



Date:

## Parse Tree: compiler building the parse tree

A parse tree is a tree representation of a derivation. The tree will be labelled tree with the following condition.

- (i) Root of the tree will always be start symbol.
- (ii) All non leaf nodes of the tree will be nonterminals.
- (iii) All leaf nodes of the tree will be terminals.

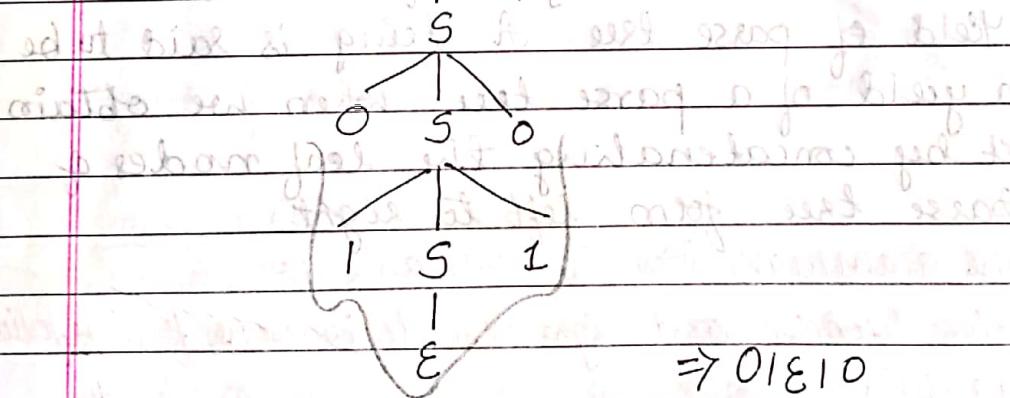
Note:  $\epsilon$  (Epsilon) can be leaf but it has to be the only child to its parent.

If A is a non terminal and  $x_1, x_2, \dots, x_n$  are its children starting from the left then there will be a production.

$A \rightarrow x_1 x_2 \dots x_n$  in the grammar.

Q: Construct a parse tree for the string 0110 in the language of a palindrome grammar.

set  $S \rightarrow 1S1 | 0S0 | 0|1 | \epsilon$   $S \rightarrow 0110$



$\Rightarrow 0110$

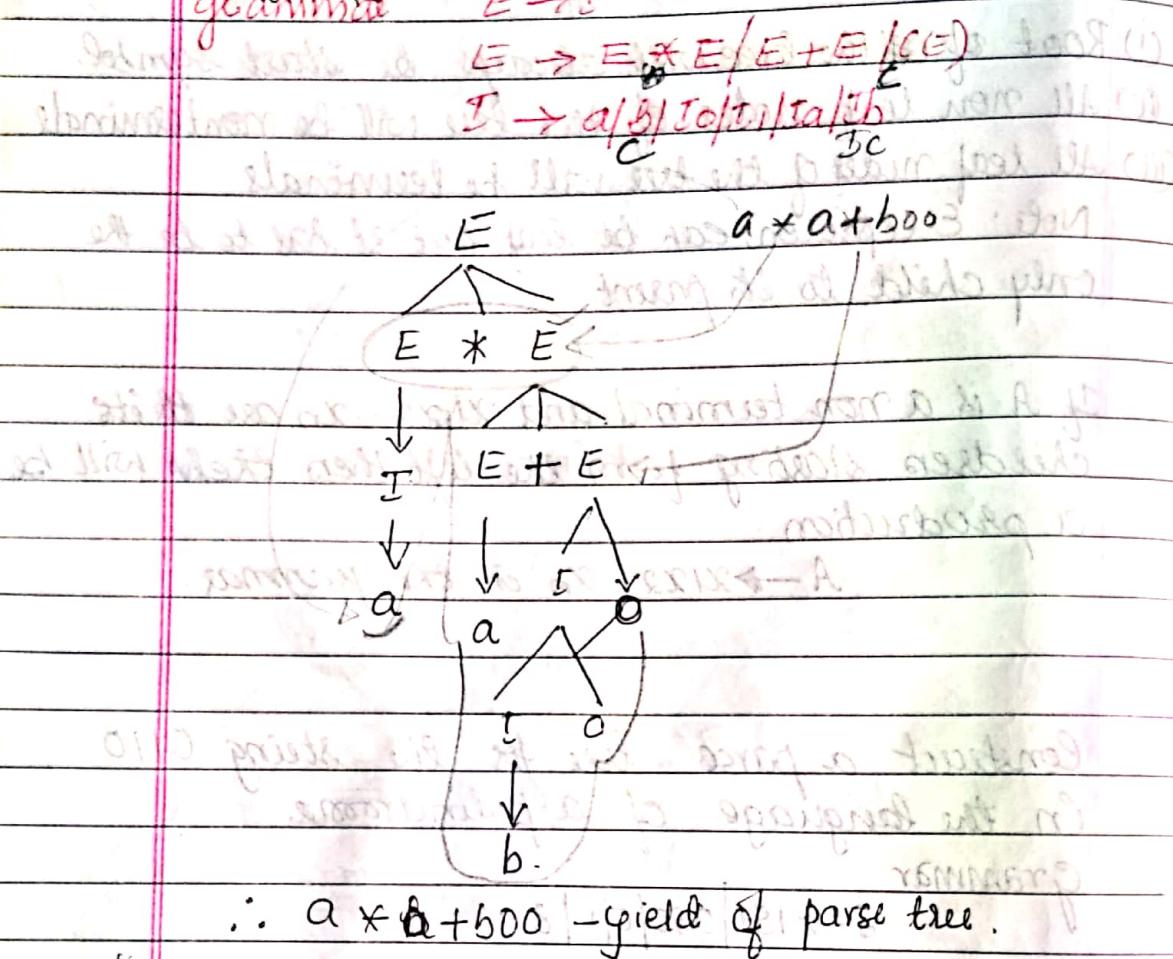
$\Rightarrow 0110 - \text{Yield}$  0110  
Parse tree.

Root vertex: Must be labelled by the start symbol.

Vertex: Labelled by Non-terminal symbols

Leaves: Labelled by terminal symbol or  $\epsilon$ .

Ques: Construct the parse trees for the string  $a * a + b o o$  using grammar using the grammar  $E \rightarrow E * E / E + E / (E) / a / b$



Yield of parse tree: A string is said to be a yield of a parse tree when we obtain it by concatenating the leaf nodes of parse tree from left to right.

5.1.2 The grammar  $S \rightarrow AS / E, A \rightarrow a / ab / ba / bb$ .

Give leftmost and rightmost derivations for the following:

(a)  $aabbba$

$\begin{array}{l} S \xrightarrow{r_1} A\delta \Rightarrow aa\delta \Rightarrow aab\delta \Rightarrow aabb\delta \Rightarrow aabbA\delta \Rightarrow aabbba\delta \Rightarrow aabbba \\ \text{L.M.D.} \end{array}$

$\begin{array}{l} S \xrightarrow{r_2} A\delta \Rightarrow AA\delta \Rightarrow AAb\delta \Rightarrow AAAE \Rightarrow AAb\delta a \Rightarrow Aabbba \Rightarrow aabbba \\ \text{R.M.D.} \end{array}$

# Parse tree



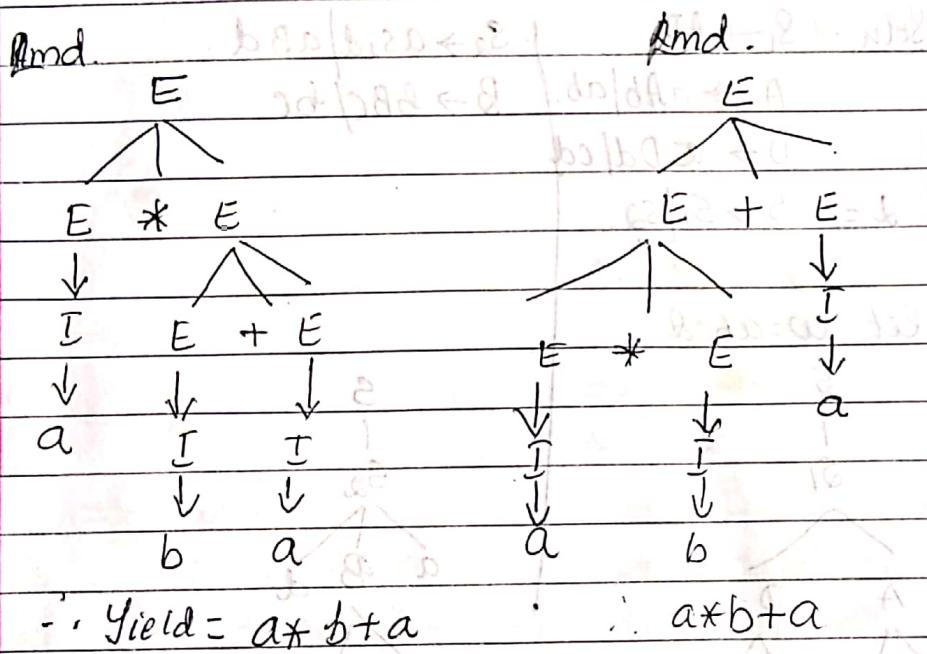
a) Obtain the string  $axb+a$  using the following grammar

$$E \rightarrow I$$

$$E \rightarrow E * E | E + E | ( E )$$

$$I \rightarrow a | b | I 0 | I b | I a$$

Soln: Parse tree.



Yield of both parse tree =  $axb+a$ .

Ambiguous grammar:

For the same string if we are able to write more than one parse tree, then the grammar is said to be ambiguous grammar.

A grammar  $G = (V, T, P, S)$  is said to be ambiguous if there is any string  $w$  belonging to  $T^*$  for which we can write more than one parse tree with the root labelled ' $S$ ' & yield ' $w$ '.

