

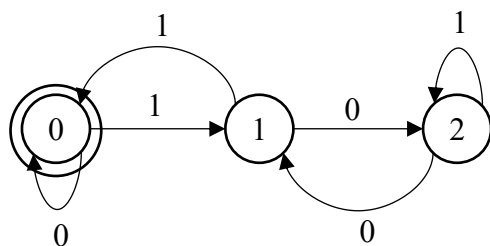
中国科学技术大学
2017--2018 学年第 2 学期考试试卷 A 卷

考试科目: 编译原理与技术 得分: _____
 学生所在系: _____ 姓名: _____ 学号: _____

一、构造一个最简的 DFA, 它接受所有能被 3 整除的二进制非负整数(包含空串)。

1. 画出最简 DFA 结构。
2. 写出对应的上下文无关文法。
3. 写出对应的正规式。(15 分)

1、



2、(5 分)

$S \rightarrow 0S \mid 1A \mid \epsilon$

$A \rightarrow 0B \mid 1S$

$B \rightarrow 0A \mid 1B$

3、(5 分)

$(0 \mid 1(01^*0)^*1)^*$

二、计算下列文法各非终结符的开始符号集合和后继符号集合, 并构造 LL(1)分析表。(15 分)

$S \rightarrow aBS \mid bAS \mid \epsilon$

$A \rightarrow bAA \mid a$

$B \rightarrow aBB \mid b$

$FIRST(S) = \{a, b, \epsilon\}$

$FIRST(A) = \{a, b\}$

$FIRST(B) = \{a, b\}$

$FOLLOW(S) = \{\$ \}$

$FOLLOW(A) = \{a, b, \$ \}$

$FOLLOW(B) = \{a, b, \$ \}$ (每个 1 分)

	A	B	\$
--	---	---	----

S	$S \rightarrow aBS$	$S \rightarrow bAS$	$S \rightarrow \epsilon$
A	$A \rightarrow a$	$A \rightarrow bAA$	
B	$B \rightarrow aBB$	$B \rightarrow b$	

(每空 1 分)

三、由文法 $E \rightarrow E=E \mid E+E \mid (E) \mid \text{num} \mid \text{id}$ 产生的表达式包括赋值表达式，即 $a=5$ 的语义是将 5 赋给 a ，而 $a=(b=5)$ 把 5 赋给 b ，然后再赋给 a 。

- 1、写一个语法制导定义，利用属性“ .val ”完成值的运算和赋值，利用属性“ .var ”检查赋值表达式的左部是否为左值。
- 2、修改该文法，使得赋值表达式的左部必然是左值。要求该文法是 SLR 的，且至多只引入两个新的非终结符。(15 分)

1、(10 分)

```

 $E \rightarrow \text{num}$    $E.\text{val}=\text{num}; E.\text{var}=\text{FALSE};$ 
 $E \rightarrow \text{id}$      $E.\text{val}=\text{id.val}; E.\text{var}=\text{TRUE};$ 
 $E \rightarrow (E_1)$    $E.\text{val}=E_1.\text{val}; E.\text{var}=E_1.\text{var};$ 
 $E \rightarrow E_1+E_2$   $E.\text{val}=E_1.\text{val}+E_2.\text{val}; E.\text{var}=\text{FALSE};$ 
 $E \rightarrow E_1=E_2$   if ( $E_1.\text{var}$ )  $E.\text{val}=E_2.\text{val}; E_1.\text{val}=E_2.\text{val}; E.\text{var}=\text{FALSE};$  else
                    print(“error”);

```

2、(5 分)

```

 $E \rightarrow V=E \mid T+E \mid T$ 
 $T \rightarrow (E) \mid \text{num} \mid V$ 
 $V \rightarrow \text{id}$ 

```

四、关于活动记录（按高地址在下的格式，long 为 8 个字节）

- 1、分别画出 C 语言和 Pascal 语言的活动记录结构。解释两者差异的原因。
- 2、若某 C 语言函数中有程序段 $\{\text{long } i=4; \text{long } a[0][4]; \text{long } j=8;\}$ ，且通过了编译。请画出该函数被调用时，这三个局部数据的内存分配情况（假设为它们分配地址前，栈顶地址为 $***100$ ）。并回答： $\text{sizeof}(a)$ 的值是多少？ $a[0][0]$ 的值是多少？并解释理由。(15 分)

