8- Properties of Context Free Languages

8.1 Useless Symbols:

Variables or Terminals that do not appear in any derivation of a terminal string from the start symbol.

Omitting Useless symbols from the grammar will not change the language of the Grammar.

8.1.1 Useful Symbols:

The symbols must be both generating and reachable.

Generating Symbols: Symbol X is generating if X $\overset{*}{\Rightarrow}$ w for some terminal string w.

Reachable Symbols: Symbol X is reachable if S $\overset{*}{\Rightarrow}$ $\alpha X \beta$ for some sentential forms α and β.

Useless symbol is not *generating* or not *reachable*.

8.1.2 Eliminating Useless Symbols:

- Eliminate symbols that are not generating.
- Eliminate symbols that are not reachable.

Example: Consider the grammar

$$S \rightarrow AB \mid a$$
$$A \rightarrow b$$

Solution:

S is reachable and also generating. ∴ S is a useful symbol.

A is generating but not reachable. ∴ S is a useless symbol.

∴ After eliminating A (the useless symbol) the grammar is

$$S \rightarrow AB \mid a$$

Example: Eliminate useless symbols from the grammar

$S \rightarrow ABC \mid AC \mid AB$

$A \rightarrow ab$

$B \rightarrow c$

$D \rightarrow cc \mid d$

Solution:

S is reachable and also generating. ∴ S is a useful symbol.

A is reachable and also generating. ∴ A is a useful symbol.

B is reachable and also generating. ∴ B is a useful symbol.

C is reachable but not generating. ∴ C is a useless symbol.

D is not reachable but generating. ∴ D is a useless symbol.

∴ After eliminating C, D (the useless symbols) the grammar is

$S \rightarrow AB$

$A \rightarrow ab$

$B \rightarrow c$

8.2 Simplification of Grammars.

8.2.1 Chomsky Normal Form (CNF):

A Grammar where all productions are of the form

- $A \longrightarrow BC$ OR
- $A \longrightarrow b$ is called Chomsky Normal Form.

Where A, B, C are Non Terminals (Variables) and b is a terminal.

To convert a CFG to its equivalent CNF

- Eliminate useless symbols.
- Eliminate ε productions i.e., productions of the form $A \longrightarrow \varepsilon$.
- Eliminate unit productions i.e., productions of the form $A \longrightarrow B$.

8.2.2 Eliminating ε productions

Nullable Symbols:

The Symbol A is nullable if $A \Rightarrow \varepsilon$.

Finding Nullable Symbols (Algorithm):

- BASIS: If $A \longrightarrow \varepsilon$ is in G, then A is nullable.
- INDUCTION: If $B \longrightarrow C_1C_2 \ldots C_k$ is a production in G, where each C_i is nullable then B is nullable.

Eliminating ε productions

- Find Nullable Symbols.
- If $A \longrightarrow X_1X_2 \ldots X_{i-1}X_iX_{i+1} \ldots X_k$ is a production in G and X_i is a nullable symbol then rewrite the production as

$$A \longrightarrow X_1X_2 \ldots X_{i-1}X_iX_{i+1} \ldots X_k \mid X_1X_2 \ldots X_{i-1}X_{i+1}X_{i+2} \ldots X_k$$

Example: Consider the Grammar

1. $S \longrightarrow AB$
2. $A \longrightarrow aAA \mid \varepsilon$
3. $B \longrightarrow bBB \mid \varepsilon$

Eliminate ε productions from the above grammar.

Solution:

Finding nullable symbols.

Basis:

- A is a nullable symbol. ($\because A \longrightarrow \varepsilon$ is a production in the Grammar)
- B is a nullable symbol. ($\because B \longrightarrow \varepsilon$ is a production in the Grammar)

Induction:

- $S \longrightarrow AB$ is in the grammar and both A and B are nullable.
- $\therefore$S is a nullable symbol.

The set N of nullable symbols N = {S, A, B}

Eliminating ε productions: If a production contains a nullable symbol then include the productions with and without nullable symbols in the grammar.

- S ⟶ AB | A | B
- A ⟶ aAA | aA | a
- B ⟶ bBB | bB | b

Example: Consider the Grammar

1. S ⟶ aTa
2. T ⟶ ABC
3. A ⟶ aA | C
4. B ⟶ Bb | C
5. C ⟶ c | ε

Eliminate ε productions from the above grammar.

Solution:

Set of nullable symbols N = {T, A, B, C}

Grammar after eliminating ε productions

1. S ⟶ aTa | aa
2. T ⟶ ABC | AB | AC | BC | A | B | C
3. A ⟶ aA | a | C
4. B ⟶ Bb | b | C
5. C ⟶ c

8.2.3 Eliminating Unit Productions

- A Unit production is a production of the form A ⟶ B where both A and B are Variables or Non Terminals.

- If A ⟶ B is a production in the grammar and B ⟶ β is in G then replace the production A ⟶ B by the production A ⟶ β.

