Unit - III

Context Free Grammar and Languages

Grammar :-

Standard way of representing the language is called grammar in Automata.

Grammar contains a set of production rules which makes the strings of language. The set of all possible strings which can be derived from grammar is known as language of the grammar.

Grammar is just like the same as English grammar. If the sentence is correct grammatically then that sentence will be the part of grammar otherwise not.

- "I am going to school". It is a valid example of grammar.
- I going am to school. It is not valid example.

Grammar Constituents

Terminal symbols | Non-Terminal symbols

i) Terminal Symbols :-

Terminal symbols are the components of the sentence that are generated using grammar and are denoted using small case letters (lower case) like a, b, c, ...

ii) Non-terminal symbols:

Non-terminal symbols take part in the generation of the sentence but are not the component of sentence. These types of symbols are also called Auxiliary symbols or variables. They are represented using capital letters like A, B, C, ...

Types of Grammar :-

Grammar	Language	Automata
Type 0	Recursively Enumerable	Turing Machine
Type 1	Context-sensitive	Linear-bounded non-deterministic machine
Type 2	Context-free	Non-deterministic push down Automata
Type 3	Regular	Finite State Automata

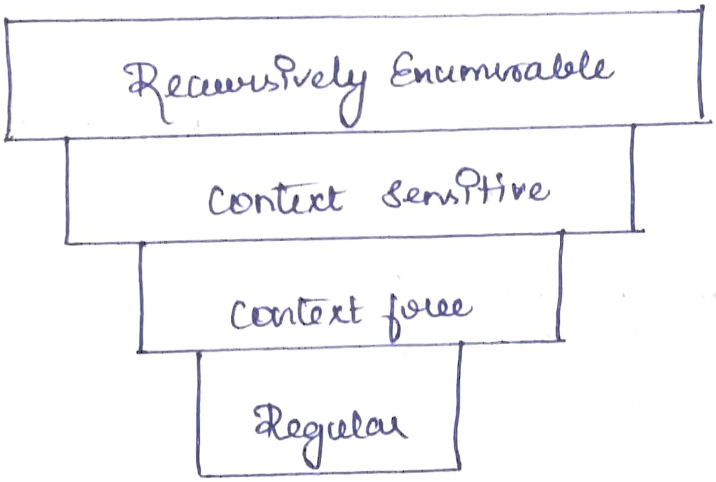

Figure : Types of Grammar in TOC

Context-free Grammar (CFG) :

Context Free Grammar is used to generate all the possible patterns of the strings in a given formal language. Context free grammar G can be defined by the four tuples as,

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

where $G \rightarrow$ G is a grammar, which consists of a set of production rule. It is used to generate the string of a language.

$T \rightarrow$ T is a final sets of Terminal Symbol.

Lowercase letters

V → V is the finite set of Non-Terminal symbols.
↓
Upper case letters

P → P is a set of production rules, which is used for replacing non-terminals symbols (on left side of productions) in a string with other terminals or non-terminal symbols (on right side of production).

S → S is the start symbol used to derive the string.

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

V → Finite set of variables/ Non-Terminals

T → Finite set of terminals

P → Finite set of Productions

S → Start symbol

Example 1:
Construct CFG for the language having any number of a's over the set $\Sigma = \{a\}$. Production rule is $P = \{S \to aS, S \to \varepsilon\}$ and derive the string "aaaaaa".

Solution: $\Sigma = \{a\}$

$L = \{\varepsilon, a, aa, aaa, aaaa, \ldots\}$

$\boxed{\therefore R.E = a^*}$

The given production rule is,

$S \to aS$ —— ①

$S \to \varepsilon$ —— ②

We have to derive the string "aaaaaa".

Hence,

$S \rightarrow aS$

$\Rightarrow aaS$ (By applying $S \rightarrow aS$)

$\Rightarrow aaaS$

$\Rightarrow aaaaS$

$\Rightarrow aaaaaS$

$\Rightarrow aaaaaaS$

$\Rightarrow \underline{aaaaaa}$ (By applying $S \rightarrow \varepsilon$)

$\downarrow$

Hence, we derived the string

Example 2:

Let $G = \{\{S\}, \{a, b\}, P, S\}$ with productions $P = (S \rightarrow aSa, S \rightarrow bSb, S \rightarrow \varepsilon)$. Find the language generated by this grammar.

Solution:

Find the minimum string obtained by this grammar.

Substitute $S \rightarrow \varepsilon$ in the productions $S \rightarrow aSa$ (&) $S \rightarrow bSb$.

