Classification of Context Free Grammars

Context **F**ree **G**rammars (CFG) can be classified on the basis of following two properties:

1) Based on number of strings it generates.

- If CFG is generating finite number of strings, then CFG is **Non-Recursive** (or the grammar is said to be recursive grammar)
- If CFG can generate infinite number of strings then the grammar is said to be **Recursive** grammar

During Compilation, the parser uses the grammar of the language to make a parse tree(or derivation tree) out of the source code. The grammar used must be unambiguous. An ambiguous grammar must not be used for parsing.

2) Based on number of derivation trees.

- If there is only 1 derivation tree then the CFG is unambiguous.
- If there are more than 1 derivation tree, then the CFG is ambiguous.

Examples of Recursive and Non-Recursive Grammars

Recursive Grammars

```
1) S->SaS
   S->b
```

The language(set of strings) generated by the above grammar is :{b, bab, babab,…}, which is infinite.

```
2) S-> Aa
   A->Ab|c
```

The language generated by the above grammar is :{ca, cba, cbba …}, which is infinite.

Non-Recursive Grammars

```
   S->Aa
   A->b|c
```

The language generated by the above grammar is :{ba, ca}, which is finite.

Types of Recursive Grammars

Based on the nature of the recursion in a recursive grammar, a recursive CFG can be again divided into the following:

- Left Recursive Grammar (having left Recursion)
- Right Recursive Grammar (having right Recursion)
- General Recursive Grammar(having general Recursion)

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above