

编译原理第五章第二次作业
李昊宸 2017K8009929044

5.3.1 下面是涉及运算符 + 和整数或浮点运算分量的表达式的文法。区分浮点数的方法是看它有无小数点。

$E \rightarrow E + T \mid T$

$T \rightarrow \text{num.num} \mid \text{num}$

- 1) 给出一个 SDD 来确定每个项 T 和表达式 E 的类型
- 2) 扩展这个得到的 SDD, 使得它可以把表达式转换成为后缀表达式。使用一个单目运算符 intToFloat 把一个整数转换为相等的浮点数。

答:

1)

PRODUCTION	SEMANTIC RULES
$E \rightarrow E_1 + T$	$E.type = (E_1.type == \text{float}) \parallel (T.type == \text{float}) ? \text{float} : \text{int}$
$E \rightarrow T$	$E.type = T.type$
$T \rightarrow \text{num.num}$	$T.type = \text{float}$
$T \rightarrow \text{num}$	$T.type = \text{int}$

2)

PRODUCTION	SEMANTIC RULES
$S \rightarrow E$	$\text{print}(E.code)$
$E \rightarrow E_1 + T$	<pre> if((E₁.type == float) (T.type == float)) { E.type = float if(E₁.type == int) { E₁.type = float E₁.val = intToFloat(E₁.val) E₁.code = E₁.val } if(T.type == int) { T.type = float T.val = intToFloat(T.val) T.code = T.val } } E.code = E₁.code T.code "+" </pre>
$E \rightarrow T$	$E.type = T.type$ $E.val = T.val$ $E.code = T.code$
$T \rightarrow \text{num.num}$	$T.type = \text{float}$ $T.val = \text{num.num.lexval}$ $T.code = \text{num.num.lexval}$

T -> num	T.type = int T.val = num.lexval T.code = num.lexval
----------	---

5.4.2 改写下面的 SDT:

A -> A {a} B | A B {b} | 0

B -> B {c} A | B A {d} | 1

使得基础文法变成非左递归的。其中 a、b、c 和 d 是语义动作，0 和 1 是终结符。

答:

A -> 0 C

C -> {a} B C | B {b} C | ε

B -> 1 D

D -> {c} A D | A {d} D | ε

5.4.6 修改图 5.25 中的 SDD，使它包含一个综合属性 B.le，即一个方框的长度。两个方框
并列后得到的方框的长度是这两个方框的长度和。然后把你的新规则加入到图 5.26 中
SDT 的合适位置上。

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

图 5-25 方框排版的 SDD

产生式	语义动作
1) $S \rightarrow B$	{ $B.ps = 10;$ }
2) $B \rightarrow B_1 B_2$	{ $B_1.ps = B.ps;$ } { $B_2.ps = B.ps;$ } { $B.ht = \max(B_1.ht, B_2.ht);$ } { $B.dp = \max(B_1.dp, B_2.dp);$ }
3) $B \rightarrow B_1 \text{ sub } B_2$	{ $B_1.ps = B.ps;$ } { $B_2.ps = 0.7 \times B.ps;$ } { $B.ht = \max(B_1.ht, B_2.ht - 0.25 \times B.ps);$ } { $B.dp = \max(B_1.dp, B_2.dp + 0.25 \times B.ps);$ }
4) $B \rightarrow (B_1)$	{ $B_1.ps = B.ps;$ } { $B.ht = B_1.ht;$ } { $B.dp = B_1.dp;$ }
5) $B \rightarrow \text{text}$	{ $B.ht = \text{getHt}(B.ps, \text{text.lexval});$ } { $B.dp = \text{getDp}(B.ps, \text{text.lexval});$ }

图 5-26 方框排版的 SDT

答:

新增加的规则:

PRODUCTION	SEMANTIC RULES
S -> B	
B -> B ₁ B ₂	B.le = B ₁ .le + B ₂ .le
B -> B ₁ sub B ₂	B.le = B ₁ .le + B ₂ .le
B -> (B ₁)	B.le = B ₁ .le
B -> text	B.le = getLe(B.ps , text.lexval)

对于每一个产生式新增加的规则，由于全部都是计算 B 的综合属性 le，故新增加的规则在 SDT 表中添加到每个产生式对应的语义动作的最末端即可（也就是跟在计算 B.dp 的后面）。