- **OBJECTIVE :** To familiarize context free grammars

UNIT II GRAMMARS **9**

Introduction - Types of Grammar - Context Free Grammars and Languages - Derivations - Parse Trees - Equivalence of Derivations and Parse Trees - Ambiguity - Normalization of CFG - Elimination of Useless symbols - Unit productions - productions - Chomsky normal form - Greibach Normal form.

GRAMMAR –TYPES OF GRAMMAR

Grammar Introduction :

- ✶ Grammar is denoted as G. which is defined as.

  G = (V, T, P, S)

where,

V = set of variables or Non-Terminals.

T = set of Terminals (V and T are disjoint $\therefore V \quad T = \phi$

P = finite set of productions each production is of the form $A \to \alpha$

where A is a variable.

α is a string of symbols from (VUT)*

S = is a special variable called the start symbol.

Example :

G = ({E}, {+, *, (,), id} p, E_

where p consists of $E \to E + E$

$E \to E * E$

$E \to (E)$

$E \to id$

Notations used in the Grammar.

1. The capital letters denote variables, S is the start symbol, unless otherwise stated.
2. The lower case letters a, b, c, d, e digits and boldface string are terminals.
3. The capital letters x, y and z denote symbols that may be either terminals or variables.
4. The lower – case letter u, v, w, x, y and z denote strings of terminals.
5. The lower – case Greek letters α, β, μ denote string of Non-terminals and Terminals.
6. If $A \to \alpha, |\alpha_2| ...\alpha_k$

   So the example can be written as,

$E \to E+E \mid E*E \mid (E) \mid id$

Types of Grammars :

1) Type 0 Grammar
2) Type 1 Grammar
3) Type 2 Grammar
4) Type 3 Grammar

Type 0 Grammar :

- It is understricted Grammar or phase structure grammar.
- A grammar without any restriction.
- The productions are of the form.

Type 1 Grammar :

- It is contest sensitive grammar or context dependent grammar.
- A production of the form.
  $\beta A \tau \to \beta \alpha \tau$ is called type 1 production if $\alpha \neq \in$
- It is accepted by linear bounded arutomata

Type 2 Grammar :

- It is context free grammar.
- A production of the form $A \to \alpha$
  where $A \in V, \text{and}\, \alpha \in (VUT)*$
- Left hand side has no left context or right context.
- It is accepted by push down automata.

Type 3 Grammar :

- It is regular grammar.
- A production of the form $A \to a \text{ or } A \to ab,$ where A, B, $\in$ v and a $\in \Sigma$
- It is accepted by finite automata.

Derivation and langauages :

- If $A \to B$ is a production of P and $\alpha \text{ and } \tau$ are any strings in (VUT)* then
  $\alpha A \tau \underset{G}{\Leftarrow} \alpha \beta \tau$
- The derivation may be
  1. left most derivation
  2. right most derivation

Left most derivation :

- If at each step of derivation, a production is applied to the left most non-terminal then the derivation is said to be left-most derivation.
- Example
  $E \to E+E \mid E*E \mid (E) \mid id$
  Deriving a string id + id * using left most derivation.
  $E \underset{lm}{\Rightarrow} E+E$

$\underset{lm}{\Rightarrow} id + E$

$\underset{lm}{\Rightarrow} id + E * E$

$\underset{lm}{\Rightarrow} id + id * E$

$\underset{lm}{\Rightarrow} id + id * id$

Right most derivation :

- If at each step in a derivation is applied to the light most variable is said to be right most derivation.
- Example

  $E \rightarrow E + E \mid E * E \mid (E) \mid id$
- Deriving a string id + id * using left most derivation.