Inherently ambiguous language: A language is said to be inherently ambiguous if we can not write any grammar for this language so that it is unambiguous.

Q. Write a grammar for  $L = a^m b^n c^m d^n / m \geq 1, n \geq 1$

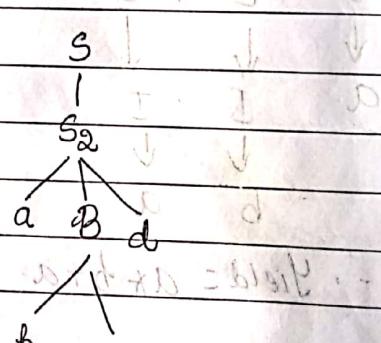
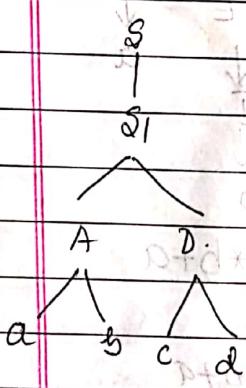
$$\text{Soln. } S_1 \rightarrow AD \quad | \quad S_2 \rightarrow aS_1d / ABd.$$

$$A \rightarrow aAb / ab \quad | \quad B \rightarrow BBC / bc$$

$$D \rightarrow aDd / cd$$

$$L = S \rightarrow S_1 / S_2.$$

Let  $uv=abcd$



abcd

abcd.

As per the above example we cannot write unambiguous grammar if it is inherently ambiguous grammar.

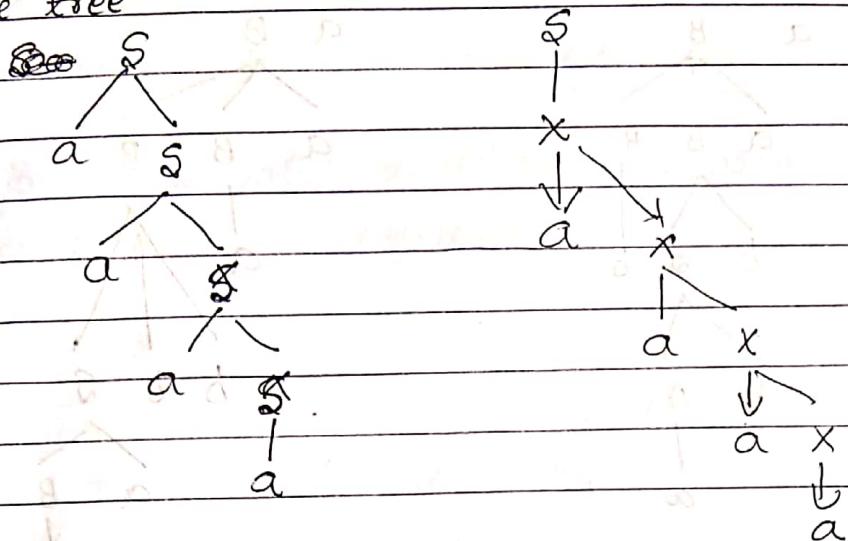
Q. Is the following grammar ambiguous?

$$S \rightarrow aS/x$$

$$x \rightarrow axa$$

Soln: let  $w = aaaa$ .

Parse tree



$$as \xrightarrow{lm} aS$$

$$\Rightarrow a \underline{a} S$$

$$\Rightarrow aa \underline{a} S$$

$$\Rightarrow aaaa$$

$$S \Rightarrow X$$

$$\Rightarrow aX$$

$$\Rightarrow a \underline{a} X$$

$$\Rightarrow a a \underline{a} X$$

$$\Rightarrow a \underline{a} a a$$

$$\underline{\underline{aaaa}}$$

As structure of parse tree are different  
left most derivatives are different for a  
string  $w = aaaa$  it is ambiguous

Q: Is the following ambiguous

$$S \xrightarrow{lm} aB | bA$$

$$A \xrightarrow{lm} aS | BAA | a$$

$$B \xrightarrow{lm} bS | aBB | b$$

Soln:  $w = aabbab$  (any string can be taken)

$$S \xrightarrow{lm} aB$$

$$\Rightarrow a \underline{a} B$$

$$\Rightarrow aa \underline{b} S B$$

$$\Rightarrow aab \underline{b} A B$$

$$\Rightarrow abb \underline{a} B$$

$$\Rightarrow aabbab$$

$$S \xrightarrow{lm} aB$$

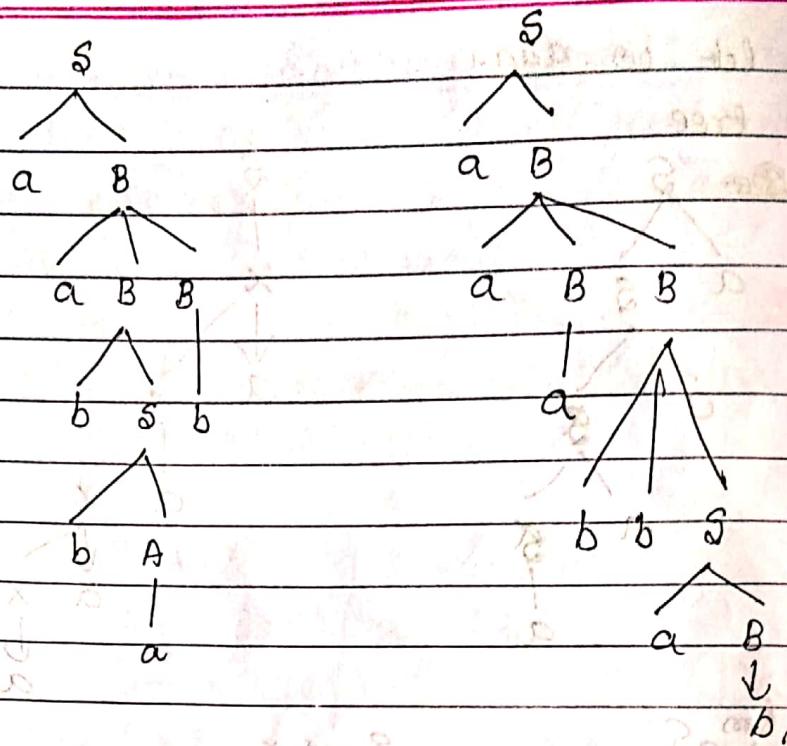
$$\Rightarrow a \underline{a} BB$$

$$\Rightarrow aa \underline{a} b B$$

$$\Rightarrow a a b B$$

$$\Rightarrow aabb a B$$

$$\Rightarrow aabba B$$



The given grammar is ambiguous.

- Q Obtain the string  $aabbabbba$  by applying leftmost derivation and the parse tree for the grammar shown below. Is it possible to obtain the same string again by applying leftmost derivation but by selecting different productions?

$$S \rightarrow aB/bA$$

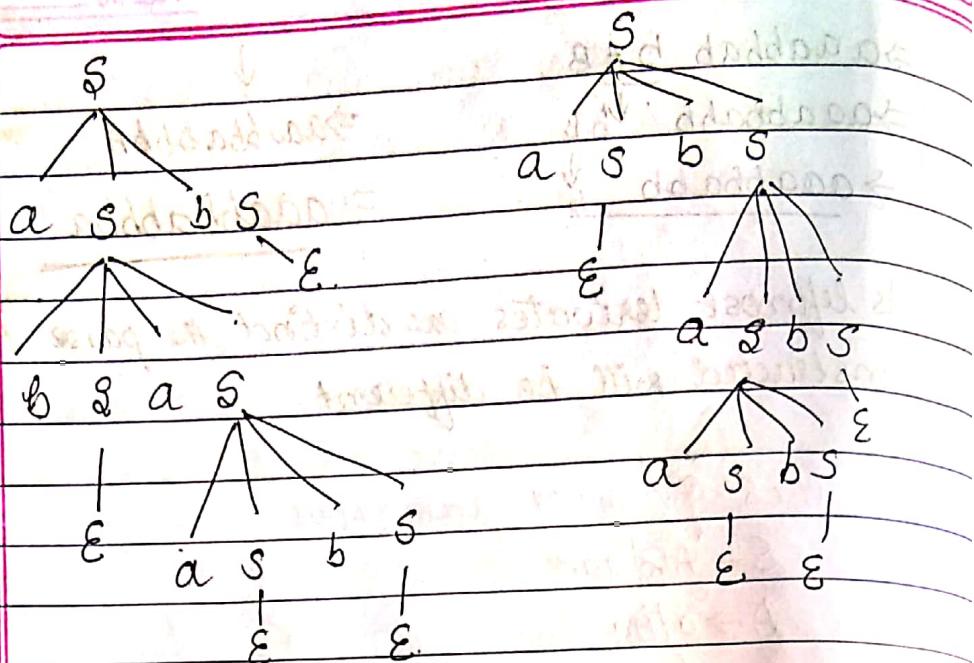
$$A \rightarrow aS/bAA/a$$

$$B \rightarrow bS/aBB/B$$

Soln:  $\begin{array}{l} S \Rightarrow aB \\ \downarrow \\ \Rightarrow a \underline{aBB} \\ \Rightarrow aa \underline{BB} B \\ \Rightarrow aaa \underline{BSBB} \\ \Rightarrow aaab \underline{BABB} \\ \Rightarrow aaabb \underline{BGB} \\ \Rightarrow aaabba \underline{BBS} \end{array}$

$$\begin{array}{l} S \Rightarrow aB \\ \Rightarrow aa \underline{B} \\ \Rightarrow aa \underline{B} B \\ \Rightarrow aa \underline{B} BB \\ \Rightarrow aa \underline{B} BB \\ \Rightarrow aaab \underline{B} \\ \Rightarrow aaabb \underline{B} \\ \Rightarrow aaabba \underline{B} \\ \Rightarrow aaabbab \underline{S} \end{array}$$





