15/11/21

Noam chomsky gave a mathematical model of Grammar which is effective for writing computer language.

The four type of Grammar according to Noam chomsky are :

Grammar Type	Grammar Accepted	Language Accepted	Automaton
Type 0	unrestricted grammar	Recursively Enumerable Language.	~~Enumerable~~ Turing machine
Type 1	Context sensitive Grammar	Context sensitive Language.	Linear Bounded Automaton
Type 2	Context free Grammar	Context free Language	Push down Automata
Type 3	Regular Grammar	Regular Language	finite state Automaton

Scanned with CamScanner

Grammar :-

A Grammar 'G' can be formally described using 4 tuples as $G = (V, T, S, P)$ where,

V = Set of Variables or Non-Terminal symbols

T = Set of ~~Terminal~~ Terminal symbols

S = Start Symbol

P = Production rules for Terminals and Non-Terminals

A production rules has the form $\alpha \to \beta$ where α and β are strings on $V \cup T$ (union) and atleast one symbol of 'α' belongs to 'V'

Example :- $G = ((S, A, B), (a, b), S, (S \to AB, A \to a, B \to b))$

$V = \{S, A, B\}$ Non Terminals

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

$S = S$ start symbol

P = Production rules

$$\left.\begin{array}{l} S \to AB \\ A \to a \\ B \to b \end{array}\right\} \text{This is The Production rules}$$

Scanned with CamScanner

Eg:- $S \rightarrow AB$

$\rightarrow A$ (insted of A we can put a)

$\rightarrow aB$ (insted of B we can put b)

$\rightarrow \underline{ab}$

Now need to ~~to~~ discuss Regular Grammar.

Regular Grammar:-

Regular Grammar can be divided into two types.

Right Linear Grammar:-

A Grammar is said to be Right linear if all productions are of the form

$A \rightarrow xB$

$A \rightarrow x$

where $A, B \in V$ (Non Terminals) and $x \in T$ (Terminals)

Scanned with CamScanner

left linear Grammar :-

A grammar is said to be left linear if all productions are of the form

$$A \longrightarrow BX$$

$$A \rightarrow X$$

where $A, B \in V$ and $X \in T$.

Ex:-

$$S \longrightarrow abS | b$$

S which is a non terminal symbol and right of S i.e, a and b are the terminal symbols so. above is the right linear grammar.

$$S \longrightarrow Sbb | b$$

S which is a non terminal symbol and b is a non terminal symbol. So this is a left. linear grammar.

Scanned with CamScanner

4/12/21 ①

Unit-3 Ambiguity in CFG

A Grammar G is ambiguous if there exists some string $w \in L(G)$ for which there are two or more distinct derivation trees, or there are two or more distinct leftmost derivations.

Example:- Consider CFG $S \rightarrow S+S \mid S*S \mid a \mid b$ and string $w = a*a+b$, and derivations as follows.

Solution.-

first leftmost derivations for $w = a*a+b$

$S \Rightarrow S*S$ (using $S \rightarrow S*S$)

$\Rightarrow a*S$ (using $S \rightarrow a$)

$\Rightarrow a*S+S$ (using $S \rightarrow S+S$)

$\Rightarrow a*a+S$ (using $S \rightarrow a$)

$\Rightarrow a*a+b$ (using $S \rightarrow b$)

Scanned with CamScanner

Second leftmost derivation for $w = a * a + b$

$S \Rightarrow S + S$ (using $S \rightarrow S + S$)

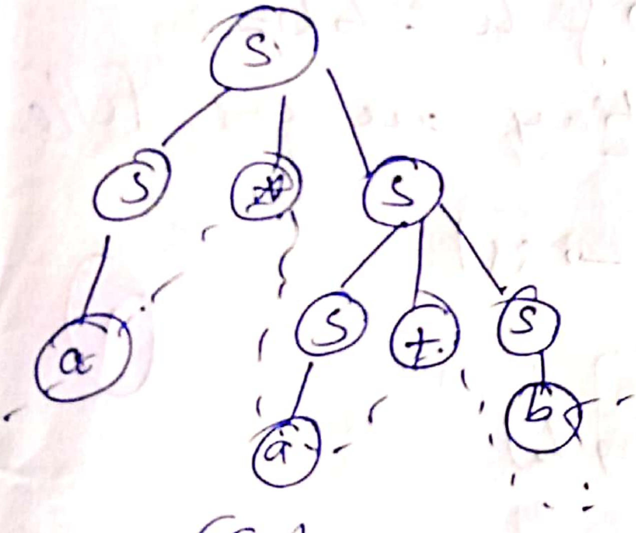
$\Rightarrow S * S + S$ (using $S \rightarrow S * S$)

$\Rightarrow a * S + S$ (using $S \rightarrow a$)

