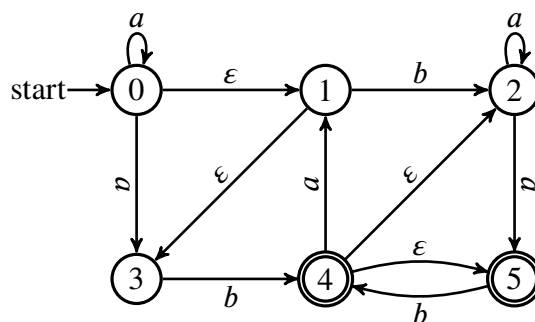


武汉大学计算机学院
2017 - 2018 学年第一学期 2015 级
《编译原理》期末考试试卷 (A)

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 “abaab” 的过程;
- (2) 设用子集构造算法求出的与 NFA N 等价的 DFA M 有 4 个状态 A, B, C 和 D , 其中 $A = \epsilon\text{-closure}(\{0\})$, 状态转换函数 $Dtrans(A, b) = B, Dtrans(B, a) = C$ 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) 求正则表达式 r , 使得 r 所生成的语言 $L(r) = L(N)$.

二、 设文法 $G(T)$ 定义如下: (25 分, 每小题 5 分)

$$\begin{aligned} T &\rightarrow a[L] \mid a \\ L &\rightarrow LL \mid T \end{aligned}$$

其中: ‘ a ’, ‘ $[$ ’, 和 ‘ $]$ ’ 为终结符, T 和 L 是非终结符, T 是文法开始符号.

- (1) 试写出语句 “ $a[aa]$ ” 的一个最左推导;
- (2) 试消除文法 $G(T)$ 中的左递归和左公因子;
- (3) 试对消除左递归和左公因子后的文法所有非终结符求 First 集和 Follow 集;
- (4) 试对消除左递归和左公因子后的文法构造 LL(1) 分析表, 从而说明消除左递归和左公因子后的文法不是 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 $G(T)$ 如题二所示：

(10 分, 5+5)

- (1) 试对语句 “ $a[aaa]$ ” 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 $G(T)$ 等价的无二义的文法，使得链接运算 (LL) 为左结合。

四、 设题二文法 $G(T)$ 的增广 (拓广) 文法 $G(T')$ 如下所示：

(20 分, 5+5+5+5)

$$T' \rightarrow T \quad (0)$$

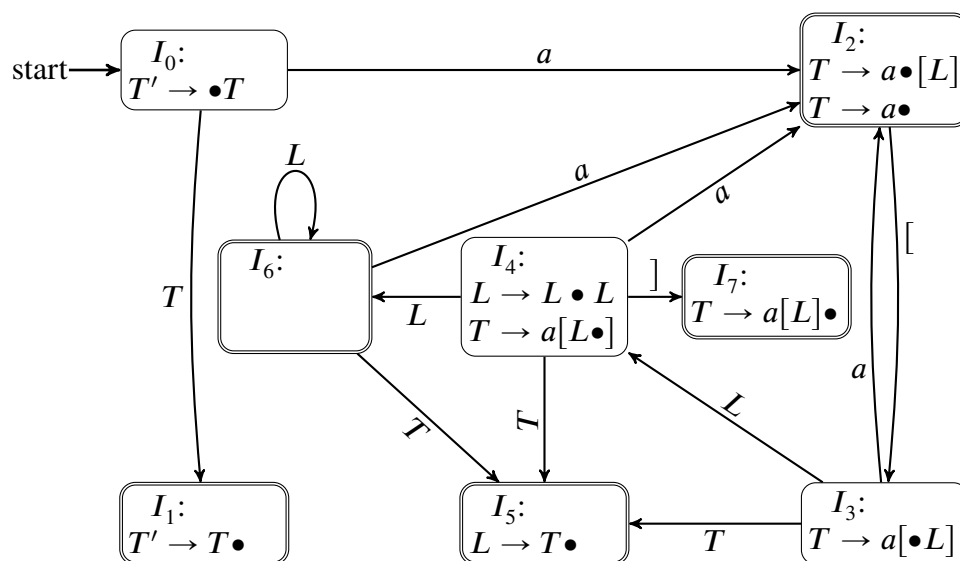
$$T \rightarrow a[L] \quad (1)$$

$$\quad \quad \quad | \quad a \quad (2)$$

$$L \rightarrow LL \quad (3)$$

$$\quad \quad \quad | \quad T \quad (4)$$

文法 $G(T')$ 的识别可行 (活) 前缀 LR(0) 自动机如下图所示 (注意每个状态仅列出了内核项，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 $a([La)^*aa^*$ 所生成的文法符号串是可行 (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 $G(T)$ 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、题二文法 $G(T)$ 所生成语句是树结构的前序遍历序列. 现需将该序列转换为后序遍历序列, 如 “a[b c[d e[f]] g]” 转换为 “[b [d [f]e]c g]a”. 为此设计以下属性: $a.lexeme$ 为终结符 a 所对应的词素(形); $T.postorder$ 和 $L.postorder$ 记录 T 和 L 所表示的语法成分所对应的后序遍历序列. (10 分, 5+5)

(1) 试为上述属性设计语法制导定义 SDD;

(2) 试画出语句 “w[x[y] z]” 的注释(附注)语法树.

六、设有如下 C 程序片段: (5 分)

```
do {
    x = y + z;
    if (a > b || !(c > d)) continue;
    else x = x + 1;
} while (e > f && !(g > h || i > j));
```

其对应的三地址码如下所示

L0: t0 := y + z		x := t1
x := t0		L1: [] (e > f) goto L__
[] (a > b) goto L__		[] (g > h) goto L__
[] (c > d) goto L__		[] (i > j) goto L__
t1 := x + 1		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.

七、设有如下 C 语言程序: (5 分)

```
#include <stdio.h>
void foo(char *s)
{
    char *t = "DEF";
    s = t;
}
int main()
{
    char *s = "ABC";
    printf("%s", s);
    foo(s);
    printf("%s\n", s);
    return 0;
}
```

编译运行后输出如下结果:

ABCABC

试分析, 为什么并没有输出期望的 “ABCDEF” .