

# L属性定义 *L-Attributed Definitions*

- ❖ 与S属性定义类似，也是一种SDD
- ❖ 属性的**计算顺序**由分析树中**结点的顺序**决定
  - 应用场景：先有计算顺序，后有语义规则（与依赖图策略相反）
  - 计算方法：采用**深度优先**遍历方式计算

# 分析树的遍历

❖ 利用 *dfvisit* 来计算属性值

```
procedure dfvisit(n:node)
```

```
{
```

```
  for each child  $m_i$  of n from left to right do //遍历子树
```

```
  {
```

```
    计算 $m_i$ 的继承属性值;
```

```
    dfvisit( $m_i$ ) //处理 $m_i$ 子树内部的语义
```

```
  }
```

```
  计算n的综合属性值 //访问根
```

```
}
```

# L属性定义

❖ 满足以下条件的SDD是L属性定义

➤ 对于任意产生式规则  $A \rightarrow X_1 \dots X_n$ ，其附加语义规则满足：

✓  $X_j$  的继承属性只依赖于： $A$ 的继承属性，文法符号  $X_1 \dots X_{j-1}$  的综合属性和继承属性

❖ S属性定义也是L属性定义：*dfvisit*支持S属性定义的语义计算

# 例1:一个不是L属性定义的SDD

产生式规则

$A \rightarrow LM$

$A \rightarrow QR$

语义规则

$L.i = f_1(A.i)$

$M.i = f_2(L.s)$

$A.s = f_3(M.s)$

$R.i = f_4(A.i)$

$Q.i = f_5(R.s)$

$A.s = f_6(Q.s)$

哪些属性是综合属性?

# 例2

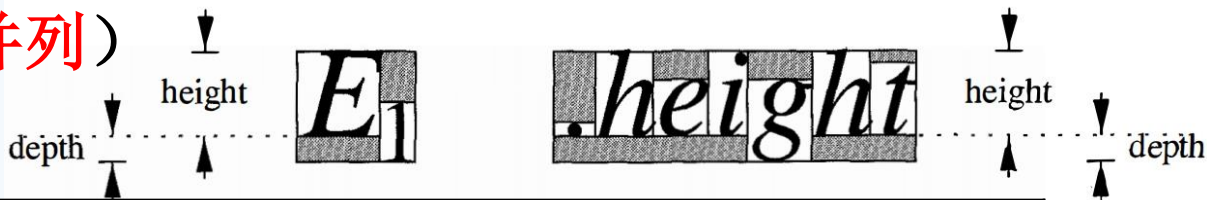
设有产生式 $A \rightarrow BCD$ 。A、B、C、D这4个非终结符号都有两个属性： $s$ 是综合属性， $i$ 是继承属性。对于以下规则：

$$A.s = D.i, \quad B.i = A.s + C.s, \quad C.i = B.s, \quad D.i = B.i + C.i$$

- (1) 是否满足S属性定义的要求
- (2) 是否满足L属性定义的要求
- (3) 是否存在和这些规则一致的求值过程

例：已知方框排版的SDD，求text text sub text的高度和深度（**右结合，sub优先级高于并列**）

P213 图5-25



产生式	语义规则
$S \rightarrow B$	$B.ps = 10$
$B \rightarrow B_1 B_2$	$B_1.ps = B.ps; B_2.ps = B.ps$ $B.ht = \max(B_1.ht, B_2.ht)$ $B.dp = \max(B_1.dp, B_2.dp)$
$B \rightarrow B_1 \text{ sub } B_2$	$B_1.ps = B.ps$ $B_2.ps = 0.7 * B.ps$ $B.ht = \max(B_1.ht, B_2.ht - 0.25 * B.ps)$ $B.dp = \max(B_1.dp, B_2.dp + 0.25 * B.ps)$
$B \rightarrow (B_1)$	$B_1.ps = B.ps; B.ht = B_1.ht; B.dp = B_1.dp$
$B \rightarrow \text{text}$	$B.ht = \text{getHt}(B.ps, \text{text.lexval})$ $B.dp = \text{getDp}(B.ps, \text{text.lexval})$

