CS 181: Formal Languages and Automata Theory **Spring 2021**

Homework 2

Assigned: Tue 4/6 *Due: Mon 4/12, 6:00pm PDT*

Problem 1

Let $\Sigma = \{a, b, c\}$. Show an NFA which recognizes the following language over Σ:

$$L_1 = \{w \in \Sigma^+ \mid w \text{ contains at least one substring consisting of two of the same symbol separated by at least three occurrences of the other two symbols}\}$$

The substring between the "two of the same symbol" can consist of one or both of the "other two symbols" but cannot contain the "same symbol".
E.g., L_1 contains: $abbcbba, caaaaaacb, bbbabcbcbaaaaa, babcbcbcaaaaaca, abbbcbababac$. L_1 does not contain: $abbaba, caacaab, bbbbcbcaaaaa$.

Specify the NFA as a state diagram. It does not have to be fully specified, but you have to use the blocking convention correctly. *Be sure to clearly indicate your initial state and accepting state(s).* You may find it helpful to use the shorthand discussed in class of putting more than one symbol on an edge of the transition diagram. Part of your score will be based on demonstrating effective use of nondeterminism in the NFA model.

Briefly describe how your design works.

Problem 2

Let $\Sigma = \{a, b\}$. Use the Pumping Lemma for the FSLs to show that the following language over Sigma is not a FSL:

$$L_2 = \{a^{(2n)}b^n \mid n \geq 0\}$$

Problem 3

Show that $L_3 = \{w \in \Sigma^* \mid \#(a, w) = 2\#(b, w)\}$ is not FSL using the technique discussed in class of proof by contradiction using the closure properties of the family of FSLs. You may find some of the other languages mentioned in this homework set useful for solving this problem.

Problem 4

Let $\Sigma = \{0, 1, \#\}$. Give a regular expression for the following language:

$$L_4 = \{w \in \Sigma^* \mid \text{in } w, \text{ to the right of every 0 there is at least one 1 before any \#'s}\}$$

E.g., L_4 contains: $1011\#1, 000, 01$. L_4 does not contain: $\#00\#11, 10101010\#$.
Briefly explain how your regular expression is designed to correctly represent this language.

Problem 5

Recall from lecture that a directed rooted tree is a connected, directed graph which is acyclic, even when viewed as an undirected graph, where one node is designated as the root and all edges in the tree are directed away from the root towards the leaves.

The height of a directed rooted tree is defined as the length of a longest path from the root to a leaf. The height can be expressed in terms of the number of edges on a longest path or in terms of the number of nodes on a longest path. I.e., "h edges" and "$h+1$ nodes" are two different ways of expressing the same tree height.

We define a k-ary directed rooted tree to be a directed rooted tree in which each node has at most k children.

Let T be a k-ary directed rooted tree of height h edges ($h+1$ nodes), where $h \geq 2$ and $k > 1$. Show by induction on h that:

If T has $(k^h) + 1$ leaf nodes, then T must have height at least h edges.