

# 'CONTEXT FREE LANGUAGE'

- A language generated by CFG or accepted by M.
- Also defined by the same 4-tuples.
- Context FG has production rule of form.

$$\alpha \rightarrow \beta$$

$\alpha \in (V_n)^+$  non-terminal (Left side pas non-terminal hi hoga)

$\beta \in (V_n \cup \Sigma)^*$  set of terminal symbols.

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

$$S \rightarrow aAb \mid \epsilon$$

$$A \rightarrow aAb \mid \epsilon$$

eg:  $S \rightarrow aAb \Rightarrow a aAb b \Rightarrow a a aAb b b \Rightarrow a a a a b b b b \Rightarrow a a a a b b b b$

$$L = \{0^n 1^n : n \geq 2\}$$

$$S \rightarrow 000011 \mid 00S1 \mid \epsilon$$

$\epsilon$  agr  $S$  k saath  
x bna diya to  $\Delta$  the area  
ho jayegi.

$$S \rightarrow 00A1$$

$$00A1 \rightarrow 00$$

$$A \rightarrow 00B1 \mid \epsilon$$

$$B \rightarrow 00B1 \mid \epsilon$$

$$010$$

$$L = \{0^n 10^n : n \geq 1\}$$

$$S \rightarrow 0S011$$

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

$$S \rightarrow a \mid aSb \mid aS$$

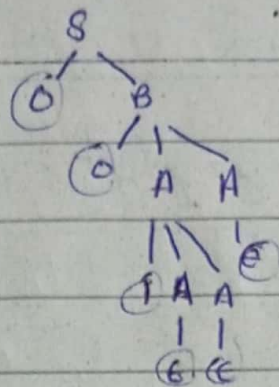
isse equal a b b  
generate honxe

or zyada a isse generate ho jayegi





## DERIVATION TREE

$$S \rightarrow OB$$
$$D \rightarrow 10A \text{ / } \epsilon \quad \Rightarrow$$
$$B \rightarrow \text{GAA}$$


→ For story 0010 →

→ The leaf nodes will represent the strings.

- There are two types of derivation trees.

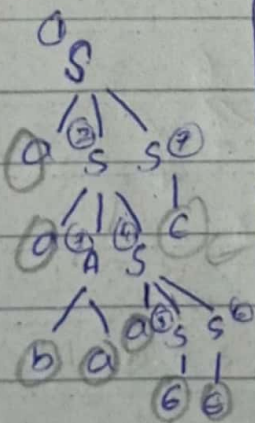
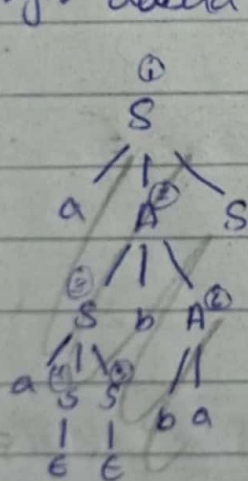
(i) Left Derivation Tree  $\Rightarrow$  is mein hum kamesha left most non-terminal ko expand karte.

② Right " "  $\Rightarrow$  "

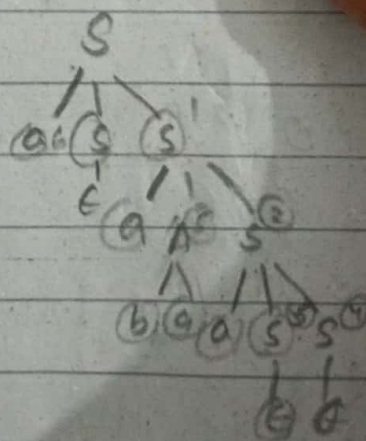
$$S \rightarrow aAS / aSS / \epsilon$$

$A \rightarrow SBA/ba$

String: adbaa



Left Dev. Tree



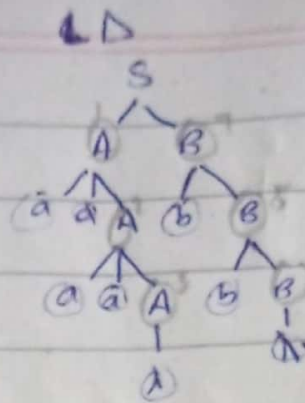
▲ Right Derivative or Tree.

