COMP 330 Fall 2023
Assignment 2
Due date: 5^{th} Oct 2023

Posted on September 19^{th} 2023

This is the second of 6 assignments. There are **5** questions for credit. The homework is due on `CrowdMark` **at 11:59pm**.

Important: Solutions have to be in **pdf** format. Every question must be on a **separate** pdf and submitted through Crowdmark. We prefer solutions prepared on LaTeX but that is **not required**. We will **not** accept jpg photos of handwritten solutions. We will not accept Word documents. Either of these formats can be exported to pdf. **TAs will remove marks for blurry/illegible submissions.**

Question 1. [20 points] Recall from Lectures 3 and 5 the *extended transition functions* δ^* and Δ^* of DFA and NFA respectively. Prove the following facts (each worth 10 points) about those functions using *mathematical induction*.

1. Given a DFA $M = (Q, \Sigma, \delta, s_0, F)$ and strings $x, y \in \Sigma^*$, $\delta^*(q, xy) = \delta^*(\delta^*(q, x), y)$.
2. Given an NFA $N = (Q, \Sigma, \Delta, S_0, F)$, subsets $A \subseteq Q, B \subseteq Q$ and a string $x \in \Sigma^*$, $\Delta^*(A \cup B, x) = \Delta^*(A, x) \cup \Delta^*(B, x)$.

Question 2. [10 points] Let Σ be an alphabet and let $L \subseteq \Sigma^*$ be a *finite* language. Prove that L is regular by

1. [5 points] Constructing an NFA or an NFA+ε which accepts L.
2. [5 points] Proving the correctness of your machine. Note: Argue directly, do not use induction.

Question 3. [30 points] Let Σ be an alphabet and let $L_1, L_2 \subseteq \Sigma^*$ be *regular* languages. Prove each of the following claims by constructing an FA (DFA, NFA, or NFA+ε). *You do not need to prove the correctness of your construction.*

1. [15 points] **Claim:** The language $\texttt{merge}(L_1, L_2) = \{x_1 y_2 : \exists x_1, x_2, y_1, y_2 \in \Sigma^* \text{ s.t. } x_1 x_2 \in L_1 \ \& \ y_1 y_2 \in L_2\}$ is regular.

   Hint: Use the following state operator in your construction. For a DFA $M = (Q, \Sigma, \delta, q_0, F)$, $\texttt{reachable}(q) = \{p \in Q : \text{ there is a walk from } q \text{ to } p\ \}$.

2. [15 points] Given some alphabet Σ, we define the dist operator between two strings $x, y \in \Sigma^*$ as the number of positions in which the strings x and y differ. That is,

$$\texttt{dist}(x, y) = \begin{cases} n & \text{if } |x| = |y| \ \ \& \ \ x, y \text{ differ in exactly } n \text{ positions} \\ \infty & \text{if } |x| \neq |y| \end{cases}$$

For instance $\texttt{dist}(aa, baa) = \infty, \texttt{dist}(aab, baa) = 2, \texttt{dist} = (aba, bab) = 3$.

Claim: The language $\texttt{dist2}(L_1) = \{w \in \Sigma^* : \exists x \in L_1 \text{ s.t. } \texttt{dist}(x, w) \leq 2\}$ is regular.

Question 4. [20 points] Provide **regular expressions** over the alphabet $\Sigma = \{a, b\}$ which describe each of the following languages (each worth 5 points). You should not need to use a conversion algorithm. A proof of correctness is not required.

1. $\{w \in \Sigma^* : n_a(w) \neq 1\}$ Note: We use the notation $n_a(w)$ to denote the number of letters $a \in \Sigma$ in the string w.

2. $\{w \in \Sigma^* :$ Every letter of w at an even position is an $a\}$ Note: In the string ab the position of a is 1 and the position of b is 2.

3. $\{vwv : v, w \in \Sigma^*, |v| = 1, |w| \geq 1\}$

4. $\{w \in \Sigma^* : w$ starts with ab or ends with $ba\}$

Question 5. [20 points] Consider the following language over the alphabet $\Sigma = \{a, b\}$

$$L = \{w : \ n_b(w) \text{ is even, } n_a(w) \text{ is odd, and } w \text{ does not contain the substring } ba\}$$

Answer each of the following questions.

1. [10 points] Design the **<u>minimal</u> deterministic** finite automata which recognizes L. You do not need to prove correctness *nor* minimality.

2. [10 points] Provide regular expressions for each of the equivalence classes of $\equiv_L$. [Hint: Use your answer from 1. A conversion algorithm is not necessary.]