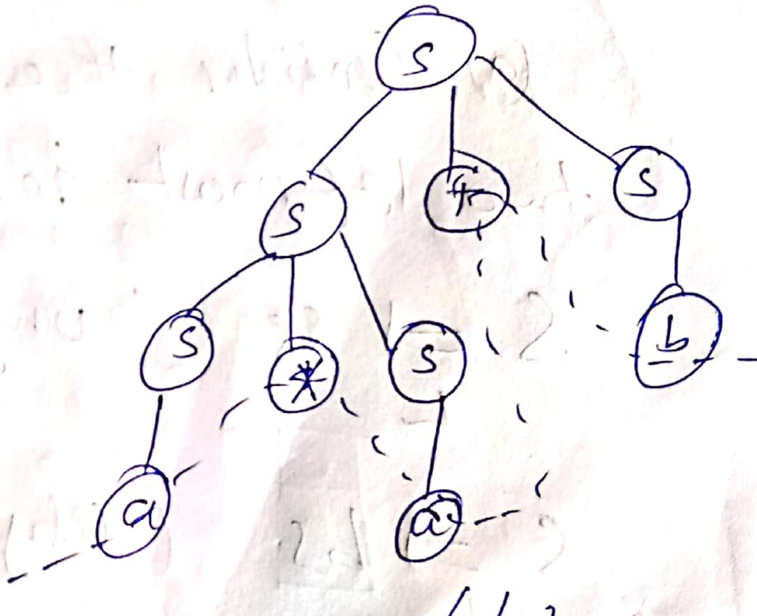
$\Rightarrow a * a + S$ (using $S \rightarrow a$)

$\Rightarrow a * a + b$ (using $S \rightarrow b$)

two distinct parse trees are shown figure (a) and figure (b)



(a)



(b)

fig(a) Parse tree for $a*a+b$

fig(b) : parse tree for $a*a+b$

Since there are two distinct leftmost derivations (two parse tree) for string w. hence w is ambiguous Then is ambiguity in Grammar G.

Ex:- Show that the Following grammar are ambiguous. ③

(a) $S \to SS \mid a \mid b$

(b) $S \to A \mid B \mid b$,
$A \to aAB \mid ab$,
$B \to abB \mid \epsilon$

Solution:-

(a) Consider the string $w = bbb$.
two leftmost derivations are as Follows

$S \underset{L}{\Rightarrow} SS$ (using $S \to SS$) ①

$S \underset{L}{\Rightarrow} bS$ (using $S \to b$)

$\underset{L}{\Rightarrow} bSS$ (using $S \to SS$)

$\underset{L}{\Rightarrow} bbS$ (using $S \to b$)

$\underset{L}{\Rightarrow} bbb$ (using $S \to b$)

Scanned with CamScanner

Or otherway

(2)

4

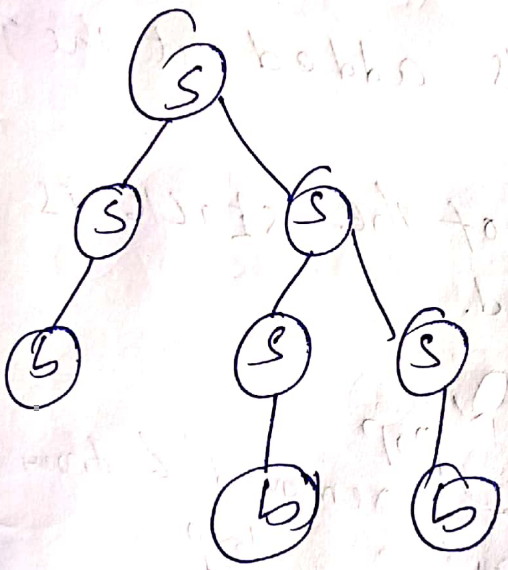
$S \underset{L}{\Rightarrow} SS$ (using $S \to SS$)

$\underset{L}{\Rightarrow} SSS$ (using $S \to SS$)

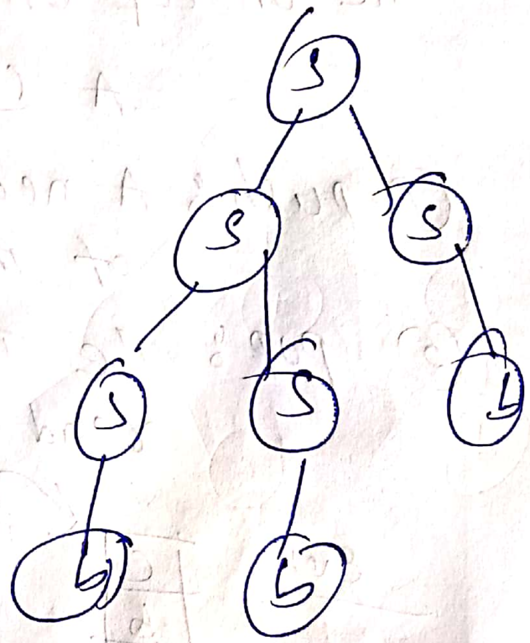
$\underset{L}{\Rightarrow} bSS$ (using $S \to b$)

$\underset{L}{\Rightarrow} bbS$ (using $S \to b$)

$\underset{L}{\Rightarrow} bbb$ (using $S \to b$)



(a) Parse tree for bbb



Parse tree for bbb

So, given two grammars are ambiguous

③ Context Free Language ①

Derivation Tree
or
Parse Tree

A Derivation Tree or Parse Tree is an ordered rooted tree that graphically represents the semantic information of strings derived from a context free Grammar.

