PUSHDOWN AUTOMATA

$$CFG \implies CFL \rightarrow PDA$$

Deterministic PDA	Non-Deterministic PDA
less powerful	more powerfull
only for "SOME" CFL we can design	for "ALL" CFL we can design
DPDA "	NDPDA "
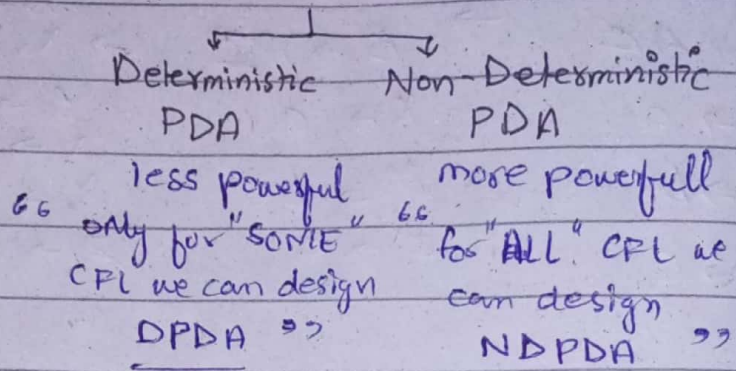
→ Every NDPDA cannot be converted to DPDA.

→ CFL is divided into ⎡→ DCFL → which can be design by DPDA
                      ⎣→ CFL → " " " " NDPDA

→ A PDA is nothing but a Finite Automata with "Stack"

S/P
story → | | | x | y | z | ε | | | → jese hi ε aygaa ut will stop
               input string

Read
header ↑
         | Finite Control | → | | |
         | Unit | | Z₀ |
                                Stack → for memory.
→ yo mohi PA ki
  state diagram

push → pop
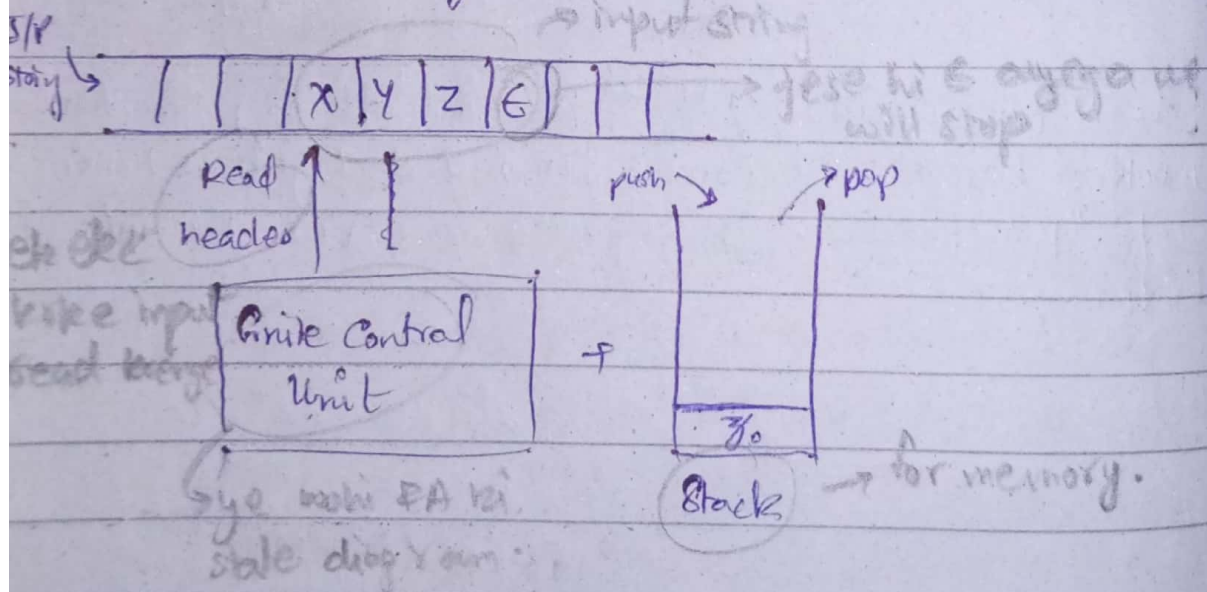
$$PDA = \{Q, \Sigma, q_0, F, z_0, \Gamma, \delta\}$$

Q = Finite set of states.