  $E \overset{lm}{\Rightarrow} E + E$
-
- $\overset{lm}{\Rightarrow} id + E$
- $\overset{lm}{\Rightarrow} id + E * E$
- $\overset{lm}{\Rightarrow} id + id * E$
- $\overset{lm}{\Rightarrow} id + id * id$

Context free languages (CFL)

- The language generated by G, L (G) is

  $L(G) = \{w \mid w \text{ is in } T^* \text{ and } S \overset{*}{\underset{G}{\Rightarrow}} w\}$
- That is,

  a string is in L (G) if

  1) The string consists of only terminals
  2) The string can be delivered
- A string of terminals and variables α is called a sentential form if
- We define grammar G1 and G2 to be equivalent if $L(G_1) = L(G_2)$

Derivation Trees (Parse tree)

- Derivation can be displayed as a derivation tree.
- The vertices of a derivation tree and labeled with terminal or variable symbols of the grammer or $\in$
- If an interior vertex is labeled A, and the sons of A are labeled x_1, x_2, ….x_k from the left then A $\rightarrow x_1\ x_2 \ldots x_k$ must be a production.
- The derivation tree.
- If we lead the leaves from left to light, we get the string (id + id) * id.

- More formally Let G (V, T, P, S) be a CFG
- A tree is a derivation tree for G if

  (1) Every vertex has a label, which is a symbol of VUTU {∈}

  (2) The label of the root is S

  (3) If a vertex is interior and has label then A must be in V

  (4) If vertex has label A and sons vertex A are labeled from left as x1, x2 … xk then $A \to x_1, x_2 \ldots x_k$ must be a production in p.

  (5) If a vertex has a label ∈, then it leaf and is the only son of its father.

LMD 1

$E \underset{lm}{\Rightarrow} E+E$

$\underset{lm}{\Rightarrow} id+E$

$\underset{lm}{\Rightarrow} id+E*E$

$\underset{lm}{\Rightarrow} id+id*E$

$\underset{lm}{\Rightarrow} id+id*id$

LMD 2

$E \underset{lm}{\Rightarrow} E*E$

$\underset{lm}{\Rightarrow} E+E*E$

$\underset{lm}{\Rightarrow} id+E*E$

$\underset{lm}{\Rightarrow} id+id*E$

$\underset{lm}{\Rightarrow} id+id*id$

Ambiguity :

- A context free grammar G such that some word has two parse trees is said to be ambiguous.
- An equivalent definition of ambiguity is that some word has more than one left most derivation or more than one right most derivation.
- Example :

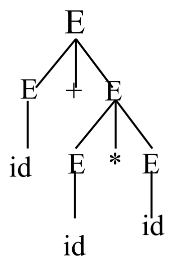
  Show that the Grammar G

  $E \to E+E \mid E*E \mid (E) \mid id$ is ambiguous.

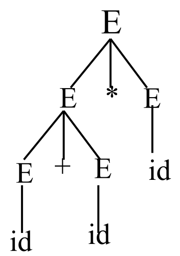
Solution :

- Deriving a string id + id * id
- For the word id+id*id, there exists two right most derivation.

Parse tree 1



parse tree 2



- For the word id + id * id has two parse trees
- So the grammar is ambiguous.

THE RELATIONSHIP BETWEEN DERIVATION AND DERIVATION TREES

Theorem :

Let G = (V, T, P, S) be a context free Grammar

Then $S \overset{*}{\Rightarrow} \alpha$ iff there is a derivation tree in Grammar G with yield α

Proof :

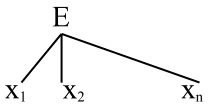
- We shall prove that for any A in V.

  $A \overset{*}{\Rightarrow} \alpha$ if and only if there is an A-tree with α as the yield.

- This can be proved by methamatical induction on the number of interior vertices in the tree.

Basis :

- If there is only one interior vertex, then the tree is



i.e) $x_1, x_2 \ldots. x_n$ must be α and $A \rightarrow \alpha$ be a production of p, by the definition of a derivation tree.

Induction :

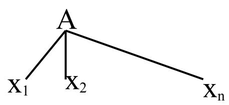
- Assume that the result ais tree for k-1 interior vertices.
- Suppose that α is the yield of an A-tree with k interior vertices for k>1
- The son's of the node A could not all be leaves.
- Let the labels of the sons be $x_1, x_2 \ldots x_n$ in order from the left.
- Then $A \rightarrow x_1, x_2 \ldots. x_n$ is a production in P.
- If the ith son is not a leaf it is the root of the subtree, and x_i must be a variable.
- The portion of α delivered from x_i must lie to the left of the symbols delivered from x_j.
- Thus we can write α as $\alpha_1 \alpha_2 \ldots \alpha_n$, where for each I between 1 and n.

