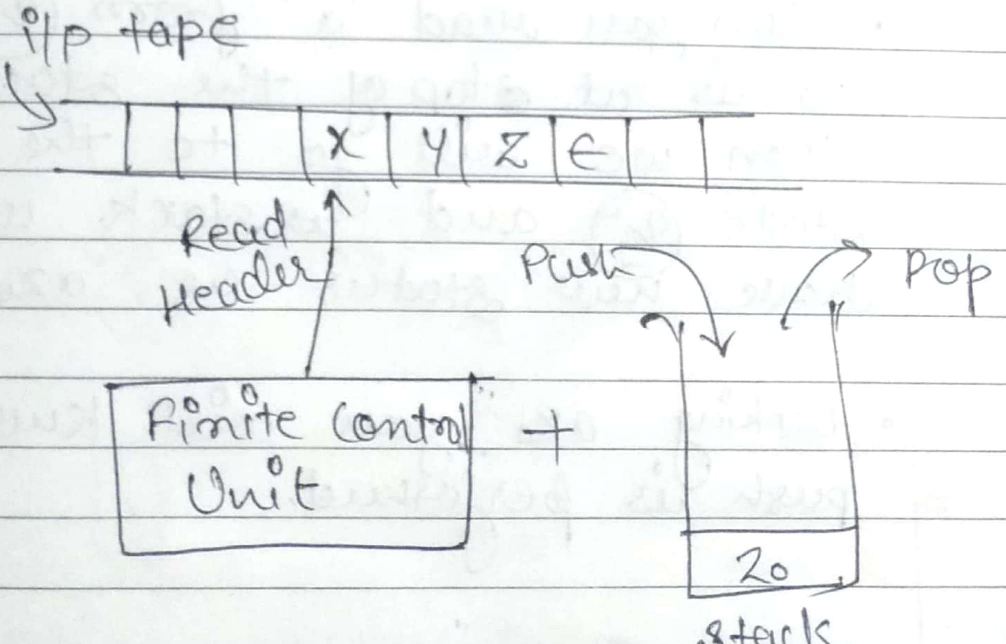
Push Down Automata

CFG $\xrightarrow{\text{generates}}$ CFL $\xrightarrow{\text{Accepted by}}$ PDA

PDA = $\{Q, \Sigma, q_0, f, z_0, \Gamma, \delta\}$

Q = Finite set of states
Σ = input alphabet
q_0 = initial state
f = set of final states
z_0 = Bottom/initial stack symbol
Γ = stack alphabet.
δ = transition function.

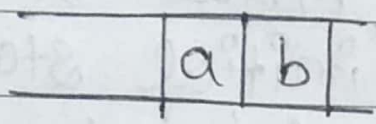


* Advantage of Stack :-
- Stack is zero address DS
- No need to give address to complete push & pop operation
- We assume that stack is of infinite length. So no need to look for overflow condition.

- Z_0 is used to indicate that stack is empty.

Operations on PDA

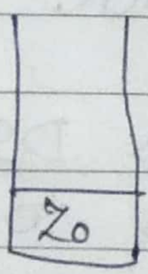
① PUSH :-



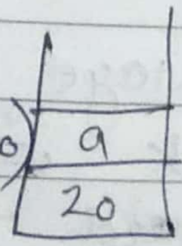
$$
\begin{tikzcd}
q_i \arrow[r, "{a, Z_0 / aZ_0}"] & q_j
\end{tikzcd}
$$

- When you read 'a' from q_i and 'Z_0' is at top of the stack then we will go to the new state q_j and stack will have new status as 'aZ_0'

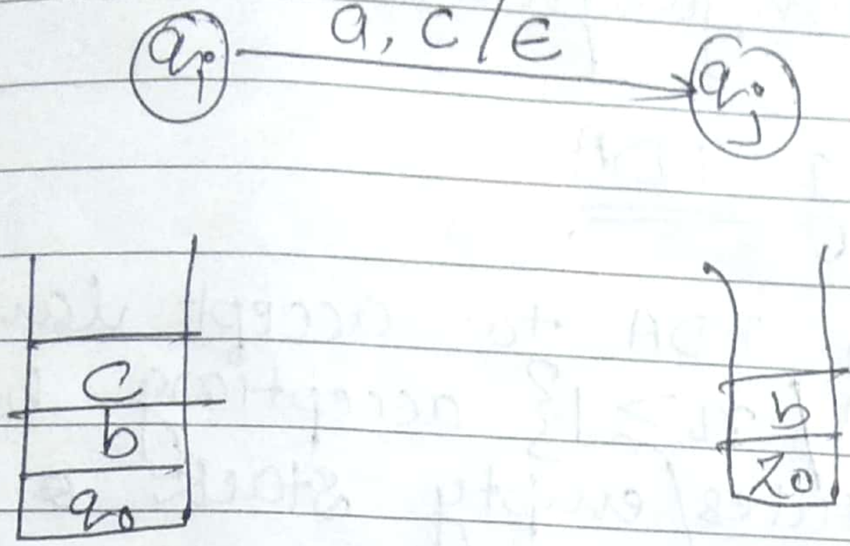
- Looking aZ_0 you will know that push is performed.



