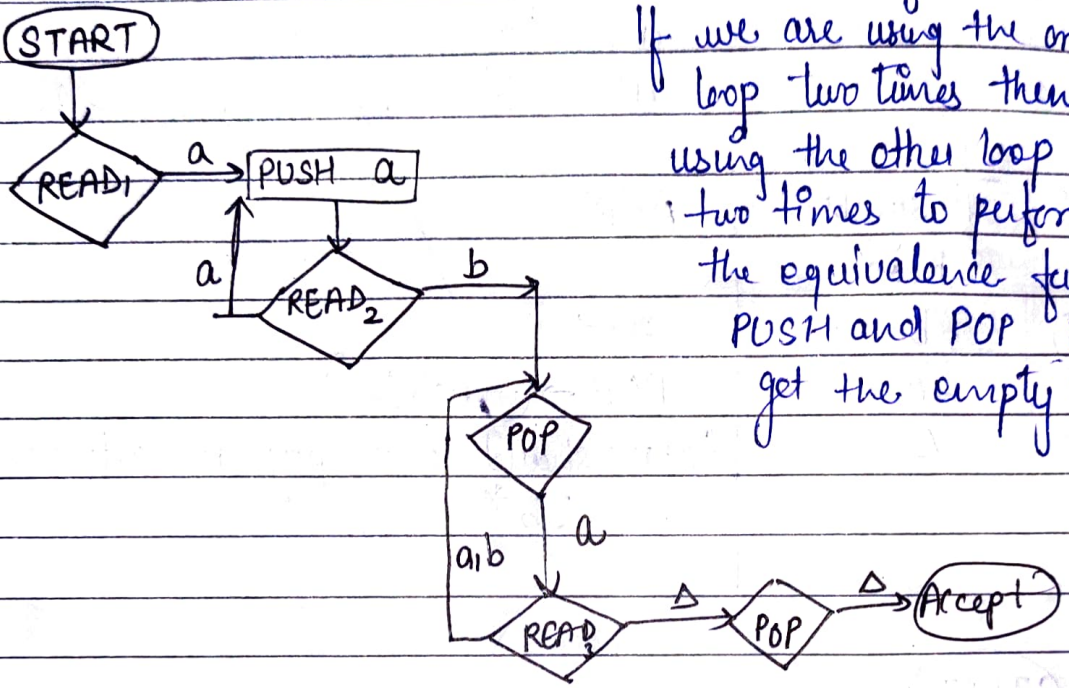
$a^* ab(a+b)^*$

Date : / /
Page No.

Ques/ Draw PDA for $(a+b)^* ab$

Ques. Determine the lg accepted by PDA.

②



If we are using the one loop two times then using the other loop two times to perform the equivalence fun^n PUSH and POP to get the empty stack.

logic: ① Complete all states (provide transition)
② L = { ✓, x x, }
② Smallest possible word (no. of read state)

