

“ 编译原理 ” 考试试卷 (A 卷) 参考答案

考试方式 闭卷 考试时间 考试时长 150 分钟

院（系） 专业班级

学 号 姓 名

题号	一	二	三	四	五	六	七	八	总分	总分人	核对人
分值	10	10	15	15	15	15	5	15	100		
得分											

分 数	
评卷人	

一、（10 分）简答。

1. （5 分）简述中间代码优化的任务和目的。

参考答案：中间代码优化的任务是对前一阶段产生的中间代码进行功能等价变换或改造，目的是使生成的代码更为高效，即更省时间或（和）更节省空间。

2. （5 分）简述 Bison 和 ANTLR 采用的语法分析方法有什么不同。

参考答案：

(1)Bison 采用自底向上的语法分析方法，ANTLR 采用的自顶向下的语法分析方法（5 分）。

具体地说 Bison 采用 LR(1), LALR(1)等分析算法，ANTLR 采用 LL(*)分析算法。

(2)BISON 需要词法分析程序(flex)配合，ANTLR 可独立完成词法分析和语法分析。

(3)ANTLR 能自动生成语法分析树，Bison 需要用户设计产生式对应的语义动作(代码)，来生成语法分析树(或 AST)。（1 分）

分 数	
评卷人	

二、(10 分) 文法与语言。

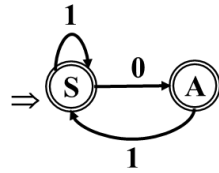
1. (5 分) 给出产生语言 $L=\{w|w \in \{0,1\}^*, \text{并且 } w \text{ 中没有两个连续的 } 0\}$ 的正规文法(仅含形如 $A \rightarrow a|aB|\varepsilon$ 的三类产生式, 其中 A, B 为非终结符, a 为终结符);

参考答案:

$G[S]:$

$S \rightarrow 1S | 0A | \varepsilon$

$A \rightarrow 1S | \varepsilon$



注意: 此题答案不唯一, 需认真审核学生写出的文法是否满足条件。右图给出了与文法等价的 DFA, 但 DFA 不是答案的一部分, 仅用来以更直观的形式表达文法。

2. (5 分) 设有文法 $G[S]:$

$S \rightarrow AB$

$A \rightarrow bA|a$

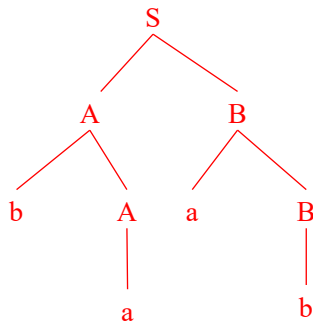
$B \rightarrow aB|b$

(1) 画出句子 baab 的语法推导树以及最左推导 (2 分)

(2) 写出该句子的短语、简单短语和句柄 (3 分)

参考答案:

(1)



最左推导: $S \Rightarrow AB \Rightarrow bAB \Rightarrow baB \Rightarrow baaB \Rightarrow baab$

(2) 短语: a,b,ba,ab,baab

简单短语: a,b

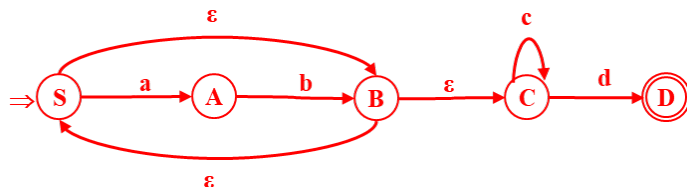
句柄: a

分 数	
评卷人	

三、(15 分) 词法分析

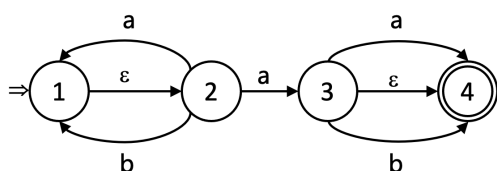
1. (5 分) 给定正规表达式 $R=(ab)^*c^*d$, 构造对应的非确定有限自动机 NFA M, 使得 $L(M)=L(R)$;