$\delta(q_i, a, Z_0) = (q_j, aZ_0)$



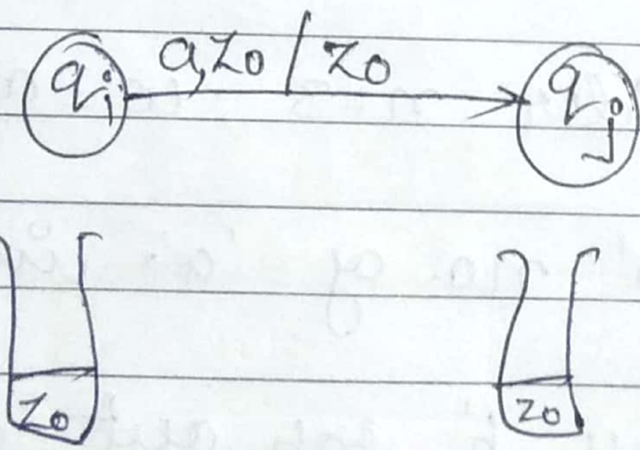
② POP :-



$$\delta(q_0, a, c) = (q_j, \epsilon)$$

ε represents pop operation.

③ Skip



$$\delta(q_i, a, Z_0) = (q_j, Z_0)$$

Representation of PDA

1. Transition Diagram
2. Transition fun^n/moves.

Designing PDA

Q1 ~~Pro~~ Design PDA to accept language $L = \{a^n b^n \mid n \geq 1\}$ accepting by final states/empty stack.

or

$L = \{0^n 1^n \mid n \geq 1\}$

Sol^n ① $L = \{a^n b^n \mid n \geq 1\}$

$L = \{$ ~~a,ab~~ ab, aabb, aaabbb - - - $\}$

② Algo (Logic)

consider $n = 3$, $w = aaabbb$

1. Push 'n' no. of 'a' into the stack
2. For every 'b' pop out an 'a' from the stack.
3. At the end of the string, the m/c stops as it reaches the final states.

(3) Implementation :-

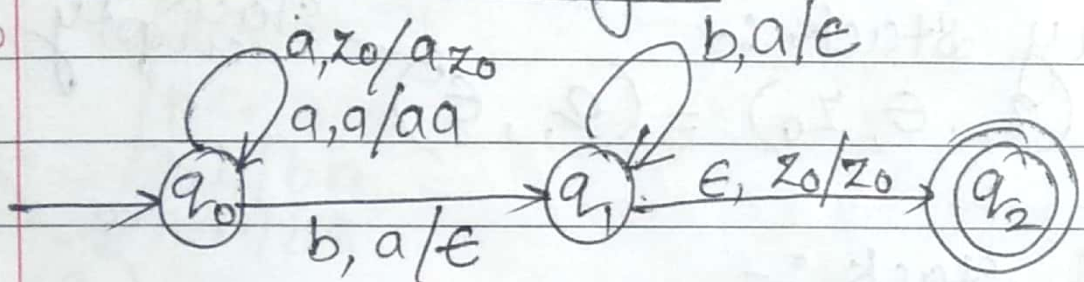
PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$Q = \{q_0, q_1, q_2\}$ $\quad q_0 = q_0$

$\Sigma = \{a, b\}$ $\quad z_0 = z_0$

$\Gamma = \{a, b, z_0\}$ $\quad F = \{q_2\}$

(4) Transition Diagram :-



(5) Transition Functions :-

push 'a' :-

$\delta(q_0, a, \underset{\substack{\uparrow \\ \text{Top of stack}}}{z_0}) = (q_0, az_0)$

push 'a' :-

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

push 'a' :-

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

push 'b' Pop

$\delta(q_0, b, a) = (q_1, \epsilon)$

push 'b'

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

push 'b'

