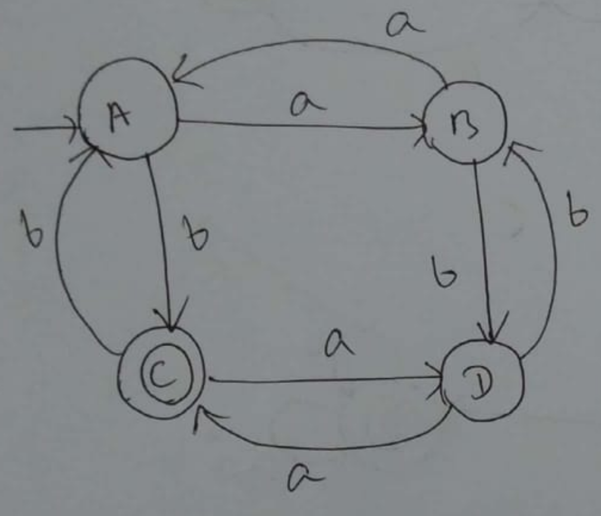
IAT - II, OCT - 2018

Scheme & Solution

Prepared By:
Sagarika Behera,
Dept. of CSE

Sub: Automata Theory & Computability (15CS54)

1. (i) $L = \{ w \in \{a,b\}^* : w$ contains an even no. of a's and an odd no. of b's$\}$ (5M)



Regular grammar

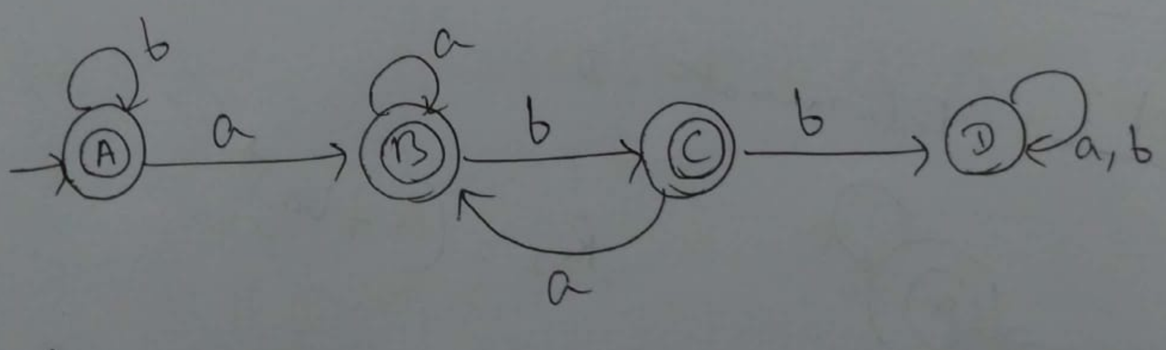
A → aB | bC

B → aA | bD

C → aD | bA | ∈

D → aC | bB

(ii) $L = \{ w \in \{a,b\}^* : w$ doesn't contain the substring abb$\}$ (5M)



