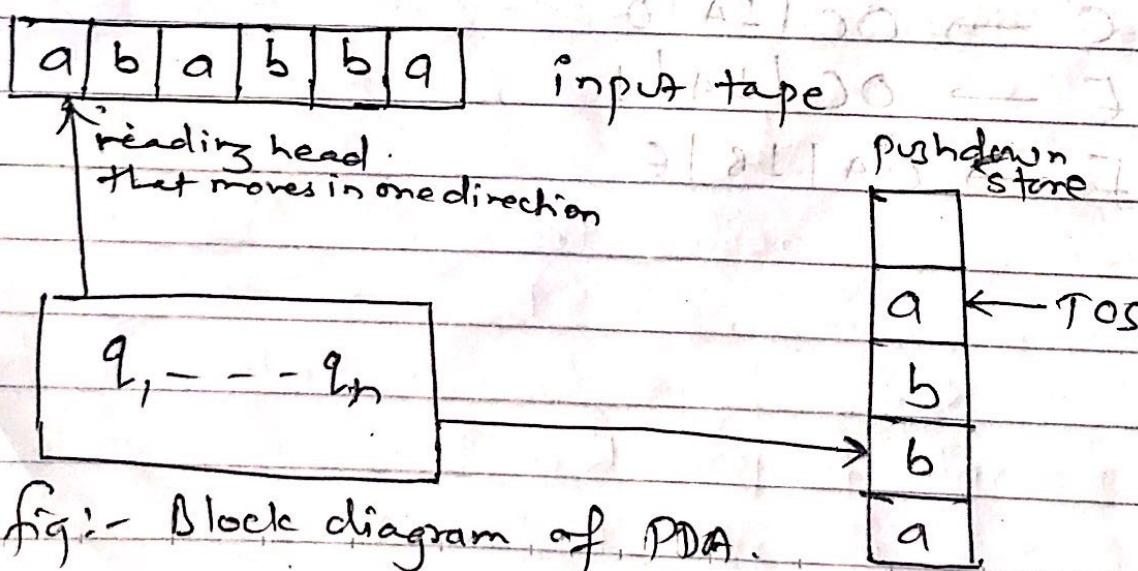


Push Down Automata (PDA)

- ⇒ Using pumping lemma for regular expression, we come to know that there are certain languages like $L = \{a^n b^n : n > 0\}$ does not belong to the class of FA. i.e. such type of languages are not recognized by FA.
- ⇒ This is because, FA is a memory less machine and to process the language like above, FA must have to remember the total no. of a's in order to process the same number of b's.
- ⇒ To process such type of languages, we have machine which can remember the number of a's such that it can match exact number of b's with respect to a's. Such machine is called pushdown Automata (PDA).
- ⇒ In pushdown Automata (PDA), stack is used in the form of memory. Since stack is also called pushdown store, it is called pushdown automata.



- PDA is an abstract (^{idealized}) machine determined by following three things.
 - Input tape
 - finite control
 - stack.
- Each moves of the machine is determined by three things.
 - The current state
 - Next input symbol
 - Symbol on the top of stack.
- The move consists of;
 - changing state / staying on same state.
 - Replacing the stack top by string of zero or more symbols.
- Popping the top symbol off the stack means replacing it by ϵ .
- Pushing Y on the stack means replacing stack's top say X , by YX . The single move of the machine contains only one stack operation either push or pop.

formal Definition :-

A PDA can be defined by seven tuples.

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

Where

Φ = finite set of states

Σ = finite set of input symbols / alphabets.

δ = transition function

q_0 = initial state.

F = finite set of final states.

Z = symbol of stack / finite set of stack alphabet.

Z_0 = initial stack symbol, $Z_0 \in T$.

Moves of PDA :-

- Each instant provided by the transition function of a PDA is called its move. The move of a PDA is defined as:-

$$\delta(q, x, y) \longrightarrow (p, z)$$

Where,

q — current state

x — symbol of the input tape under reading head.

y — top of stack

p — next state

z — new top of stack that replaces y .

- This move tells that, whenever we are in state q , if we read x from the input tape and y is the current top of stack then state is changed to p and top of stack y is replaced by z , which becomes new top of stack.

at 10 $\delta(q, a, b) \rightarrow (p, a)$
 replace los ~~b~~ by a

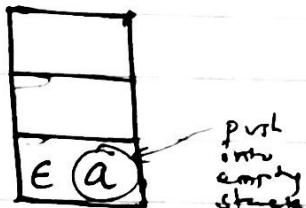
Data

Page

To push an input symbol in the empty stack, we use the transition as;

$$\delta(q, a, \epsilon) \longrightarrow (p, a)$$

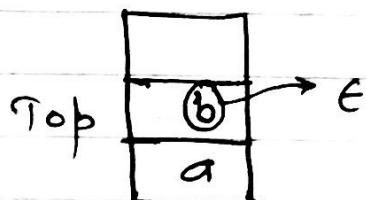
Pushes 'a' to the stack



To pop an input symbol from stack we use transition as;

$$\delta(q, a, b) \longrightarrow (p, e)$$

Pops 'b' from the stack.



