 dtrived5@stevens.edu

1. (a) L is the complement of the language $\{a^n b^n | n \geq 0\}$

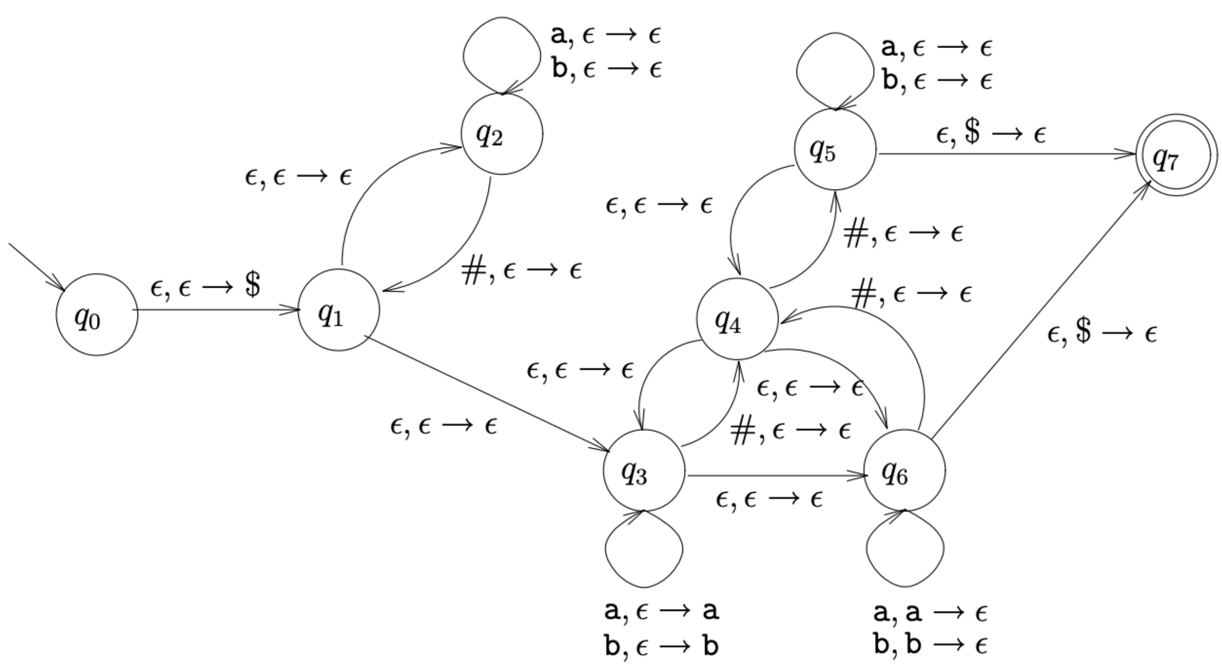
$L = \{a^n b^m : n \neq m\} \cup \{(a \cup b)^* ba(a \cup b)^*\}$

A string in the complement of the language $\{a^n b^n | n \geq 0\}$ must be in one of the following forms:
(i) Contains a substring ba
(ii) Equals to $a^x b^y$ for some $x \neq y$

S -> S1 | S2
S1 -> ba | XS1 | S1X
X -> a | b
S2 -> PQ | QR
P -> a | aP
R -> b | bR
Q -> ab | aQb

(b) A string in the language $\{x_1 \# x_2 \# \ldots \# x_k | k \geq 1,$ each $x_i \in \{a, b\}^*$, and for some i and j, $x_i = x_j^R\}$ must be in one of the following forms:
(i) Contains a palindrome x_i for some i
(ii) Contains distinct i and j such that $x_i = x_j^R$

S -> S1 | S2
S1 -> P | X#P | P#X
P -> aPa | bPb | a | b | e
X -> a | b | # | e
S2 -> Q | X#Q | Q#X
Q -> aQa | bQb | #X#



2. We have a Context free language C and a Regular language R. Since C is given to be a CFL we know that there exists a PDA M1 to recognize C. Since R is given to be a Regular we have a DFA M2 to recognize R. To prove $C \cap R$ is a CFL we demonstrate a PDA M to recognize $C \cap R$. The proof is by construction. We construct M from M1 and M2.

Let M1 recognize C where $M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_1, F_1)$
Let M2 recognize R where $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$
Construct M to recognize $C \cap R$ where $M = (Q, \Sigma, \Gamma, \delta, q, F)$

(i) $Q = \{(r_1, r_2) \,|\, r_1 \in Q_1 \ and \ r_2 \in Q_2\}$
(ii) Σ is assumed the same in M1 and M2
(iii) $\Gamma = \Gamma_1$
(iv) δ is defined as: for each $(r_1, r_2) \in Q$; each $a \in \Sigma$ and each $b \in \Gamma$ let

$$\delta((r_1, r_2), a, b) = (\delta_1(r_1, a, b), \delta_2(r_2, a))$$

(v) $q = (q_1, q_2)$
(vi) $F = \{(r_1, r_2) \,|\, r_1 \in F_1 \ and \ r_2 \in F_2\}$

