ADESINA ALAMEEN B
18/SCI01/099

CSC 312

In formal language theory, a grammar (when the context is not given, often called a formal grammar for clarity) describes how to form strings from a language's alphabet that are valid according to the language's syntax. A grammar does not describe the meaning of the strings or what can be done with them in whatever context—only their form. A formal grammar is defined as a set of production rules for strings in a formal language.

A formal grammar is a set of rules for rewriting strings, along with a "start symbol" from which rewriting starts. Therefore, a grammar is usually thought of as a language generator. However, it can also sometimes be used as the basis for a "recognizer"—a function in computing that determines whether a given string belongs to the language or is grammatically incorrect. To describe such recognizers, formal language theory uses separate formalisms, known as automata theory.

Derivation : In formal language theory, a context-free grammar (CFG) is a formal grammar in which every production rule is of the form

$A \to \alpha$

Where A is a single nonterminal symbol, and α is a string of terminals and/or nonterminals (α can be empty). A formal grammar is considered "context free" when its production rules can be applied regardless of the context of a nonterminal. No matter which symbols surround it, the single nonterminal on the left hand side can always be replaced by the right hand side. This is what distinguishes it from a context-sensitive grammar.

A grammar **G** can be formally written as a 4-tuple (N, T, S, P) where −

- **N** or **V*N*** is a set of variables or non-terminal symbols.

- **T** or ∑ is a set of Terminal symbols.
- **S** is a special variable called the Start symbol, $S \in N$
- **P** is Production rules for Terminals and Non-terminals. A production rule has the form $\alpha \rightarrow \beta$, where α and β are strings on $V_N \cup \sum$ and least one symbol of α belongs to VN.

Production : A production or production rule in computer science is a rewrite rule specifying a symbol substitution that can be recursively performed to generate new symbol sequences. A finite set of productions is the main component in the specification of a formal grammar (specifically a generative grammar). The other components are a finite set of nonterminal symbols, a finite set (known as an alphabet) of terminal symbols that is disjoint from and a distinguished symbol that is the start symbol.

To generate a string in the language, one begins with a string consisting of only a single *start symbol*, and then successively applies the rules (any number of times, in any order) to rewrite this string. This stops when we obtain a string containing only terminals. The language consists of all the strings that can be generated in this manner. Any particular sequence of legal choices taken during this rewriting process yields one particular string in the language. If there are multiple different ways of generating this single string, then the grammar is said to be ambiguous.

For example, assume the alphabet consists of a and b, with the start symbol *S*, and we have the following rules:

1. *S* -> a *S* b
2. *S* -> ba

Sentence : A sentential form is a member of (Σ U N) that can be derived in a finite number of steps from the start symbol S ; that is, a sentential form is a member of $(w \in (\Sigma \cup N)^* \mid S \rightarrow W)$. A sentential form that contains no nonterminal symbols (i.e. is a member of Σ^*) is called a sentence.

The *language* of *G*, denoted as **L**(*G*), is defined as all those sentences that can be derived in a finite number of steps from the start symbol *S*; that is, the set $(w \in \Sigma^* \mid S \rightarrow W)$

Null Symbol: it is sometimes useful to specify that a symbol can be replaced by nothing at all. To indicate this, we use the null symbol ε, e.g., A –> B | ε.