$$S \rightarrow AB$$

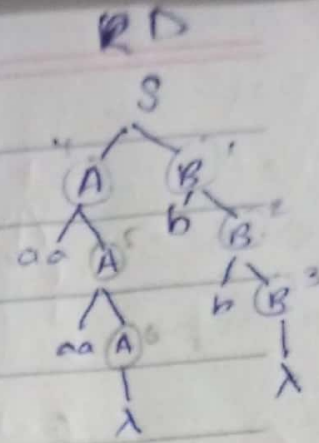
$$A \rightarrow aaA | \lambda$$

$$B \rightarrow bB | \lambda$$

derive  $aaaaabb$



aaaaabb



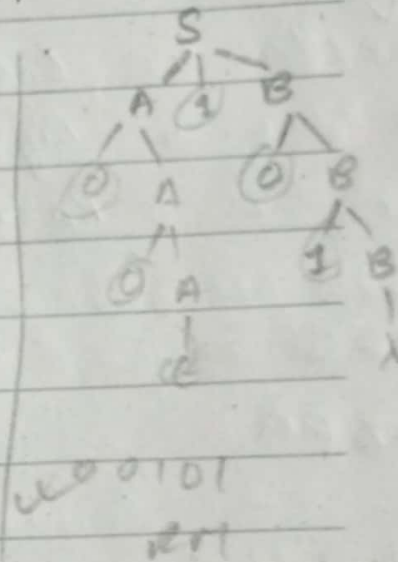
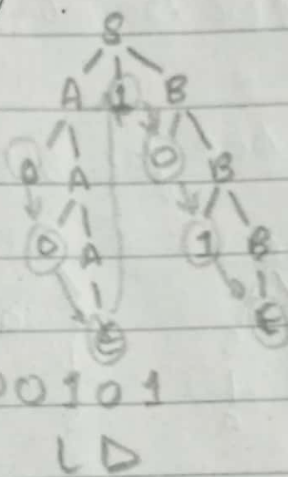
dono se same tree generate hoga hai.

Q: Derive "00101".

$$S \rightarrow A1B$$

$$A \rightarrow 0A | \epsilon$$

$$B \rightarrow 0B | 1B | \epsilon$$

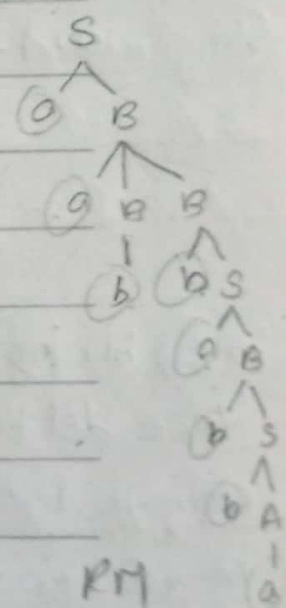


Q: Derive "aabbabba".

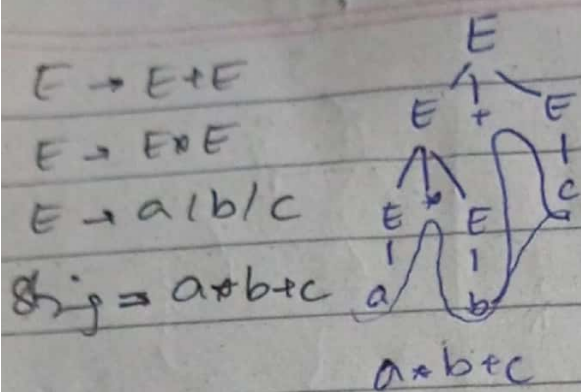
$$S \rightarrow aB / bA$$

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

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





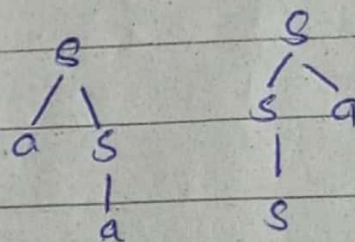


## "Ambiguous Grammar"

- A grammar is said to be ambiguous if there exist
- more than one Leftmost Derivation (or)
  - " " " " Right " " (or)
  - " " " " Parse tree for given input string.

