following languages. The alphabet is \( \{0, 1\}^* \).

1. \( A_1 = \{ w | w \text{ contains at least two 0's}\} \).
   \textbf{Solution}: \( (0 \cup 1)^*0(0 \cup 1)^*0(0 \cup 1)^* \).

2. \( A_2 = \{ w | w \text{ contains an even number of 0's}\} \).
   \textbf{Solution}: \( 1^*(01^*1)^* \).

Problem 3: Proving non-regularity: the Pumping Lemma. Prove that the following languages are not regular.

1. \( L_4 = \{ i^j2^k | i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k \} \).
   \textbf{Solution}: Define \( L_4' = \{ i^j2^{k'} | j, k \geq 0 \} \cup \{ 0^i1^j2^k | i > 1, j, k \geq 0 \} \). It is easily seen that \( L_4' \) is regular. Now, observe that \( L_4 - L_4' = \{ 0^i1^j2^k | j \geq 0 \} \) is not regular.

Problem 4: The size of the minimal DFA for a regular language \( L \). Consider the regular language \( L = \{ w | w \text{ contains at least three 1's}\} \). Prove that any DFA for this language has at least 4 states.

\textbf{Solution}: The crucial fact to use is that, if strings \( x \) and \( y \) lead from the start state to the same state \( q \), then for every string \( z \), \( xz \in L \) if and only if \( yz \in L \). More formally, \( \delta^*(q_0, x) = \delta^*(q_0, y) \) implies \( \forall z \in \Sigma^*, xz \in L \) if and only if \( yz \in L \). (Think about it and convince yourself that this is true).

Now, note that strings \( \epsilon, 111, 111 \) must lead to different states. For instance, suppose \( \delta(q_0, \epsilon) = \delta(q_0, 1) \). Then, setting \( z = 11 \), we see that 11 \( \not\in L \) whereas 111 \( \in L \). This is a contradiction, and therefore the strings \( \epsilon \) and 1 lead to different states.