~~abab~~  
 abaabb

Grammar is ambiguous

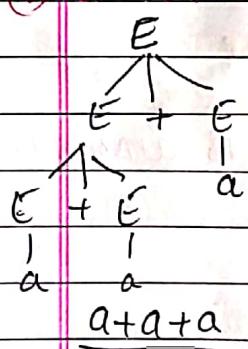
Ambiguous

$E \rightarrow E+E$

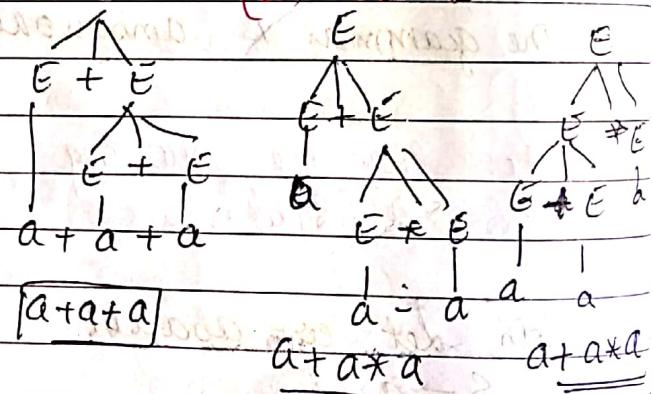
$E \rightarrow E \cdot E$

$E \rightarrow a$

(i)  $a+a+a$

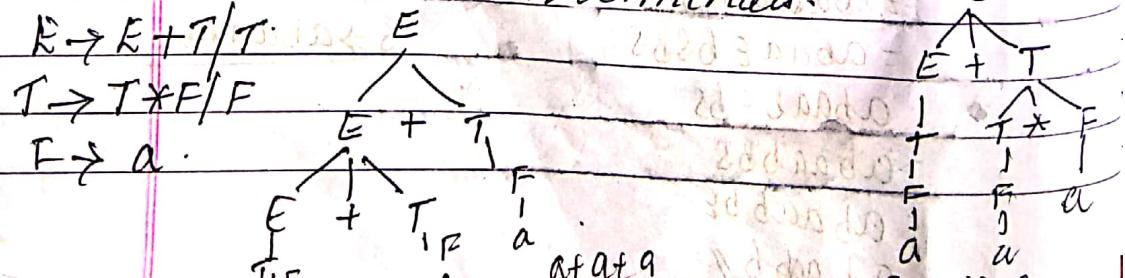


(ii)  $a+a \cdot a$



Unambiguous: To remove ambiguity follow associativity using left recursion.

To have precedence levels should be introduced and non terminals.



## Removing ambiguity in grammars

The reason for ambiguity in grammar is because the precedence of operators are not taken into consideration. If we can force the precedence then we can make grammar unambiguous.

- \* To obtain an unambiguous grammar we can add additional nonterminals to force the precedence factor is an expression that can not be broken apart by any adjacent operator, either  $*$  or  $\wedge$ .

The only factor is our language use

- (i) Identifier: It is not possible to separate the letters of an identifier by attaching an operator all terminals
  - (ii) Any parenthesis expression no matter what appears <sup>inside</sup> prevents from becoming the operand of any operator outside the parenthesis.
  - (iii) A term is an expression that cannot be broken by + or - operator. It has expression of highest precedence that make use of terms & factors.
  - Term is a product of one or more factors.
  - Expression: May be either be terms or necessarily defined using expressions and terms for lower precedence operators.
  - \* takes precedence over +, \* and / left associativity problems avoided by using left precedence its solved by using left associativity

→ left excursion

Q: Convert the following grammar to unambiguous grammar. If the grammar is ambiguous, state so.

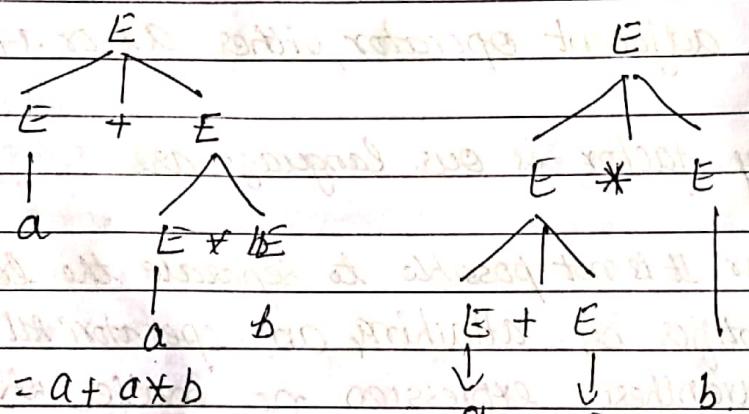
$$E \rightarrow E+E$$

$$E \rightarrow E * E$$

$$E \rightarrow E/E$$

$$E \rightarrow (E) / a/b$$

for  $a+b$ .



So above grammar is ambiguous.

So let us remove the ambiguity.

Step 1: Group all terminals under identifier

$\rightarrow (a, b - \text{name it } I)$

Step 2: Write factors

Step 3: Highest precedence operator ( $T, F$ )  $*$ ,  $/$

Step 4: Lower precedence: (using  $T \& E$ )  $+$

$$I \rightarrow a/b$$

$$F \rightarrow I | CE$$

$$T \rightarrow F | T * F | T / F$$

$$E \rightarrow T | E + T$$

$$E \rightarrow E + T / F$$

$$T \rightarrow T * F | T / F$$

$$F \rightarrow I | (E)$$

$$E \rightarrow a/b$$



Date :

Page No.:

as 14

Q: Show that the following grammar is ambiguous  
& write an unambiguous grammar

$$S \rightarrow S+S \mid S*S \mid S-S \mid b \mid c \mid (S)$$

Soln:

$$\begin{array}{l} S \rightarrow T \mid S+T \mid S-T \\ T \rightarrow F \mid F \times F \\ F \rightarrow I \mid (S) \\ E \rightarrow b \mid c \end{array}$$

↑ higher precedence  
↓ 4 levels

### Precedence constraints (Rule 1)

- The level at which the production is present defines the priority of the operator contained in it
- The higher level → lower priority
- The lower level → higher "
- 

### Associativity

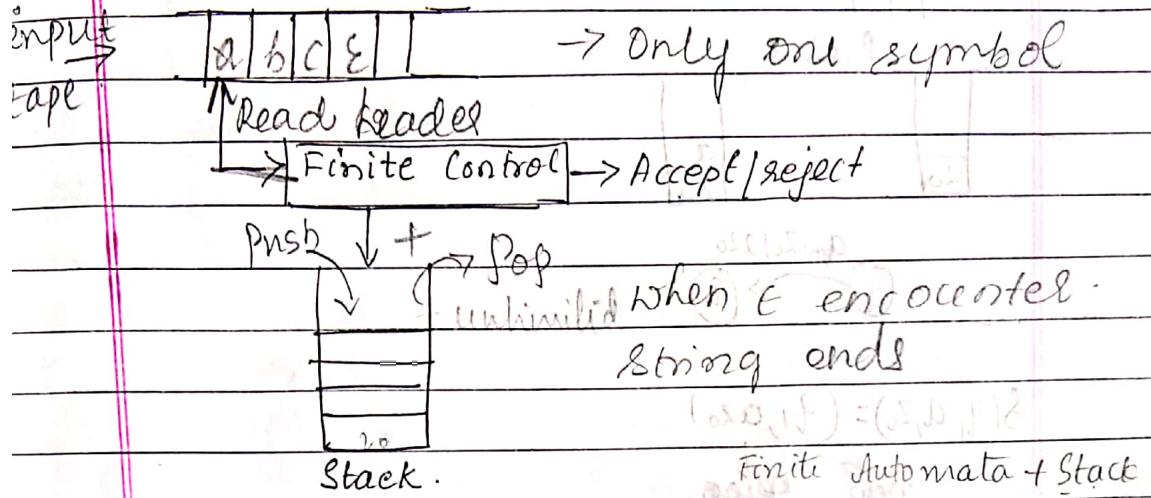
→ If the operator is left associative, introduce left enclosers.



## PUSH DOWN AUTOMATA (PDA).

PDA is E-NFA with an additional stack.

It accepts context free language.



Formal definition: A PDA is defined by the 7 tuples  $M = (Q, \Sigma, T, \delta, q_0, z_0, F)$ .

where  $Q$  = Finite set of states (alphabet).

$\Sigma$  = Set of symbols that may be present in PDA

$T$  = set of all symbols that may be present in stack.  $T \subseteq \Sigma$

$q_0$  = Start state

$z_0$  = Initial symbol on stack.

$F$  = Finite set of accepting states  $F \subseteq Q$ .

$\delta$  = Transition function.  $\delta: Q \times \Sigma \rightarrow Q \times T$

It is defined as  $\delta(q, a, x) = (p, r)$

$q$  = current state  $p$  = final state  $\{top\}$  stack

$a$  = input symbol  $r$  = stack symbol

$x$  = current top of the stack.

$p$  = state to which the machine goes after processing symbol  $a$ .

$r \rightarrow ax$  if  $a$  is pushed onto stack

$\Rightarrow \epsilon$  if stack is popped.

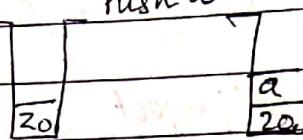
$-x$  if stack is left alone.

$z_0$  = Bottom/Top of stack.

Push

$a \ b$

Push a



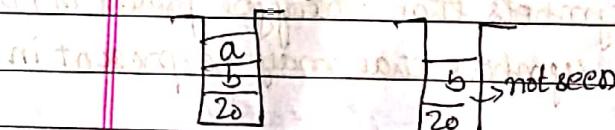
$a_0/z_0/a_0$   
 $(q_i) \rightarrow (q_j)$

$$S(q_i, a, z_0) = (q_j, a_0 z_0)$$

new  $\downarrow$   
 Stack

Pop  
 $a, c, e$  means  
 pop.

