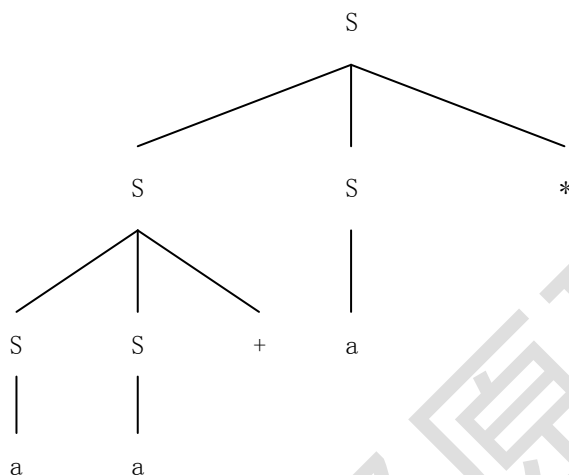


## 第二章作业

### 练习 2.2.1

1)  $S \Rightarrow SS^* \Rightarrow SS+S^* \Rightarrow aS+S^* \Rightarrow aa+S^* \Rightarrow aa+a^*$  (答案以最左推导为例, 不唯一)

2) 语法分析树如下:



3) 由字符  $a$  与运算符  $+$ 、 $*$  构成的后缀表达式。

证明需要包括两方面: 该文法可生成语言的任意串; 该文法只能生成属于该语言的串。

4) 没有二义性。

### 练习 2.2.5

1) 使用数学归纳法, 对该文法最终获得句子进行的推导步数  $n$  进行归纳。

当  $n=1$  时, 仅进行 1 次推导, 仅有

$num \Rightarrow 11$  和  $num \Rightarrow 1001$

两种可能, 两个句子分别表示数值 3 和 9, 均能被 3 整除。

设  $n \leq k(k \geq 1)$  时, 得到句子的数值都能被 3 整除, 对于  $n=k+1$ , 第 1 步推导必然为

$num \Rightarrow num_1 0$  或  $num \Rightarrow num_1 num_2$  (下标仅为区分  $num$  符号的多次出现)

对于  $num \Rightarrow num_1 0$ , 若  $num_1$  经过  $k$  步推导得到符号串  $x$ , 则  $num$  最终推导得到终结符号串  $x0$ , 由归纳假设知  $x$  可被 3 整除; 故  $x0$  必能被 3 整除。

对于  $num \Rightarrow num_1 num_2$ ,  $num_1$  和  $num_2$  分别可经过不超过  $k$  步推导获得终结符号串  $x$ 、 $y$ ; 因此  $num$  经过  $k+1$  步推导获得终结符号串  $xy$ ; 由归纳假设知  $x$ 、 $y$  均可被 3 整除; 故  $xy$  可被 3 整除。

综上, 该文法获得的句子表示的数值都可被 3 整除。

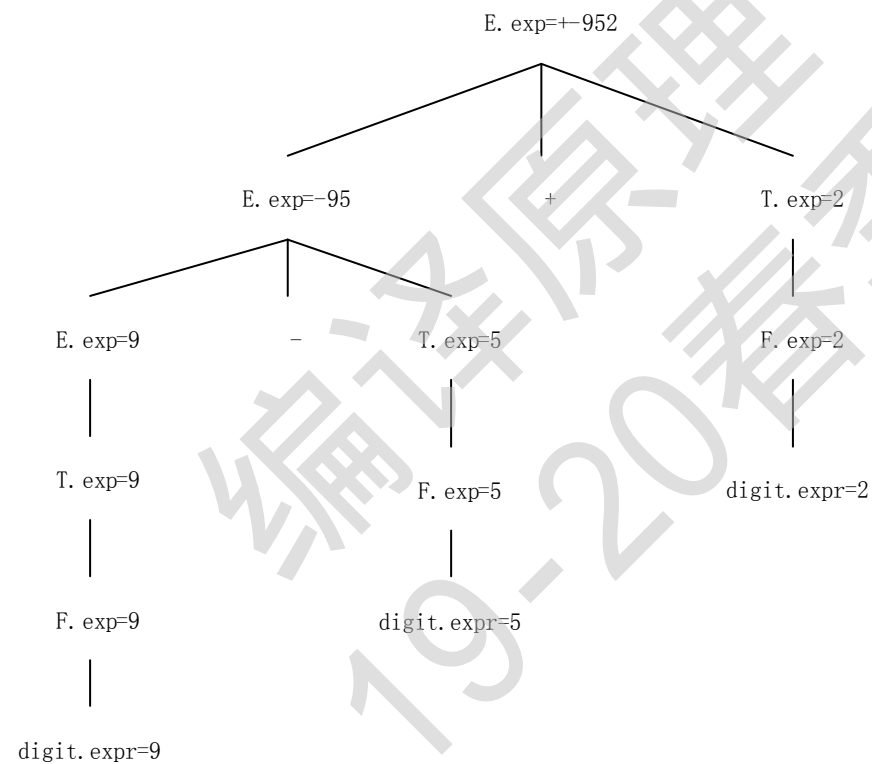
2) 不能。可举例说明, 例如该文法无法表示  $0(0_2)$ 、 $33(100001_2)$  等。

