Compiler Design

Fatemeh Deldar

Isfahan University of Technology

1403-1404

FIRST and FOLLOW

- The construction of both top-down and bottom-up parsers is aided by two functions, **FIRST** and **FOLLOW**, associated with a grammar G

- ***FIRST*(*X*) for all grammar symbols *X***

1. If X is a terminal, then $\text{FIRST}(X) = \{X\}$.

2. If X is a nonterminal and $X \rightarrow Y_1 Y_2 \cdots Y_k$ is a production for some $k \geq 1$, then place a in $\text{FIRST}(X)$ if for some i, a is in $\text{FIRST}(Y_i)$, and ϵ is in all of $\text{FIRST}(Y_1), \ldots, \text{FIRST}(Y_{i-1})$; that is, $Y_1 \cdots Y_{i-1} \stackrel{*}{\Rightarrow} \epsilon$. If ϵ is in $\text{FIRST}(Y_j)$ for all $j = 1, 2, \ldots, k$, then add ϵ to $\text{FIRST}(X)$. For example, everything in $\text{FIRST}(Y_1)$ is surely in $\text{FIRST}(X)$. If Y_1 does not derive ϵ, then we add nothing more to $\text{FIRST}(X)$, but if $Y_1 \stackrel{*}{\Rightarrow} \epsilon$, then we add $\text{FIRST}(Y_2)$, and so on.

3. If $X \rightarrow \epsilon$ is a production, then add ϵ to $\text{FIRST}(X)$.

FIRST and FOLLOW

- ***FOLLOW*(*A*) for all nonterminals *A***

1. Place \$ in FOLLOW(S), where S is the start symbol, and \$ is the input right endmarker.
2. If there is a production $A \to \alpha B \beta$, then everything in FIRST(β) except ϵ is in FOLLOW(B).
3. If there is a production $A \to \alpha B$, or a production $A \to \alpha B \beta$, where FIRST(β) contains ϵ, then everything in FOLLOW(A) is in FOLLOW(B).

FIRST and FOLLOW

- **Example**

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathbf{id}
\end{array}
$$

  - $FIRST(F) = FIRST(T) = FIRST(E) = \{(, id\}$
  - $FIRST(E') = \{+, \epsilon\}$
  - $FIRST(T') = \{*, \epsilon\}$
  - $FOLLOW(E) = FOLLOW(E') = \{), \$\}$
  - $FOLLOW(T) = FOLLOW(T') = \{+,), \$\}$
  - $FOLLOW(F) = \{+, *,), \$\}$

$LL(1)$ Grammars

- $\boldsymbol{LL(1)}$
  - The first "L" in $LL(1)$ stands for scanning the input from left to right
  - The second "L" for producing a leftmost derivation
  - The "1" for using one input symbol of lookahead at each step to make parsing action decisions
- **A grammar $\boldsymbol{G}$ is $\boldsymbol{LL(1)}$ if and only if whenever $\boldsymbol{A \to \alpha | \beta}$ are two distinct productions of $\boldsymbol{G}$, the following conditions hold:**
  1. $FIRST(\alpha)$ and $FIRST(\beta)$ are disjoint sets
  2. If ϵ is in $FIRST(\beta)$, then $FIRST(\alpha)$ and $FOLLOW(A)$ are disjoint sets, and likewise if ϵ is in $FIRST(\alpha)$
- Predictive parsers can be constructed for $LL(1)$ grammars

$LL(1)$ Grammars

- **Construction of a predictive parsing table**

INPUT: Grammar G.

OUTPUT: Parsing table M.

METHOD: For each production $A \to \alpha$ of the grammar, do the following:

1. For each terminal a in $\text{FIRST}(\alpha)$, add $A \to \alpha$ to $M[A, a]$.
2. If ϵ is in $\text{FIRST}(\alpha)$, then for each terminal b in $\text{FOLLOW}(A)$, add $A \to \alpha$ to $M[A, b]$. If ϵ is in $\text{FIRST}(\alpha)$ and $\$$ is in $\text{FOLLOW}(A)$, add $A \to \alpha$ to $M[A, \$]$ as well.

- If, after performing the above, there is no production at all in $M[A, a]$, then set $M[A, a]$ to error

$LL(1)$ Grammars

- **Example**

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \textbf{id}
\end{array}
$$

NON - TERMINAL	INPUT SYMBOL					
	id	$+$	$*$	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow \textbf{id}$			$F \rightarrow (E)$		

$LL(1)$ Grammars

- For every $LL(1)$ grammar, each parsing-table entry uniquely identifies a production or an error
- **Example**

$$
\begin{aligned}
S &\rightarrow iEtSS' \mid a \\
S' &\rightarrow eS \mid \epsilon \\
E &\rightarrow b
\end{aligned}
$$

NON - TERMINAL	INPUT SYMBOL					
	a	b	e	i	t	$\$$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$\boxed{S' \rightarrow \epsilon \quad S' \rightarrow eS}$			$S' \rightarrow \epsilon$
E		$E \rightarrow b$				