1. Consider the following DFA.

(a) Give a one-sentence description of the language accepted by this DFA.

(b) Give another DFA that is equivalent to the one above but using as few states as possible.

(c) Justify that your DFA accepts the same language as the one above. (You do not have to write a formal proof.)

2. Give an NFA and a RE for each of the following languages. Give a brief justification that your answers are correct, i.e., explain why your NFA accepts every string in the language but no other, and why your RE describes every string in the language but no other — no formal proof required — just brief English explanations.

(a) \( L_1 = \{ s \in \{a, b, d\}^* : s \text{ contains exactly one occurrence of the substring "dab" and no occurrence of the substring "bad"} \} \)

(b) \( L_2 = \{ s \in \{0, 1, 2\}^* : s \text{ contains some pair of 2's separated by exactly five characters other than a 2} \} \)

(c) \( L_3 = \{ s \in \{0, 1, 2\}^* : \text{the integer value of } s \text{ (in ternary notation) is one less than a multiple of 4} \} \)

(For example, \( 21 \in L_3 \) because the integer value of “21” in base three is \( 2 \cdot 3 + 1 \cdot 1 = 7 = 8 - 1 \).)

3. For every language \( L \subseteq \Sigma^* \), define \( \text{Init}(L) = \{ x \in \Sigma^* : \exists y \in \Sigma^*, xy \in L \} \). (Intuitively, \( \text{Init}(L) \) is the set of all strings that can be “completed” to become a string in \( L \).) For example, \( \text{Init} \{a, ab, bab\} = \{ \varepsilon, a, ab, b, ba, bab\} \).

Prove that the class of regular languages is closed under the \( \text{Init} \) operation, i.e., that for every regular expression \( R \), there exists a regular expression \( R_I \) such that \( L(R_I) = \text{Init}(L(R)) \).