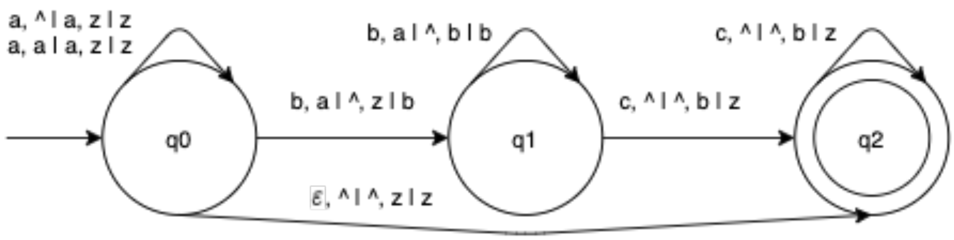
3. To show that language L is not a CFL, we will make the assumption that it is CFL then contradict it. From the above question, under this assumption we are guaranteed that if we intersected some Regular language with L then the resulting language would be a CFL. So if we show that for some Regular language R and some language K which is not a CFL that $L \cap R = K$, then we have derived the contradiction.

To see what this R and K might be consider all these languages, L, K and R as capturing some properties. For L this property is equality of a, b and c. For $K = \{a^n b^n c^n \,|\, n \geq 0\}$ has the property of equality and order. Then R should have the property of order: $R = a^*b^*c^*$ which makes it regular. Since we have a contradiction, it must be that L is not a CFL.

4. (a) A 2 stack PDA to recognize the language $\{a^n b^n c^n : n \geq 0\}$.

1. Use the first stack for checking $a^n b^n$, this can be done by pushing a whenever you see an a and then popping a when you see a b.
2. Use the second stack for checking $b^n c^n$, this can be done by pushing b whenever you see a b and then popping b when you see a c.
3. Accept if both stacks are empty at the end of this process, otherwise do not accept it.

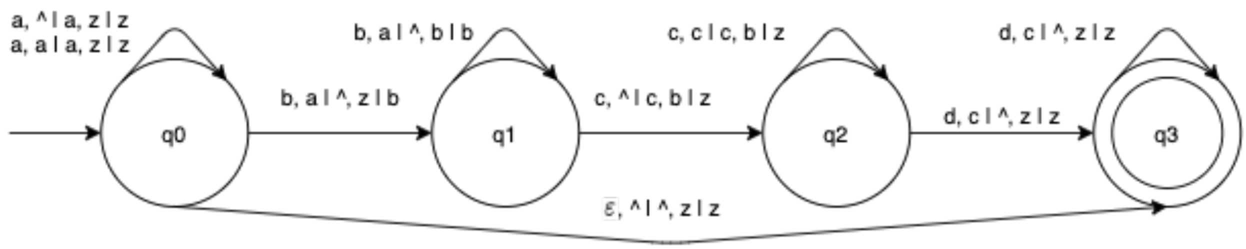
   Let ^ and z be the stack symbol of stack 1 and 2 respectively, then the transition diagram of the above PDA will be as follows:



(b) 2 stack PDA to recognize the language $\{a^n b^n c^n d^n : n \geq 0\}$.

1. Use the first stack for checking $a^n b^n$, this can be done by pushing a whenever you see an a and then popping a when you see a b.
2. Use the second stack for checking $b^n c^n$, this can be done by pushing b whenever you see a b and then popping b when you see a c.
3. Use the first stack for checking $c^n d^n$, this can be done by pushing c whenever you see a c and then popping c when you see a d.
4. Accept if both stacks are empty at the end of this process, otherwise do not accept it.

   Let ^ and z be the stack symbol of stack 1 and 2 respectively, then the transition diagram of the above PDA will be as follows:

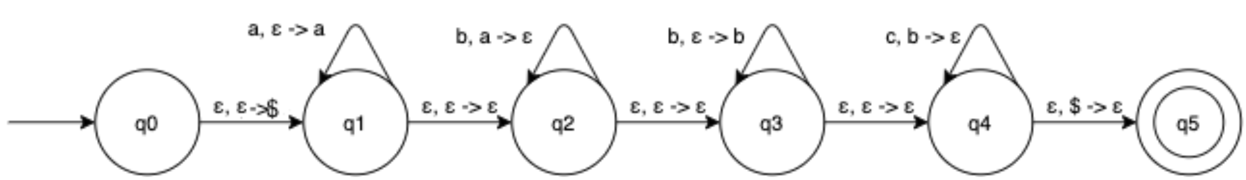


(c) The 2 stack PDA is used for the context sensitive language, the power of 2 stack PDA is equivalent to the power of the Turing machine as the Turing machine uses the infinite size of tape as a memory element which can also be implemented by the 2 stacks, we can pop to read and push to write. We can also simulate the Turing machine by popping from the right stack and pushing to the left to move right, and vice versa. If we hit the bottom of the stack we behave accordingly: that is halt or reject, or stay where you are. Only difference between Turing machine and 2 stack PDA is that is size of the stack is considered to be limited whereas in Turing machine it is unlimited.

5. $L_{add} = \{a^i b^{i+j} c^j \,|\, i, j \geq 0\}$ is a CFL.

$G = (V, \Sigma, R, S)$, $V = \{S, T, U\}$, $\Sigma = \{a, b, c\}$

$$
\begin{aligned}
R = \{ & S \to TU, \\
& T \to aTb \,|\, \epsilon, \\
& U \to bUc \,|\, \epsilon \}
\end{aligned}
$$



$L_{mult} = \{a^i b^{ij} c^j \,|\, i, j \geq 0\}$ is a TM-decidable language.