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

Empty stack :- stack empty

$\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$

final stack :-

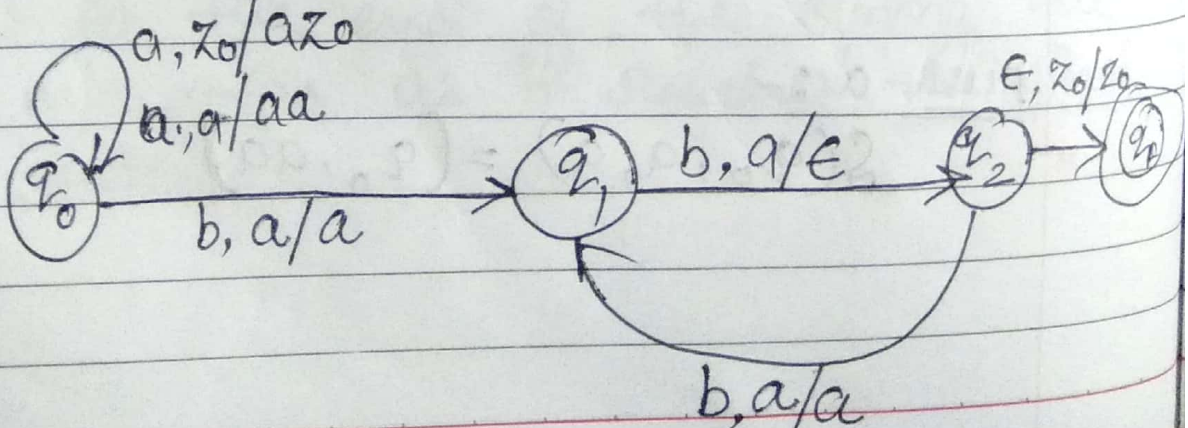
$\delta(q_1, \epsilon, z_0$

Q2 $L = \{a^n b^{2n} \mid n \geq 1\}$

a	a	a	b	b	b	b	b	b	ϵ

∵ a = 2 times of b

∵ we cannot push 2 b's at a time to pop 1 a



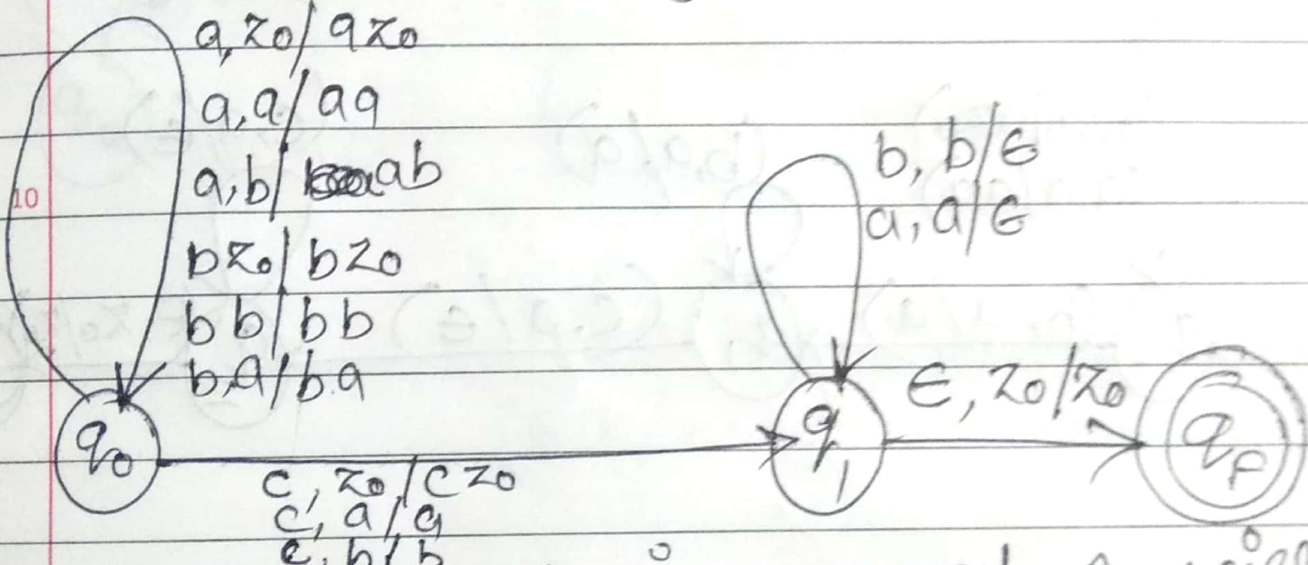
Q3 $L = \{wcw^R \mid w \in (a,b)^*\}$

a	b	b	c	b	b	a	ε

string seperator

Logic • c will always tell that w is finished

• stack is always poped in reverse.



• If c is scanned I will skip & change to remember that the string w is finished.