Example:- for the Grammar $G = \{V, T, P, S\}$

where $S \to 0B$, $A \to 1AA \mid \epsilon$,

$B \to 0AA$

Root vertex : must be labelled by the start symbol

Vertex : Labelled by non-Terminal symbol

leaves :- labelled by Terminal symbols or ϵ (Epsilon)

Scanned with CamScanner

Context free Grammars :

A grammar $G = (V, T, P, S)$ is said to be a CFG if the productions of G are of the form:

$$A \rightarrow \alpha \text{ , where } \alpha \in (V \cup T)^*$$

The right hand side of a CFG is not restricted and it may be null or a combination of variables and terminals. The possible length of right hand sentential form ranges from 0 to ∞ ie, $0 \leq |\alpha| \leq \infty$

As we know that a CFG has no context neither left nor right. This is why, it is known as context-free.

Ex:- Consider grammar $G = (V, T, P, S)$ (2)
having productions $S \rightarrow aSa \mid bSb \mid \epsilon$.
check the productions and find the language generated.

Solution:-

let P_1: $S \rightarrow aSa$ (RHS is terminal variable terminal)

P_2: $S \rightarrow bSb$ (RHS is terminal variable terminal)

P_3: $S \rightarrow \epsilon$ (RHS is null string)

Since, all productions are of the form $A \rightarrow \alpha$, where $\alpha \in (V \cup T)^*$. hence given grammar G is a CFG.

Language Generated:- $S \rightarrow aSa$ or bSb

③

$\Rightarrow aSa$ or bSb

$\Rightarrow a^n S a^n$ or $b^n S b^n$ (using n-step derivation)

$\Rightarrow a^n b^m S b^m a^n$ or $b^n a^m S a^m b^n$ (using m-step derivation)

$\Rightarrow a^n b^m b^m a^n$ or $b^n a^m a^m b^n$ (using $S \to \epsilon$)

$\therefore$ so, $L(G) = \{ww^R : w \in (a+b)^*\}$

Scanned with CamScanner

(1) Derivations from a Grammar

The set of all strings that can be derived from a Grammar is said to be the Language generated from that Grammar.

Ex:- (1)

Consider the Grammar $G_1 = (\{S, A\}, (a, b), S, \{S \to aAb, aA \to aaAb, A \to \epsilon\})$

$S \to aAb$ (by $S \to aAb$)

$\to aaAbb$ [by $aA \to aaAb$]

$\to aaaAbbb$ [by $aA \to aaAb$]

$\to aaabbb$ [by $A \to \epsilon$]

(2)

Ex (2)

Grammar $G_2 = \{ \{S, A, B\}, \{a, b\}, S, S \to AB, A \to a, B \to b \}$

$S \to AB$
$\to ab$

Language generated by $G_2 \Rightarrow L(G_2) = \{ab\}$

Ex (3)

$G_3 = \{ (S, A, B), \{a, b\}, S, \{S \to AB, A \to aA | a, B \to bB | b\} \}$

(1) $S \to AB$ [$A \to a$ $B \to b$)
$\to ab$

(2) $S \to AB$
$\to aAbB$
$\to aabb$

(3) $S \to AB$
$\to aAb$
$\to aab$

(4) $S \to AB$
$\to abB$
$\to abb$

depend on choice production we take we can generate string.

Scanned with CamScanner

$L(G_3) = \{ab, a^2b^2, a^2b, ab^2, --\}$

$\boxed{L(G_3) = \{a^m b^n \mid m \geq 0 \text{ and } n \geq 0\}}$

This is the generalised way to generate the grammar G.

Scanned with CamScanner

① Language To Context free (Conversion)

Problem:- Show that the language $L = \{a^m b^n \mid m \neq n\}$ is Context free

Solution:-

If it is possible to construct a CFG to Generate this language then we say that the language is Context free.

Let us construct the CFG for the language defined.

Assume that $m = n$

ie, n number of a's should be followed by n number of b's.

So, the CFG for this can be

$$S \rightarrow aSb / \epsilon \text{ ——— } ①$$

But, $L = \{a^m b^n \mid m \neq n\}$ means, a's should be followed by b's and number of a's should not be equal to number of b's.

Scanned with CamScanner

② i.e, $m \neq n$.

Let us see the different cases when $m > n$ and when $m < n$.

Case-1 :-

$m > n$:-

This case occurs if the number of a's more compared to number of b's. ~~number of a's~~

The Extra a's can be generated using the production

$$A \rightarrow aA \mid a$$

and the Extra a's generated from this production should be appended towards left of the string generated from the production shown in production 1.

This can be achived by introducing one more production.

$$S_1 \rightarrow AS$$

Scanned with CamScanner

(3) So, even though from S we get n number of a's followed by n number of b's since it is preceded by a variable A from which we could generate extra a's, number of a's, ~~number of~~ followed by number of b's are different.

Case 2:-

$m < n$:-

This case occurs if the number of b's are more compared to number of a's. The extra b's can be generated using the production

$$B \rightarrow bB / b$$

and the extra b's generated from this production should be appended towards right of the string generated from the production shown in production (1)

Scanned with CamScanner

This can be achieved by introducing one more production.

$$S_1 \rightarrow SB$$

The Context free grammar $G = (V, T, P, S)$ where

$V = \{S_1, S, A, B\}$, $T = \{a, b\}$

$P = \{$

$S_1 \rightarrow AS \mid SB$

$S \rightarrow aSb \mid \epsilon$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

$\}$ S_1 is the start symbol.

generates the language

$$L = \{a^m b^n \mid m \neq n\}.$$

Since a CFG Exists for the language. So, The language is Context free.

Hence Proved.

Scanned with CamScanner

Problem:- Draw a CFG to generate a Language Consisting of Equal number of a's and b's.