Σ = Input Alphabet

q_0 = Initial State (only one)

F = Final State (0 or many more)

z_0 = Bottom / Initial Stack Symbol / (last element of stack when we encounter it means stack is empty)

Γ = Stack Alphabet (ye humare string ya input bhi hai's or alphabet bhi hosakte hai. R alphabets ko represent kr sakte hai symbol se hum string)

δ = Transition Function $\Sigma \subseteq \Gamma$, $z_0 \in \Gamma$)

  ↳ Seperate δ for NDPDA & DPDA.

DPDA : $Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow Q \times \Gamma^*$

NDPDA : $Q \times \Sigma \times F \rightarrow 2^{(Q \times \Gamma^*)}$

transition lete waqt teen cheezon ko dekhne hoga.

① Current State ② Input Symbol ③ Top element of Stack.

After transition we move to next state and can perform any three operations on stack ① push ② pop ③ skip (mean donot do anything)

→ Ek baari mein ek hi baar push/pop krsakte hai

→ Basic Operations on Stack :

① Push	② Pop	③ Skip (no push, no pop)
$\dfrac{a \mid b \mid}{}$	$(q_i) \xrightarrow{a,c/\epsilon} (q_j)$	$(q_i) \xrightarrow{a,b/b} (q_j)$
on state q_i if we have input 'a' and top element is z_0 we	ϵ iska likho gye c pop krle except uske reeche	jo phele tha wahi wapes ohega status change nhi hoga stack ka
$(q_i) \xrightarrow{a, z_0/az_0} (q_j)$		
move to q_j with new status of stack as az_0		
$\delta(q_i, a, z_0) = (q_j, az_0)$	$\delta(q_i, a, c) = (q_j, \epsilon)$ jabbhi ϵ aye to pop op. use karta	$\delta(q_i, a, b) = (q_j, b)$

→ hum phele saare "a's" ko push karenge
or phir fir har ek 'a' pop karenge or 'b' generate
hote rhenge so no. of a = no. of b's

$$L = \{a^n b^n, \; n \geq 1\}$$

→ agr a ho or stack par
z_0 ho to a push kardenge

$a, z_0 / a z_0$

$a, a/aa$

$b, a / \epsilon$

q_0 — $b, a / \epsilon$ → q_1

$\epsilon, z_0 / z_0$

→ ye bs ensure karne ke liye hai
ki jab string end hojaye to
stack empty hona chahiye
tabhi string accept
hoga.

q_f

→ input a ho or
stack mein top par
a ho to 'a' push
kardenge

yaha state
isliye change
ki 'b' paa toake
order maintain
rhe. jese hi b
aayega 'a'
ko pop kardenge

→ yaha humne multiple
entri ki q_0 par bike alag
alag transitions hai

q_f $\epsilon, z_0 / \epsilon$

z_0 pop krne
ka matlab string
is accepted.

Dono methods se
string accept karaski
hai

→ Hum har state par scasi transitions show nhi karenge warna
bahut complex hojayega. Bs kisi ek transition ke liye
muliple options nhi denge in DPDA.
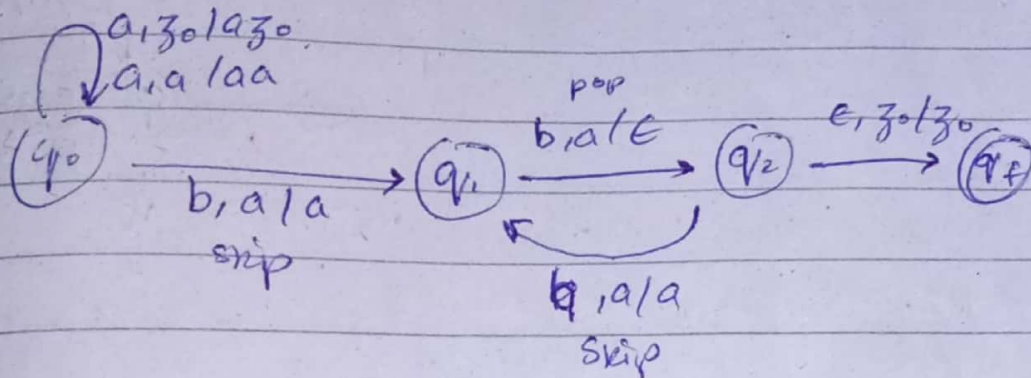
→ No Transition Table.

