

第9讲(语法制导翻译 2)要点

- ▶语义分析要解决的两个问题
 - ▶ 语义信息的表示: 语义属性 (表达式的值val、变量的类型type、...)
 - >语义信息的计算: 语义规则

SDD = CFG + 语义属性 + 语义规则

- ▶ 无循环依赖SDD的判定
 - > 充分条件 继承属性不依赖右兄弟节点的属性和父节点的综合属性
 - ➤ S-SDD if LR文法 then LR分析+语义翻译
 - ► L-SDD if LL(1)文法 then LL(1)分析+语义翻译 bottom_up分析+语义翻译

第9讲(语法制导翻译_2)要点

▶ 语法制导翻译方案 (SDT) SDD的具体实施方案

SDD定义了各属性的计算方法(计算规则)怎么算?

SDT进一步明确了各属性的计算时机(计算顺序) 怎么算? +何时算?

SDD

	无循环依赖SDD的SDT

	产生式	语义规则
(1)	$T \rightarrow F T'$	T'.inh = F.val
		T.val = T'.syn
(2)	$T' \rightarrow * F T_I'$	T_1' .inh = T' .inh \times F .val
		$T'.syn = T_1'.syn$
(3)	$T' \rightarrow \varepsilon$	T'.syn = T'.inh
(4)	$F \rightarrow \text{digit}$	F.val = digit.lexval

SDT

- 1) $T \rightarrow F \{ T'.inh = F.val \} T' \{ T.val = T'.syn \}$
- 2) $T' \rightarrow *F \{ T_1'.inh = T'.inh \times F.val \} T_1' \{ T'.syn = T_1'.syn \}$
- 3) $T' \rightarrow \varepsilon \{ T'.syn = T'.inh \}$
- 4) $F \rightarrow \text{digit} \{ F.val = \text{digit.} lexval \}$

S-SDD的SDT

- 》综合属性的计算时机
 - ▶所有子节点分析完
 - ▶语义动作置于产生式末尾

L-SDD的SDT

- ▶继承属性的计算时机
 - ▶即将分析A之前
 - ▶语义动作置于紧靠在A的本次出现之前的位置上
- ▶综合属性的计算时机
 - ▶所有子节点分析完毕
 - ▶语义动作置于产生式末尾

例

>L-SDD

一 产生式 语义规则

(1)
$$T \rightarrow F'T'$$
 $T'.inh = F.val$ $T.val = T'.syn$

(2) $T' \rightarrow *F'T'$ $T_1'.inh = T'.inh \times F.val$ $T'.syn = T_1'.syn$

(3) $T' \rightarrow \varepsilon$ $T'.syn = T'.inh$

(4) $F \rightarrow \text{digit}$ $F.val = \text{digit.} lexval$

>SDT

T → F { T'.inh = F.val } T' { T.val = T'.syn }
 T' → *F { T₁'.inh = T'.inh × F.val } T₁' { T'.syn = T₁'.syn }
 T' → ε { T'.syn = T'.inh }
 F → digit { F.val = digit.lexval }

S-SDD的自底向上翻译

 $S ext{-}SDD$: if LR文法 then LR分析+语义翻译 if LR文法 + $S ext{-}SDD$ then LR分析+语义翻译

- ▶语法分析器的扩展
 - >为每个栈记录增加属性值字段, 存放文法符号的综合属性值
 - 产在每次归约时调用计算综合属性值的语义子程序

L-SDD的自顶向下翻译

LL(1)文法 + *L-SDD*

```
输入:
1) T \rightarrow F \{ T'.inh = F.val \} T' \{ T.val = T'.syn \}
2) T' \rightarrow *F \{ T_1'.inh = T'.inh \times F.val \} T_1' \{ T'.syn = T_1'.syn \}
3) T' \rightarrow \varepsilon \{ T'.syn = T'.inh \}
                                                                                                        digit * digit
4) F \rightarrow \text{digit} \{ F.val = \text{digit.} lexval \}
                                                  T_{val=15}
                                                                      inh=3 T'_{syn=15} \{ T.val = T'.syn \}
                                           \{ T'.inh = F.val \}
                         F_{val=3}
                      digit { F.val = digit.lexval }
                                                                    F_{val} = T'.inh \times F.val \} T' \{T'.syn = T_1'.syn \}
inh=15 \quad syn=15
                        (3)
                                                                   digit
                                                                              \{F.val = digit.lexval\} \varepsilon \{T'.syn = T'.inh\}
                                                                    (5)
```

