Grammars and Context Free Language

Mathematical model of Grammar

⇒ Noam Chomsky gave a mathematical model of Grammar which is effective for writing Computer languages.

⇒ The four types of Grammar according to Noam Chomsky are:

Grammar Type	Grammar Accepted	Language Accepted	Automaton
Type 0	Unrestricted Grammar	Recursively Enumerable Language	Turing Machine
Type 1	Context Sensitive Grammar	Context Sensitive Language	Linear Bounded Automaton
Type 2	Context Free Grammar	Context Free Language	Pushdown Automata
Type 3	Regular Grammar	Regular Language	Finite State Automaton

Grammar

⇒ A Grammar 'G' can be formally described using 4 tuples as $G_1 = (V, T, S, P)$ where,

V = Set of variables or Non-Terminal Symbols

T = Set of Terminal Symbols

S = Start Symbol

P = Production rules for Terminal and Non-Terminals

A production rule has the form $\alpha \to \beta$ where α and β are strings on $V \cup T$ and atleast one symbol of α belongs to V.

Example: $G_1 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB, A \to a, B \to b\})$

$V = \{S, A, B\}$

$T = \{a, b\}$

$S = S$

$P = S \to AB, A \to a, B \to b$

Eg. $S \to AB$
$\to aB$
$\to ab$

Derivations from a Grammar

The set of all strings that can be derived from a Grammar is said to be the the Language generated from that Grammar.

Example 1: Consider the Grammar

$G_1 = (\{S, A\}, \{a, b\}, S, \{S \to aAb, aA \to aaAb, A \to \epsilon\})$

$S \to aAb$
$\to aaAbb$
$\to aaaAbbb$
$\to aaabbb$

$L(G_1) = \{a^n b^n \mid n > 0\}$

Example 2: $G_2 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB, A \to aA \mid a, B \to bB \mid b\})$

(1) $S \to AB$
$\to aB$
$\to ab$

(2) $S \to AB$
$\to aAB$
$\to aAbB$
$\to aabb$

(3) $S \to AB$
$\to aAB$
$\to aAb$
$\to aab$

(4) $S \to AB$
$\to abB$
$\to abb$

$L(G_2) = \{ab, a^2b^2, a^3b, ab^2, \ldots\}$
$= \{a^m b^n \mid m \geq 0 \text{ and } n \geq 0\}$

Context Free Language

⇒ In formal language theory, a Context Free Language is a language generated by some Context Free Grammar.

⇒ The set of all CFL is identical to the set of languages accepted by Pushdown Automata.

Context Free Grammar is defined by 4 tuples as $G = \{V, \Sigma, S, P\}$ where

V = Set of variables or Non-Terminal symbols

Σ = Set of Terminal symbols

S = Start Symbol

P = Production Rule

Context Free Grammar has Production Rule of the form

$$A \rightarrow \alpha$$

where, $\alpha = \{V \cup \Sigma\}^*$ and $A \in V$

CFL – Example

⇒ For generating a language that generates equal number of a's and b's in the form $a^n b^n$, The CFG will be defined as

$$G_1 = \{(S, A), (a, b), (S \to aAb,\ A \to aAb \mid \epsilon)\}$$

$S \to aAb$

$\to aaAbb$ (by $A \to aAb$)

$\to aaaAbbb$ (by $A \to aAb$)

$\to aaabbb$ (by $A \to \epsilon$)

$\Rightarrow a^3 b^3 \Rightarrow a^n b^n$

Method to find whether a string belongs to a Grammar or not.

① Start with the Start Symbol and choose the closest Production that matches to the given string.

② Replace the variables with its most appropriate Production. Repeat the process until the string is generated or until no other productions are left.

Example: Verify whether the Grammar
$S \to 0B|1A$, $A \to 0|0S|1AA|\wedge$, $B \to 1|1S|0BB$
generates the string 00110101

$S \to 0\underline{B}$ (S → 0B)
$\to 00BB$ (B → 0BB)
$\to 001B$ (B → 1)
$\to 0011S$ (B → 1S)
$\to 00110B$ (S → 0B)
$\to 001101\underline{S}$ (B → 1S)
$\to 00110101$ (B → 1)

Example: verify whether the Grammar
$S \to aAb$, $A \to aAb|\wedge$ generates the string aabbb.