  1) $\alpha_1 = x_i$ if x_i is a terminal
  2) $x_1 \Rightarrow \alpha_i$ if x_i is a variable

Case 1 :

(x$_i$ is a terminal)

- If $\alpha_i = x_i$ then x_1 is a terminal such that the derivation tree is.



Case 2 :

(x$_i$ is a variable)

- If x is a variable, then the derivation of α_i from x_i must take fewer than k steps.
- Thus by the inductive hypothesis, for each x_i that is a variable, there is x_i tree with α_i
- Let this tree b T_i

<u>SIMPLIFICATION OF CFG</u>

Simplification of CFG:

- There are several ways to restrict the format of productions without reducing the generative power of context – free grammar.

- If L is a non empty context free language, then it can be generated by a context free grammar G with the following properties.
- Each variable and each terminal of G appears in the derivation of some word in L.
- There are no productions of the form $A \to B$, where A and B are variable.
- If $\in$- is not in L, there need be no productions of the form $A \to E$
- If E is not in L, every production be of one of the forms $A \to BC$ and $A \to O$
- Also we can make every productions of the form $A \to a\alpha$, where α is a string.
- There are two special forms of CFG
  1. Chomsky Normal form.
  2. Greibach Normal form.

Elemination of Useless symbols :

- The useless symbols can be eliminated from a grammar.
- Let G = (V, T, P, S) be a Grammar
- A symbol x is useful if there is a derivation $S \overset{*}{\Rightarrow} \alpha x \beta \overset{*}{\Rightarrow} w$, for some α,β and w, where w is in T^*
- Otherwise x is useless
- There are two aspects of usefulness
  1. Some terminal string must be derivation from x
  2. x must occur in some string derivation from s

Lemma 1 :

Given a CFG, G = (V, T, P, S) with L (G) = ϕ we can find an equivalent CFG G^1.

$G^1 = (V^1, T, P^1, S)$ such that for each A in V^1, there is some w in T^* for which $A \overset{*}{\Rightarrow} w$

Lemma 2:

Given a CFG G = (V, T, P, S), we can find an equivalent Grammar G1 = (V^1, T^1, P^1, S) such that for each x in V^1 U T^1there exist α and β in $(V^1 \text{ U } T^1)^*$ for which $S \overset{*}{\Rightarrow} \alpha x \beta$

Example :

Consider the Grammar, and find the useless symbol.

$S \to AB \mid a$
$A \to a$

Solution :

We find that no terminal string is derivable from B. we therefore eliminate B and the production $S \to ab$

$S \to a$

$A \to a$

G = {s}, {a}, $\{s \to a\}$, s) is an equivalent Grammar with no useless symbol.

Elemination of E-Productions :

- The productions of the form $A \rightarrow E$ is called E-production
- If E-is in L(G). we cannot eliminate all E-productions from G.
- If E-is not in L(G), we can eliminate the E-productions from G.
- The method is to determine for each variable A, whether
- $A \overset{*}{\Rightarrow} E$, if so it is called as A-nullable
- We may replace each production $B \rightarrow x_1, x_2, ... x_i ... x_n$ by all productions striking out some subsets of those x_1's that are nullable. but we do not include $B \rightarrow E$, even if all x_i,s are nullable.

Example :

Consider the grammar

$S \rightarrow as \mid bA \mid \in$

$A \rightarrow \in$

Eliminate the E-productions.

Solution :

we find that E-is in L(G), so we eliminate $S \rightarrow \in$ and A-is nullable

so the resultant Grammar

$S \rightarrow as \mid b \mid \in$

Elimination of Unit productions :

- The productions of the form $A \rightarrow B$ is called is unit productions
- There need be no productions of the form $A \rightarrow B$
- Unit Productions can be eliminated form the grammar

Examble :

Consider the grammar

$S \rightarrow A$

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow a$

Eliminate the unit productions

Solution :

By eliminating the unit productions, the resultant grammar is

$S \rightarrow A$

CHOMSKY NORMAL FORM

Chomsky normal form (CNF)

- In Chomsky normal form, the productions are of the form

$A \rightarrow B$

$A \rightarrow a$

where A, B, C are variable, a is a term.

