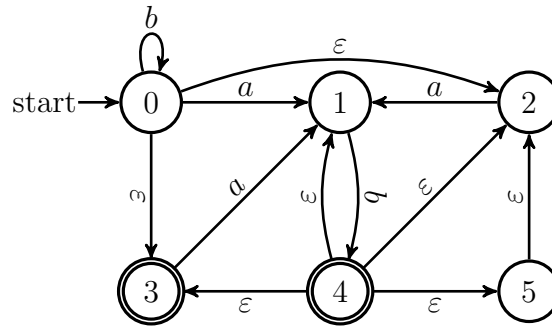


武汉大学计算机学院
2012-2013学年第一学期2010级
《编译原理》期末考试试卷(A)

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: (25分, 每小题5分)



- (1) 试写出NFA N 接受字符串“babab”的过程;
- (2) 设用子集构造法求出的与NFA N 等价的DFA M 有3个状态 A , B 和 C , 其中 $A = \varepsilon\text{-closure}(\{0\})$, $Dtrans(A, a) = B$, 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;
- (5) 求正规表达式 r , 使得 $L(r) = L(N)$ 。

二、设C语言的部分表达式文法 $G(E)$ 定义如下: (25分, 每小题5分)

$$E \rightarrow *E \mid E[E] \mid id$$

其中: ‘id’, ‘*’, ‘[’和‘]’为终结符, ‘*’引用运算符, ‘[]’下标运算符。

- (1) 试写出语句“id[*id]”的一个最左推导;
- (2) 试消除文法 $G(E)$ 中的左递归;
- (3) 试对消除左递归后的文法所有非终结符求First集和Follow集;
- (4) 试对消除左递归后的文法构造LL(1)分析表, 从而说明 $G(E)$ 不是LL(1)文法;
- (5) 试写出语句“id[*id]”的一个正确的分析过程。

三、设C语言的部分表达式文法 $G(E)$ 定义如下 (10分, 5+5)

$$E \rightarrow E + E \mid *E \mid E[E] \mid id$$

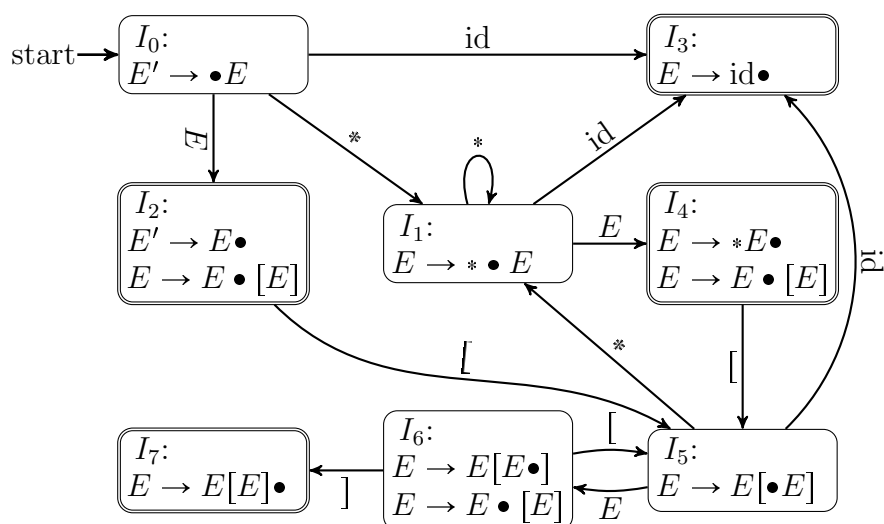
其中: ‘id’, ‘*’, ‘[’, ‘]’和‘+’为终结符, ‘*’引用运算符, ‘[]’下标运算符, ‘+’为二元加运算符。

- (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，使得下标运算‘[]’优先级高于引用运算‘*’；引用运算的优先级高于二元加运算‘+’。且二元加为左结合运算。

四、 设C语言的部分表达式拓广文法 $G(E')$ 定义如下： (20分, 5+5+5+5)

$$\begin{aligned}
 E' &\rightarrow E & (0) \\
 E &\rightarrow *E & (1) \\
 &| E[E] & (2) \\
 &| id & (3)
 \end{aligned}$$