Q Design a PDA for the language.

$$L = \{a^n b^n : n > 0\}$$

Sol'n: Let $M = (\Phi, \Sigma, \delta, q_0, F, \Gamma, z_0)$ be a PDA.

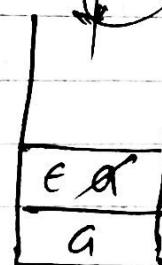
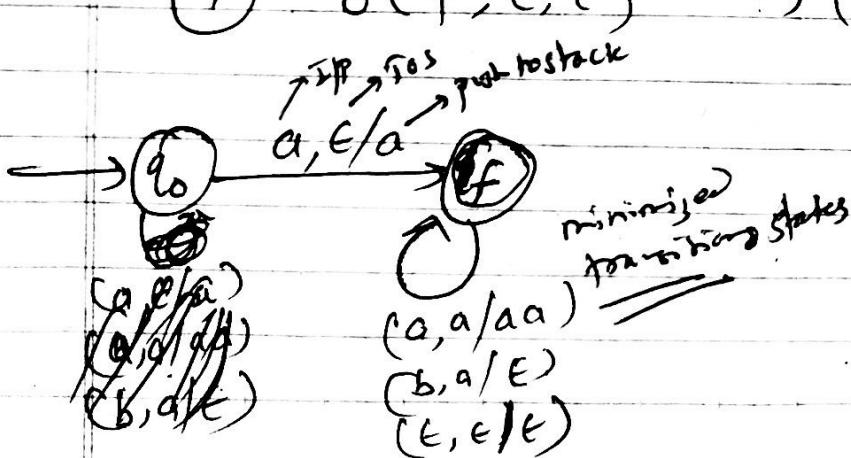
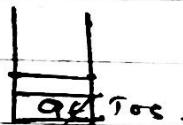
S can be defined as:

$$\textcircled{1} \quad \delta(q_0, a, \epsilon) \longrightarrow (f, a)$$

$$\textcircled{2} \quad \delta(f, a, a) \longrightarrow (f, aa)$$

$$\textcircled{3} \quad \delta(f, b, a) \rightarrow (f, e)$$

$$(4) \quad \delta(f, \epsilon, \epsilon) \rightarrow (f, \epsilon)$$



for $aaaabb$ string

state	unread i/p	stack content	Transition used
q_0	$aaaabb$	ϵ	-
f	$aabb$	a	①
f	$aabb$	aa	②
f	abb	aaa	②
f	bb	aa	③
f	b	a	③
f	ϵ	ϵ	③
f	-	ϵ	④

Since, the stack becomes empty after the termination of string, the language is accepted by PDA.

OR

$$\textcircled{1} \quad \delta(q_0, \epsilon, \epsilon) \rightarrow (q_1, z_0)$$

This step initializes the stack to indicate the bottom of stack.

$$\textcircled{2} \quad \delta(q_1, a, z_0) \rightarrow (q_1, az_0)$$

$$\textcircled{3} \quad \delta(q_1, a, a) \rightarrow (q_1, aa)$$

$$\textcircled{4} \quad \delta(q_1, b, a) \rightarrow (q_2, \epsilon)$$

$$\textcircled{5} \quad \delta(q_2, \epsilon, z_0) \rightarrow (q_2, \epsilon)$$

graphically



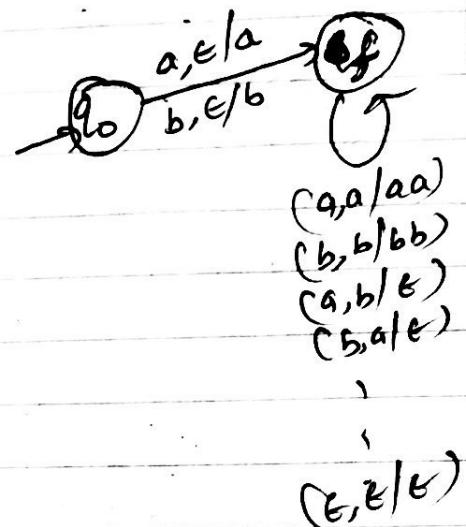
$(a, z_0/az_0)$ $(\epsilon, a/aa)$ $(b, a/\epsilon)$

Design a PDA for language

$L = \{ \text{consisting all the strings of } a\text{'s and } b\text{'s}$
 $\text{over } \Sigma = \{a, b\}, \text{ having equal no. of}$
 $a\text{'s \& } b\text{'s} \}$.

