**编译原理第六、七章作业**

2154312 郑博远

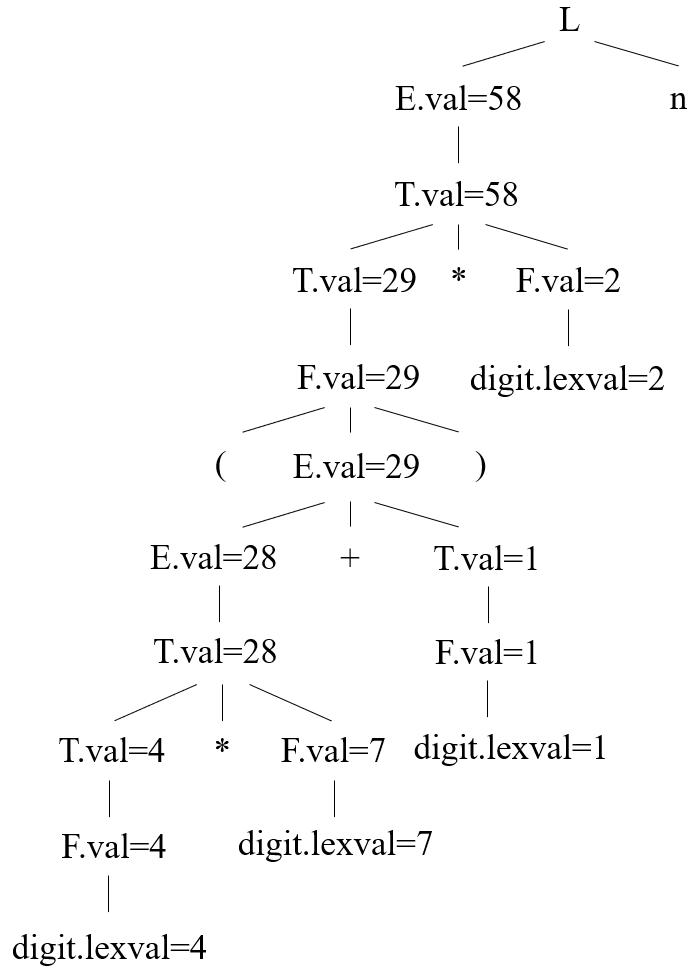
**第六章**

1. **按照表6.1所示的属性文法，构造表达式 (4\*7+1)\*2的附注语法树。**

**表6.1 一个简单台式计算器的属性文法**

|  |  |
| --- | --- |
| **产生式** | **语义规则** |
| L→En  E→E1+T  E→T  T→T1\*F  T→F  F→(E)  F→digit | print(E.val) →  E.val := E1.val + T.val→  E.val := T.val→  T.val := T1.val \* F.val→  T.val := F.val→  F.val := E.val→  F.val:= digit.lexval→ |

**答：**

表达式末尾加上换行符n作为结束：(4\*7+1)\*2n，附注语法树如下：

以最底最左侧内部结点为例，对应产生式F→digit，语义规则F.val:= digit.lexval。由于其子结点的属性digit.lexval=4，所以决定了F的属性F.val也为3。同理，在F结点的父结点处，T.val也算得为3。其他digit（lexval = 7, 1, 2）的属性向上传递计算亦然。

考虑左下方产生式为T→T1\*F的结点。对应语义规则T.val := T1.val \* F.val，其两个子结点的属性值分别为4、7，因此该结点中算得T.val的值为28。同理，产生式为E→E1+T的结点对应语义规则E.val := E1.val \*+T.val，其两个子结点的属性值分别为28、1，因此该结点中算得E.val的值为29。

其他部分的结点属性值计算与传递均同理。最后，包含开始符号L的产生式L→En对应的语义规则打印出通过E得到的表达式的值。

**第七章**

1. **给出下面表达式的逆波兰表示（后缀式）。**

**a\*(-b+c) not A or not (C or not D)**

**a+b\*(c+d/e) (A and B) or (not C or D)**

**答：**

(1) a b uminus c + \*

(2) A not C D not or not or

(3) a b c d e / + \* +

(4) A B and C not D or or

1. **请将表达式 -(a+b) \* (c+d) - (a+b+c) 分别表示成三元式、间接三元式和四元式序列。**

**答：**三元式：

|  |  |  |  |
| --- | --- | --- | --- |
|  | op | arg1 | arg2 |
| (0) | + | a | b |
| (1) | uminus | (0) |  |
| (2) | + | c | d |
| (3) | \* | (1) | (2) |
| (4) | + | a | b |
| (5) | + | (4) | c |
| (6) | - | (3) | (5) |