(1)

Solution:-

Note that initial Production can be of the form

$S \rightarrow aB \mid bA$

If the first symbol is 'a', the second symbol should be a non-terminal from which we can obtain either 'b' or one more 'a' followed by two B's denoted by 'aBB' or a 'b' followed by S denoted by 'bS'.

Note that from all these symbols definitely we obtain Equal number of a's and b's.

The Productions Corresponding to these can be of the form

$B \rightarrow b \mid aBB \mid bS$

Scanned with CamScanner

on similar lines we can write A-productions as

(2)

$A \rightarrow a \mid bAA \mid aS$

from which we obtain a 'b' followed by Either

1. 'a' or
2. a 'b' followed by AA's denoted by

   bAA or

3. Symbol 'a' followed by S denoted by aS

The Context free grammar $G = (V, T, P, S)$ where $V = \{S, A, B\}$, $T = \{a, b\}$

$$P = \{ S \rightarrow aB \mid bA$$
$$A \rightarrow aS \mid bAA \mid a$$
$$B \rightarrow bS \mid aBB \mid b$$
$$\} \text{ \& } S \text{ is the start symbol}$$

generates the language consisting of Equal number of a's and b's.

Hence Proved

Problem :- obtain a CFG to generate integers.

Solution :-

The sign of a number can be '+' or '-' or ϵ.

The production for this can be written as

$$S \rightarrow + \mid - \mid \epsilon$$

A number can be formed from any of the digits 0, 1, 2, -- 9. the production to obtain these digits can be written as

$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid -- \mid 9$$

A number N can be recursively defined as follows.

1. A number N is a digit D (i.e, $N \rightarrow D$)
2. The number N followed by digit D is also a number (i.e, $N \rightarrow ND$)

Scanned with CamScanner

The production for this recursive definition can be written as

$N \rightarrow D$

$N \rightarrow ND$

An Integer number I can be a number N or the sign S of a number followed by number N.

The production for this can be written as $I \rightarrow N|SN$

So, the grammar G to obtain integer number can be written as $G=\{V,T,P,S\}$

where $V=\{D,S,N,I\}$

$T=\{+,-,0,1,2,3,--9\}$

$P=\{$

$I \rightarrow N|SN$

$N \rightarrow D|ND$

$S \rightarrow +|-|\epsilon$

$D \rightarrow 0|1|2|3|--|9$

$\}$ $S=I$ which is start symbol.

Hence Proved

Scanned with CamScanner

18/11/21

UNIT-3
Continue

LEFTMOST AND RIGHTMOST DERIVATIONS

Left most derivation:-

If $G=(V,T,P,S)$ is a CFG and $w \in L(G)$ Then a derivation $S \overset{L}{\Rightarrow} w$ is called Leftmost derivation if and only if all steps involved in derivation have leftmost variable replacement only.

Right most derivation:-

If $G=(V,T,P,S)$ is a CFG and $w \in L(G)$ then a derivation $S \underset{R}{\Rightarrow} w$ is called righmost derivation if and only if all steps involved in derivation have rightmost variable replacment only.

Examples:-

Scanned with CamScanner

Example 1:- Consider the grammar
$S \to S+S \mid S*S \mid a \mid b$. find leftmost and right most derivations for string w.
i.e., $w = a*a+b$.

Solution:-

Leftmost derivation for $w = a*a+b$

$S \underset{L}{\Rightarrow} S*S$ (using $S \to S*S$)

$\underset{L}{\Rightarrow} a*S$ (the first left hand symbol is a so using $S \to a$)

$\underset{L}{\Rightarrow} a*S+S$ (using $S \to S+S$, in order to get $a+b$)

$\underset{L}{\Rightarrow} a*a+S$ (second symbol from the left is a, so using $S \to a$)

$\underset{L}{\Rightarrow} a*a+b$ (the last symbol from the left is b, using $S \to b$)

Scanned with CamScanner

Right most derivation:-

for $w = a*a+b$

$S \underset{R}{\Rightarrow} S*S$ (using $S \to S*S$)

$\underset{R}{\Rightarrow} S*S+S$ (since, in the above sentential form second symbol from the right is * so we can not use $S \to a|b$ therefore, we use $S \to S+S$)

$\underset{R}{\Rightarrow} S*S+b$ (using $S \to b$)

$\underset{R}{\Rightarrow} S*a+b$ (using $S \to a$)

$\underset{R}{\Rightarrow} a*a+b$ (using $S \to a$)

Scanned with CamScanner

Problem 2:-

Consider a CFG $S \to bA \mid aB$, $A \to aS \mid aAA \mid a$, $B \to bS \mid aBB \mid b$. Find leftmost and rightmost derivations for $w = aaabbabbba$.