Soln:

- ① $\delta(q_0, a, \epsilon) \rightarrow (f, a)$
- ② $\delta(q_0, b, \epsilon) \rightarrow (f, b)$
- ③ $\delta(f, a, a) \rightarrow (f, aa)$
- ④ $\delta(f, b, b) \rightarrow (f, bb)$
- ⑤ $\delta(f, a, b) \rightarrow (f, \epsilon)$
- ⑥ $\delta(f, b, a) \rightarrow (f, \epsilon)$
- ⑦ $\delta(f, a, \epsilon) \rightarrow (f, a)$
- ⑧ $\delta(f, b, \epsilon) \rightarrow (f, b)$
- ⑨ $\delta(f, \epsilon, \epsilon) \rightarrow (f, \epsilon)$

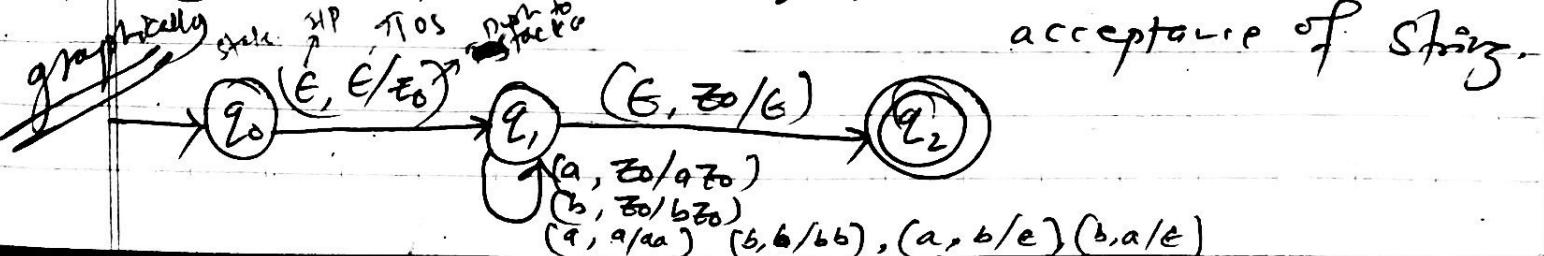


OR

- ① $\delta(q_0, \epsilon, \epsilon) \rightarrow (q_1, z_0)$ // initialize the stack to indicate the bottom of stack.

- ② $\delta(q_1, a, z_0) \rightarrow (q_1, az_0)$
- ③ $\delta(q_1, b, z_0) \rightarrow (q_1, bz_0)$
- ④ $\delta(q_1, a, a) \rightarrow (q_1, aa)$
- ⑤ $\delta(q_1, b, b) \rightarrow (q_1, bb)$
- ⑥ $\delta(q_1, a, b) \rightarrow (q_1, \epsilon)$
- ⑦ $\delta(q_1, b, a) \rightarrow (q_1, \epsilon)$
- ⑧ $\delta(q_1, \epsilon, z_0) \rightarrow (q_2, \epsilon)$ // indicate the acceptance of string.

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



trace for $w = aabbbaab$ also
 $w = ababa$

~~$L = \{ww^R \mid w \in \{a, b\}^*\}$~~

rewrite.

$a \in a$

$b \in b$

$ab \in ba$

$ba \in ab$

left string right string

$$\begin{array}{ll} eq & aa \\ = & bb \\ & ab \\ & ba \\ w & ab \\ & ba \\ w^R & ab \end{array}$$

we need a separator. such that
 we can identify the first half & second half

~~So it's~~ So consists of:

A set to push w on the stack.

$\delta(q_0, a, a) \rightarrow f(q_0, \overset{\text{TOS}}{aa}) ? \quad a \text{ on top of } a$

$\delta(q_0, b, b) \rightarrow f(q_0, \overset{\text{TOS}}{bb}) ? \quad b \text{ on top of } b$

$\delta(q_0, a, b) \rightarrow f(q_0, \overset{\text{TOS}}{ab}) ? \quad a \text{ on top of } b$

$\delta(q_0, b, a) \rightarrow f(q_0, \overset{\text{TOS}}{ba}) ?$

$\delta(q_0, a, z_0) \rightarrow f(q_0, \overset{\text{TOS}}{az_0}) ?$

$\delta(q_0, b, z_0) \rightarrow f(q_0, \overset{\text{TOS}}{bz_0}) ?$

\Rightarrow A set to guess the middle of stack when PDA switches from state q_0 to q_1 , without reading anything.

$$\textcircled{7} \quad \delta(q_0, \epsilon, a) \rightarrow \{(\bar{z}_1, a)\}$$

$$\textcircled{8} \quad \delta(q_0, \epsilon, b) \rightarrow \{(\bar{z}_1, b)\}$$

\Rightarrow A set to match w^L against the contents of stack.

$$\textcircled{9} \quad \delta(z_1, a, a) \rightarrow \{(\bar{z}_1, \epsilon)\}$$

if
TOS & Top
match then
pop TOS

$$\textcircled{10} \quad \delta(z_1, b, b) \rightarrow \{(\bar{z}_1, \epsilon)\}$$

\Rightarrow and finally.

$$\textcircled{11} \quad \delta(z_f, \epsilon, z_0) \rightarrow \{(\bar{z}_f, z_0)\}$$

(12)
also

$$\delta(q_0, \epsilon, \epsilon) \rightarrow (z_0, z_0) \quad \cancel{\times}$$

Instantaneous Description (ID) of PDA:-

⇒ Any configuration of a PDA ^{at a particular instant} can be described by a triplet (q, w, z)

Where

q - is the ^{present} state of PDA.

w - is the remaining input (yet to be read by ^{Diagram})

z - is the stack contents.