Example: Remove Unit Productions from the grammar

S ⟶ XY

$X \longrightarrow A$

$A \longrightarrow B \mid a$

$B \longrightarrow b$

$Y \longrightarrow T$

$T \longrightarrow Y \mid c$

Solution:

$S \longrightarrow XY$		$S \longrightarrow XY$
$X \longrightarrow B \mid a$		$X \longrightarrow b \mid a$
$A \longrightarrow b \mid a$	$\Longrightarrow$	$A \longrightarrow b \mid a$
$B \longrightarrow b$		$B \longrightarrow b$
$Y \longrightarrow Y \mid c$		$Y \longrightarrow c$
$T \longrightarrow T \mid c$		$T \longrightarrow c$

8.2.4 Converting a Grammar to CNF

1. Eliminate ε Productions.
2. Eliminate Unit Productions.
3. Eliminate Useless Symbols.
4. If the length of the Right Hand Side of a production is >= 2 and there is a terminal a on the RHS, then replace the terminal a by a nonterminal T_a and add a new production $T_a \longrightarrow a$ to the grammar.
5. If the length of the Right Hand Side of a production is > 2, i.e., the production is of the form $A \longrightarrow X_1X_2 \ldots X_k$ $(k > 2)$ rewrite the production as

$$A \longrightarrow X_1M_1$$
$$M_1 \longrightarrow X_2M_2$$
$$M_2 \longrightarrow X_3M_3$$
$$.$$
$$.$$
$$.$$
$$M_{k-2} \longrightarrow X_{k-1}X_k$$

Where $M_1, M_2, \cdots, M_{k-2}$ are the new non terminals.

Example: Convert the below CFG to CNF.

$S \longrightarrow aACa$

$A \longrightarrow B \mid a$

$B \longrightarrow C \mid c$

$C \longrightarrow cC \mid \varepsilon$

1. Eliminate ε Productions:

$N = \{C, B, A\}$

$S \longrightarrow aACa \mid aAa \mid aCa \mid aa$

$A \longrightarrow B \mid a$

$B \longrightarrow C \mid c$

$C \longrightarrow cC$

2. Eliminate Unit productions:

$S \longrightarrow aACa \mid aAa \mid aCa \mid aa$

$A \longrightarrow B \mid a$

$B \longrightarrow C \mid c$

$C \longrightarrow cC$

$S \longrightarrow aACa \mid aAa \mid aCa \mid aa$

$A \longrightarrow C \mid c \mid a$

$B \longrightarrow cC \mid c$

$C \longrightarrow cC$

$S \longrightarrow aACa \mid aAa \mid aCa \mid aa$

$A \longrightarrow cC \mid c \mid a$

$B \longrightarrow cC \mid c$

$C \longrightarrow cC$

3. Eliminate Useless Symbols.

$S \longrightarrow aACa \mid aAa \mid aCa \mid aa$

A ⟶ cC | c | a

B ⟶ cC | c

C ⟶ cC

B is a useless symbol because it is not reachable from the start symbol.

Eliminate B. The grammar is

S ⟶ aACa | aAa | aCa | aa

A ⟶ cC | c | a

C ⟶ cC

4. If the length of the Right Hand Side of a production is >= 2

S ⟶ aACa | aAa | aCa | aa

A ⟶ cC | c | a

C ⟶ cC

$\Downarrow$

S ⟶ T_aACT_a | T_aAT_a | T_aCT_a | T_aT_a

A ⟶ T_cC | c | a

C ⟶ T_cC

T_a ⟶ a

T_c ⟶ c

5.

S ⟶ T_aACT_a | T_aAT_a | T_aCT_a | T_aT_a

A ⟶ T_cC | c | a

C ⟶ T_cC

T_a ⟶ a

T_c ⟶ c

$S \longrightarrow T_aM_1 \mid T_aM_2 \mid T_aM_3 \mid T_aT_a$

$A \longrightarrow T_cC \mid c \mid a$

$C \longrightarrow T_cC$

$T_a \longrightarrow a$

$T_c \longrightarrow c$ **CNF**

$M_1 \longrightarrow AM_3$

$M_2 \longrightarrow AT_a$

$M_3 \longrightarrow CT_a$

Example: Convert the below CFG to CNF.

$S \longrightarrow aSa$

$S \longrightarrow B$

$B \longrightarrow bbC$

$B \longrightarrow bb$

$C \longrightarrow \varepsilon$

$C \longrightarrow cC$

Example: Convert the below CFG to CNF.

$S \longrightarrow ABC$

$A \longrightarrow aC \mid D$

$B \longrightarrow bB \mid \varepsilon \mid A$

$C \longrightarrow Ac \mid \varepsilon \mid Cc$

$D \longrightarrow aa$

8.3 Context Free Languages:

Languages defined by the CFG are called context free languages.

Language of a CFG:

If G is a CFG then

$$L(G) = \{ w \mid S \overset{*}{\Rightarrow} w \}$$

8.3.1 Properties of Parse Tree (Introduction)

- Let G be the CFG i.e., G = (V, T, P, S).
- Let m be the number of non terminals or variables in G.
- Let b be the branching factor i.e., the length of the longest right hand side of any production in the Grammar G. In other words b is the maximum number of children of any node in the parse tree.
- Let h be the height of the parse tree, i.e., the length of the longest path from root to any of the leaf nodes.