Smallest word = ab

L = { ab, aaba, aabb, aaabaa, aaabab, ... }

L = $lg(a^n b (a|b)^{n-1})$ OR $a^n b (a+b)^{n-1}$

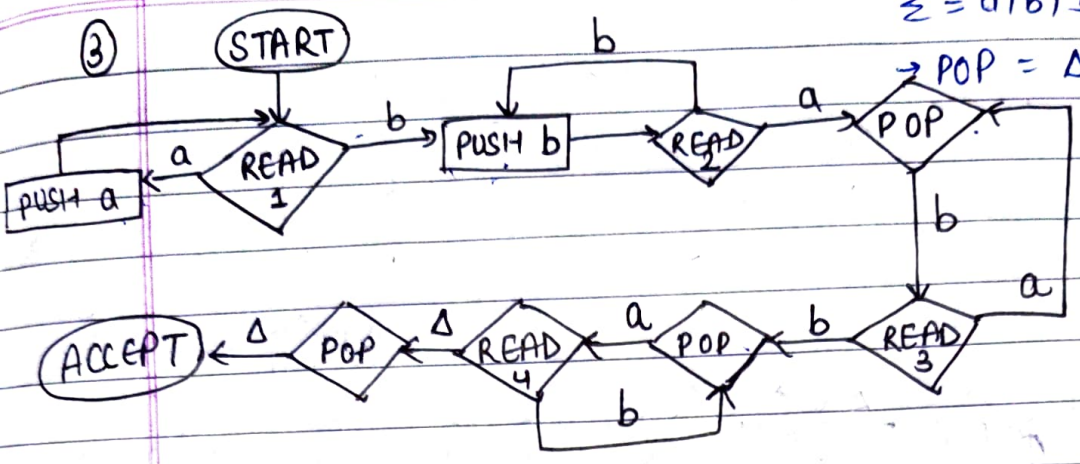
$a^n b^m a^m b^n$

Date : / /
Page No.

→READ
$\Sigma = a, b, \Delta$

→POP = Δ, a, b

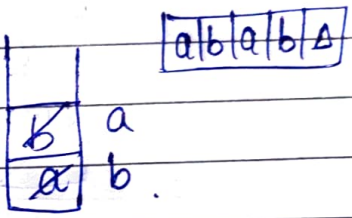
③



$L = \{abab, \cancel{aabb}\ aabbaabb, aabbbbaaabb, \ldots\}$

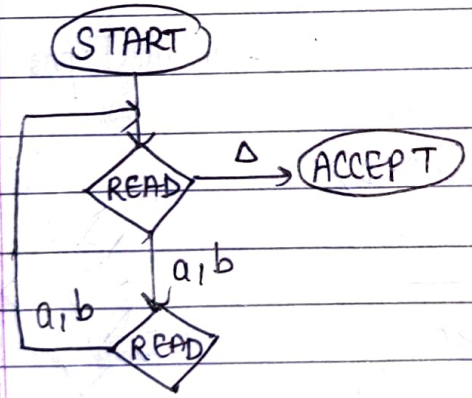
$L = Lg(a^n b^m a^m b^n)$.

→



smallest word = abab

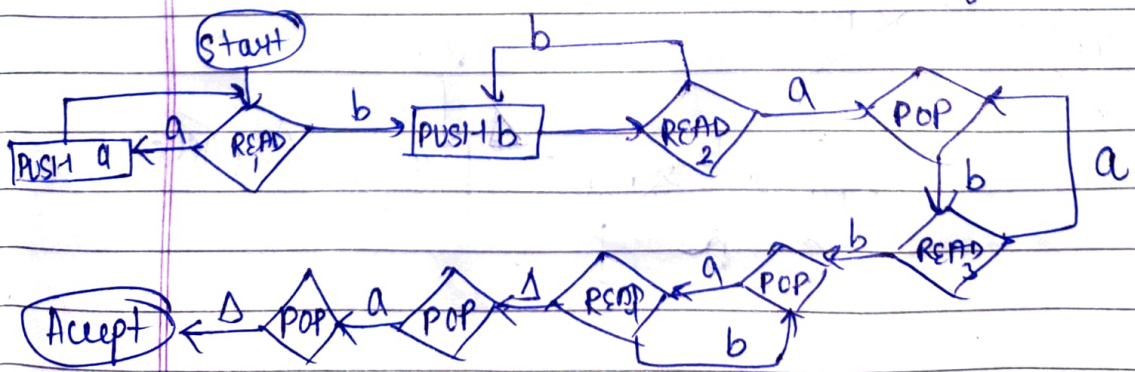
④



smallest word = $\{aa, ab, ba, bb, \ldots\}$

L = even length words.

⑤ $a^{n+1} b^m a^m b^n$ (~~at least~~ one a should be there at last after reading b)