⇒ Such a description by triplet is called ID of PDA.

Let $P = (Q, \Sigma, \Delta, q_0, F, I, Z_0)$ be a PDA

Then we define a relation \vdash , "yields" as

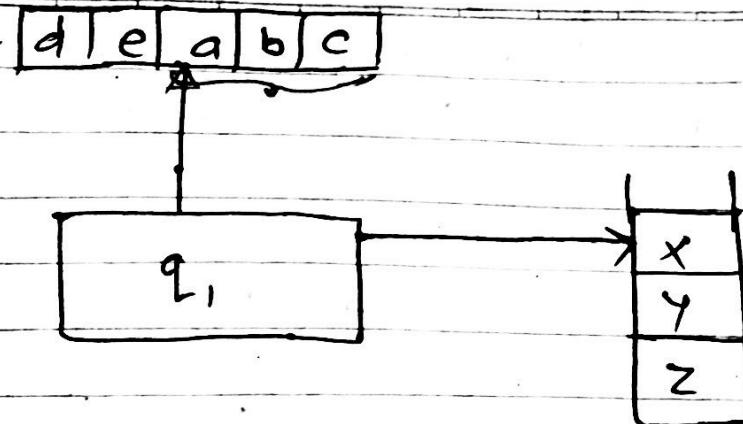
$$(q, aw, za) \vdash (P, w, \beta z)$$

from P ,
configuration P will reach to this configuration
in only one move.

- read 'a' from I/p tape, current TOS is γ after one move PDA reaches to next configuration.

$\delta(q, a, \gamma)$ contains (P, β) .

This moves reflects the idea that, by consuming "a" from the input tape, and replacing γ on the TOS by β , we can go from state q to state P .



(q_1, abc, xyz) is the instantaneous description of above figure.

e.g:-

ID of PDA

\Rightarrow for the PDA described earlier accepting language of equal a's & b's the accepting sequence of ID's for string abba can be shown as;

$$(q_0, abba, \epsilon) \xleftarrow{} (q_1, abba, z_0)$$

$$\xleftarrow{} (q_1, bba, az_0)$$

$$\xleftarrow{} (q_1, ba, z_0)$$

$$\xleftarrow{} (q_1, a, bz_0)$$

$$\xleftarrow{} (q_1, \epsilon, z_0)$$

$$\xleftarrow{} (q_2, \epsilon, \epsilon) \text{ Accepted}$$

Q Design a PDA accepting language
 $L = \{ w c w^R \mid w \in (0+1)^*\}$

- Soln.
- ① $\delta(q_0, \epsilon, \epsilon) \rightarrow (q_0, z_0)$
 - ② $\delta(q_0, 0, z_0) \rightarrow (q_0, 0z_0)$
 - ③ $\delta(q_0, 1, z_0) \rightarrow (q_0, 1z_0)$
 - ④ $\delta(q_0, 0, 0) \rightarrow (q_0, 00)$
 - ⑤ $\delta(q_0, 1, 1) \rightarrow (q_0, 11)$
 - ⑥ $\delta(q_0, 0, 1) \rightarrow (q_0, 01)$
 - ⑦ $\delta(q_0, 1, 0) \rightarrow (q_0, 10)$
 - ⑧ $\delta(q_0, C, 0) \rightarrow (q_1, 0)$ only state change
no charge to S

Charge the state when center is reached

- ⑨ $\delta(q_0, C, 1) \rightarrow (q_1, 1)$ only state change
no charge to S
- ⑩ $\delta(q_0, C, z_0) \rightarrow (q_1, z_0)$ case only one shift
- ⑪ $\delta(q_1, 0, 0) \rightarrow (q_1, \epsilon)$
- ⑫ $\delta(q_1, 1, 1) \rightarrow (q_1, \epsilon)$
- ⑬ $\delta(q_1, \epsilon, z_0) \rightarrow (q_f, \epsilon)$

* Shifting process for $w =$

<u>eg</u>	<u>w = aabbcc</u>	<u>unread l/p</u>	<u>stack contents</u>	<u>transition</u>
<u>state</u>	<u>aabbcc</u>			
q	aabbcc	c		
f	a b b b c	a		(1)
f	b b b c	aa		(2)
f	b b c	ba		(3)
f	b c	a		(4)
f	c	b		(5)
f	-	c		
f	-	c		(6)

Language of PDA:-

⇒ We can define acceptance of any string by a PDA in 2 ways.