Regular grammar

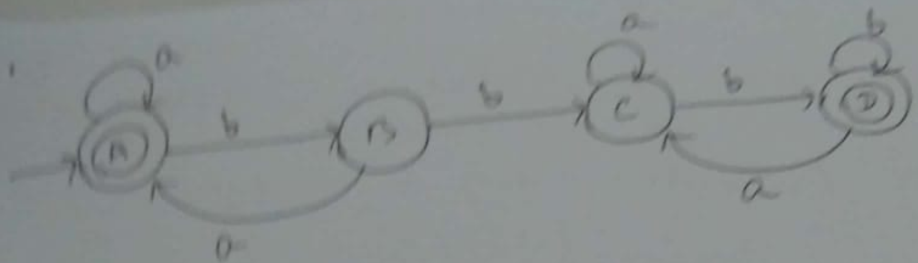
A → aB | bA | ∈

B → aB | bC | ∈

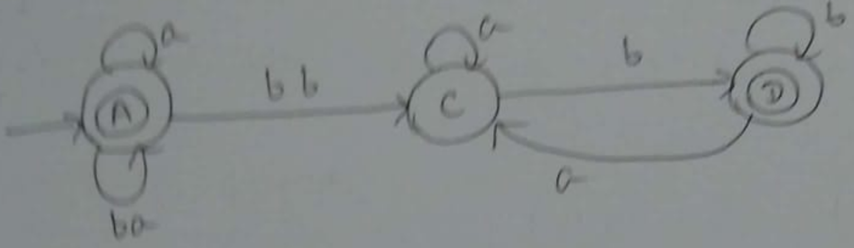
C → aB | bD | ∈

D → aD | bD

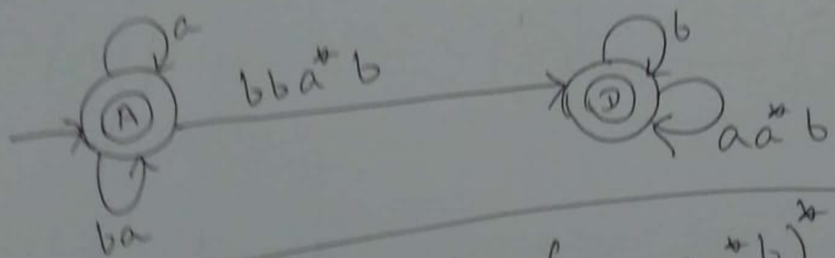
2.



Step 1 Eliminate B

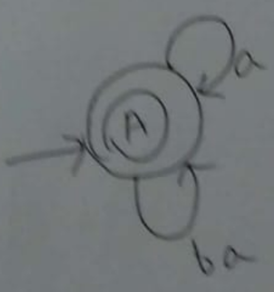


Step 2 Eliminate C



$$\boxed{R_1 = (a+ba)^* \, bba^*b \, (b + aa^*b)^*}$$

Step 3 Eliminate D



$$R_2 = (a+ba)^*$$

Final r.e is $R = R_1 + R_2$
$= (a+ba)^* \left[bba^*b \, (b + aa^*b)^* + \epsilon \right]$

Scanned by CamScanner

Ambighous Grammar

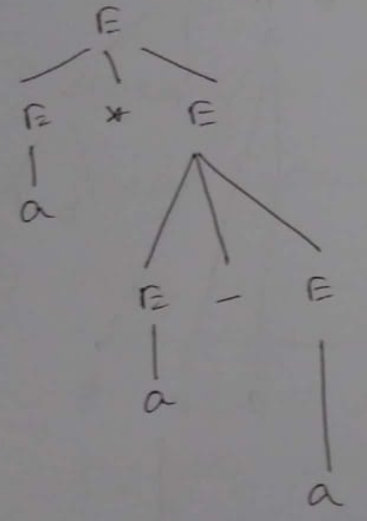
[6+4]

→ A grammar is ambighous, if there exist more than one LMD or more than one RMD for the same string.

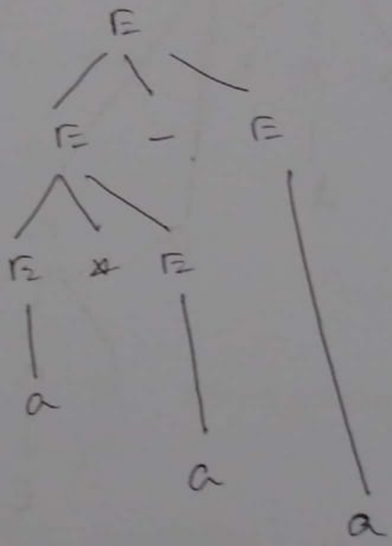
$E \to E+E \mid E-E \mid E*E \mid E/E \mid (E) \mid E \wedge E \mid a$

$w = a*a-a$

LMD1



LMD2



Yes, the grammar is ambighous.