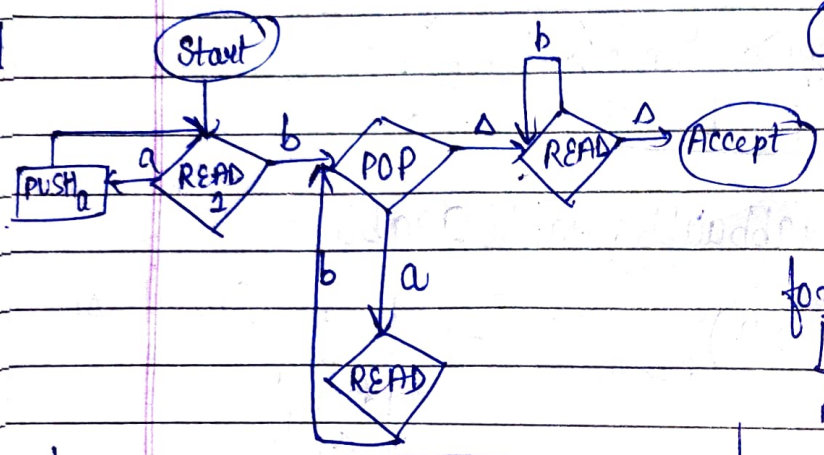
Date : / /
Page No.

Ques Draw the PDA for.

a) $a^k b^l$, $k < l$

$a^k b^k b^m$, $m \geq 1$

Logic	Symbol Read	Action
①	a	→ PUSH
②	b	→ POP (match a)
③	b is being read	→ Δ



(There is atleast one extra b. Then read b until you encounter Δ)

for eg.

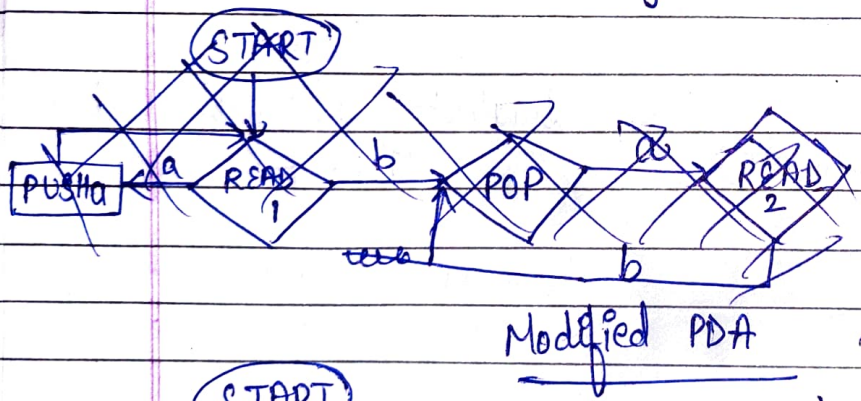
a	a	b	b	b	Δ

PUSH PUSH POP POP POP Read (b)

a a Δ

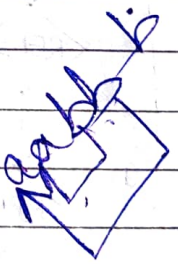
In this PDA, single b is getting accepted, we need to modify this.