→ Only Transition ID's.

$$\delta(q_0, a, z_0) = (q_0, a z_0)$$
$$\delta(q_0, a, a) = (q_0, aa)$$
$$\delta(q_0, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, z_0) = (q_f, z_0) \quad \text{or} \quad \delta(q_1, \epsilon, z_0) = (q_f, \epsilon)$$

$L = \{wcw\}$ ✗

$\{a^n b^n c^n, n \geq 1\}$ ✗

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

phele saare a's ko push. Phir ek 'b'
ko skip or and b k liye 'a' ko pop.

$a, z_0 / a z_0$
$a, a / a a$

$b, a / a$
skip

$q_0 \xrightarrow{} q_1 \xrightarrow{\text{pop}\; b, a / \epsilon} q_2 \xrightarrow{\epsilon, z_0 / z_0} q_f$

$b, a / a$
skip

$L = \{w \textcircled{c} w^R, w \in \{a, b\}^*\}$ [palindrome]

$w = aaba$ → a seperator which tells
$w^R = abaa$ (reverse   that it has ended

→ phele poori w ko push kadenge or jab c ayega to
state change karke hum top most element ko apne
current input se match karenge agr wo same hai to
pop.
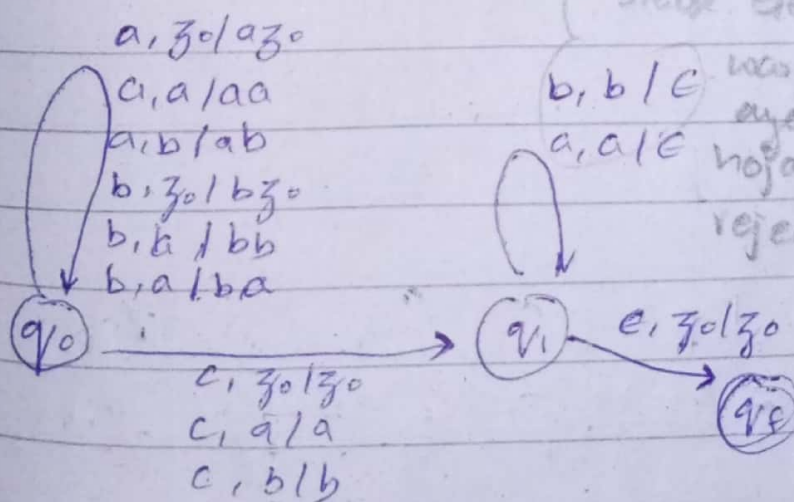
agr jo input hai wohi
[stack element hai to pop
wrna ishe dawa ruch
aya to system halt
hojayega or string
reject

$a, z_0 / a z_0$
$a, a / a a$
$a, b / a b$
$b, z_0 / b z_0$
$b, a / b b$
$b, a / b a$

$b, b / \epsilon$
$a, a / \epsilon$

$q_0 \xrightarrow{} q_1 \xrightarrow{\epsilon, z_0 / z_0} q_f$

$c, z_0 / z_0$
$c, a / a$
$c, b / b$
skip
bs transition change
karhe hai

$ab\overset{c}{a}b\epsilon$

$aa\overset{b}{b}ba$

babbaabaa   a, z0|a...