参考答案:



注意, 答案不唯一。

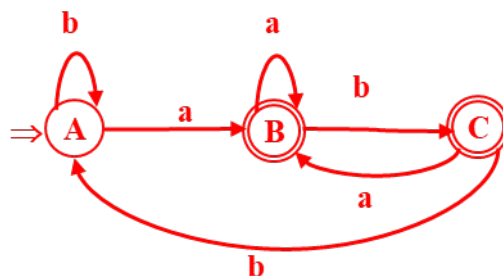
2. (6 分) 给定如下 NFA, 请将其转换成 DFA;



参考答案:

状态子集	a	b
{1, 2}	{1, 2, 3, 4}	{1, 2}
{1, 2, 3, 4}	{1, 2, 3, 4}	{1, 2, 4}
{1, 2, 4}	{1, 2, 3, 4}	{1, 2}

将状态子集依次命名为 A, B, C,
由状态转换图得 DFA。



3. (4 分) 请问 2 中所得到的 DFA 是否是最小化的?如果是请给出证明;如果不是,请将其最小化。
(4 分)

参考答案:

2 得到的 DFA 是最小化的 (结论 1 分)。因为:

首先, A、B、C 均是可达的 (始态 A 接受 a 到状态 B, B 接受 b 到状态 C), 没有无用状态; (1 分)

其次, A 与 B、C 不等价 (B、C 是终态, 而 A 不是) (分割 A 与 BC 1 分)

B 与 C 也不是等价状态, 因为 B 接受 b 到达状态 C (终态), 而 C 接受 b 到达状态 A (非终态)。
(分割 B、C 1 分)

所以, 状态 A、B、C 互不等价。

综上, 所得 DFA 已是最小化状态。

分 数	
评卷人	

四、(15 分) 考虑文法 $G(\{a, b, m\}, \{S, M, N\}, \{S\}, P)$, 其中产生式集合 P 由下列产生构成:

$$\begin{aligned} S &\rightarrow MN \\ M &\rightarrow aMb \mid amb \\ N &\rightarrow aNb \mid m \end{aligned}$$

- (1) (3 分) 求文法 G 所表示的语言;
- (2) (7 分) 该文法是 LL(1) 文法吗? 如果是, 给出理由; 如果不是, 请对方法进行等价变换, 并说明变换后的文法是 LL(1) 文法;
- (3) (5 分) 根据 (2) 所给出的 LL(1) 文法, 建立 LL(1) 分析表。

参考答案:

(1) $L = \{a^k m b^k a^n m b^n \mid k > 0, n \geq 0\}$ 3 分

(2) 因为 M 的两个候选式有左公因子, 故文法不是 LL(1) 的。对文法进行提取左公因子, 变换后得等价文法 G' , 并计算每个产生式的 SELECT 集:

$$\begin{aligned} S &\rightarrow MN && \{a\} \\ M &\rightarrow aM' && \{a\} \\ M' &\rightarrow Mb && \{a\} \\ M' &\rightarrow mb && \{m\} \\ N &\rightarrow aNb && \{a\} \\ N &\rightarrow m && \{m\} \end{aligned}$$

因为 M' 和 N 各自的两个产生式的 SELECT 集的交集均为空集, 故等价变换后的文法是 LL(1) 的。

7 分, 其中产生式 3 分, SELECT 集 3 分, 理由 1 分。

(3) 根据 G' 及各产生式的 SELECT 集, 构造 LL(1) 分析表如下: (5 分)。

$V_N \backslash V_T$	a	m	b	#
S	$S \rightarrow MN$			
M	$M \rightarrow aM'$			
M'	$M' \rightarrow Mb$	$M' \rightarrow mb$		
N	$N \rightarrow aNb$	$N \rightarrow m$		

别解: (2) 文法改写成:

$$\begin{aligned} S &\rightarrow aMbN && S \rightarrow aMbM \\ M &\rightarrow aMb \mid m && \text{或} && M \rightarrow aMb \mid m, \text{显见改写后文法是 LL(1) 的,} \\ N &\rightarrow aNb \mid m \end{aligned}$$

