Chapter 4
Syntax Analysis

Computing First(X) : All Grammar Symbols

1. **If X is a terminal, First(X) = {X}**
2. **If $X \rightarrow \in$ is a production rule, add $\in$ to First(X)**
3. **If X is a non-terminal, and $X \rightarrow Y_1 Y_2 \ldots Y_k$ is a production rule**

   Place First(Y_1) in First(X)

   if $Y_1 \overset{*}{\Rightarrow} \in$, Place First($Y_2$) in First(X)

   if $Y_2 \overset{*}{\Rightarrow} \in$, Place First($Y_3$) in First(X)

   …

   if $Y_{k-1} \overset{*}{\Rightarrow} \in$, Place First($Y_k$) in First(X)

   NOTE: As soon as $Y_i \overset{*}{\Rightarrow} \in$,/Stop.

   Repeat above steps until no more elements are added to any First( ) set.

   Checking "$Y_j \overset{*}{\Rightarrow} \in$?" essentially amounts to checking whether $\in$ belongs to First(Y_j)

2

Example 1

Given the production rules:

S → aABb

A → c | ∈

B → d | ∈

Verify that

First(S) = {a }

First(A) = { c, ∈ }

First(B) = { d, ∈ }

3

Computing First(X) : All Grammar Symbols - continued

Informally, suppose we want to compute

First(X_1 X_2 … X_n) = First (X_1) "+"
First(X_2) if ∈ is in First(X_1) "+"
First(X_3) if ∈ is in First(X_2) "+"
…
First(X_n) if ∈ is in First(X_{n-1})

Note 1: Only add ∈ to First(X_1 X_2 … X_n) if ∈ is in First(X_i) for all i

Note 2: For First(X_1), if $X_1 \rightarrow Z_1 Z_2 \ldots Z_m$, then we need to compute First(Z_1 Z_2 … Z_m) !

4

Example 2

Given the production rules:

S → i E t SS' | a

S' → eS | ∈

E → b

Verify that

First(S) = { i, a }

First(S') = { e, ∈ }

First(E) = { b }

5

Example - 3

1. S → A a
2. A → B D
3. B → b
4. B → ϵ
5. D → d
6. D → ϵ

Example - 3

1. $S \rightarrow A\ a$
2. $A \rightarrow B\ D$
3. $B \rightarrow b$
4. $B \rightarrow \epsilon$
5. $D \rightarrow d$
6. $D \rightarrow \epsilon$

$First(S) = \{b, d, a\}$
$First(A) = \{b, d, \epsilon\}$
$First(B) = \{b, \epsilon\}$
$First(D) = \{d, \epsilon\}$

Example 4

Computing First for:

E → TE'
E' → + TE' | ∈
T → FT'
T' → * FT' | ∈
F → (E) | id

First(E)

↓

First(TE') → **First(T) U First(E')**

Not First(E') since $T \overset{*}{\not\Rightarrow} \in$

↓

First(T)

↓

First(F) U First(T')

Not First(T') since $F \overset{*}{\not\Rightarrow} \in$

↓

First(F)

↓

First((E)) U First(id)

↓

"(" and "id"

Overall: First(E) = { (, id } = First(F)
First(E') = { + , ∈ } First(T') = { * , ∈ }
First(T) → First(F) = { (, id }

Example - 5

S→ ABC| CbB| Ba
A→ da | BC
B→ g | ϵ
C→ h | ϵ

First(B) = {g, ϵ}
First(C) = {h, ϵ}

.

Example - 5

S→ ABC| CbB| Ba
A→ da | BC
B→ g | ϵ
C→ h | ϵ

First(A) → First(da) U First(BC)
→ {d} U First(B)-{ϵ} U First (C)
→{d} U {g, ϵ} –{ϵ }U {h, ϵ}
→{d, g, h, ϵ}

First(B) = {g, ϵ}
First(C) = {h, ϵ}

Example - 5

S→ ABC| CbB| Ba
A→ da | BC
B→ g | ϵ
C→ h | ϵ

First(S) → First(ABC) U First(CbB) U First(Ba)
→ {d, g, h, ϵ} U {h, b} U { g, a}
→ {a, b, d, g, h, ϵ}

First(ABC)→ First(A) –{ϵ} U First(B)-{ϵ} U First(C)
→ {d, g, h} U {g} U { h, ϵ}
→ {d, g, h, ϵ}

First(CbB) → First(C) U First(bB)
→ {h, ϵ} –{ϵ} U {b}
→ {h, b}

First(Ba) → First(B) U First(a)
→ {g, ϵ} –{ϵ }U {a}
→ { g, a}

First(A) → First(da) U First(BC)
→ {d} U First(B)-{ϵ} U First (C)
→{d} U {g, ϵ} –{ϵ }U {h, ϵ}
→{d, g, h, ϵ}

First(B) = {g, ϵ}
First(C) = {h, ϵ}

Example - 6

E → T X
X → + E
X → ε
T → int Y
T → (E)
Y → * T
Y → ε

The **First** of a terminal is that terminal.

Symbol	First
(	(
)	)
+	+
*	*
int	int
Y	
X	
T	
E	

Example - 6

E → T X
X → + E
X → ε
T → int Y
T → (E)
Y → * T
Y → ε

Symbol	First
(	(
)	)
+	+
*	*
int	int
Y	ε
X	ε
T	
E	

Example - 6

E → T X
X → + E
X → ε
T → int Y
T → (E)
Y → * T
Y → ε

Symbol	First
(	(
)	)
+	+
*	*
int	int
Y	ε, *
X	ε, +
T	int, (
E	int, (

Example 6

1. C → P F class id X Y
2. P → public
3. P → ϵ
4. F → final
5. F → ϵ
6. X → extends id
7. X → ϵ
8. Y → implements I
9. Y → ϵ
10. I → id J
11. J → , I
12. J → ϵ

Example - 6

1. C → P F class id X Y
2. P → public
3. P → ϵ
4. F → final
5. F → ϵ
6. X → extends id
7. X → ϵ
8. Y → implements I
9. Y → ϵ
10. I → id J
11. J → , I
12. J → ϵ

First(C) = {public, final, class}
First(P) = {public, ϵ}
First(F) = {final, ϵ}
First(X) = {extends, ϵ}
First(Y) = {implements, ϵ}
First(I) = {id}
First(J) = {',', ϵ}