Assignment 5 – CS 301 - Theory of Automata – Fall 2020

Due: Thursday, December 3, 2020 (online – 8 AM)

If any assignment is deemed to be copied from any other student or internet, you may be awarded an F in this course.

1. [50 = 10 * 5]

   For the following grammars, determine whether the language is a CFL or non-CFL. If the language is CFL, you are required to give a CFG. Otherwise, use pumping lemma to show that the language is a non-CFL language.

   a. A = {$a^n ww^R a^n$ | n > 0 and w ∈ {a, b}* } Note that w^R stands for reverse of word w
   b. B = {w | w ∈ {a, b}* and w is not a palindrome}
   c. C = {w | w ∈ {a, b, c}* where na(w) = 2nb(w) and n(w) = na(w) + nb(w)} Note: na(w) stands for number of a's in w.
   d. D = {$a^n b^m c^k$ | n >= m and k = n – m}
   e. E = { $a^{2n} b^{2m} c^n d^m$: n, m>=0}
   f. F = {$a^n b^n c^k$ | n <= k <= 2n }

Part a and b from http://www.cs.williams.edu/~tom/courses/361/notes/Lect17.pdf

Part c from final fall 2018

Part d from …

Part e from lums

Part f from lums HW8 SOL.pdf

2. [15] https://courses.engr.illinois.edu/cs373/fa2009/hws/hw_04.pdf
   In class we have seen that $a^n b^n c^n$ is not a CFL as we have just one stack and 2 comparisons cannot be done with one stack only. Assume that we have a special kind of PDA which has two stacks. Prove that a PDA with two stacks will be able to accept some non-CFL languages as well using $a^n b^n c^n$ as an example. Write down in steps how would the new PDA accept this language.

   List another example that can be accepted by 2-stack PDA but not by one-stack PDA

   How about PDAs with queue? Can it also accept our example language? Show complete working

3. [10]
   In class, we have seen CNF form of CFG, according to which all rules must be of the following form
   S → BC
   S → a

Assume new-CNF of the following format
S→ABC
S→a

All other CNF rules are still valid. For example, start variable symbol cannot appear on RHS of the rule.

Will it be possible to find grammar for every CFL?

Counter example A->aa is a CFG but cannot be converted into the desired form. Infact any string with length two cannot be converted into the form given in question.

4. Given two NPDA's, is it possible to verify whether both NPDA's accept the same language. If yes then how? If no then why?
Hint: consider how would someone check if two sets are equal or not. Also, consider the properties of CFL to do this question.