1、(8 分)

C 语言活动记录结构

临时数据
局部数据
机器状态
控制链
返回值

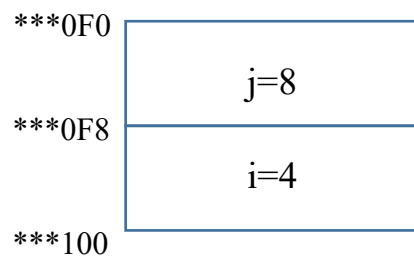
参数

Pascal 语言活动记录结构

临时数据
局部数据
机器状态
访问链
控制链
返回值
参数

C 语言没有访问链，因为 C 语言的函数声明不能嵌套，运行时只能访问全局变量（静态数据区）和局部变量（栈顶活动记录），因此无须使用访问链。

2、（7 分）



sizeof(a)=0。因为 sizeof 的定义是元素个数乘上元素类型长度
a[0][0]=4。因为 a 指向 ***0F8，但未实际分配存储单元

五、为 C 语句 $x=(a-b)+(a-c)$ 产生代码（汇编语言），假定所有变量都是静态的，有 R0、R1 两个寄存器可用于保存中间计算结果。

- 1、写出三地址中间代码（不做优化）。
- 2、写出代价最小的目标代码，并计算总代价。
- 3、对应（2）中产生的每一条目标代码，给出该代码执行后的每个寄存器的寄存器描述，以及该代码执行后发生了变化的名字地址描述。（20 分）

1、（4 分）

t1=a-b

t2=a-c

t3=t1+t2

x=t3

2、（6 分）

MOV a, R0 代价 2

MOV R0, R1 代价 1

SUB b, R0 代价 2

SUB c, R1 代价 2

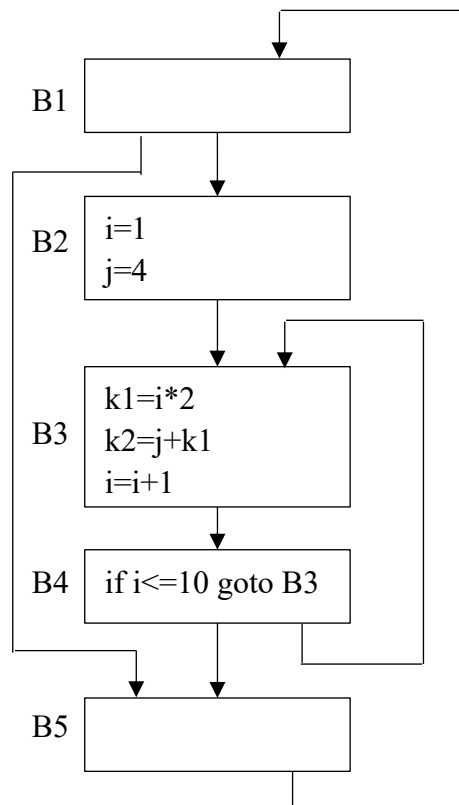
ADD R0, R1 代价 1
 MOV x, R1 代价 2
 总代价 10
 3、(10 分)

寄存器描述
 MOV a, R0 R0 中含 a, R1 未用
 MOV R0, R1
 SUB b, R0 R0 中含 t1, R1 中含 a
 SUB c, R1 R0 中含 t1, R1 中含 t2
 ADD R0, R1 R0 中含 t1, R1 中含 t3
 MOV x, R1 R0 中含 t1, R1 中含 t3

名字地址描述
 a 在 a 和 R0 中
 R0, R1 中含 aa 在 a, R0, R1 中
 a 在 a, R1 中, t1 在 R0 中
 a 在 a 中, t2 在 R1 中
 t3 在 R1 中
 x 在 x 和 R1 中

六、考虑如下程序流图，进行如下针对“循环”的分析：

- 1、设计一个数据流问题来计算每个节点的支配节点集合，包括相应的迁移函数、合流方程、边界条件、各节点的初始化条件。
- 2、根据定义直接写出程序块 B1、B2、B3、B4、B5 的支配节点集合。
- 3、列出流图中的所有回边，并写出对应的自然循环。
- 4、流图中给出了部分程序代码，优化它们（k2 在 B3、B5 块均有引用，其它变量没有其它使用）。(20 分)



