

2004 年编译原理试题

1. (20 分) 写出字母表 $\Sigma = \{a, b\}$ 上语言 $L = \{w \mid w \text{ 的最后两个字母是 } aa \text{ 或 } bb\}$ 的正规式, 并画出接受该语言的最简 DFA。

2. (15 分) 说明下面的文法不是 SLR(1)文法, 并重写一个等价的 SLR(1)文法。

$S \rightarrow M a \mid b M c \mid d c \mid b d a$

$M \rightarrow d$

3. (10 分) 为下面的语言写一个无二义的文法:

ML 语言中用分号分隔语句的语句块, 例如:

$((s; s); (s; s; s); s); (s; s)$

4. (20 分) 考虑一个类 Pascal 的语言, 其中所有的变量都是整型 (不需要显式声明), 并且仅包含赋值语句、读语句、写语句, 条件语句和循环语句。下面的产生式定义了该语言的语法 (其中 **lit** 表示整型常量; OP 的产生式没有给出, 因为它和下面讨论的问题无关)。

定义 Stmt 的两个属性: *MayDef* 表示它可能定值的变量集合, *MayUse* 表示它可能引用的变量集合。

(1) 写一个语法制导定义或翻译方案, 它计算 Stmt 的 *MayDef* 和 *MayUse* 属性。

(2) 基于 *MayDef* 和 *MayUse* 属性, 说明 $\text{Stmt}_1; \text{Stmt}_2$ 和 $\text{Stmt}_2; \text{Stmt}_1$ 在什么情况下有同样的语义。

$\text{Program} \rightarrow \text{Stmt}$

$\text{Stmt} \rightarrow \text{id} := \text{Exp}$

$\text{Stmt} \rightarrow \text{read}(\text{id})$

$\text{Stmt} \rightarrow \text{write}(\text{Exp})$

$\text{Stmt} \rightarrow \text{Stmt}; \text{Stmt}$

$\text{Stmt} \rightarrow \text{if}(\text{Exp}) \text{ then begin Stmt end else begin Stmt end}$

$\text{Stmt} \rightarrow \text{while}(\text{Exp}) \text{ do begin Stmt end}$

$\text{Exp} \rightarrow \text{id}$

$\text{Exp} \rightarrow \text{lit}$

$\text{Exp} \rightarrow \text{Exp OP Exp}$

5. (10 分) 下面是一个 C 语言程序:

main()

{

long i;

long a[0][4];

long j;

```

        i = 4; j = 8;
        printf("%d, %d\n", sizeof(a), a[0][0]);
    }

```

虽然出现 `long a[0][4]` 这样的声明，在 X86/Linux 机器上该程序还是能通过编译并生成目标代码。请回答下面两个问题：

- (1) `sizeof(a)` 的值是多少，请说明理由。
- (2) `a[0][0]` 的值是多少，请说明理由。

6. (15 分) 考虑下面的三地址语句序列：

```

    b := 1
    b := 2
    if w <= x goto L2
    e := b
    goto L2
L1: goto L3
L2: c := 3
    b := 4
    c := 6
L3: if y <= z goto L4
    goto L5
L4: g := g + 1
    h := 8
    goto L1
L5: h := 9

```

- (1) 在该代码中用水平的横线将代码分成基本块，并给每个基本块一个序号。
- (2) 画出该代码的控制流图，每个基本块就用 (1) 的序号表示。
- (3) 若有循环的话，列出构成每个循环的结点。

7. (5 分) 如果

- (1) 用编译命令 `cc test.c` 会报告有未定义的符号；
- (2) 用编译命令 `cc test.c -lusr.a` 会得到可执行程序 (`-lusr.a` 表示连接库 `libusr.a`)。

那么，用编译命令 `cc test.c -lusr.a -lusr.a` 是否会报告有多重定义的符号？请说明理由。

8. (5 分) C++ 中的对象声明语句应如何翻译成 C 语句？如书上图 11.11 程序中的

```

    Point _center;

```

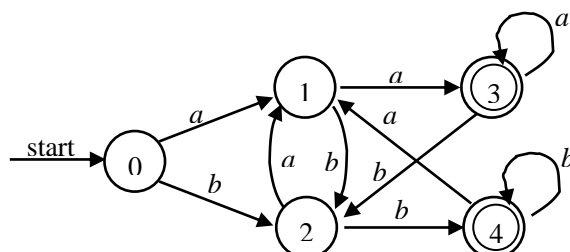
应翻译成什么？

2004 年编译原理试题参考答案

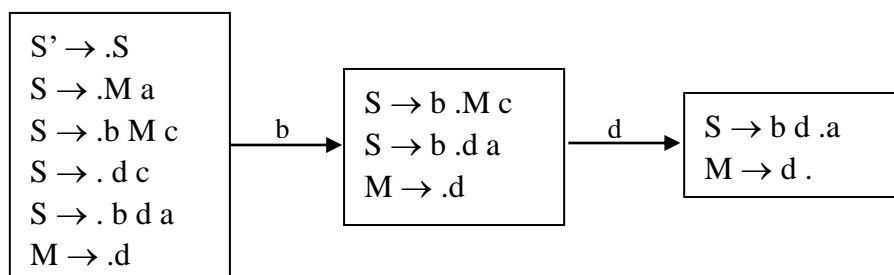
1. 语言 L 的正规式是:

$$(a|b)^*(aa|bb)$$

接受该语言的最简 DFA 是:



2. $S' \rightarrow S$ $S \rightarrow M a | b M c | d c | b d a$ $M \rightarrow d$



因为 a 是 M 的后继符号之一，因此在上面最右边一个项目集中有移进-归约冲突。

等价的 SLR(1)文法是

$$S \rightarrow d a | b d c | d c | b d a$$

3. $SL \rightarrow S | SL ; S$

$$S \rightarrow s | (SL)$$

4. (1) $\text{Program} \rightarrow \text{Stmt}$

$$\text{Stmt} \rightarrow \text{id} := \text{Exp}$$

$$\{ \text{Stmt.MayDef} := \{\text{id.name}\}; \\ \text{Stmt.MayUse} := \text{Exp.MayUse} \}$$

$$\text{Stmt} \rightarrow \text{read}(\text{id})$$

$$\{ \text{Stmt.MayUse} := \phi; \text{Stmt.MayDef} := \{\text{id.name}\} \}$$

$$\text{Stmt} \rightarrow \text{write}(\text{Exp})$$

$$\{ \text{Stmt.MayDef} := \phi; \text{Stmt.MayUse} := \text{Exp.MayUse} \}$$

$$\text{Stmt} \rightarrow \text{Stmt}_1 ; \text{Stmt}_2$$

$$\{ \text{Stmt.MayUse} := \text{Stmt}_1.\text{MayUse} \cup \text{Stmt}_2.\text{MayUse}; \\ \text{Stmt.MayDef} := \text{Stmt}_1.\text{MayDef} \cup \text{Stmt}_2.\text{MayDef} \}$$

$\text{Stmt} \rightarrow \text{if (Exp) then begin Stmt}_1 \text{ end else begin Stmt}_2 \text{ end}$
 $\{ \text{Stmt.MayUse} := \text{Stmt}_1.\text{MayUse} \cup \text{Stmt}_2.\text{MayUse} \cup \text{Exp.MayUse};$
 $\text{Stmt.MayDef} := \text{Stmt}_1.\text{MayDef} \cup \text{Stmt}_2.\text{MayDef} \}$
 $\text{Stmt} \rightarrow \text{while (Exp) do begin Stmt}_1 \text{ end}$
 $\{ \text{Stmt.MayUse} := \text{Stmt}_1.\text{MayUse} \cup \text{Exp.MayUse};$
 $\text{Stmt.MayDef} := \text{Stmt}_1.\text{MayDef} \}$
 $\text{Exp} \rightarrow \text{id} \quad \{ \text{Exp.MayUse} := \{ \text{id.name} \} \}$
 $\text{Exp} \rightarrow \text{lit} \quad \{ \text{Exp.MayUse} := \phi \}$
 $\text{Exp} \rightarrow \text{Exp}_1 \text{ OP Exp}_2$
 $\{ \text{Exp.MayUse} := \text{Exp}_1.\text{MayUse} \cup \text{Exp}_2.\text{MayUse} \}$
 (2) $\text{Stmt}_1.\text{MayDef} \cap \text{Stmt}_2.\text{MayUse} = \phi$ and
 $\text{Stmt}_2.\text{MayDef} \cap \text{Stmt}_1.\text{MayUse} = \phi$ and
 $\text{Stmt}_1.\text{MayDef} \cap \text{Stmt}_2.\text{MayDef} = \phi$

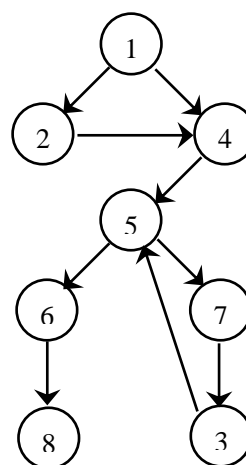
5. (1) 按照数组 size 的计算公式, sizeof(a) 的值一定是 0。

(2) a[0][0] 的值是 4。虽然 a 的 size 是 0, 但它仍然有起始地址, 并且 a[0][0] 的地址等于 a 的起始地址。由于 X86/Linux 机器上, 活动记录栈向低地址方向增长, 另外由于低地址放低位字节, 因此 a[0][0] 的地址和 i 的地址一致, 其值和 i 的值一样, 等于 4。

6. (1)

(2)

$b := 1$	
$b := 2$	
<u>if w <= x goto L2</u>	(1)
$e := b$	
<u>goto L2</u>	(2)
<u>L1: goto L3</u>	(3)
L2: $c := 3$	
$b := 4$	
$c := 6$	(4)
<u>L3: if y <= z goto L4</u>	(5)
<u>goto L5</u>	(6)
L4: $g := g + 1$	
$h := 8$	
<u>goto L1</u>	(7)
L5: $h := 9$	(8)



(3) 结点 5、7 和 3 构成一个循环, 其中 5 是入口结点。

7. 不会。连接时, 第一次遇到库 libusr.a 便能解决所有的外部引用。这样在第二次遇到库 libusr.a 时什么东西也不会加入目标程序。

8. C++ 语言中类的对象声明不加翻译就成了 C 语言中相应结构类型的变量声明, 不管对象声明出现在程序中的什么地方。