COMPILER LECTURES

COMPUTER SCIENCE

3RD CLASS

M.SC. SAMER AL-YASSIN

2017-2018

LECTURE 5

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 $\in$ can be added to any **FIRST** set.

1) If ***X*** is **Terminal**, then **FIRST**(***X***) is **{*X*}**.
2) If $X \longrightarrow \in$ is a production, then add $\in$ 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 $\in$ is in all of FIRST (Y_1). . . FIRST (Y_{i-1}); that is, $Y_1 \ldots Y_{i-1} \xrightarrow{*} \in$. If $\mathcal{E}$ is in FIRST (Y_j) for all j = 1, 2,..., k, then add $\mathcal{E}$ to FIRST(***X***). For example, everything in FIRST (Y_1) is surely in FIRST(***X***). If Y_1 does not derive $\in$, then we add nothing more to FIRST(***X***), but if $Y_1 \xrightarrow{*} \in$, 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-$\in$ symbols of FIRST(X_1). Also add the non-$\in$ symbols of FIRST(X_2) if $\in$ is in FIRST(X_1), the non-$\in$ symbols of FIRST(X_3) if $\in$ is in both FIRST(X_1) and FIRST(X_2), and so on. Finally, add $\in$ to FIRST($X_1\ X_2\ \ldots\ X_n$) if, for all **i**, F1RST (X_i) contains $\in$.

First Function Examples:

Example1: Consider the following grammar

$S \longrightarrow aSb \mid ba \mid \in$

FIRST(S) = {**a**, **b**, $\in$**}**

Example2: Consider the following grammar

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

Sol:

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

FIRST (T) = {**c**, $\in$**}**

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

$S \longrightarrow XYB \mid ccb$
$X \longrightarrow xX \mid \in$
$Y \longrightarrow yY \mid Xz \mid \in$
$B \longrightarrow bbc \mid b$

Sol:

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

FIRST (X) = {**x,** $\in$**}**

FIRST (Y) = {**y, x, z,** $\in$**}**

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

Example5: Consider the following grammar

$$S \longrightarrow abS \mid bX$$
$$X \longrightarrow \mathcal{E} \mid cN$$
$$N \longrightarrow Nb \mid c$$

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

$$S \longrightarrow abS \mid bX$$
$$X \longrightarrow \in \mid cN$$
$$N \longrightarrow cN'$$
$$N' \longrightarrow bN' \mid \in$$

Sol:

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

FIRST (X) = {**c,** $\in$**}**

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

FIRST (N') = {**b,** $\in$**}**

Follow Function

To compute **FOLLOW** (A) for all Non-terminals 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 $\in$ is placed in **FOLLOW** (B).
3) If there is a production $A \longrightarrow \alpha B$, or
4) A production $A \longrightarrow \alpha B\beta$ where **FIRST** (β) contains $\in$ (i.e., $\beta \xrightarrow{*} \in$), Then everything in **FOLLOW** (A) is in **FOLLOW** (B).

Follow Function Examples:

Example1: Consider the following grammar

$S \longrightarrow bXY$
$X \longrightarrow b \mid c$
$Y \longrightarrow b \mid \in$

Nonterminals	First	Follow
S	b	$
X	b, c	b, $
Y	b, ∈	$

Example2: Consider the following grammar

$S \longrightarrow aSb \mid X$
$X \longrightarrow cXb \mid b$
$X \longrightarrow bXZ$
$Z \longrightarrow n$

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

Example3: Consider the following grammar

$S \longrightarrow ABb \mid bc$
$A \longrightarrow abAB \mid \in$
$B \longrightarrow bc \mid cBS$

Nonterminals	First	Follow
S	a, b, c	$, a, b, c
A	a, ∈	b, c
B	b, c	a, b, c

How to convert RE to CFG

•General rules:

• a+ b S →a | b

• ab S →aA

A →b

• a* S→ aS | λ

How to convert RE to RG

•Use a new nonterminal for every new character

•Each loop state turns into a recursive definition on a non-terminal

Regular Expression = ab*ab
Regular Grammar =
S→ aA
A→ bA
A→ aB
B→ b
Regular Expression = a(a+ b)*b
Regular Grammar =
S→ aA
A→ aA
A→ bA
A→ b
Regular Expression = ab*a
Regular Grammar =
S→ aA
A→ bA
A→ a