(1) Acceptance by final state :-

Given a PDA M , the language accepted by final state $L(M)$ is

$$\{ w \mid (q_0, w, z_0) \xrightarrow{*} (P, \epsilon, r) \}$$

where,

$P \in F$ and $r \in T^*$

Date: _____ Page: _____

2) Acceptance by empty stack: -

- Given a PDA M , the language accepted by empty stack, $L(M)$ is

$$\{w \mid (q, w, z_0) \xrightarrow{*} (p, \epsilon, \epsilon)\}$$

where,

$$p \in Q$$

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

$$\Rightarrow \begin{array}{lll} w = ab & w = a b b b & w = a b a \\ w^* = ba & w^* = b b b a & w^* = a b a \\ ww^* = abba & ww^* = a b b b b b b a & ww^* = a b a a b a \end{array}$$

we don't have middle marker //
we cannot count the length.

one way:

abbba

middle

part
already
reached

=

a b b b b b a

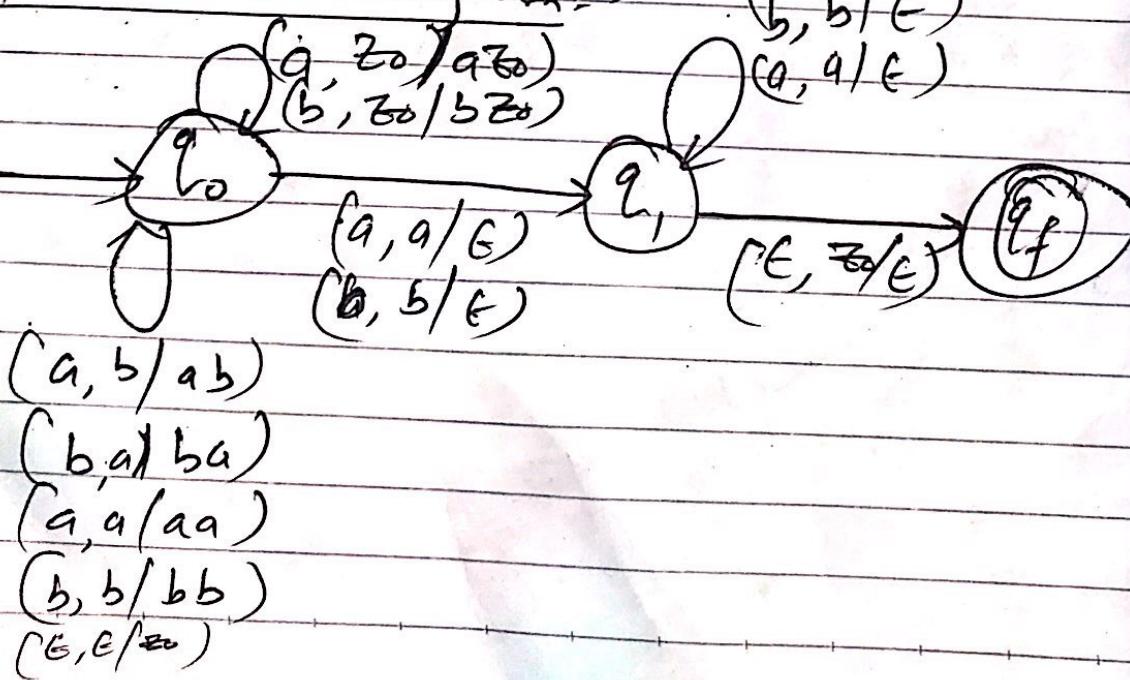
we cannot
guess
middle part
has reached.

so there can be
more middle parts

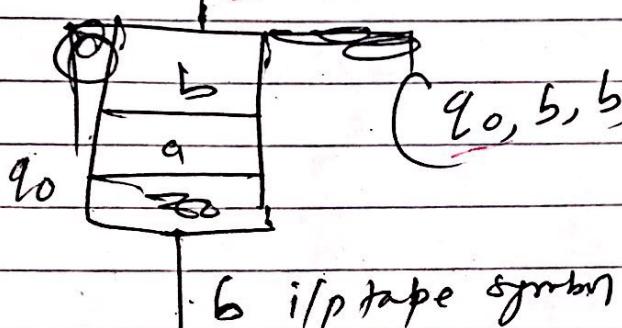
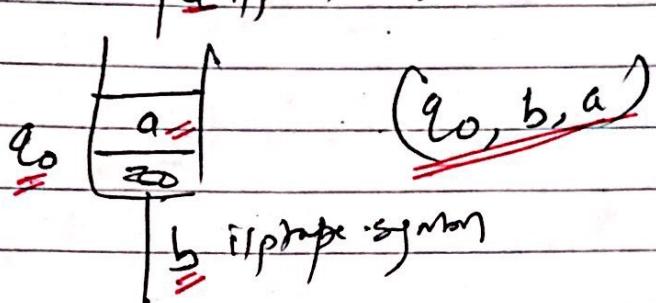
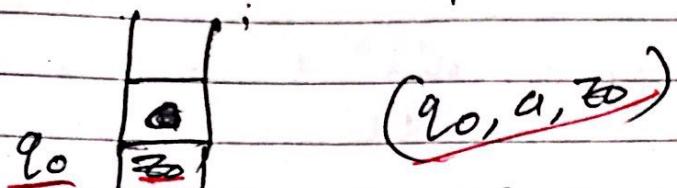
abagba

