

## 第九章作业

### 9.1

#### 参考答案

9.1

(1) 为便于说明，将上述代码加上语句编号后如下所示：

```
(1)      read (n)
(2)      i:=1
(3)      fen:=1
(4)  L1:  if i<=n goto L2
(5)      goto L3
(6)  L2:  t1:=fen*i
(7)      fen:=t1
(8)      i:=i+1
(9)      goto L1
(10) L3:  write(fen)
```

根据基本块划分方法，首先确定基本块的入口语句，这里语句(1)、(4)、(5)、(6)、(10)都是入口语句；然后确定基本块。这段代码共划分为 5 个基本块，其中  $B_1=\{(1), (2), (3)\}$ ， $B_2=\{(4)\}$ ， $B_3=\{(5)\}$ ， $B_4=\{(6), (7), (8), (9)\}$ ， $B_5=\{(10)\}$ 。

(2) 基本块划分结果及相应的流图如图 1 所示。

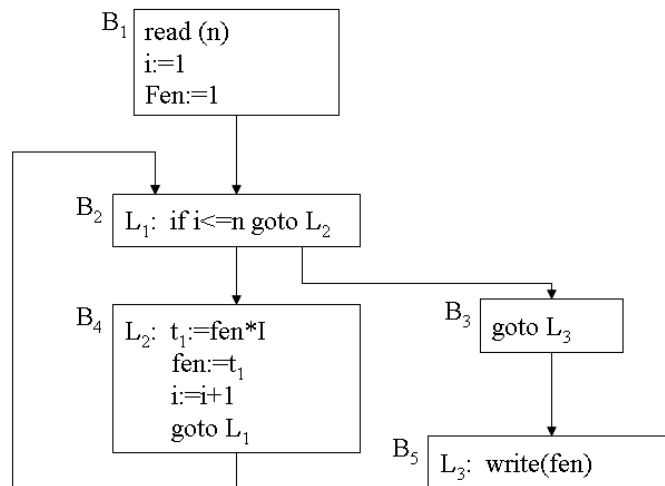


图 1 基本块划分结果及相应的流图

第十章作业

10.3

10.4

参考答案

10.3

(1) 这段代码中有三个入口语句，分别是第一条语句、语句标号 L 标识的语句和最后一条语句，故可以划分为 3 个基本块。相应的流图如图 1 所示。

(2) 首先对  $B_2$  进行优化。

削弱计算强度。对语句  $A:=K*I$  和  $B:=J*I$  进行计算强度削弱，由于循环控制变量  $I$  每次加 10，可以用  $T_1:=K*10$  和  $A:=A+T_1$  代替  $A:=K*I$ ，用  $T_2:=J*10$  和  $B:=B+T_2$  代替  $B:=J*I$ ，这就需要把对  $A$  和  $B$  赋初值的语句  $A:=K*I$  和  $B:=J*I$  提到循环之前，放在基本块  $B_1$  中。

对基本块  $B_1$  进行优化，经过常数传播和常数合并之后得到语句  $A:=K, B:=J$ 。而  $B_2$  中的语句  $T_1:=K*10$  和  $A:=A+T_1$ 、以及  $T_2:=J*10$  和  $B:=B+T_2$  需要放在对  $A$  和  $B$  的引用之后，可以放在 goto 语句之前。这样，得到如图 2 所示的基本块和流图。