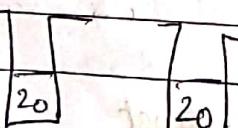
$q_i^0$        $q_j$



$$S(q_i^0, a, c) = (q_j^0, e)$$

Skip

$q_i^0$        $q_j$





Q 1

Design a PDA for the language  
 $L = \{a^n b^n | n \geq 1\}$

Soln:

$L = \{aab, aabb, aaabbb, aaaabbbb, \dots\}$

Acceptance      aaabbb

by

empty  
stack

Since we need to identify

same no. of 'a's & 'b's

We keep pushing 'a's (call)

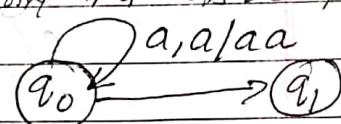
then whenever 'b' is encountered pop a, if after  
popping all b's from stack,  $a_0/a_0/\epsilon$  is accepted.

Step 1: Push a top of stack

first move:  $\delta(q_0, a, z_0) = (q_0, a z_0)$

Current state  $\xrightarrow{\text{top}} q_0$   $\xrightarrow{\text{next state}} z_0$  Top of stack

2nd move  $\delta(q_0, a, a) = (q_0, a a)$



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

till we encounter a

we are at  $q_0$

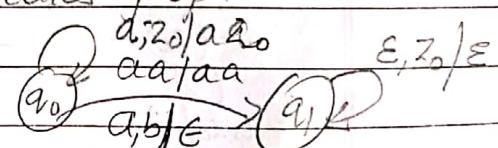
once b encountered  $\rightarrow$  goes  
to  $q_1$

indicates pop

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

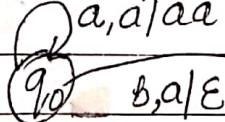
$\delta(q_1, b, a) = (q_1, \epsilon)$

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



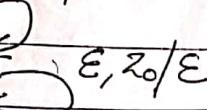
$a, z_0 / a z_0$

$a, a / aa$

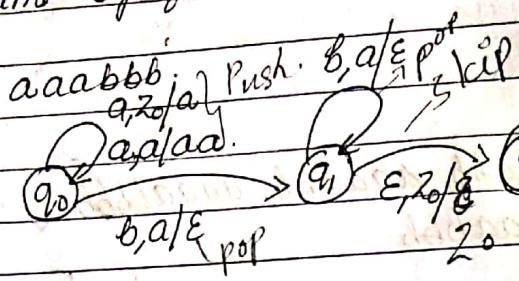


$b, a / \epsilon$

$\epsilon, z_0 / \epsilon$



Acceptance by final state -



$$\delta(q_0, a, z_0) = (q_0, \alpha_{z_0})$$

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

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

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

$$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$$

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F) = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \{q_2\})$$

$$M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \{q_2\})$$

Q.2 Design a push accept @ string

$$L = \{abb, aa\}$$

Soln For 1st sec  $\beta$

$$\delta(q_0, a, z_0) =$$

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

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

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

$$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$$

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

$$\delta(q_2, \epsilon, z_0) = (q_2, z_0)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

$$\delta(q_2, b, b) = (q_2, bb)$$

$$\delta(q_2, a, b) = (q_2, ba)$$

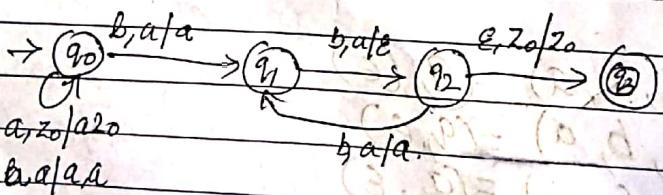
$$\delta(q_2, b, a) = (q_2, ab)$$

$$\delta(q_2, a, a) = (q_2, aa)$$

Q.3 Design a PD

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

$$L = \{ababa, \dots\}$$



Soln:

$$\delta(q_0, a, z_0) =$$

$$\delta(q_0, b, z_0) =$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Q.2 Design a pushdown automata (PDA) to accept a string of the form  $L = \{a^n b^n | n \geq 1\}$ .

$$L = \{ abb, aabb, aaabb, \dots \}$$

Soln For 1a sec push two a's.

$$\delta(q_0, a, z_0) = (q_0, aa z_0) \quad \text{push } a \quad \text{push two element}$$

*(skip)*

$$\delta(q_0, a, a) = (q_0, aaa) \rightarrow \text{push } a \quad \text{push two element}$$

*(skip)*

$$\delta(q_0, b, a) = (q_1, \epsilon) \rightarrow \text{pop } a \quad \text{pop two element}$$

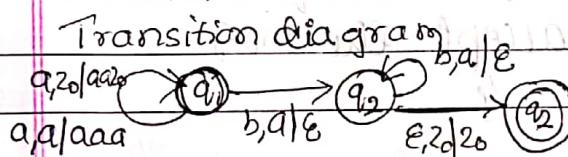
*(skip)*

$$\delta(q_1, b, a) = (q_1, \epsilon) \rightarrow \text{pop } a \quad \text{pop two element}$$

*(skip)*

$$\delta(q_1, \epsilon, z_0) = (q_2, z_0) \rightarrow \text{skip } a \quad \text{skip two element}$$

*(skip)*



$$M = \{ \{q_0, q_1, q_2\}, \{a, b\}, \Sigma, \delta, q_0, z_0 \mid q_2 \}$$

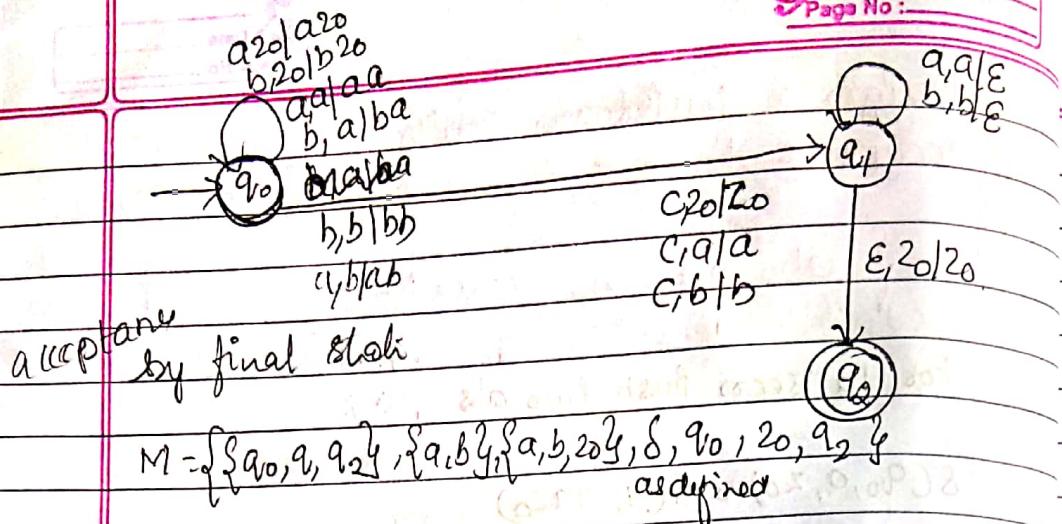
Q.3 Design a PDA for the language  $L = \{w \in \Sigma^* \mid w = w^R \text{ over } \Sigma = \{a, b\}\}$  over

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

$$L = \{ abca, aaca, abacab, bacab, \dots \}$$

Soln:

$\delta(q_0, a, z_0) = (q_0, a z_0)$	$w = ab$	$\text{Push } a$
$\delta(q_0, b, z_0) = (q_0, b z_0)$	$w^R = ba$	$\text{Reverse } z_0$
$\delta(q_0, a, a) = (q_0, aa)$	$a, b$	$\text{Push } a, b$
$\delta(q_0, b, a) = (q_0, ba)$		$\text{Push } b, a$
$\delta(q_0, a, b) = (q_0, ab)$		$\text{Push } a, b$
$\delta(q_0, b, b) = (q_0, bb)$		$\text{Push } b, b$
$\delta(q_0, \epsilon, z_0) = (q_1, z_0)$		$\text{skip } z_0$
$\delta(q_0, c, a) = (q_1, a)$		$\text{skip } a$
$\delta(q_0, c, b) = (q_1, b)$		$\text{skip } b$
$\delta(q_1, a, a) = (q_1, \epsilon)$		$\text{skip } a$
$\delta(q_1, b, b) = (q_1, \epsilon)$		$\text{skip } b$
$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$		$\text{skip } z_0$



**Q4** Design a PDA for the language  $a^i b^j$  such that  $i \leq j$  and  $i, j \geq 1$ .

Soln: Push on  $a$ 's, Pop on  $b$ 's & if stack is empty accept the string  
 $L = \{abb, abbb\}$

$$\delta(q_0, a, z_0) = (q_0, a20)$$

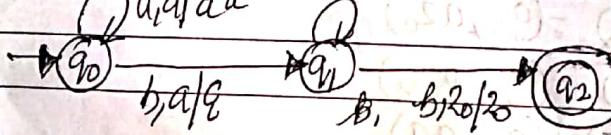
$$\delta(q_0, aa) = (q_0, aa)$$

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

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

$$\delta(q_1, b, z_0) = (q_2, z_0)$$

$$a, z_0/a20 \quad b/p/\epsilon.$$



$$M = \{q_0, q_1, q_2\}, \{a, b\}, \{a, 20, q_1, q_2\}, \delta \text{ as defined}$$

$$a, 20/a20 \quad b, 20/z_0$$

$$a, a/a \quad a, a/\epsilon$$

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

**Q5**

Design a PDA  
 $L(M) = \{ \}$

Soln:

Same sign  
 Different sign

$$\delta(q_0, a, z_0)$$

$$\delta(q_0, b, z_0)$$

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

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

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

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

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

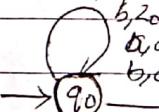
$$a, b/z_0$$

$$b, a/z_0$$

$$M = \{q_0, q_1\}$$

**Q6.** Design a PDA

Soln:  $L = \{aba, abba\}$   
 This is palin



$$a, 20/a20 \quad b, 20/z_0$$

$$a, a/a \quad a, a/\epsilon$$

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



Q5

Design a PDA that accept language  $L(M) = \{ w \mid w \in (a+b)^* \text{ and } n_a(w) = n_b(w) \}$

dpa

Soln:

Same symbol push  $(a, a)$

Different symbol Pop  $(a, b)$

$$\delta(q_0, a, z_0) = (q_0, a z_0)$$

$$\delta(q_0, b, z_0) = (q_0, b z_0)$$

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

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

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

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

$$\delta(q_0, \epsilon, z_0) = (q_1, z_0)$$

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

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

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

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

$$\delta(q_2, \epsilon, z_0) = (q_3, z_0)$$

$$\delta(q_3, a, a) = (q_3, aa)$$

$$\delta(q_3, b, b) = (q_3, bb)$$

$$\delta(q_3, a, b) = (q_4, \epsilon)$$

$$\delta(q_3, b, a) = (q_4, \epsilon)$$

$$\delta(q_4, \epsilon, z_0) = (q_5, z_0)$$

$$\delta(q_5, a, a) = (q_5, aa)$$

$$\delta(q_5, b, b) = (q_5, bb)$$

$$\delta(q_5, a, b) = (q_6, \epsilon)$$

$$\delta(q_5, b, a) = (q_6, \epsilon)$$

$$\delta(q_6, \epsilon, z_0) = (q_7, z_0)$$

$$\delta(q_7, a, a) = (q_7, aa)$$

$$\delta(q_7, b, b) = (q_7, bb)$$

$$\delta(q_7, a, b) = (q_8, \epsilon)$$

$$\delta(q_7, b, a) = (q_8, \epsilon)$$

$$\delta(q_8, \epsilon, z_0) = (q_9, z_0)$$

$$\delta(q_9, a, a) = (q_9, aa)$$

$$\delta(q_9, b, b) = (q_9, bb)$$

$$\delta(q_9, a, b) = (q_{10}, \epsilon)$$

$$\delta(q_9, b, a) = (q_{10}, \epsilon)$$

$$\delta(q_{10}, \epsilon, z_0) = (q_{11}, z_0)$$

$$\delta(q_{11}, a, a) = (q_{11}, aa)$$

$$\delta(q_{11}, b, b) = (q_{11}, bb)$$

$$\delta(q_{11}, a, b) = (q_{12}, \epsilon)$$

$$\delta(q_{11}, b, a) = (q_{12}, \epsilon)$$

$$\delta(q_{12}, \epsilon, z_0) = (q_{13}, z_0)$$

$$\delta(q_{13}, a, a) = (q_{13}, aa)$$

$$\delta(q_{13}, b, b) = (q_{13}, bb)$$

$$\delta(q_{13}, a, b) = (q_{14}, \epsilon)$$

$$\delta(q_{13}, b, a) = (q_{14}, \epsilon)$$

$$\delta(q_{14}, \epsilon, z_0) = (q_{15}, z_0)$$

$$\delta(q_{15}, a, a) = (q_{15}, aa)$$

$$\delta(q_{15}, b, b) = (q_{15}, bb)$$

$$\delta(q_{15}, a, b) = (q_{16}, \epsilon)$$

$$\delta(q_{15}, b, a) = (q_{16}, \epsilon)$$

$$\delta(q_{16}, \epsilon, z_0) = (q_{17}, z_0)$$

$$\delta(q_{17}, a, a) = (q_{17}, aa)$$

$$\delta(q_{17}, b, b) = (q_{17}, bb)$$

$$\delta(q_{17}, a, b) = (q_{18}, \epsilon)$$

$$\delta(q_{17}, b, a) = (q_{18}, \epsilon)$$

$$\delta(q_{18}, \epsilon, z_0) = (q_{19}, z_0)$$

$$\delta(q_{19}, a, a) = (q_{19}, aa)$$

$$\delta(q_{19}, b, b) = (q_{19}, bb)$$

$$\delta(q_{19}, a, b) = (q_{20}, \epsilon)$$

$$\delta(q_{19}, b, a) = (q_{20}, \epsilon)$$

$$\delta(q_{20}, \epsilon, z_0) = (q_{21}, z_0)$$

$$\delta(q_{21}, a, a) = (q_{21}, aa)$$

$$\delta(q_{21}, b, b) = (q_{21}, bb)$$

$$\delta(q_{21}, a, b) = (q_{22}, \epsilon)$$

$$\delta(q_{21}, b, a) = (q_{22}, \epsilon)$$

$$\delta(q_{22}, \epsilon, z_0) = (q_{23}, z_0)$$

$$\delta(q_{23}, a, a) = (q_{23}, aa)$$

$$\delta(q_{23}, b, b) = (q_{23}, bb)$$

$$\delta(q_{23}, a, b) = (q_{24}, \epsilon)$$

$$\delta(q_{23}, b, a) = (q_{24}, \epsilon)$$

$$\delta(q_{24}, \epsilon, z_0) = (q_{25}, z_0)$$

$$\delta(q_{25}, a, a) = (q_{25}, aa)$$

$$\delta(q_{25}, b, b) = (q_{25}, bb)$$

$$\delta(q_{25}, a, b) = (q_{26}, \epsilon)$$

$$\delta(q_{25}, b, a) = (q_{26}, \epsilon)$$

$$\delta(q_{26}, \epsilon, z_0) = (q_{27}, z_0)$$

$$\delta(q_{27}, a, a) = (q_{27}, aa)$$

$$\delta(q_{27}, b, b) = (q_{27}, bb)$$

$$\delta(q_{27}, a, b) = (q_{28}, \epsilon)$$

$$\delta(q_{27}, b, a) = (q_{28}, \epsilon)$$

$$\delta(q_{28}, \epsilon, z_0) = (q_{29}, z_0)$$

$$\delta(q_{29}, a, a) = (q_{29}, aa)$$

$$\delta(q_{29}, b, b) = (q_{29}, bb)$$

$$\delta(q_{29}, a, b) = (q_{30}, \epsilon)$$

$$\delta(q_{29}, b, a) = (q_{30}, \epsilon)$$

$$\delta(q_{30}, \epsilon, z_0) = (q_{31}, z_0)$$

$$\delta(q_{31}, a, a) = (q_{31}, aa)$$

$$\delta(q_{31}, b, b) = (q_{31}, bb)$$

$$\delta(q_{31}, a, b) = (q_{32}, \epsilon)$$

$$\delta(q_{31}, b, a) = (q_{32}, \epsilon)$$

$$\delta(q_{32}, \epsilon, z_0) = (q_{33}, z_0)$$

$$\delta(q_{33}, a, a) = (q_{33}, aa)$$

$$\delta(q_{33}, b, b) = (q_{33}, bb)$$

$$\delta(q_{33}, a, b) = (q_{34}, \epsilon)$$

$$\delta(q_{33}, b, a) = (q_{34}, \epsilon)$$

$$\delta(q_{34}, \epsilon, z_0) = (q_{35}, z_0)$$

$$\delta(q_{35}, a, a) = (q_{35}, aa)$$

$$\delta(q_{35}, b, b) = (q_{35}, bb)$$

$$\delta(q_{35}, a, b) = (q_{36}, \epsilon)$$

$$\delta(q_{35}, b, a) = (q_{36}, \epsilon)$$

$$\delta(q_{36}, \epsilon, z_0) = (q_{37}, z_0)$$

$$\delta(q_{37}, a, a) = (q_{37}, aa)$$

$$\delta(q_{37}, b, b) = (q_{37}, bb)$$

$$\delta(q_{37}, a, b) = (q_{38}, \epsilon)$$

$$\delta(q_{37}, b, a) = (q_{38}, \epsilon)$$

$$\delta(q_{38}, \epsilon, z_0) = (q_{39}, z_0)$$

$$\delta(q_{39}, a, a) = (q_{39}, aa)$$

$$\delta(q_{39}, b, b) = (q_{39}, bb)$$

$$\delta(q_{39}, a, b) = (q_{40}, \epsilon)$$

$$\delta(q_{39}, b, a) = (q_{40}, \epsilon)$$

$$\delta(q_{40}, \epsilon, z_0) = (q_{41}, z_0)$$

$$\delta(q_{41}, a, a) = (q_{41}, aa)$$

$$\delta(q_{41}, b, b) = (q_{41}, bb)$$

$$\delta(q_{41}, a, b) = (q_{42}, \epsilon)$$

$$\delta(q_{41}, b, a) = (q_{42}, \epsilon)$$

$$\delta(q_{42}, \epsilon, z_0) = (q_{43}, z_0)$$

$$\delta(q_{43}, a, a) = (q_{43}, aa)$$

$$\delta(q_{43}, b, b) = (q_{43}, bb)$$

$$\delta(q_{43}, a, b) = (q_{44}, \epsilon)$$

$$\delta(q_{43}, b, a) = (q_{44}, \epsilon)$$

$$\delta(q_{44}, \epsilon, z_0) = (q_{45}, z_0)$$

$$\delta(q_{45}, a, a) = (q_{45}, aa)$$

$$\delta(q_{45}, b, b) = (q_{45}, bb)$$

$$\delta(q_{45}, a, b) = (q_{46}, \epsilon)$$

$$\delta(q_{45}, b, a) = (q_{46}, \epsilon)$$

$$\delta(q_{46}, \epsilon, z_0) = (q_{47}, z_0)$$

$$\delta(q_{47}, a, a) = (q_{47}, aa)$$

$$\delta(q_{47}, b, b) = (q_{47}, bb)$$

$$\delta(q_{47}, a, b) = (q_{48}, \epsilon)$$

$$\delta(q_{47}, b, a) = (q_{48}, \epsilon)$$

$$\delta(q_{48}, \epsilon, z_0) = (q_{49}, z_0)$$

$$\delta(q_{49}, a, a) = (q_{49}, aa)$$

$$\delta(q_{49}, b, b) = (q_{49}, bb)$$

$$\delta(q_{49}, a, b) = (q_{50}, \epsilon)$$

$$\delta(q_{49}, b, a) = (q_{50}, \epsilon)$$

$$\delta(q_{50}, \epsilon, z_0) = (q_{51}, z_0)$$

$$\delta(q_{51}, a, a) = (q_{51}, aa)$$

$$\delta(q_{51}, b, b) = (q_{51}, bb)$$

$$\delta(q_{51}, a, b) = (q_{52}, \epsilon)$$

$$\delta(q_{51}, b, a) = (q_{52}, \epsilon)$$

$$\delta(q_{52}, \epsilon, z_0) = (q_{53}, z_0)$$

$$\delta(q_{53}, a, a) = (q_{53}, aa)$$

$$\delta(q_{53}, b, b) = (q_{53}, bb)$$

$$\delta(q_{53}, a, b) = (q_{54}, \epsilon)$$

$$\delta(q_{53}, b, a) = (q_{54}, \epsilon)$$

$$\delta(q_{54}, \epsilon, z_0) = (q_{55}, z_0)$$

$$\delta(q_{55}, a, a) = (q_{55}, aa)$$

$$\delta(q_{55}, b, b) = (q_{55}, bb)$$

$$\delta(q_{55}, a, b) = (q_{56}, \epsilon)$$

$$\delta(q_{55}, b, a) = (q_{56}, \epsilon)$$

$$\delta(q_{56}, \epsilon, z_0) = (q_{57}, z_0)$$

$$\delta(q_{57}, a, a) = (q_{57}, aa)$$

$$\delta(q_{57}, b, b) = (q_{57}, bb)$$

$$\delta(q_{57}, a, b) = (q_{58}, \epsilon)$$

$$\delta(q_{57}, b, a) = (q_{58}, \epsilon)$$

$$\delta(q_{58}, \epsilon, z_0) = (q_{59}, z_0)$$

$$\delta(q_{59}, a, a) = (q_{59}, aa)$$

$$\delta(q_{59}, b, b) = (q_{59}, bb)$$

$$\delta(q_{59}, a, b) = (q_{60}, \epsilon)$$

$$\delta(q_{59}, b, a) = (q_{60}, \epsilon)$$

$$\delta(q_{60}, \epsilon, z_0) = (q_{61}, z_0)$$

$$\delta(q_{61}, a, a) = (q_{61}, aa)$$

$$\delta(q_{61}, b, b) = (q_{61}, bb)$$

$$\delta(q_{61}, a, b) = (q_{62}, \epsilon)$$

$$\delta(q_{61}, b, a) = (q_{62}, \epsilon)$$

$$\delta(q_{62}, \epsilon, z_0) = (q_{63}, z_0)$$

$$\delta(q_{63}, a, a) = (q_{63}, aa)$$

$$\delta(q_{63}, b, b) = (q_{63}, bb)$$

$$\delta(q_{63}, a, b) = (q_{64}, \epsilon)$$

$$\delta(q_{63}, b, a) = (q_{64}, \epsilon)$$

$$\delta(q_{64}, \epsilon, z_0) = (q_{65}, z_0)$$

$$\delta(q_{65}, a, a) = (q_{65}, aa)$$

$$\delta(q_{65}, b, b) = (q_{65}, bb)$$

$$\delta(q_{65}, a, b) = (q_{66}, \epsilon)$$

$$\delta(q_{65}, b, a) = (q_{66}, \epsilon)$$

$$\delta(q_{66}, \epsilon, z_0) = (q_{67}, z_0)$$

$$\delta(q_{67}, a, a) = (q_{67}, aa)$$

$$\delta(q_{67}, b, b) = (q_{67}, bb)$$

$$\delta(q_{67}, a, b) = (q_{68}, \epsilon)$$

$$\delta(q_{67}, b, a)$$

for 3 states M = {q0, q1, q2} and 2 inputs A = {a, b}

$$f(w) : M = \{q_0, q_1, q_2\} \text{ and } A = \{a, b\} \Rightarrow M = \{M\}.$$

$$\delta(q_0, a, z_0) = (q_0, a, z_0)$$

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

$$\delta(q_0, b, z_0) = (q_0, b, z_0)$$

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\delta(q_2, a, a) = (q_2, a)$$