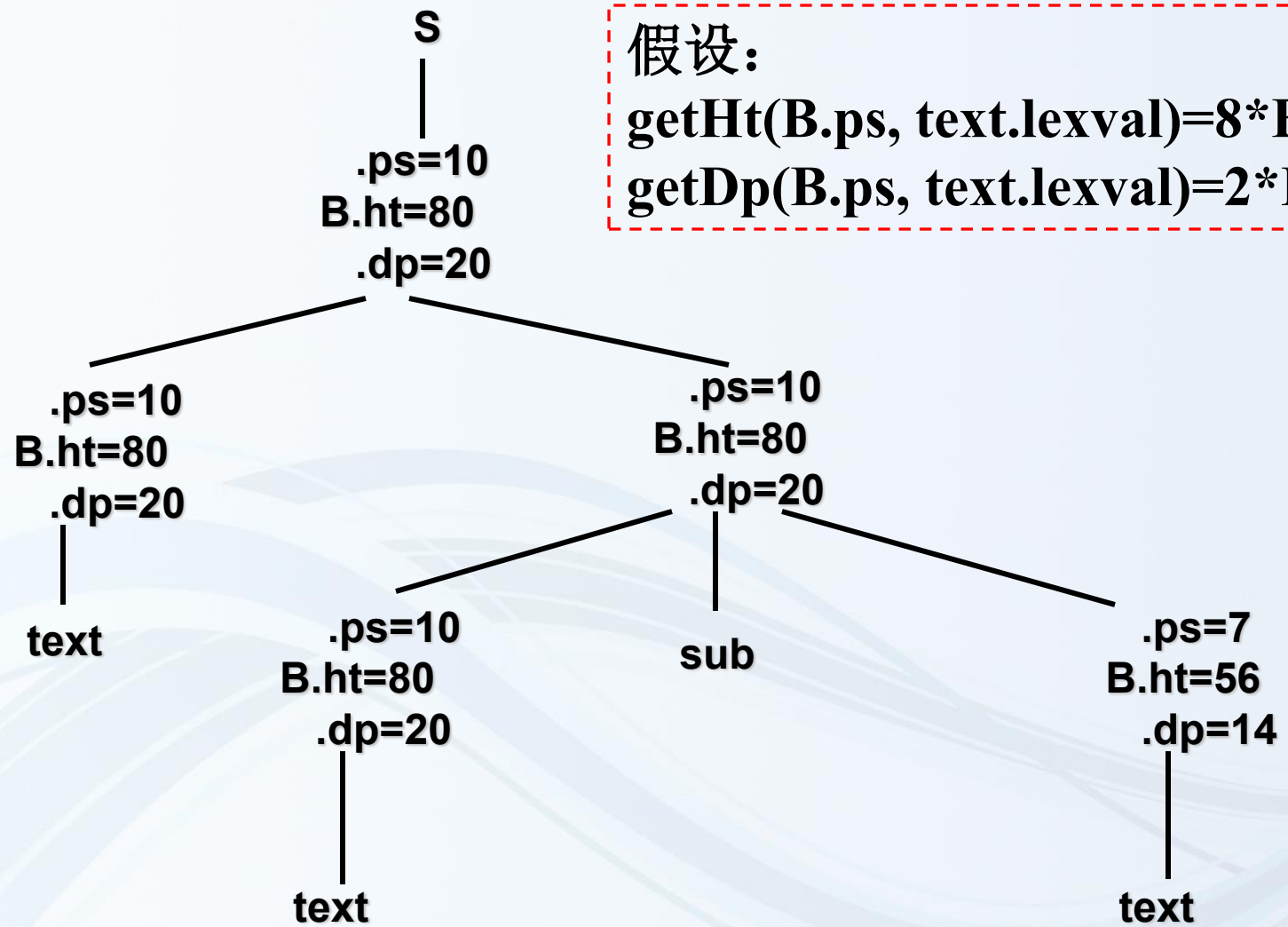
哪些属性是继承属性？

假设：

$\text{getHt}(B.ps, \text{text.lexval}) = 8 * B.ps$

$\text{getDp}(B.ps, \text{text.lexval}) = 2 * B.ps$

假设：  
 $getHt(B.ps, text.lexval) = 8 * B.ps$   
 $getDp(B.ps, text.lexval) = 2 * B.ps$



# 翻译方案 *Syntax-Directed Translation Scheme*

- ❖ 第2种语义描述方法，也译成**翻译模式**，缩写为SDT。
- ❖ 语义动作嵌入文法中（与语法制导定义不同）

例

$\text{expr} \rightarrow \text{expr} + \text{term} \{ \text{print}(\text{"+"}) ; \}$