题六图

1、(5 分)

迁移函数: $OUT[B] = IN[B] \cup B$

合流方程: $IN[B] = \bigcap_{P \text{ 是 } B \text{ 的所有前驱}} OUT[P]$

边界条件: $OUT[ENTRY] = \emptyset$

各节点的初始化条件: $OUT[B] = U$

2、(5 分)

B1 的支配节点集合为 {B1}

B2 的支配节点集合为 {B1、B2}

B3 的支配节点集合为 {B1、B2、B3}

B4 的支配节点集合为 {B1、B2、B3、B4}

B5 的支配节点集合为 {B1、B5}

3、(4 分)

回边 $B4 \rightarrow B3$, 对应的循环为 {B3,B4}

回边 $B5 \rightarrow B1$, 对应的循环为 {B1,B2,B3,B4,B5}

4、(6 分)

B2: $k2=4$

B3: $k2=k2+2$

B4: if $k2 < 24$ goto B3

中国科学技术大学

2019--2020 学年第 2 学期考试试卷 A 卷标准答案

考试科目: 编译原理 B

得分: _____

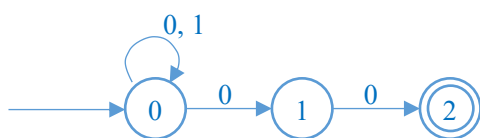
学生所在院系: _____ 姓名: _____ 学号: _____

1. 在字母表 $\Sigma = \{0,1\}$ 上,

- 手工构造识别语言“以 00 结尾的串”的 NFA, 且非 DFA。
- 手工构造识别 a) 中语言的最简 DFA。
- 构造识别 a) 中语言的正规式。
- 手工构造识别语言: 能被 3 整除的二进制数 (不含空串) 的最简 DFA。(15 分)

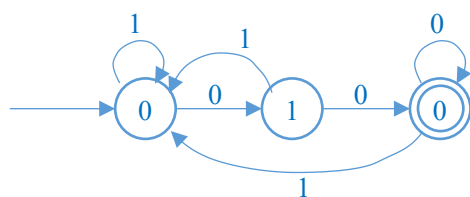
答:

a)



(4 分)

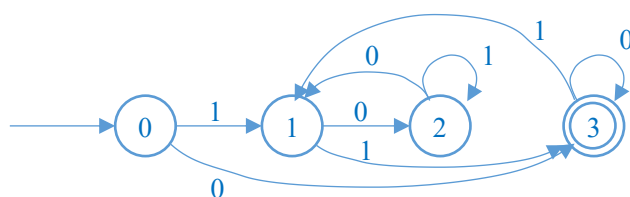
b)



(4 分)

c) $(0|1)^*00$ (4 分)

d)



(3 分)

2. 根据以下递归下降的预测分析函数伪代码 (其中, 带引号的都表示终结符),

```

void S() {
    if (lookahead=='for') {
        L(); S(); }
}

void L() {
    if (lookahead=='for') {
        match('for'); E(); S();
        match('next'); }
    else error();}
  
```

```

void E() {
    if (lookahead=='id') {
        match('id'); match('=');
        match('num'); match('to');
        match('num');}
    else error();}
  
```

- 写出对应的文法。
- 写出所有非终结符的 **first** 集合与 **follow** 集合。
- 构造该文法的 LL(1) 预测分析表。(15 分)

答:

a)

$S \rightarrow LS \mid \varepsilon$

$L \rightarrow \text{for } E \text{ } S \text{ next}$

$E \rightarrow \text{id} = \text{num to num}$ (4 分)

b)

$\text{first}(S) = \{\text{for}, \varepsilon\}, \text{follow}(S) = \{\text{next}, \$\}$

$\text{first}(L) = \{\text{for}\}, \text{follow}(L) = \{\text{for}, \text{next}, \$\}$

$\text{first}(E) = \{\text{id}\}, \text{follow}(E) = \{\text{for}, \text{next}\}$ (6 分)

c)

