PAGE NO:
DATE:

UNIT-3

Context free grammar

Grammar: A grammar G is a 4 tuple machine where G=(V, T, P, S) where V is set of variables or non terminals
T is set of terminals
P is set of productions
S is start symbol.

Ex:- Aditi ate slowly.
noun verb adverb

<sentence> ⟶ <noun><verb><adverb>
<noun> ⟶ Aditi
<verb> ⟶ ate
<adverb> ⟶ slowly

In the above grammar 'sentence' is the start symbol.

Types of grammar

There are 4 types of grammar:

1) Type 0 grammar

Type 0 grammar is a unrestricted grammar or phrase structured grammar. if all the productions are of the form $\alpha \longrightarrow \beta$ where $\alpha \in (V \cup T)^+$ at least one variable or terminal must be there and $\beta \in (V \cup T)^*$

Ex:- $S \longrightarrow aAb / \varepsilon$
variables

PAGE NO :
DATE :

$aA \longrightarrow aA/a$ capital letters are variables and small letters are terminals

terminals are just like the inputs we give in DFA or NFA

2) Type 1 grammar

A grammar $Gr = (V, T, P, S)$ is said to be Type 1 grammar or context sensitive grammar if all the productions are of the form $\alpha \longrightarrow \beta$ like Type 0 grammar but there is a restriction on the length of β.

The length of β must be at least as much as the length of α where α and $\beta \in (V \cup T)^*$ one or more

(another way to represent β)

Ex:- $S \longrightarrow aAb$

$aA \longrightarrow aA/ab$

α has length 2 at least 2 or more for b both

3) Type 2 grammar

A grammar $Gr = (V, T, P, S)$ is said to be a Type 2 grammar or context free grammar where all the productions are of the form $A \longrightarrow \alpha$ where $\alpha \in (V \cup T)^*$

only 1 variable

no restriction on length

Ex:- $S \longrightarrow aAb/\varepsilon$

$A \longrightarrow aA/ab$

4) Type 3 grammar

A grammar $G_4 = (V, T, P, S)$ is said to be a type 3 grammar or regular grammar if all the productions are of the form $A \rightarrow wB$ or $A \rightarrow w$ where $w \in T^*$ (zero or more terminals)

Ex:- $S \rightarrow aA/\varepsilon$

$A \rightarrow aA/ab$

* Type 0 grammar is a grammar for recursively enumerable machines used in unit 5 for tiring machines

* Type 1 grammar is used to construct linear block automata

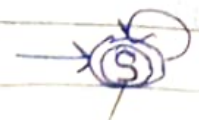
* Type 3 grammar is regular grammar used to construct finite automata.

* Type 2 grammar is used to construct push down automata.

Finite automata to Grammar

1) Obtain a grammar to generate strings consisting of any number of a's

→ First construct a DFA for this



S is start symbol

$\delta(S, a) = S$ is the delta transition.

$\delta(S, a) = S$

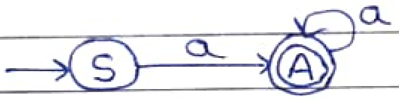
$S \longrightarrow aS$

$S \longrightarrow \varepsilon$ because start state itself is the final state so no input will also work

$S \longrightarrow aS/\varepsilon$

2) obtain a grammar to generate strings consisting of at least one 'a'.

→ The DFA is



delta transitions

$\delta(S, a) = A \Rightarrow S \longrightarrow aA$

$\delta(A, a) = A \Rightarrow A \longrightarrow aA/\varepsilon$ → we may stop after only 1 a also

3) obtain a grammar to generate strings consisting of any number of a's and b's

→ The DFA is

delta transitions

$\delta(S, a) = S$

$\delta(S, b) = S$

$\therefore S \longrightarrow aS/bS/\varepsilon$

4). Obtain a grammar to generate strings consisting of at least two a's

→ The DFA is

delta transitions

$\delta(S, a) = A \Rightarrow S \rightarrow aA$

$\delta(A, a) = B \Rightarrow A \rightarrow aB$

$\delta(B, a) = B \Rightarrow B \rightarrow aB/\varepsilon$

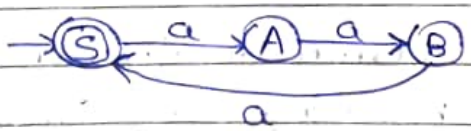
We can even write without constructing DFA by doing logical substitution (logically)

$S \rightarrow aa$

$S \rightarrow aS$

5) Obtain a grammar to generate strings consisting of multiple of 3a's.

→ The DFA is



delta transitions

$\delta(S, a) = A \Rightarrow S \rightarrow aA/\varepsilon$

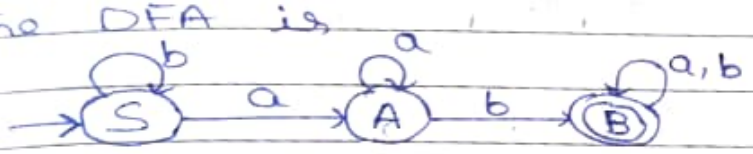
$\delta(A, a) = B \Rightarrow A \rightarrow aB$

$\delta(B, a) = S \Rightarrow B \rightarrow aS$

6) Generate a grammar for strings of a's & b's containing ab

PAGE NO :
DATE :

→ The DFA is



delta transition.