a	b
a	b

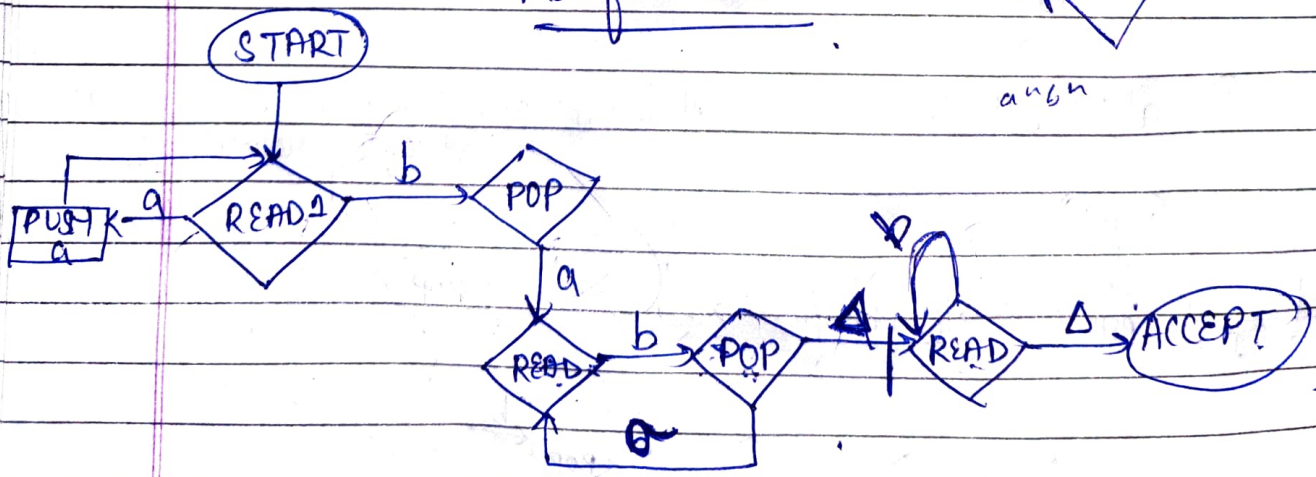


Modified PDA

→ smallest possible word = abb



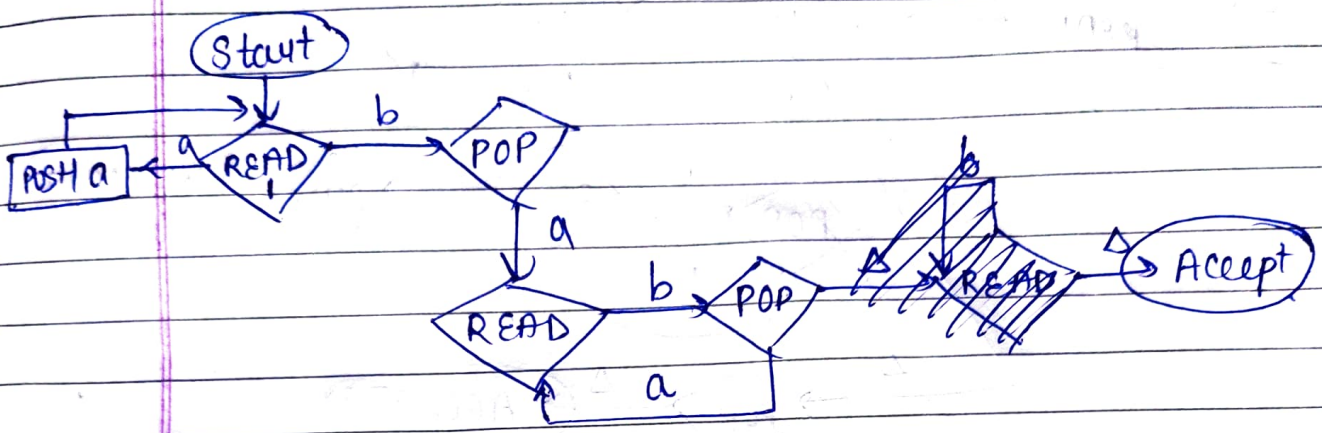
a n b n



Date : 11/11
Page No.

