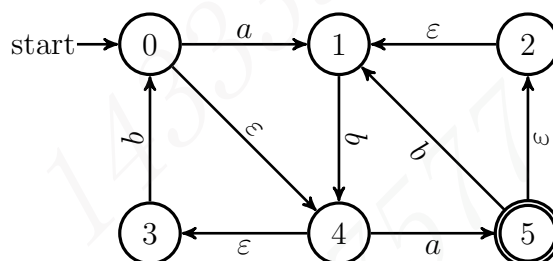


武汉大学计算机学院  
2010-2011学年第一学期2008级  
《编译原理》期末考试试卷(A)

学号: \_\_\_\_\_ 姓名: \_\_\_\_\_ 专业: \_\_\_\_\_ 成绩: \_\_\_\_\_

(注: ①考试时间为120分钟; ②所有的解答必须写在答题纸上, 并注明题号。)

一、设NFA  $N$  的状态转换图如下所示: (20分, 5+5+5+5)



- (1) 试写出NFA  $N$  接受字符串“bbaba”的过程;
- (2) 设用子集构造法求出的与NFA  $N$  等价的DFA  $M$  有三个状态  $A$ 、 $B$  和  $C$ , 其中  $A = \varepsilon\text{-closure}(\{0\})$ ,  $\text{Dtrans}(A, a) = B$ , 试求与状态  $A$ 、 $B$  和  $C$  所对应的NFA  $N$  的状态集, 并画出DFA  $M$  的状态转换图;
- (3) 求DFA  $M$  的最小状态自动机;
- (4) 试用自然语言描述  $N$  所生成的语言。

二、设LL(2)文法  $G(S)$  定义如下: (30分, 每小题5分)

$$\begin{aligned} S &\rightarrow aAaBb \mid bBbAa \\ A &\rightarrow aA \mid \varepsilon \\ B &\rightarrow bB \mid \varepsilon \end{aligned}$$

- (1) 试写出语句“aaabb”的一个最左推导;
- (2) 试对非终结符  $S$ 、 $A$  和  $B$  求First集和Follow集;
- (3) 试画出文法  $G$  的LL(1)分析表, 从而说明  $G$  不是LL(1)文法;
- (4) 已知该文法生成的语言  $L(G)$  是正则语言, 试设计正规表达式  $r$ , 使得  $L(r) = L(G)$ ;
- (5) 试设计一个与文法  $G$  等价的LL(1)文法;
- (6) 试简要说明文法  $G$  不是SLR(1)文法。

三、设有后缀表达式文法  $G(E)$  定义如下: (10分, 5+5)

$$E \rightarrow EE - \mid E - \mid id$$

其中: ‘ $id$ ’ 和 ‘ $-$ ’ 为终结符, “ $-$ ” 为一元减法或二元减法运算符号。

- (1) 试对语句“ $id\ id\ -\ -$ ”画出两颗不同的语法树从而说明该文法为二义文法;
- (2) 试设计一个与文法 $G(E)$ 等价的无二义的文法, 且在该文法中一元减法的运算优先级别低于二元减运算。

四、设后缀表达式的拓广文法 $G(E')$ 定义如下: (20分, 5+5+5+5)

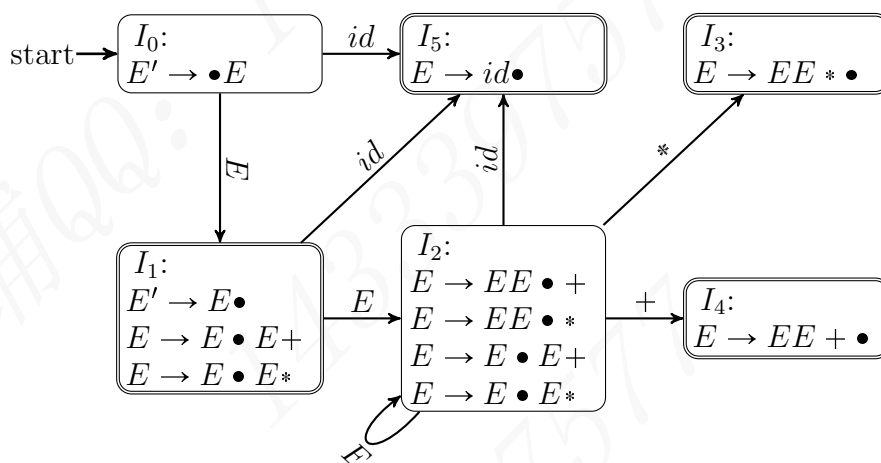
$$E' \rightarrow E \quad (0)$$

$$E \rightarrow EE + \quad (1)$$

$$| EE * \quad (2)$$

$$| id \quad (3)$$

文法 $G(E')$ 的识别活前缀LR(0)项目自动机如下图所示(注意每个状态仅列出了核心项目):



- (1) 试问文法符号串“ $EEE + E$ ”和“ $EEEE$ ”是否为文法 $G(E')$ 的活前缀, 如果是, 请写出其对应的有效项目集;
- (2) 试对非终结符 $E$ 求其First集和Follow集;
- (3) 试画出该文法的SLR分析表;
- (4) 利用你的分析表写出输入表达式“ $id\ id\ +\ id\ *$ ”的分析过程。