Solution:-

Leftmost derivation for $w = aaabbabbba$

$S \underset{L}{\Rightarrow} aB$ (using $S \to aB$ to generate first symbol of w)

$\underset{L}{\Rightarrow} aaBB$ (since, second symbol is a, so we use $B \to aBB$)

$\underset{L}{\Rightarrow} aaaBBB$ (since, third symbol is a, so we use $B \to aBB$)

$\underset{L}{\Rightarrow} aaabBB$ (since, fourth symbol is b, so we use $B \to b$)

$\underset{L}{\Rightarrow} aaabbB$ (since, fifth symbol is b, so we use $B \to b$)

Scanned with CamScanner

$\underset{L}{\Rightarrow}$ aaabbaBB (since first symbol is a, so we use $B \rightarrow aBB$)

$\underset{L}{\Rightarrow}$ aaabbabB (since seventh symbol is b, so we use $B \rightarrow b$)

$\underset{L}{\Rightarrow}$ aaabbabbS (since Eight symbol is b, so we use $B \rightarrow bS$)

$\underset{L}{\Rightarrow}$ aaabbabbbA (since, ninth symbol is b, so we use $S \rightarrow bA$)

$\underset{L}{\Rightarrow}$ aaabbabbba (since, the tenth symbol is a, so using $A \rightarrow a$)

$\therefore$ $\boxed{S \Rightarrow aaabbabbba}$

Rightmost derivation:- for w = aaabbabbba

$S \underset{R}{\Rightarrow} aB$ (Using $S \rightarrow aB$ to generate first symbol of w)

$\underset{R}{\Rightarrow} aaBB$ (we need a as the rightmost symbol and second symbol from the left side, so we use $B \rightarrow aBB$

$\Rightarrow aaBbS$ (we need a as right most symbol and this is obtained from A only, we use $B \to bS$)

$\Rightarrow aaBbbA$ (using $S \to bA$)

$\Rightarrow aaBbba$ (using $A \to a$)

$\Rightarrow aaaBBbba$ (we need b as the fourth symbol from the right)

$\Rightarrow aaaBbbba$ (using $B \to b$)

$\Rightarrow aaabSbbba$ (using $B \to bS$)

$\Rightarrow aaabbAbbba$ (using $S \to bA$)

$\underset{R}{\Rightarrow} aaabbabbba$ (using $A \to a$)

Rightmost derivation $S \underset{R}{\Rightarrow} aaabbabbba$

∴

problem 1:-

Let $G = (V, T, P, S)$ where $V = \{S, C\}$, $T = \{a, b\}$

$P = \{ S \to aCa$

$C \to aCa \mid b$

$\}$ S is the starting symbol.

what is the language generated by this grammar?

Solution:-

Consider the derivation

$S \Rightarrow aCa \Rightarrow aba$ (by applying 1st & 3rd production)

So, the string $aba \in L(G)$

Consider the derivation

$S \Rightarrow aCa$ by applying $S \to aCa$

$\Rightarrow aaCaa$ by applying $C \to aCa$

$\Rightarrow aaaCaaa$ by applying $C \to aCa$

Scanned with CamScanner

$\Rightarrow$ - -

$\Rightarrow$ - - -

$\Rightarrow a^n c a^n$ by applying $C \rightarrow aCa$ (n times)

$\Rightarrow a^n b a^n$ by applying $C \rightarrow b$

So, the Language L accepted by the grammar G is $L(G) = \{a^n b a^n \mid n \geq 1\}$

i.e, the Language L derived from the grammar G is "The string consisting of n number of a's followed by a 'b' followed by n number of a's.

Scanned with CamScanner

Problem 2:

what is the language generated by the grammar $S \to 0A \mid \epsilon$

$A \to 1S$

Solution: The null string ϵ can be obtained by applying the productions $S \to \epsilon$ and the derivations is shown bellow

$S \Rightarrow \epsilon$ (By applying $S \to \epsilon$)

Consider the derivation

$S \Rightarrow 0A$ (By applying $S \to 0A$)

$\Rightarrow 01S$ (By applying $A \to 1S$)

$\Rightarrow 010A$ (By applying $S \to 0A$)

$\Rightarrow 0101S$ (By applying $A \to 1S$)

$\Rightarrow 0101$ (By applying $S \to \epsilon$)

Scanned with CamScanner

(2) So, alternatively applying the productions $S \to 0A$ and $A \to 1S$ and finally applying the production $S \to \epsilon$, we get string consisting of only 01's.

So, both null string i.e. ϵ and string consisting 01's can be generated from this grammar.

So, The language generated by this grammar is

$$\boxed{L = \{ w \mid w \in \{01\}^* \} \text{ or } L = \{ (01)^n \mid n \geq 0 \}}$$

Scanned with CamScanner

Problem 3:-

Construct CFG for the Language L which has all the strings which are all palindromes over $T = \{a, b\}$

Solution:-

As we know the strings are palindrome if they posses same alphabets from forward as well as from backward

Ex:- Example string 'LIRIL' is palindrome because

LIRIL

$\longrightarrow$ read

$\longleftarrow$ read

$\Downarrow$

It is the same

Scanned with CamScanner

②

Since the language L is over $T=\{a,b\}$.
we want the production rules to be build a's and b's. As ϵ can be the palindrome, a can be palindrome even b can be palindrome. So we can write the production rules as

$G = (\{S\}, \{a,b\}, P, S)$

P can be $S \rightarrow aSa$

$S \rightarrow bSb$

$S \rightarrow a$

$S \rightarrow b$

$S \rightarrow \epsilon$

The string can be 'abaaba' will be derived as

Scanned with CamScanner

(3)

$S \rightarrow aSa$

$\rightarrow abSba \quad (S \rightarrow bSb)$

$\rightarrow abaSaba \quad (S \rightarrow aSa)$

$\rightarrow aba\epsilon aba \quad (S \rightarrow \epsilon)$

$\rightarrow abaaba.$

which is a palindrome

Scanned with CamScanner

19/11/21

DERIVATION TREES

Let $G = (V, T, P, S)$ is a CFG, Each production of G is represented with a tree satisfying the following Conditions:

① If $A \rightarrow \alpha_1 \alpha_2 \alpha_3 \cdots \alpha_n$ is a production in G, then A becomes the parent of nodes labeled $\alpha_1, \alpha_2, \alpha_3 \cdots \alpha_n$ and

② The Collection of children from left to right yields $\alpha_1 \alpha_2 \alpha_3 \cdots \alpha_n$

Example:-

Consider a CFG $S \rightarrow S+S \mid S*S \mid a \mid b$ and Construct the derivation trees for all Productions.