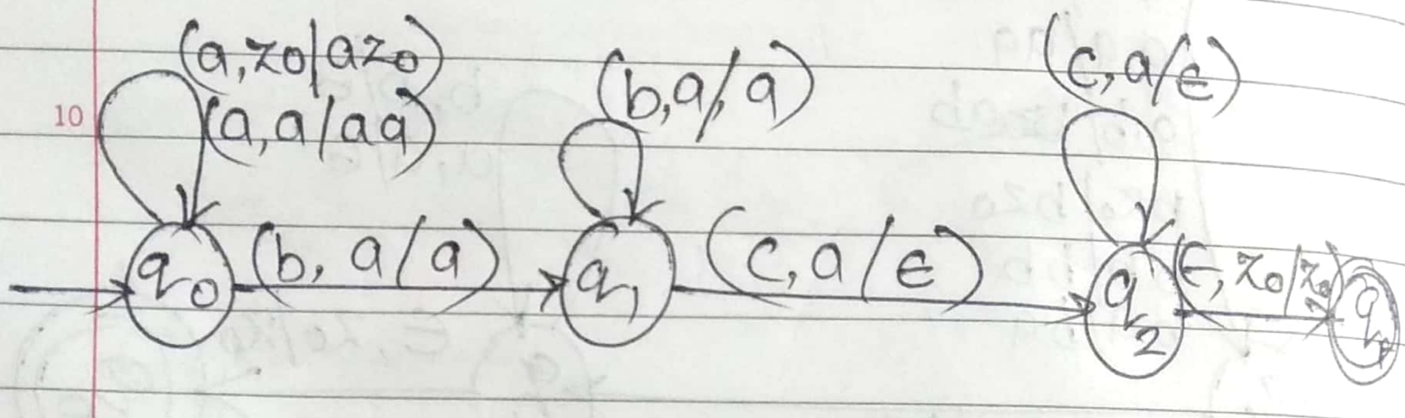
• when reverse string starts, if the scanned symbol is matched with stack top, that only the pop operation is performed.

Q4 $L=\{a^n b^m c^n \mid n \geq 1\}$

Logic :- Since the sequence abcabc
a b c

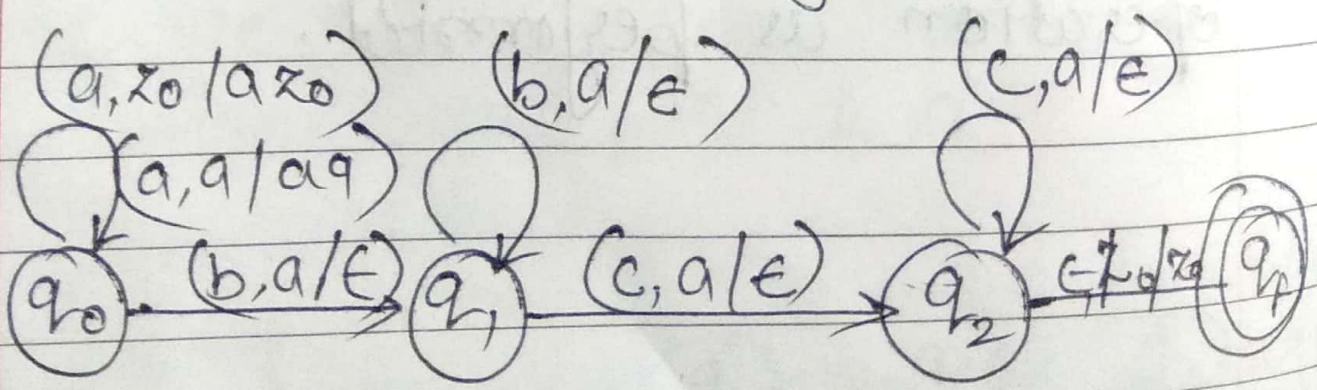
So after 'a' if 'b' comes simply skip the operation (NO PUSH/POP)

- we need to match ~~no~~ no. of c_s with a_s



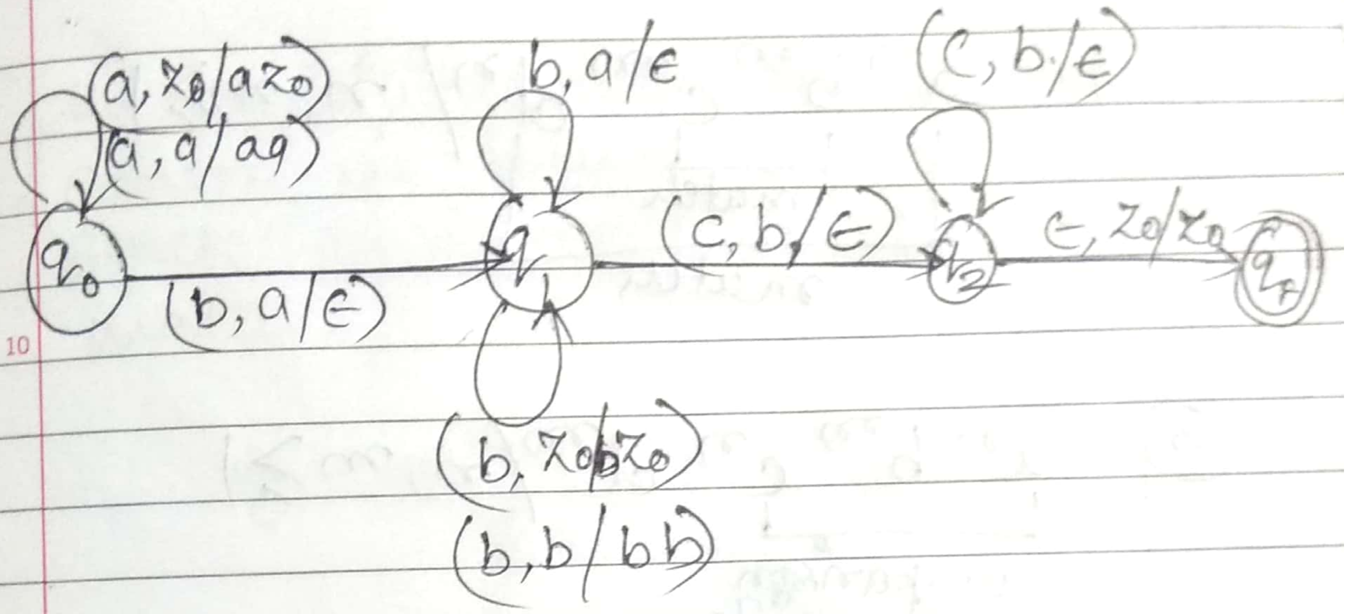
Q5 $L=\{a^{m+n} b^m c^n \mid m, n \geq 1\}$

