**FIRST Set in Syntax Analysis**

* Difficulty Level : [Medium](https://www.geeksforgeeks.org/medium/)
* Last Updated : 26 Jan, 2023

 Read

 Discuss

FIRST(X) for a grammar symbol X is the set of terminals that begin the strings derivable from X.

FIRST set is a concept used in syntax analysis, specifically in the context of LL and LR parsing algorithms. It is a set of terminals that can appear immediately after a given non-terminal in a grammar.

The FIRST set of a non-terminal A is defined as the set of terminals that can appear as the first symbol in any string derived from A. If a non-terminal A can derive the empty string, then the empty string is also included in the FIRST set of A.

The FIRST set is used to determine which production rule should be used to expand a non-terminal in an LL or LR parser. For example, in an LL parser, if the next symbol in the input stream is in the FIRST set of a non-terminal, then that non-terminal can be safely expanded using the production rule that starts with that symbol.

It is worth noting that FIRST set is also used in computing FOLLOW set, which is a set of terminals that can appear immediately after a non-terminal in a grammar. FOLLOW set is used in LR parsing, which requires more information than LL parsing.

To compute the FIRST set of a grammar, one can start with all terminals having the respective terminal in their FIRST set and continue the process by adding the first terminal of the right-hand side of the production to the set of the non-terminal in the left-hand side of the production. Repeat this process until no new element can be added to any set.

FIRST set is a fundamental concept in syntax analysis, and it is used in many parsing algorithms and techniques. Its computation is a

**Rules to compute FIRST set:**

1. If x is a terminal, then FIRST(x) = { ‘x’ }
2. If x-> Є, is a production rule, then add Є to FIRST(x).
3. If X->Y1 Y2 Y3….Yn is a production,
   1. FIRST(X) = FIRST(Y1)
   2. If FIRST(Y1) contains Є then FIRST(X) = { FIRST(Y1) – Є } U { FIRST(Y2) }
   3. If FIRST (Yi) contains Є for all i = 1 to n, then add Є to FIRST(X).

**Example 1:**

Production Rules of Grammar

E -> TE’

E’ -> +T E’|Є

T -> F T’

T’ -> \*F T’ | Є

-> (E) | id

**FIRST sets**

FIRST(E) = FIRST(T) = { ( , id }

FIRST(E’) = { +, Є }

FIRST(T) = FIRST(F) = { ( , id }

FIRST(T’) = { \*, Є }

FIRST(F) = { ( , id }

**Example 2:**

Production Rules of Grammar

S -> ACB | Cbb | Ba

A -> da | BC

B -> g | Є

C -> h | Є

**FIRST sets**

FIRST(S) = FIRST(ACB) U FIRST(Cbb) U FIRST(Ba)

= { d, g, h, b, a, Є}

FIRST(A) = { d } U FIRST(BC)

= { d, g, h, Є }

FIRST(B) = { g , Є }

FIRST(C) = { h , Є }

**Notes:**

1. The grammar used above is Context-Free Grammar (CFG). Syntax of most programming languages can be specified using CFG.
2. CFG is of the form A -> B, where A is a single Non-Terminal, and B can be a set of grammar symbols ( i.e. Terminals as well as Non-Terminals)

**Advantages and Disadvantages:**

**Advantages of using FIRST set in syntax analysis include:**

* Improved parsing: FIRST set can be used to determine which production rule should be used to expand a non-terminal in an LL or LR parser, which helps to improve the accuracy and efficiency of the parsing process.
* Ambiguity resolution: FIRST set can be used to resolve ambiguities in the grammar, by determining which production rule should be used in cases where multiple production rules can apply to the same non-terminal.
* Simplified error handling: By determining which production rule should be used based on the FIRST set, an LL or LR parser can detect errors in the source code more quickly and accurately.

**Disadvantages of using FIRST set in syntax analysis include:**

* Complexity: Computing FIRST set can be a complex process, especially for grammars with many non-terminals and production rules.
* Limited applicability: FIRST set is mainly used in LL and LR parsing algorithms, and may not be applicable to other types of parsing algorithms.
* Limitations of LL parsing: LL parsing is limited in its ability to handle certain types of grammars, such as those with left-recursive rules, which can lead to an infinite loop in the parser.

Overall, the use of FIRST set in syntax analysis can improve the accuracy and efficiency of the parsing process, but it should be balanced against the complexity and limitations of the parsing algorithm being used.  
In the next article “FOLLOW sets in Compiler Design” we will see how to compute Follow sets.

This article is compiled by [**Vaibhav Bajpai**](https://disqus.com/by/vaibhav2992/). Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.