COT 5310 Theory of Automata and Formal Languages January 29, 2024

Problem Set 2
Due February 8, 2024

1. (10 pts) The following problems relate to regular languages. You can use theorems proved in the text related to the problems.
(a) Let L, L' be regular languages. Show that L – L' is a regular language.
(b) Let L, L' be languages such that L is regular, L ∪ L' is regular, and L ∩ L' = ∅. Prove that L' is regular.
(c) Suppose that we know that L ∪ L' is regular and that L is finite. Can we conclude that L' is regular? Explain.

2. (10 pts) Let A = {*a*, *b*} be the alphabet of an f.s.a. Let *x* be an input string over A. The number of *a*'s in the input string *x* is defined to be $N_a(x)$. Similarly, $N_b(x)$ is the number of *b*'s in the input string *x*. Construct an f.s.a that for any input string *x*, accepts only those strings such that $N_a(x) \bmod 5$ is equal to 2 and $N_b(x) \bmod 3$ is equal to 1. If such a d.f.a. does not exist, explain why.

3. (10 pts) First construct an n.f.a with the fewest states you can that accepts the set:
$\{bab^na : n \geq 0\} \cup \{ba^nb : n \geq 0\}$.
Next convert the n.f.a. you constructed to a d.f.a.

4. (10 pts) Show that the following languages L are not regular using the pumping lemma for regular languages.
(a) $L = \{a^*b^sc^t \mid 0 < t \leq s\}$. For this problem, suppose for an *n* being the number of states of the reference f.s.a, you choose $w = a^nb^nc^n$. Is this a valid string to choose. Will this work for proving your assertion? Explain.
(b) $L = \{a^p \mid p \text{ is a prime number}\}$.

5. (10 pts) Which of the following languages are regular languages. Justify your answer.
(a) $\{ (0110)^{2n} : n \geq 0 \}$
(b) $\{ a^nb^mc^k : n,m,k \geq 0, n + m = k \}$
(c) $\{ x x^R : x \in \{a,b,c\}^* \}$ (note that that x^R denotes the *reverse* of the string x.)
(d) $\{w x w^R : w, x \in \{a,b\}^+ \}$
(e) $\{ a^{n\wedge 2} : n \geq 1 \}$

6. The following problem will not be graded but will be covered in the solution set. However, it is important and useful to try to do the problem.
Construct a d.f.a. that accepts strings over {*0, 1*} if and only if the value of the string interpreted as a binary integer is equivalent to 3 modulo *7*. (Note that the binary input string *100* is the integer *4* and the binary inputs are accepted left to right.)