Unit- III

3.1 Definition of CFG (Context Free Grammar) and problems

3.2 Derivation

3.3 Ambiguity in Grammar
   3.3.1 Inherent Ambiguity
   3.3.2 Ambiguous to Unambiguous CFG

3.4 Simplification of CFGs
   3.4.1 Removal of useless symbols
   3.4.2 Elimination of $\epsilon$ production
   3.4.3 Removal of unit production

3.5 Normal Forms
   3.5.1 Chomsky’s Normal Forms (CNF)
   3.5.2 Greibach Normal Form (GNF)

3.6 Closure Properties of CFL

3.7 Decision Properties of CGL

3.8 Undecidable Problems

3.9 Applications of CFG

3.10 Pumping Lemma for CFL

3.11 CFG to finite Automata
3.1 Definition of CFG (Context Free Grammar):

The CFG can be formally defined by $G = \{V,T,P,S\}$ where

- $V =$ set of non terminals or variables
- $T =$ set of terminals
- $P =$ set of production
- $S =$ start symbol

**Problems based on CFG**

1. Write CFG for the following languages:
   
   a) Having any number of a’s over the set $\{a\}$.
   
   $S \rightarrow aS$
   
   $S \rightarrow \epsilon$

   b) Having regular expression $(0+1)^*$.
   
   $S \rightarrow 0S \mid 1S$
   
   $S \rightarrow \epsilon$

   c) Containing strings of at least two a’s.
   
   $S \rightarrow Aa Aa A$
   
   $A \rightarrow aA \mid bA \mid \epsilon$

   d) $L = wcw^T$ where $w \in \{a,b\}^*$
   
   $S \rightarrow aSa$
   
   $S \rightarrow bSb$
   
   $S \rightarrow C$

   e) Which has all the strings which are all palindromes over $\{a,b\}$.
   
   $S \rightarrow aSa$
   
   $S \rightarrow bSb$
   
   $S \rightarrow a \mid b \mid \epsilon$
f) Which consists of all the strings having at least one occurrence of 000.

\[ S \rightarrow ATA \\
A \rightarrow 0A \mid 1A \mid \varepsilon \\
T \rightarrow 000 \]

g) There are no consecutive b’s, the string may or may not have consecutive a’s.

\[ S \rightarrow aS \mid bA \mid a \mid b \mid \varepsilon \\
A \rightarrow aS \mid a \mid \varepsilon \]

h) At least one occurrence of double a.

\[ S \rightarrow BAB \\
A \rightarrow aa \\
B \rightarrow aB \mid bB \mid \varepsilon \]

i) All the string of different first and last symbols over \{0,1\}.

\[ S \rightarrow 0A1 \mid 1A0 \\
A \rightarrow 0A \mid 1A \mid \varepsilon \]

j) \( L = a^n b^{2n} \) where \( n \geq 1 \).

\[ S \rightarrow aSbb \mid abb \]

k) \( L = a^x b^y \) where \( x \neq y \)

\[ S \rightarrow aSb \mid A \mid B \\
A \rightarrow aA \mid a \\
B \rightarrow bB \mid b \]

l) For the regular expression \((110+11)^*(10)^*\)

\[ S \rightarrow AB \\
A \rightarrow 110A \mid 11A \mid \varepsilon \\
B \rightarrow 10B \mid \varepsilon \]

m) For generating the integers.

\[ S \rightarrow GI \\
G \rightarrow + \mid - \\
I \rightarrow DI \mid D \\
D \rightarrow 0 \mid 1 \mid 2 \ldots \mid 9 \]
n) \( L = \{0^i1^j2^k \mid j > i+k\}. \)
\[
\begin{align*}
S & \rightarrow ABC \\
A & \rightarrow 0A1 \mid \varepsilon \\
B & \rightarrow 1B \mid 1 \\
C & \rightarrow 1C2 \mid \varepsilon
\end{align*}
\]