Logic ⇒ we have to save & store all a's in stack compare ~~the~~ a's count with b_s and c_s together.

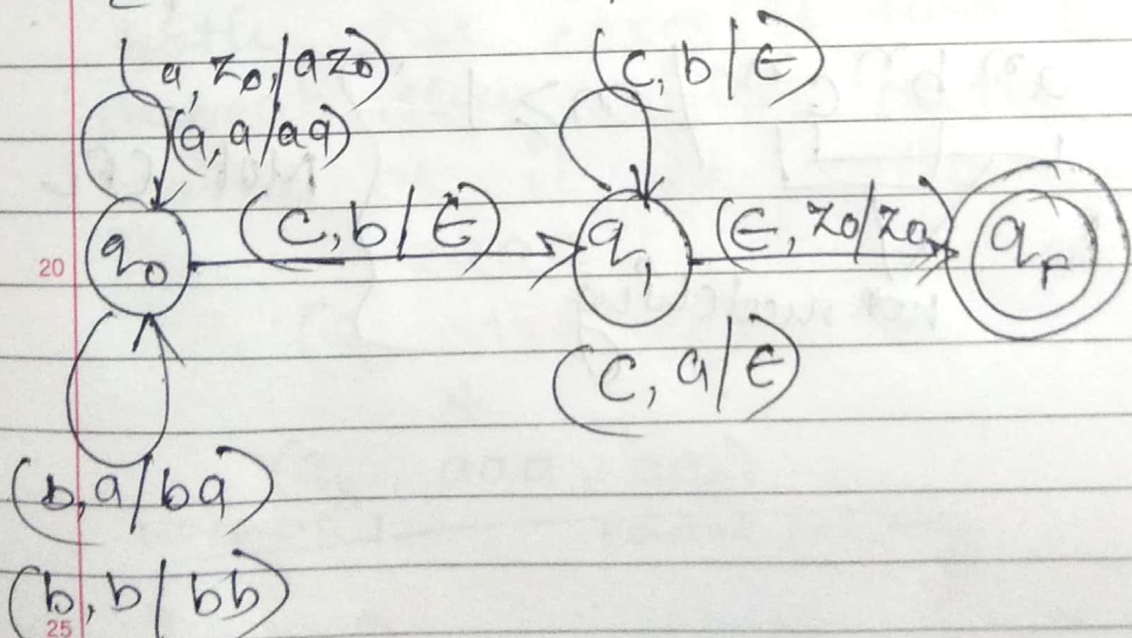


Q6 $L = \{a^n b^{n+m} c^m \mid n, m \geq 1\}$

$\underbrace{a^n b^n}_{\text{compare \& pop}} \underbrace{b^m c^m}_{\text{compare \& pop}}$



Q7 $L = \{a^n b^m c^{n+m} \mid n, m \geq 1\}$



[even if a comes after b there will not be any trasitions]

Practice

(1) $\underbrace{a^n b^n}_{\text{Match \& pop}} \underbrace{c^m d^m}_{\text{match \& pop}} \,/\, n, m \geq 1$

(2) $\underbrace{a^n \underbrace{b^m c^m}_{\text{match}} d^n}_{\text{match}} \,/\, n, m \geq 1$

(3) $\underbrace{a^n b^m c^n}_{} d^m \,/\, n, m \geq 1$

comparison not possible

Hence it is not CFL

(4) $a^n b^n c^n \,/\, n \geq 1$

not matching

Not CFL

Q $L = \{ww^R \mid w \in (a,b)^+\}$

eg aba aba

how to find centre (string separator)

The possibility ~~that the symbol~~ of centre is when top of the stack and input symbol must match.

eg aa | aa

So for this situation we need NPDA which will consider both the cases (if two symbol repeat then either centre has come or it is not)

aaaa

$(q_s, aaaa, z_0)$

$\downarrow$

(q_s, aaa, az_0)

No centre → (q_s, aa, aaz_0)

Centre has come → (q_1, aa, z_0) × stops

(q_s, aa, aaz_0):

