FIRST and FOLLOW

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 Function

To compute **FIRST**(***X***) for all grammar symbols ***X***, apply the following rules until no more terminals or ε can be added to any **FIRST** set.

1) If ***X*** is **Terminal**, then **FIRST**(***X***) is **{*X*}**.
2) If $X \longrightarrow \varepsilon$ is a production, then add ε to **FIRST**(X).
3) If X is **Nonterminal** and $X \longrightarrow Y_1\ Y_2 \ldots Y_k$ is a production, then place **a** in **FIRST**(***X***) if for some **i**, **a** is in F1RST (Y_i), and ε is in all of FIRST (Y_1). . . FIRST (Y_{i-1}); that is, $Y_1 \ldots Y_{i-1} \overset{*}{\longrightarrow} \varepsilon$. If ε is in FIRST (Y_j) for all j = 1, 2,..., k, then add ε to FIRST(***X***). For example, everything in FIRST (Y_1) is surely in FIRST(***X***). If Y_1 does not derive ε, then we add nothing more to FIRST(***X***), but if $Y_1 \overset{*}{\longrightarrow} \varepsilon$, then we add FIRST ($Y_2$) and so on.

Now, we can compute **FIRST** for any string $X_1\ X_2 \ldots X_n$ as follows. Add to FIRST($X_1\ X_2 \ldots X_n$) all the non-ε symbols of FIRST(X_1). Also add the non-ε symbols of FIRST(X_2) if ε is in FIRST(X_1), the non-ε symbols of FIRST(X_3) if ε is in both FIRST(X_1) and FIRST(X_2), and so on. Finally, add ε to FIRST($X_1\ X_2 \ldots X_n$) if, for all **i**, F1RST (X_i) contains ε.

First Function Examples:

Example1: Consider the following grammar

$S \longrightarrow aSb \mid ba \mid \varepsilon$

FIRST(S) = {**a**, **b**, ε }

Example2: Consider the following grammar

$S \longrightarrow TabS \mid X$
$T \longrightarrow cT \mid \varepsilon$
$X \longrightarrow b \mid bX$

Sol:

FIRST (S) = {**c**, **a**, **b** }

FIRST (T) = {**c**, ε }

FIRST(X) = {**b**}

Example3: Consider the following grammar

$S \longrightarrow AB \mid bS$
$A \longrightarrow aB \mid BB$
$B \longrightarrow b \mid cB$

Sol:

FIRST (S) = {a, **b, c**}

FIRST (A) = {a, **b, c**}

FIRST (B) = {**b, c**}

Example4: Consider the following grammar

$S \longrightarrow XYB \mid ccb$
$X \longrightarrow xX \mid \varepsilon$
$Y \longrightarrow yY \mid Xy \mid \varepsilon$
$B \longrightarrow bbc \mid b$

Sol:

FIRST (S) = {**c, x, y, b**}

FIRST (X) = {**x**, ε }

FIRST (Y) = {**y, x**, ε }

FIRST (B) = {**b**}

Example5: Consider the following grammar

$S \longrightarrow \mathbf{ab}S \mid \mathbf{b}X$
$X \longrightarrow \varepsilon \mid \mathbf{c}N$
$N \longrightarrow N\mathbf{b} \mid \mathbf{c}$

There is the left recursion problem, so we must solve this problem before finding the first function

$S \longrightarrow \mathbf{ab}S \mid \mathbf{b}X$
$X \longrightarrow \varepsilon \mid \mathbf{c}N$
$N \longrightarrow \mathbf{c}N'$
$N' \longrightarrow \mathbf{b}N' \mid \varepsilon$

Sol:

FIRST (S) = {**a, b**}

FIRST (X) = {**c,** ε }

FIRST (N) = {**c**}

FIRST (N') = {**b,** ε }

Follow Function

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

1) Place **$** in **FOLLOW**($S$), where S is the start symbol And **$** is the input right endmarker.
2) If there is a production $A \longrightarrow \alpha B\beta$, then everything in **FIRST** (**β**) except for ε is placed in **FOLLOW** (B).
3) If there is a production $A \longrightarrow \alpha B$, or a production $A \longrightarrow \alpha B\beta$ where **FIRST** (**β**) contains ε (i.e., $\beta \overset{*}{\longrightarrow} \varepsilon$), then everything in **FOLLOW** (A) is in **FOLLOW** (B).

Follow Function Examples:

Example1: Consider the following grammar

S ⟶ **b*XY***
X ⟶ **b | c**
Y ⟶ **b | Ɛ**

Nonterminals	First	Follow
S	**b**	**$**
X	**b, c**	**b, $**
Y	**b, Ɛ**	**$**

Example2: Consider the following grammar

S ⟶ **a*S*b | *X***
X ⟶ **c*X*b | b**
X ⟶ **b*XZ***
Z ⟶ **n**

Nonterminals	First	Follow
S	**a, b, c**	**$, b**
X	**b, c**	**b, n, $**
Z	**n**	**b, n, $**

Example3: Consider the following grammar

S ⟶ ***AB*b | bc**
A ⟶ **ab*AB*| Ɛ**
B ⟶ **bc | c*BS***

Nonterminals	First	Follow
S	**a, b, c**	**$, a, b, c**
A	**a, Ɛ**	**b, c**
B	**b, c**	**a, b, c**