

中国科学技术大学

2005—2006 学年第二学期考试试卷

考试科目：编译原理和技术

得分：_____

学生所在系：_____ 姓名：_____ 学号：_____

1、(15 分)

(a) 用正规式表示字母表{a, b}上, a 不会相邻的所有串。

(b) 画出一个最简的确定有限自动机, 它接受所有大于 101 的二进制整数。

2、(10 分) 构造下面文法的 LL(1)分析表。

$S \rightarrow aBS \mid bAS \mid \varepsilon$

$A \rightarrow bAA \mid a$

$B \rightarrow aBB \mid b$

3、(10 分) 下面的文法是二义文法

$S \rightarrow E$

$E \rightarrow \text{while } E \text{ do } E \mid \text{id} := E \mid E + E \mid \text{id} \mid (E)$

请你为该语言重写一个规范的 LR(1)文法, 它为该语言中的各种运算体现通常的优先级和结合规则。不需要证明你的文法是规范 LR(1)的。

4、(10 分) 为下面文法写一个语法制导的定义, 它完成一个句子的 **while-do** 最大嵌套层次的计算并输出这个计算结果。

$S \rightarrow E$

$E \rightarrow \text{while } E \text{ do } E \mid \text{id} := E \mid E + E \mid \text{id} \mid (E)$

5、(15 分) 考虑一个类似 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: *Def* 表示在 **Stmt** 中一定会定值且在该定值前没有引用的变量集合, *MayUse* 表示在 **Stmt** 中有引用且在该引用前可能没有定值的变量集合。

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

(b) 基于上面的计算, 程序可能未赋初值的变量集合从哪儿可以得到?

可能未赋初值的变量是这样定义的: 若存在从程序开始点到达变量 **a** 某引用点的一条路径, 在这条路径上没有对变量 **a** 赋值, 则变量 **a** 属于程序可能未赋初值的变量集合。

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

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

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


```
typedef struct{long i;}S;
```

```
S f(k)
long k;
{
    S s;
    s.i =k;
    return s;
}
```

```
m()
{
    S s;
    s.i = 20;
    s = f(10);
}
```

汇编文件 return.s 的内容如下:

```
.file "return.c"
.text
.globl f
.type f, @function
f:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $4, %esp
    movl     8(%ebp), %eax
    movl     12(%ebp), %edx
    movl     %edx, -4(%ebp)
    movl     -4(%ebp), %edx
    movl     %edx, (%eax)
    leave
    ret
    .size f, .-f
.globl m
.type m, @function
m:
    pushl    %ebp
    movl     %esp, %ebp
    subl     $4, %esp
    movl     $20, -4(%ebp)
    leal     -4(%ebp), %eax
```

```

pushl    $10
pushl    %eax
call     f
addl     $8, %esp
leave
ret
.size m, .-m
.section .note.GNU-stack,"",@progbits
.ident   "GCC: (GNU) 3.3.5 (Debian 1:3.3.5-13)"

```

8、(5 分)把下面左边的文件 file1.c 提交给编译器,编译器没有报告任何错误。而把文件 file2.c 提交给编译器, 错误报告如下:

file2.c: 2: error: conflicting types for ‘func’

file2.c: 1: error: previous declaration of ‘func’

试分析原因。(在这两个文件中, 第 1 行都是函数 func 的原型, 第 2 行都是函数 func 的定义, 函数体为空。)

file1.c

```

int func(double);
int func(f) float f; {}

```

file2.c

```

int func(double);
int func(float f) {}

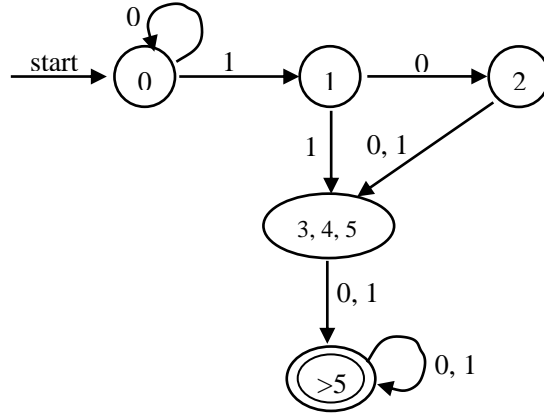
```

9、(5 分)教材上第 342 页倒数第 7 行说“将 C++语言中一个类的所有非静态属性构成一个 C 语言的结构类型, 取类的名字作为结构类型的名字”。在这一章都学过后, 你认为这句话需要修改吗?

2006 年编译原理和技术试题参考答案

1、 (a) $b^*(abb^*)^*(a|\varepsilon)$

(b) 见下图。若不允许前缀的无效 0，则去掉状态 0 上的 0 转换。



2、 开始符号集合和后继符号集合

	First	Follow
S	a, b, ε	\$
A	a, b	a, b, \$
B	a, b	a, b, \$

LL(1)分析表

	a	b	\$
S	$S \rightarrow a B S$	$S \rightarrow b A S$	$S \rightarrow \varepsilon$
A	$A \rightarrow a$	$A \rightarrow b A A$	
B	$B \rightarrow a B B$	$B \rightarrow b$	

3、
 $S \rightarrow E$
 $E \rightarrow \text{while } E \text{ do } E \mid A$
 $A \rightarrow \text{id} := A \mid T$
 $T \rightarrow T + F \mid F$
 $F \rightarrow \text{id} \mid (E)$

4、 语法制导的定义如下：

$S \rightarrow E$	print(S.loop);
$E \rightarrow \text{while } E_1 \text{ do } E_2$	$E.\text{loop} := \max(E_1.\text{loop}, E_2.\text{loop}) + 1;$
$E \rightarrow \text{id} := E$	$E.\text{loop} := E_1.\text{loop};$
$E \rightarrow E_1 + E_2$	$E.\text{loop} := \max(E_1.\text{loop}, E_2.\text{loop});$
$E \rightarrow \text{id}$	$E.\text{loop} := 0;$
$E \rightarrow (E_1)$	$E.\text{loop} := E_1.\text{loop};$