$$\delta(q_2, b, b) = (q_2, b)$$

$$\delta(q_2, a, b) = (q_2, ab)$$

$$\delta(q_2, b, a) = (q_2, ba)$$

$$\delta(q_3, a, a) = (q_3, a)$$

$$\delta(q_3, b, b) = (q_3, b)$$

$$\delta(q_3, a, b) = (q_3, ab)$$

$$\delta(q_3, b, a) = (q_3, ba)$$

$$\delta(q_4, a, a) = (q_4, a)$$

$$\delta(q_4, b, b) = (q_4, b)$$

$$\delta(q_4, a, b) = (q_4, ab)$$

$$\delta(q_4, b, a) = (q_4, ba)$$

$$\delta(q_5, a, a) = (q_5, a)$$

$$\delta(q_5, b, b) = (q_5, b)$$

$$\delta(q_5, a, b) = (q_5, ab)$$

$$\delta(q_5, b, a) = (q_5, ba)$$

$$\delta(q_6, a, a) = (q_6, a)$$

$$\delta(q_6, b, b) = (q_6, b)$$

$$\delta(q_6, a, b) = (q_6, ab)$$

$$\delta(q_6, b, a) = (q_6, ba)$$

$$\delta(q_7, a, a) = (q_7, a)$$

$$\delta(q_7, b, b) = (q_7, b)$$

$$\delta(q_7, a, b) = (q_7, ab)$$

$$\delta(q_7, b, a) = (q_7, ba)$$

$$\delta(q_8, a, a) = (q_8, a)$$

$$\delta(q_8, b, b) = (q_8, b)$$

$$\delta(q_8, a, b) = (q_8, ab)$$

$$\delta(q_8, b, a) = (q_8, ba)$$

$$\delta(q_9, a, a) = (q_9, a)$$

$$\delta(q_9, b, b) = (q_9, b)$$

$$\delta(q_9, a, b) = (q_9, ab)$$

$$\delta(q_9, b, a) = (q_9, ba)$$

$$\delta(q_{10}, a, a) = (q_{10}, a)$$

$$\delta(q_{10}, b, b) = (q_{10}, b)$$

$$\delta(q_{10}, a, b) = (q_{10}, ab)$$

$$\delta(q_{10}, b, a) = (q_{10}, ba)$$

$$\delta(q_{11}, a, a) = (q_{11}, a)$$

$$\delta(q_{11}, b, b) = (q_{11}, b)$$

$$\delta(q_{11}, a, b) = (q_{11}, ab)$$

$$\delta(q_{11}, b, a) = (q_{11}, ba)$$

$$\delta(q_{12}, a, a) = (q_{12}, a)$$

$$\delta(q_{12}, b, b) = (q_{12}, b)$$

$$\delta(q_{12}, a, b) = (q_{12}, ab)$$

$$\delta(q_{12}, b, a) = (q_{12}, ba)$$

$$\delta(q_{13}, a, a) = (q_{13}, a)$$

$$\delta(q_{13}, b, b) = (q_{13}, b)$$

$$\delta(q_{13}, a, b) = (q_{13}, ab)$$

$$\delta(q_{13}, b, a) = (q_{13}, ba)$$

$$\delta(q_{14}, a, a) = (q_{14}, a)$$

$$\delta(q_{14}, b, b) = (q_{14}, b)$$

$$\delta(q_{14}, a, b) = (q_{14}, ab)$$

$$\delta(q_{14}, b, a) = (q_{14}, ba)$$

$$\delta(q_{15}, a, a) = (q_{15}, a)$$

$$\delta(q_{15}, b, b) = (q_{15}, b)$$

$$\delta(q_{15}, a, b) = (q_{15}, ab)$$

$$\delta(q_{15}, b, a) = (q_{15}, ba)$$

$$\delta(q_{16}, a, a) = (q_{16}, a)$$

$$\delta(q_{16}, b, b) = (q_{16}, b)$$

$$\delta(q_{16}, a, b) = (q_{16}, ab)$$

$$\delta(q_{16}, b, a) = (q_{16}, ba)$$

$$\delta(q_{17}, a, a) = (q_{17}, a)$$

$$\delta(q_{17}, b, b) = (q_{17}, b)$$

$$\delta(q_{17}, a, b) = (q_{17}, ab)$$

$$\delta(q_{17}, b, a) = (q_{17}, ba)$$

$$\delta(q_{18}, a, a) = (q_{18}, a)$$

$$\delta(q_{18}, b, b) = (q_{18}, b)$$

$$\delta(q_{18}, a, b) = (q_{18}, ab)$$

$$\delta(q_{18}, b, a) = (q_{18}, ba)$$

$$\delta(q_{19}, a, a) = (q_{19}, a)$$

$$\delta(q_{19}, b, b) = (q_{19}, b)$$

$$\delta(q_{19}, a, b) = (q_{19}, ab)$$

$$\delta(q_{19}, b, a) = (q_{19}, ba)$$

$$\delta(q_{20}, a, a) = (q_{20}, a)$$

$$\delta(q_{20}, b, b) = (q_{20}, b)$$

$$\delta(q_{20}, a, b) = (q_{20}, ab)$$

$$\delta(q_{20}, b, a) = (q_{20}, ba)$$

$$\delta(q_{21}, a, a) = (q_{21}, a)$$

$$\delta(q_{21}, b, b) = (q_{21}, b)$$

$$\delta(q_{21}, a, b) = (q_{21}, ab)$$

$$\delta(q_{21}, b, a) = (q_{21}, ba)$$

$$\delta(q_{22}, a, a) = (q_{22}, a)$$

$$\delta(q_{22}, b, b) = (q_{22}, b)$$

$$\delta(q_{22}, a, b) = (q_{22}, ab)$$

$$\delta(q_{22}, b, a) = (q_{22}, ba)$$

$$\delta(q_{23}, a, a) = (q_{23}, a)$$

$$\delta(q_{23}, b, b) = (q_{23}, b)$$

$$\delta(q_{23}, a, b) = (q_{23}, ab)$$

$$\delta(q_{23}, b, a) = (q_{23}, ba)$$

$$\delta(q_{24}, a, a) = (q_{24}, a)$$

$$\delta(q_{24}, b, b) = (q_{24}, b)$$

$$\delta(q_{24}, a, b) = (q_{24}, ab)$$

$$\delta(q_{24}, b, a) = (q_{24}, ba)$$

$$\delta(q_{25}, a, a) = (q_{25}, a)$$

$$\delta(q_{25}, b, b) = (q_{25}, b)$$

$$\delta(q_{25}, a, b) = (q_{25}, ab)$$

$$\delta(q_{25}, b, a) = (q_{25}, ba)$$

$$\delta(q_{26}, a, a) = (q_{26}, a)$$

$$\delta(q_{26}, b, b) = (q_{26}, b)$$

$$\delta(q_{26}, a, b) = (q_{26}, ab)$$

$$\delta(q_{26}, b, a) = (q_{26}, ba)$$

$$\delta(q_{27}, a, a) = (q_{27}, a)$$

$$\delta(q_{27}, b, b) = (q_{27}, b)$$

$$\delta(q_{27}, a, b) = (q_{27}, ab)$$

$$\delta(q_{27}, b, a) = (q_{27}, ba)$$

$$\delta(q_{28}, a, a) = (q_{28}, a)$$

$$\delta(q_{28}, b, b) = (q_{28}, b)$$

$$\delta(q_{28}, a, b) = (q_{28}, ab)$$

$$\delta(q_{28}, b, a) = (q_{28}, ba)$$

$$\delta(q_{29}, a, a) = (q_{29}, a)$$

$$\delta(q_{29}, b, b) = (q_{29}, b)$$

$$\delta(q_{29}, a, b) = (q_{29}, ab)$$

$$\delta(q_{29}, b, a) = (q_{29}, ba)$$

$$\delta(q_{30}, a, a) = (q_{30}, a)$$

$$\delta(q_{30}, b, b) = (q_{30}, b)$$

$$\delta(q_{30}, a, b) = (q_{30}, ab)$$