$\delta(S, a) = A \Rightarrow S \rightarrow aA$
$\delta(S, b) = S \Rightarrow S \rightarrow bS$
$\delta(A, a) = A \Rightarrow A \rightarrow aA$
$\delta(A, b) = B \Rightarrow A \rightarrow bB$
$\delta(B, a) = B \Rightarrow B \rightarrow aB / \varepsilon$
$\delta(B, b) = B \Rightarrow B \rightarrow bB / \varepsilon$

or

$S \rightarrow AabA$
$A \rightarrow aA / bA / \varepsilon$

$aA \rightarrow aaA \rightarrow aaaA$
$\rightarrow abA \rightarrow abbA$

7) generate / obtain a grammar for strings of a's and b's ending with ab

→

$S \rightarrow Aab$
$A \rightarrow aA / bA / \varepsilon$

8) obtain a grammar for the language $L = \{a^n b^n / n \geq 0\}$

→

$S \rightarrow aSb / \varepsilon$ → for $n = 0$

$aaSbb$

$aaaSbbb$

9) obtain a grammar for the language $L = \{a^n b^n / n \geq 1\}$.

→

$S \rightarrow aSb / ab$

PAGE NO :
DATE :

10) $L = \{ a^{n+1} b^n / n \geq 0 \}$

→

$S \rightarrow aSb / a$

11) $L = \{ a^{2n} b^n / n \geq 0 \}$

→

$S \rightarrow aaSb / \varepsilon$

12) $L = \{ ww^R / w \in (a+b)^* \}$

→

$S \rightarrow aSa / bSb / \varepsilon$

13) $L = \{ \underbrace{0^m 1^m}_{1} \underbrace{2^n}_{2} / m \geq 1, n \geq 0 \}$

let $0^m 1^m$ is A and 2^n is B

$S \rightarrow AB$
$A \rightarrow 0A1 / 01$
$B \rightarrow 2B / \varepsilon$

14) strings of balanced parenthesis of $(), \{\}, []$

→

$S \rightarrow (S) / \{S\} / [S] / \varepsilon$

15) $L = \{ a^{n+2} b^m / m > n, n \geq 0 \}$

→ $n = 0 \Rightarrow aab$
$n = 1 \Rightarrow aaabb$
$n = 2 \Rightarrow aaaabbb$ Minimum required

$S \rightarrow aAB$
$A \rightarrow aAb / a$
$B \rightarrow bB / b$

PAGE NO :
DATE :

16) $L = \{ a^n b^{n-3} / n \geq 3 \}$

$n = 3 \Rightarrow aaa$

$n = 4 \Rightarrow aaaab$ after 3 a's we have equal no. of a's & b's

$n = 5 \Rightarrow aaaaabb$

$S \rightarrow aaaA$

$A \rightarrow aAb/\varepsilon$

17) $L = \{ \underset{A}{a^n} \underset{B}{b^n c^m} / n, m \geq 0 \}$

→ $S \rightarrow AB$

$A \rightarrow aAb/\varepsilon$

$B \rightarrow cB/\varepsilon$

18) strings of 0's and 1's having substring 3 consecutive 0's

→ $(0+1)^* 000 (0+1)^*$

$S \rightarrow A000A$

$A \rightarrow 0A/1A/\varepsilon$

19) strings of a's and b's containing not more than 3a's

→ $b^*(a+\varepsilon)b^*(a+\varepsilon)b^*(a+\varepsilon)b^*$

$S \rightarrow BABABAB$

$B \rightarrow bB/\varepsilon$

$A \rightarrow a/\varepsilon$

Derivation Tree or Parse Tree

Let $G_1 = (V, T, B, S)$ be a context free grammar. The derivation tree is defined with the following properties.

1) root has the label 'S'

2) Every vertex has a label ($V \cup T \cup \varepsilon$)
3) leaf node has the label T and Interior vertex has the label 'V'
4) Let (if a vertex is labelled 'A' and if $X_1, X_2, \ldots X_n$ are children of 'A' from left then) $A \longrightarrow X_1, X_2, \ldots X_n$ be the production P_N

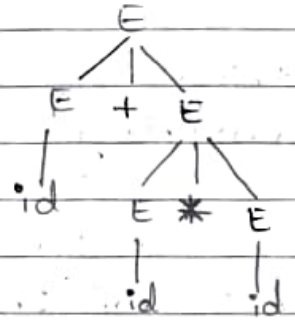
Leftmost Derivation

The leftmost derivation to get the string id + id * id is

$E \longrightarrow E + E$

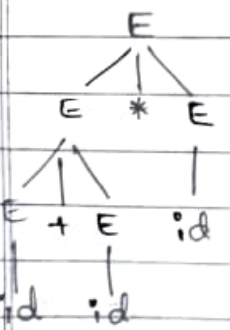
$E \longrightarrow E * E$

$E \longrightarrow id$

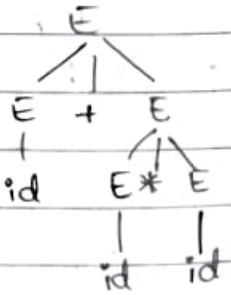


Rightmost derivation

The right most derivation to get the string id + id * id



or



Ambiguous grammar

A grammar $G = (V, T, B, S)$ is said

PAGE NO :
DATE :

to be ambiguous grammar if there exists atleast one string $w \in T^*$ for which two or more different parse trees exist by applying either leftmost or rightmost derivation

1) Is the grammar ambiguous?

