1. Automata:
   (a) State, alphabet, transition, starting state, and accepting states.
   (b) DFA: moves to a unique state $y = \delta(x, a)$ when at state $x$, and input is $a$.
   (c) NFA: moves to a set of possibly empty states, plus $\epsilon$-transitions.
   (d) Simulating DFAs/NFAs: follow the arrows, maintain (set) of current states.
   (e) Equivalent DFA of an NFA.

2. Operations on languages, regular operations
   (a) DFA for the complement of a regular language: set all accepting states to non-accepting, and all non-accepting states to accepting.
   (b) Regular Languages: languages accepted by some DFA.
   (c) Regular operations on languages: union / concatenation / Kleen $\ast$.
   (d) Explicit construction of a DFA accepting the union of two DFA.
   (e) Regular expression: start with ‘base languages’ $a \in \Sigma$, $\epsilon$, or $\emptyset$, then recursively combine them using regular operations.
   (f) Constructing NFAs for the results of applying each of the regular operations to regular languages.
   (g) Construction of DFAs/NFAs for languages corresponding to regular expressions.

3. Pumping Lemma
   (a) If $L$ is regular,
      i. there is pumping length $p$,
      ii. such that for all $w \in L$ with $|w| \geq p$, 
      iii. there is split of $w = xyz$ with $|xy| \leq p$ and $|y| > 0$ such that
       iv. for all $i \geq 0$, $xy^iz \in L$.
   (b) Proof of pumping lemma.
   (c) To show that $L$ is not regular, it suffices to show:
      i. for any pumping length $p$,
ii. we can exhibit a string \( w \in L \) (based on \( p \), designed to make all splits (next step) simple)

iii. such that for any split \( w = xyz \) with with \(|xy| \leq p\) and \(|y| > 0\),

iv. we can exhibit some \( i \geq 0 \) such that \( xy^i z \not\in L \).

Additional problems:

1. For a fixed \( k \), let

\[
L_k = \{ w \in \{0,1\}^* \mid \text{the } k^{th} \text{ symbol from the right is } 1 \}.
\]

Design a DFA for \( L_k \).

2. An all-NFA \( M \) is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\) that accepts a string \( x \in \Sigma^* \) if every possible state that \( M \) could be in after reading input \( x \) is a state from \( F \).

Note, in contrast, that an ordinary NFA accepts a string if some state among these possible states is an accept state.

Given an all-NFA \( M \), show how to construct a DFA \( D \) such that \( L(D) = L(M) \).

3. Let \( A \) be a regular language. Show that the following language is also regular:

\[
\{ x \mid \text{there exists } y \text{ such that } |y| = 2|x| \text{ and } xy \in A \}.
\]