Subject Name: **Theory of Computation**

Subject Code: **IT-5001**

Semester: **5th**

## Unit-III : Context Free Grammar

### 3.1 Context Free Grammar

A CFG is way of describing languages by recursive rules called productions. A CFG consists of set of variables V, set of terminal symbols T , a start symbol S which is element of V and set pf rules or productions P. Therefore CFG is defined as

$$G = (V, T, P, S)$$

Each production consists of a head and body like head → body . The head consists of one variable form V and body is sring of one or more variable and terminal symblos ( V U T)*.

This class of grammar or language contains all regular languages. Moreover, this class contains languages such as $\{0^n 1^n : n \geq 0\}$, which, as we have already seen is not a regular language.

### 3.2 Derivation

To infer any string by replacing variable using productions of given CFG is known as derivation. Consider CFG G = ( V, T, P , S) and αAβ be a string of terminals and variables with A is variable from V and α, β are string of terminals and variables. Lets there is production A → γ in grammar G then there will be derivation αAβ => α γ β.

When derivation begins from start symbol S and end in string of terminals T then these inferred string of terminal is language of Grammars G.

**Example :**

G = ({S,A} , { 0,1}, P, S )

P ={ S → 0S1,S → A, A → ε }.

**Derivation**

S ⇒ 0S1
⇒ 00S11
⇒ 000S111
⇒ 0000S1111
⇒ 0000A1111
⇒ 00001111

### 3.3 Left Most Derivation (LMD)

In each step of derivation the left most variable is replaced by one of its production .

### 3.4 Right Most Derivation (RMD)

In each step of derivation the righ most variable is replaced by one of its production .

**Note:** LMD and RMD is applicable when in process of derivation a dentantial form consists more than one varibles.

### 3.5 The Language of Grammar

If G = (V,T,P,S) is CFG G then language of G denotes as L(G) is the set of terminal strings w that can be inferred by derivation from start symbol.

L(G) = { w in T* | S =*>w }

L(G) is called context free language.

Therefore , start from start symbol , derivation of terminal strings is called string of language and set of such string is called Language of CFG.

### 3.6 Derivation Tree or Parse tree

It is tree like structure to perform derivation .

A derivation or parse tree consists of followings

- the root is the start symbol for G
- the interior nodes are the nonterminals of G
- the leaf nodes are the terminal symbols of G.
- the children of a node T (from left to right) correspond to the symbols on the right hand side of some production for T in G.

Every terminal string generated by a grammar has a corresponding parse tree; every valid parse tree represents a string generated by the grammar called the yield of the parse tree .

**3.7 Yield of tree**

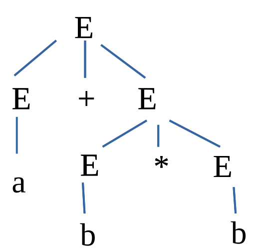
Concatenating all leaf node symbols from leaf right of a parse tree is called yield of parse tree.
When root of the parse tree is labeled by start symbol and all leaf nodes are terminal symbols than yield of tree is valid string of language of CFG.

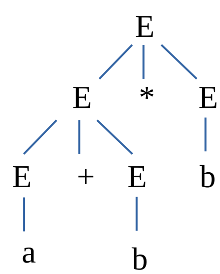
**3.8 Ambiguous Grammars**

A grammar for which there exist more than one different parse trees for the same terminal string is said ambiguous grammar.

**Example**

G = ({E},{a,b,+,*},{E -> E + E | E * E |a |b},{E})
We can prove this grammar is ambiguous by demonstrating two parse trees for the same
yield




Yield of these tow parse trees are a + b * b. Therefore give CFG is ambiguous.

While in general it may be difficult to prove a grammar is ambiguous, the demonstration of two distinct parse trees for the same terminal string is sufficient proof that a grammar is ambiguous.

**3.9 Simplification of CFG**

A CFG is said to be in simple form if it is
1. Free from useless symbols.
2. Free from epsilon productions such as A → ε
3. Free from unit productions A → B

Therefor to simplify given CFG , following steps have to be followed
1. Eliminate all epsilon productions.
2. Eliminate all unit productions
3. Eliminate all useless symbols.

### 3.10 Chomsky Normal Form(CNF)

A CFG G = (V,T,P,S) is said to be in CNF if all its productions are in following forms

$A \rightarrow BC$

$A \rightarrow a$

Where A,B,C are variables in V and a is terminal symbol in T.

Production $S \rightarrow \varepsilon$ is allowed in CNF provided start symbol S does not appear in body of any production.

### 3.11 Greibach Normal forms (GNF)

A CFG G = (V,T,P,S) is said to be in GNF if all its production are of the form $A \rightarrow a\beta$, where a is terminal symbol and $\beta$ is string of zeros and more variables.