$$\delta(q_{30}, b, a) = (q_{30}, ba)$$

$$\delta(q_{31}, a, a) = (q_{31}, a)$$

$$\delta(q_{31}, b, b) = (q_{31}, b)$$

$$\delta(q_{31}, a, b) = (q_{31}, ab)$$

$$\delta(q_{31}, b, a) = (q_{31}, ba)$$

$$\delta(q_{32}, a, a) = (q_{32}, a)$$

$$\delta(q_{32}, b, b) = (q_{32}, b)$$

$$\delta(q_{32}, a, b) = (q_{32}, ab)$$

$$\delta(q_{32}, b, a) = (q_{32}, ba)$$

$$\delta(q_{33}, a, a) = (q_{33}, a)$$

$$\delta(q_{33}, b, b) = (q_{33}, b)$$

$$\delta(q_{33}, a, b) = (q_{33}, ab)$$

$$\delta(q_{33}, b, a) = (q_{33}, ba)$$

$$\delta(q_{34}, a, a) = (q_{34}, a)$$

$$\delta(q_{34}, b, b) = (q_{34}, b)$$

$$\delta(q_{34}, a, b) = (q_{34}, ab)$$

$$\delta(q_{34}, b, a) = (q_{34}, ba)$$

$$\delta(q_{35}, a, a) = (q_{35}, a)$$

$$\delta(q_{35}, b, b) = (q_{35}, b)$$

$$\delta(q_{35}, a, b) = (q_{35}, ab)$$

$$\delta(q_{35}, b, a) = (q_{35}, ba)$$

$$\delta(q_{36}, a, a) = (q_{36}, a)$$

$$\delta(q_{36}, b, b) = (q_{36}, b)$$

$$\delta(q_{36}, a, b) = (q_{36}, ab)$$

$$\delta(q_{36}, b, a) = (q_{36}, ba)$$

$$\delta(q_{37}, a, a) = (q_{37}, a)$$

$$\delta(q_{37}, b, b) = (q_{37}, b)$$

$$\delta(q_{37}, a, b) = (q_{37}, ab)$$

$$\delta(q_{37}, b, a) = (q_{37}, ba)$$

$$\delta(q_{38}, a, a) = (q_{38}, a)$$

$$\delta(q_{38}, b, b) = (q_{38}, b)$$

$$\delta(q_{38}, a, b) = (q_{38}, ab)$$

$$\delta(q_{38}, b, a) = (q_{38}, ba)$$

$$\delta(q_{39}, a, a) = (q_{39}, a)$$

$$\delta(q_{39}, b, b) = (q_{39}, b)$$

$$\delta(q_{39}, a, b) = (q_{39}, ab)$$

$$\delta(q_{39}, b, a) = (q_{39}, ba)$$

$$\delta(q_{40}, a, a) = (q_{40}, a)$$

$$\delta(q_{40}, b, b) = (q_{40}, b)$$

$$\delta(q_{40}, a, b) = (q_{40}, ab)$$

$$\delta(q_{40}, b, a) = (q_{40}, ba)$$

$$\delta(q_{41}, a, a) = (q_{41}, a)$$

$$\delta(q_{41}, b, b) = (q_{41}, b)$$

$$\delta(q_{41}, a, b) = (q_{41}, ab)$$

$$\delta(q_{41}, b, a) = (q_{41}, ba)$$

$$\delta(q_{42}, a, a) = (q_{42}, a)$$

$$\delta(q_{42}, b, b) = (q_{42}, b)$$

$$\delta(q_{42}, a, b) = (q_{42}, ab)$$

$$\delta(q_{42}, b, a) = (q_{42}, ba)$$

$$\delta(q_{43}, a, a) = (q_{43}, a)$$

$$\delta(q_{43}, b, b) = (q_{43}, b)$$



## Instantaneous Description.

During execution the PDA goes through a series of a series of configurations called instantaneous description given by  $(q, aw, bx)$

$q$  - current state

$aw$  - input string yet to be processed with left most symbol ( $a$ ) & current  $\epsilon/p$ .

$bx$  - string on the stack, with left most symbol.

' $b$ ' as top of stack.

move: It is change in instantaneous description represented by  $\leftarrow$  (Sleeping T).

Language of PDA: A PDA accepts a string in two ways

(i) By accepting state

(ii) By empty stack.

(i) Language by accepting state: The language of PDA  $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$  is given by  $L(M) = \{ w \mid (q_0, w, z_0) \xrightarrow{*} (q_f, \epsilon, \lambda) \text{ where } q_f \in F \text{ and stack may or may not be empty}$

(ii) Language of PDA by empty stack:

Language of PDA  $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \emptyset)$  is given by  $L(M) = \{ w \mid (q_0, w, z_0) \xrightarrow{*} (q_f, \epsilon, \emptyset) \text{ where } q_f \in Q \text{ and stack should be empty}$

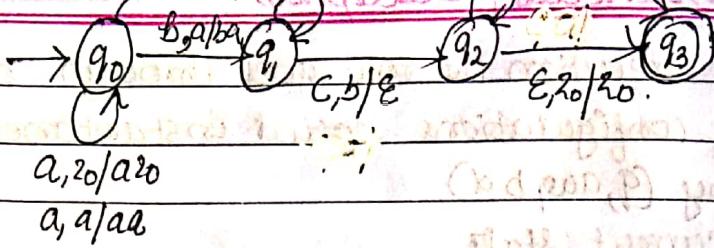
Q.T. Design a PDA for the language  $a^m b^n c^{n+m}$  m>0. Show the moves on the string  $abc^2$ .

Soln:  $\delta = \{ ab^2, abb^2, aabb^2 \}$  (Q.P. 18)

As long as  $a$  &  $b$  are inputs push the elements when ' $c$ ' is the input pop.

E, z<sub>0</sub>/z<sub>0</sub>

b, b/bb  
C, b/E



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

$$\delta(q_0, a, z_0) = (q_0, a z_0)$$

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

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

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

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

$$\delta(q_1, C, a) = (q_2, \epsilon)$$

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

$$\delta(q_2, a, z_0) = (q_3, z_0)$$

Moves for abc<sup>2</sup>

$$(q_0, abcc, z_0) \xrightarrow{} (q_0, bcc, a z_0)$$

$$\xrightarrow{} (q_1, cc, ba z_0)$$

$$\xrightarrow{} (q_2, c, a z_0)$$

$$\xrightarrow{} (q_2, \epsilon, z_0)$$

$$\xrightarrow{} (q_3, \epsilon, z_0)$$

When input ended, the MC was in an accepting state q<sub>3</sub>; ∴ the string abc<sup>2</sup> gets accepted.

Q8

Design a PDA for language abc<sup>i</sup> b<sup>j</sup> c<sup>k</sup> i, j, k ≥ 0 and moves for aabbcc jk.