Theorem :

- ✶ Any context free language without E is generated by a Grammar in which the productions are of the form $A \to BC$ or $A \to a$. Here A, B and C are variable and a is a terminal.

Proof :

- ✶ Let G be a CFG, generating a language not containing E.
- ✶ We can find an equivalent grammar G_1 = (V, T, P, S) such that p contains no unit productions or E-productions.
- ✶ Thus if a productions has a single symbol on the right, that symbol is a terminal then the production is an acceptable form.
- ✶ If production is of the form $A \to x_1, x_2 \ldots. x_m$ where $m \geq 2$

  If xi is a terminal a, introduce a new variable c_a and a production $c_a \to a$ then replace x_i by c_a
- ✶ Let the new set of variables be V1 and the few set of productions be P^1 consider G_2 (V^1, T, P^1, S)
- ✶ We modify G_2 by adding some additional symbols to V1 and replacing some productions of P^1
- ✶ For each production

  $A \to B_1 B_2 \ldots.. B_m$ of P^1 where $m \geq 3$ we create a new variables $D_1, D_2, \ldots. D_{m-2}$ and replace $A \to B_1 B_2 \ldots\ldots B_m$ by the set of productions.

  $A \to B_1 D_1, D_1 \to B_2 D_2, .. D_2 \to B_3 D_3, \ldots.$

  $D_{m-3} \to B_{m-2} \to D_{m-2}, D_{m-2} \to B_{m-1} B_m$
- ✶ Let V" be the now set of non-terminals and P" be the new set of productions G3 = (V", T, P", S) is in CNF

Examble :

$S \to bA \mid ab$

$A \to bAA \mid as \mid a$

$B \to aBB \mid bs \mid b$

Find an equivalent grammar in CNF

Solution :

(1) $S \to bA$ is replaced by

$S \to C_b A$

$C_b \to b$

(2) $S \to aB$ is replaced by

$S \to C_a B$

$C_a \to a$

(3) $A \to bAA$ is replaced by

$A \to C_b AA$

$C_b \to b$

$A \to c_b AA$ is replaced by

$A \to C_b D_1$

$D_1 \to AA$

(4) $A \to aS$ is replaced by

$A \to C_a S$

$C_a \to a$

(5) $A \to a$ is in proper form

(6) $B \to aBB$ is replaced by

$B \to C_a BB$

$C_a \to a$

(7) $B \to C_a BB$ is replaced by

$B \to C_a D_2$

$D_2 \to BB$

(8) $B \to bS$ is replaced by

$B \to C_b S$

$C_b \to b$

(9) $B \to b$ is in proper form

Resultant grammar :

1. $S \to C_b A | C_a B$

   $A \to C_b D_1 | C_a S | a$

   $B \to C_b S | C_a D_2 | b$

   $D_1 \to AA$

   $D_2 \to BB$

   $C_a \to a$

   $C_b \to b$

GREIBACH NORMAL FORM

Greibach normal form :

- ✶ GNF uses productions whose light hand sides each start with a terminal symbol followed by some variable

Lemma 1:

- ✶ Let G = (V, T, P, S) be a FCG

  Let $A \to \alpha_1 B \alpha_2$ be a production in p and $B \to B_1 | B_2 | \ldots | B_r$ be the set of all B-productions

* Let G_1 = (V, T, P, S) be obtained from G by deleting the production $A \to \alpha_1 B \alpha_2$ from p and adding the productions
  $A \to \alpha_1 B_1 \alpha_2 | \alpha_1 B_2 \alpha_2 | \alpha_1 B_3 \alpha_2 | \ldots | \alpha_1 B_r \alpha_2$
  Then L (G) = L (G_1)

Lemma 2:

* Let G = (V, T, P, S) be a CFG
  Let $A \to A\alpha_1 | A\alpha_2 | \ldots. | A\alpha_r$ be the set of A-Productions for which A is the left most symbol of RHS
* Let $A \to B_1 + B_2 | \ldots | B_s$ be the remaining A-Productions.
* Let G1 = (VU {B}, T, P, S) be the GFG formed by adding the variable B to V and replacing all the A-productions by the productions

  1) $\left.\begin{array}{l} A \to B_i \\ A \to B_i B \end{array}\right\} i \le i \le s$

  2) $\left.\begin{array}{l} B \to \alpha_i \\ B \to \alpha_i B \end{array}\right\} i \le i \le r$

Theorem :

Every context free language without ε- can be generated by a grammar for which every production is the form $A \to a\alpha$, where A is a variable a is and α is a string of variables.