Equivalent Unambighous grammar is

$E \to E+T \mid E-T \mid T$

$T \to T*F \mid T/F \mid F$

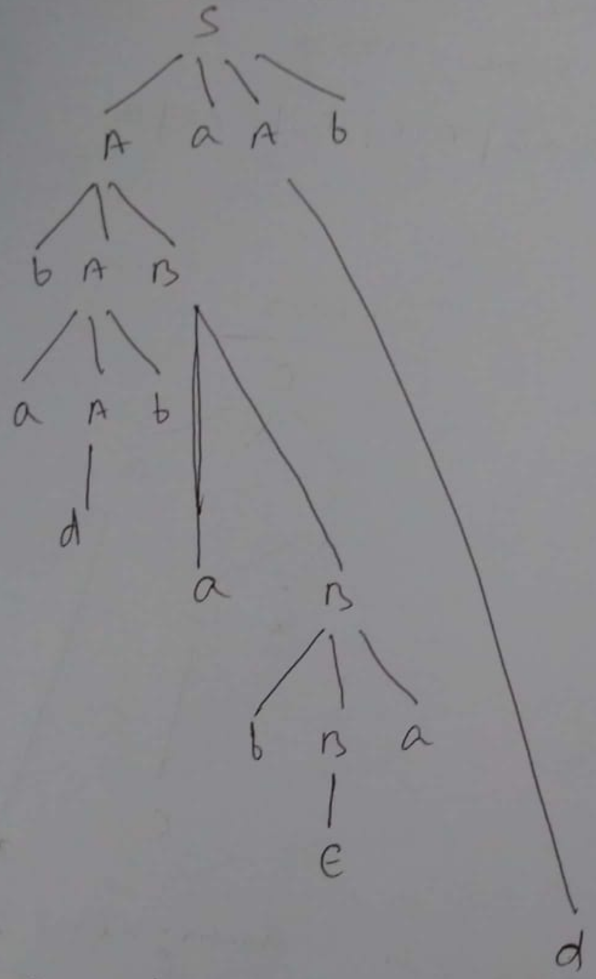
$F \to G \wedge F \mid G$

$G \to (E) \mid a$

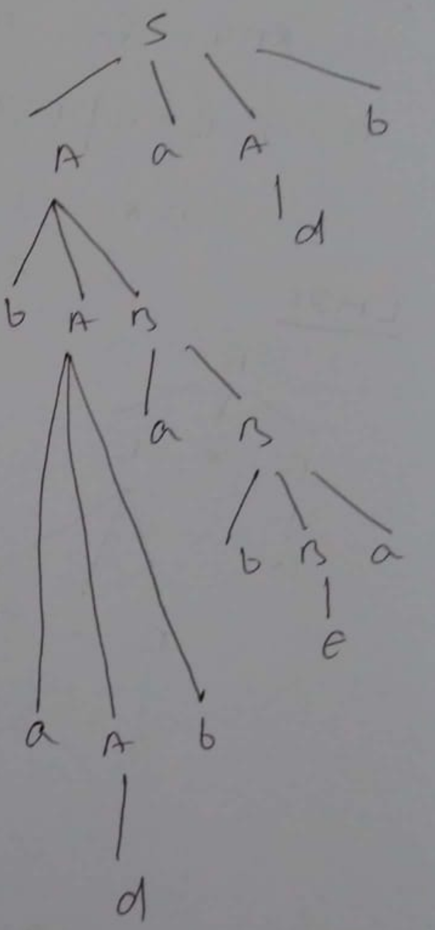
4.

W = badbabaadb (S

LMD Parse Tree



RMD Parse Tree



$S \Rightarrow AaAb$

$\Rightarrow bABaAb \quad (A \to bAB)$

$\Rightarrow baAbBaAb \quad (A \to aAb)$

$\Rightarrow badbBaAb \quad (A \to d)$

$\Rightarrow badbaBaAb \quad (B \to aB)$

$\Rightarrow badbabBaaAb \quad (B \to bBa)$

$\Rightarrow badbab\epsilon aaAb \quad (B \to \epsilon$

$\Rightarrow badbabaadb \quad (A \to d)$

LMD

$S \Rightarrow AaAb$

$\Rightarrow Aadb \quad (A \to d)$

$\Rightarrow bABadb \quad (A \to bAB)$

$\Rightarrow bAaBadb \quad (B \to aB)$

$\Rightarrow bAabBaadb \quad (B \to bBa)$

$\Rightarrow bAab\epsilon aadb \quad (B \to \epsilon)$

$\Rightarrow baAbabaadb \quad (A \to aAb)$

$\Rightarrow badbabaadb \quad (A \to d)$

5. (a) <u>Push Down Automata</u> (4M)

→ It is an automata to recognize the context free language.

It is defined by 7 - tuples.

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

where Q is a finite set of states

Σ is finite set of input alphabets.

