

第5章 语法制导翻译

语法制导定义和语法制导翻译

语法制导定义

- 将文法符号和某些属性相关联
- 并通过**语义规则**来描述如何计算属性的值
- $E \rightarrow E_1 + T$ $E.code = E_1.code || T.code || '+'$

SDD是对上下文无关文法的推广。将每个文法符号和一个语义属性集合相关联，将每个产生式和一组语义规则相关联，这些规则用于计算该产生式中各文法符号的属性值。

主要就是为CFG中的文法符号设置语义属性，具体来说，主要就是将一些语义属性附加到代表语言构造的文法符号上，从而把信息和语言构造联系起来。

语法制导翻译

- **在产生式体中加入语义动作，并在适当的时候执行这些语义动作**
- $E \rightarrow E_1 + T$ $\{ \text{print '+'}; \}$

继承属性和综合属性

- **综合属性**(synthesized attribute)：在分析树结点N上的非终结符号A的属性值由N对应的产生式所关联的语义规则来定义
 - 通过N的**子结点**或**N本身**的属性值来定义
- **继承属性**(inherited attribute)：结点N的属性值由N的父结点所关联的语义规则来定义
 - 依赖于N的**父结点**、**N本身**和N的**兄弟结点**上的属性值
- 不允许N的继承属性通过N的子结点上的属性来定义，但是允许N的综合属性依赖于N本身的继承属性
- 终结符号有综合属性（由词法分析获得），但是没有继承属性

S属性的SDD

只包含综合属性的SDD称为S属性的SDD

- 每个语义规则都根据产生式体中的属性值来计算头部非终结符号的属性值

S属性的SDD可以和LR语法分析器一起实现

- 栈中的状态可以附加相应的属性值
- 在进行归约时，按照语义规则计算归约得到的符号的属性值

S属性的SDD一定可以按照自底向上的方式求值。

L属性的SDD

- 每个属性
 - 要么是综合属性
 - 要么是继承属性，且产生式 $A \rightarrow X_1 X_2 \cdots X_n$ 中计算 X_i 的规则只能使用
 - A的继承属性
 - X_i 左边的文法符号 X_j 的继承属性或综合属性
 - X_i 自身的继承或综合属性，且这些属性之间的依赖关系不形成环

语法分析树上的SDD求值

实践中很少先构造语法分析树再进行SDD求值，但在分析树上求值有助于翻译方案的可视化，便于理解

注释语法分析树：包含了各个结点的各属性值的语法分析树

步骤：对于任意的输入串，首先构造出相应的分析树。给各个结点（根据其文法符号）加上相应的属性值。按照语义规则计算这些属性值即可。

抽象语法树

每个结点代表一个语法结构；对应于一个运算符

节点的每个子结点代表其子结构；对应于运算分量

表示这些子结构按照特定方式组成了较大的结构

可以忽略掉一些标点符号等非本质的东西

产生式	语义规则
1) $E \rightarrow E_1 + T$	$E.node = \text{new Node}('+', E_1.node, T.node)$
2) $E \rightarrow E_1 - T$	$E.node = \text{new Node}('-', E_1.node, T.node)$
3) $E \rightarrow T$	$E.node = T.node$
4) $T \rightarrow (E)$	$T.node = E.node$
5) $T \rightarrow \text{id}$	$T.node = \text{new Leaf}(\text{id}, \text{id.entry})$
6) $T \rightarrow \text{num}$	$T.node = \text{new Leaf}(\text{num}, \text{num.val})$

消除左递归：

产生式	语义规则
1) $E \rightarrow T E'$	$E.node = E'.syn$ $E'.inh = T.node$
2) $E' \rightarrow + T E'_1$	$E'_1.inh = \text{new Node}('+', E'.inh, T.node)$ $E'.syn = E'_1.syn$
3) $E' \rightarrow - T E'_1$	$E'_1.inh = \text{new Node}('-', E'.inh, T.node)$ $E'.syn = E'_1.syn$
4) $E' \rightarrow \epsilon$	$E'.syn = E'.inh$
5) $T \rightarrow (E)$	$T.node = E.node$
6) $T \rightarrow \text{id}$	$T.node = \text{new Leaf}(\text{id}, \text{id.entry})$
7) $T \rightarrow \text{num}$	$T.node = \text{new Leaf}(\text{num}, \text{num.val})$

语法制导的翻译方案

课件P45之后都没讲