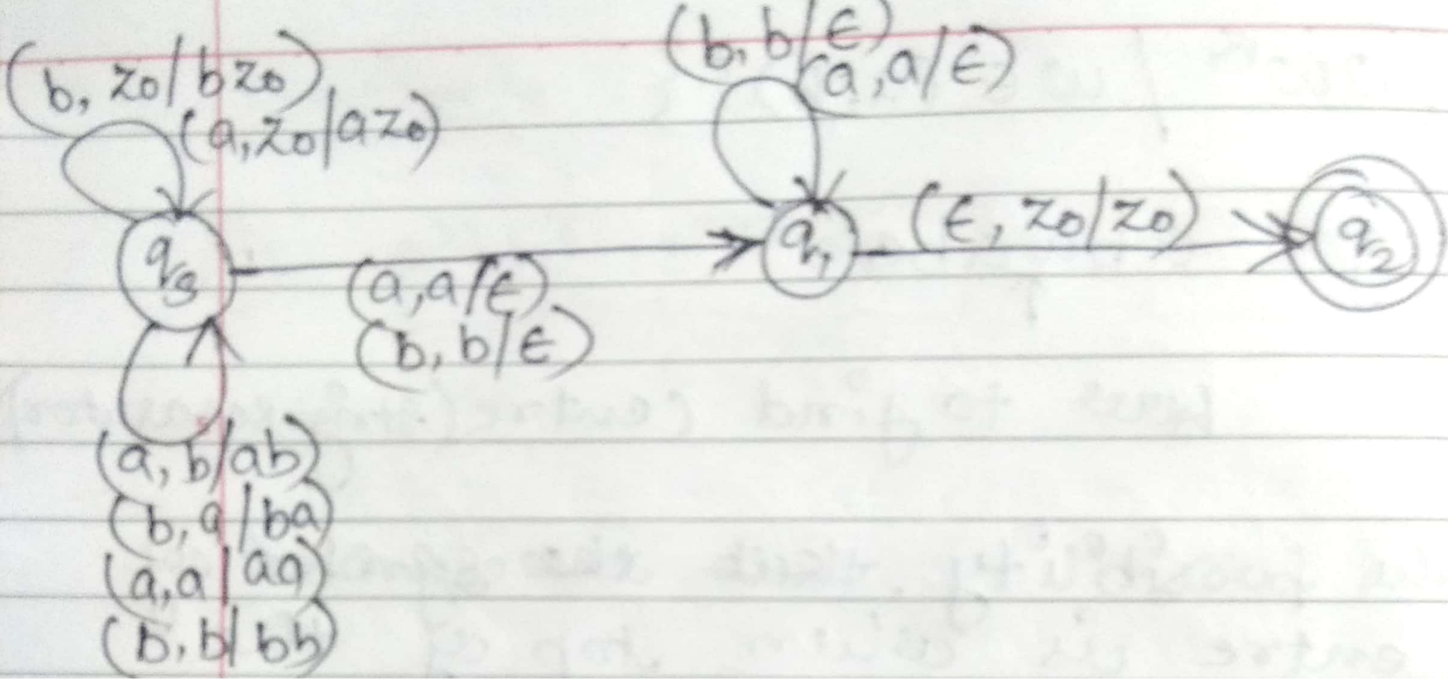
NC → $(q_s, a, aaaz_0)$

centre → (q_1, a, az_0)

$(q_s, a, aaaz_0)$ → $(q_s, \epsilon, aaaaz_0)$ ×

$(q_s, a, aaaz_0)$ → (q_1, ϵ, aaz_0) ×

(q_1, a, az_0) → (q_1, ϵ, z_0) → (q_f, ϵ, z_0) ✓


(b, z0/bz0)
(a, z0/az0)
q0
(a, a/ε)
(b, b/ε)
(a, b/ab)
(b, a/ba)
(a, a/aa)
(b, b/bb)
(b, b/ε)
(a, a/ε)
q1
(ε, z0/z0)
q2

PDA to CFG conversion

Theorem:- If L is a CFL then we can construct a PDA A accepting L by empty stack.

Method ⇒ Let $L = L(G)$, where $G = \{V_n, T, P, S\}$ is a context free Grammer. We construct a PDA A as

$$A = (\{q\}, \Sigma, V_n \cup T, \delta, q, z_0, F)$$

where δ is defined by the following rules:-

$R_1 \Rightarrow \delta(q, \epsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ in } \text{ir}$

$R_2 \Rightarrow \delta(q, a, a) = \{(q, \epsilon)\}$ for every a i

Q. CFG → PDA

$S \to aSa$

$S \to bSb$

$S \to c$

Sol^n **Steps :-** To convert CFG to PDA

① Write ~~production rules~~ transition rules

② Pop rules.