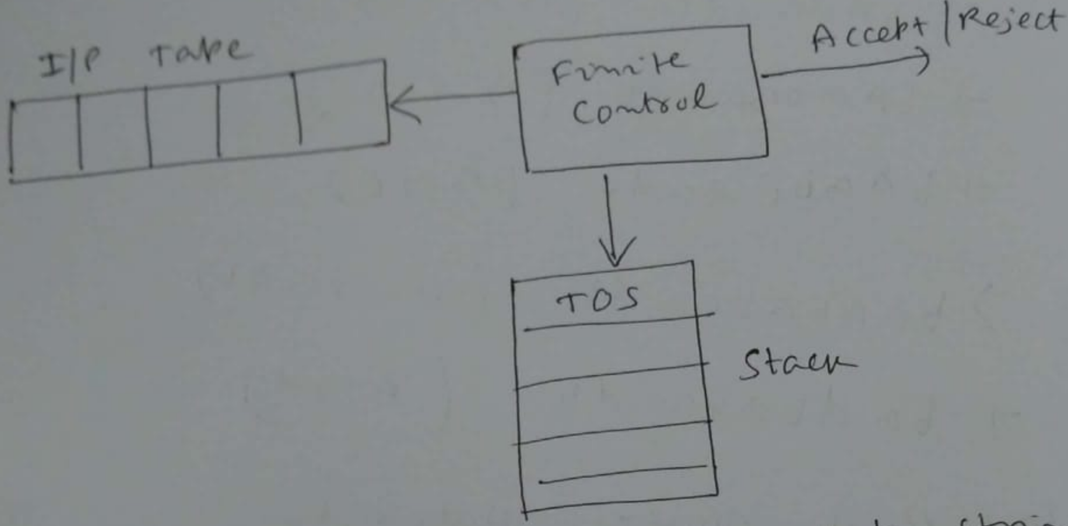
Γ is set of stack alphabets.

δ is transition function.

$\delta : Q \times \Sigma \cup \{\epsilon\} \times \Gamma \to Q \times \Gamma^*$

q_0 is start state.
Z_0 is initial stack symbol
F is final state.

PDA Block diagram



→ The I/P tape contains input string.

→ Finite control checks the current input, top of stack (TOS), then decides which operation to do (Push or POP).

→ When input becomes empty, stack is empty or it reached the final state, then it is accepted, otherwise it is rejected.

(b) $L = \{ w c w^R \mid w \in \{a,b\}^* \}$ (6M)

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, b, Z_0) = (q_0, bZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, c, a) = (q_1, a)$

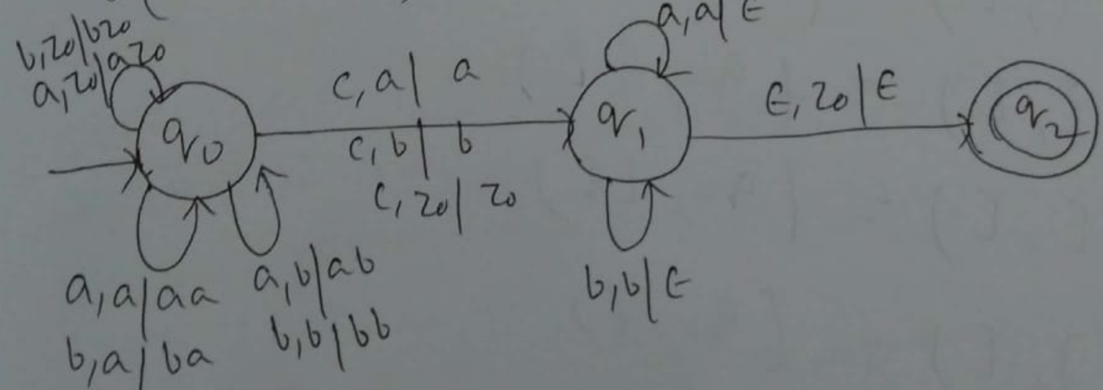
$\delta(q_0, c, b) = (q_1, b)$

$\delta(q_0, c, Z_0) = (q_1, Z_0)$

$\delta(q_1, a, a) = (q_1, \epsilon)$

$\delta(q_1, b, b) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$



ID for $w = abcba$

$(q_0, abcba, Z_0) \vdash (q_0, bcba, aZ_0) \vdash (q_0, cba, baZ$

$\vdash q_1, ba, baZ_0 \vdash q_1, a, aZ_0 \vdash q_1, \epsilon, Z_0$

$\vdash q_2, \epsilon, \epsilon$ Accepted

(2

Def^n of DPDA & NPDA (2M)

Design of PDA (8M)