~~b) $(a+b)^k c^k$~~

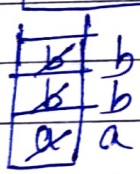
(b) $a^n b^{n+1}$



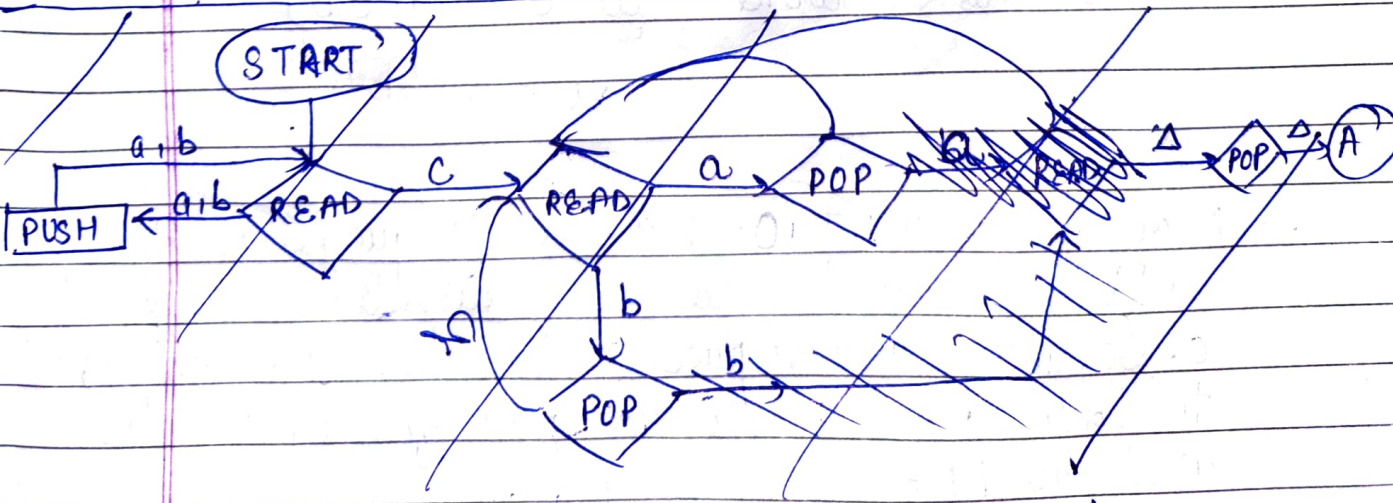
(c) odd Palindrome

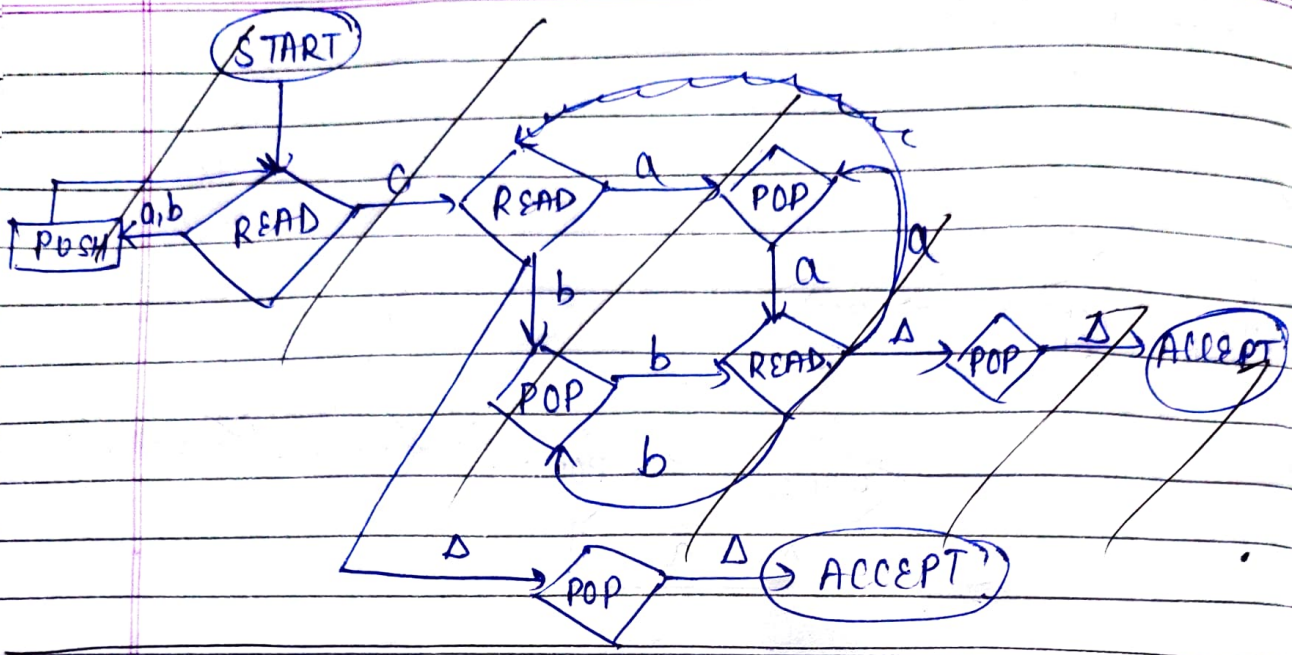
w c rev(w)

