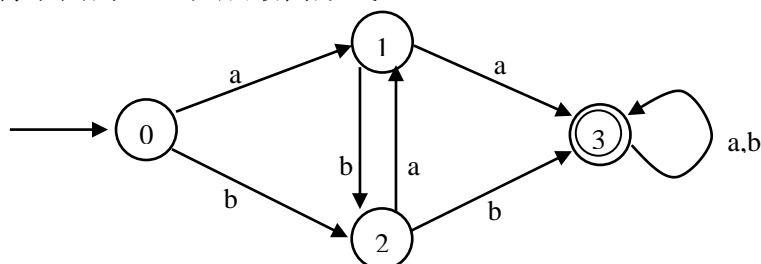


# 1999 年编译原理试题

1. (10 分) 将下面的 DFA 化成最简形式。



2. (5 分) 为什么 LR 分析在访问转移表时决不会进入出错条目。

3. (15 分) 为语言

$L = \{ w \mid w \in (a|b)^* \text{ 并且在 } w \text{ 的任何前缀中, } a \text{ 的个数不少于 } b \text{ 的个数} \}$

写一个 LR (1) 文法, 不准超过 6 个产生式。(若超过 6 个产生式, 不给分。若所写文法不是 LR (1) 文法, 最多给 5 分。)

4. (20 分) 程序的文法如下:

$P \rightarrow D$

$D \rightarrow D ; D \mid \text{id} : T \mid \text{proc id} ; D ; S$

(1) 写一个语法制导定义, 打印该程序一共声明了多少个 **id**。

(2) 写一个翻译方案, 打印该程序每个变量 **id** 的嵌套深度。

5. (10 分) 按 P<sub>127</sub> 图 4.21 的方式, 给出图 4.18 中非终结符 T 的函数。

6. (5 分) 一个 C 语言程序如下:

```

typedef struct _a{
    char    c1;
    long    i;
    char    c2;
    double  f;
}a;
typedef struct _b{
    char    c1;
    char    c2;
    long    i;
    double  f;
}b;
main()
{
    printf("Size of double, long, char = %d, %d, %d\n",
           sizeof(double), sizeof(long), sizeof(char));
    printf("Size of a, b = %d, %d\n", sizeof(a), sizeof(b));
}
    
```

该程序在 SUN 工作站上的运行结果如下:

Size of double, long, char = 8, 4, 1

Size of a, b = 24, 16

结构类型 a 和 b 的域都一样, 仅次序不同, 为什么它们需要的存储空间不一

样。

7. (15 分) 一个 C 语言函数如下:

```
func(i)
long i;
{
    long j;
    j=i-1;
    func(j);
}
```

该函数在 PC 机 linux 操作系统上编译生成的汇编代码如下:

```
.file "stack.c"
gcc2_compiled.:
__gnu_compiled_c:
.text
    .align 2
.globl _func
    .type _func,@function
_func:
    pushl %ebp
    movl %esp,%ebp
    subl $4,%esp
    movl 8(%ebp),%edx
    decl %edx
    movl %edx, -4(%ebp)
    movl -4(%ebp),%eax
    pushl %eax
    call _func
    addl $4,%esp
L1:
    leave
    ret
Lf1:
    .size _func,Lf1-_func
```

试画出该函数的一个活动记录的内容, 包括活动记录的每个单元存放什么东西、执行 `movl 8(%ebp), %edx` 指令时栈顶指针所指的的位置、与活动记录有关的另一个指针所指的位置和地址增长方向。

8. (10 分) 一个 C 语言函数如下:

```
main()
{
    int i, j, k;
    i=5;
    j=1;
    while(j<100) {
        k=i+1;
```

```

        j=j+k;
    }
}

```

经优化编译后，生成的代码如下：

```

        .file "optimize.c"

gcc2_compiled.:
__gnu_compiled_c:
.text
    .align 2
.globl _func
    .type _func,@function
_func:
    pushl %ebp
    movl %esp,%ebp
    movl $1,%eax
    movl $6,%edx
    .align 2,0x90
L4:
    addl %edx,%eax
    cmpl $99,%eax
    jle L4
    leave
    ret
Lfel:
    .size _func,Lfel-_func

```

试说明编译器对这个程序作了哪些种类的优化（只需要说复写传播、删除公共子表达式等，不需要说怎样完成这些优化的）。

9.（5 分）书上 P<sub>279</sub> 的表 10.1 给出了函数调用的翻译。但是这种翻译方式不能用于虚函数的情况。试说明 P<sub>281</sub> 例中的虚函数调用 `obj.scale (zoom_factor)` 应该翻译成什么形式的 C 语句。

10.（5 分）把 SFP 表达式  $(\lambda x.1+x)3$  翻译成 FAM 抽象机指令序列。