

Syntax Analysis::

Context-free grammar::

A context-free grammar is denoted by (V, T, S, P)

$V \rightarrow$ variables

$T \rightarrow$ Terminals

$S \rightarrow$ Start symbol

$P \rightarrow$ Productions

Variables also called as non-terminals or syntactic variables

Terminals in the syntax analysis are tokens. They are the elementary symbols of the language defined by the grammar

Production has on its left side a non-terminal which is called head of the production and on its right side it has a sequence of terminals and/or non-terminals which is called the body of the productions.

The string of terminals and non-terminals is called “sentential form”.

The string of only terminals is called a “sentence”.

Ex::

$E \rightarrow E + E$

$E \rightarrow (E)$

$E \rightarrow -E$

$E \rightarrow a \mid b$

Context-free grammar is a tool used by the parser to analyze the syntax of the source code

At any stage during parsing, when we have derived some sentential form that is not yet a sentence we will have two choices to proceed for the next derivation

1. to which non-terminal in the sentential form a production rule to be applied
2. which production rule for that non-terminal to apply

If we always choose the leftmost non-terminal in a sentential form to apply a production rule to - this is called a leftmost derivation

If we always choose the rightmost non-terminal in a sentential form to apply a production rule to - this is called a rightmost derivation

A parse tree for a grammar G is a tree where
the root is the start symbol for G
the interior nodes are the non-terminals of G
the leaf nodes are the terminal symbols of G .
the children of a node T (from left to right) correspond to the symbols on the right hand side
of some production for T in G .

If there exists several different derivations of the same string the grammar is said to be ambiguous.
In other words there exists two different parse trees for the same string.

Ex.: $E \rightarrow E+E \mid E-E \mid a \mid b \mid c$

then the sentence $a+b-c$

there will be two parse trees for this string corresponding to $(a+b)-c$ and $a+(b-c)$

1. $E \rightarrow E-E$	2. $E \rightarrow E+E$
$\rightarrow E-c$	$\rightarrow a+E$
$\rightarrow E+E-c$	$\rightarrow a+E-E$
$\rightarrow a+b-c$	$\rightarrow a+b-c$

The parser invokes lexical analyzer for tokens and the tokens are returned by the lexical analyzer to the parser

A parser reports error when the program is not consistent with grammar. It also tells about the loopholes and ambiguity in grammar .