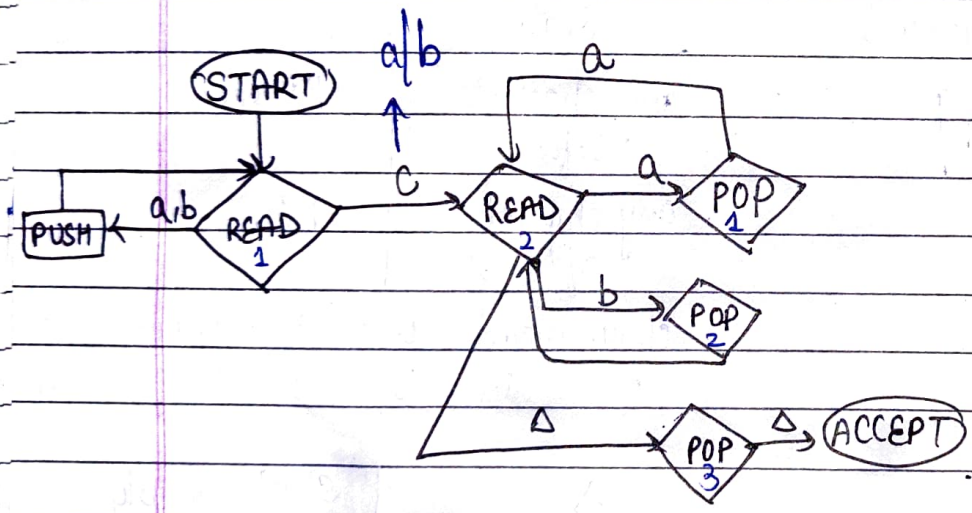
abb c bba



I/P Symbol Read	Action
any character before C	PUSH
any character after C	POP
char. read a →	ele. popped a
" " b →	" " b
Δ	stack empty







This PDA is for word $w\ c\ rev(w)$

abb c bba.

→ Now if we want or have a language with a/b in middle in odd Palindrome except of c. So, we clearly replace c by a/b. Then we need to choose a a correct path for a particular word in PDA.

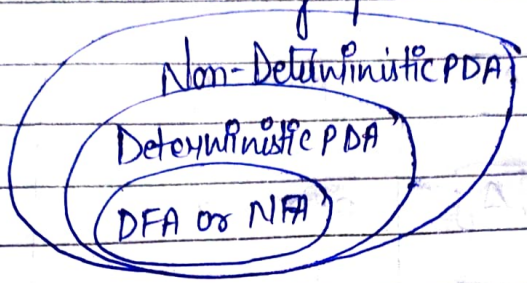
eg. aba → Path → PUSH → READ 1 → READ 2 → POP 1 → READ 2 → POP 3 → ACCEPT

Date : / /
Page No.

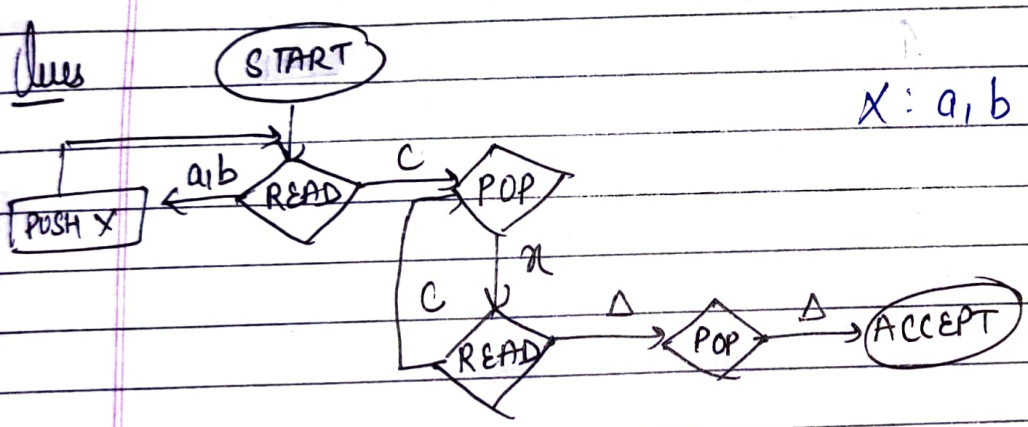
Input

Note * Deterministic PDA, every string has a unique path to the machine or to the PDA.

