

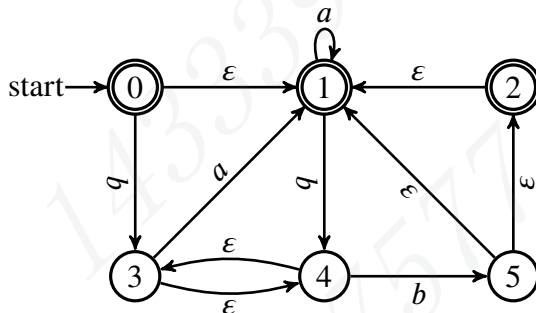
武汉大学计算机学院  
2021 - 2022 学年第 2 学期  
《编译原理》期末考试试卷 (A)

[闭卷, 试卷共 4 页, 答题时间 120 分钟]

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

注意: (1) 请将答案一律写在答卷纸上, 注意标明题号并按顺序解答。答在试卷上不得分。  
(2) 离开考场时将试卷及答卷纸一起上交。

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 “ababa” 的一个过程;
- (2) 设用子集构造法求出的与 NFA  $N$  等价的 DFA  $M$  有 4 个状态  $A, B, C$  和  $D$ , 其中  $A = \varepsilon\text{-closure}(\{0\})$ ,  $Dtrans(A, a) = B$ ,  $Dtrans(A, b) = C$  试求与状态  $A, B, C$  和  $D$  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 NFA  $N$  所接受的语言;
- (5) 求正规表达式  $r$ , 使得  $L(r) = L(N)$ .

二、 设含有解引用和函数调用的 C 语言表达式文法  $G(E)$  定义如下:

$$E \rightarrow *E \mid E(E) \mid (E) \mid id$$

其中: ‘\*’, ‘(’, ‘)’, 和 ‘id’ 为终结符,  $E$  是文法开始符号. (25 分, 每小题 5 分)

- (1) 试写出语句 “(\*id)(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, 从而说明消除左递归后的文法不是 LL(1) 文法;
- (5) 试利用你的分析表写出语句 “id(\*id)” 的一个正确的分析过程.

三、 设文法  $G(E)$  如题二所示：

(10 分, 5+5)

- (1) 试对语句 “\*id(id)” 画出两棵不同的语法树，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，使得函数调用运算优先于解引用运算。

四、 设题二文法  $G(E)$  的拓广文法  $G(E')$  如下所示：

(20 分, 5+5+5+5)

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

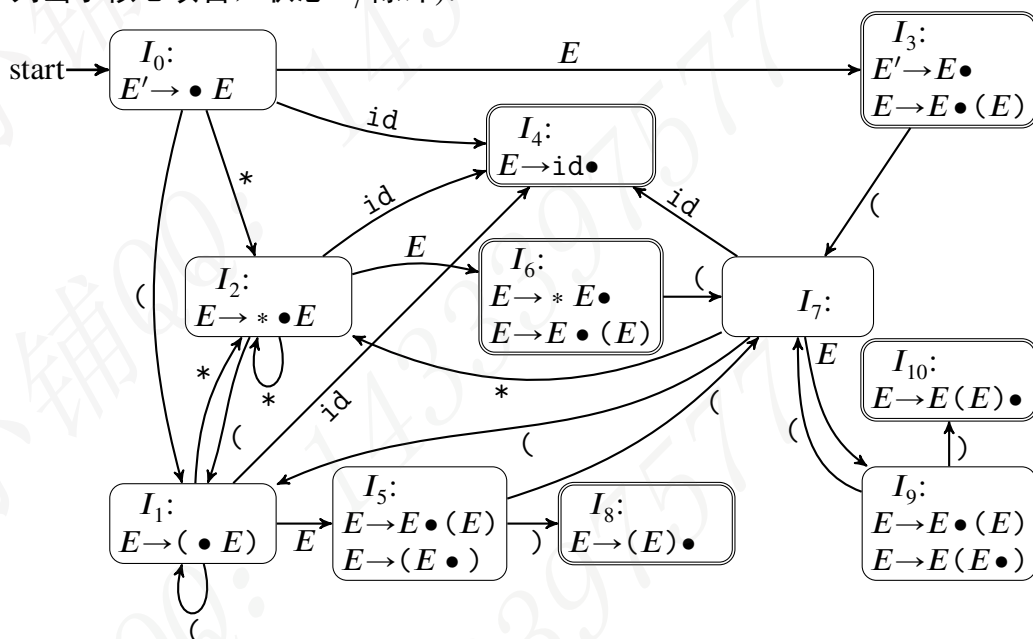
$$E \rightarrow *E \quad (1)$$

$$| E(E) \quad (2)$$

$$| (E) \quad (3)$$

$$| id \quad (4)$$

文法  $G(E')$  的识别活前缀 LR(0) 项目自动机  $M$  如下图所示 (注意每个状态仅列出了核心项目，状态  $I_7$  除外)：



- (1) 试求状态  $I_7$  所对应的 LR(0) 项目集；
- (2) 试求活前缀 “ $((**)^n E$ ” ( $n \geq 1$ ) 至少要经过多少次归约才能把整个前缀归约到  $E$ ，为什么？
- (3) 试构造该文法的 SLR 分析表，并对分析表中的移进/归约和归约/归约冲突选择正确的移进或归约动作，使得文法  $G(E)$  的所有语句能被正确地分析且运算的优先级与结合次序与题三所规定的一致；
- (4) 试利用你的分析表写出 “\*id(id)” 的分析过程。

- 五、现需化简题二文法  $G(E)$  所生成的 C 语言表达式，即尽可能地删除括号对，并使得化简后的表达式按题三规定的运算优先级结合后与原表达式运算次序一致。如： (10 分, 5+5)

序号	C 语言表达式	最简表达式
1	$(*(id(id)))$	$*id(id)$
2	$((**id)((id)))$	$(**id)(id)$

为此设计为非终结符  $E$  设计综合属性  $exp$ ，其值为该非终结符所表示的语法成分对应的最简表达式；综合属性  $E.is\_star$  取值为布尔量  $\{true, false\}$ 。当  $E$  表示的表达式最后一次运算为解引用时，其取值为  $true$ ，否则为  $false$ 。  $id.lexeme$  取值为  $id$  对应的字符串。

- (1) 试设计将文法  $G(E)$  生成的 C 语言表达式转换成最简表达式的 SDD；
- (2) 试写出表达式 “ $((**((*(a)(b))(c)(d)(e)))(h))$ ” 的最简表达式。

- 六、设有如下 Pascal 程序片段： (5 分)

```
while a > b and not (c > d) do
begin
    x := x + 1;
    if e > f and g > h or not (i > k) then continue;
    else x := y + 2;
end;
```

其对应的三地址码如下所示

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

试为其中空白 “\_\_” 填上正确的标号编号，并为空白 “[ ]” 填上  $if$  或  $ifnot$ 。

第七题见下页！

七、 设有如下 C 语言程序:

(5 分)

```
1 #include <stdio.h>
2
3 void init(char *(*p)[3]) {
4     static char *s[] = {"foo", "bar", "baz"};
5     p = &s;
6 }
7
8 int main()
9 {
10     char *(*p)[3];
11     init(p);
12     for (int i=0; i<3; i++) printf("%s", (*p)[i]);
13     return 0;
14 }
```

该程序在 Intel x86/Linux 下用 gcc 编译没有任何警告, 但运行编译后的程序报错如下:

Segmentation fault (core dumped)

试分析产生运行错误的原因. 若将代码第 11 行修改为 `init(&p)`, 试问如何修改函数 `init()`, 使得程序能输出 `foobarbaz` 并正常退出.