COMP 330 Fall 2023
Assignment 3
Due date: 19^{th} Oct 2023

Posted on October 5^{th} 2023

This is the third 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. [15 points] Provide algorithms which solve the following decision problems. Justify the correctness of your algorithms using facts shown in class.

1. [5 points] Given a DFA M and an integer n, is the index of $\equiv_{L(M)}$ equal to n?
2. [10 points] Given two regular expressions r_1, r_2 over a non-empty alphabet Σ, is $L(r_1) \subseteq L(r_2)$?

Question 2. [15 points] Prove, **using the Myhill-Nerode theorem**, that the following language over the alphabet $\Sigma = \{a, b, c\}$ is not regular. **Note: You will receive no marks if you use the pumping lemma and/or closure properties.**

$$L = \{a^i b^j c^k : i, j, k \in \mathbb{N}, 0 \leq i < j < k\}$$

Question 3. [30 points] Prove that the following languages are not regular using the Pumping lemma and/or closure properties. **You will receive no marks if you use the Myhill-Nerode theorem.**

1. [15 points] $L_1 = \{w \in \{a, b\}^* : n_a(w) + n_b(w) = 3 \cdot n_b(w)\}$ Note: We use $n_a(w)$ to denote the number of a's in the string w.
2. [15 points] $L_2 = \{a^n b^m c^k : n, m, k \in \mathbb{N}, n < m \text{ or } n < k\}$

Question 4. [20 points] Consider the following languages over the alphabet $\Sigma = \{a, b\}$

1. $L_1 = \{xwx : x, w \in \Sigma^*\}$
2. $L_2 = \{xwx : x, w \in \Sigma^*, |w| = 1\}$
3. $L_3 = \{xwx : x, w \in \Sigma^*, |x| = 1\}$

For each language, decide whether it is regular or not regular. Prove each of your claims (using any method/fact seen in class). If you construct a FA or a regular expression, you do not need to prove the correctness of your construction.

Question 5. [20 points] Suppose L_1, L_2 are two regular languages over the alphabet $\Sigma = \{a, b\}$. Consider the following two new languages:

1. `shuffle1`$(L_1, L_2) = \{a_1 b_1 \ldots a_n b_n : a_1 \ldots a_n \in L_1, b_1 \ldots b_n \in L_2, a_i, b_i \in \Sigma, a_i = b_i, i \in [1 \ldots n]\}$
2. `shuffle2`$(L_1, L_2) = \{a_1 b_1 \ldots a_n b_n : a_1 \ldots a_n \in L_1, b_1 \ldots b_n \in L_2, a_i, b_i \in \Sigma, a_{n-(i-1)} = b_i, i \in [1 \ldots n]\}$

Notice how `shuffle1`, `shuffle2` are similar, but not quite the same, as the original `shuffle` shown in Lecture 6. One of (1) and (2) is *necessarily* regular and the other is *not necessarily* regular. Which is which? Prove your answer. If part of your proof relies on a construction, you do not need to prove its correctness.