间接三元式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 间接代码 |  | 三元式表 | | | |
| （1） |  |  | op | arg1 | arg2 |
| (2) |  | (1) | + | a | b |
| (3) |  | (2) | uminus | (1) |  |
| (4) |  | (3) | + | c | d |
| (1) |  | (4) | \* | (2) | (3) |
| (5) |  | (5) | + | (1) | c |
| (6) |  | (6) | - | (4) | (5) |

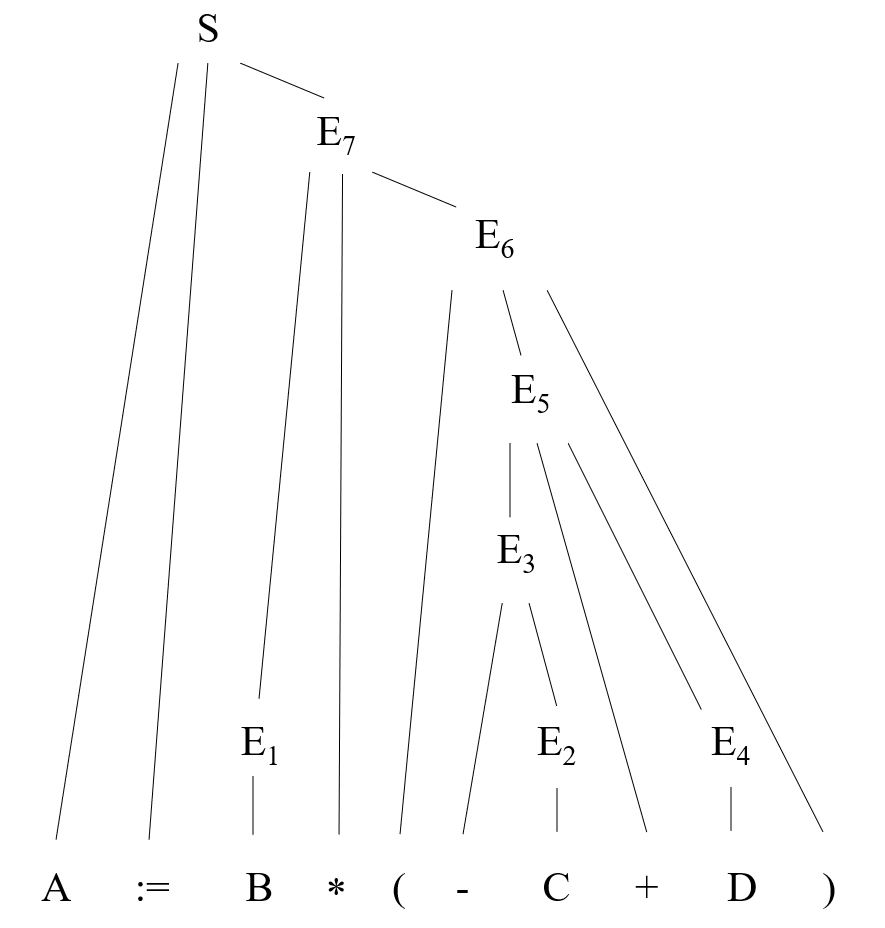
四元式：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | op | arg1 | arg2 | result |
| (0) | + | a | b | T1 |
| (1) | uminus | T1 |  | T2 |
| (2) | + | c | d | T3 |
| (3) | \* | T2 | T3 | T4 |
| (4) | + | a | b | T5 |
| (5) | + | T5 | c | T6 |
| (6) | - | T4 | T6 | T7 |

1. **按7.3节所说的办法，写出下面赋值语句**

**A:=B\*(-C+D)**

**的自下而上语法制导翻译过程。给出所产生的三地址代码。**

**答：**画出语法分析树如下：

根据语法树，进行如下翻译：

|  |
| --- |
| E1.place = B  E1.code = ‘’ |
| E2.place = C  E2.code = ‘’ |
| E3.place = T1  E3.code = ‘T1:=uminus C’ |
| E4.place = D  E4.code = ‘’ |
| E5.place = T2  E5.code = ‘T1:=uminus C  T2:=T1 + D’ |
| E6.place = T2  E6.code = ‘T1:=uminus C  T2:=T1 + D’ |
| E7.place = T3  E7.code =‘T1:=uminus C  T2:=T1 + D  T3:=B\*T2’ |
| S.code = ‘T1:=uminus C  T2:=T1 + D  T3:=B\*T2  A:=T3’ |

因此，三地址代码为：

T1:=uminus C

T2:=T1 + D

T3:=B\*T2

A :=T3