### 练习 2.3.1

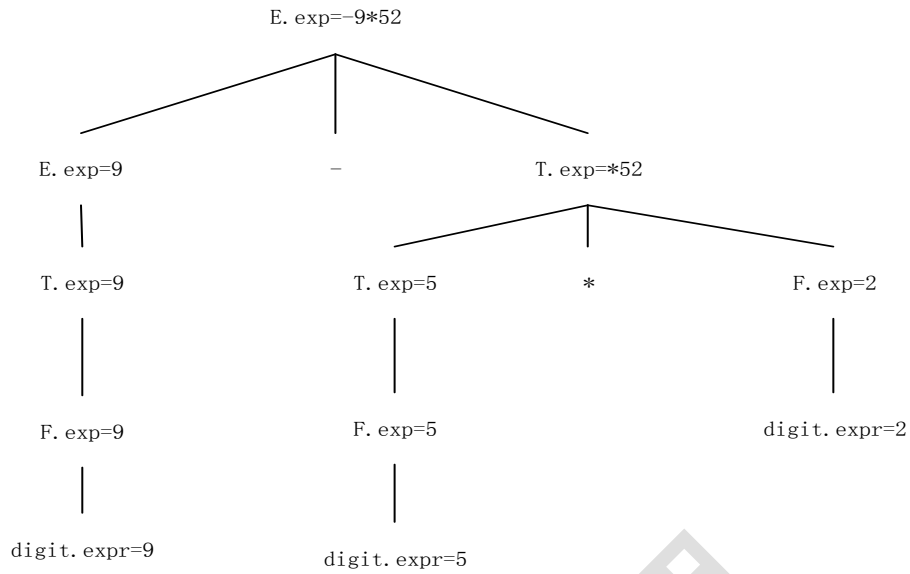
语法制导翻译方案如下：

$E \rightarrow \{ \text{print}('+') \} E_1 + T$   
       $| \{ \text{print}('-') \} E_1 - T$   
       $| T$   
 $T \rightarrow \{ \text{print}('*') \} T_1 * F$   
       $| \{ \text{print}('/') \} T_1 / F$   
       $| F$   
 $F \rightarrow \text{digit} \{ \text{print}(\text{digit}) \}$   
       $| (\text{expr})$

9-5+2 的注释语法分析树如下：



9-5\*2 的注释语法分析树如下：



注释语法分析树的概念请参考龙书第 2 版中文版 P196 页。

### 练习 2.8.1

```

class For extends Stmt{
    Expr E1,E2,E3;
    Stmt S;
    Label start,after;
    public For(Expr x, Expr y, Expr z, Stmt s) {
        E1=x;
        E2=y;
        E3=z;
        S=s;
        start=newlabel();
        after=newlabel();
    }
    public void gen(){
        E1.gen();
        emit(start+":");
        Expr n = E2.rvalue();
        emit("ifFalse "+n.toString()+" goto "+after);
        S.gen();
        E3.gen();
        emit("goto "+start);
        emit(after+":");
    }
}
  
```