(3) 分析表分别如下:

$V_N \backslash V_T$	a	m	b	#
S	$S \rightarrow aMbN$			
M	$M \rightarrow aMb$	$M \rightarrow m$		
N	$N \rightarrow aNb$	$N \rightarrow m$		

$V_N \backslash V_T$	a	m	b	#
S	$S \rightarrow aMbM$			
M	$M \rightarrow aMb$	$M \rightarrow m$		

分 数	
评卷人	

五、(15 分) 给定文法 $G[S]$:

$S \rightarrow bAC \mid bBd \mid aBS \mid aAe$

$A \rightarrow f$

$B \rightarrow f$

$C \rightarrow c \mid \varepsilon$

(1) (7 分) 构造文法 $G[S]$ 的识别 LR (1) 活前缀的 DFA;

(2) (5 分) 构造 LR(1) 分析表;

(3) (3 分) $G[S]$ 是 LALR(1) 文法吗? 请给出理由。

参考解答:

(1) 拓展文法并对产生式编号:

0: $S' \rightarrow S$

1: $S \rightarrow bAC$

2: $S \rightarrow bBd$

3: $S \rightarrow aBS$

4: $S \rightarrow aAe$

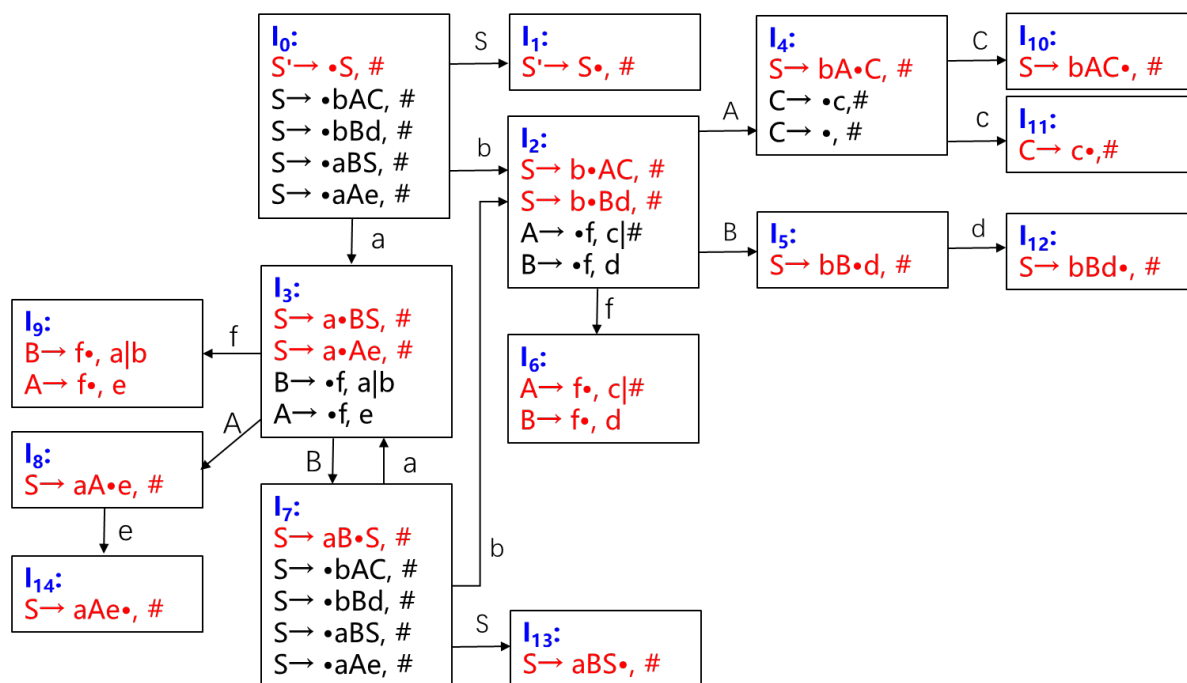
5: $A \rightarrow f$

6: $B \rightarrow f$

7: $C \rightarrow c$

8: $C \rightarrow \varepsilon$

构造 LR(1) 自动机:



(2) 查 DFA 无移进-归约、归约-归约冲突，文法是 LR(1) 的，构造 LR(1) 分析表如下：

	ACTION							GOTO			
	a	b	c	d	e	f	#	S	A	B	C
0	S3	S2						1			
1							ACC				
2						S6			4	5	
3						S9			8	7	
4			S11				r8				10
5				S12							
6			r5	r6			r5				
7	S3	S2						13			
8					S14						
9	r6	r6			r5						
10							r1				
11							r7				
12							r2				
13							r3				
14							r4				

(3) LR(1)DFA 中，I6 和 I9 是同心项目集，合并同心项目集, 得：

I6, I9:

$A \rightarrow f\bullet, c|e|\#$

$B \rightarrow f\bullet, a|b|d$

两个归约项目的向前搜索符号集的交集为空集，即合并同心项目集后不存在归约-归约冲突，故文法是 LALR(1) 的。(注：两个向前搜索符号集分别是 A、B 的 FOLLOW 集)

分 数	
评卷人	

六、(15 分) 下面是某语言文法的部分产生式及相应的翻译模式片断:

$S \rightarrow \text{'if' ' (' } \{ E.\text{true} := \text{newlabel}; E.\text{false} := S.\text{next} \} E \text{')' } \{ S_1.\text{next} := S.\text{next} \} S_1$
 $\{ S.\text{code} := E.\text{code} || \text{gen}(E.\text{true} ':') || S_1.\text{code} \}$
 $S \rightarrow \text{'for' ' (' } \{ T_1.\text{next} := \text{newlabel} \} T_1 \text{' ;' }$
 $\{ E.\text{true} := \text{newlabel} ; E.\text{false} := S.\text{next} \} E ;$
 $\{ T_2.\text{next} := T_1.\text{next} \} T_2 \text{')' } \{ S_1.\text{next} := \text{newlabel} \} S_1$
 $\{ S.\text{code} := T_1.\text{code} || \text{gen}(T_1.\text{next} ':') || E.\text{code} || \text{gen}(E.\text{true} ':')$
 $\quad || S_1.\text{code} \quad || \text{gen}(S_1.\text{next} ':') || T_2.\text{code} || \text{gen}(\text{'goto' } T_1.\text{next}) \}$
 $S \rightarrow \{ T.\text{next} := S.\text{next} \} T \{ S.\text{code} = T.\text{code} \}$
 $T \rightarrow \text{id '=' } A \{ T.\text{code} := A.\text{code} || \text{gen}(\text{id}.\text{place} ':=' A.\text{place}) \}$
 $A \rightarrow \text{id } \{ A.\text{place} := \text{id}.\text{place}; A.\text{code} := \text{' ' } \}$
 $A \rightarrow \text{int } \{ A.\text{place} := \text{newtemp}; A.\text{code} := \text{gen}(A.\text{place} ':=' \text{int.val}) \}$
 $A \rightarrow A_1 \text{ op } A_2$
 $\{ A.\text{place} := \text{newtemp}; A.\text{code} := A_1.\text{code} || A_2.\text{code} || \text{gen}(A.\text{place} ':=' A_1.\text{place op } A_2.\text{place}) \}$
 $E \rightarrow A_1 \text{ rop } A_2 \{ E.\text{code} := A_1.\text{code} || A_2.\text{code} || \text{gen}(\text{'if' } A_1.\text{place rop } A_2.\text{place 'goto' } E.\text{true}) || \text{gen}$
 $\text{'goto' } E.\text{false}) \}$

解答内容不得超过装订线

注: S代表语句(特别地, T为赋值语句), id代表标识符, A代表算术表达式, E代表布尔表达式, rop代表关系比较运算符(如'<=', '=='等), op代表算术运算符(如'+', '/', '%'等)。