6. $\delta(q_0, (, Z_0) = (q_0, (Z_0)$

$\delta(q_0, \{, Z_0) = (q_0, \{Z_0)$

$\delta(q_0, [, Z_0) = (q_0, [Z_0)$

$\delta(q_0, (, () = (q_0, (()$

$\delta(q_0, \{, () = (q_0, \{()$

$\delta(q_0, [, () = (q_0, [()$

$\delta(q_0,), () = (q_0, \epsilon)$

$\delta(q_0, (, \{) = (q_0, (\{)$

$\delta(q_0, \{, \{) = (q_0, \{\{)$

$\delta(q_0, [, \{) = (q_0, [\{)$

$\delta(q_0, \}, \{) = (q_0, \epsilon)$

$\delta(q_0, (, [) = (q_0, ([)$

$\delta(q_0, \{, [) = (q_0, \{[)$

$\delta(q_0, [, [) = (q_0, [[)$

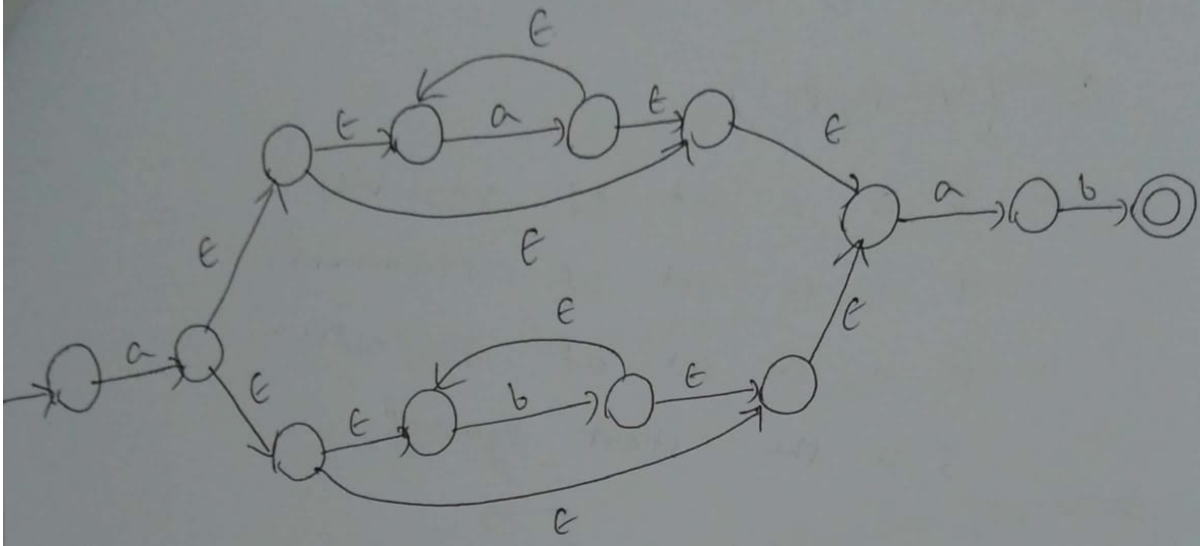
$\delta(q_0,], [) = (q_0, \epsilon)$

$\delta(q_0, \epsilon, Z_0) = (q_1, \epsilon)$

It is <u>NPDA</u>. Since there is a transition on $\delta(q_0, (, Z_0)$ & $\delta(q_0, \epsilon, Z_0)$.

Scanned by CamScanner

(a) $r = a(a^* + b^*)ab$ (6M)



(4M)

(b) Sentence

→ A sentence is a string which contains only terminals.

Sentential form

When we derive a string from the start symbol S, the intermediate steps are called sentential form, which is a combination of terminals & non terminals.

Ex $E \Rightarrow E + E$ } sentential form

$\Rightarrow a + E$

$\Rightarrow a + a$ → sentence

8. CFG (10n)

A CFG G is defined by 4-tuples

$G = (V, T, P, S)$

where V is a set of variables

T is a set of terminals

P is a set of productions

S is the start symbol.

(i) $L = \{ a^n b^{n+3} \mid n \geq 0 \}$

$S \rightarrow aSb \mid bbb$

(ii) $L = \{$ Palindrome over $\{0,1\}^* \}$

$S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon$

(iii) $L = \{ a^i b^j c^k \mid j = i + 2k,\ i, j, k \geq 0 \}$

$a^i \ b^j \ c^k$

$a^i \ b^{i+2k} \ c^k$

$a^i \ b^i \ b^{2k} \ c^k$

$S \rightarrow AB$

$A \rightarrow aAb \mid \epsilon$

$B \rightarrow bbBc \mid \epsilon$