Compiladores: Parsing descendente

Christiano Braga
Universidade Federal Fluminense

Março 2021

Algoritmo descendente

void $A()$ {
    Choose an A-production, $A \to X_1 X_2 \cdots X_k$;
    for ($i = 1$ to k) {
        if (X_i is a nonterminal)
            call procedure $X_i()$;
        else if (X_i equals the current input symbol a)
            advance the input to the next symbol;
        else /* an error has occurred */;
    }
}

Figure 1: Procedimento para parser descendente

Exemplo

$$
\begin{array}{lcl}
S & \rightarrow & c\ A\ d \\
A & \rightarrow & a\ b \mid a
\end{array}
$$

Figure 2: Exemplo de parsing descendente

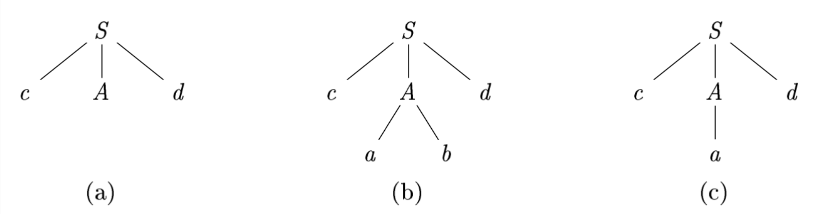

Figure 3: Exemplo de parsing descendente: derivação

FIRST I

1. If X is a terminal, then $\text{FIRST}(X) = \{X\}$.
2. If X is a nonterminal and $X \to Y_1 Y_2 \cdots Y_k$ 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, $Y_1 \cdots Y_{i-1} \overset{*}{\Rightarrow} \epsilon$. 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 $Y_1 \overset{*}{\Rightarrow} \epsilon$, then we add $\text{FIRST}(Y_2)$, and so on.
3. If $X \to \epsilon$ is a production, then add ϵ to $\text{FIRST}(X)$.

Figure 4: FIRST de um símbolo

FIRST II

Now, we can compute FIRST for any string $X_1X_2 \cdots X_n$ as follows. Add to FIRST$(X_1X_2 \cdots X_n)$ all non-ϵ symbols of FIRST(X_1). Also add the non-ϵ symbols of FIRST(X_2), if ϵ is in FIRST(X_1); the non-ϵ symbols of FIRST(X_3), if ϵ is in FIRST(X_1) and FIRST(X_2); and so on. Finally, add ϵ to FIRST$(X_1X_2 \cdots X_n)$ if, for all i, ϵ is in FIRST(X_i).

Figure 5: FIRST de uma string

FOLLOW

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).

Exemplo I

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

Figure 6: Gramática 4.28

Exemplo II

1. $\text{FIRST}(F) = \text{FIRST}(T) = \text{FIRST}(E) = \{(, \textbf{id}\}$. To see why, note that the two productions for F have bodies that start with these two terminal symbols, **id** and the left parenthesis. T has only one production, and its body starts with F. Since F does not derive ϵ, $\text{FIRST}(T)$ must be the same as $\text{FIRST}(F)$. The same argument covers $\text{FIRST}(E)$.

2. $\text{FIRST}(E') = \{+, \epsilon\}$. The reason is that one of the two productions for E' has a body that begins with terminal $+$, and the other's body is ϵ. Whenever a nonterminal derives ϵ, we place ϵ in FIRST for that nonterminal.

3. $\text{FIRST}(T') = \{*, \epsilon\}$. The reasoning is analogous to that for $\text{FIRST}(E')$.

4. $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{), \$\}$. Since E is the start symbol, $\text{FOLLOW}(E)$ must contain \$. The production body $(\ E\ )$ explains why the right parenthesis is in $\text{FOLLOW}(E)$. For E', note that this nonterminal appears only at the ends of bodies of E-productions. Thus, $\text{FOLLOW}(E')$ must be the same as $\text{FOLLOW}(E)$.

5. $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{+,), \$\}$. Notice that T appears in bodies only followed by E'. Thus, everything except ϵ that is in $\text{FIRST}(E')$ must be in $\text{FOLLOW}(T)$; that explains the symbol $+$. However, since $\text{FIRST}(E')$ contains ϵ (i.e., $E' \stackrel{*}{\Rightarrow} \epsilon$), and E' is the entire string following T in the bodies of the E-productions, everything in $\text{FOLLOW}(E)$ must also be in $\text{FOLLOW}(T)$. That explains the symbols \$ and the right parenthesis. As for T', since it appears only at the ends of the T-productions, it must be that $\text{FOLLOW}(T') = \text{FOLLOW}(T)$.

6. $\text{FOLLOW}(F) = \{+, *,), \$\}$. The reasoning is analogous to that for T in point (5).

Tabela de parsing

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.

Figure 8: Algoritmo para construção da tabela de parsing

Exemplo para a Grm. 4.28

Non - terminal	Input Symbol					
	id	$+$	$*$	(	)	\$
E	$E \to TE'$			$E \to TE'$		
E'		$E' \to +TE'$			$E' \to \epsilon$	$E' \to \epsilon$
T	$T \to FT'$			$T \to FT'$		
T'		$T' \to \epsilon$	$T' \to *FT'$		$T' \to \epsilon$	$T' \to \epsilon$
F	$F \to \mathbf{id}$			$F \to (E)$		

Figure 9: Tab. de parsing para Grm. 4.28