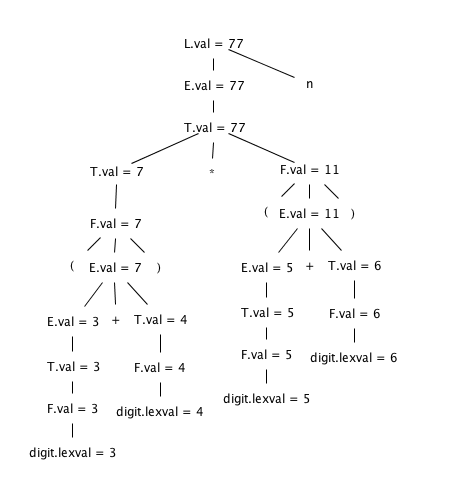
## 第五章作业

### 练习5.1.1



### 练习5.1.2

|  |  |
| --- | --- |
| 产生式 | 语义规则 |
| L -> En | L.val = E.val |
| E -> TE’ | E’.inh = T.val  E.val = E’.syn |
| E’ -> +TE1’ | E1’.inh = E’.inh + T.val  E’.syn = E1’.syn |
| E’ -> ε | E’.syn = E’.inh |
| T -> FT’ | T’.inh = F.val  T.val= T’.syn |
| T’ -> \*FT1’ | T1’.inh = T’.inh\*F.val  T’.syn = T1’.syn |
| T’ ->ε | T’.syn = T’.inh |
| F -> (E) | F.val = E.val |
| F -> digit | F.val = digit.lexval |

### 练习5.2.3

（1）不是；是；存在

（2）不是；是；存在

（3）是；是；存在

（4）不是；不是；不存在

### 练习5.3.1

1)

|  |  |
| --- | --- |
| 产生式 | 语义规则 |
| E🡪E1+T | if (E1.type==integer && T.type==integer){  E.type = integer  }else{  E.type = real  } |
| E🡪T | E.type = T.type |
| T🡪**num.num** | T.type = real |
| T🡪**num** | T.type = integer |

2)

|  |  |
| --- | --- |
| 产生式产生式 | 语义规则 |
| E🡪E1+T | if (E1.type==integer && T.type==integer){  E.type = integer  E.post = E1.post || T.post || 'int+'  }else{  E.type = real  if (E1.type==integer){  E1.type = real  E1.post = E1.post || "inttoreal"  }  if( T.type==integer){  T.type =real  T.post = T.post || "inttoreal"  }  E.post = E1.post || T.post || 'float+"  } |
| E🡪T | E.type = T.type  E.post = T.post |
| T🡪**num.num** | T.type = real  T.post = **num.num** |
| T🡪**num** | T.type = integer  T.post:= **num** |

其中post属性为后缀符号串，'||'符号为连接运算

### 练习5.4.2

A -> 0A’

A' -> {a}BA' | B{b}A' | ε

B -> 1B'

B' -> {c}AB' | A{d}B' | ε

如果a、b、c、d涉及到属性计算的话，变换的结果要更复杂一些。

### 练习5.4.6

SDD与SDT中修改的部分使用粗体表示。

SDD：

S -> B B.ps = 10

B -> B1B2 B1.ps = B.ps

B2.ps = B.ps

**B.le = B1.le + B2.le**

B.ht = max(B1.ht, B2.ht)

B.dp = max(B1.dp, B2.dp)

B -> B1 sub B2 B1.ps = B.ps

B2.ps = 0.7 \* B.ps

**B.le = B1.le + 0.7\*B2.le**

B.ht = max(B1.ht, B2.ht - 0.25 \* B.ps)

B.dp = max(B1.dp, B2.dp + 0.25 \* B.ps)

B -> ( B1 ) B1.ps = B.ps

**B.le = B1.le**

B.ht = B1.ht

B.dp = B1.dp

B -> text **B.le = getLe(B.ps, text.lexval)**

B.ht = getHt(B.ps, text.lexval)

B.dp = getDp(B.ps, text.lexval)

SDT：

S -> { B.ps = 10;}

B

B -> {B1.ps = B.ps;}

B1 {B2.ps = B.ps;}

B2 {**B.le = B1.le + B2.le;**

B.ht = max(B1.ht, B2.ht);

B.dp = max(B1.dp, B2.dp);}

B -> {B1.ps = B.ps;}

B1 sub {B2.ps = 0.7 \* B.ps;}

B2 {**B.le = B1.le + 0.7\*B2.le;**

B.ht = max(B1.ht, B2.ht - 0.25 \* B.ps);

B.dp = max(B1.dp, B2.dp + 0.25 \* B.ps);}

B -> ( {B1.ps = B.ps;}

B1 ) {**B.le = B1.le;**

B.ht = B1.ht

B.dp = B1.dp;}

B -> text {**B.le = getLe(B.ps, text.lexval);**

B.ht = getHt(B.ps, text.lexval);

B.dp = getDp(B.ps, text.lexval);}

### 5.4.4

（1）

S -> if (C) S1 else S2 L1 = new()

L2 = new()

C.true = L1

C.false = L2

S1.next = S.next

S2.next = S.next

S.code = C.code || label || L1 || S1.code || goto S.next || label || L2 || S2.code

（2）

S -> do S1 while(C) L1 = new()

L2 = new()

C.true = L1

C.false = S.next

S1.next = L2

S.code = label || L1 || S1.code || label || L2 || C.code

（3）

S -> ‘{‘ L ‘}’ L.next = S.next

S.code = L.code

L -> L1S M = new()

L1.next = M

S.next = L.next

L.code = L1.code || label || M || S1.code

L -> ε L.code = “”

### 5.5.4

a)



b)



c) 原文法包含左递归，需要先消除左递归。

对L –> LS | ε消除左递归，得到文法：

S -> ‘{’ L ’}’

L ->L’

L’ –> SL’ | ε



### 练习5.5.5

a)



b)



c)原文法包左递归，消除左递归后可得文法：

S -> ‘{’ L ’}’

L ->L’

L’ –> SL’ | ε