语义属性说明:

id.place : 对应id的存储位置;

A.place: 存放A的值的存储位置

A.code|E.code : 对A|E求值的三地址代码序列;

E.true和E.false分别表示布尔表达式为真和假时, 程序要跳转到的位置, 即标号。

S.code|T.code : S或T 的三地址代码序列 ;

S.next: 表示 S 之后要执行的首条 TAC 语句的标号。

语义函数/过程说明:

gen() : 生成一条三地址代码;

newtemp : 在符号表中新建一个从未使用过的名字(从t1开始), 并返回该名字的存储位置;

newlabel 返回一个新的语句标号(从 L1 开始)。

||: 是三地址代码序列之间的链接运算;

1. (5 分) 根据题设给出的文法画出语句:

$\text{for } (i = 1; i < n; i = i + 1) \text{ if } (n \% i == 0) s = s + i$

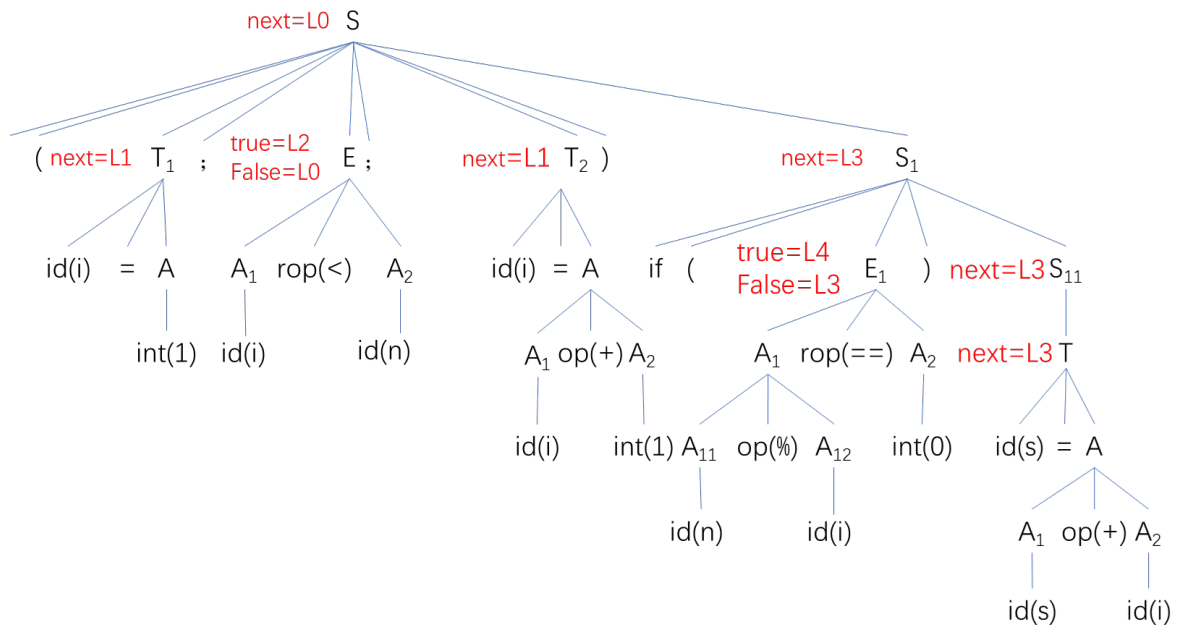
的语法分析树, 同一子树下的同名符号请用下标加以区别;

2. (4 分) 根据题设给出的翻译模式, 在语法树上标注各结点的继承属性值(设根结点的 next 属性值为 L0);

3. (6 分) 根据题设给出的翻译模式, 将(1)给出的语句翻译成中间代码, 直接写出三地址码序列。

参考解答：

(1) 语法分析树



(2) 各结点的继承属性标在语法中对应结点的左侧，如上图所示。

(3) 三地址码序列：

t1 = 1

i = t1

L1:

if i < n goto L2

goto L0

L2:

t4 = n % i

t5 = 0

if t4==t5 goto L4

goto L3

L4:

t6 = s + i

s = t6

L3:

t2 = 1

t3 = i + t2

i = t3

goto L1

L0:

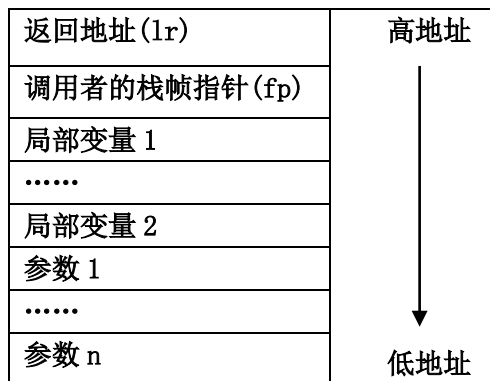
分 数	
评卷人	

七、(5 分) 有以下 C 语言程序:

```
void setarr(int a[],int pos, int val){
    a[pos]=val;
    // 栈帧观察点
}

int main(){
    int a[3] = {31,32,33};
    setarr(a,1,50);
    return a[1];
}
```

在运行 Raspberry Pi OS (Raspbian GNU/Linux 10) 操作系统的树莓派 (Raspberry 4B) 硬件设备上经某编译器 (未开启任何优化选项) 编译成 armv7-a (32 位) 目标代码后运行。设函数的栈帧布局如下图所示 (仅展示了主要内容, 栈从高地址向低地址增长):



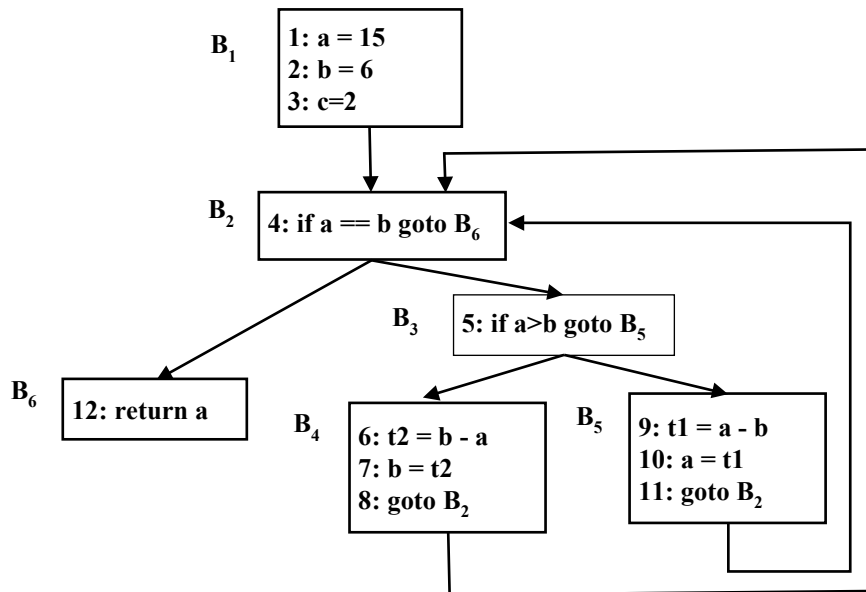
在寄存器 lr 的值不被破坏的前提下, 其值也可以不保存在栈帧中。下表记录了一次实际运行时函数 main 和 setarr 栈帧的内容。请在下表空白的地方填写当程序运行至“栈帧观察点”时, 函数 main 和 setarr 栈帧中缺失的存储单元内容(符号/含义及值), 每空 0.5 分。

地址	值	含义	备注
0xbefff544	0xb6e6c718	返回地址(lr)	main 的栈帧
0xbefff540	0x00000000	调用者帧指针(fp)	
0xbefff53c	33	a[2]	
0xbefff538	50	a[1]	
0xbefff534	31	a[0]	
0xbefff530	0x000102e0	_start 入口地址	
0xbefff52c	0xbefff544	main 的帧指针(fp)	setarr 的栈帧
0xbefff528	0x00000000	保留	
0xbefff524	0xbefff534	数组 a 的首地址	
0xbefff520	1	参数 pos	
0xbefff51c	0x00000032	参数 val	