Solution:- for production $S \rightarrow S+S$.

Scanned with CamScanner

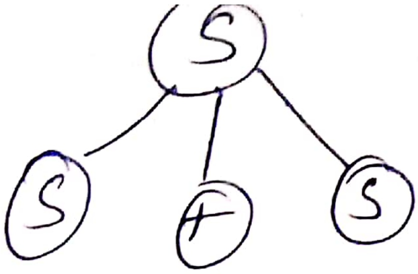
$\Rightarrow$ 

Fig (a)

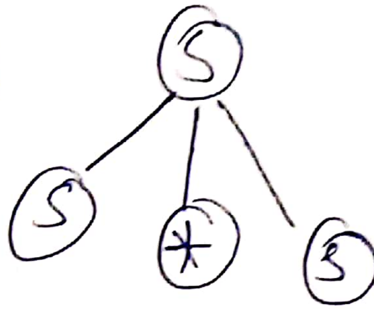
for production $S \rightarrow S * S \Rightarrow$ 

Fig (b)

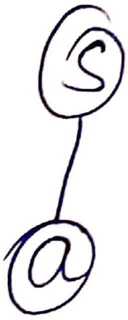
For production $S \rightarrow a \Rightarrow$ 

fig (c)

for production $S \rightarrow b \Rightarrow$ 

fig (d)

If $w \in L(G)$ then it is represented by a tree called derivation tree or parse tree satisfying the following conditions:

Scanned with CamScanner

1. The root has label S (the starting symbol)
2. The all internal vertices (or nodes) are labeled with variables.
3. The leaves (or terminals) nodes are labeled with ϵ or terminal symbol.
4. If $A \rightarrow \alpha_1 \alpha_2 \alpha_3 -- \alpha_n$ is a production in G then A becomes the parent of nodes labeled $\alpha_1 \alpha_2 \alpha_3 -- \alpha_n$ and
5. ~~The~~ The collection of leaves from left to right yields the string w.

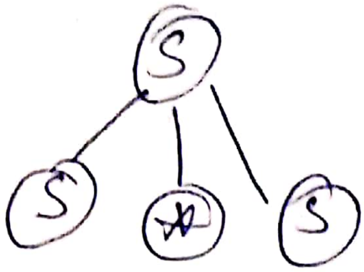
Example:- Consider the grammar $S \rightarrow S+S \mid S*S \mid a \mid b$. Construct derivation tree for string $w = a*b+a$.

Solution:- The derivation tree or parse tree is shown as

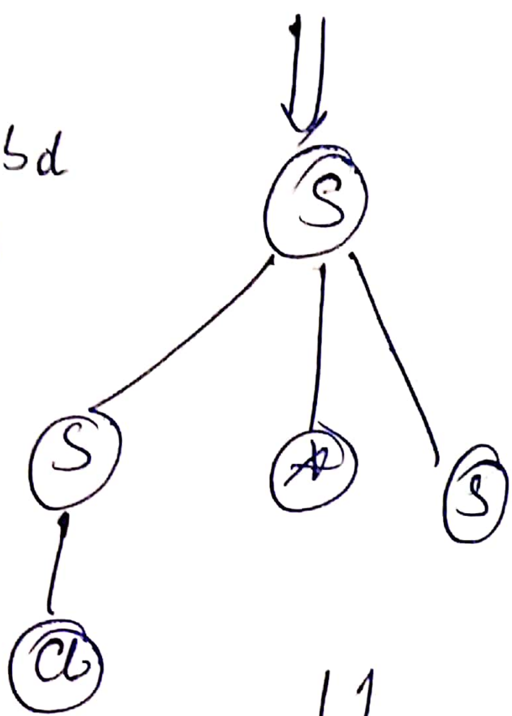
Scanned with CamScanner

left most derivation: for $w = a*b+a$.

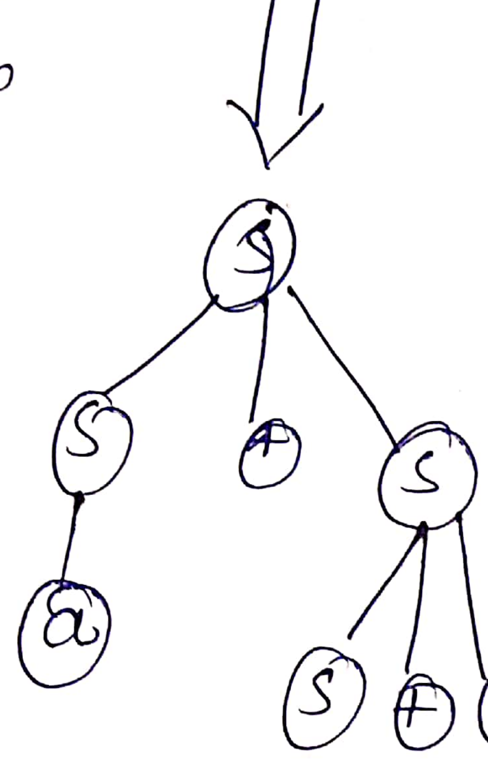
$S \Rightarrow S*S$ (using $S \rightarrow S*S$) $\Rightarrow$



