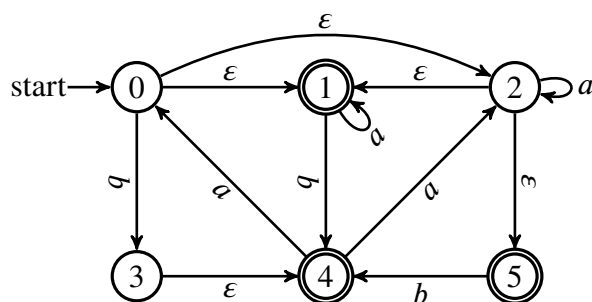


武汉大学计算机学院  
2019 - 2020 学年第一学期 2018 级  
《编译原理》期末考试试卷 (A)

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 “baaba” 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  $A = \epsilon\text{-closure}(\{0\})$ ,  $Dtrans(A, a) = B$ ,  $Dtrans(A, b) = C$  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、 设含有否定  $\neg$  和蕴含  $\rightarrow$  运算的命题公式文法  $G(F)$  定义如下:

$$F \rightarrow F \rightarrow F \mid \neg F \mid (F) \mid a$$

其中: ‘a’, ‘ $\rightarrow$ ’, ‘ $\neg$ ’, ‘(’ 和 ‘)’ 为终结符,  $F$  是文法开始符号. (25 分, 每小题 5 分)

- (1) 试写出语句 “ $\neg(a \rightarrow a)$ ” 的一个最左推导;
- (2) 试消除文法  $G(F)$  中的左递归和左公因子;
- (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 “ $\neg(a \rightarrow a)$ ” 的一个正确的分析过程.



五、现需对题四文法  $G(F')$  所生成的命题公式转换为析取 ( $\vee$ )、合取 ( $\wedge$ ) 和仅有对原子取否的逻辑等价公式，如：

| 序号 | 原命题公式                             | 转换后的命题公式                    |
|----|-----------------------------------|-----------------------------|
| 1  | $A \rightarrow B$                 | $\neg A \vee B$             |
| 2  | $\neg \neg A$                     | $A$                         |
| 3  | $\neg(A \rightarrow B)$           | $(A) \wedge (\neg B)$       |
| 4  | $A \rightarrow (B \rightarrow C)$ | $\neg A \vee \neg B \vee C$ |
| 5  | $(A \rightarrow B) \rightarrow C$ | $(A) \wedge (B) \vee C$     |

为此设计继承属性  $F.is\_neg$ , 其取值为布尔量 True 和 False; 综合属性  $F.nnf$ , 其取值为  $F$  所表示的语法成分对应的转换后的命题公式 (字符串);  $a.lexeme$  取值为  $a$  所对应的字符串.  $F.is\_neg$  的语义规则如下所示: (10 分, 5+5)

| 产生式                                 | 语义规则                                                    |
|-------------------------------------|---------------------------------------------------------|
| $F' \rightarrow F$                  | $F.is\_neg = \text{False}$                              |
| $F \rightarrow F_1 \rightarrow F_2$ | $F_1.is\_neg = \neg F.is\_neg; F_2.is\_neg = F.is\_neg$ |
| $F \rightarrow \neg F_1$            | $F_1.is\_neg = \neg F.is\_neg$                          |
| $F \rightarrow (F_1)$               | $F_1.is\_neg = F.is\_neg$                               |

- (1) 试写出属性  $F.nnf$  的语法制导定义;
- (2) 试求 “ $\neg(((A \rightarrow \neg B) \rightarrow C) \rightarrow \neg(D \rightarrow E))$ ” 转换后的命题公式.

六、设有如下 Pascal 程序片段: (5 分)

```
repeat
  if g < h or not i < j then continue;
  else x := y + 1;
  z := x + y;
until not (not (a < b) or c < d and e < f);
```

其对应的三地址码如下所示

```
L0: [   ] (g<h) goto L__ |      z := t1
    [   ] (i<j) goto L__ | L1: [   ] (a<b) goto L__
    t0 := y+1             |      [   ] (c<d) goto L__
    x := t0               |      [   ] (e<f) goto L__
    t1 := x+y             | 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 **s)
4  {
5      char a[][8] = {"Stay", "Safe", "From", "Covid"};
6      for (int i = 0; i < 4; i++)
7          s[i] = a[i];
8  }
9
10 int main()
11 {
12     char *s[4];
13     init(s);
14     for (int i = 0; i < 4; i++)
15         printf("%s\n", s[i]);
16     return 0;
17 }
```

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输出乱码:

Hã}HÉJ IH ...

但若把第 5 行修改为:

```
char *a[] = {"Stay", "Safe", "From", "Covid"};
```

则程序可正确地输出:

StaySafeFromCovid

试分析产生运行错误的原因.