分 数	
评卷人	

八、(15 分) 数据流分析及中间代码优化

1. (5 分) 已知流图如下:



- (1) 求能到达基本块 B_4 出口的定值点集合 $\text{Out}[B_4] = \{ 1, 3, 6, 7, 9, 10 \}$;
- (2) 求基本块 B_5 入口处的活跃变量集合 $\text{LiveIn}[B_5] = \{ a, b \}$;
- (3) 求变量 a 在定值点 10 的 DU 链 = $\{ 4, 5, 6, 9, 12 \}$;
- (4) 求变量 b 在引用点 9 的 UD 链 = $\{ 2, 7 \}$;
- (5) B_6 的支配(必经)结点集合 = $\{ B_1, B_2, B_5 \}$;

2. (10 分) 数组 a 存储了一支股票连续 10 个交易日价格(int 型, 小于 1 万). 你只能选择某个交易日买入, 并在未来的另一个交易日卖出该股票. 以下是求解 10 个交易日所能获取的最大利润的 C 语言程序片段所对应的三地址码序列 (添加了多余代码并对最大利润作了附加奖励):

```

1: low = 10000
2: high = 0
3: i = 0
4: s = 1
5: j = 4 * i
6: t1 = a[j]
7: if t1 >= low goto 12
8: low = i
9: t2 = 4 * i
10: t3 = a[t2]
11: low = t3
12: t4 = 4 * i
13: t5 = a[t4]
14: t6 = t5 - low
15: b = 4 * s
16: if t6 <= high goto 22
17: high = i
18: t7 = 4 * i
19: t8 = a[t7]
20: t9 = t8 - low
21: high = t9
22: t10 = i + s
23: i = t10
24: if i < 10 goto 5
25: t11 = high + b
26: return t11
  
```

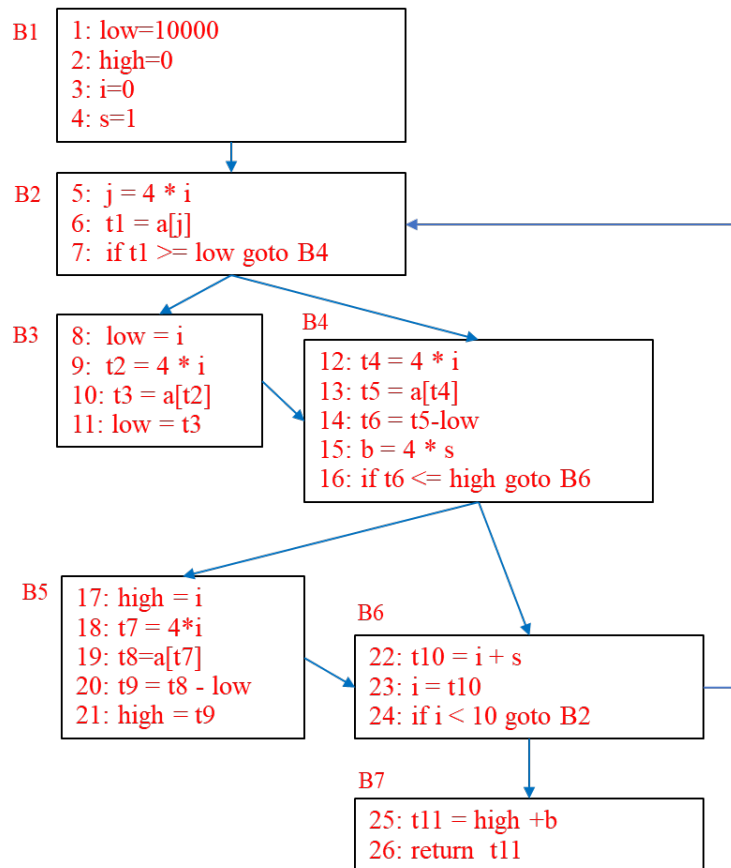
- (1) (3 分) 请将以上三地址代码序列划分为基本块, 绘出其流图;
- (2) (3 分) 对流图进行常量传播、常量折叠、CSE、复写传播、DCE 等优化, 简述优化过程, 绘出优化后的流图;

