FIRST and *FOLLOW* Computation Example

- Consider the following grammar
  $S' \to S\$,\ S \to aAS \mid c,\ A \to ba \mid SB,\ B \to bA \mid S$
- $FIRST(S') = FIRST(S) = \{a, c\}$ because
  $S' \Rightarrow S\$ \Rightarrow \underline{c}\$$, and $S' \Rightarrow S\$ \Rightarrow \underline{a}AS\$ \Rightarrow \underline{a}baS\$ \Rightarrow \underline{a}bac\$$
- $FIRST(A) = \{a, b, c\}$ because
  $A \Rightarrow \underline{b}a$, and $A \Rightarrow SB$, and therefore all symbols in $FIRST(S)$ are in $FIRST(A)$
- $FOLLOW(S) = \{a, b, c, \$\}$ because
  $S' \Rightarrow \underline{S\$}$,
  $S' \Rightarrow^* aAS\$ \Rightarrow a\underline{S}BS\$ \Rightarrow aS\underline{b}AS\$$,
  $S' \Rightarrow^* a\underline{S}BS\$ \Rightarrow a\underline{S}SS\$ \Rightarrow aS\underline{a}ASS\$$,
  $S' \Rightarrow^* a\underline{S}SS\$ \Rightarrow aS\underline{c}S\$$
- $FOLLOW(A) = \{a, c\}$ because
  $S' \Rightarrow^* a\underline{A}S\$ \Rightarrow aA\underline{a}AS\$$,
  $S' \Rightarrow^* a\underline{A}S\$ \Rightarrow aA\underline{c}$

Computation of *FIRST*: Terminals and Nonterminals

```
{
 for each (a ∈ T) FIRST(a) = {a}; FIRST(ϵ) = {ϵ};
 for each (A ∈ N) FIRST(A) = ∅;
 while (FIRST sets are still changing) {
     for each production p {
        Let p be the production A → X1X2...Xn;
        FIRST(A) = FIRST(A) ∪ (FIRST(X1) - {ϵ});
        i = 1;
        while (ϵ ∈ FIRST(Xi) && i ≤ n − 1) {
            FIRST(A) = FIRST(A) ∪ (FIRST(Xi+1 − {ϵ}); i + +;
        }
        if (i == n) && (ϵ ∈ FIRST(Xn))
           FIRST(A) = FIRST(A) ∪{ϵ}
 }
}
```

Computation of $FIRST(\beta)$: β, a string of Grammar Symbols

```
{ /* It is assumed that FIRST sets of terminals and nonterminals
    are already available /*
    FIRST(β) = ∅;
    while (FIRST sets are still changing) {
        Let β be the string X1X2...Xn;
        FIRST(β) = FIRST(β) ∪ (FIRST(X1) - {ε});
        i = 1;
        while (ε ∈ FIRST(Xi) && i ≤ n − 1) {
            FIRST(β) = FIRST(β) ∪ (FIRST(Xi+1 − {ε}); i + +;
        }
        if (i == n) && (ε ∈ FIRST(Xn))
            FIRST(β) = FIRST(β) ∪{ε}
    }
}
```

FIRST Computation: Algorithm Trace - 1

- Consider the following grammar
  $S' \rightarrow S\$, \; S \rightarrow aAS \mid \epsilon, \; A \rightarrow ba \mid SB, \; B \rightarrow cA \mid S$
- Initially, FIRST(S) = FIRST(A) = FIRST(B) = $\emptyset$
- Iteration 1
  - FIRST(S) = $\{a, \epsilon\}$ from the productions $S \rightarrow aAS \mid \epsilon$
  - FIRST(A) = $\{b\} \cup$ FIRST(S) - $\{\epsilon\} \cup$ FIRST(B) - $\{\epsilon\}$ = $\{b, a\}$
    from the productions $A \rightarrow ba \mid SB$
    (since $\epsilon \in$ FIRST(S), FIRST(B) is also included;
    since FIRST(B)=ϕ, ϵ is not included)
  - FIRST(B) = $\{c\} \cup$ FIRST(S) - $\{\epsilon\} \cup \{\epsilon\}$ = $\{c, a, \epsilon\}$
    from the productions $B \rightarrow cA \mid S$
    (ϵ is included because $\epsilon \in$ FIRST(S))

