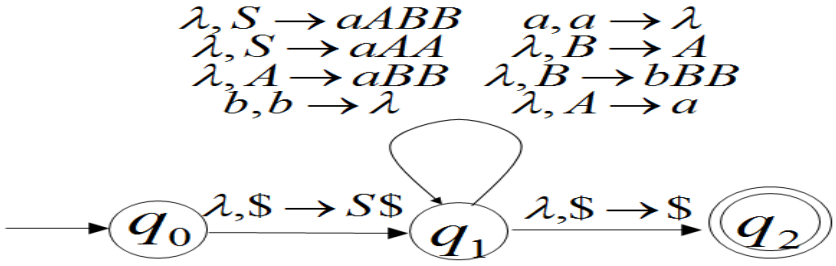
CONCORDIA UNIVERSITY
Dept. of Computer Science and Software Engineering
COMP 335 - Introduction to Theoretical Computer Science
Fall 2017
Solution to Assignment 4

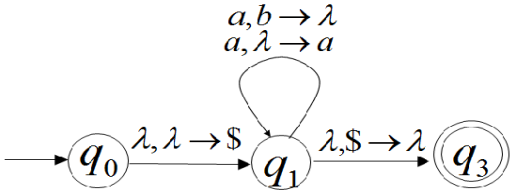
1. The NPDA that accepts the language generated by the following grammar:

   S → aABB | aAA
   A → aBB | a
   B → bBB | A

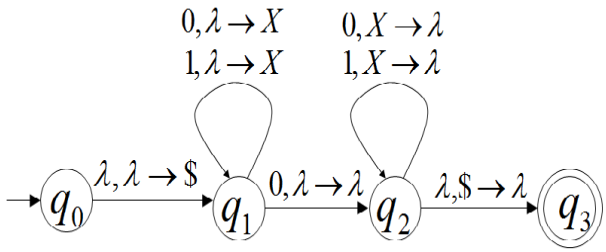


2. The accepted by PDAs

   a) L={λ}

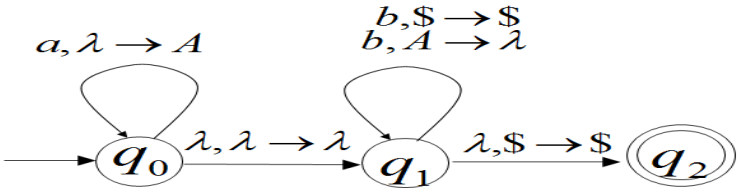


   b) $L = \{x0y \in \{0,1\}^* : |x| = |y|\ \}$

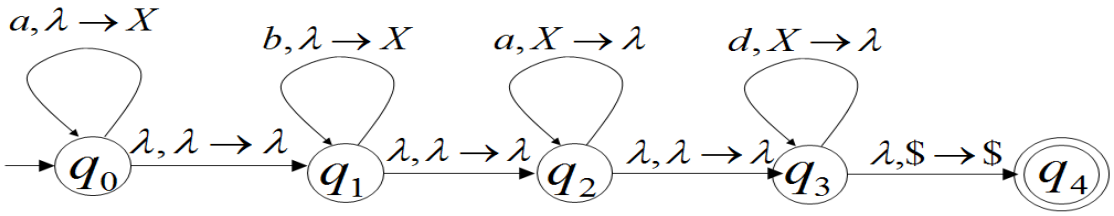


3. The PDAs are

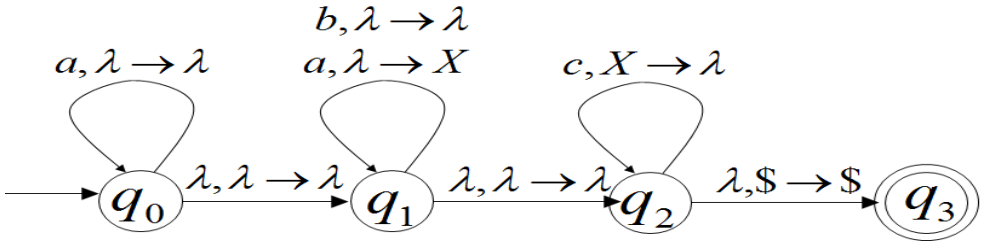
   a)



b)