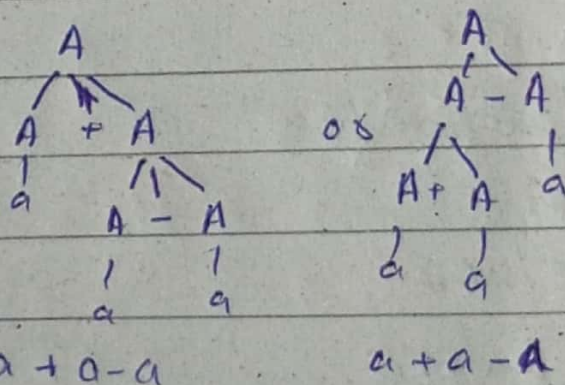
$$S \rightarrow aS / Sa / a$$

$$\text{String} = aa$$



$$A \rightarrow A + A / A - A / a$$

$$\text{String} = a + a - a$$



→ Koi bhi tree Leftmost Der hai ya RMDT hai wo hum tree dekh kr nhi kr sakte to is waja se agr ek string ke liye alag alag RM or LM Derivation tree bn rhe hai to wo Ambiguous hai.

→ If tree is Unambiguous then LMDT = RMDT

→ Grammar which is both left & right recursive is always ambiguous, but the ambiguous grammar need not to be both left & right recursive. (vice versa is not true)

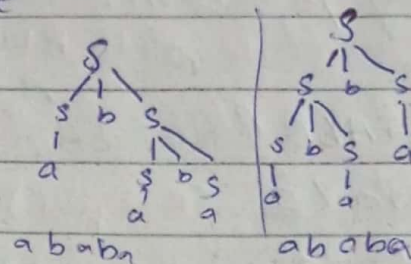
$S \rightarrow (Sa / bS) / \epsilon$  extreme right par hai to right recursive

extreme left par same symbol hai to left recursive

dono left & right recursive hai so this is ambi.

$S \rightarrow (S) b (S) / a$   
LR RR

→ ambiguous →



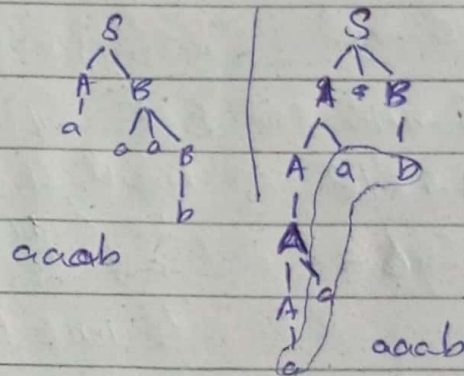
$S \rightarrow AB / aaB$

$A \rightarrow a / Aa$

$B \rightarrow b$

Not LR or RR

So we need to check.



$S \rightarrow aSbS / bSaS / \lambda$

ye left or right recursive nhi hai q k extreme left par terminal hai hai.

So ambiguous

$S \rightarrow SAB / \epsilon$

$A \rightarrow AaB / a$

$B \rightarrow AS / b$

} ye LR & RR nhi hai

lekn ags  $S \rightarrow SAB$  mein B ki

jaga AS likhdein to LR & RR hojayege

$S \rightarrow SAAS \Rightarrow$  So ambiguous.



# "SIMPLIFICATION OR MINIMIZATION OF CFL"

① Removal of Null or empty Production.

→ The production of form  $A \rightarrow \epsilon$  is called Null production or empty production.

②  $S \rightarrow AB \mid (A \mid B) \mid \epsilon$   
 $A \rightarrow a \mid \epsilon$  (Null prod.) → hum ab inki values  $S$  k andex  
 $B \rightarrow b \mid \epsilon$  (Null prod.) daal denge.  
 In dono ko yaha se remove kr denge.

③  $S \rightarrow AB \mid A \mid B \mid \epsilon$  → if we put  $\epsilon$  for both  $A$  &  $B$  then we will get  $\epsilon$  also. Iske matlab hum is language se  $\epsilon$  bhi generate kr sakte hai. Hum grammar ki language change nhi kr sakte to iske liye we write  $S' \rightarrow S \cup \{\epsilon\}$   
 $S \rightarrow AB \mid A \mid B$   
 $A \rightarrow a$   
 $B \rightarrow b$

→ Ags language mein hi  $\epsilon$  hai to 2nd waale method ki krna karna parega.