aababaabba

b
b

$$L = \{ /w/_{a=b} \;,\; w \in \{a,b\}^* \}$$

a, z0|az0
a, a|aa
b, a|ε
b,b|bb
a,b|ε

a, z0|az0
a,a|aa
b,a|ε
b,z0|bz0
b,b|bb
a,b|ε

q_0 → b, z0|bz0 → q_1 ⟹ q_0 ε, z0|z0 q_f

ε, z0|z0 → q_f

aabbaababb

bbababbaaa

Q: $L = \{ a^m b^n \;,\; m \geq n \}$

4, z0|4z0
1,1|11
0,0|ε

0,1|ε
ε,1|ε
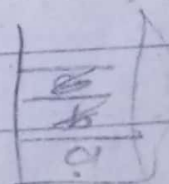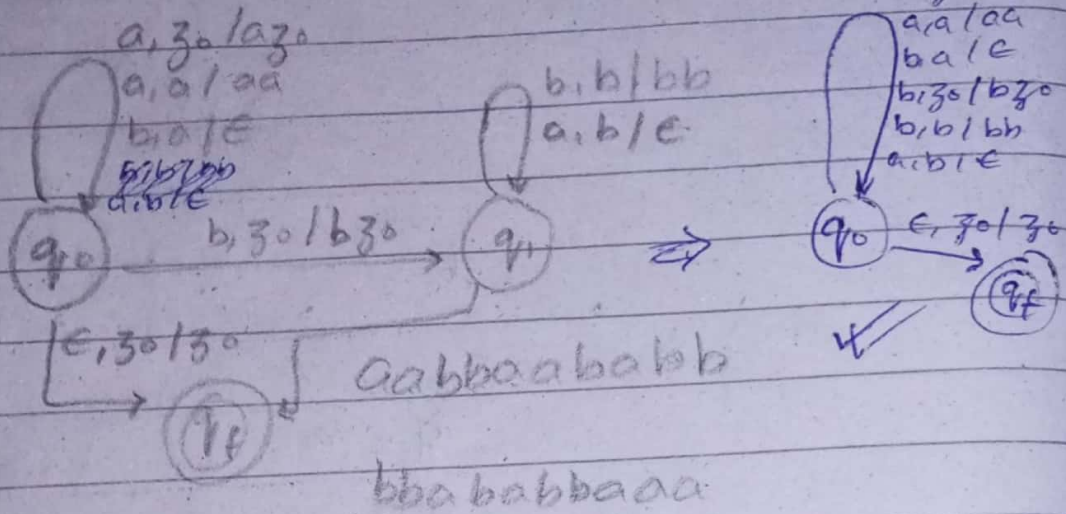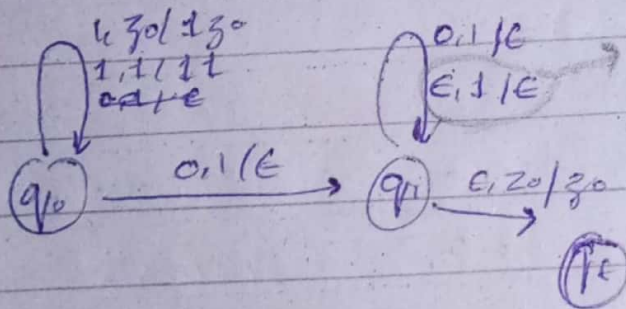
iska matlab string end
hogai lekin abhi bhi
1 mojood hai matlab
1's zero se zyada hai.

q_0 → 0,1|ε → q_1 ε, z0|z0 → q_f

---

i/p   pop   push
↓   ↓   ↓
q_1 → a, b → c → q_2

b → c
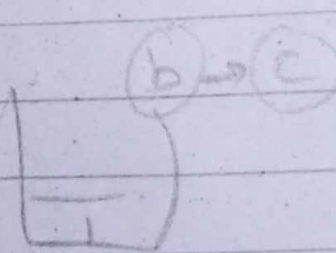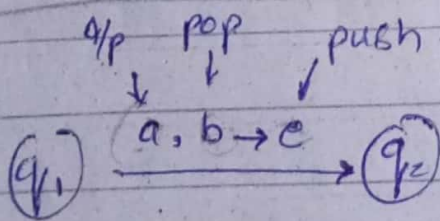
$\{1, 0\}$
↑
$\{a, b\}$

Push	Pop	Skip	Replace
q_1 —a, ε → c→ q_2	q_1 —a,b → ε→ q_2	q_1 —a, ε → ε→ q_2	q_1 —a,b→c→ q_2
pushing c when a is i/p	poping b when a is input	neither pop nor push	replacing b with c

"NON-DETERMINISTIC"