	For	next	id	= num to	\$
S	$S \rightarrow LS$	$S \rightarrow \varepsilon$			$S \rightarrow \varepsilon$
L	$L \rightarrow \text{for } E \text{ } S \text{ next}$				
E			$E \rightarrow \text{id} = \text{num to num}$		

(5 分)

3. 为{0, 1}构成的能被 2 整除的非空二进制串,

- 编写一个 LR 文法。(最好是 SLR 文法)
- 为该文法编写一个语法制导定义, 判断该二进制串是否能够被 4 整除。
- 为该文法编写一个语法制导定义, 判断该二进制串是否能够被 8 整除。
- 如果你编写的是 SLR 文法, 请写出项目集, 并构造 SLR 分析表; 如果你编写的不是 SLR 文法, 请找出其中的移进-规约和规约-规约冲突 (如果有的话)。

提醒: 0 是可以被 4 和 8 整除的。(20 分)

答:

a) SLR 文法: $S \rightarrow L0$; $L \rightarrow L1 \mid L0 \mid \varepsilon$ (5 分)

b)

$L \rightarrow L1 \mid \varepsilon \{L.\text{len}=0\}$

$L \rightarrow L_1 0 \mid \{L.\text{len}=L_1.\text{len}+1\}$

$S \rightarrow L0 \{ \text{if } (L.\text{len} \geq 1) \text{ print("能被 4 整除"); else print("不能被 4 整除"); } \}$ (5 分)

c)

$L \rightarrow L1 \mid \varepsilon \{L.\text{len}=0\}$

$L \rightarrow L_1 0 \mid \{L.\text{len}=L_1.\text{len}+1\}$

$S \rightarrow L0 \{ \text{if } (L.\text{len} \geq 2) \text{ print("能被 8 整除"); else print("不能被 8 整除"); } \}$ (5 分)

d)

$S' \rightarrow S$; $S \rightarrow L0$; $L \rightarrow L1 \mid L0 \mid \varepsilon$

*** I0 ***

$S' \rightarrow \cdot S$

$S \rightarrow \cdot L0$

$L \rightarrow \cdot L1$

$L \rightarrow \cdot L0$

$L \rightarrow \cdot$

*** I1 ***

$S' \rightarrow S \cdot$

*** I2 ***

$S \rightarrow L \cdot 0$

$L \rightarrow L \cdot 1$

$L \rightarrow L \cdot 0$

*** I3 ***

$S \rightarrow L0 \cdot$

$L \rightarrow L0 \cdot$

*** I4 ***

$L \rightarrow L1 \cdot$

	0	1	\$	S	L
0	$r, L \rightarrow \epsilon$	$r, L \rightarrow \epsilon$		1	2
1			acc		
2	s3	s4			
3	$r, L \rightarrow L0$	$r, L \rightarrow L0$	$r, S \rightarrow L0$		
4	$r, L \rightarrow L1$	$r, L \rightarrow L1$			

(5 分)

4. 简要说明什么是类型化语言？什么是类型检查？请列举类型化语言的优点（不少于 3 条）。
(10 分)

答：

类型化语言是为每种运算的各运算对象和运算结果都定义了所允许的类型语言。（2 分）

类型检查是检查程序中运算对象和运算结果是否符合类型系统定义，发现类型错误。（2 分）

优点是：编译时尽早发现错误；程序多模块可以相互独立的开发；更有效地安排空间和访问方式。（6 分）

5. 访问链记录的是什么？用途是什么？C 语言为什么没有访问链？控制链记录的是什么？作用是什么？C 语言为什么需要控制链？（10 分）

答：

访问链记录了名字作用域覆盖到本函数/过程的函数/过程的活动记录在活动记录栈中的位置，以便访问它们的名字。（3 分）

C 语言只有全局变量和局部变量之分。在函数内只能访问本活动记录的名字和全局的名字，因此不需要访问链。（2 分）

控制链用于记录调用本函数/过程的上一级函数/过程的活动记录基地址，以便函数/过程返回时恢复基地址。（3 分）

C 语言支持函数间调用，因此需要控制链来恢复活动记录栈中的基地址。（2 分）

6. 程序片段 `int i=1; printf("%d, %d, %d, %d, ", i++, i++, i=2, i); printf("%d", i);` 在某编译环境下的执行结果为 2, 2, 2, 1, 4。请分析该编译器可能采取了什么策略，导致了这个结果？（10 分）