$\text{expr} \rightarrow \text{expr} - \text{term} \{ \text{print}(\text{"-"}) ; \}$

$\text{expr} \rightarrow \text{term}$

$\text{term} \rightarrow 0 \{ \text{print}(\text{"0"}) ; \}$

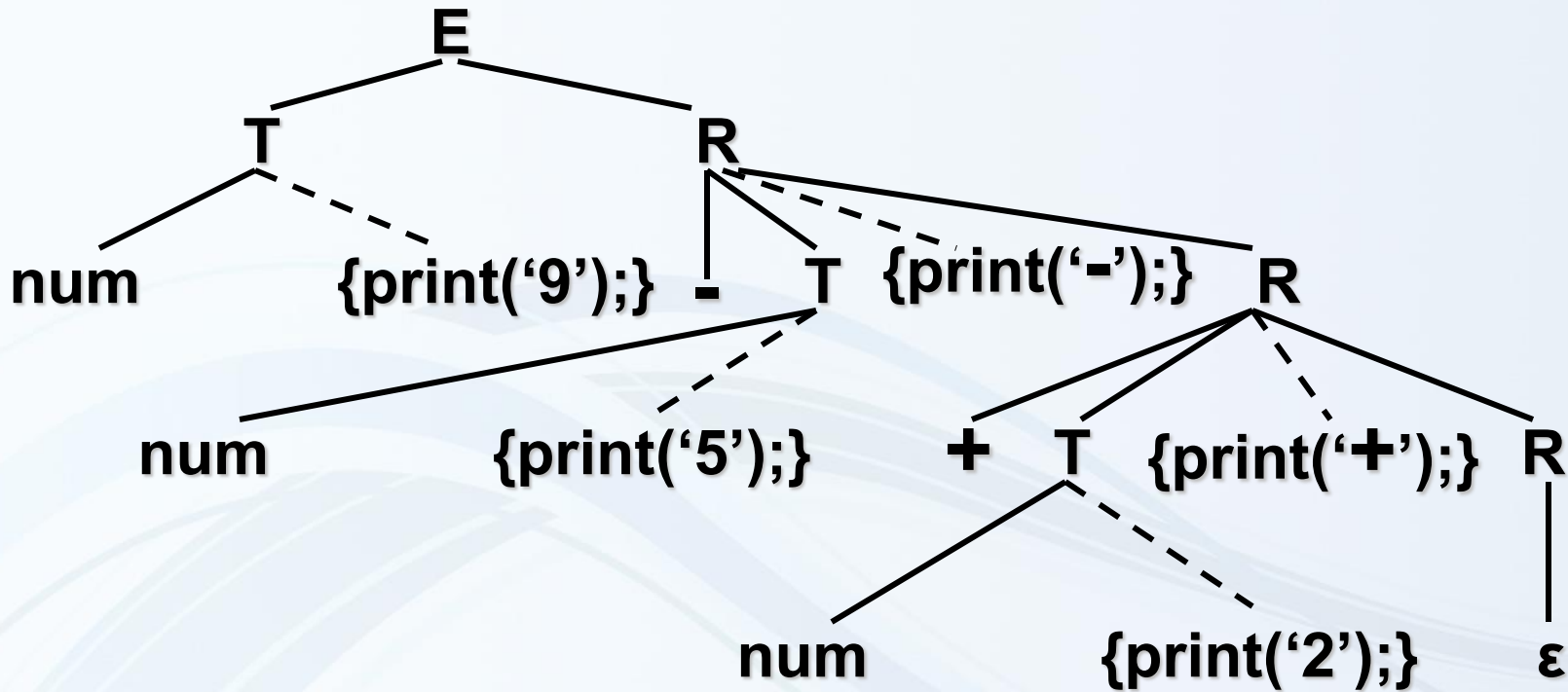
...