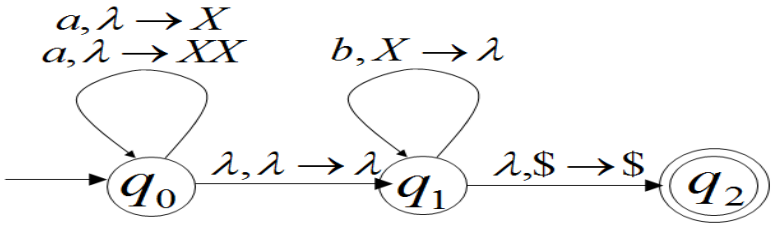


c)

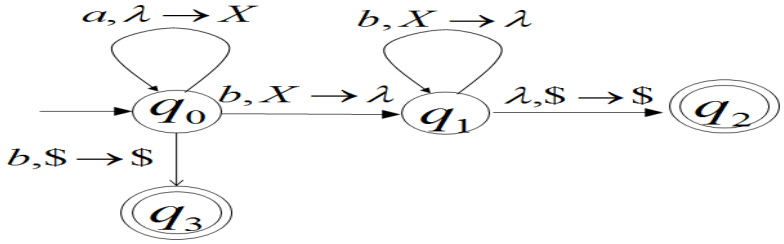


4.

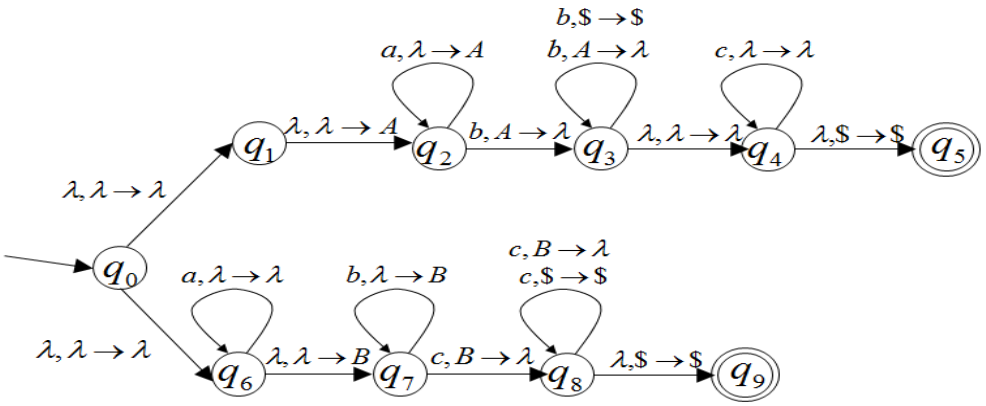
a) This is a **non-determinitic** CFL as there is a NDPA that accepts it; to get n=m, we need to push X each time a is read. To get 2n=m, we need to push XX for every time a is read; to get n<m<2n, we need to push X for some of the a's read and push XX for the other a's read. This choice to push an X or XX for any a read made it non-determinstic CFL.



b) It is **a deterministic** CFL as it is accepted by the following DPDA.



c) This is a **non deterministic** CFL. The NDPA that accepts it is:



It is clear there are two given choices: upper or lower branch.
To get m<n, the pda needs to count the number of a's read using the stack, then use it to ensure the number of b's is more. For n<p, the pda needs to count the number of b's using the stack, and use it to ensure the number of c's is more than the b's.
To have either m<n or n<p, the pda needs to decide (make a choice) whether to keep track of a's to compare with b'a, or keep track of b's to compare with c's. It is not possible to achieve this in a deterministic way.