8.3.2 Pumping Lemma for Context Free Languages

Statement: Let L be a CFL. There exists a constant n such that , if z is any string in L and |z| >= n, then we can write z=uvwxy, subject to the following conditions:

1. |vwx| <= n. That is the middle portion is not too long.
2. $vx \neq \varepsilon$. Since v and x are the strings to be "pumped ", at least one of them must not be empty.
3. For all i >= 0 , uv^iwx^iy is in L. That is two strings v and x may be pumped in any number of times, the resulting string still will be a member of L.

Example: Prove that the language $L = \{0^n1^n2^n \mid n \geq 1\}$ is a non-context free language.

Proof:

- Let L be a context Free Language.
- Let $z = 0^n 1^n 2^n$.
- Split z as z = uvwxy, where $|vwx| \leq n$ and v and x are not both ε.
- vwx cannot contain both 0's and 2's since the last 0 and the first 2 are separated by n + 1 positions.
- We shall prove L contains some string $\notin$ L, thus contradicting the assumption.

There are two cases

1. vwx has no 2's.
    a. Then vx consists of only 0's and 1's and has at least one of these symbols.
    b. uwy has n 2's but has fewer than n 0's or fewer than n 1's or both.
    c. $\therefore$ uwy $\notin$ L and L is not a CFL.

2. vwx has no 0's.

    Similarly uwy has n 0's but has fewer than n 1's or fewer than n 2's or both.

Example: Prove that the language $L = \{0^i 1^j 2^i 3^j \mid i,j \geq 1\}$ is a non context free language.

Proof:

- Let L be a context Free Language.
- Pick $z = 0^n 1^n 2^n 3^n$.
- Split z as z = uvwxy, where $|vwx| \leq n$ and v and x are not both ε.
- vwx either contains single symbol or two adjacent symbols.
- We shall prove L contains some string $\notin$ L, thus contradicting the assumption.

There are two cases

1. vwx consists of only one symbol.

   a. Then uwy consists of n occurrences of three symbols and fewer than n occurrences of the fourth symbol.
   b. ∴ uwy ∉ L and L is not a CFL.

2. vwx consists of two adjacent symbols (say 1's and 2's).
   a. Then uwy is missing some 1's or some 2's or both.
   b. ∴ uwy ∉ L and L is not a CFL.

8.3.3 Closure Properties of Context Free Languages

1. Context Free Languages are closed Under Union.
2. Context Free Languages are closed Under Concatenation.
3. Context Free Languages are closed Under Kleene Closure.
4. Context Free Languages are closed Under Reversal.

1. Context Free Languages are closed Under Union.

If L1 and L2 are Context Free Languages then L1∪ L2 is a Context Free Language.

Proof:

L1 is a CFL ∴ L1 = L(G1) where G1 = (V1, T1, P1 , S1)
L2 is a CFL ∴ L2 = L(G2) where G2 = (V2, T2, P2 , S2)

Build a new Grammar G such that

$L(G) = L1 \cup L2 = L(G1) \cup L(G2)$

Where G = (V, T, P, S) and

$V = V1 \cup V2$

$T = T1 \cup T2$

$P = P1 \cup P2$ and

$S \longrightarrow S1 \mid S2$

2. Context Free Languages are closed Under Concatenation.

If L1 and L2 are Context Free Languages then L1L2 is a Context Free Language.

Proof:

L1 is a CFL $\therefore$ L1 = L(G1) where G1 = (V1, T1, P1 , S1)

L2 is a CFL $\therefore$ L2 = L(G2) where G2 = (V2, T2, P2 , S2)

Build a new Grammar G such that

L(G) = L1 L2 = L(G1) L(G2)

Where G = (V, T, P, S) and

V = V1 $\cup$ V2

T = T1 $\cup$ T2

P = P1 $\cup$ P2 and

S $\longrightarrow$ S1S2

3. Context Free Languages are closed Under Closure.

If L is a Context Free Language then L* is a Context Free Language.

Proof:

L is a CFL $\therefore$ L = L(G) where G = (V, T, P , S)

Build a new Grammar G1 such that

L(G1) = L* = L(G)*

Where G1 = (V1, T1, P1, S1) and

V1 = V $\cup$ S1

T1 = T

P1 = P $\cup$ { S1 $\rightarrow \varepsilon$, S1 $\rightarrow$ S1S } and

S1 $\longrightarrow$ ε | S1S

4. Context Free Languages are closed Under Reversal.

If L is a Context Free Language then L^R is a Context Free Language.

Proof:

L is a CFL $\therefore$ $L = L(G)$ where $G = (V, T, P, S)$

Build a new Grammar G1 such that

$L(G1) = L^R = L(G)^R$

Where $G1 = (V1, T1, P1, S1)$ and

$V1 = V$

$T1 = T$

$P1$ = Each production $A \rightarrow \alpha$ is replaced by $A \rightarrow \alpha^R$ and

$S1 = S$