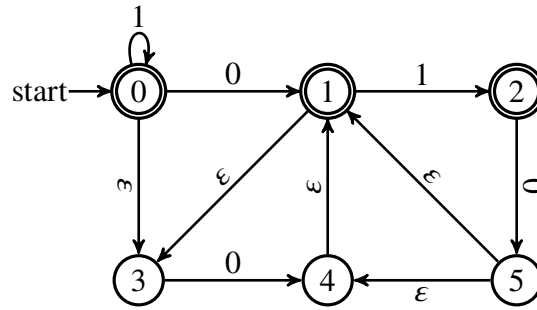


武汉大学计算机学院  
2018 - 2019 学年第一学期 2016 级  
《编译原理》期末考试试卷 (A)

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 “100101” 的过程;
- (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  $A = \varepsilon\text{-closure}(\{0\})$ ,  $D_{\text{trans}}(A, 0) = B$ ,  $D_{\text{trans}}(B, 1) = C$ . 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、 嵌套链表文法  $G(S)$  定义如下:

$$\begin{aligned} S &\rightarrow [L] \\ L &\rightarrow L; L \mid S \mid a \end{aligned}$$

其中: ‘ $a$ ’, ‘[’, ‘]’ 和 ‘;’ 为终结符,  $S$  是文法开始符号. (25 分, 每小题 5 分)

- (1) 试写出语句 “[ $a$ ; [ $a$ ]]” 的一个最左推导;
- (2) 试消除文法  $G(S)$  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 “[ $a$ ; [ $a$ ]]” 的一个正确的分析过程.

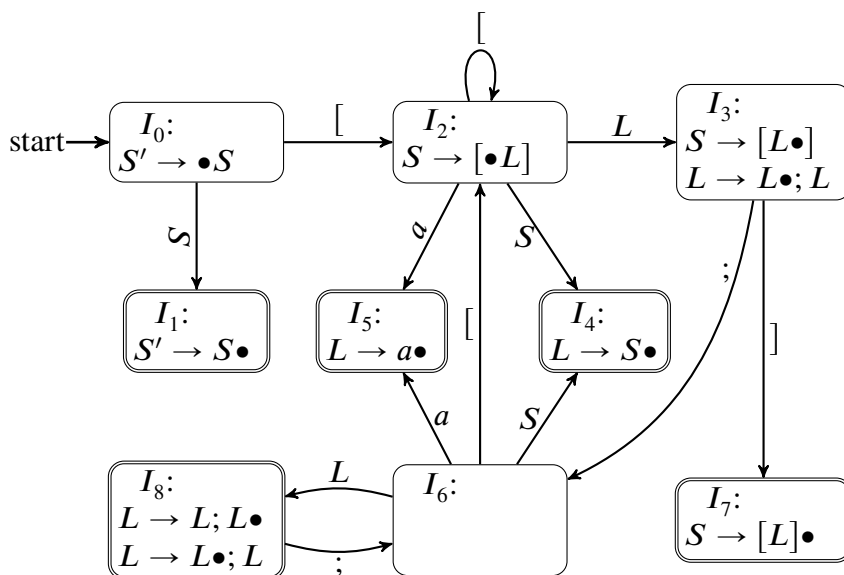
三、 设文法  $G(S)$  如题二所示: (10 分, 5+5)

- (1) 试对语句 “[a; a; a]” 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  $G(S)$  等价的无二义的文法, 使得  $L; L$  为 右结合.

四、 设题二文法  $G(S)$  的拓广文法  $G(S')$  如下所示: (20 分, 5+5+5+5)

$$\begin{aligned}
 S' &\rightarrow S & (0) \\
 S &\rightarrow [L] & (1) \\
 L &\rightarrow L; L & (2) \\
 &| S & (3) \\
 &| a & (4)
 \end{aligned}$$

文法  $G(S')$  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, 状态  $I_6$  除外):



- (1) 试求状态  $I_6$  所对应的 LR(0) 项目集;
- (2) 正则表达式  $[(L;)^*a]$  所生成的文法符号串是活前缀吗? 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  $G(S)$  的所有语句能被正确地分析且运算的结合次序与题三所规定的一致;
- (4) 试利用你的分析表写出语句 “[a; a]” 的分析过程.

五、 设二叉树的线性表示文法  $G(T)$  定义如下: (10 分, 5+5)

$$T \rightarrow \Lambda(T, T) \mid a \mid \perp$$

其中: ‘ $a$ ’, ‘ $\Lambda$ ’, ‘ $\perp$ ’, ‘(’, ‘)’ 和 ‘;’ 为终结符,  $T$  是文法开始符号.  
 现需将题二文法  $G(S)$  按题三的结合次序所生成的嵌套链表转换成二叉树的线性表示. 如

文法 $G(S)$ 语句	文法 $G(T)$ 语句
[a]	$\Lambda(a, \perp)$
[a;b]	$\Lambda(a, \Lambda(b, \perp))$
[[a];b]	$\Lambda(\Lambda(a, \perp), \Lambda(b, \perp))$

为此设计综合属性  $S.tree$  和  $L.tree$ , 其取值为  $S$  或  $L$  所表示的语法成分所对应的二叉树线性表示; 为了判断  $L_1; L_2$  合成  $L$  时,  $L_2$  是否为链尾, 特设计综合属性  $L.is\_list$ , 其取值为 True 或 False.  $L.is\_list = \text{True}$  当且仅当  $L$  是由  $L_1; L_2$  合成;  $a.lexval$  为终结符  $a$  所对应的词素 (词形).

- (1) 试设计将文法  $G(S)$  语句翻译为文法  $G(T)$  语句的 SDD;
- (2) 试求  $[[[a]];b];[c]]$  的二叉树线性表示.

六、 设有如下 Pascal 程序片段: (5 分)

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

其对应的三地址码如下所示

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

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)

```
#include <stdio.h>
void foo(char *p[2])
{
    static char *s[2] = {"happy", "2019!"};
    p = s;
}

int main()
{
    char *p[2];
    foo(p);
    printf("%s %s\n", p[0], p[1]);
    return 0;
}
```

该程序编译正确, 但运行时输出乱码, 并没有输出期望的 “happy 2019!”. 试分析原因.