1. Using the alphabet $\Sigma_1 = \{a, b\}$, design NFAs for each of the following languages:

   (a) $L_1 = \{w \mid 1 \leq |w| \leq 3\}$.

   (b) $L_2 = \{w \mid \text{every } b \text{ in } w \text{ is followed by } aa \}$.

   (c) $L_3 = L_1 \cdot L_2$, using the concatenation construction discussed in class.

2. Using the same alphabet, give a regular expression for each of the following languages:

   (a) $L_4 = \{w \mid w \text{ starts and ends with the same symbol}\}$.

   (b) $L_5 = \{w \mid w \text{ has at least three } a\}$.

   (b) $L_6 = \{w \mid w \text{ starts with } baa\}$.

3. Let’s practice translating between NFAs, DFAs, and regular expressions:

   (a) Design a NFA to recognize the language $(aa \cup aba)^*$.\(^1\)

   (b) Construct an equivalent DFA using the construction from Theorem 1.39. You can leave out the states of the DFA that are unreachable from the start state.

   (c) Now convert the DFA back to a regular expression using Lemma 1.60.

4. **Extra Credit:** Problem 1.45 in the textbook.

---

\(^1\)You’ll want to make this NFA as small as you can. Three states are enough.