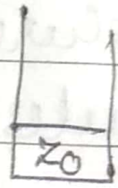
① $\delta(q_0, \epsilon, z_0) = (q_0, z_0)$

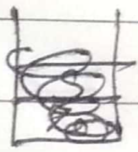
② $\delta(q_0, \epsilon, S) = (q_0, aSa)$

③ $\delta(q_0, \epsilon, S) = (q_0, bSb)$

④ $\delta(q_0, \epsilon, S) = (q_0, c)$

~~Production~~ transition rules.





⑤ $\delta(q_0, a, a) = (q_1, \epsilon)$

⑥ $\delta(q_1, b, b) = (q_2, \epsilon)$

⑦ $\delta(q_2, c, c) = (q_3, \epsilon)$

Pop rules.

Transition Table

Sr No	State	unread i/p	top of stack	transition
1	q_0	abbcbba	z_0	1
2	q_0	abbcbba	S	1
3	q_0	~~a~~bbcbba	~~a~~Sa	2
4	q_1	bbcbba	Sa	5
5	q_0	~~b~~bcbba	~~b~~Sba	3
6	q_2	bcbba	Sba	6
7	q_0	~~b~~cbba	~~b~~Sbba	3
8	q_2	cbba	Sbba	6
9	q_0	~~c~~bba	~~c~~bba	4
10	q_3	~~b~~ba	~~b~~ba	7
11	q_2	~~b~~a	~~b~~a	6
12	q_2	~~a~~	~~a~~	~~7~~ 6
13	$q_{..}$	z_0	z_0	5

Q Construct a PDA A which is equivalent to the following given CFG

$S \to 0CC$
$C \to 0S$
$C \to 1S$
$C \to 0$

Test whether 010000 is accepted PDA A

Sol^n PDA is as follow

$A = \{Q, \{0,1\}, \{S, C, 0, 1\}, \delta, q_0, z_0, F\}$

δ is defined by following rules:-

$$\left.\begin{array}{ll} ① & \delta(q_0, \epsilon, z_0) = (q_0, z_0) \\ ② & \delta(q_0, \epsilon, S) = (q_0, 0CC) \\ ③ & \delta(q_0, \epsilon, C) = (q_0, 0S) \\ ④ & \delta(q_0, \epsilon, C) = (q_0, 1S) \\ ⑤ & \delta(q_0, \epsilon, C) = (q_0, 0) \end{array}\right\} \text{transition rules}$$

$$\left.\begin{array}{ll} ⑥ & \delta(q, 0, 0) = (q_1, \epsilon) \\ ⑦ & \delta(q, 1, 1) = (q_2, \epsilon) \end{array}\right\} \text{Pop rules.}$$

Sr No	State	Unread i/p	top of stack	transition No.
1	q_0	010000	z_0	1
2	q_0	010000	S	1
3	q_0	~~0~~10000	~~0~~CC	2
4	q_1	10000	CC	6
5	q_0	~~1~~0000	~~100~~ ~~1~~SC	4
6	q_1	0000	SC	~~6~~7
7	q_0	~~0~~000	~~0~~CCC	2
8	q_1	000	CCC	6
9	q_0	~~0~~00	~~0~~CC	5
10	q_1	00	CC	6

PDA to CFG conversion

If $A = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ is a PDA then CFG is defined as
$G = (V, \Sigma, P, S)$

① Construction of set of Nonterminal

$$V = \{S\} \cup \{[q, z, q'] \mid q, q' \in Q, z \in \Gamma\}$$

② (i) S- Production

$$S \rightarrow [q_0, z_0, q], \; q \in Q$$

(ii) Pop operation

$$\delta(q, a, z) \longrightarrow (q', \Lambda)$$

$$[q, z, q'] \rightarrow a$$

(iii) for Push & no-operation

$$\delta(q, a, z) \longrightarrow (q_1, z_1 z_2 z_3)$$

$$[q, z, q'] \rightarrow a [q_1, z_1, q_2] [q_2, z_2, q_3] [q_3, z_3, q_4]$$

PDA δ to Production rules

examples

① Push & no operations.