- we have to identify correct middle part

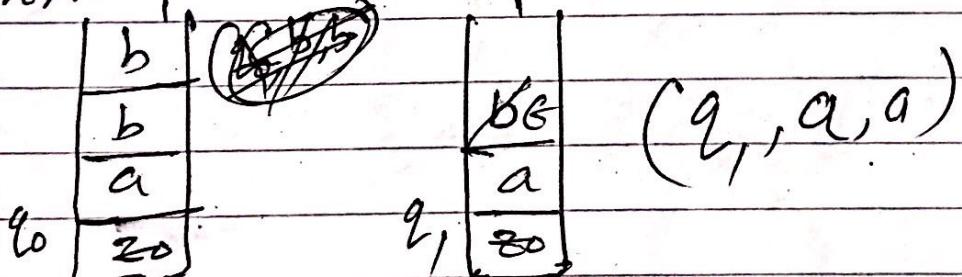
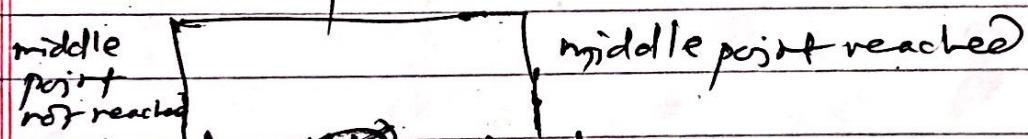
Transition Diagram:



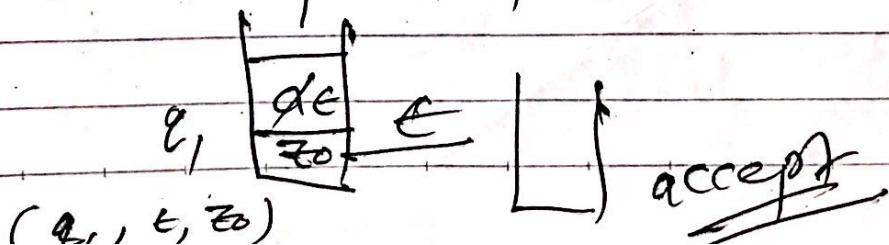
eg $w = \underline{ab}ba$



- npda
two transitions =
- ① middle point not reached $(b, b/bb)$
 - ② middle point reached $(b, b/b)$



dead



Equivalence of CFG & PDA :-

⇒ CFG to PDA conversion :-

- ⇒ To convert a given CFG to its equivalent PDA, it is needed to convert all the production rules of the given CFG into their equivalent transition functions.
- ⇒ Non-terminals of the given CFG are pushed to the stack and terminal symbols are popped from the stack.
- ⇒ We can define pushdown automata for the CFG with two steps P and q, where P being start state.
 - ⇒ Here, idea is that, the stack symbol initially is supposed to be ϵ and PDA starts with state P and on reading ϵ symbol, insert start symbol 'S' of CFG into stack.
 - ⇒ PDA changes the state to q. Then all transition occur in state q.

i.e. PDA can be defined as,

$$M = (\{P, q\}, T, VUT, \delta, P, \{q\}),$$

Stack top is ϵ ,
then δ can be defined as,

$$\delta(P, \epsilon, \epsilon) = \{(\bar{q}, S) \mid S \text{ is start symbol of grammar}\}$$

$$\delta(q, \epsilon, A) = \{(\bar{q}, \alpha) \mid A \rightarrow \alpha \text{ is a production P of G}\}$$

$$\delta(q, a, a) = \{(\bar{q}, \epsilon)\} \text{ for all } a \in \Sigma.$$

Eg:- Convert the following CFG to its equivalent PDA.

$$\begin{aligned} S &\rightarrow OS1 \mid OAA \mid 1BB \\ A &\rightarrow 1A \mid 0 \\ B &\rightarrow 0B \mid 1 \end{aligned}$$

Check whether $w = 0010101$ is accepted by the PDA.

Sol'n:

Let $M = (\Phi, \Sigma, \delta, q_0, F, I)$ where
 $\Phi = \{q, f\}$
 $\Sigma = \{0, 1\}$

δ consists of

- ① $\delta(q, \epsilon, \epsilon) \rightarrow (f, s)$
- ② $\delta(f, \epsilon, s) \rightarrow (f, OS1), (f, OAA)$
- ③ $\delta(f, \epsilon, A) \rightarrow (f, 1A), (f, 0)$
- ④ $\delta(f, \epsilon, B) \rightarrow (f, 0B), (f, 1)$
- ⑤ $\delta(f, 0, 0) \rightarrow (f, \epsilon)$
- ⑥ $\delta(f, 1, 1) \rightarrow (f, \epsilon)$
- ⑦ $\delta(f, \epsilon, \epsilon) \rightarrow (f, \epsilon)$

$$q_0 = q$$

$$F = f$$

$$\mathcal{L} = \{0, 1, S, A, B\}$$

<u>State</u>	<u>unread i/p</u>	<u>stack content</u>	<u>transition used</u>
q	(0010101	ε	—
f	push S ε 0010101	S	①
f	0010101	0S1	②
f	ε 010101	S L	⑤
f	0010101	0A41	②
f	(1 0101	·441	⑤
f	1 0101	1A41	③
f	(0101	A1	⑥
f	0101	041	③
f	(1 01	1A1	③
f	0 L	A1	⑥
f	0 1	01	③
f	1	L	③
f	ε	ε	⑥
f	—	ε	⑦

