FIRST function –

FIRST(α) is a set of terminal symbols that are first symbols appearing at RHS in derivation of α. If $\alpha \Rightarrow \varepsilon$ then ε is also in FIRST (α).

Following are the rules used to compute the FIRST functions.

1. If the terminal symbol a the FIRST(a) = {a}.
2. If there is a rule $X \rightarrow \varepsilon$ then FIRST(X)={ε}.
3. For the rule $A \rightarrow X_1\ X_2\ X_3\ \ldots X_k$ FIRST(A) = (FIRST(X_1) U FIRST(X_2) U FIRST(X_3)…FIRST(X_k).

   Where k $Xj \leq n$ such that $1 \leq j \leq k-1$

FOLLOW function –

FOLLOW (A) is defined as the set of terminal symbols that appear immediately to the right of A. In other words

FOLLOW(A) = { a | $S \overset{*}{\Rightarrow} \alpha\ Aa\ \beta$ where α and β are some grammar symbols may be terminal or non-terminal.

The rules for computing FOLLOW function are as given below -

1. For the start symbol S place \$ in FOLLOW(S).
2. If there is a production $A \rightarrow \alpha\ B\ \beta$ then everything in FIRST(β) without ε is to be placed in FOLLOW(B).
3. If there is a production $A \rightarrow \alpha\ B\ \beta$ or $A \rightarrow \alpha\ B$ and FIRST(β)= {ε} then FOLLOW(A) = FOLLOW(B) or FOLLOW(B)=FOLLOW(A). That means everything in FOLLOW(A) is in FOLLOW(B).

$A \rightarrow \alpha B \beta$

$A \rightarrow \alpha B$

$A \rightarrow \alpha B \beta$ fi(β)

foll(B) = {

Let us take some examples to compute FIRST and FOLLOW functions using above rules.

Example 3.7 : *Consider grammar*

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \varepsilon$

$F \rightarrow (E) \mid id$

E → TE
T → FT'
F → (E) | id

Find the FIRST and FOLLOW functions for the above grammar.

Solution : AS $E \rightarrow TE'$ is a rule in which the first symbol at RHS is T. Now $T \rightarrow FT'$ in which the first symbol at RHS is F and there is a rule for F as $F \rightarrow (E) \mid id$.

FIRST(E) = FIRST(T) = FIRST(F)

As $F \rightarrow (E)$

$F \rightarrow id$

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

left A Right

FIRST(E') = (+, ε)

As $E' \rightarrow +TE'$

(E)
F → (E)
A → αBβ

$E' \rightarrow \varepsilon$ by referring computation rule 2

foll(E) =)

The first symbol appearing at RHS of production rule for E' is added in the FIRST function.

FIRST(T')=(*, ε)

As $T' \rightarrow * TE'$ FT'

T' $E' \rightarrow \varepsilon$

The first terminal symbol appearing at RHS of production rule for T' is added in the FIRST function. Now we will compute FOLLOW function.

FOLLOW(E) –

i) As there is a rule $F \rightarrow (E)$ the symbol ')' appears immediately to the right of E. Hence ')' will be in FOLLOW(E).

ii) The computation rule is $A \rightarrow \alpha B \beta$ we can map this rule with $F \rightarrow (E)$ then $A = F$, $\alpha = ($, $B = E$, $\beta =)$.

FOLLOW(B) = FIRST(β) = FIRST( )) = {) }
FOLLOW(E) = {)}

Since E is a start symbol, add \$ to follow of E.

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

FOLLOW(E') –

$E \to TE'$

i) $E \to TE'$ the computational rule is $A \to \alpha B \beta$.

$\therefore$ A = E' α = T, B = E', β = ε then by computational rule 3 everything in FOLLOW(A) is in FOLLOW(B) i.e. everything in FOLLOW(E) is in FOLLOW(E').

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

$E' \to +TE'$

ii) $E' \to +TE'$ the computational rule is $A \to \alpha B \beta$.

$\therefore$ A = E', α = +T, B = E', β = ε then by computational rule 3 everything in FOLLOW(A) is in FOLLOW(B) i.e. everything in FOLLOW(E') is in FOLLOW(E').

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

We can observe in the given grammar that) is really following E.

FOLLOW(T) –

We have to observe two rules

$$E \to TE'$$

$$E' \to +TE'$$

i) Consider

$E \to TE'$ we will map it with $A \to \alpha B \beta$

A = E, α = ε, B = T, $\cancel{+}$ β = E' by computational rule 2 FOLLOW(B) = {FIRST (β) – ε}.

That is FOLLOW(T) = {FIRST(E') – ε}

$= \{\{+, \varepsilon\} - \varepsilon\}$

$= \{+\}$

$E' \to +TE'$

$A \to \alpha B \beta$

ii) Consider $E' \to +TE'$ we will map it with $A \to \alpha B \beta$

A = E', α = +, B = T, β = E' by computational rule 3 FOLLOW(A) = FOLLOW(B) i.e. FOLLOW(E') = FOLLOW(T)

FOLLOW(T) = {),\$}

Finally FOLLOW(T) = {+} U {),\$}

= {+ ,), \$}

We can observe in the given grammar that + and) are really following T.

FOLLOW(T') –

$T \to FT'$ (A α B β)

$$T \to FT'$$

We will map this rule with $A \to \alpha B\beta$ then $A = T$, $\alpha = F$, $B = T'$, $\beta=\varepsilon$ then FOLLOW(T')=FOLLOW(T) = {+,),$}

$T \to {*}FT'$ (α B)

$$T \to {*}FT'$$

We will map this rule with $A \to \alpha B\beta$ then $A = T$, $\alpha = {*}F$, $B = T'$, $\beta= \varepsilon^*$ then FOLLOW(T') = FOLLOW(T) = {+,),$}

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

FOLLOW(F) –

Consider $T \to FT'$ or $T' \to {*}FT'$ then by computational rule 2,

$T \to FT'$

$T \to FT'$ (α B β)

$A \to \alpha B\beta$

$A = T$, $\alpha = \varepsilon$, $B = F$, $\beta = T'$

FOLLOW(B) = {FIRST (β) – ε}

FOLLOW(F) = {FIRST(T') – ε}

FOLLOW(F) = {*}

$T' \to {*}FT'$

$T' \to {*}FT'$ (A → α B β)

$A \to \alpha B\beta$

$A = T'$, $\alpha = {*}$, $B = F$, $\beta = T'$

FOLLOW(B) = {FIRST(β) – ε}

FOLLOW(F) = {FIRST(T') – ε}

FOLLOW(F) = {*}

Consider $T' \to {*}FT'$ by computational rule 3

$T' \to {*}FT'$

$A \to \alpha B\beta$

$A = T'$, $\alpha ={*}$, $B = F$, $\beta = T'$

FOLLOW(A) = FOLLOW (B)

FOLLOW(T') = FOLLOW(F)

Hence FOLLOW(F) = {+,)$}

Finally FOLLOW(F) = {*} U {+,),$}

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

To summarize above computation

Symbols	FIRST	FOLLOW
E	{(,id}	{),$}
E'	{+, ε}	{),$}
T	{(,id}	{+,),$}
T'	{*, ε}	{+,),$}
F	{(,id}	{+,*,),$}