Non-Deterministic PDA, for a particular string we may have to choose among possible paths to the machine.



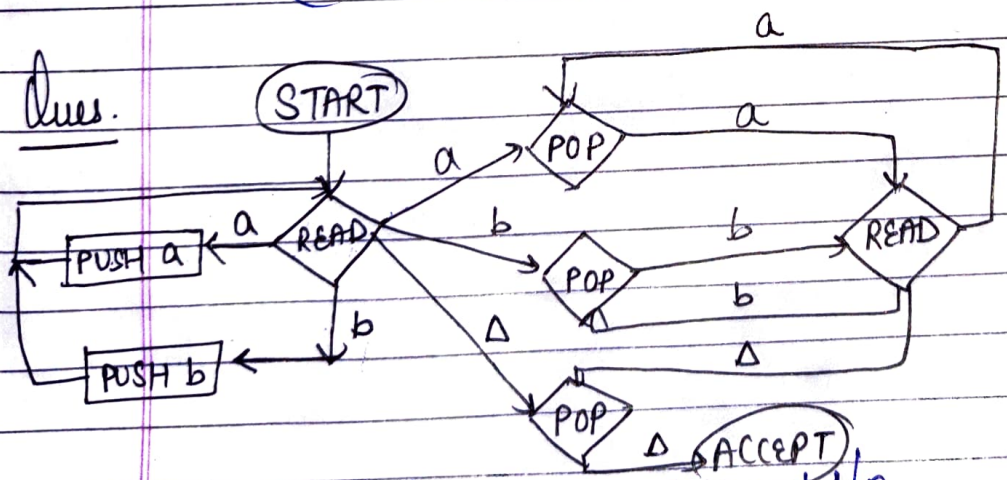
Ques



$\Sigma : a, b$

$L = (a+b)^n c^n$

Ques.



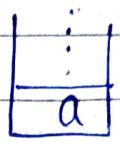
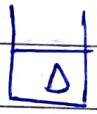
→ Even Palindrome
w. w^R (w).
eg. Δ, ab, abba

logic:

I/P	Action
Initial half a's or b's	PUSH
final half a's or b's	POP
Δ	POP = Δ
char a read	→ popped a
char b read	→ popped b

Date : / /
Page No.

eg.

Input tape	Action	Stack
a b b a Δ ..	PUSH a	
a b b a Δ ..	PUSH b	
a b b a Δ	POP b	
a b b a Δ	POP a	
a b b a Δ	POP = Δ (accept)	

Date : / /
Page No.

*. Properties of CFL

A language L is CFL if it can be represented by a CFG.

CFL is closed under
1) Union
2) Concatenation
3) Kleene Closure

i.e. if L_1 & L_2 are CFLs, then so are
$L_1 + L_2$
$L_1 . L_2$
$(L_1)^*$.

① If L_1 — CFL — CFG1
L_2 — CFL — CFG2

$L_1 + L_2 \rightarrow$ CFL $\rightarrow$ CFG3.

$L_1 = a(a+b)^*$
CFG1
$S \rightarrow aX$
$X \rightarrow aX | bX | \lambda$

$L_2 = b(a+b)^*$
CFG2
$S \rightarrow bX$
$X \rightarrow aX | bX | \lambda$

→ Proof of ~~Conten~~ Concatenation Union

① Rename non-terminal of CFG1 & CFG2.

CFG1:
$S_1 \rightarrow aX_1$ — PR1
$X_1 \rightarrow aX_1 | bX_1 | \lambda$ — PR2

CFG2:
$S_2 \rightarrow bX_2$ — PR3
$X_2 \rightarrow aX_2 | bX_2 | \lambda$ — PR4

Now, CFG3 $\rightarrow L_1 + L_2$

$S \rightarrow S_1 | S_2$
$S_1 \rightarrow aX_1$, $S_2 \rightarrow bX_2$
$X_1 \rightarrow aX_1 | bX_1 | \lambda$, $X_2 \rightarrow aX_2 | bX_2 | \lambda$

Date: / /
Page No.

