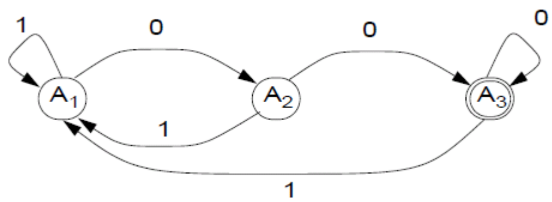
Theory of Computation

Spring 2025, Homework # 1

Due: March 25, 2025

1. (10 pts) A string u is an *anagram* of a string w if u is obtained from w by rearranging the symbols. For example, *listen* is an anagram of *silent*. Formally, if $w = w_1 w_2 \cdots w_n$, then u is its anagram if $u = w_{\sigma(1)} w_{\sigma(2)} \cdots w_{\sigma(n)}$ for some permutation σ. Given a regular language L, is $L_A = \{u | u \text{ is an anagram of } w \in L\}$ always regular? Justify your answer.

2. (20 pts) Consider the following DFA,



we associate the DFA with the following system of equations:

$$A_1 = 1A_1 + 0A_2$$
$$A_2 = 0A_3 + 1A_1$$
$$A_3 = 0A_3 + 1A_1 + \epsilon$$

The solution of A_i ($1 \leq i \leq 3$) corresponds to set of strings that can lead the DFA from state A_i to accepting state A_3. Solve the system of equations to obtain the regular expression corresponding to the solution of A_1, which is the language accepted by the DFA as A_1 is the initial state.
Hint: you may use the fact that the solution of $X = \alpha X + \beta$ is $\alpha^* \beta$. For instance, $A_3 = 0A_3 + 1A_1 + \epsilon$ implies $A_3 = 0^*(1A_1 + \epsilon)$. Also use the idea similar to Gaussian elimination in solving systems of linear equations.

3. (10 pts) Use the pumping lemma to prove that $L = \{0^{n!} | n \geq 0\}$ is not regular, where $n!$ denotes the factorial of n.

4. (15 pts) Let $\Sigma = \{0, 1\}$. For each of the following languages, decide whether it is regular or not. Justify your answers.
   (a) $L_1 = \{1^k y \mid y \text{ contains at least } k\ 1's, k \geq 1.\}$
   (b) $L_2 = \{1^k 0 y \mid y \text{ contains at least } k\ 1's, k \geq 1.\}$.
   (c) $L_3 = \{1^k y \mid y \text{ contains at most } k\ 1's, k \geq 1.\}$

5. (15 pts) If L is a language over the alphabet $\Sigma = \{0, 1\}$, define L_{ERROR} so that a string is in L_{ERROR} iff it is the result of flipping a bit in a string in L; i.e.,

$$L_{ERROR} = \{w \mid w = uxv, u, v \in \Sigma^*, x \in \{0, 1\},\ u\overline{x}v \in L\}$$

where $\overline{0} = 1, \overline{1} = 0$.
Prove that if L is regular, then L_{ERROR} is also regular.
Hint: Suppose M is a DFA accepting L, construct an NFA M' so that $L(M') = L_{ERROR}$.

6. (20 pts) For a language L over an alphabet Σ, define

$$CYCLE(L) = \{x_1x_2 \mid \exists x_1, x_2 \in \Sigma^* \text{ such that } x_2x_1 \in L\}.$$

For example, if $abc \in L$, abc, bca, $cab \in CYCLE(L)$. Prove that if L is regular, then so is $CYCLE(L)$.
Hint: Given an FA $M = (Q, \Sigma, \delta, q_0, F)$ accepting L, construct an NFA $M' = (Q', \Sigma, \delta', q'_0, F')$ to accept $CYCLE(L)$. You may let a state of Q' be of the form (p, q, d), $d \in \{1, 2\}$. The d is used to indicate whether the current input symbol is a part of x_1 or x_2.

7. (10 pts) Consider a new kind of finite automata called $\forall$-NFA. A $\forall$-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that recognizes x if *every* possible computation of M on x ends in a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* computation ends in an accept state. Prove that $\forall$-NFA recognize the class of regular languages.