答：

该编译器计算实参时，按照从右到左的顺序执行，导致第一个 `printf` 语句的后两个显示结果为 2, 1，并且 `i` 的值改为 2。在处理 `++` 运算时，先取值，后运算，而该运算延迟到了 `printf` 函数执行之后才进行。因此第一个 `printf` 语句的前两个 `i++` 的取值均取为 2。在 `printf` 函数后，两次 `++` 执行，`i` 的值改为 4。因此第二个 `printf` 语句的结果为 4。

7. 在代码生成过程中，需要维护寄存器描述和名字的地址描述。请问，什么是寄存器描述？什么是名字的地址描述？它们是如何发挥作用的？（10 分）

答：

寄存器描述记录了每个寄存器当中保存了哪些名字的值。（3 分）

名字的地址描述记录了每个名字的值可以在哪些位置找到。（3 分）

生成目标代码时，可以通过名字的地址描述找到适合取值的地址，通过寄存器描述来优化分配寄存器，来提高所生成代码的执行效率。（4 分）

8. 有以下中间代码，该段代码之前已对 i 进行了赋值，之后仅变量 d 保持活跃。

```

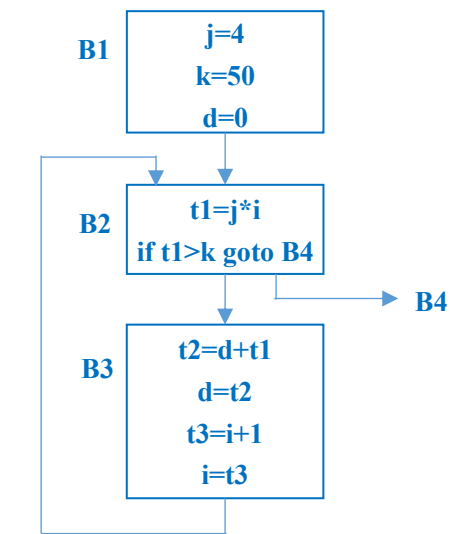
j=4
k=50
d=0
L1: t1=j*i
    if t1>k goto L2
    t2=d+t1
d=t2
t3=i+1
i=t3
goto L1
L2:

```

- 划分程序块，并画出程序流图。
- 手工优化该代码。（10 分）

答：

a)



(5 分)

b)

```

t1=4*i
d=0
L1: if t1>50 goto L2
d=d+t1
t1= t1+4
goto L1
L2:
(5 分)

```

中国科学技术大学
2018--2019 学年第 2 学期考试试卷答案 A 卷

考试科目: 编译原理与技术

得分: _____

学生姓名: _____ 学号: _____

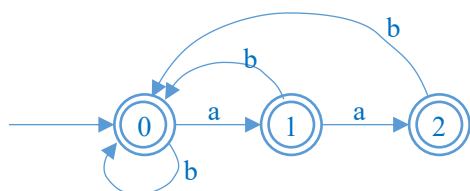
一、在字母表 $\Sigma = \{a, b\}$ 上, 为语言: 不含连续 3 个 a 的串 (包括空串),

1. 构造描述该语言的正规式。
2. 构造识别该语言的最简 DFA。
3. 构造描述该语言的 LL 文法。 (15 分)

答:

1. $(b^* (a|aa)b)^* (a|aa|\epsilon)$ (5 分)

2.



(5 分)

3. (5 分)

$S \rightarrow bS \mid aA \mid \epsilon$

$A \rightarrow bS \mid aB \mid \epsilon$

$B \rightarrow bS \mid \epsilon$

二、为文法 $S \rightarrow aSbS \mid \epsilon$,

1. 构造递归下降的预测分析函数。其中辅助函数

```
void match (terminal t) {  
    if (lookahead == t) lookahead = nextToken();  
    else error();  
}
```

2. 计算 $\text{first}(S)$ 和 $\text{follow}(S)$, 构造 LL(1) 分析表。
3. 构造 LR(1) 项目集及分析表。
4. 它是 SLR 文法吗? 为什么? (20 分)

答:

1. (5 分)

```
void S() {  
    if (lookahead == 'a') {  
        match('a'); S(); match('b'); S();  
    }  
}
```