$a,b \to c$ → q_2

q_1

$a,b \to c$ → q_2

$q_1 \xrightarrow{\lambda, b \to c} q_2$

λ - transition

$$L = \{a^n b^n \; ; \; n \geq 0\}$$

$a, \lambda \to a$ $b, a \to \lambda$

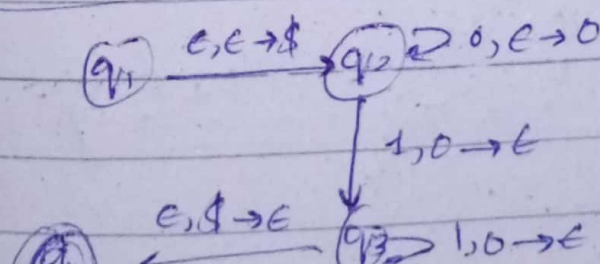$q_0 \xrightarrow{\lambda, \lambda \to \lambda} q_1 \xrightarrow{b, a \to \lambda} q_2 \xrightarrow{\lambda, \$ \to \$} q_3$
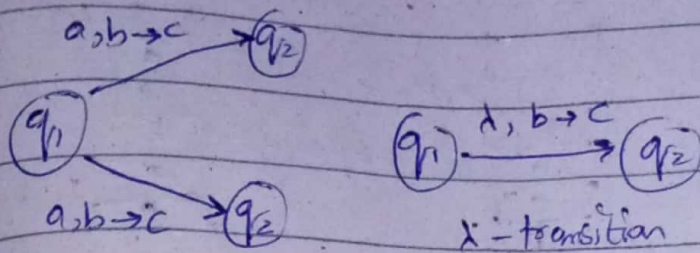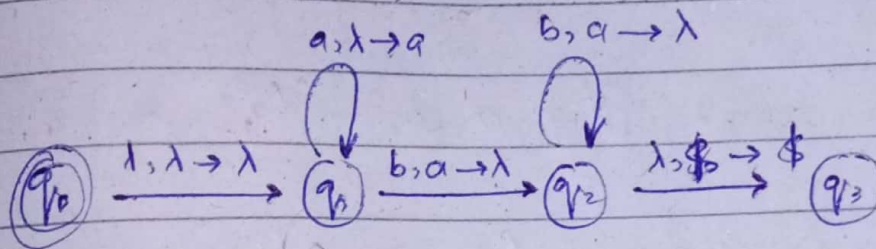
→ A string is accepted if there is a computation such that
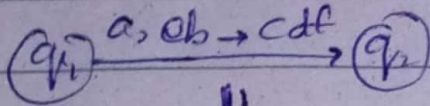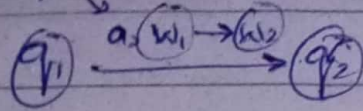① All inputs are consumed && ② The last state is an accepting state.

SIPSER VERSION

Input:	0			1			ϵ		
Stack:	0	\$	ϵ	0	\$	ϵ	0	\$	ϵ
q_1			$\{(q_2,0)\}$			$\{(q_3,\epsilon)\}$			$\{(q_2,\$)\}$
q_2			$\{(q_2,0)\}$	$\{(q_3,\epsilon)\}$					
q_3				$\{(q_3,\epsilon)\}$				$\{(q_4,\epsilon)\}$	
q_4									

$q_1 \xrightarrow{\epsilon, \epsilon \to \$} q_2 \circlearrowright 0, \epsilon \to 0$

$\downarrow 1, 0 \to \epsilon$

$q_4 \xleftarrow{\epsilon, \$ \to \epsilon} q_3 \circlearrowright 1, 0 \to \epsilon$

→ Pushing & Poping Strings:

ⁱ/ₚ   pop   push string

$(q_1) \xrightarrow{a, w_1 \to w_2} (q_2)$

$(q_1) \xrightarrow{a, cb \to cdf} (q_2)$

||

pop   push

$(q_1) \xrightarrow{a, e \to \lambda} \bigcirc \xrightarrow{\lambda, b \to \lambda} \bigcirc \xrightarrow{\lambda, \lambda \to f} \bigcirc \xrightarrow{\lambda, \lambda \to d} \bigcirc \xrightarrow{\lambda \to \lambda \to c} \bigcirc$
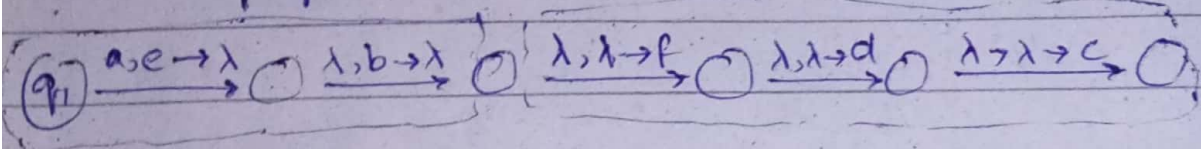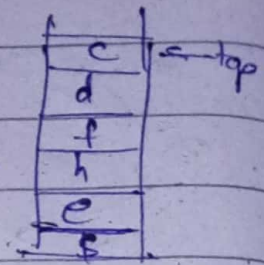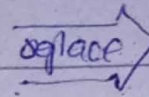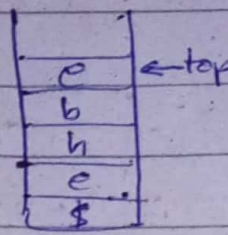
Transition function: $\delta(q_1, a, w_1) = \{(q_2, w_2)\}$