$S \rightarrow aSa$

$\boxed{S \rightarrow aa}$ (By applying $S \rightarrow \varepsilon$)

$S \rightarrow bSb$

$\boxed{S \rightarrow bb}$ (By applying $S \rightarrow \varepsilon$)

Substitute the minimum strings obtained in the production

$S \rightarrow aSa$

$\Rightarrow aaaa$ (applying $S \rightarrow aa$)

$S \rightarrow aSa$

$\Rightarrow abba$ (applying $S \rightarrow bb$)

$$
\begin{aligned}
S &\Rightarrow aSa \\
&\Rightarrow abSba \qquad (\text{applying } S \to bSb) \\
&\Rightarrow abbbba \qquad (\text{applying } S \to bb) \\
&\Rightarrow \underbrace{abb}_{w}\ \underbrace{bba}_{w^R}.
\end{aligned}
$$

Hence, the language generated by this grammar is,

$$\boxed{L = \{ww^R \mid w \in \{a, b\}^*\}}$$

Example 3:

Construct a Context free grammar for the language,

$$L = \{wcw^R \mid \text{where } w \in (a, b)^*\}$$

Solution:

$$L = \{\underbrace{w}_{\text{String}}\underbrace{c}_{\text{Constant}}\underbrace{w^R}_{\text{Reverse String}} \mid \text{where } w \in (a, b)^*\}$$

$\Rightarrow L = \{aca, bcb, aacaa, bbcbb, abcba, abbcbba, \ldots\}$

The grammar could be,

$$
\begin{aligned}
S &\to aSa \quad \text{——(1)} \\
S &\to bSb \quad \text{——(2)} \\
S &\to c \quad \text{——(3)}
\end{aligned}
$$

$$\boxed{\therefore G = \{\{S\}, \{a, b, c\}, \{S \to aSa \mid bSb \mid c\}, S\}}$$

Derivations and Languages :-

Derivation is a sequence of production rules. It is used to get i/p strings. During parsing, we have to take two decisions :

--→ capital letters

i) we have to decide the non-terminal which is to be replaced

ii) We have to decide the production rule by which the non-terminal will be replaced.

We have two options to decide which non-terminal to be placed with production rule.

Derivation → Left most Derivation

Derivation → Right most Derivation

i) Left most derivation :-

In the left most derivation, the i/p is scanned and replaced with the production rule from left to right. So, we have to read i/p string from left to right.

Example :

Production rules : $E = E+E$

$E = E-E$

$E = a/b$

Input : $a-b+a$

The left most derivation is,

$$E = E+E$$
$$E = E-E+E$$
$$E = a-E+E$$
$$E = a-b+E$$
$$E = a-b+a$$

$\therefore E = a-b+a$ is obtained

(ii) Right Most derivation:

In Right most derivation, the i/p is scanned and replaced with the production rule from right to left. So, we have to replace the i/p string from right to left.

Example:

production rule: $E = E + E$ Input: $a - b + a$

$E = E - E$

$E = a / b$

The right most derivation is,

$E = E - E$

$= E - E + E$ (applying $E \to E + E$)

$= E - E + a$ (replace $E \to a$)

$= E - b + a$ (replace $E \to b$)

$E = a - b + a$ (replace $E \to a$)

Example 1: Derive the string "abb" for the left most derivation & right most derivation using the CFG given by,

$S \to AB / \varepsilon$

$A \to aB$

$B \to Sb$

Solution:

Left most derivation:-

$S \to A\,B$

$\Rightarrow aB\,B$ (replacing $A \to aB$)

$\Rightarrow aSb\,B$ (replacing $B \to Sb$)

$\Rightarrow ab\,B$ (replacing $S \to \varepsilon$)

$\Rightarrow abSb$ (replacing $B \rightarrow Sb$)

$\Rightarrow abb$ // (replacing $S \rightarrow \varepsilon$)

∴ The string 'abb' is obtained in left most derivation

Right Most derivation :-

$S \rightarrow AB$

$\Rightarrow ASb$ (replacing $B \rightarrow Sb$)

$\Rightarrow Ab$ (replacing $S \rightarrow \varepsilon$)

$\Rightarrow aBb$ (replacing $A \rightarrow aB$)

$\Rightarrow aSbb$ (replacing $B \rightarrow Sb$)

$\Rightarrow abb$ // (replacing $S \rightarrow \varepsilon$)

∴ The string 'abb' is obtained in right most derivation

Example 2: Derive the string "00101" for the left most and right most derivation using the CFG given,

~~S→A1B~~ $S \rightarrow A1B$

$A \rightarrow 0A \mid \varepsilon$

$B \rightarrow 0B \mid 1B \mid \varepsilon$

Solution:

Left most derivation :-

$S \rightarrow A1B$

$\Rightarrow 0A1B$ (replacing $A \rightarrow 0A$)

$\Rightarrow 00A1B$ (replacing $A \rightarrow 0A$)

$\Rightarrow 001B$ (replacing $A \rightarrow \varepsilon$)

$\Rightarrow 0010B$ (replacing $B \rightarrow 0B$)

$\Rightarrow 00101B$ (replacing $B \rightarrow 1B$)

$\Rightarrow 00101$ (replacing $B \rightarrow \varepsilon$)

Obtained //

Right most derivation :-

$S \rightarrow A1B$

$\Rightarrow A10B$ (replacing $B \rightarrow 0B$)

$\Rightarrow A101B$ (replacing $B \rightarrow 1B$)

$\Rightarrow A101$ (replacing $B \rightarrow \epsilon$)

$\Rightarrow 0A101$ (replacing $A \rightarrow 0A$)

$\Rightarrow 00A101$ (replacing $A \rightarrow 0A$)

$\Rightarrow \underline{00101}$ (replacing $A \rightarrow \epsilon$)

Obtained //

Example 3: Obtain the left most derivation for the string 'aaabbabba' using the gramma

$S \rightarrow aB \mid bA$

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

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