$\Rightarrow a*S$ (the first left hand symbol is a, so using $S \rightarrow a$)



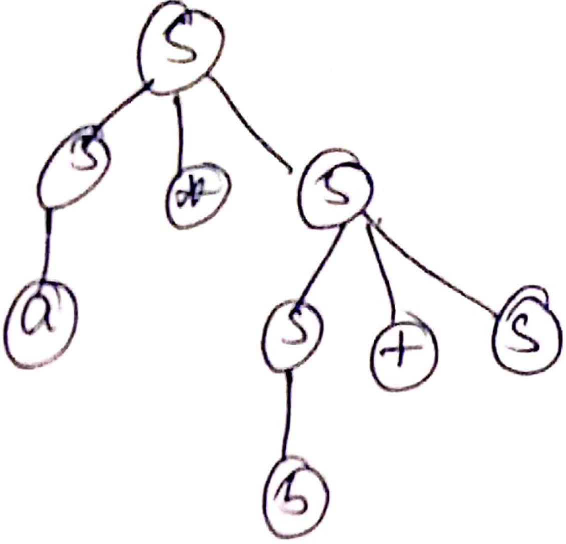
$\Rightarrow a*S+S$ (using $S \rightarrow S+S$, in order to get $b+a$)



Scanned with CamScanner

⇒ a * b + S (second symbol from left is b, so using S → b)

⇓



⇒ a * b + a (the last symbol from the left is a, so using S → a)

⇓

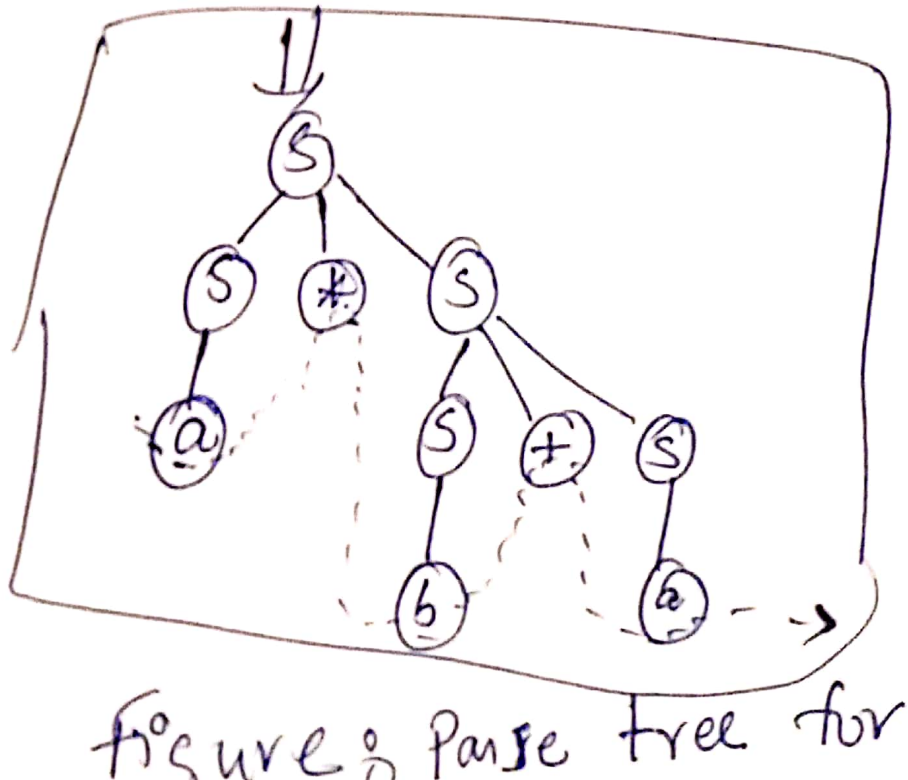


figure: Parse tree for a * b + a.

Scanned with CamScanner

Example 2:-

Consider a grammar G having productions $S \rightarrow aAS \mid a$, $A \rightarrow SbA \mid SS \mid ba$. Show that $S \Rightarrow aabbaa$ and construct a derivation tree whose yield is aabbaa.

Parse Tree

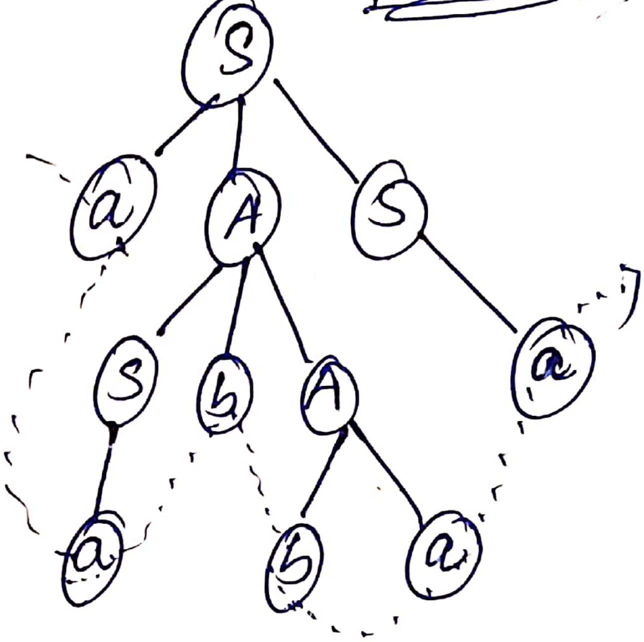


Fig: Parse tree yielding aabbaa.

