Birla Institute of Technology and Science, Pilani
First Semester 2020 - 21
Test-3, OPEN BOOK, October 06, 2020
Theory of Computation (CS F351)

Partial Marking is given wherever required.

Q1. [**4 Marks**] True/False. Justify your answer.

"Let P be a PDA with a property that for any input string it cannot put more than two symbols on the stack. If the number of stack symbols are k, then L(P) is regular."

Sol: TRUE. Number of possible configurations with stack are $k^2 + k + 1$. These can be captured by the states of a corresponding NFA.

Q2. [**5 Marks**] Consider the alphabet $\sum = \{a, b, c\}$. Assume that the bottom of stack symbol is Z0 and # is the end marker of the input string. Let $L = \{a^i b^j c^k \mid i > 0, j \geq 0, k \geq 0, \text{and } j=k\}$. Consider the following deterministic PDA M = ({q0, q1, q2, q3, q4, qf}, {a, b, c}, {b, Z0}, Δ, q0, qf}, where Δ is defined as follows (e denotes string of length zero):

((q0, e, e) (q1, Z0))

((q1, a, Z0) (q2, Z0))

((q3, c, b) (q4, e))

((q4, #, Z0) (qf, e))

Few transitions are missing in the given DPDA. Write these missing transitions. You do not require to add any new states. *Also, can the above PDA be deterministic if # is not the end marker of the input string?*

Sol: ((q2, a, e)(q2, e)), ((q2, #, Z0)(qf, e))

((q2, b, e)(q3, b)), ((q3, b, e) (q3, b)), ((q4, c, b)(q4, e))

No, the given PDA will not be deterministic if # is not the end-marker of the input symbol. The branch from q2 to qf will make the PDA non-deterministic.

Q3. [**4 Marks**] Over $\sum = \{a, b\}$, let $L1 = \{a^n b^n \mid n \geq 0\}$. Using pumping lemma for CFL's, the following proof claims to prove that L1 is not a CFL. Is the proof right? Justify your answer. Symbol e denotes string of length zero.

<u>Proof</u>: Suppose L1 is a CFL and p be the pumping length. Let $w = a^p b^p$. From pumping theorem we know that w can be divided in five parts uvxyz such that $uv^ixy^iz \in L1$ for all $i \geq 0$.
Let $v = a^k$ such that $0 < k < p$. Now, $u = a^{p-k}$, $v = a^k$, $x = e$, $y = e$, and $z = b^p$.
Therefore, now for i = 0 the resulting string $w = a^{p-k} b^p \notin L1$.
Therefore, we can conclude that L1 is not a CFL.

Sol: Above Proof is not correct because of following reasons:
Pumping theorem says that each string <u>w can be divided in five parts uvxyz such that $uv^ixy^iz \in$ L1 for all $i \geq 0$.</u>
Now, for the given string there exist five parts such that the conditions are true.
Let $v = a^{p/2}$ and $y = b^{p/2}$. Now, $u = a^{p/2}$, $v = a^{p/2}$, $x = e$, $y = b^{p/2}$, and $z = b^{p/2}$.
With this, $uv^ixy^iz \in L1$ for all $i \geq 0$.

Q4. [**5 Marks**] Consider the CFG G = ({S, A, B}, {0, 1, 2}, {S->AB, A->0A1|e, B->2B|e}, S). Complete the following deterministic PDA M for the language of G such that it is capable of generating the leftmost derivation of the parse tree of a given input string. Assume $ is the bottom of the stack, # is the end-marker of the input string, and e denotes the string of length zero.

M = ({p, q, q0, q1, q2, q#, qf}, {0, 1, 2, #}, {0, 1, 2, S, A, B, $}, Δ, p, {qf}), where Δ is given as:
((p, e, e) (q, S$))
((q, 0, e) (q0, e))
((q, 1, e) (q1, e))
((q, 2, e) (q2, e))
((q, #, e) (q#, e))
((q0, e, 0) (q, e))
((q1, e, 1) (q, e))
((q2, e, 2) (q, e))
…..

Sol:

((q0, e, S)(q0, AB)), ((q0, e, A)(q0, 0A1))	1M
((q1, e, A)(q1, e))	1M
((q2, e, B) (q2, 2B))	1M
((q#, e, B)(q#, e))	1M
((q#, e, $)(qf, e))	1M

Q5. [**5 Marks**] For a given language L, define EVEN (L) as the set of all strings of L with even length. In other words, EVEN(L) = {x | x $\in$ L and (|x| mod 2) = 0}. Is the following statement TRUE/FALSE? Prove your answer.

"If L is a CFL, EVEN (L) is also a CFL".

Sol: The language of even length strings is regular. For example, let L1 = (00 + 01 + 10 + 11)*. Now, for any given language L, EVEN (L) = L intersection L1. Since, CFL intersection RL is a CFL, therefore, the statement is TRUE.

Q6. [**7 Marks**] If a stack contains K symbols, let the middle symbol of a stack (denoted as Γ_{MID}) be defined as the $\lceil (K/2) \rceil$th symbol from the bottom of the stack. Now, let the six tuples of a general PDA (called as M_PDA) be defined as:

M_PDA M = (K, Σ, Γ, Δ, s, F), where K is the finite set of states, Σ is the finite set of input symbols, Γ is the finite set of stack symbols, s is the start state, F is the set of final states, and Δ is given by:

(K x {Σ U ε} x {Γ U ε} x {Γ_{MID} U ε}) to (K x {Γ U ε}). Note that the symbol corresponding to Γ_{MID} is not popped out of the stack, and is just read.

This means that the transition of a PDA will depend upon the current state, input symbol read, top of the stack, and middle symbol of the stack. Depending upon these four tuples, it can change the state, and push a symbol on top of the stack.
Claim: M_PDA is more powerful than a normal PDA, i.e. if a PDA can read the middle symbol of the stack, it is computationally more powerful than the one which cannot read the middle symbol.

Proof: M_PDA accepts the language L = {a^n b^n c^n | n > 0}. Check if the following high level design of M_PDA is correct or not. If it is correct, just write "DESIGN is CORRECT". Else, state where (or why) the M_PDA design is wrong and how it can be corrected.

Step-1: For each input symbol "a", the M_PDA pushes a symbol "X" on the stack.
Step-2: For each input symbol "b", if Γ_{MID} is "X", push "Y" on top of the stack. Rather, on input symbol "b" if Γ_{MID} is not "X", or after reading a "b" if it reads an "a", the string is rejected.
Step-3: For each input symbol "c", if top of the stack is "Y", pop it out. Rather, if an "a" or "b" is read after a "c", the string is rejected.
Step-4: Finally, if the input is finished and top of the stack is "X", accept the string.
Sol: The proof is wrong in the step 3. The given PDA accepts the string a^{10} b^9 c^9. Therefore, in step 3 we need to check for the first "c", there has to be "Y" at mid of the stack (Γ_{MID}). This ensures that number of a's = number of b's. Then follow step -3 and step-4.

Marking Scheme:

1M for saying that PDA is wrong.

2M for identifying the string which would not be accepted.

4M for correct modification.