Hence accepted

Date. _____

OR $(q, 0010101, \epsilon) \xrightarrow{\quad} (f, 0010101, s)$

 $\vdash (f, 0010101, 0S1)$
 $\vdash (f, 010101, S1)$
 $\vdash (f, 010101, 0AA1)$
 $\vdash (f, 10101, AA1)$
 $\vdash (f, 10101, 1AA1)$
 $\vdash (f, 0101, AA1)$
 $\vdash (f, 0101, 0A1)$
 $\vdash (f, 101, A1)$
 $\vdash (f, 101, 1A1)$
 $\vdash (f, 01, A1)$
 $\vdash (f, 01, 01)$
 $\vdash (f, 1, 1)$
 $\vdash (f, \epsilon, \emptyset)$

~~#~~

Q Consider the CFG,

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

P is

$$E \rightarrow T \mid E + T \quad V = \{E, T, F\}$$

$$T \rightarrow F \mid T * F$$

$$F \rightarrow a \mid (E)$$

Construct a equivalent PDA q to recognize acceptance
of $a + Ca + a$)

Design a PDA for the grammar

$$G = (V, T, P, S) \text{ where}$$

$$V = \{S\}$$

$$T = \{a, b, c\}$$

P consists of

$$S \rightarrow aSa$$

$$S \rightarrow bSb$$

$$S \rightarrow C$$

check whether $w = abbcbba$ is accepted by PDA,

P consists of

$$S \rightarrow aAs / bS / a$$

$$A \rightarrow bB / a$$

$$B \rightarrow aa / b$$

$S \in \Sigma^n$

Let $M = (\Phi, \Sigma, \delta, q_0, F, T)$

where,

$$\Phi = \{q, f\}$$

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

δ consists of :-

$$\textcircled{1} \quad \delta(q, \epsilon, \epsilon) \rightarrow (f, s)$$

$$\textcircled{2} \quad \delta(f, \epsilon, s) \rightarrow (f; qAs), (f; bs), (f, a)$$

$$\textcircled{3} \quad \delta(f, \epsilon, A) \rightarrow (f, bb), (f, a)$$

$$\textcircled{4} \quad \delta(f, \epsilon, B) \rightarrow (f, aa), (f, b)$$

- (5) $\delta(f, a, a) \rightarrow (f, \epsilon)$
 - (6) $\delta(f, b, b) \rightarrow (f, \epsilon)$
 - (7) $\delta(f, \epsilon, \epsilon) \rightarrow (f, \epsilon)$
- ~~*~~

$q_0 = q$ initial state.

$f = f$ final state.

$\Sigma = \{S, A, B, a, b\}$

$w = q^* b bba$

State	Unread ip	Stack after	Trans used
q	qabbba	ϵ	—
f	abbba	S	(1)
f	bbbba	qAS	(2)
f	bbb a	AS	(3)
f	bbb a	BBS	(3)
f	bb a	BS	(2)
f	ba	BS	(4)
f	a	S	(6)
f	a	a	(2)
f	ϵ	ϵ	(5)
f	—	ϵ	(7)

~~*~~

Pumping lemma for CFL:-

Let L be a context free language and $z \in L$. Let $|z|$ be the length of the string. Then, z can be decomposed into sub-strings u, v, w, x and y such that

- ① $|Vx| \neq \emptyset$ i.e. V, x not empty at a time.
- ② $|Vwx| \leq n$
- ③ $UV^k w x^k y \in L$, for all $k \geq 0$

$V \neq \emptyset$ pumped any no. of times belongs to CFL.

Soln:-

$$S \rightarrow AB$$

$$A \rightarrow aB|a$$

$$B \rightarrow bA|b \text{ be the CFG}$$

let $z = ababb$ be the string such that $z \in L$.