$\text{term} \rightarrow 9 \{ \text{print}(\text{"9"}) ; \}$

**3+5时，输出什么？**

**3+5是字符流  
还是记号流？**

# 中綴转后綴 9-5+2



$E \rightarrow TR$

$R \rightarrow +T\{\text{print}('+' );\}R \mid -T\{\text{print}('-' );\}R \mid \varepsilon$

$T \rightarrow \text{num}\{\text{print}(\text{num.val});\}$

# 构建翻译方案

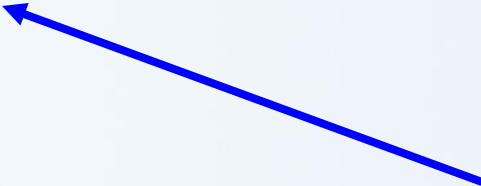
- ❖ 基本原则：**不在属性未赋值时使用它**
- ❖ 综合属性：**建议将语义规则嵌入文法规则的最右部**
- ❖ 继承属性：
  - 必须在该文法符号之前计算（建议**与该文法符号相邻**）
  - 不能使用其右部文法符号的综合属性和继承属性
- ❖ L属性定义满足以上限制

# 例

错误:

$S \rightarrow A_1 \{S.s = A_1.s + A_2.s\} A_2$

$A \rightarrow a \{A.s = 1\}$



综合属性应该在后

正确:

$S \rightarrow A_1 A_2 \{S.s = A_1.s + A_2.s\}$

$A \rightarrow a \{A.s = 1\}$


# 例II

错误:

$S \rightarrow A_1 A_2 \{A_1.in = 1; A_2.in = 2\}$

$A \rightarrow a \{print(A.in)\}$

继承属性应该在前



正确:

$S \rightarrow \{A_1.in = 1; A_2.in = 2\} A_1 A_2$

$A \rightarrow a \{print(A.in)\}$

也正确:

$S \rightarrow \{A_1.in = 1\} A_1 \{A_2.in = 2\} A_2$

$A \rightarrow a \{print(A.in)\}$

# 例III

对于L属性定义：  
继承属性在哪里赋值  
肯定没错？

SDT

$S \rightarrow \{B.ps = 10\} B$

$B \rightarrow \{B_1.ps = B.ps\} B_1 \{B_2.ps = B.ps\} B_2$   
 $\{B.ht = \max(B_1.ht, B_2.ht); B.dp = \max(B_1.dp, B_2.dp)\}$

$B \rightarrow \{B_1.ps = B.ps\} B_1 \text{ sub } \{B_2.ps = 0.7 * B.ps\} B_2$   
 $\{B.ht = \max(B_1.ht, B_2.ht - 0.25 * B.ps)$   
 $B.dp = \max(B_1.dp, B_2.dp + 0.25 * B.ps)\}$

$B \rightarrow (\{B_1.ps = B.ps\} B_1) \{B.ht = B_1.ht; B.dp = B_1.dp\}$

$B \rightarrow \text{text}$

$\{B.ht = \text{getHt}(B.ps, \text{text.lexval})$   
 $B.dp = \text{getDp}(B.ps, \text{text.lexval})\}$

## 5.4.4 翻译方案（消除左递归）

含有综合属性的翻译方案（文法有左递归）

$$A \rightarrow A_1 Y \{A.a = g(A_1.a, Y.y)\}$$

$$A \rightarrow X \{A.a = f(X.x)\}$$

消除左递归后的翻译方案（文法的左递归已经消除）：

$$A \rightarrow X \{R.i = f(X.x)\} R \{A.a = R.s\}$$

$$R \rightarrow Y \{R_1.i = g(R.i, Y.y)\} R_1 \{R.s = R_1.s\}$$

$$R \rightarrow \varepsilon \{R.s = R.i\}$$

# 翻译方案（消除左递归） 例:9-5+2

## 翻译方案

$E \rightarrow E_1 + T \quad \{E.val = E_1.val + T.val\}$

$E \rightarrow E_1 - T \quad \{E.val = E_1.val - T.val\}$

$E \rightarrow T \quad \{E.val = T.val\}$

$T \rightarrow (E) \quad \{T.val = E.val\}$

$T \rightarrow \text{num} \quad \{T.val = \text{num.val}\}$

输入是字符流还是记号流?

## 消除左递归后的翻译方案:

$E \rightarrow T \{R.i = T.val\} R \{E.val = R.s\}$

$R \rightarrow + T \{R_1.i = R.i + T.val\} R_1 \{R.s = R_1.s\}$

$R \rightarrow - T \{R_1.i = R.i - T.val\} R_1 \{R.s = R_1.s\}$

$R \rightarrow \epsilon \quad \{R.s = R.i\}$

$T \rightarrow (E) \{T.val = E.val\}$

$T \rightarrow \text{num} \{T.val = \text{num.lexval}\}$