

University of Asia Pacific
Department of Computer Science & Engineering
CSE 430: Compiler Design Lab
Lab Problem 4: FIRST & FOLLOW Function

Introduction:

The construction of a predictive parser is aided by two functions associated with a grammar G. These functions, **FIRST** and **FOLLOW**, allow us to fill in the entries of a predictive parsing table for G, whenever possible. First and Follow sets are needed so that the parser can properly apply the needed production rule at the correct position.

FIRST(α)

First(α) is a set of terminal symbols that begin in strings derived from α .

Rule 1:

If FIRST (α) is the set of terminals that begin the strings derived from α .

Example 1:

Consider the production rule-
 $A \rightarrow abc \mid def\ ghi$

Then, we have-

$$\text{First}(A) = \{a, d, g\}$$

Rule 2:

If $\alpha \rightarrow \epsilon$, then ϵ , is also in FIRST (α).

Example 2:

For a production rule $X \rightarrow \epsilon$, First(X) = { ϵ }

Rule 3:

If FIRST (α) is the set of non-terminals:

If $\alpha \rightarrow Y_1 Y_2 Y_3 \dots Y_k$ is a rule then

If a is in First (Y_1) then

Add a to First (α)

If ϵ is in First (Y_1) and a is in First (Y_2) then

Add a to First (α)

If ϵ is in First (Y_1) and ϵ is in First (Y_2) and a is in First (Y_3)
then Add a to First (α)

.....

If ϵ is in First (Y_i) for all Y_i then

Add ϵ to First (α)

Example 3:

First (Y1) $\rightarrow \{a, b, \epsilon\}$

First (Y2) $\rightarrow \{c, d, \epsilon\}$

First (Y3) $\rightarrow \{e, f, \epsilon\}$

Then First (X) $\rightarrow \{a, b, c, d, e, f, \epsilon\}$

Example 4:

First (Y1) $\rightarrow \{a, b, \epsilon\}$

First (Y2) $\rightarrow \{c, d, \epsilon\}$

First (Y3) $\rightarrow \{e, f\}$

Then First (X) $\rightarrow \{a, b, c, d, e, f\}$

Example 5:

First (Y1) $\rightarrow \{a, b, \epsilon\}$

First (Y2) $\rightarrow \{c, d\}$

First (Y3) $\rightarrow \{e, f, \epsilon\}$

Then First (X) $\rightarrow \{a, b, c, d\}$

Example 6:

First (Y1) $\rightarrow \{a, b\}$

First (Y2) $\rightarrow \{c, d, \epsilon\}$

First (Y3) $\rightarrow \{e, f, \epsilon\}$

Then First (X) $\rightarrow \{a, b\}$

FOLLOW (A):

FOLLOW (A), for nonterminal A, to be the set of terminals that can appear immediately to the right of A in some sentential form. That is, the set of terminals such that there exists a derivation of the form $S \rightarrow aA\alpha\beta$ for some α and β .

To compute FOLLOW (A) for all nonterminal A, apply the following rules until nothing can be added to any FOLLOW set:

Rule 1:

Place \$ in FOLLOW(S), where S is the start symbol and \$ is the input right end marker.

Rule 2:

If there is a production $A \rightarrow \alpha B \beta$, then everything in FIRST (β), except for ϵ , is placed in FOLLOW (B).

Rule 3:

If there is a production $A \rightarrow \alpha B$, or a production $A \rightarrow aB\beta$ where FIRST (β) contains ϵ (i.e., $\beta \in \epsilon$),

Then everything in FOLLOW (A) is in FOLLOW (B).

Example 1:

$S \rightarrow aSe \mid B$	$FIRST(S) = \{a, b, c, d, \varepsilon\}$
$B \rightarrow bBCf \mid C$	$FIRST(B) = \{b, c, d, \varepsilon\}$
$C \rightarrow cCg \mid d \mid \varepsilon$	$FIRST(C) = \{c, d, \varepsilon\}$

According to Rule 1:

$FOLLOW(S) = \{\$ \}$

According to Rule 2:

$FOLLOW(C) = \{f, g\}$

$FOLLOW(B) = \{c, d, f\}$

$FOLLOW(S) = \{\$, e\}$

According to Rule 3:

$FOLLOW(C) = \{f, g\} \cup FOLLOW(B) = \{c, d, e, f, g, \$ \}$

$FOLLOW(B) = \{c, d, f\} \cup FOLLOW(S) = \{c, d, e, f, \$ \}$

$FOLLOW(S) = \{\$, e\}$

Example 2:

Consider the expression grammar stated below:

$E \rightarrow T E'$

$E' \rightarrow + T E' \mid e$

$T \rightarrow F T'$

$T' \rightarrow * F T' \mid e$

$F \rightarrow (E) \mid id$

Then:

$FIRST(E) = FIRST(T) = FIRST(F) = \{ (, id \}$

$FIRST(E') = \{ +, e \}$

$FIRST(T') = \{ *, e \}$

$FOLLOW(E) = FOLLOW(E') = \{), \$ \}$

$FOLLOW(T) = FOLLOW(T') = \{ +,), \$ \}$

$FOLLOW(F) = \{ +, *,), \$ \}$

Example 3:

Consider the expression grammar stated below:

$S \rightarrow aBDh$

$B \rightarrow cC$

$C \rightarrow bC \mid \varepsilon$

$D \rightarrow EF$

$E \rightarrow g \mid \varepsilon$

$F \rightarrow f \mid \epsilon$

The first and follow functions are as follows-

$\text{First}(S) = \{a\}$

$\text{First}(B) = \{c\}$

$\text{First}(C) = \{b, \epsilon\}$

$\text{First}(D) = \{\text{First}(E) - \epsilon\} \cup \text{First}(F) = \{g, f, \epsilon\}$

$\text{First}(E) = \{g, \epsilon\}$

$\text{First}(F) = \{f, \epsilon\}$

$\text{Follow}(S) = \{\$ \}$

$\text{Follow}(B) = \{\text{First}(D) - \epsilon\} \cup \text{First}(h) = \{g, f, h\}$

$\text{Follow}(C) = \text{Follow}(B) = \{g, f, h\}$

$\text{Follow}(D) = \text{First}(h) = \{h\}$

$\text{Follow}(E) = \{\text{First}(F) - \epsilon\} \cup \text{Follow}(D) = \{f, h\}$

$\text{Follow}(F) = \text{Follow}(D) = \{h\}$

Sample Input/Output:

You can take input from a **text file/console**. Instead of **epsilon** (ϵ) use **hash** (#) symbol.

Input 1:

```
E -> TR
R -> +T R | #
T -> F Y
Y -> *F Y | #
F -> (E) | i
```

Output 1:

```
First (E) = {(, i,}
First(R) = {+, #,}
First (T) = {(, i,}
First(Y) = {*, #,}
First (F) = {(, i,}
```

```
Follow (E) = {$, ),}
Follow(R) = {$, ),}
```

```
Follow (T) = {+, $, ),}
Follow(Y) = {+, $, ),}
Follow (F) = {*, +, $, ),}
```

Input 2:

```
S -> aSe | B
B -> bBCf | C
C -> cCg | d | #
```

Output 2:

$\text{FIRST}(S) = \{a, b, c, d, \#, \}$

$\text{FIRST}(B) = \{b, c, d, \#, \}$

$\text{FIRST}(C) = \{c, d, \#, \}$

$\text{FOLLOW}(S) = \{\$, e\}$

$\text{FOLLOW}(B) = \{c, d, e, f, \$\}$

$\text{FOLLOW}(C) = \{c, d, e, f, g, \$, \}$