文法 $G(T')$ 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“ $E[**]$ ”和“ $*E[*E]$ ”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) 试构造该文法的SLR分析表，并使得运算的优先级和结合次序与题三所规定的一致；
- (4) 利用你的分析表写出表达式“*id[id]”的分析过程。

五、 设C语言的部分表达式文法 $G(E)$ 定义如下： (10分, 5+5)

$$E \rightarrow E + E \mid *E \mid id$$

现需将文法 $G(E)$ 所生成的表达式翻译为三地址码。在翻译过程中需要符号表与类型的支持。设支持指针和数组的类型表达式递归定义如下：

- (1) (int, 4)和(char, 1)是类型表达式，其中第二分量是该类型变量所占内存的字节数。

- (2) 如果 T 是类型表达式, 则 $(ptr(T), 4)$ 和 $(array(T), n)$ 是类型表达式, 其中 n 是数组所占内存的字节数。如说明语句“`int a[3][5]`”对应的类型表达式为 $(array(array(int, 4), 20), 60)$ 。而说明语句“`char *b[3]`”对应的类型表达式为 $(array(ptr(char, 1), 4), 12)$ 。

为了正确地完成翻译, 特设定如下属性与辅助函数: id 有属性 $lexeme$, 为 id 所对应的词形(字符串)。 E 具有 $temp$ 属性, 其取值为字符串, 它记录保存表达式 E 计算结果的变量或临时变量名。 E 还具有 $type$ 属性, 其取值为上述定义的类型表达式, 它记录表达式 E 的类型。在翻译过程中, 三地址码通过函数 $emit(char *)$ 直接打印到屏幕上。函数 $error(char *)$ 打印出错信息并终止分析过程。函数 $newtemp()$ 产生一个新的临时变量。访问符号表函数 $getsym(char *)$ 返回变量所对应的类型表达式。函数 $getsize(T)$ 返回类型表达式类型的第二分量。

C语言在处理表达式的相关规则如下:

- (1) `char`类型变量在表达式中转换为`int`类型。
- (2) `array`类型转换为指向该数组第一个元素的指针。
- (3) 表达式 $E_1 + E_2$ 中, 如果其中之一的类型是指针, 则另一个的类型必须是`int`, 否则类型出错。例如, E_1 的类型是 $ptr(T, 4)$, 则 E_2 的类型必须是`int`。其运算结果为 $E_1.temp + E_2.temp * getsize(T)$, 运算结果的类型也是 $ptr(T, 4)$ 。
- (4) 表达式 $E_1[E_2]$ 等同于 $*(E_1 + E_2)$ 。
- (5) $*E$ 要求 E 的类型必须是 $ptr(T, 4)$ 。如果 T 是 $array(T_1, n)$, 运算结果为 E 本身, 其运算结果的类型为 $ptr(T_1, 4)$ 。否则, 运算结果为 $*E.temp$, 结果类型为 T 。

请完成如下小题:

- (1) 设计上述文法翻译为三地址码的语法制导语义定义;
- (2) 设有如下说明语句
`int *a[3][4][5], x, y, z;`
 试写出变量 a 的类型表达式并翻译表达式“`*a[x][y][z]`”为三地址码。

六、设有如下Pascal程序片段: (5分)

```
repeat
  x := x + 1;
  if a > b or c > d then continue;
  else x := x + 2;
until not(e > f or g > h) or i > k;
```

其对应的三地址码如下所示

L1: t0 := x+1		x := t1
x := t0		L2: [] (e>f) goto L__
[] (a>b) goto L__		[] (g>h) goto L__
[] (c>d) goto L__		L0: [] (i>k) goto L__
t1 := x+2		L3:

试为其中空白“__”填上正确的标号编号，并为空白“[]”填上if或ifnot.

七、设有如下两个C语言程序：

(5分)

```
Program 1:
int a[2][5] = {0, 1, 2, 3, 4,
               5, 6, 7, 8, 9 };
int foo(int (*b)[5])
{
    return b[1][3];
}

int main()
{
    printf ("%d\n", foo(a));
    return 0;
}
-----
Program 2:
int a[2][5] = {0, 1, 2, 3, 4,
               5, 6, 7, 8, 9 };
int foo(int **b)
{
    return b[1][3];
}

int main()
{
    printf ("%d\n", foo(a));
    return 0;
}
```

编译运行Program 1输出8后正常退出，而Program 2报段错误后异常退出，试解释其原因。