五、设文法 $G(E)$ 是对题四的文法进行消除左递归变换后所得到的文法如下所示:

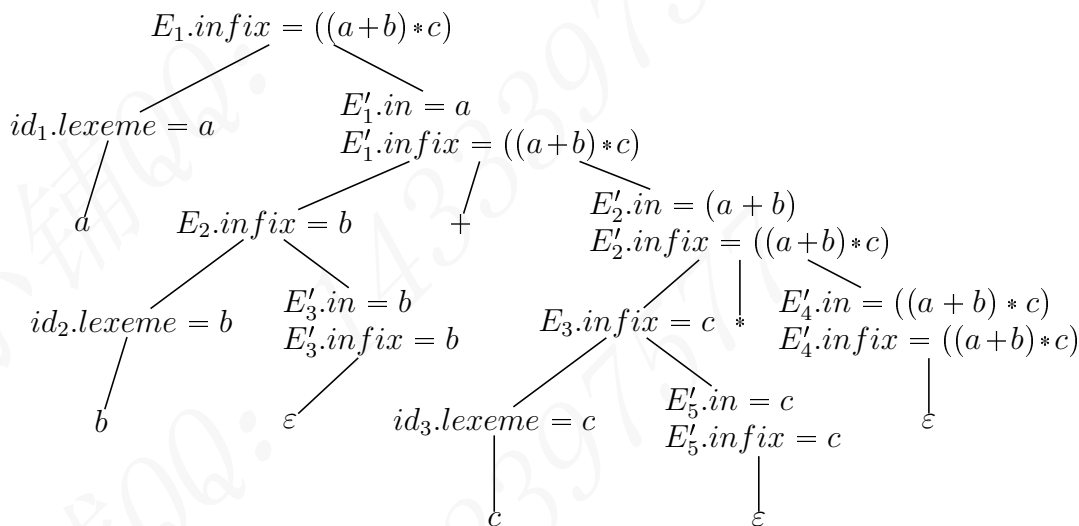
$$E \rightarrow id\ E'$$

$$E' \rightarrow E + E' \mid E * E' \mid \varepsilon$$

现需将文法 $G(E)$ 所生成的后缀表达式翻译为中缀表达式。为此对非终结符 $E$ 定义综合属性 $infix$ , 其取值为 $E$ 所生成的后缀表达式所对应的中缀表达式, 如下图中的结点 $E_1$ ,  $E_2$ 和 $E_3$ ; 对终结符 $id$ 定义综合属性 $lexeme$ , 其取值为该终结符所对应的字符串, 如下图中的结点 $id_1$ ,  $id_2$ 和 $id_3$ ; 在语法树中, 如果 $E'$ 的左兄弟是 $id$ , 其继承属性 $E'.in$ 为其左兄弟的综合属性 $id.lexeme$ , 如图中的 $E'_1$ ,  $E'_3$ 和 $E'_5$ ; 如果 $E'$ 的最左兄弟是 $E$ , 则紧靠 $E'$ 左兄弟一定是运

算符号‘+’或‘\*’，这时 $E'.in$ 应为其父结点的继承属性 $in$ 和其最左兄弟的综合属性 $infix$ 加上对应的运算符所组成的中缀表达式，如图中的 $E'_2$ 和 $E'_4$ ；非终结符 $E'$ 的综合属性 $infix$ 定义为语法树中派生 $E'$ 的第一个祖先 $E$ 所对应的属性 $infix$ ，如图中的 $E'_1$ ， $E'_2$ 和 $E'_4$ 的祖先均为 $E_1$ ，所以其 $infix$ 属性均为“ $((a+b)*c)$ ”。

(10分, 5+5)



图：“ $ab + c*$ ”的附注语法树

- (1) 试为 $G(E)$ 设计后缀表达式到中缀表达式翻译的语法制导定义（注：字符串连接运算用“+”表示即可， $id.lexeme$ 的计算在词法分析时完成，在语法规则中是已知量）；
- (2) 画出语句“ $abc + *$ ”的附注语法树。

六、设有如下Pascal程序片段：

(5分)

```
while (a > b and c > d) do
begin
  if not (e > f and not (i > j)) then
    x := x + 1;
  else
    break;
  if (m > n) then continue;
  y := y + 2;
end;
```

其对应的三地址码如下所示(见下页)：

17: if (a > b) goto l_	goto l_
goto l_	13: goto l_
10: if (c > d) goto l_	15: if (m > n) goto l_
goto l_	goto l_
18: if (e > f) goto l_	14: goto l_
goto l_	16: t1 := y + 2
11: if (i > j) goto l_	y := t1
goto l_	goto l_
12: t0 := x + 1	19:
x := t0	

试为其中的goto填上正确的标号编号。

七、设有如下C语言程序片段：

(5分)

```
void foo()
{
    int i;
    int a[] = {10, 20};
    for (i = 0; i <= 2; i++) a[i] -= 3;
    .....
    return;
}
```

测试发现在for语句处出现了死循环，试分析为什么会出现死循环。