5、 (a) 计算 Stmt 的 Def 和 MayUse 属性的语法制导定义如下：

```

Program → Stmt {Program.MayUse := Stmt.MayUse }
Stmt → id := Exp
      { Stmt.Def := {id.name} - Exp.MayUse ; Stmt.MayUse := Exp.MayUse }
Stmt → read (id )
      { Stmt.MayUse :=  $\phi$  ; Stmt.Def := {id.name} }
Stmt → write ( Exp )
  
```

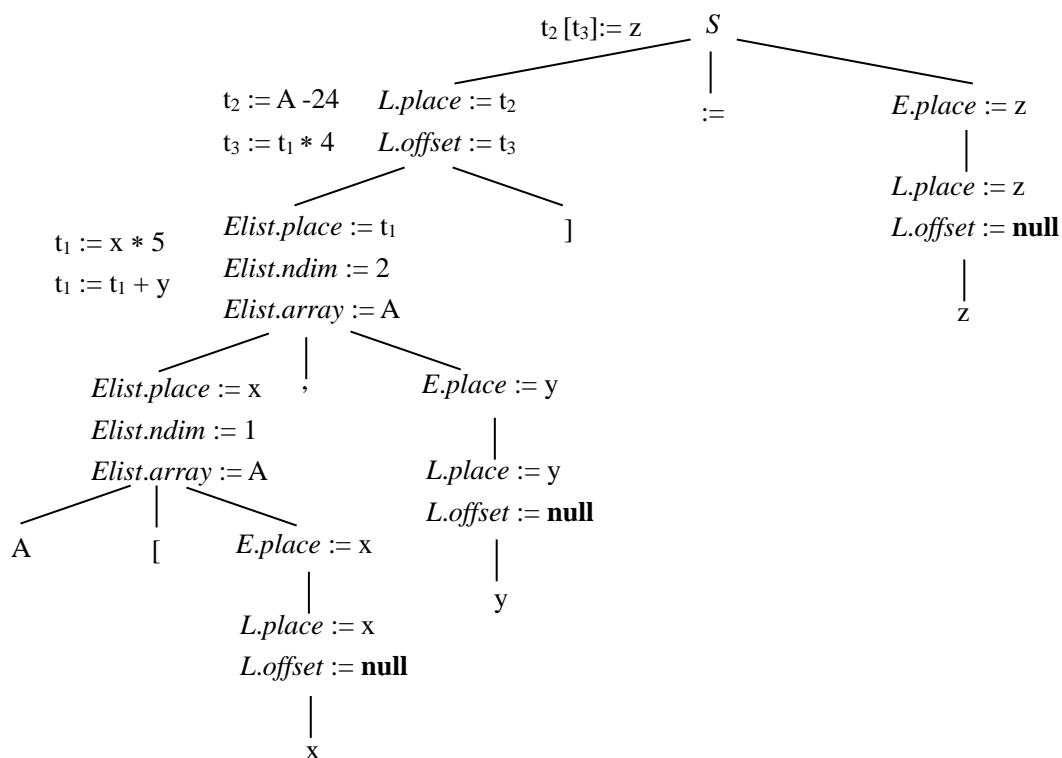
```

{ Stmt.Def :=  $\phi$  ; Stmt.MayUse := Exp.MayUse }
Stmt  $\rightarrow$  Stmt1 ; Stmt2
{   Stmt.MayUse := Stmt1.MayUse  $\cup$  (Stmt2.MayUse - Stmt1.Def) ;
  Stmt.Def := Stmt1.Def  $\cup$  (Stmt2.Def - Stmt1.MayUse) }
Stmt  $\rightarrow$  if ( Exp ) then begin Stmt1 end else begin Stmt2 end
{   Stmt.MayUse := Stmt1.MayUse  $\cup$  Stmt2.MayUse  $\cup$  Exp.MayUse;
  Stmt.Def := (Stmt1.Def  $\cap$  Stmt2.Def) - Exp.MayUse }
Stmt  $\rightarrow$  while ( Exp ) do begin Stmt1 end
{   Stmt.MayUse := Stmt1.MayUse  $\cup$  Exp.MayUse;
  Stmt.Def := Stmt1.Def - Exp.MayUse }
Exp  $\rightarrow$  id { Exp.MayUse := {id.name} }
Exp  $\rightarrow$  lit { Exp.MayUse :=  $\phi$  }
Exp  $\rightarrow$  Exp1 OP Exp2 { Exp.MayUse := Exp1.MayUse  $\cup$  Exp2.MayUse }

```

(b) Program 的属性 MayUse 就是程序可能未赋初值的变量集合。

6、



7、 (a) 调用函数 m 把实在参数压栈后,将存放返回值的地址压栈,然后调用 f.f 在返回前,将结果送到 m 提供的存放返回值的地址。

(b) m 要为 3 次 f 调用分配存放结果值的空间,并且这 3 次存放结果值的空间不重叠。在每次调用 f 时,将相应的存放返回值的地址压栈。

8、文件 file1.c 中,函数 func 的定义采用传统的方式,形式参数 f 的类型被提升到 double。

函数原型中该参数也声明成 `double` 类型，两者类型相同，因此没有错误。

而在文件 `file2.c` 中，函数 `func` 的定义采用现在提倡的形式，形式参数的类型就是 `float`，不会提升。它的类型和函数原型中声明的类型不相同，所以编译器会报告错误。

9、需要修改，增加虚方法表指针作为第一个域。