Soh: Let L = {ε, aabbcc, a<sup>i</sup>b<sup>j</sup>c<sup>k</sup> | i=j=k}

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

$$\delta(q_0, a, z_0) \xrightarrow{} (q_0, a z_0)$$

$$\delta(q_0, b, z_0) \xrightarrow{} (q_0, b z_0)$$



$$\delta(q_0, a, a) = (q_0, aa) \quad (\text{Push all } a's)$$

$$s(q_0, b, a) = (q_1, \epsilon) \quad \text{if } b \text{ encountered.}$$

$$S(q_1, B, \alpha) (= q_1, \varepsilon) \quad \text{pop. \& go to } q_1$$

$s(a, b, z_0) = (q_1, b z_0)$  at  $q_1$  if only  $b$ 's push

$$S(q_1, B, b) = \delta(q_1, BB) \quad \text{if cis isom } B^{\text{cap}}$$

$$\delta(q_1, c, b) = (q_2, \varepsilon)$$

$$\delta(q_2, e, B) = (q_2, \varepsilon)$$

$$f(a, f(b)) = f(a, b)$$

$$f(q_2, \varepsilon, \lambda_0) = f(q_3, \lambda_0) \xrightarrow{q_3 \text{ is } \frac{\partial f}{\partial q} \neq 0} (q_1, \lambda_0) \xrightarrow{q_1 \text{ is } \frac{\partial f}{\partial q} \neq 0} (q_2, \lambda_0)$$

10.000000 10.000000 10.000000 10.000000

$S(q_0, aabbcc, 20) \models (q_0, aabbcc, a20)$

$\vdash (q_0, bbbbcc, aa20)$

1- (q<sub>1</sub>, Bbbcc, a<sub>20</sub>)

$\vdash (q_1, \underline{bbcc}, z_0)$

$\vdash (q_1 \text{ Bcc}, B2)$

$\vdash (a_1, cc, bb2)$

$\vdash (q_2, c, B20)$

$$I - (q_2, \epsilon)$$

$$1 - (q_3, z_0)$$

Q.9 Design a PDA to accept strings such that  
 $L = \{a^i b^j | i \neq j\}$  & show the moves on  
 the string  $a^2 b^3$ . b, aL

Soln:  $L = \{a, aabb, ab\}^*$

$B, a \in E$

$$\delta(q_0, a, z_0) = (q_0, a z_0)$$

$$\delta(q_0, a, a) = (q_0, aa) \quad a, a/a/a \quad \epsilon, \epsilon/20$$

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

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

$$\delta(q_1 b, \epsilon, a) = (q_2, \epsilon) \text{ for } j=0$$

- This problem is of the same type as the previous one.



Date \_\_\_\_\_  
Page No. \_\_\_\_\_

**Q11** Design a PDA to accept  
a string of them  $S a^{2n} b^n \mid n \geq 0$

Soln:  $\lambda = \{\epsilon, aab, aaadbb, \dots\}$

$$\begin{aligned}\delta(q_0, \epsilon, z_0) &= (q_0, z_0) \\ \delta(q_0, a, z_0) &= (q_1, z_0) \\ \delta(q_0, b, z_0) &= (q_0, z_0) \\ \delta(q_1, a, a) &= (q_0, a) \\ \delta(q_1, b, a) &= (q_1, \epsilon) \\ \delta(q_1, \epsilon, z_0) &= (q_2, z_0)\end{aligned}$$

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

$$\delta(q_0, a, z_0) = (q_1, z_0) \rightarrow \text{first } a$$

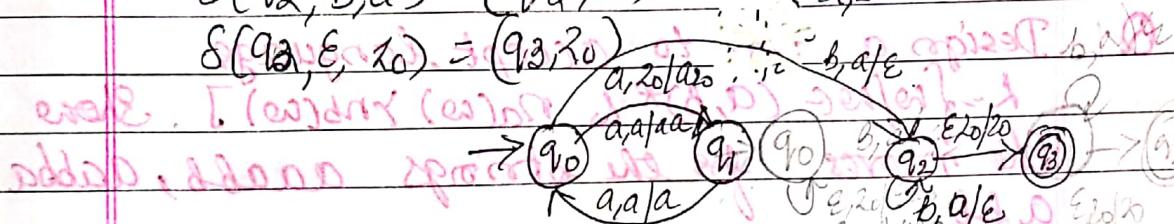
$$\delta(q_1, a, a) = (q_0, a) \rightarrow \text{skip second } a$$

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

$$\delta(q_0, b, a) = (q_2, \epsilon)$$

$$\delta(q_2, b, a) = (q_2, \epsilon)$$

$$\delta(q_2, \epsilon, z_0) = (q_3, z_0)$$



**Q12** Design a PDA for  $\lambda = \text{wwr over } \Sigma = \{0, 1\}$ . Show the moves on the string 1111.

Soln:

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

$$\delta(q_0, 0, z_0) = (q_0, 0z_0)$$

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

$$\delta(q_0, 0, 0) = (q_0, 00)$$

$$\delta(q_0, 1, 1) = (q_0, 11)$$

$$\delta(q_0, 0, 1) = (q_0, 01)$$

$$\delta(q_0, 1, 0) = (q_0, 10)$$

$$\delta(q_0, 0, 0) = (q_1, \epsilon)$$

$$\delta(q_0, 1, 1) = (q_1, \epsilon)$$

$$\delta(q_1, 0, 0) = (q_1, \epsilon)$$

$$(\delta(q_1, \epsilon, z_0) = (q_2, z_0))$$

### Moves

$S(q_0, 111, \epsilon_20) \vdash (q_0, 111, \epsilon_20)$  for priority 1  
 $\vdash (q_0, 11, \epsilon_20)$   
 $\vdash (q_0, 1, \epsilon_20)$   
 $\vdash (q_1, \epsilon, \epsilon_20)$   
 $\vdash ((q_1, \epsilon), (\epsilon, \epsilon_20))$

(1)  $\rightarrow$  (2)  $\rightarrow$

for priority 2  
 $\vdash (q_1, \epsilon, \epsilon_20) \vdash (q_1, \epsilon, \epsilon_20)$   
 $\vdash ((q_1, \epsilon), (\epsilon, \epsilon_20))$   
 $\vdash (q_1, \epsilon, \epsilon_20) \vdash (q_1, \epsilon, \epsilon_20)$   
 $\vdash ((q_1, \epsilon), (\epsilon, \epsilon_20))$

Q.13 Design a PDA to accept language  
 $L = \{a^m b^n c^m | (a,b)^* \in n\alpha(\omega) \wedge m\beta(\omega)\}$ . Show  
the moves for the strings aaabb, aabba,  
abb.

Soln: 12.  $L = \{aab, aabba, \dots\}$  for 3rd QM in npda  
 $\vdash 111$  priority 3 for nbs  $n\alpha(\omega) \wedge m\beta(\omega)$

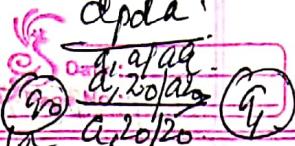
$\delta(q_0, a, \epsilon_20) = (q_0, a_20)$  OR  
 $\delta(q_0, b, \epsilon_20) = (q_0, b_20) \rightarrow (q_0, \epsilon, a_20) \rightarrow (q_1, \epsilon, a_20)$   
 $\delta(q_0, a, a) = (q_0, aa_20)$   
 $\delta(q_0, b, a) = (q_0, \epsilon) \rightarrow (q_0, a_20)$   
 $\delta(q_0, b, b) = (q_0, bb_20) \rightarrow a, a|aa_20$   
 $\delta(q_0, a, b) = (q_0, \epsilon), \delta(q_0, b, \epsilon) \rightarrow q$   
 $\delta(q_1, a, \epsilon_20) = (q_1, \epsilon_20) \rightarrow (q_1, b_20)$   
 $\delta(q_1, \epsilon, a) = (q_1, a) \rightarrow (q_1, b, b_20)$

### Moves

(1) aaabb

$\delta(q_0, aaabb, \epsilon_20) \vdash (q_0, aabb, a_20)$   
 $\vdash (q_0, abb, aa_20)$   
 $\vdash (q_0, b, aa_20)$   
 $\vdash (q_1, b, aa_20)$   
 $\vdash (q_1, \epsilon, a_20)$   
 $\vdash (q_1, a)$

dpda



(ii) aabba

$$\begin{aligned} \delta(q_0, aabba, z_0) &\vdash (q_0, abba, a_20) & a, b / \epsilon \\ &\vdash (q_0, bba, aa_20) & b, b / bb \\ &\vdash (q_1, ba, a_20) & b, a / \epsilon. \end{aligned}$$

(iii) abb

$$\delta(q_0, abb, z_0) \vdash (q_0, bb, a_20)$$

$$\vdash (q_0, b, z_0)$$

$\vdash (q_1, \epsilon, b_20)$  - Halted so rejected.

Q14 Design a PDA for palindrome (odd length also).

Soln:  $L = \{010, 00100, 10101, \dots\}$

6: Extra transition need to include for  $\in Q12$ .

$$\delta(q_0, 1, z_0) = (q_2, z_0)$$

$\delta(q_0, 0, z_0) = (q_2, z_0)$  } if contains only one  $\epsilon$

$$\delta(q_0, 1, 0) = (q_1, 0)$$

$\delta(q_0, 1, 1) = (q_1, 1)$  } '1' is at middle.

$$\delta(q_0, 0, 0) = (q_1, 0)$$

$$\delta(q_0, 0, 1) = (q_1, 1)$$

} 0 is at middle

101

## Applications of Context free grammar

1. Parsers : To design programming language compilers & interpreters CFGs are very important. Usually languages are balanced. Paranthesis which can be expressed using CFG. Arithmetic and conditional expression along with various operators can also be easily expressed using CFG. Using CFG we can check syntax not semantics.
2. YACC parser - generator Yet another compiler. Compiles a program in Unix which efficient parser can be generated. The input to this program command in CFG but represented using different notation.
3. Markup language The most familiar markup language HTML major function are: for creating links between documents and describing the look of a document. The structure of a part of HTML can be with some tags.
4. XML & document type definitions XML describes the semantics of the text.