o) \( L = \{0^i1^j2^k \mid i = j\}. \)
\[
\begin{align*}
S & \rightarrow AB \\
A & \rightarrow 0A1 \mid \varepsilon \\
B & \rightarrow 2B \mid \varepsilon
\end{align*}
\]

p) \( L = \{0^i1^j2^k \mid j \leq k\}. \)
\[
\begin{align*}
S & \rightarrow AB \\
A & \rightarrow 0A \mid \varepsilon \\
B & \rightarrow 1B2 \mid C \\
C & \rightarrow 2C \mid \varepsilon
\end{align*}
\]
3.2 Derivation:

a. Left Most Derivation

b. Right Most Derivation

Example: Derive the string 1000111 for leftmost and rightmost derivation.

\[ S \rightarrow T00T \]

\[ T \rightarrow 0T \mid 1T \mid \epsilon \]

<table>
<thead>
<tr>
<th>Left Most Derivation</th>
<th>Right Most Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow T00T )</td>
<td>( S \rightarrow T00T )</td>
</tr>
<tr>
<td>( 1T00T \rightarrow T \rightarrow 1T )</td>
<td>( T001T \rightarrow T \rightarrow 1T )</td>
</tr>
<tr>
<td>( 10T00T \rightarrow T \rightarrow 0T )</td>
<td>( T0011T \rightarrow T \rightarrow 1T )</td>
</tr>
<tr>
<td>( 10\epsilon 00T \rightarrow T \rightarrow \epsilon )</td>
<td>( T00111T \rightarrow T \rightarrow \epsilon )</td>
</tr>
<tr>
<td>( 1000T )</td>
<td>( T0011\epsilon \rightarrow T \rightarrow \epsilon )</td>
</tr>
<tr>
<td>( 10001T \rightarrow T \rightarrow 1T )</td>
<td>( T0011 )</td>
</tr>
<tr>
<td>( 100011T \rightarrow T \rightarrow 1T )</td>
<td>( 1T00111 \rightarrow T \rightarrow 1T )</td>
</tr>
<tr>
<td>( 1000111T \rightarrow T \rightarrow 1T )</td>
<td>( 10T00111 \rightarrow T \rightarrow 0T )</td>
</tr>
<tr>
<td>( 1000111\epsilon \rightarrow T \rightarrow \epsilon )</td>
<td>( 10\epsilon 00111 \rightarrow T \rightarrow \epsilon )</td>
</tr>
<tr>
<td>( 1000111 )</td>
<td>( 1000111 )</td>
</tr>
</tbody>
</table>

3.2.1 Derivation Tree (Parse Tree):

Derivation Tree is a graphical representation for the derivation of the given production rules for a given CFG. Following are properties of any derivation tree:

a. The root node is always a node starting symbol.

b. The derivation is read from left to right.

c. The leaf nodes are always terminal nodes.

d. The interior nodes are always the non terminal nodes.
Example: Construct the derivation tree for the string “aabbabba” from the CFG given below:

\[
S \rightarrow aB \mid bA \\
A \rightarrow a \mid aS \mid bAA \\
B \rightarrow b \mid bS \mid aBB
\]

Solution:

3.3 Ambiguity in Grammar:

The grammar can be derived in either leftmost or rightmost derivation but if there exists more than one left parse tree or more than one right parse tree than the grammar is said to be ambiguous grammar.
Example: Remove ambiguity from the following example:

\[ E \rightarrow E + E \mid E * E \mid \text{id} \]

Solution:

\[ E \rightarrow E + T \]

\[ E \rightarrow T \]

\[ T \rightarrow T * F \]

\[ T \rightarrow F \]

\[ F \rightarrow \text{id} \]
3.3.1 Inherent Ambiguity:

A context free grammar is called inherently ambiguous if all the productions rules in the grammar are ambiguous.

Example:

\[
S \rightarrow XYZ \mid aaYbb \\
X \rightarrow aaY \mid aa \\
Y \rightarrow baZ \mid ba \\
Z \rightarrow bZb \mid bb
\]

3.3.2 Ambiguous to Unambiguous CFG:

3.4 Simplification of CFGs:

There are three steps for simplification of CFG (reduced grammar):

a) Removal of useless symbols.

b) Elimination of \( \varepsilon \) production.

c) Removal of unit production.