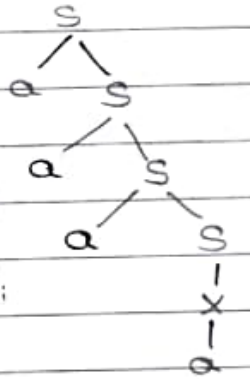
$S \rightarrow aS/X$

$X \rightarrow aX/a$

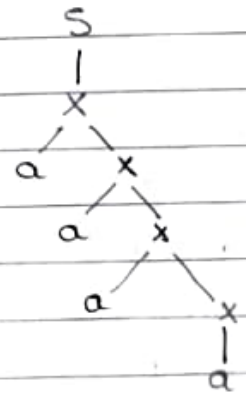
→

S	
aS	$S \rightarrow aS$
aaS	$S \rightarrow aS$
aaaS	$S \rightarrow aS$
aaaX	$S \rightarrow X$
aaaa	$X \rightarrow a$

⇒



S	
X	$S \rightarrow X$
aX	$X \rightarrow aX$
aaX	$X \rightarrow aX$
aaaX	$X \rightarrow aX$
aaaa	$X \rightarrow a$



Trees are different so it is ambiguous grammar

2) Is the grammar ambiguous?

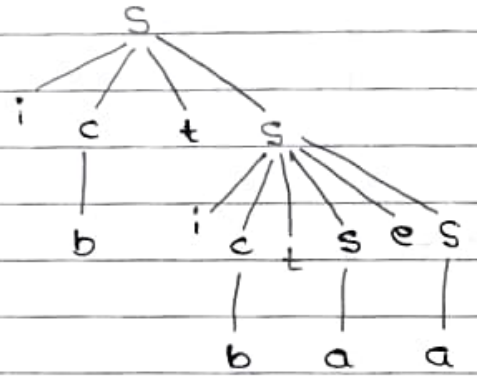
$S \rightarrow ictS/ictSeS/a$

$C \rightarrow b$

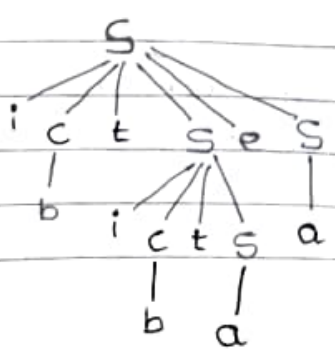
PAGE NO :
DATE :

→

S	
ictS	S → ictS
ibtS	c → b
ibtictSeS	S → ictSeS
ibtibtSeS	c → b
ibtibtaeS	S → a
ibtibtaea	S → a



S	
ictSeS	S → ictSeS
ibtSeS	c → b
ibtictSeS	S → ictS
ibtibtSeS	c → b
ibtibtaeS	S → a
ibtibtaea	S → a



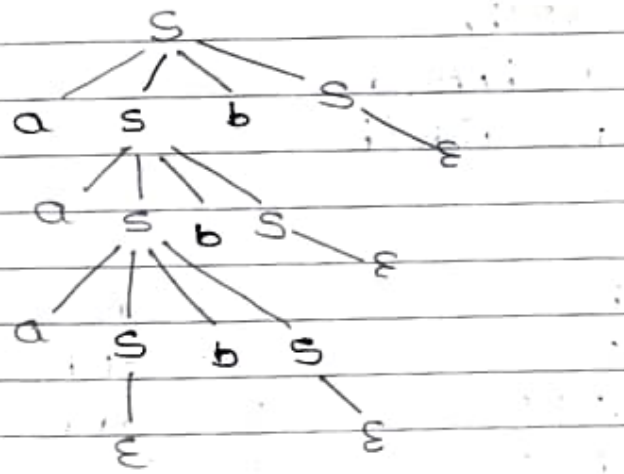
The trees are different so the grammar is ambiguous

PAGE NO :
DATE :

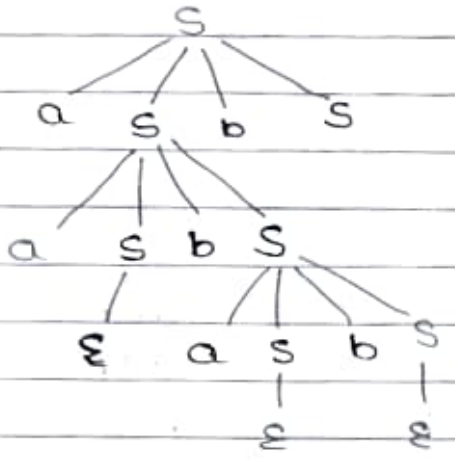
3) $S \rightarrow aSbS$
$S \rightarrow bSaS$
$S \rightarrow \varepsilon$

→

S	
aSbS	$S \rightarrow aSbS$
aaSbSbS	$S \rightarrow aSbS$
aabSaSbSbS	$S \rightarrow bSaS$
aabaSbSbS	$S \rightarrow \varepsilon$
aababSbS	$S \rightarrow \varepsilon$
aababbS	$S \rightarrow \varepsilon$
aababb	$S \rightarrow \varepsilon$



S	
aSbS	$S \rightarrow aSbS$
aaSbSbS	$S \rightarrow aSbS$
aabSbS	$S \rightarrow \varepsilon$
aabaSbSbS	$S \rightarrow aSbS$
aababSbS	$S \rightarrow \varepsilon$
aababbS	$S \rightarrow \varepsilon$
aababb	$S \rightarrow \varepsilon$



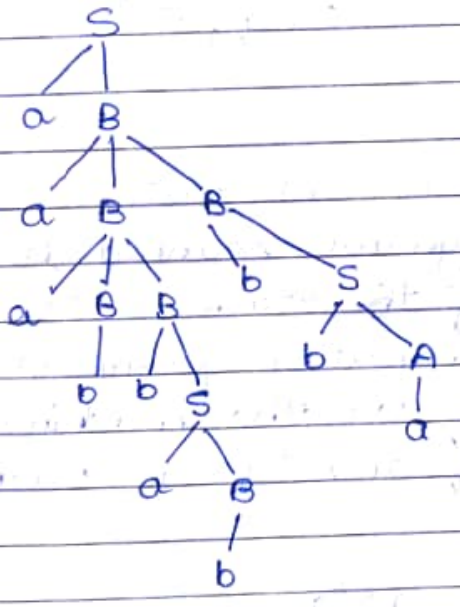
Different trees : the grammar is ambiguous

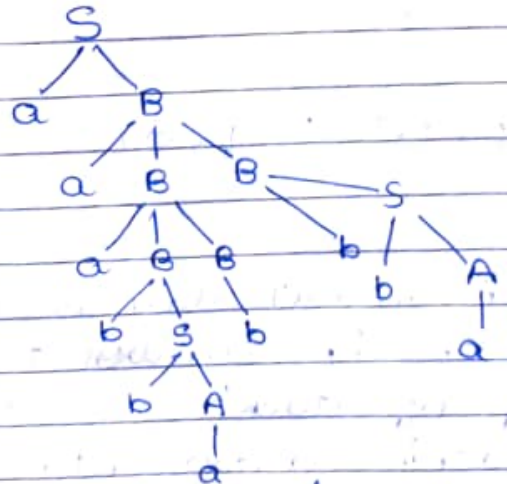
4) $S \longrightarrow aB/bA$
$A \longrightarrow aS/bAA/a$
$B \longrightarrow bS/aBB/b$

S	
aB	$S \rightarrow aB$
aaBB	$B \rightarrow aBB$
aaaBBB	$B \rightarrow aBB$
aaabBB	$B \rightarrow b$
aaabbSB	$B \rightarrow bS$
aaabbaBB	$S \rightarrow aB$
aaabbabB	$B \rightarrow b$
aaabbabbS	$B \rightarrow bS$
aaabbabbbA	$S \rightarrow bA$
aaabbabbba	$A \rightarrow a$

S	
aB	$S \rightarrow aB$

aaBB	B → aBB
aaaBBB	B → aBB
aaabSBB	b → bS
aaabbAbB	S → bA
aaabbaBB	A → a
aaabbabB	B → b
aaabbabbS	B → bS
aaabbabbbA	S → bA
aaabbabbba	A → a





Trees are different ∴ the grammar is ambiguous.

PAGE NO :
DATE :

PushDown Automata (PDA)

A PDA is a six tuple machine
$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ where

$Q \rightarrow$ set of finite states in the machine
$\Sigma \rightarrow$ set of inputs
$\Gamma \rightarrow$ set of stack alphabets

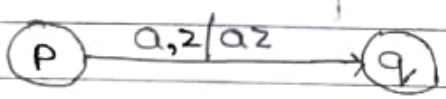
$\delta \rightarrow Q \times (\Sigma \cup \varepsilon) \times \Gamma \longrightarrow \Gamma^* \text{ or } (Q \times \Gamma^*)$

(current state, top of stack)

1) $\delta(p, a, z) = (q, az)$ $\quad z \Rightarrow \frac{a}{z}$

(input symbol; push operation)

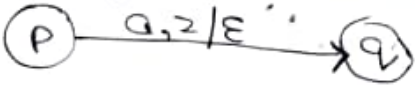
If the current state is 'p' and input symbol scanned is 'a' and if top of the stack is 'z' then there is a transition from state p to state q or the machine enters state q and 'a' is pushed onto the stack



2) $\delta(p, a, z) = (q, \varepsilon)$

(pop operation)

If the current state is p and 'a' is the i/p symbol scanned and top of stack is z then the machine enters into state q and the topmost element 'z' is popped

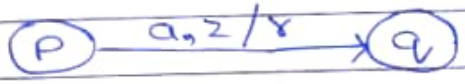


PAGE NO :
DATE :

or deleted from the stack.

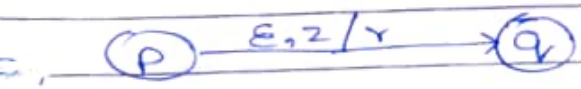
3) $\delta(p, a, z) = (q, r)$ (replace)

If the current state is 'p' and the input symbol scanned is 'a' and if top of stack is z then the machine enters into state 'q' and the topmost element 'z' is replaced by 'r'



4) $\delta(p, \varepsilon, z) = (q, r)$

If the current state is 'p' and the input symbol scanned is 'ε' (empty string) and if top of the stack is 'z' then the machine enters into the state 'q' and the topmost symbol 'z' in the stack is replaced by 'r'



5) $\delta(p, \varepsilon, \varepsilon) = (q, z)$ (push)

If the current state is 'p' and the input symbol scanned is 'ε' (empty string) and when the stack is empty then the machine enters into state 'q' and 'z' is pushed into the stack

PAGE NO :
DATE :

(p) —ε, ε/z→ (q)

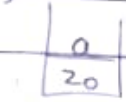
here also we will design machines but with the help of stack

1) $L = \{a^n b^n / n \geq 1\}$ obtain a PDA for this

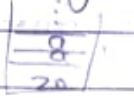
→ initially,

$\delta(q_0, a, z_0) = (q_0, az_0)$

when stack is empty, top of stack is denoted by z_0

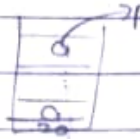


$\delta(q_0, a, a) = (q_0, aa)$ this line can be used to push any no. of a's into the stack.



Suppose all the a's are empty and we got our first b, we start push

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



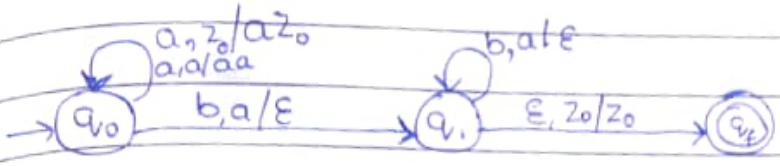
$\delta(q_1, b, a) = (q_1, \varepsilon)$ this can be used to pop any no. of a's from stack suppose there are equal no of a's and b's then when b's are exhausted stack will be empty.

$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

write the above in sentence form also

Now draw machines for above δ transitions.

PAGE NO :
DATE :



Example:-

$\delta(q_0, aaabbb, z_0)$

scan the first 'a' and put into top of stack

$= (q_0, aabbb, az_0)$ (push)

scan the second 'a' and put it into top of stack

$= (q_0, abbb, aaz_0)$ (push)

$= (q_0, bbb, aaaz_0)$

$= (q_1, bb, aaz_0)$

$= (q_1, b, az_0)$

$= (q_1, \varepsilon, z_0)$

$= (q_f, \varepsilon, z_0)$

2) $L = \{wcw^R / w \in (a+b)^*\}$

wabaa

w^R aaba

abaacaaba

push reading pop

here starting symbol can be a or b (two cases)

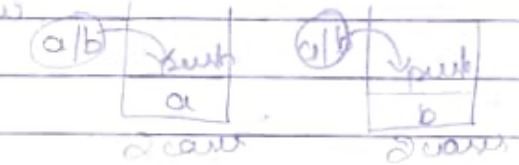
$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

a	b
z_0	z_0

PAGE NO :
DATE :

Now



total 4 cases to address

If top of stack is a

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

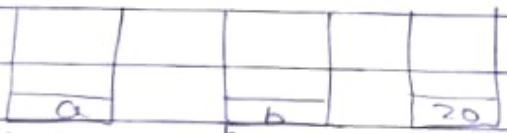
$\delta(q_0, b, a) = (q_0, ba)$

If top of stack is b

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

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

now from above cases we can complete entire string w
now when we get 'c' we have 3 cases for top of stack



a on TOS

b on TOS

w was empty string

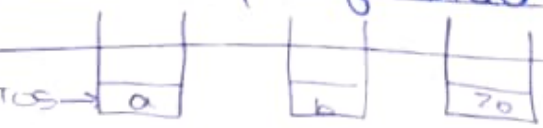
$\delta(q_0, c, a) = (q_1, a)$

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

$\delta(q_0, c, Z_0) = (q_1, Z_0)$

replace with same symbol on top of stack

now we go to string w^R
here also we have 3 cases for top of stack



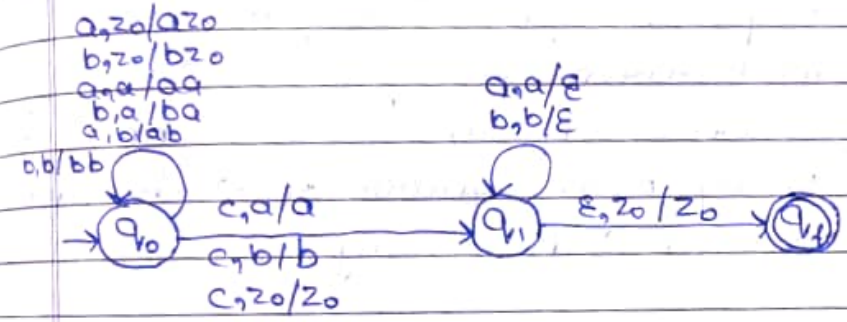
PAGE NO :
DATE :

$\delta(q_1, a, a) = (q_1, \varepsilon)$ pop the elements

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

after popping all the elements TOS becomes z_0

$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$. This indicates that w and w^R are correct



Example :- abaaCaaba

$\delta(q_0, abaaCaaba, z_0)$

$= (q_0, baaCaaba, az_0)$

$= (q_0, aaCaaba, baz_0)$

$= (q_0, aCaaba, abaz_0)$

$= (q_0, Caaba, aabaz_0)$

$= (q_1, aaba, aabaz_0)$

$= (q_1, aba, abaz_0)$

$= (q_1, ba, baz_0)$

PAGE NO :
DATE :

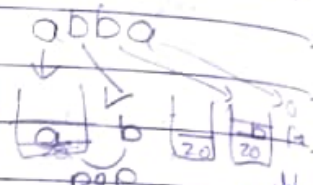
$= (q_1, a, az_0)$

$= (q_f, \varepsilon, z_0)$

~~$= (q_f, \varepsilon, z_0)$~~

e) Obtain a PDA for language
$L = \{ w : n_a(w) = n_b(w), w \in (a+b)^* \}$

→ Add first symbol to stack and if you get opposite symbol pop it in between if stack becomes empty (for ex:- abba) add very next symbol to stack and proceed same way.



here there is no specific order so stat remains same q_0

→

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

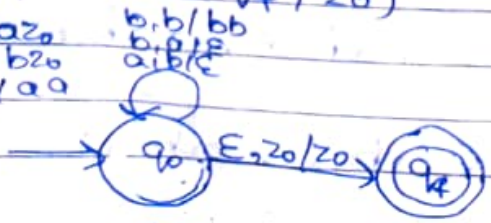
$\delta(q_0, a, a) = (q_0, aa)$

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

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

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

$\delta(q_0, \varepsilon, z_0) = (q_f, z_0)$



PAGE NO :
DATE :

Ex:- abab

$\delta(q_0, abab, z_0)$

$= (q_0, bab, az_0)$

$= (q_0, ab, z_0)$

$= (q_0, b, az_0)$

$= (q_f, \varepsilon, z_0)$

3) $L = \{a^n b^{2n} / n \geq 0\}$

→ for every one 'a' scanned push 2a's into the stack so we will have equal no. of a's and b's then for every b pop one a from the stack.

$\delta(q_0, a, z_0) = (q_0, aaz_0)$

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

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

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

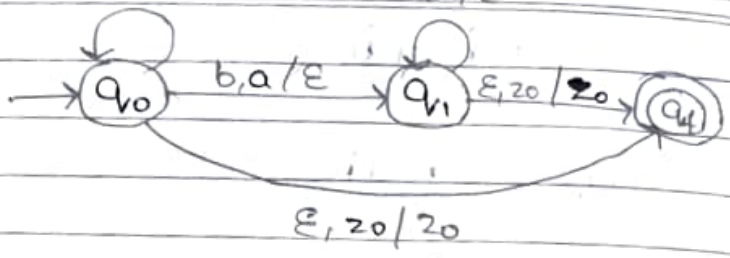
$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

since there can be no string also $n=0$ so

$\delta(q_0, \varepsilon, z_0) = (q_f, z_0)$

a, z0 / a a z0
a, a / a a a

b, a / ε



Example:- a a b b b b

$\delta(q_0, aabbbb, z_0)$

$= (q_0, abbbb, aaz_0)$

$= (q_0, bbbb, aaaaz_0)$

$= (q_1, bbb, aaaz_0)$

$= (q_1, bb, aaz_0)$

$= (q_1, b, az_0)$

$= (q_f, \varepsilon, z_0)$

Obtain a PDA for balancing parenthesis (,), [,]
for every open brace push it into the stack and for close brace pop it out of the stack

$\delta(q_0, (, z_0) = \delta(q_0, (z_0)$
$\delta(q_0, (, () = \delta(q_0, ((}$
$\delta(q_0, (, [) = \delta(q_0, ([)$
$\delta(q_0, [, z_0) = \delta(q_0, [z_0)$

PAGE NO :
DATE :

$\delta(q_0, [, C) = \delta(q_0, [C)$
$\delta(q_0, [, [) = \delta(q_0, [[)$
$\delta(q_0,), C) = \delta(q_1, \varepsilon)$
$\delta(q_0,], [) = \delta(q_1, \varepsilon)$
$\delta(q_1,), C) = \delta(q_1, \varepsilon)$
$\delta(q_1,], [) = \delta(q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) =$

Deterministic PDA (push down Automata)
Let $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ be a pushdown automata. The PDA is Deterministic if

- $\delta(q, a, z)$ has only one element
  $\delta(q_0, a, a) = (q_1, aa)$ → should be only once.
- If $\delta(q, \varepsilon, z)$ is not empty then $\delta(q, a, z)$ should be empty.
  $\delta(q_0, \varepsilon, z) = (q_0, z_0)$ if one is
  $\delta(q_0, a, z_0) = (q_0, \varepsilon)$ there then the other must also be there

WCW^R

Ex:- $\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, b, z_0) = (q_0, bz_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$

PAGE NO :
DATE :

$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, c, z_0) = (q_1, z_0)$
$\delta(q_0, c, a) = (q_1, a)$
$\delta(q_0, c, b) = (q_1, b)$
$\delta(q_1, a, a) = (q_1, \varepsilon)$
$\delta(q_1, b, b) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$ → non empty but no other can with (q_1, z_0)

first all satisfied

The above transitions satisfy both the cases ∴ it is a Deterministic PDA

$a^n b^n / n \geq 1$

1) $\delta(q_0, a, z_0) = (q_0, az_0)$ → non-Empty
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \varepsilon)$
$\delta(q_1, b, a) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_2, \varepsilon)$ → empty

first condition satisfied

The above transition satisfies both cases ∴ it is a Deterministic PDA

$n_a w / n_a \in w = n_b \in w\}$

3) $\delta(q_0, a, z_0) = (q_0, az_0)$ must have been empty
$\delta(q_0, b, z_0) = (q_0, bz_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, \varepsilon)$
$\delta(q_0, b, a) = (q_0, \varepsilon)$
$\delta(q_0, \varepsilon, z_0) = (q_1, z_0)$ → not empty

→ Non-Deterministic PDA

$n_a \in w > n_b \in w$

4) $\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, b, z_0) = (q_0, bz_0)$

PAGE NO :
DATE :

$\delta(q_0, a, a) = (q_0, aa)$ → should have been empty
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, \varepsilon)$
$\delta(q_0, b, a) = (q_0, \varepsilon)$
$\delta(q_0, \varepsilon, a) = (q_1, a)$ → not empty

→ Non-Deterministic PDA

$a^n b^{2n} / n \geq 1$

e) $\delta(q_0, a, z_0) = (q_0, aaz_0)$ — first case satisfied
$\delta(q_0, a, a) = (q_0, aaa)$ — since empty we cannot write and case so it is Deterministic
$\delta(q_0, b, a) = (q_1, \varepsilon)$
$\delta(q_1, b, a) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_2, \varepsilon)$

→ Deterministic PDA

d) $L = \{ wcw^R / w \in (a+b)^* \}$
$\delta(q_0, a, z_0) = (q_0, az_0)$ } must have been empty
$\delta(q_0, b, z_0) = (q_0, bz_0)$
$\delta(q_0, a, a) = (q_0, aa), (q_1, \varepsilon)$ ✓ state has multiple transitions
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$
$\delta(q_0, b, b) = (q_0, bb), (q_1, \varepsilon)$ ✓
$\delta(q_1, a, a) = (q_1, \varepsilon)$
$\delta(q_1, b, b) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$ → non Empty

→ Non-deterministic PDA

Equivalence of PDA

CFG to PDA

Procedure of conversion

* Convert the grammar into GNF (Greibach Normal Form)

$A \to a\alpha$
GNF form

PAGE NO :
DATE :

step2:- Let 'q_0' be the start state and 'z_0' be initial symbol on stack. without consuming any input push the start symbol s onto the stack and change the state to q_1

$$\delta(q_0, \varepsilon, z_0) = (q_1, Sz_0)$$

step3:- For each production of the form $A \longrightarrow a\alpha$

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

(remain in same state)

$A \rightarrow aB$

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

step4:- finally in state q_1, without consuming any input, change the state to q_f and the transition can be written in the form

$$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$$

Q) $S \longrightarrow aABC$
$A \longrightarrow aB/a$
$B \longrightarrow bA/b$
$C \longrightarrow a$

→ first $\delta(q_0, \varepsilon, z_0) \longrightarrow (q_1, Sz_0)$

$A \longrightarrow a\alpha$	$\delta(q_1, a, A) \longrightarrow (q_1, \alpha)$
$S \longrightarrow aABC$	$\delta(q_1, a, S) = (q_1, ABC)$
$A \longrightarrow aB$	$\delta(q_1, a, A) = (q_1, B)$

PAGE NO :
DATE :

$A \rightarrow a$	$\delta(q_1, a, A) = (q_1, \varepsilon)$
$B \rightarrow ba$	$\delta(q_1, b, B) = (q_1, A)$
$B \rightarrow b$	$\delta(q_1, b, B) = (q_1, \varepsilon)$
$C \rightarrow a$	$\delta(q_1, a, C) = (q_1, \varepsilon)$

last step $\delta(q_1, \varepsilon, Z_0) = (q_f, Z_0)$

After this write machine M=
and write all δ transitions
also one example

(d) $S \rightarrow aABB/aAA$
$A \rightarrow aBB/a$
$B \rightarrow bBB/A$ replace with aBB so it will be in CNF form
$C \rightarrow a$
convert the grammar into PDA

→ first step is
$\delta(q_0, \varepsilon, Z_0) \rightarrow (q_1, SZ_0)$

$A \rightarrow a\alpha$	$\delta(q_1, a, A) \rightarrow (q_1, \alpha)$
$S \rightarrow aABB$	$\delta(q_1, a, S) = (q_1, ABB)$
$S \rightarrow aAA$	$\delta(q_1, a, S) = (q_1, AA)$
$A \rightarrow aBB$	$\delta(q_1, a, A) = (q_1, BB)$
$A \rightarrow a$	$\delta(q_1, a, A) = (q_1, \varepsilon)$
$B \rightarrow bBB$	$\delta(q_1, b, B) = (q_1, BB)$

PAGE NO :
DATE :

$B \to A$ (CNF form) $B \to aBB$	$\delta(q_1, a, B) = (q_{1,} BB)$
don't write this $B \to A$ $B \to a$ CNF form	$\delta(q_1, a, B) = (q_1, \varepsilon)$
$C \to a$	$\delta(q, a, C) = (q_1, \varepsilon)$

finally, $\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

S
aABB
aaBBB
aabBBBB
aabaBBB
aabaaBB
aabaaaB
aabaaaa

$\delta(q_0, aabaaaa, z_0)$
$\delta(q_1, aabaaaa, Sz_0)$
$\delta(q_1, abaaaa, ASz_0)$ → a replaced with A
$\delta(q_1, baaaa, AASz_0)$

PAGE NO :
DATE :

PDA to CFG

procedure of convertion

step 1:- The input symbol of PDA will be the terminal of CFG (small letters)

step 2:- If PDA moves from state q_i to q_j on consuming the input symbol 'a' when 'z' is on top of the stack then the non-terminal of CFG are the triplet of the form $q_i z q_j$

$$\delta(q_i, a, z) = (q_j, \dots)$$

$q_i z q_j$ S/A/B... $S \rightarrow aA$ (non-terminal part)

step 3:- If q_0 is the start state and q_f is the final state then the start symbol of CFG will be '$q_0 z q_f$'

step 4:- The production of CFG can be obtained from transitions of PDA

i) For each transition of the form

$\delta(q_i, a, z) = (q_j, AB)$ push

introduce the production

$$(q_i z q_k) \longrightarrow a(q_j A q_\ell)(q_\ell B q_k)$$

q_j and q_k will take all possible values of Q (all states present here $i \neq j$)

ii) $\delta(q_i, a, z) = (q_j, \varepsilon)$ pop

PAGE NO :
DATE :

$\delta(q_i, \varepsilon, z) = (q_i, \varepsilon)$

for transition of this form

$q_i Z q_j \longrightarrow a$

Suppose we have replace operation

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

we cannot directly convert it into grammar so replace statement with below statements

$\delta(q_i, a, A) = (q_n, \varepsilon)$
$\delta(q_n, \varepsilon, z) = (q_i, Az)$ } convert this into CFG

eg) $\delta(q_0, A, z) = (q_0, Az)$
$\delta(q_0, b, A) = (q_0, AA)$
$\delta(q_0, a, A) = (q_1, \varepsilon)$ } 2 states q_0, q_1