This proves, $L_1 + L_2$ is CFL.

② Concatenation

$L_1 = a(a+b)^*$

CFG1

$S \rightarrow aX$

$X \rightarrow aX \mid bX \mid \lambda$

$L_2 = b(a+b)^*$

CFG2

$S \rightarrow bX$

$X \rightarrow aX \mid bX \mid \lambda$

Proof of Concatenation

① Rename non-terminal of CFG1 & CFG2

$S_1 \rightarrow aX_1$ — PR1
$X_1 \rightarrow aX_1 \mid bX_1 \mid \lambda$ — PR2] CFG1 | CFG2 [$S_2 \rightarrow bX_2$ — PR3
$X_2 \rightarrow aX_2 \mid bX_2 \mid \lambda$ — PR4

Now, CFG3 $\longrightarrow L_1 . L_2$

$S \rightarrow S_1 . S_2$

$S_1 \rightarrow$

$S_2 \rightarrow$

$X_1 \rightarrow$

$X_2 \rightarrow$

③ Kleene Closure (*)

eg:

$L_1 = a(a+b)^*$

CFG1

$S \rightarrow aX$

$X \rightarrow aX \mid bX \mid \lambda$

Date : / /
Page No.

Proof of kleene Closure

① Rename non-terminal of CFG 1

$S_1 \rightarrow aX_1$
$X_1 \rightarrow aX_1 | bX_1 | \lambda$

$\therefore L(aw, CFG_2) = (L_1)^*$ ✱

$S_1 \rightarrow SS_1 | \lambda$
$S_1 \rightarrow aX_1$
$X_1 \rightarrow aX_1 | bX_1 | \lambda$

✱. Are Context Free languages closed under intersection?

Case1

eg:- L_1 = Palindrome excluding λ — L_2 = string of all a's (aa^*) excluding λ

$S_1 \rightarrow aS_1a | bS_1b$
$S_1 \rightarrow a | b$
$S_1 \rightarrow aa | bb$

$S_2 \rightarrow aS_2 | a$

$L_3 = L_1 \cap L_2 = aa^*$.

Case2

eg:- $L_1 = \{a^n b^n c^i ; n, i \geq 1\}$ — $L_2 = \{a^i b^n c^n , n, i \geq 1\}$

$S_1 \rightarrow X_1 Y_1$
$X_1 \rightarrow aX_1b | ab$
$Y_1 \rightarrow cY_1 | c$

$L = \{abc, abcc, aabbc, aabbcc, \ldots\}$

$S_2 \rightarrow X_2 Y_2$
$X_2 \rightarrow aX_2 | a$
$Y_2 \rightarrow bY_2 c | bc$

$L = \{abc, aabc, abbcc, aabbcc \ldots\}$

Date : / /
Page No.

$L_1 \cap L_2 = \{a^n b^n c^n\} \neq CFL$

→ We cannot write the CFL for this example, so, CFL are not closed under intersection.

*. Are Context Free languages closed under Complement?

Method 1 $L_1 \rightarrow CFL \rightarrow CFG$

$\hookrightarrow L_1' \rightarrow CFL \rightarrow CFG$

eg. $L_1 = a^{2n} b^n c^i$ Now, $L_1' \rightarrow CFL \rightarrow CFG$

CFG →

$S \rightarrow aaSb$ $a^n b^n c^n \subseteq L_1'$

$\hookrightarrow \neq CFG$

Method 2 (all possible words defined over $\{a,b,c\}$ excluding $a^{2n} b^n c^i$)

Assumption ① Let us assume that L_1 & L_2 are closed under complementation.

If L_1 & L_2 are CFL

L_1', L_2' — CFL (by assumption 1)

$L_1' + L_2'$ — CFL (CFL are closed under union)

$(L_1' + L_1'')'$ — CFL (by assumption ①)

$= L_1 \cap L_2$ (As CFLs are not closed under intersection. Thus, assumption 1 failed)

and, CFL are not closed under Complement!