① Starting Symbol (S) production

$S \rightarrow (q_0, z_0\ q_0)$

$S \rightarrow (q_0, z_0, q_1)$

② Push Operation

$\delta(q_0, a, z_0) = (q_0, a z_0)$ $\quad [2^2 = 4]$

$[q_0\ z_0\ q_0] \longrightarrow a[q_0\ a\ q_0]\ [q_0, z_0, q_0]$

$[q_0\ z_0\ q_0] \longrightarrow a[q_0\ a\ q_1]\ [q_1, z_0, q_0]$

$[q_0\ z_0\ q_1] \longrightarrow a[q_0\ a\ q_0]\ [q_0, z_0, q_1]$

$[q_0\ z_0\ q_1] \longrightarrow a[q_0\ a\ q_1]\ [q_1, z_0, q_1]$

$\delta(q_0, b, a) \rightarrow (q_1, a)$ $\quad 2^1 = 2$

$[q_0, a\ q_0] = b[q_1, a, q_0]$

$[q_0, a, q_1] = b[q_1, a, q_1]$

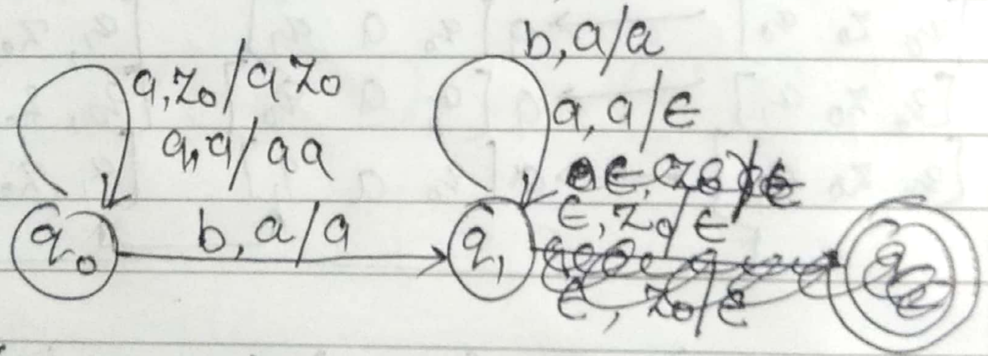
③ Pop operation.

$\delta(q_1, a, a) = (q_1, n)$ $\{2^{\epsilon} = 0\}$

$[q_1, a, q_0] \longrightarrow a$

Construct a PDA accepting $\{a^n b^m a^n \mid m, n \geq 1\}$ by null store and construct CFG accepting same

$L = \{aba, aabbaa, aabaa \ldots \ldots\}$



$\delta(q_0, a, Z_0) \rightarrow (q_0, aZ_0)$.

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

$\delta(q, b, a) \longrightarrow (q_1, a)$

$\delta(q_1, b, a) \longrightarrow (q_1, a)$

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

$\delta(q_1, \epsilon, Z_0) \longrightarrow (q_1, \epsilon)$

③

Production Rules

① S production

$$\boxed{\begin{array}{l} S \rightarrow [q_0, z_0, q_0] \\ S \rightarrow [q_0, z_0, q_1] \end{array}}$$

$\delta(q_0, a, z_0) \longrightarrow (q_0, az_0) \leftarrow$ Push operation

$[q_0, z_0, q_0] \longrightarrow a\,[q_0, a, q_0]\,[q_0, z_0, q_0]$

$[q_0, z_0, q_0] \longrightarrow a\,[q_0, a, q_1]\,[q_0, z_0, q_0]$

$[q_0, z_0, q_1] \longrightarrow a\,[q_0, a, q_0]\,[q_0, z_0, q_1]$

$[q_0, z_0, q_1] \longrightarrow a\,[q_0, a, q_1]\,[q_0, z_0, q_1]$

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

$[q_0, a, q_0] \longrightarrow a\,[q_0, a, q_0]\,[q_0, a, q_0]$

$[q_0, a, q_0] \longrightarrow a\,[q_0, a, q_1]\,[q_0, a, q_0]$

$[q_0, a, q_1] \longrightarrow a\,[q_0, a, q_0]\,[q_0, a, q_1]$

$[q_0, a, q_1] \longrightarrow a\,[q_0, a, q_1]\,[q_1, a, q_1]$

$\delta(q_0, b, a) \longrightarrow (q_1, a) \leftarrow$ (skip) no operation

$[q_0, a, q_0] \longrightarrow b\,[q_1, a, q_0]$

$[q_0, a, q_1] \longrightarrow b\,[q_1, a, q_1]$

$\delta(q_1, b, a) \longrightarrow (q_1, a)$

$[q_1, a, q_0] \longrightarrow b\,[q_1, a, q_0]$

$[q_1, a, q_1] \longrightarrow b\,[q_1, a, q_1]$

$\delta(q_1, a, a) \longrightarrow (q_1, \epsilon)$ — Pop

$\boxed{[q_1, a, q_1] \longrightarrow a}$

$\delta(q_1, \epsilon, z_0) \longrightarrow (q_1, \epsilon)$

$\boxed{[q_1, z_0, q_1] \longrightarrow \epsilon}$

Scanned with CamScanner

Tutorial Questions

(1) Design PDA to accept language $L = \{a^n b^n \mid n \geq 1\}$ accepting by final states / empty stack.

(2) Design PDA to accept language $L = \{a^n b^{2n} \mid n \geq 1\}$ accepting by final states or empty stack.

(3) Design PDA to accept language $L = \{wcw^R \mid w \in (a,b)^*\}$

(4) $L = \{ww^R \mid w \in (a,b)^+\}$
(Non-deterministic PDA)

(5) Construct a PDA which is equivalent to the following given CFG

$$S \rightarrow aSa$$
$$S \rightarrow bSb$$
$$S \rightarrow c$$

(6) Construct a PDA accepting $\{a^n b^m a^n \mid m, n \geq 1\}$ by null store & construct CFG accepting same.