④  $S \rightarrow aSb \mid \epsilon$  →  $S' \rightarrow S \cup \{\epsilon\}$   
 $S \rightarrow aSb \mid ab$

⑤  $S \rightarrow AbaC$   
 $A \rightarrow BC$   
 $B \rightarrow b \mid \epsilon$   
 $C \rightarrow d \mid \epsilon$   
 $D \rightarrow d$   
 yaha Null variable  $B, C$  &  $A$  hai  
 jaha jaha ye teen variables hai waha  $\lambda$  put krke values likh denge  
 $S \rightarrow AbaC \mid baC \mid Aba \mid ba$   
 $A \rightarrow BC \mid B \mid C$   
 $B \rightarrow b$   
 $C \rightarrow d$   
 $D \rightarrow d$

### 3) Removal of Unit Production.

→ The production of the form  $A \rightarrow B$  where  $A, B \in V_n$ ,  $|A| = |B| = 1$ , is known as unit production.

$$A \rightarrow B \Rightarrow A \rightarrow a \checkmark$$

$$B \rightarrow a$$

$$S \rightarrow Aa \quad S \rightarrow Aa$$

$$(A) \rightarrow a (B) \Rightarrow A \rightarrow a/d$$

$$B \rightarrow d$$

~~$B \rightarrow d$  se khatam na khatam nahi hai~~

→ gaba change nhi krta q k single non-terminal to single non-terminal nhi change production or gaba non-terminal k saath terminal bhi involved hai

$$S \rightarrow aAb \quad S \rightarrow aAb$$

$$A \rightarrow b/a \quad d/c/b/a \quad A \rightarrow a/b/c/d$$

$$B \rightarrow c/b \quad d/c/b \Rightarrow A \rightarrow a/b/c/d$$

$$(C) \rightarrow (D)c = d/c$$

$$D \rightarrow d \checkmark$$

$$S \rightarrow aSb/E \quad \text{No simplification req.}$$

$$S \rightarrow Aa/B$$

$$B \rightarrow A/bb$$

$$A \rightarrow a/bc/B$$

Asab se phle unit wala productions likh denge jisme unit productions hai without ussitting unit productions

$$S \rightarrow Aa$$

$$A \rightarrow Aa$$

$$A \rightarrow Aa$$

$$A \rightarrow a/bc$$

$$S \rightarrow B$$

$$S \rightarrow Aa/bb/a/bc$$

$$B \rightarrow A$$

$$B \rightarrow Aa/bc$$

$$B \rightarrow bb/a/be$$

$$A \rightarrow B$$

$$A \rightarrow Aa/bb$$

$$A \rightarrow a/bc/bb$$



### ③ Removal of Useless symbols:

→ The variables which are not involved in the derivation of any string is known as useless symbol.

↳ Select the variable that cannot be reached from the start symbol of the grammar & remove them along with their all production.

↳ Select variables

$$S \rightarrow aAb$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$[C \rightarrow d] \rightarrow \text{useless symbol.}$$

↳ Select variable that are reachable from the start symbol but which does not derive any terminal, remove them along with their production.

$$S \rightarrow aA / bB \quad \times \text{ useless symbol of } B \text{ "kuch terminal generate nhi krta."}$$

$$A \rightarrow b$$

$$① S \rightarrow aAB / bA / c \rightarrow \text{useless}$$

$$A \rightarrow aB / b$$

$$B \rightarrow c / d$$

→ koi bhi rule kisi bhi order mein apply krsktte hai lekin ek rule apply krne k baad wapis simplification check krenge.

Q: Simplify by applying all rules.

$$S \rightarrow aA / aBB$$

$$S \rightarrow aA / aabb$$

$$A \rightarrow aA / \lambda$$

$$A \rightarrow aA / aa$$

$$B \rightarrow bB / bba$$

→ yaha jaha A hai waha  $\lambda$  put koderge

$$C \rightarrow B$$

→ C & B are like dead states koi terminal generate nhi krsktte appar mein looping hoshi hai sig. To jaha jaha B ya C hoga wo sab hat jayenge.

Simplify using only Rule (3)

$$Q: S \rightarrow AB / a$$

$$A \rightarrow BC / b$$

$$B \rightarrow aB / c$$

$$C \rightarrow aC / B$$

~~so~~ same left with

$$S \rightarrow a$$

$$A \rightarrow b$$

1st identify useful symbols. All the terminals are useful & the non-terminals generating them are also useful.  
useful = {a, b, S, A, B}

$B \rightarrow aB$  [is mein a to useful hai lekin B nhi production k right side k same elements useful set se belong krne chahie.]

$$B \rightarrow c \times$$

$$C \rightarrow aC \times$$

$$C \rightarrow B \times \rightarrow \text{Ab jaha jaha } B \text{ hai wo bhi useless.}$$

$$A \rightarrow BC \times$$

$$S \rightarrow AB \times$$

but yaha A is unreachable

$S \rightarrow a$  final answer.