2. (5 分)

$\text{first}(S) = \{a, \epsilon\}$, $\text{follow}(S) = \{b, \$\}$

	a	b	\$
S	$S \rightarrow aSbS$	$S \rightarrow \varepsilon$	$S \rightarrow \varepsilon$

3. (8 分)

I0: $S \rightarrow aS \cdot bS, \$$

$S' \rightarrow \cdot S, \$$

$S \rightarrow \cdot aSbS, \$$

$S \rightarrow \cdot, \$$

I1:

$S' \rightarrow S \cdot, \$$

I2:

$S \rightarrow a \cdot SbS, \$$

$S \rightarrow \cdot aSbS, b$

$S \rightarrow \cdot, b$

I3:

I4:

$S \rightarrow a \cdot SbS, b$

$S \rightarrow \cdot aSbS, b$

$S \rightarrow \cdot, b$

I5:

$S \rightarrow aSb \cdot S, \$$

$S \rightarrow \cdot aSbS, \$$

$S \rightarrow \cdot, \$$

I6:

$S \rightarrow aS \cdot bS, b$

I7:

$S \rightarrow aSbS \cdot, \$$

I8:

$S \rightarrow aSb \cdot S, b$

$S \rightarrow \cdot aSbS, b$

$S \rightarrow \cdot, b$

I9:

$S \rightarrow aSbS \cdot, b$

	a	b	\$	S
0	s2		$r, S \rightarrow \varepsilon$	1
1			acc	
2	s4	$r, S \rightarrow \varepsilon$		3
3		s5		
4	s4	$r, S \rightarrow \varepsilon$		6
5	s2		$r, S \rightarrow \varepsilon$	7
6		s8		
7			$r, S \rightarrow aSbS$	
8	s4	$r, S \rightarrow \varepsilon$		9
9		$r, S \rightarrow aSbS$		

4. (2 分)

是 SLR 文法。

因为：遇到 a 就移进；遇到 b 就先做 S 的规约，再将 b 移进；遇到 \$ 就做 S 的规约。

所有动作无须参考搜索符。

也可以列出 SLR 的项目集，表示其中无冲突：

I0:

$S' \rightarrow \cdot S$

$S \rightarrow \cdot aSbS$

$S \rightarrow \cdot$

I1:

$S' \rightarrow S \cdot$

I2:

$S \rightarrow a \cdot SbS$

$S \rightarrow \cdot aSbS$

$S \rightarrow \cdot$

I3:

$S \rightarrow aS \cdot bS$

I5:

$S \rightarrow aSb \cdot S$

$S \rightarrow \cdot aSbS$

$S \rightarrow \cdot$

I7:

$S \rightarrow aSbS \cdot$

三、为 {0, 1} 构成的非空二进制串构造一个 SLR 文法。为该文法写出一个语法制导定义，输出该二进制串的科学记数法表示。例如输入 00011000，输出 1.1000E4；

输入 001, 输出 1. E0; 输入 000, 输出 0. E0。要求只使用综合属性。 (15 分)

答:

文法: (5 分)

$S \rightarrow BL$

$L \rightarrow BL \mid \varepsilon$

$B \rightarrow 0 \mid 1$

语法制导定义: (输出 01, 2 分, 输出指数 3 分, 正确处理前导 0, 2 分, 正确处理小数点 1 分, 指数值正确 2 分)

$B \rightarrow 0 \quad \{B.value=0;\}$

$B \rightarrow 1 \quad \{B.value=1;\}$

$S \rightarrow BS \quad \{$

$L \rightarrow LB \quad \{if \ L.count==0 \ \&\& \ B.value==1$

$\quad \{ \ L.count=1; \ print(B.value); \ print('.') \}; \}$

$\quad \text{else if } L.count>0$

$\quad \{ \ L.count+=1; \ print(B.value); \}$

$\quad \}$

$L \rightarrow B \quad \{if \ B.value==1$

$\quad \{ \ L.count=1; \ print(B.value); \ print('.') \}; \}$

$\quad \text{else } L.count=0;$

$\quad \}$

$S \rightarrow L \quad \{if \ L.count==0$

$\quad \ print('0.E0');$

$\quad \text{else}$

$\quad \{ \ print('E'); \ print(L.value-1); \}$

四、下列两个类型表达式是否存在合一代换? 如存在, 写出最一般的合一代换。

1. $pointer(\beta_1) \rightarrow (pointer(\beta_1) \rightarrow \beta_2)$ 和 $\alpha_1 \rightarrow \alpha_2$

