1. Read the notes above carefully.

2. For each of the following languages, construct a finite automaton (deterministic, nondeterministic, or nondeterministic finite automaton with $\epsilon$-transitions) that accepts the language. Give the key idea(s) for your construction, and brief and precise interpretations for the states of the machine.

   (a) \[
   \{ x \in \{0,1\}^* \mid \#_0(x) = \#_1(x) \text{ and every prefix of } x \text{ has at most one more 0 than 1s and at most one more 1 than 0s } \}.
   \]
   (Note: $\#_u(v)$ denotes the number of occurrences of a substring $u$ in a string $v$.)

   (b) \[
   \{ x \in \{0,1\}^* \mid \text{there exist two 0s in } x \text{ that are separated by a string of length 5k for some } k \geq 0 \}.
   \]

   (c) The set of all strings over the alphabet \{a, b, c\} that yield the same value when evaluated from left to right as right to left by “multiplying” according to the following table in Figure 1.

   For examples: \((a \circ b) \circ b = (c \circ b) = a\) and \((a \circ (b \circ b)) = (a \circ a) = a\), whereas \(((a \circ b) \circ c) = (c \circ c) = b\) and \((a \circ (b \circ c)) = (a \circ c) = c.\)

   \[
   \begin{array}{c|ccc}
   \circ & a & b & c \\
   \hline
   a & a & c & c \\
   b & b & a & c \\
   c & c & a & b \\
   \end{array}
   \]

   Figure 1: A non-associative multiplication table for $\circ$.

3. Convert the following nondeterministic finite automaton with $\epsilon$-transitions, $M$, to an equivalent nondeterministic finite automaton $M_1$, and then using the Subset Construction to convert $M_1$ to an equivalent deterministic finite automaton $M_2$ with its inaccessible states removed. Explicitly and briefly write down each step which you perform, such as: (1) Computing all the $\epsilon$-closures of the states of $M$, and showing complete state-transition diagrams of $M_1$ and $M_2$. 


4. Do [Sip06]/[Sip13] Do Chapter 1, problems 1.33 and 1.41 by machine construction (deterministic, nondeterministic, or nondeterministic finite automaton with \( \epsilon \)-transitions). Give the key idea(s) for your construction, and brief and precise interpretations for the states of each machine.


6. Let \( \Sigma = \{0, 1\} \). Give a regular expression for each of the the following languages. Briefly and precisely annotate your answers.

   (a) All strings in \( \Sigma^* \) with at most three 0s.
   (b) All strings in \( \Sigma^* \) with a number of 0s divisible by three.
   (c) All strings in \( \Sigma^* \) with exactly one occurrence of the substring 000.

7. Recall that for two regular expressions (over an alphabet) \( r \) and \( s \), \( r = s \) means that \( L(r) = L(s) \). Prove or disprove the following for regular expressions \( r \) and \( s \) over an alphabet \( \Sigma \).

   (a) \((rs + r)^*r = r(sr + r)^*\)
   (b) \((r^*s)^* = (r + s)^*\)

8. Do [Sip06]/[Sip13] Do Chapter 1, exercise 1.21 (a). Notes: You may use the procedure presented in lecture. Show all the (important) intermediate work.

9. A finite automaton is called non-crossing if its state-transition diagram can be drawn in the plane without having any edge cross.

   (a) Prove that every regular language is accepted by a non-crossing finite automaton.
   (b) Construct a regular language that is not accepted by any non-crossing deterministic finite automaton.

10. Let \( L \) be a language over an alphabet \( \Sigma \) such that \( L \neq \emptyset, L \neq \{\epsilon\} \), and \( L^2 = L \). For each of the following parts, prove or disprove that:

    (a) \( \epsilon \in L \);
    (b) \( L \) is not a finite language (that is, \(|L| \) is not finite).