(3) (1 分)找出流图中的循环;

(4) (3 分)对循环进行不变计算外提、循环归纳变量强度削弱、删除循环基本归纳变量等优化。简述优化过程,并绘出优化后的流图。

参考解答:

(1)基本块与流图



(2) 15 和 22 对 s 的引用来自唯一定值点 4, 且定值是常量, 可以执行常量传播优化, 并对 15 进行常量折叠:

15: b = 4

22: t10 = i + 1

5, 9, 12, 18 中的 $4*i$ 是公共子表达式, 可执行 CSE 优化, 9, 12, 18 改用复写语句, 直接引用 5 计算的结果:

9: t2=j

12: t4=j

18: t7=j

对 10, 13, 19 中执行复写传播, 用 j 分别代替 t2, t4, t7。4, 9, 12, 18 变成死代码, 可删除。

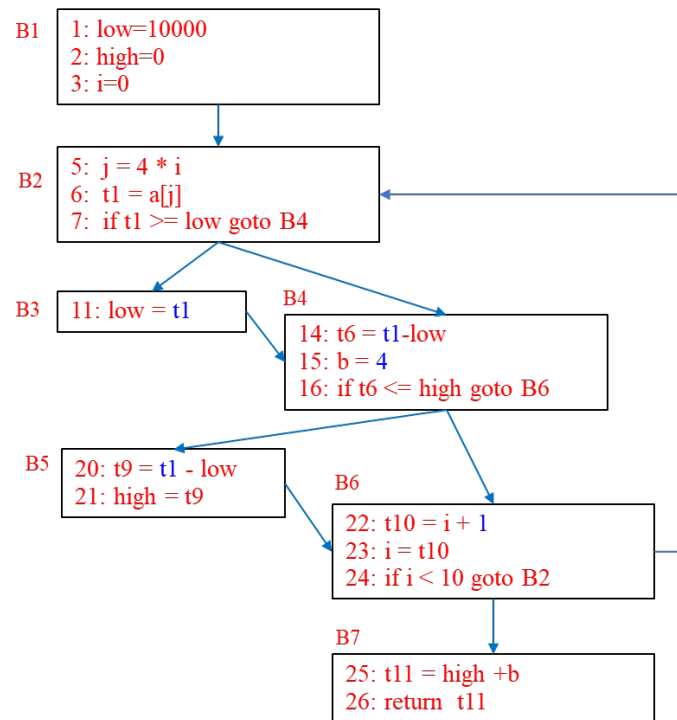
6, 10, 13, 19 中的 $a[j]$ 是公共子表达式, 可执行 CSE, 10, 13, 19 改用复写语句, 直接引用 6 的计算结果:

10: t3=t1

13: t5=t1

19: t8=t1

执行复写传播, 11, 14, 20 中的 t1, t3, t5 均改引用 t1, 10, 13, 19 变成死代码, 可删除。此外, 8 对 low 的定值, 17 对 high 的定值都是死代码, 可删除。得流图:



(3) {B2, B3, B4, B5, B6} 构成循环 (回边 B6→B2)

(4) 15 是循环不变计算, 且是循环出口的必经结点, 循环内没有对 b 的引用或定值, 可以外提至新设的前置结点, 再与 B1 合并。

变量 i 是基本归纳变量, j 是 i 同族的归纳变量。可以进行强度削弱, 用 $j=j+4$ 取代 $j=4*i$, 并将该定值移至循环体尾部。并在前置结点对 j 作初始化: $j=0$; 用 j 表达循环控制条件, 可删除归纳变量 i。此外, $b=4$ 是常量, 执行常量传播后, 变成死代码, 可删除。优化后的流图:

