Date
29/11/22

Unit - III

Grammar

A grammar can be represented by four tuples :- V, T, P, S

V → Set of all variables / auxiliary symbols / non-terminal symbols
T → Set of all terminal
P → Set of all productions
S → Start symbol

$$P = \begin{Bmatrix} S \to aSB \\ S \to aB \\ B \to b \end{Bmatrix} \qquad V = \{S, B\}$$
$$T = \{a, b\}$$

Derivation :-

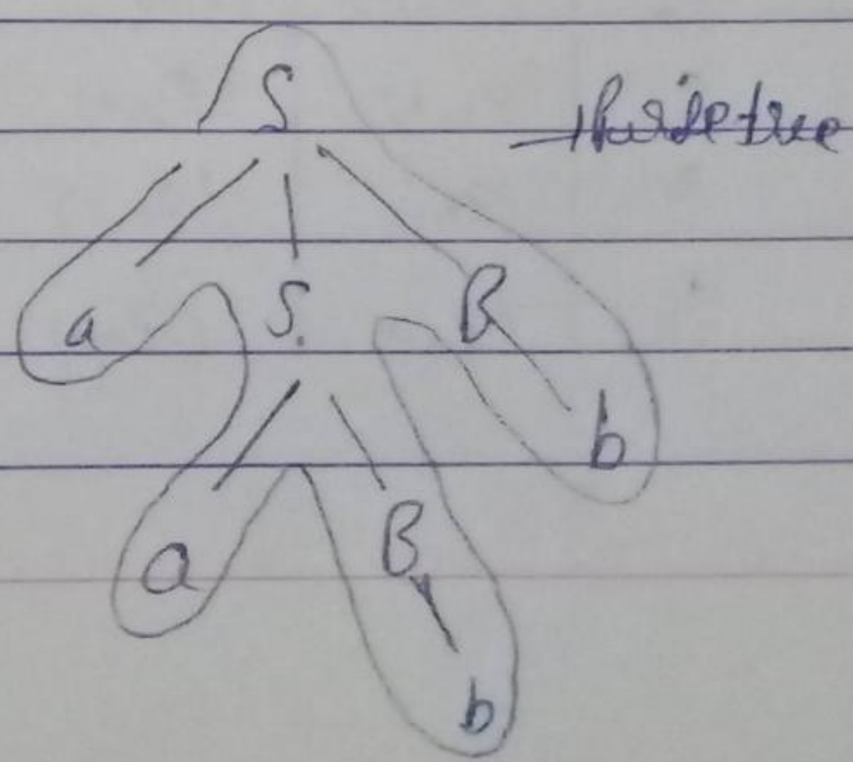
Deriving a string from the grammar starting from the start symbol. At every step only one variable will be replaced by its right hand side.

* If you start replacing the left most symbol first then it is called leftmost derivation.
* If you start replacing the right most symbol first then it is called right most derivation.
* Entire process is called derivation of string.
* Intermediate steps are known as sentential form or sequential form.

One of the way of representing the derivation is called derivation tree or parse tree.

e.g. aabb

$S \to aSB$
$\to aaBB$
$\to aabB$
$\to aabb$

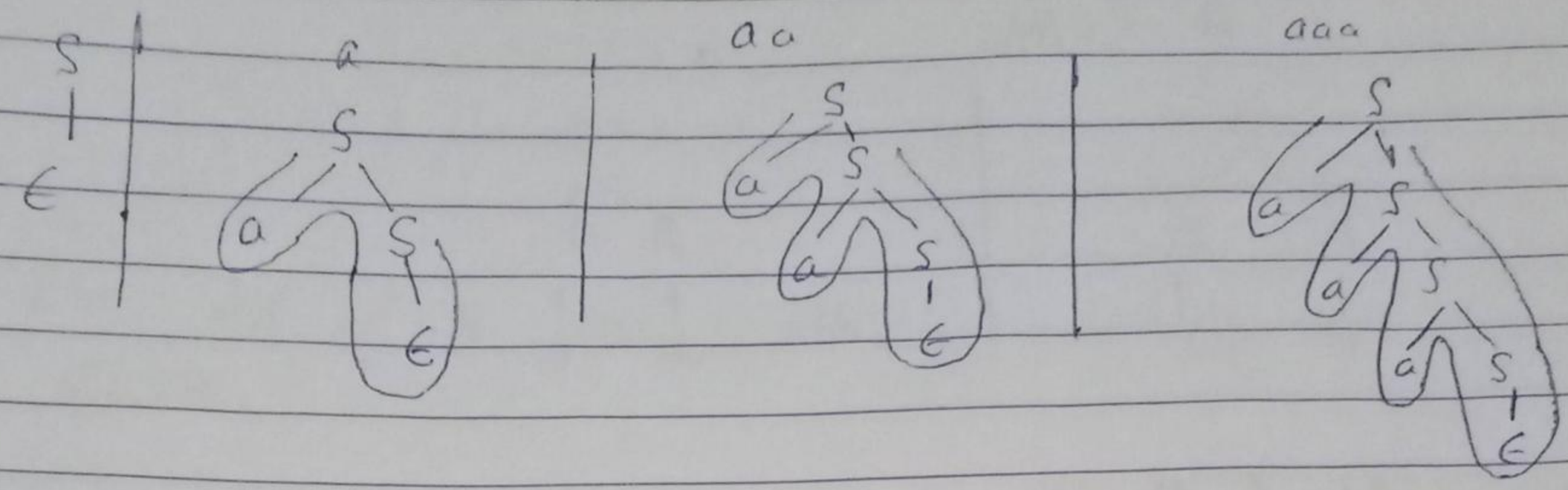


Parse tree

$L = a^n \mid n \geq 0$

Ans- $L = \{\epsilon, a, aa, aaa, \ldots\}$

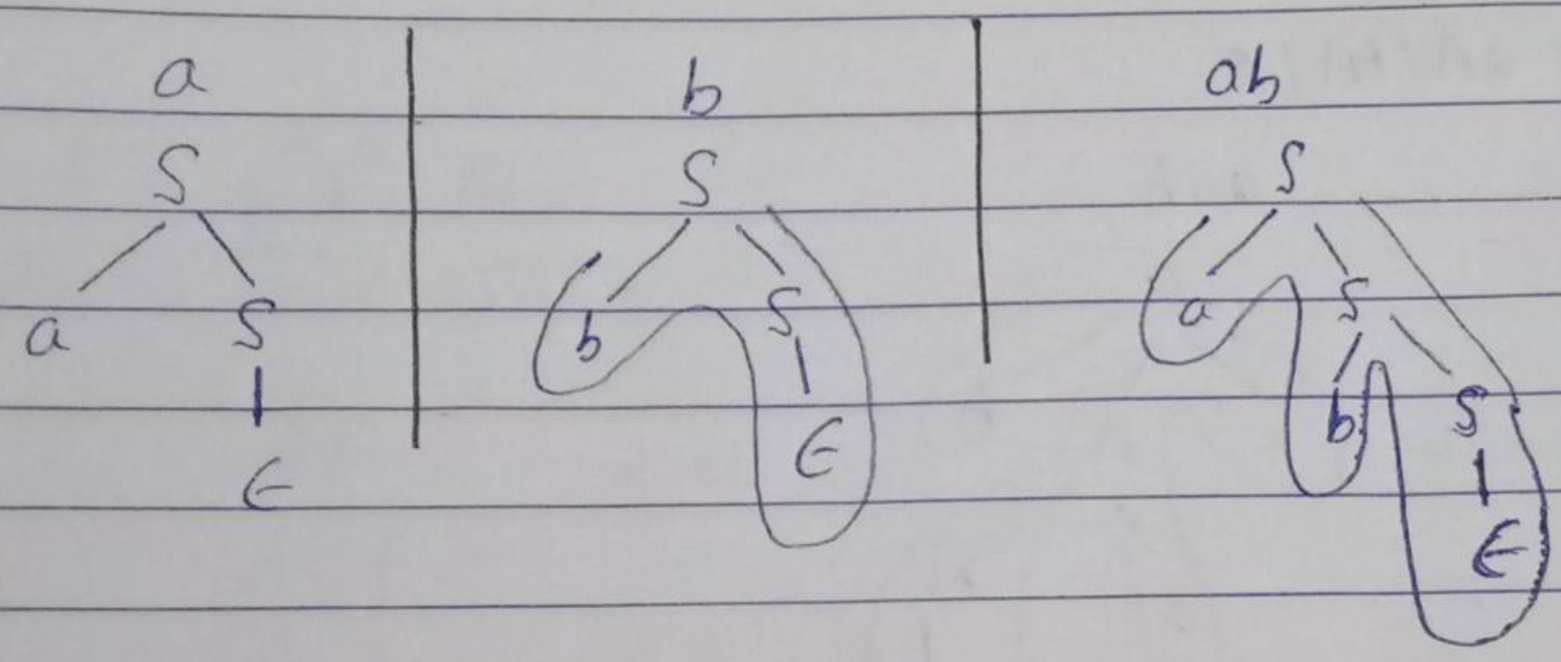
$S \rightarrow aS \mid \epsilon$



(2) $L =$ ~~Set of all~~ $(a+b)^*$

$L = \{\epsilon, a, b, aa, bb, aaa, \ldots\}$

$S \rightarrow aS \mid bS \mid \epsilon$



(3) $L =$ Set of all strings of length atleast 2. over $\{a, b\}$

Ans- $L = \{a, b, aa, ab, ba, bb, aaa, aba$

~~$S \rightarrow aS \mid bS \mid a$~~ $(a+b)(a+b)(a+b)^*$

$S \rightarrow AAB$

$A \rightarrow a \mid b$

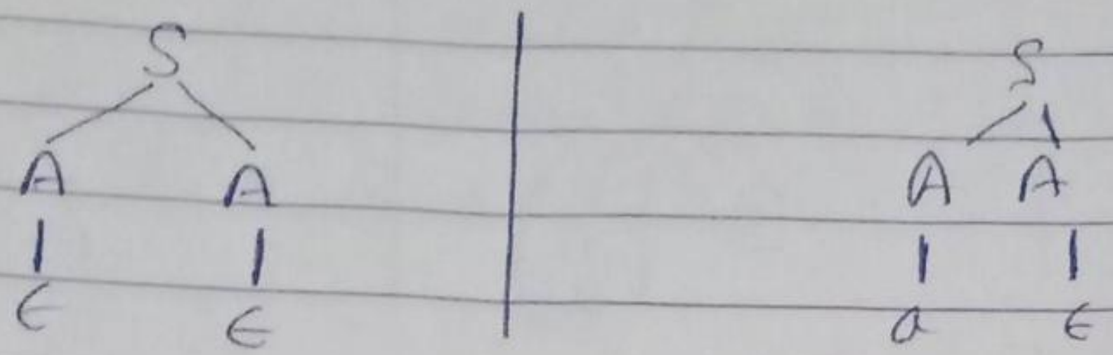
$B \rightarrow aB \mid bB \mid \epsilon$

(4) L = Set of all strings of length atleast 2.

Ans= R.E = $(a+b)^*(a+b+\epsilon)(a+b+\epsilon)$

$S \rightarrow AA$

$A \rightarrow a/b/\epsilon$

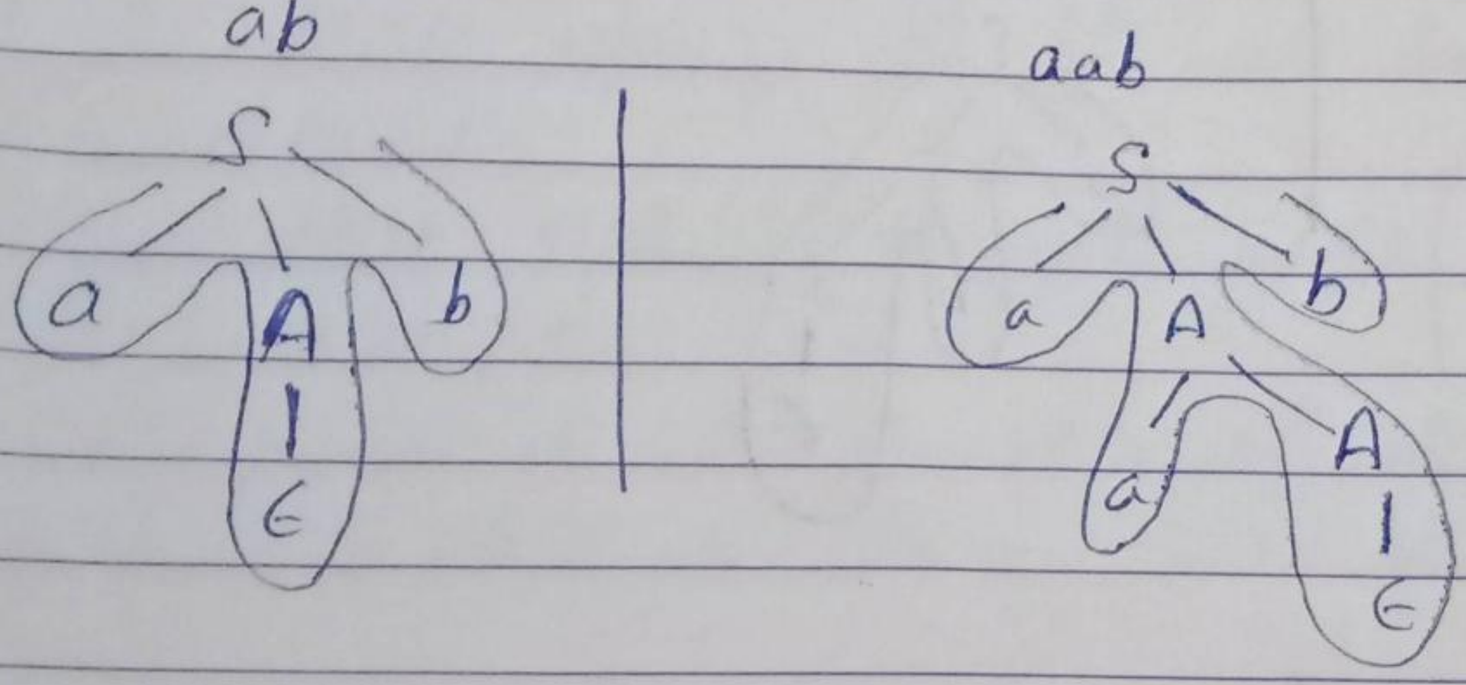


(5) L = Set of all strings starting with a and ending with b.

Ans= R.E = ~~$(ab)(a+$~~ $a(a+b)^*b$

$S \rightarrow aAb$

$A \rightarrow aA/bA/\epsilon$

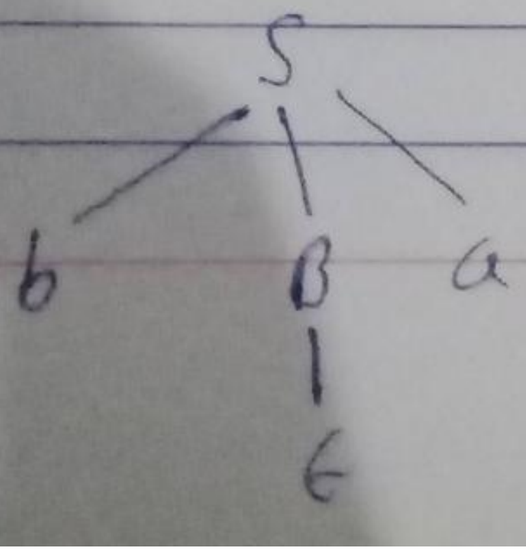


L = Set of all strings starting with b and ending with a.

Ans= R.E = $b(a+b)^*a$

$S \rightarrow bBa$

$B \rightarrow aA/bA/\epsilon$



⑦ L = Set of all strings starting with and ending with different symbols

Ans= $R.E = a(a+b)^*b + b(a+b)^*a$

$S \rightarrow aAb / bAa$

$A \rightarrow aA / bA / \epsilon$

⑧ L = Set of all strings starting and ending with same symbol.

Ans= $R.E = a(a+b)^*a + b(a+b)^*b + a + b + \epsilon$

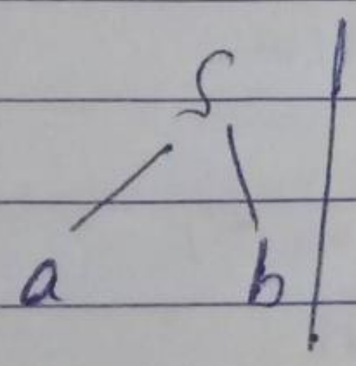
$S \rightarrow aAa / bAb / a / b / \epsilon$

$A \rightarrow aA / bA / \epsilon$

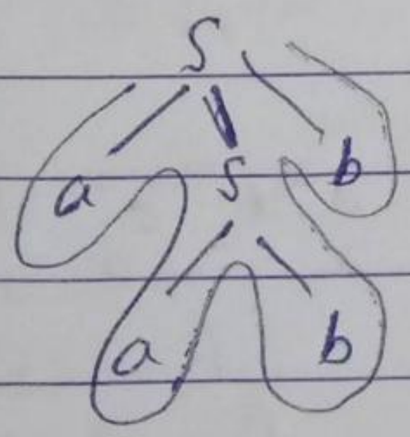
⑨ $L = a^n b^n, n \geq 1$

Ans= $S \rightarrow aSb / ab$

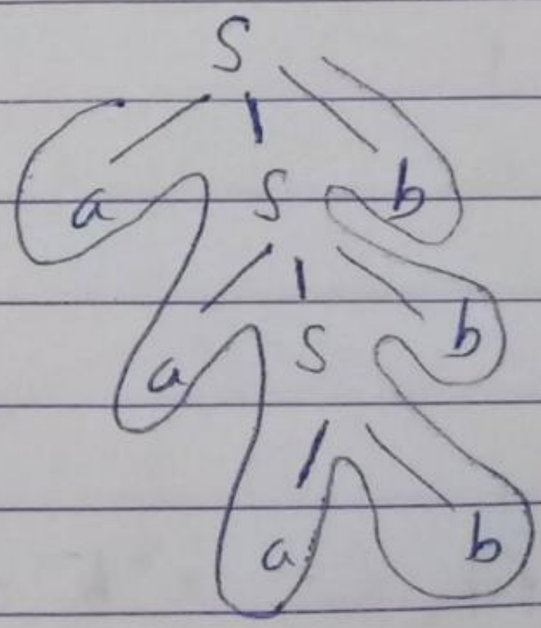
ab



aabb



aaabbb



⑩ $L = WW^R \cup WaW^R \cup WbW^R, W \in \{a,b\}^*$

Ans= $S \rightarrow aSa / bSb / a / b / \epsilon$

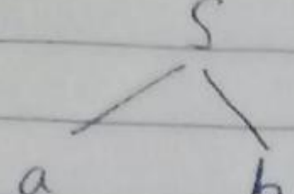
Note:- only if even length

$S \rightarrow aSa / bSb / \epsilon$

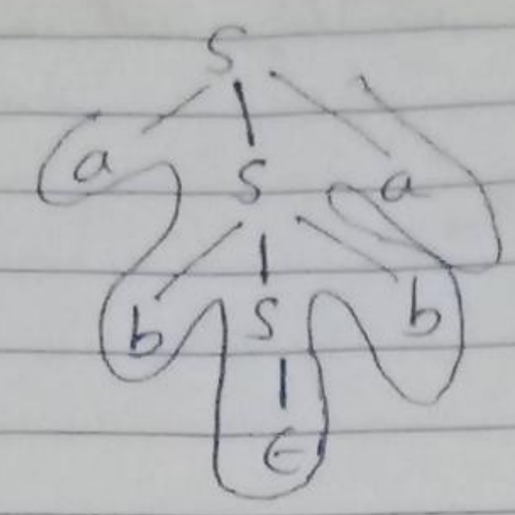
$S \rightarrow aSa / bSb / a / b$

a*
S → aS/ϵ

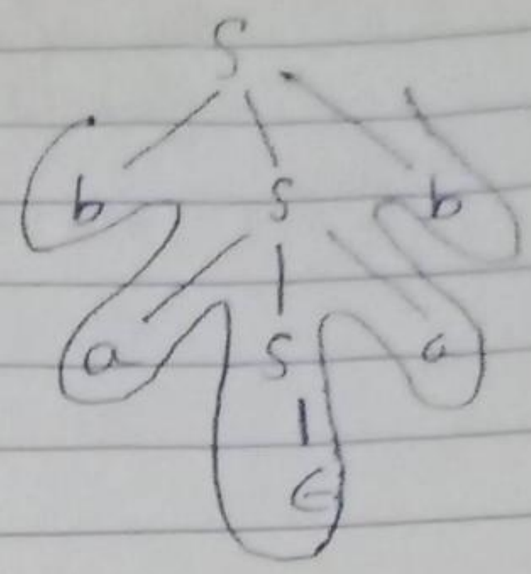
ab



abba



baab



(11) Construct a grammar over $\Sigma = \{a, b\}$ of even length string.

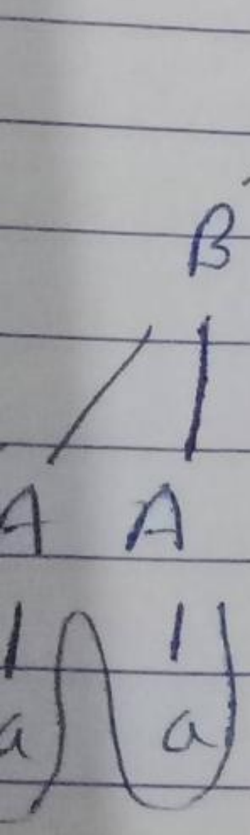
Ans= $R.E = ((a+b)(a+b))^*$

$S \rightarrow BS/\epsilon$

$B \rightarrow AA$

$A \rightarrow a/b$

aa



(12) $L = a^n b^m \ / n, m \geq 1$

Ans= $S \rightarrow AB$

$B \rightarrow bB/b$

$A \rightarrow aA/a$

(13) $L = a^n b^n c^m \ / n, m \geq 1$

Ans= $S \rightarrow AB$

$B \rightarrow cB/c$

$A \rightarrow aAb/ab$

(14) $L = a^n b^n c^m d^m \ / n, m \geq 1$

Ans= $S \rightarrow AB$

$B \rightarrow cBd/cd$

$A \rightarrow aAb/ab$

(15) $L = a^n b^m c^n \quad | n, m \geq 1$

Ans-

~~$S \rightarrow AB$~~

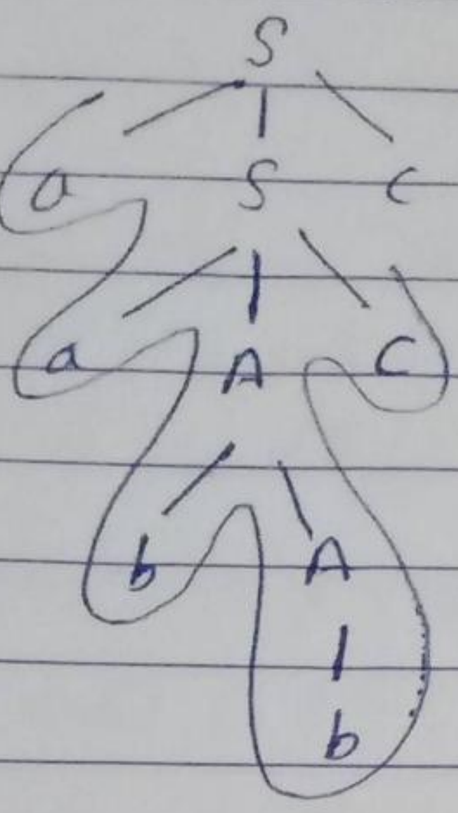
~~$B \rightarrow bB \mid b$~~

~~$A \rightarrow aAc \mid ac$~~

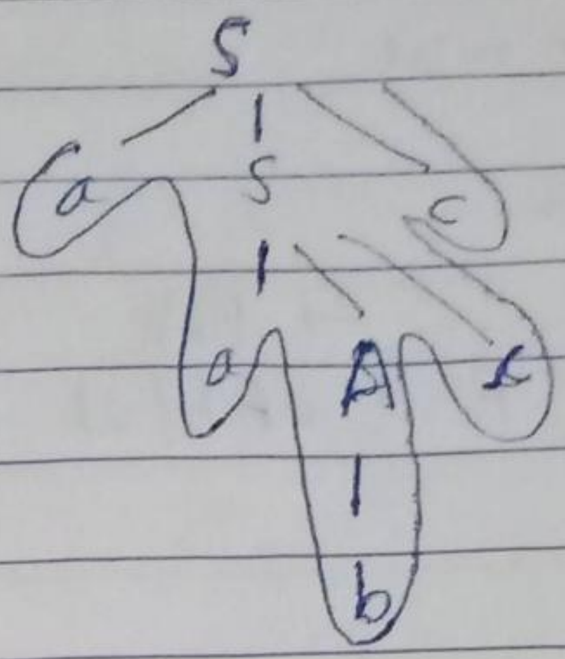
$S \rightarrow aSc \mid aAc$

$A \rightarrow bA \mid b$

aabbcc



aabcc



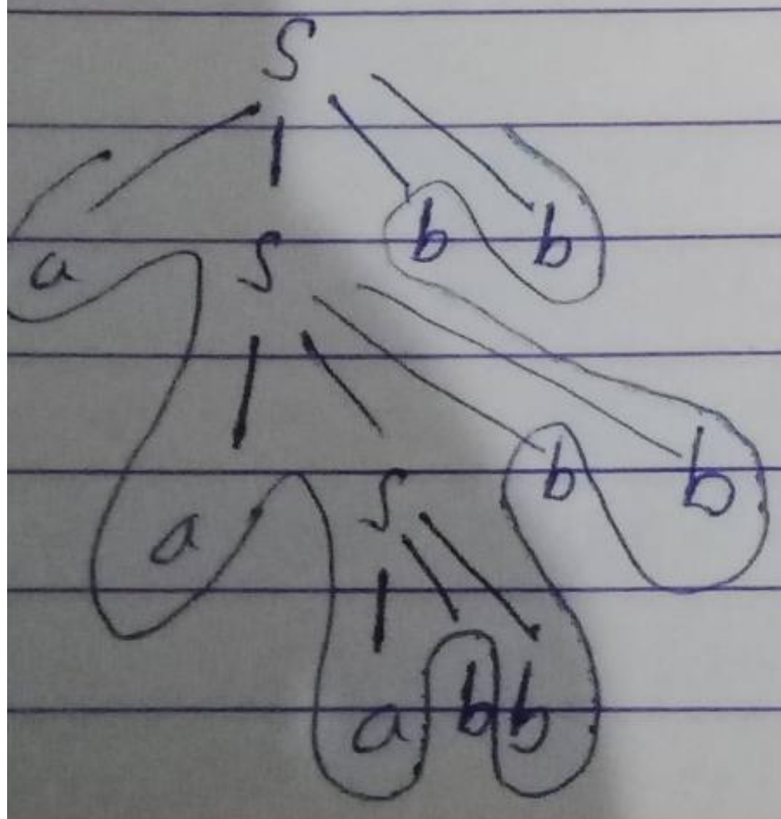
(16) $L = a^n b^{2n} \quad | n \geq 1$

Ans= ~~$S \rightarrow aSbA$~~

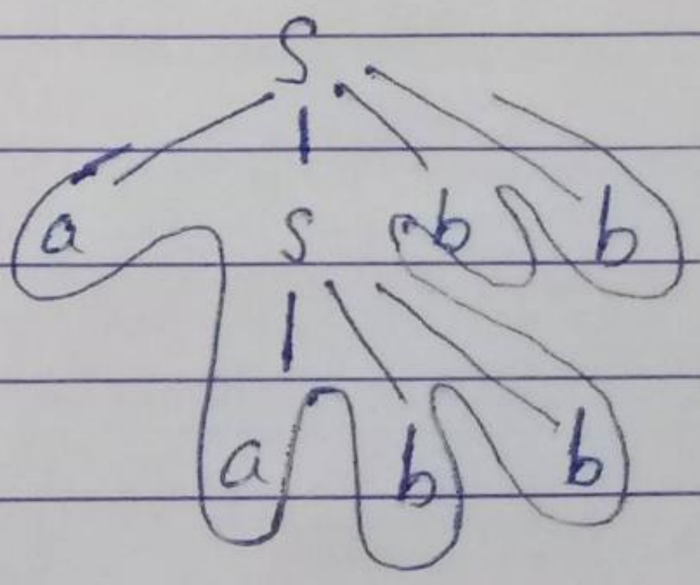
$S \rightarrow aSbb \mid abb$

~~$A \rightarrow aAb \mid ab^2$~~

aaabbbbbb



aabbbb



(17) $L = a^n b^m c^m d^n \mid n,m \geq 1$

Ans=

$S \to aSd \mid aAd$

$A \to bAc \mid bc$

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

Ans= $L = a^n a^m b^m c^n \mid n,m \geq 1$

$S \to aSc \mid aAc$

$A \to aAb \mid ab$

(19) $a^n b^{m+m} c^m \mid n,m \geq 1$

Ans= $L = a^n b^n \cdot b^m \cdot c^m$

$S \to BA$

~~$S \to aSc \mid a$~~ $B \to aBb \mid ab$

$A \to bAc \mid bc$

(20) $L = a^n b^m \cdot c^{m+n} \mid n,m \geq 1$

Ans= $L = a^n b^m \cdot c^m \cdot c^n$

$S \to aSc \mid aAc$

$A \to bAc \mid bc$

(21)

Classification of Grammar

According to Chomsky, Grammars are classified into four types:-

1) Type 3 Grammar ⟶ Regular Grammar
2) Type 2 Grammar ⟶ Context Free Grammar (CFG)
3) Type 1 Grammar ⟶ Context Sensitive Grammar (CSG)
4) Type 0 Grammar ⟶ Unrestricted Grammar (UG)

1) Type 3:- It can be written in the form:-

Right Linear Grammar (RLG)

$A \to \alpha B / \beta$

$\{A, B\} \in V$ (variables)

$\{\alpha, \beta\} \in T^*$ (Terminal)

e.g. $A \to aB/a$

$B \to aB/bB/a/b$

Left Linear Grammar (LLG)

$A \to B\alpha/\beta$

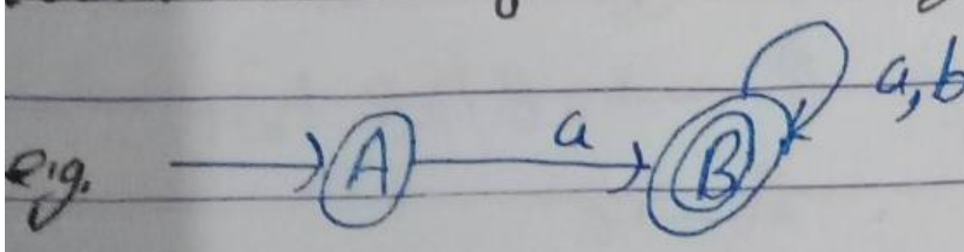
$\{A, B\} \in V$

$\{\alpha, \beta\} \in T^*$

e.g. $A \to Ba/a$

$B \to Ba/Bb/a/b$

Conversion of F.A to Regular Grammar

e.g. 

$\delta(A, a) = B$

$\delta(B, a) = B$

$\delta(B, b) = B$

$\delta(q, a) \to p$

$q \to ap$ (if q is not a final state)

$\delta(q, a) \to p$

$q \to ap/\epsilon$ (if q is a final state)

$A \to aB$

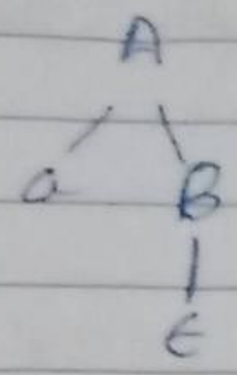
$B \to aB/bB/\epsilon$ (RLG)

$A \to Ba$

$B \to Ba/Bb/\epsilon$ (LLG) ✗

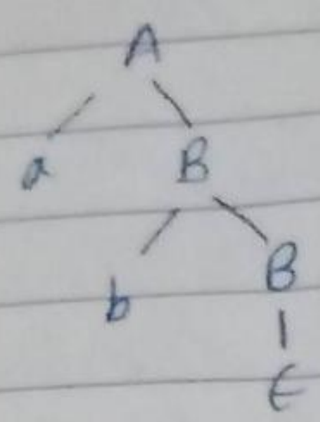
Note:- We cannot direct convert RLG into LLG.

a

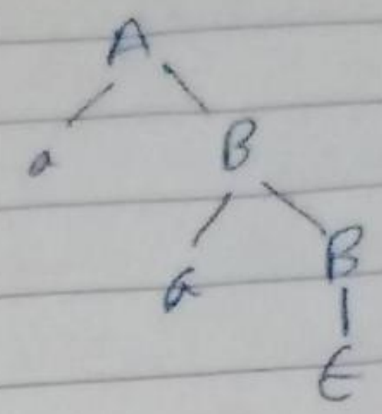


(RLG)

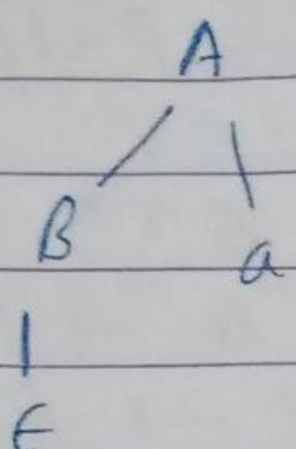
ab



aa

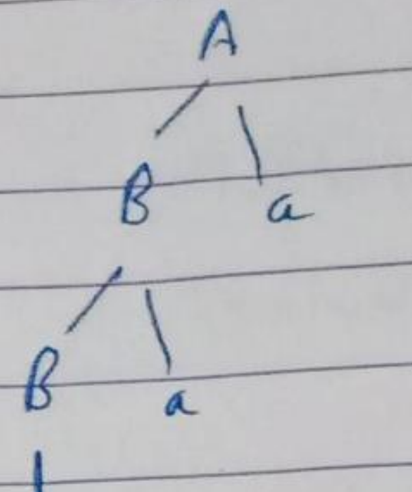


a



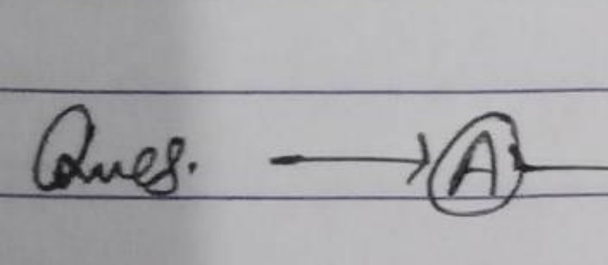
(LLG)

aa



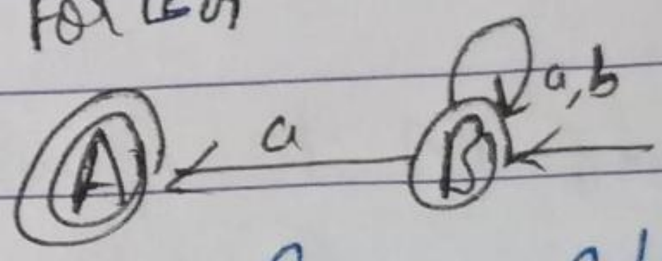
ϵ

ϵ

Ques. 

$\Rightarrow$ Reversal

For LLG



For RLG :-

$\delta(A,a) = B$

$\delta(B,a) = B$

$\delta(B,b) = B$

R-LG $\begin{cases} A \to aB \\ B \to aB/bB/\epsilon \end{cases}$

R.L.G $\begin{cases} B \to aB/bB/aA \\ A \to \epsilon \end{cases}$

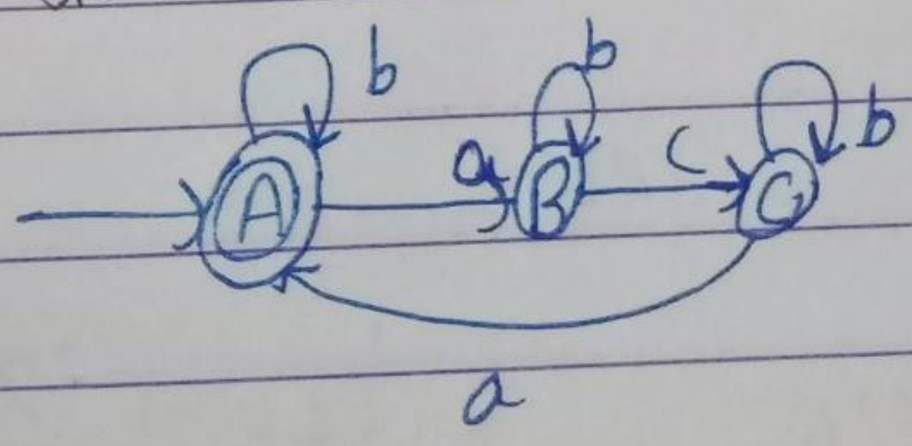
L.L.G $\begin{cases} B \to Ba/Bb/Aa \\ A \to \epsilon \end{cases}$

Conversion of Regular Grammar to F.A

$A \to aB/bA/b$

$B \to cc/bB$

$C \to aA/bC/a$



Type 2 Grammar:- If productions are of the form $A \to \alpha$ where $A \in V$, $\alpha \in (V \cup T)^*$ then such a grammar is called type 2 Grammar or context free grammar. Languages generated by context-free grammar are known as ~~lang~~ context-free languages and machine used as a acceptor for context free language is PDA (Push Down Automata).

$E \longrightarrow E+E / E*E / id$

$V = \{E\}$

$T = \{+, *, id\}$

To generate string id + id * id

OR

$E \to E*E$

$\to E+E*E$

$\to id+E*E$

$\to id + id*E$

$\to id + id * id$

$E \to E+E$

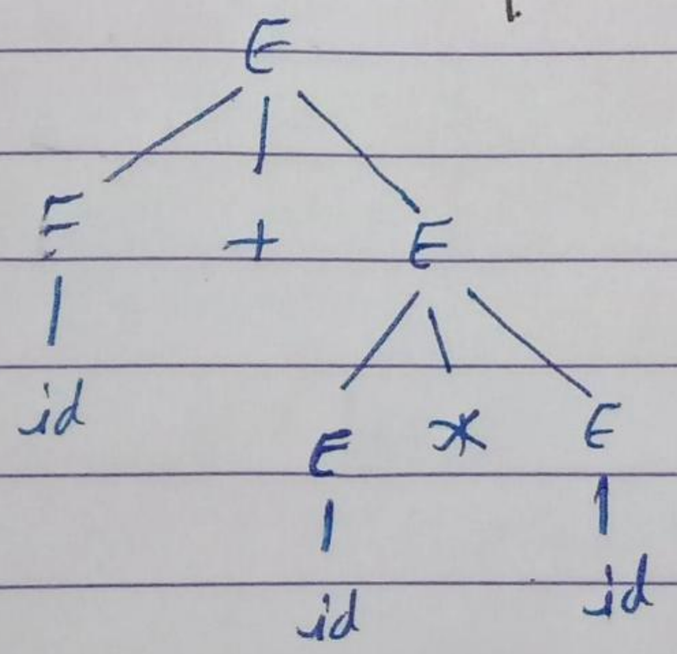
$\to E + E*E$

$\to id + E*E$

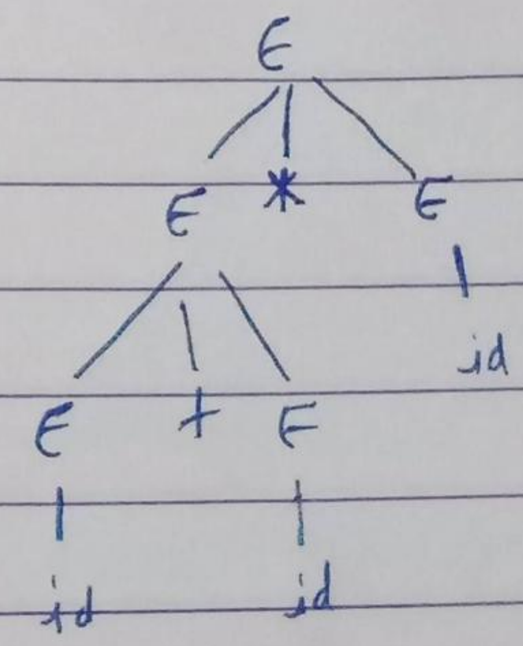
$\to id + id*E$

$\to id + id* id$

Parse tree:-



Parse tree



Right most derivation

$E \to E+E / E*E / id$

$E \to +EE / *EE / id$

Left most derivation

$E \to E+E / E*E / id$

$E \to EE+ / EE* / id$

For a given Grammar, for a given string if more than one LMD (Leftmost Derivation), RMD, Parse tree exist then we can say that grammar is ambiguous

1. $S \to aS/Sa/a$. To generate aa

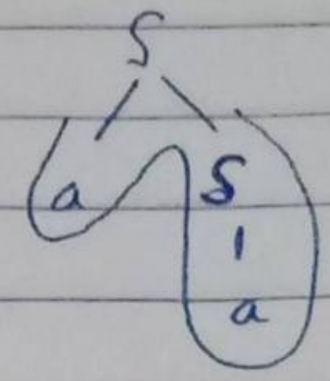
$S \to aS$

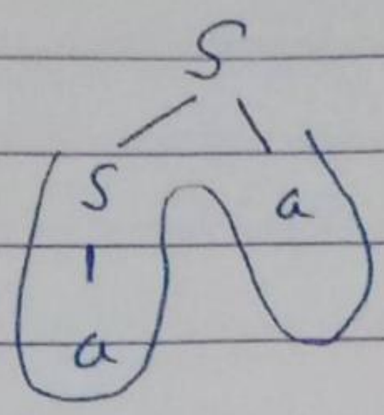
$\to aa$

OR

$S \to Sa$

$\to aa$





It is an ambiguous grammar.

2. $S \to aSbS / bSaS / \epsilon$ To generate abab

$S \to aSbS$

$\to abSaSbS$

$\to ab\epsilon a\epsilon b\epsilon$

$\to abab$

OR