$\delta(q_i, a, z) = (q_j, AB)$	$q_i z q_k \rightarrow a (q_j A q_e)(q_e B q_k)$
• $\delta(q_0, a, z) = (q_0, Az)$	$q_0 z q_{0\,k=0} \rightarrow a(q_0 A q_{0\,e=0})(q_0 z q_{0\,k=0})$ / $a(q_0 A q_{0\,e=1})(q_1 z q_{0\,k=1})$
	$q_0 z q_{0\,k=1} \rightarrow a(q_0 A q_{0\,e=0})(q_0 z q_{1\,k=1})$ / $a(q_0 A q_{0\,e=1})(q_1 z q_{1\,k=1})$
• $\delta(q_0, b, A) = (q_0, AA)$	$q_0 A q_{0\,k=0} \rightarrow b(q_0 A q_{0\,e=0})(q_0 A q_{0\,k=0})$ / $b(q_0 A q_{0\,e=1})(q_1 A q_{0\,k=1})$
	$q_0 A q_{0\,k=1} \rightarrow b(q_0 A q_{0\,e=0})(q_0 A q_{1\,k=1})$ / $b(q_0 A q_{0\,e=1})(q_1 A q_{1\,k=1})$

PAGE NO :
DATE :

$\delta(q_i, a, z) = (q_j, \varepsilon)$ $\quad q_i z q_j \Rightarrow a$

$\delta(q_0, a, A) = (q_1, \varepsilon)$ $\quad q_0 A q_1 \rightarrow a$ & pop operation

a) $\delta(q_0, a, z) = (q_0, Az)$ push
$\delta(q_0, a, A) = (q_0, A)$ → replace
$\delta(q_0, b, A) = (q_1, \varepsilon)$ pop
$\delta(q_1, \varepsilon, z) = (q_2, \varepsilon)$

k and l have 4 values

$\delta(q_0, a, A) = (q_3, \varepsilon)$
$\delta(q_3, \varepsilon, z) = (q_0, Az)$

$\delta(q_i, a, z) = (q_j, AB)$ $\quad q_i z q_k \rightarrow (q_i A q_l)(q_l B q_k)$

$\delta(q_0, a, z) = (q_0, Az)$ $\quad q_0 z q_k \rightarrow a(q_0 A q_l)(q_l z q_k)$

$q_0 z q_0$ ($k=0$) $\rightarrow a(q_0 A q_0)(q_0 z q_0)$ ($l=0$)
$/\, a(q_0 A q_0)(q_1 z q_0)$ ($l=1$)
$/\, a(q_0 A q_0)(q_2 z q_0)$ ($l=2$)
$/\, a(q_0 A q_0)(q_3 z q_0)$ ($l=3$)

$q_0 z q_0$ ($k=1$) $\rightarrow a(q_0 A q_0)(q_0 z q_1)$ ($l=0$)
$/\, a(q_0 A q_1)(q_1 z q_1)$ ($l=1$)
$/\, a(q_0 A q_2)(q_2 z q_1)$ ($l=2$)
$/\, a(q_0 A q_3)(q_3 z q_1)$ ($l=3$)

$q_0 z q_2$ ($k=2$) $\rightarrow a(q_0 A q_0)(q_0 z q_2)$ ($l=0$)
$a(q_0 A q_1)(q_1 z q_2)$ ($l=1$)
$a(q_0 A q_2)(q_2 z q_2)$ ($l=2$)
$a(q_0 A q_3)(q_3 z q_2)$ ($l=3$)

$q_0 z q_3$ ($k=3$) $\rightarrow a(q_0 A q_0)(q_0 z q_3)$ ($l=0$)
$/\, a(q_0 A q_1)(q_1 z q_3)$ ($l=1$)
$/\, a(q_0 A q_2)(q_2 z q_3)$ ($l=2$)
$/\, a(q_0 A q_3)(q_3 z q_3)$ ($l=3$)

PAGE NO :
DATE :

$\delta(q_i, a, z) = (q_j, \varepsilon)$ $\quad q_i z q_j \to a$

- $\delta(q_0, a, A) = (q_3, \varepsilon)$ $\quad q_0 A q_3 \to a$
- $\delta(q_0, b, A) = (q_1, \varepsilon)$ $\quad q_0 A q_1 \to b$
- $\delta(q_1, \varepsilon, z) = (q_2, \varepsilon)$ $\quad q_1 z q_2 \to \varepsilon$

$\delta(q_i, a, z) = (q_j, AB)$ $\quad q_i z q_k \to a (q_j A q_l)(q_l B q_k)$

$\delta(q_3, \varepsilon, z) = (q_0, Az)$ $\quad q_3 z q_0 \to (q_0 A q_0)(q_0 z q_0)$

$k=0$, $l=0$

don't write ε

$/ \; (q_0 A q_0)(q_1 z q_0)$
$/ \; (q_0 A q_1)(q_2 z q_0)$
$/ \; (q_0 A q_2)(q_3 z q_0)$
$/ \; (q_0 A q_3)(q_3 z q_0)$

$q_3 z q_0 \to$ $k=1$
$(q_0 A q_0)(q_0 z q_1)$
$/ \; (q_0 A q_1)(q_1 z q_1)$
$/ \; (q_0 A q_2)(q_2 z q_1)$
$/ \; (q_0 A q_3)(q_3 z q_1)$

$q_3 z q_2 \to$ $k=2$
$(q_0 A q_0)(q_0 A q_2)$
$/ \; (q_0 A q_1)(q_1 A q_2)$
$/ \; (q_0 A q_2)(q_2 A q_2)$
$/ \; (q_0 A q_3)(q_3 A q_2)$

$q_3 z q_3 \to$ $k=3$
$(q_0 A q_0)(q_0 A q_3)$
$/ \; (q_0 A q_1)(q_1 A q_3)$
$/ \; (q_0 A q_2)(q_2 A q_3)$
$/ \; (q_0 A q_3)(q_3 A q_3)$