FIRST Computation: Algorithm Trace - 2

- The grammar is
  $S' \to S\$,\ S \to aAS \mid \epsilon,\ A \to ba \mid SB,\ B \to cA \mid S$
- From the first iteration,
  FIRST(S) = $\{a, \epsilon\}$, FIRST(A) = $\{b, a\}$, FIRST(B) = $\{c, a, \epsilon\}$
- Iteration 2
  (values stabilize and do not change in iteration 3)
  - FIRST(S) = $\{a, \epsilon\}$ (no change from iteration 1)
  - FIRST(A) = $\{b\} \cup$ FIRST(S) - $\{\epsilon\} \cup$ FIRST(B) - $\{\epsilon\} \cup \{\epsilon\}$
    = $\{b, a, c, \epsilon\}$ (changed!)
  - FIRST(B) = $\{c, a, \epsilon\}$ (no change from iteration 1)

Computation of *FOLLOW*

```
{ for each (X ∈ N ∪ T) FOLLOW(X) = ∅;
  FOLLOW(S) = {$}; /* S is the start symbol of the grammar */
  repeat {
    for each production A → X1X2...Xn {/* Xi ≠ ε */
      FOLLOW(Xn) = FOLLOW(Xn) ∪ FOLLOW(A);
      REST = FOLLOW(A);
      for i = n downto 2 {
        if (ε ∈ FIRST(Xi)) { FOLLOW(Xi−1) =
            FOLLOW(Xi−1) ∪ (FIRST (Xi) − {ε})∪ REST;
            REST = FOLLOW(Xi−1);
        } else { FOLLOW(Xi−1) = FOLLOW(Xi−1) ∪ FIRST (Xi) ;
                 REST = FOLLOW(Xi−1); }
      }
    }
  } until no FOLLOW set has changed
}
```

FOLLOW Computation: Algorithm Trace

- Consider the following grammar
  $S' \rightarrow S\$,\ S \rightarrow aAS \mid \epsilon,\ A \rightarrow ba \mid SB,\ B \rightarrow cA \mid S$
- Initially, $follow(S)$ = {\$}; $follow(A) = follow(B) = \emptyset$
  $first(S) = \{a, \epsilon\}$; $first(A) = \{a, b, c, \epsilon\}$; $first(B) = \{a, c, \epsilon\}$;
- Iteration 1 /* In the following, $x \cup = y$ means $x = x \cup y$ */
  - $S \rightarrow aAS$: $follow(S) \cup = \{\$\}$; $rest = follow(S) = \{\$\}$
    $follow(A) \cup = (first(S) - \{\epsilon\}) \cup rest = \{a, \$\}$
  - $A \rightarrow SB$: $follow(B) \cup = follow(A) = \{a, \$\}$
    $rest = follow(A) = \{a, \$\}$
    $follow(S) \cup = (first(B) - \{\epsilon\}) \cup rest = \{a, c, \$\}$
  - $B \rightarrow cA$: $follow(A) \cup = follow(B) = \{a, \$\}$
  - $B \rightarrow S$: $follow(S) \cup = follow(B) = \{a, c, \$\}$
  - At the end of iteration 1
    $follow(S) = \{a, c, \$\}$; $follow(A) = follow(B) = \{a, \$\}$

FOLLOW Computation: Algorithm Trace (contd.)

- $\mathit{first}(S) = \{a, \epsilon\}$; $\mathit{first}(A) = \{a, b, c, \epsilon\}$; $\mathit{first}(B) = \{a, c, \epsilon\}$;
- At the end of iteration 1
  $\mathit{follow}(S) = \{a, c, \$\}$; $\mathit{follow}(A) = \mathit{follow}(B) = \{a, \$\}$
- Iteration 2
- $S \rightarrow aAS$: $\mathit{follow}(S) \cup = \{a, c, \$\}$;
  $\mathit{rest} = \mathit{follow}(S) = \{a, c, \$\}$
  $\mathit{follow}(A) \cup = (\mathit{first}(S) - \{\epsilon\}) \cup \mathit{rest} = \{a, c, \$\}$ (changed!)
- $A \rightarrow SB$: $\mathit{follow}(B) \cup = \mathit{follow}(A) = \{a, c, \$\}$ (changed!)
  $\mathit{rest} = \mathit{follow}(A) = \{a, c, \$\}$
  $\mathit{follow}(S) \cup = (\mathit{first}(B) - \{\epsilon\}) \cup \mathit{rest} = \{a, c, \$\}$ (no change)
- At the end of iteration 2
  $\mathit{follow}(S) = \mathit{follow}(A) = \mathit{follow}(B) = \{a, c, \$\}$;
- The *follow* sets do not change any further