Proof :

* Let G = (V, T, P, S) be a CFG in CNF generating the CFL L.
* Assume V = {A_1, A_2, … A_m}
* The first step in the construction is to modify the productions so th
* at if
  $A_i \to A_j \mu$ is a production, then j>i
* Starting with A_1 and proceeding to A_m we do this as follows
* We now modify the A_k productions
* If $A_k \to A_j \mu$ is a production with j<k, we generate new set of production by substituting for A_j, the RHS of each Aj production according to Lemma1.
* By repeating the process k-1 times atmost we obtain productions of the form $A_k \to A_\gamma \mu 1 \ge k$ according to Lemma 2 introducting a new variable B_k
* By repeating the above process for each variable, we have only the productions of the form.

  1) $A_i \to A_j \mu \quad j > i$

  2) $A_i \to a\mu \quad a \text{ in } T$

  3) $B_i \to \mu \quad \mu \text{ in } (VU\{B_1, B_2, \ldots B_{i-1})^*$

- RHS of any production for A_m must be a terminal since A_m is the highest numbered variable
- The left most symbol on the RHS of any production for A_{m-1} must be A_{m-1} or a terminal symbol.
- At the last step examine the productions for the new variables $B_1, B_2, \ldots B_m$
- No B_i-productions can start with another B_j therefore all B_i-productions have RHS beginning with terminals or A_i's

Example :

Convert into GNF from the Grammar

$G = (\{A_1, A_2, A_3\}, \{a, b\}, P, A_1)$

where P consist of

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 | b$

$A_3 \rightarrow A_1A_2 | a$

Solution :

Step 1 :

Since RHS of the productions for A_1 and A_2 start with terminal or higher numbered variables we begin with the productions.

$A_3 \rightarrow A_1A_2$

$A_3 \rightarrow A_2A_3 | A_2$

$A_3 \rightarrow A_3A_1A_3A_2 | bA_3A_2 \quad (\therefore A_2 \rightarrow A_3A_1 | b)$

- The new resultant set of productions

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 | b$

$A_3 \rightarrow A_3A_1A_3A_2 | bA_3A_2 | a$

- We now apply lemma 2 to the productions

$A_3 \rightarrow A_3A_1A_3A_2 | bA_3A_2 | a$

symbol B3 is introduced and the production $A_3 \rightarrow A_3A_1A_3A_2$ is replaced by

$A_3 \rightarrow bA_3A_2 | a$

$A_3 \rightarrow bA_3A_2 | B_3 | aB_3$

$B_3 \rightarrow A_1A_3A_2$

$B_2 \rightarrow A_1A_3A_2B_3$

- The resultant set of production

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 | b$

$A_3 \rightarrow bA_3A_2B_3$

$B_3 \rightarrow A_1A_3A_2B_3$

- Now all the productions with A_3 on the RHS that start with terminals.

- These are used to replace A3 in the productions $A_2 \rightarrow A_3A_1$ and then the productions with A_2 on the left are used to replace A2 in the production $A_1 \rightarrow A_2A_3$
- The new set of productions

  $A_3 \rightarrow bA_3A_2B_3|aB_3|bA_3A_2|a$

  $A_2 \rightarrow bA_3A_2B_3A_1|aB_3A_1|bA_3A_2A_1|aA_1|b$

  $A_1 \rightarrow bA_3A_2B_3A_1A_3|aB_3A_1A_3|bA_3A_2A_1|aA_1A_3|aA_1A_3|bA_3$

  $B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2|aB_3A_1A_3A_3A_2|bA_3A_2A_1A_3A_3A_2|$
  $aA_1A_3A_3A_2|aA_1A_3A_3A_2|bA_3A_3A_2$

  $B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2B_3|aB_3A_1A_3A_3A_2B_3|bA_3A_2A_1A_3A_3A_2B_3|$
  $aA_1A_3A_3A_2B_3|aA_1A_3A_3A_2B_3|bA_3A_3A_2B_3$