Instantaneous Description: (q, u, s)

current state → | remaining input ↑ | ← Current stack contents
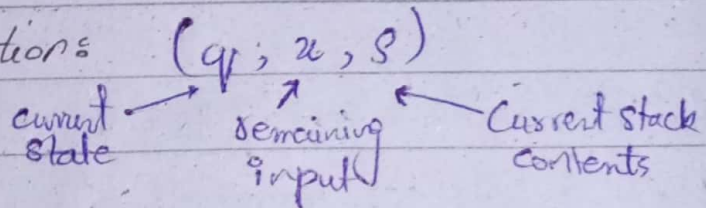
Example:

ⁱ/ₚ : aaabbb
        ↑

$(q_1, bbb, coa\$)$
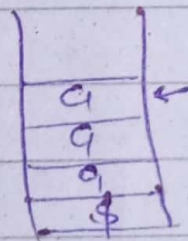
See Slide # 7 (kleek #14 PDA.pdf)

____

$L = \{ w \in \{0,1\}^* , w = w^R$ & length of $|w| = odd \}$

$0, \epsilon \to \epsilon$

$(q_1) \xrightarrow{e, \epsilon \to \$} (q_2) \xrightarrow{1, \epsilon \to \epsilon} (q_3) \xrightarrow{\epsilon, \$ \to \epsilon} (q_4)$

$0, \epsilon \to 0$ $0, 0 \to \epsilon$
$1, \epsilon \to 1$ $1, 1 \to$

$S \rightarrow aSb$

$S \rightarrow ab$ jo bhi

① $\delta(q, \epsilon, \epsilon) = (q, z_0)$ ⟶ Sabse phele z_0 push karege
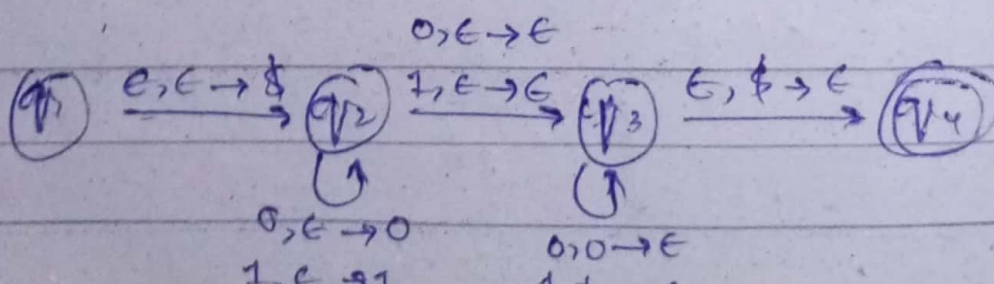
② $\delta(q, \epsilon, \epsilon) = (q, S)$ ⟹ Phir Start symbol push hoga

③ $\delta(q, \epsilon, S) = (q, aSb)$ ⟹ agr top of Stack mein S hai to hum poori Right hand side ko push kadenge.

④ $\delta(q, \epsilon, S) = (q, ab)$

Rule: Ssase non-terminals ke liye ychi karenge ke agr wo top of stack mein encounter ho to unki production ki right hand side push kadenge.

"aSb" ko ek saath push nhi kar sakte is waja se multiple states bnaake push kadenge.

$\bigcirc \xrightarrow{\delta, S \rightarrow aSb} \bigcirc \quad = \quad \bigcirc \xrightarrow{\delta, S \rightarrow b} \bigcirc \xrightarrow{\epsilon, \epsilon \rightarrow S} \bigcirc \xrightarrow{\epsilon, \epsilon \rightarrow a} \bigcirc$

⑤ $\delta(q, a, a) = (q, \epsilon)$ ⟹ jo bhi terminal top of stack mein milega usko pop kadenge

⑥ $\delta(q, b, b) = (q, \epsilon)$ ↗

⑦ $\delta(q, \epsilon, z_0) = (q, z_0)$ ⟶ accept state (ao we seach) final state