Solution:-

$$
\begin{aligned}
S &\Rightarrow aAS \\
&\Rightarrow aSbAS \\
&\Rightarrow aabAS \\
&\Rightarrow aabbaS \\
&\Rightarrow aabbaa
\end{aligned}
$$

Hence $S \Rightarrow aabbaa$

Parse tree is shown in figure.

Example 3 :- Consider the grammar G whose productions are

$S \rightarrow 0B \mid 1A$, $A \rightarrow 0 \mid 0S \mid 1AA$, $B \rightarrow 1 \mid 1S \mid 0BB$.

Find (a) left most (2) Right most derivation for string 00110101 and construct derivation tree also.

Solution :-

Left most derivation :-

$$S \Rightarrow 0B \Rightarrow 00BB$$
$$\Rightarrow 001B \Rightarrow 0011S$$
$$\Rightarrow 00110B \Rightarrow 001101S$$
$$\Rightarrow 0011010B \Rightarrow 00110101$$

Right most derivation

$$S \Rightarrow 0B \Rightarrow 00BB$$

Scanned with CamScanner

$\Rightarrow 00B1 \Rightarrow 001S1$

$\Rightarrow 0011A1 \Rightarrow 00110S1$

$\Rightarrow 001101A1 \Rightarrow 00110101$

(c) Derivation tree

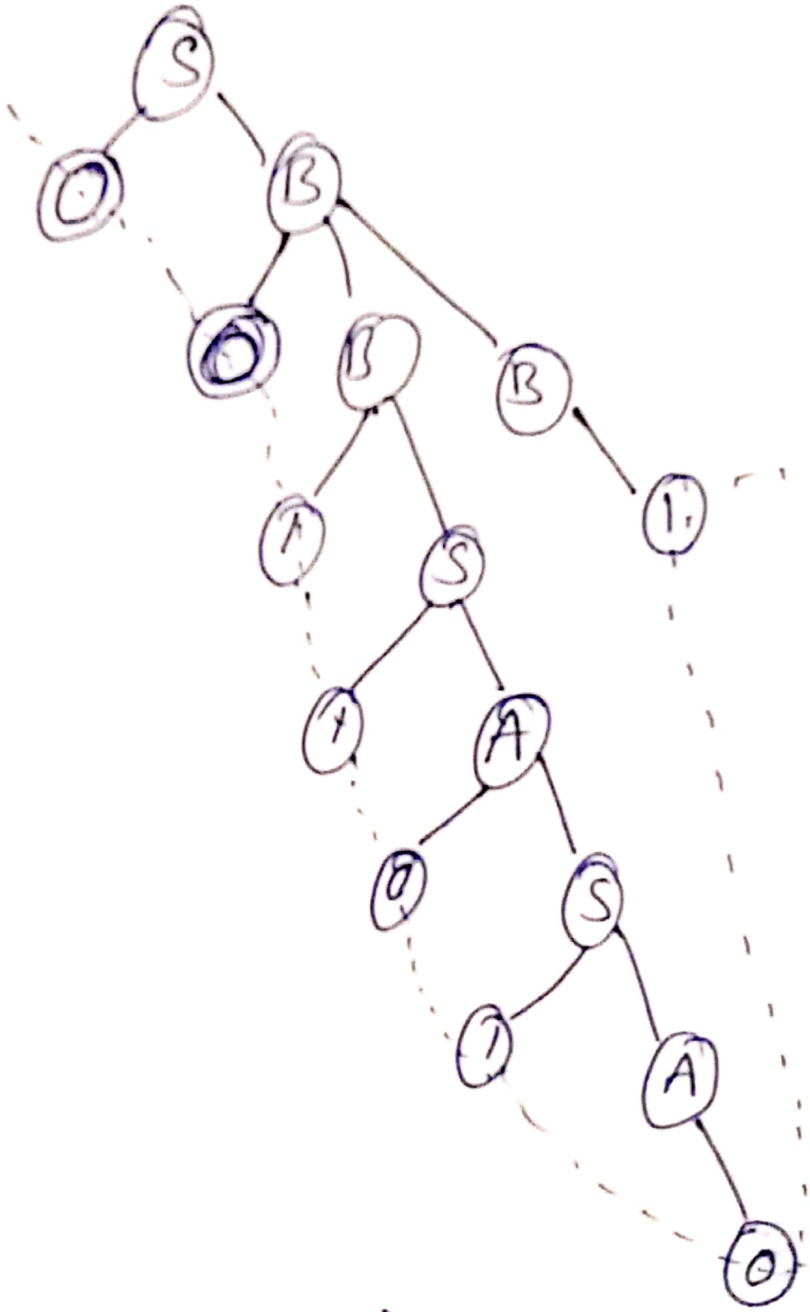


Fig. $\Rightarrow$ 0 0110101

is the derivation tree

Scanned with CamScanner