According to pumping lemma,

$$u = a$$

$$v = ba$$

$$w = b$$

$$x = \epsilon$$

$$y = b$$

$$S \rightarrow AB$$

$$\rightarrow aB|B$$

$$\rightarrow a\cancel{b}AB$$

$$\rightarrow a\cancel{b}B|B$$

$$\rightarrow abab\cancel{B}$$

$$\rightarrow abab\cancel{b}$$

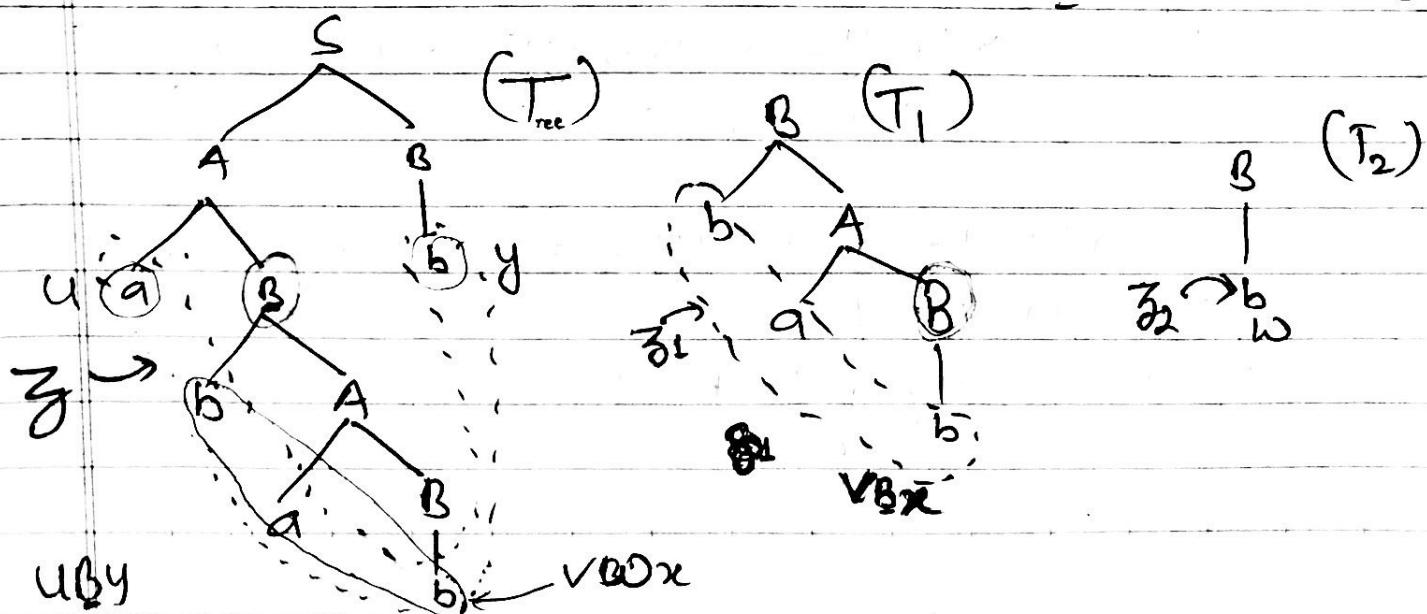
$$\begin{array}{c} A \rightarrow a \\ \cancel{B} \rightarrow bA \end{array}$$

$$B \rightarrow bA$$

$$A \rightarrow aB$$

$$B \rightarrow b$$

$$B \rightarrow b$$



→ Here, T_1 is the proper subtree of T and T_2 is the proper subtree of T_1 . Therefore, we can write,

$$S \longrightarrow U g_1 Y \quad \text{and} \quad Z_1 = V g_2 X \quad \text{and} \\ Z_2 = W$$

Since, g is the yield of S-tree and g_1 and g_2 are the yields of B-tree, we can write production rules of the form

$\zeta \rightarrow \cup \beta y$,

$B \rightarrow V B X$ and

$\beta \rightarrow \omega$

Also, $\sqrt{w_x} |v_{wx}| \geq |w|$

~~✓ TUTTUS~~

1930-1931

卷之四十一

Using above production rules,

$S \rightarrow uBy \rightarrow uwY \rightarrow uvwxy$,
which is true for $k=0$.

Again,

$S \rightarrow UBY \rightarrow UVBXy \rightarrow UVWxy$,
which is true for $k=1$.

$S \rightarrow UVy \rightarrow UVBxy \rightarrow UVVBxy \rightarrow$
 $UV^2\cancel{B}x^2y$, which is true for $k=2$.

∴ we can say that UV^kWx^ky is also true.
 $\therefore UV^kWx^ky \in L$.

\Rightarrow

Show that the language, such that

$$L = \{a^n b^n c^n : n > 0\}$$

$$\text{eg: } w = a a a b b b c c c$$

such variables i.e. 3 variables cannot be dealt by CCB or PDA. no need TMs.

Soln: Let L be the content free language and $g \in L$.

$$\text{let } g = a^p b^p c^p$$

According to pumping lemma, g can be decomposed into u, v, w, x and y such that,

$$u = a^q$$

$$v = a^r, r > 0$$

$$w = a^s$$

$$x = a^t, t > 0$$

$$y = \cancel{a^p - t} \cdot a^{p-(q+r+s+t)} b^p c^p$$

now, $\cancel{q+r+s+t} = 0$

$$\begin{aligned} uv^2wx^2y &= a^q (a^r)^2 a^s (a^t)^2 a^{p-(q+r+s+t)} b^p c^p \\ &= a^{p+r+t} b^p c^p \end{aligned}$$

here, $r > 0$ and $t > 0$, $p+r+t > p$

Therefore, $a^{p+r+t} b^p c^p$ is not of the form $a^p b^p c^p$

i.e. $uv^2wx^2y \notin L$.

Hence, L is not content free \blacksquare proved