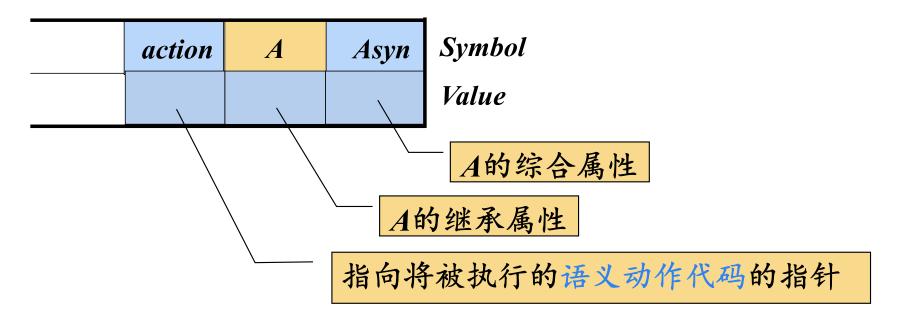
L-SDD的自顶向下翻译 LL(1)文法 + L-SDD

- 产在预测分析的同时实现语义翻译
 - 产在非递归的预测分析过程中进行翻译
 - 产在递归的预测分析过程中进行翻译

在非递归的预测分析过程中进行翻译

户扩展语法分析栈

- (1) 增加属性值 (value) 字段
- (2) 将继承属性和综合属性存放在不同的记录中
- (3) 增加动作记录用来存放语义动作代码的指针
- (4) 不光是动作记录, 其实分析栈中的每一个记录都对应着一段执行代码



分析栈中的每一个记录都对应着一段执行代码

- ▶综合记录出栈时,要将综合属性值复制给后面特定 的语义动作
- ▶变量展开时(即变量本身的记录出栈时),如果其 含有继承属性,则要将继承属性值复制给后面特定 的语义动作

L-SDD的设计思路分析

例:L-SDD

$$\begin{array}{c}
T \to T * F \mid F \\
F \to \text{digit}
\end{array}$$

	产生式	语义规则
(1)	$T \rightarrow F T'$	T'.inh = F.val
		T.val = T'.syn
(2)	$T' \rightarrow * F T_1'$	$T_1'.inh = T'.inh \times F.val$
		$T'.syn = T_1'.syn$
(3)	$T' \rightarrow \varepsilon$	T'.syn = T'.inh
(4)	$F \rightarrow \text{digit}$	F.val = digit.lexval

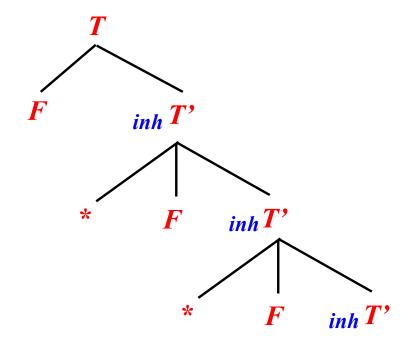
语义翻译的主要任务: 计算T的值

L-SDD的设计思路分析

例:L-SDD

	产生式	语义规则
(1)	$T \rightarrow F T'$	T'.inh = F.val
		T.val = T'.syn
(2)	$T' \rightarrow * F T_1'$	$T_1'.inh = \underline{T'.inh \times F.val}$
		$T'.syn = T_1'.syn$
(3)	$T' \rightarrow \varepsilon$	T'.syn = T'.inh
(4)	$F \rightarrow \text{digit}$	F.val = digit.lexval

语义翻译的主要任务: 计算T的值



L-SDD的设计思路分析

例:L-SDD

	产生式	语义规则
(1)	$T \rightarrow F T'$	T'.inh = F.val
		T.val = T'.syn
(2)	$T' \rightarrow * F T_1'$	$T_1'.inh = T'.inh \times F.val$
		$T'.syn = T_1'.syn$
(3)	$T' \rightarrow \varepsilon$	T'.syn = T'.inh
(4)	$F \rightarrow \text{digit}$	F.val = digit.lexval

语义翻译的主要任务: 计算T的值

