FOLLOW SET

Lecture 11
15CSE311 Compiler Design
Kavitha K R
Department of Computer Science

FOLLOW Sets

Definition: Given a grammar G = (V, T, P, S), for any nonterminal $A \in V$:

FOLLOW(A) = { $\mathbf{a} \in T \mid S \Rightarrow^* \alpha A \mathbf{a} \beta$ for some α, β}.

i.e., FOLLOW(A) contains those terminals that can appear after A in something derivable from the start symbol S.

if $S \Rightarrow^* \alpha A$ then $ is also in FOLLOW(A).   ($ ≡ EOF, "end of input.")

Example:

$E \rightarrow E + E \mid \mathbf{id}$

FOLLOW(E) = { +, $ }.

Computing FOLLOW Sets

Given a grammar G = (V, T, P, S):

1. add $ to FOLLOW(S);

2. repeat {

    for each production $A \rightarrow \alpha B \beta$ in P, add every non-ε symbol in FIRST(β) to FOLLOW(B).

    for each production $A \rightarrow \alpha B \beta$ in P, where $\varepsilon \in$ FIRST(β), add everything in FOLLOW(A) to FOLLOW(B).

    for each production $A \rightarrow \alpha B$ in P, add everything in FOLLOW(A) to FOLLOW(B).

    } until no change to any FOLLOW set.

Example (FOLLOW Sets)

X → YZ | a
Y → b | ε
Z → c | ε

X is start symbol: add $ to FOLLOW(X);

X → YZ, so add everything in FOLLOW(X) to FOLLOW(Z).
► add $ to FOLLOW(Z).

X → YZ, so add every non-ε symbol in FIRST(Z) to FOLLOW(Y).
► add c to FOLLOW(Y).

X → YZ and ε ∈ FIRST(Z), so add everything in FOLLOW(X) to FOLLOW(Y).
► add $ to FOLLOW(Y).