$S \rightarrow AB / AC$  "B is useful & A also So S is also usefull

$$A \rightarrow aAb / bAa / a$$

$$B \rightarrow bba / aAb / AB \rightarrow B \rightarrow bba \rightarrow \text{ye useful symbol ka combination hai so B is also useful.}$$

$$C \rightarrow aBcA / aAb \times$$

$$D \rightarrow aAb / aC \times \text{ not useful.}$$

Useful = {a, b, A, B, S}

Ab useful wale symbols ki production mein jaha jaha not useful hai unko hata denge.

$$\begin{matrix} S \rightarrow AB \\ A \rightarrow aAb / bAa / a \\ B \rightarrow bba / aAb / AB \end{matrix}$$



## CHOMSKY NORMAL FORM

→ The grammar  $G$  is said to be in Chomsky Normal Form if every production is in the form.

$$A \rightarrow BC / a \rightarrow \text{ya ek terminal par.}$$

$B, C \in V$   $\hookrightarrow A$  is a non-terminal  
 $a \in \Sigma$   $\hookrightarrow$  par jasktte hai

→ The derivation tree from CNF is always binary tree.

→ Every grammar can be converted to CNF.

$$S \rightarrow aSb / ab \quad \text{only two non-terminals}$$

$$S \rightarrow ASB / AB$$

$$S \rightarrow ASB / AB$$

$$A \rightarrow A \quad \text{ishe change krna}$$

$$AS \rightarrow AS$$

$$B \rightarrow b \quad \text{Parag krz only two non-terminals are req.}$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$S \rightarrow aAb / bB$$

$$S \rightarrow CAB / BB$$

$$S \rightarrow CAB / BB$$

$$A \rightarrow a / b$$

$$\rightarrow A \rightarrow a / b$$

$$\rightarrow C_A \rightarrow CA$$

$$B \rightarrow b$$

$$B \rightarrow b$$

$$A \rightarrow a / b$$

$$C \rightarrow a$$

$$B \rightarrow b$$

$$C \rightarrow a$$

→ The no. of steps required to derive a string of length  $|w|$  is  $(2|w| - 1)$ .

$w = abcd \rightarrow |w| = 4$  so  $2(4) - 1 = 7 \rightarrow$  it will take 7 steps to generate "abcd".

→ Advantage: If a string  $w$  is not generated in  $2|w| - 1$  steps for a CNF grammar it means it does not belong to the language.

jitne bli terminals  
hai unke jaga new variable  
introduce karenge.

$S \rightarrow ABa$	$S \rightarrow \textcircled{AB}X$	$S \rightarrow A_BX$
$A \rightarrow aab$	$A \rightarrow \textcircled{XX}Y$	$A_B \rightarrow AB$
$B \rightarrow AC$	$B \rightarrow A\textcircled{Z}$	$A \rightarrow X_BY$
$X \rightarrow a$	$X \rightarrow a$	$X_X \rightarrow XX$
$Y \rightarrow b$	$Y \rightarrow b$	$B \rightarrow AZ$
$Z \rightarrow c$	$Z \rightarrow c$	$X \rightarrow a$
		$Y \rightarrow b$
		$Z \rightarrow c$

inke X, Y, Z mein change  
nhi karenge.

$S \rightarrow \textcircled{b}A/aB$	$S \rightarrow XA/YB$	$S \rightarrow XA/YB$
$A \rightarrow bAA/aS/a$	$A \rightarrow XAA/XS/X$	$A \rightarrow XAA/YS/Y$
$B \rightarrow aBB/bS/b$	$B \rightarrow YBB/XS/X$	$X_A \rightarrow XA$
$X \rightarrow a$	$X \rightarrow b$	$B \rightarrow Y_B B/XS/X$
$Y \rightarrow a$	$Y \rightarrow a$	$Y_B \rightarrow YB$
		$X \rightarrow B$
		$Y \rightarrow A$

yaha B blize nhi likha q k "B" k saath do os production  
involved hai to unke choices bli include hogati os humein  
cisf "b" change is way a se new variable introduce  
karna.

"Jab variables ko interchange krke hnge to  
right side se start karege."

$S \rightarrow \textcircled{ABCD} \Rightarrow S \rightarrow ABV_1 \Rightarrow S \rightarrow AV_2$   
 $V_1 \rightarrow CD$        $V_2 \rightarrow BV_1$   
 $V_1 \rightarrow CD$

random kisi bli  
do ko choose  
nhi kr sakte.