3.4.1 Removal of useless symbols:

A symbol \( P \) is useful if there exists some derivation in the following form:

\[
\star S \rightarrow \alpha P \beta \star \\
\star \alpha P \beta \rightarrow w
\]

Then \( p \) is said to be useful symbol.
Example:
S → aA | a | Bb | cC
A → aB
B → a | Aa
C → cCD
D → ddd

Solution:
S → aA | a | Bb
A → aB
B → a | Aa

3.4.2 Elimination of ε production

Example:
S → XYX
X → 0X | ε
Y → 1Y | ε

Solution:
S → XYX | XY | YX | Y | X
X → 0X | 0
Y → 1Y | 1

3.4.3 Removal of unit production.

Example:
S → 0A | 1B | C
A → 0S | 00
B → 1 | A
C → 01

Solution
S → 0A | 1B | 01
A → 0S | 00
B → 1 | 0S | 00
C → 01

Exercise: Simplify the following CFGs

S → A | 0C1
A → B | 01 | 10
C → ε | CD

Solution
S → 01 | 10
3.5 Normal Forms:

3.5.1 Chomsky’s Normal Forms (CNF):

The CNF can be defined in the following form:

<table>
<thead>
<tr>
<th>Non terminal</th>
<th>→</th>
<th>Non Terminal . Non Terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non terminal</td>
<td>→</td>
<td>Terminal</td>
</tr>
</tbody>
</table>

Example: Convert the following CFG into CNF:

1. $S \rightarrow aaaaS$
   $S \rightarrow aaaa$

   **Solution:**
   
   $S \rightarrow C_a C_1$
   $C_1 \rightarrow C_a C_2$
   $C_2 \rightarrow C_a C_3$
   $C_3 \rightarrow C_a C_4$
   $S \rightarrow C_a C_5$
   $C_5 \rightarrow C_a C_a$
   $C_a \rightarrow a$

2. $S \rightarrow aSa \mid bSb \mid a \mid b$

   **Solution:**
   
   $S \rightarrow C_b C_1$
   $C_1 \rightarrow SC_a$
   $C_a \rightarrow a$
   $S \rightarrow C_b C_2$
   $C_2 \rightarrow SC_b$
   $C_b \rightarrow b$
   $S \rightarrow a$
   $S \rightarrow b$

3.5.2 Greibach Normal Form (GNF):

The GNF can be defined in the following form:

<table>
<thead>
<tr>
<th>Non terminal</th>
<th>→</th>
<th>One Terminal . Any number of non-terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non terminal</td>
<td>→</td>
<td>One terminal</td>
</tr>
</tbody>
</table>
To convert given CFG into GNF we can use two lemmas based on which it is easy to convert given CFG to GNF:

A. Lemma 1:

\[ G = (V, T, P, S) \text{ be a given CFG and if there is a production} \]
\[ A \rightarrow Ba \text{ and } B \rightarrow \beta_1 | \beta_2 | \ldots | \beta_n \]

Then we can convert A rule to GNF as
\[ A \rightarrow \beta_1 a | \beta_2 a | \ldots | \beta_n a \]

B. Lemma 2:

\[ G = (V, T, P, S) \text{ be a given CFG and if there is a production} \]
\[ A \rightarrow Aa_1 | Aa_2 | \ldots | Aa_n | \beta_1 | \beta_2 | \ldots | \beta_n \]

Such that \( \beta_i \) do not start with \( A \) then equivalent grammar in GNF can be:
\[ A \rightarrow \beta_1 | \beta_2 | \ldots | \beta_n \]
\[ A \rightarrow \beta_1 Z | \beta_2 Z | \ldots | \beta_n Z \]
\[ Z \rightarrow a_1 | a_2 | \ldots | a_n \]
\[ Z \rightarrow a_1 Z | a_2 Z | \ldots | a_n Z \]

Example: Convert the following grammar to GNF:

Example 1.
\[ S \rightarrow abSb \]
\[ S \rightarrow aa \]

Solution:
\[ S \rightarrow aBSB \]
\[ S \rightarrow aA \]
\[ B \rightarrow b \]
\[ A \rightarrow a \]

Example 2.
\[ S \rightarrow ABA \]
\[ A \rightarrow aA | \epsilon \]
\[ B \rightarrow bB | \epsilon \]
Remove $\epsilon$ in the CFG:

$$S \rightarrow ABA \mid AB \mid BA \mid AA \mid AA \mid A \mid B$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

Remove unit production:

$$S \rightarrow ABA \mid AB \mid BA \mid AA \mid AA \mid aA \mid a \mid bB \mid b$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

In GNF we can write:

$$S \rightarrow aABA \mid aAB \mid aBA \mid AA \mid AA \mid aA \mid a \mid bBA \mid bA \mid bB \mid b$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

Example 3.

$$S \rightarrow AA \mid 0$$

$$A \rightarrow SS \mid 1$$

Step 1: (for $A$)

$$A \rightarrow SS \mid 1$$

Put the value of $S$ in left most

$$A \rightarrow AAS \mid 0S \mid 1$$

From lemma 2: $a_1 = AS$, $\beta_1 = 0S$, $\beta_2 = 1$
Step 2:

\[ S \rightarrow AA \mid 0 \]

Put the above value of A in left most:

\[ S \rightarrow 0SA \mid 1A \mid 0SZA \mid 1ZA \mid 0 \]

Put the value of A for Z:

\[ Z \rightarrow 0SS \mid 1S \mid 0SZS \mid 1ZS \]

\[ Z \rightarrow 0SSZ \mid 1SZ \mid 0SZSZ \mid 1ZSZ \]

The final solution is:

\[ S \rightarrow 0SA \mid 1A \mid 0SZA \mid 1ZA \mid 0 \]

\[ A \rightarrow 0S \mid 1 \mid 0SZ \mid 1Z \]

\[ Z \rightarrow 0SS \mid 1S \mid 0SZS \mid 1ZS \mid 0SSZ \mid 1SZ \mid 0SZSZ \mid 1ZSZ \]

Exercise: convert the following CFG into GNF

\[ S \rightarrow AB \]

\[ A \rightarrow BS \mid b \]

\[ B \rightarrow SA \mid a \]
3.6 Closure Properties of CFL:

a) The context free languages are closed under union.
b) The context free languages are closed concatenation.
c) The context free languages are closed under kleen closure.
d) The context free languages are not closed under intersection.
e) The context free languages are not closed under complement.

3.7 Decision Properties of CGL:

a) Emptiness: The given context free grammar cannot generate any string at all.
b) Finiteness: The given context free grammar generates a finite language.
c) Membership: whether given string belongs to given grammar.

3.8 Undecidable Problems:

There are no algorithms to answer these questions. Hence these problems are known as Undecidable Problems:

a) Whether or not two different context free languages define the same language?
b) Whether given CFL is ambiguous or not?
c) Whether complement of given CFL is context free language?
d) Whether the intersection of two context free languages is context free?

3.9 Applications of CFG:

When any high level program like C or PASCAL is compiled, the compiler checks the syntax of every programming statement by constructing syntax tree. And for building the syntax tree, it is necessary to write context free grammar for each statement in the program.
There are various **Parsing Techniques**:

3.10 Pumping Lemma for CFL:

Let $L$ be any context free language, then there is a constant $n$, which depends only upon $L$, such that there exist a string $w \in L$ and $|w| \geq n$ where $w = pqrst$ such that

- a) $|qs| \geq 1$
- b) $|prs| \leq n$ and
- c) For all $i \geq 0$ $pq^irs^t$ is in $L$.

**Example:** Proof whether the given language $L = \{SS^T | S \in \{a,b\}^*\}$ is context free or not?
3.11 CFG to Finite Automata:

S → 0A | 1B | 0 | 1
A → 0S | 1B | 1
B → 0A | 1S