2. $pointer(\beta_1) \rightarrow (pointer(\beta_1) \rightarrow \beta_2)$ 和 $\alpha_1 \rightarrow (\alpha_2 \rightarrow \alpha_3)$

3. $pointer(\beta_1) \rightarrow (pointer(\beta_1) \rightarrow \beta_2)$ 和 $\alpha_1 \rightarrow (\alpha_2 \rightarrow \alpha_2)$

4. $pointer(\beta_1) \rightarrow (pointer(\beta_1) \rightarrow \beta_2)$ 和 $\alpha_1 \rightarrow (\alpha_1 \rightarrow \alpha_1)$ (10 分)

答:

1. $S(\alpha_1)=pointer(\beta_1), S(\alpha_2)=pointer(\beta_1) \rightarrow \beta_2$ (2 分)

2. $S(\alpha_1)=pointer(\beta_1), S(\alpha_2)=pointer(\beta_1), S(\alpha_3) \rightarrow \beta_2$ (3 分)

3. $S(\alpha_1)=pointer(\beta_1), S(\alpha_2)=pointer(\beta_1), S(\beta_2) \rightarrow pointer(\beta_1)$ (3 分)

4. $S(\alpha_1)=pointer(\beta_1), S(\beta_2) \rightarrow pointer(\beta_1)$ (2 分)

五、画出 C 语言活动记录的结构。说明程序(省略头文件包含部分)

```
void main() {  
    int i=10, j=20;  
    printf("%d, %d\n");  
}
```

的输出是什么, 为什么? (假定 printf 函数从右到左计算参数并入栈) (10 分)

答:

从高地址到低地址依次为: 形参、返回值、控制链、机器状态、局部数据 (5 分)

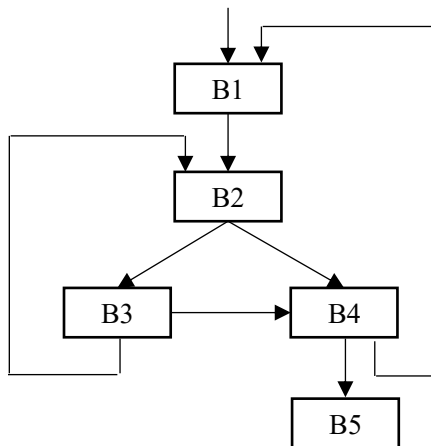
输出 20, 10 (2 分)

printf 依次向栈中寻找参数, 缺少时找到的是主调函数的局部变量。 (3 分)

六、为问题: 计算流图中每个结点的支配结点集合

1. 设计数据流方程, 写出数据流分析的方向、迁移函数、汇合函数、边界条件、初始值。

2. 直接写出以下流图中各程序块出口的支配结点集合(无须计算过程)。 (15 分)



答:

1. (10 分)

分析方向: 正向

迁移函数: $OUT[B] = \{B\} \cup IN[B]$

汇合函数: $IN[B] = \bigcap_{P \text{ 是 } B \text{ 的前驱}} OUT[P]$

边界条件: $IN[ENTRY] = \Phi$

初始值: $OUT[B] = \Phi$

2. (5 分)

$OUT[B1] = \{B1\}$, $OUT[B2] = \{B1, B2\}$, $OUT[B3] = \{B1, B2, B3\}$

$OUT[B4] = \{B1, B2, B4\}$, $OUT[B5] = \{B1, B2, B4, B5\}$

七、结构化高级语言不建议使用 goto 语句。对编译而言, 请分别从生成代码的角度和从优化的角度考虑, goto 语句的存在是否有影响? (10 分)

答:

从生成代码角度考虑, 没有影响。汇编本身就包含转向语句。 (5 分)

从优化角度考虑, 有影响。Goto 语句的存在不利于分析循环, 流图可能不可规约。 (5 分)

八、用不超过 50 个字描述, 本课程的实验难度、工作量如何? (5 分)

答: 略。