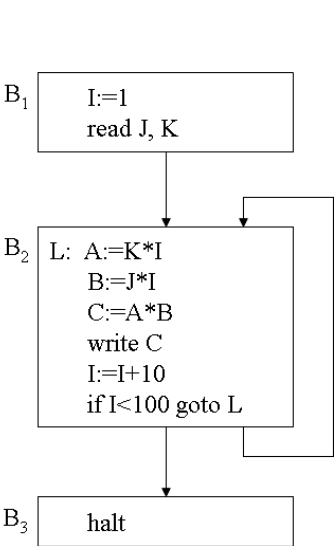


图 1 代码对应的基本块和流图

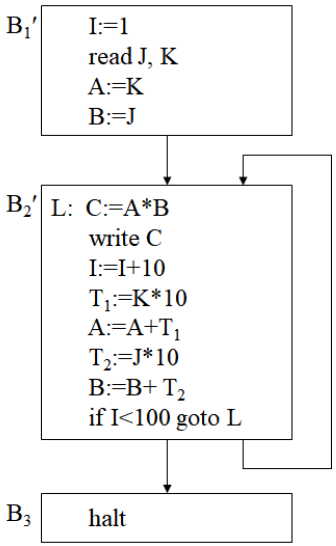


图 2 削弱计算强度后的基本块和流图

删除归纳变量。

$I$  是基本归纳变量，由于存在线性关系  $A:=K*I$  和  $B:=J*I$ ，所以， $A$ 、 $B$  和  $I$  是同族归纳变量。所以，循环条件  $I<100$  可以用  $A<K*100$  或  $B<J*100$  代替，假如选前者，则 if 语句就可以变换为：

$T_3:=K*100$   
if  $A<T_3$  goto L

这样，就可以删除归纳变量  $I$ ，语句  $I:=I+10$  删除后，得到如图 3 所示的基本块和流图。

代码外提。

由于  $K$  和  $J$  的值在程序执行过程中保持不变，即  $K$  和  $J$  是循环不变量，故  $K*10$ 、 $J*10$  和  $K*100$  也是循环不变量，可以将他们提到循环之外，放在循环之前，即  $B_1$  中，删除  $B_1$  中的死代码  $I:=1$ ，得到如图 4 所示的基本块和流图。

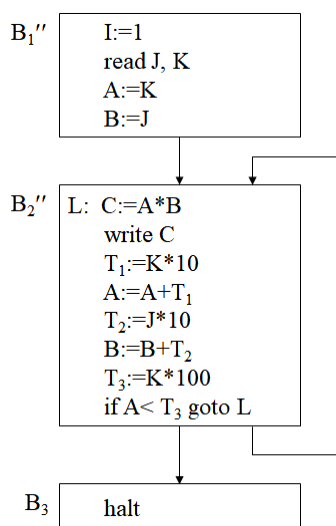


图3 删除归纳变量后的基本块和流图

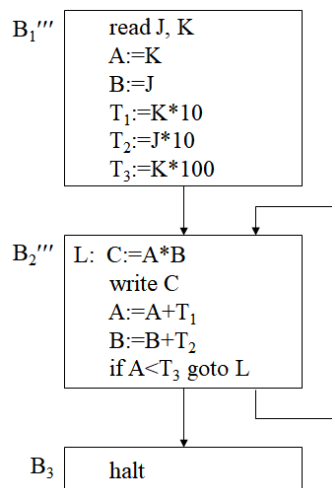


图4 代码外提后的基本块和流图

#### 10.4

(1) 假设数组  $a$  和  $b$  的存储空间起始地址分别用  $A$  和  $B$  表示，其元素地址计算公式中的常量分别用  $a$  和  $b$  表示，整型数的域宽为 4，则根据主教材第 8 章的翻译方案，可以将该程序段中的可执行语句翻译为如下的三地址代码。

- (1)  $i:=1$
- (2) if  $i>m$  goto (21)
- (3)  $j:=1$
- (4) if  $j>n$  goto (18)
- (5)  $t_1:=i*n$
- (6)  $t_1:=t_1+j$
- (7)  $t_2:=A-a$
- (8)  $t_3:=4*t_1$
- (9)  $t_4:=i*n$
- (10)  $t_4:=t_4+j$
- (11)  $t_5:=B-b$
- (12)  $t_6:=4*t_4$
- (13)  $t_7:=t_5[t_6]$
- (14)  $t_2[t_3]:=t_7$
- (15)  $t_8:=j+1$
- (16)  $j:=t_8$
- (17) goto (4)
- (18)  $t_9:=i+1$
- (19)  $i:=t_9$
- (20) goto (2)
- (21) ...

(2) 为将三地址代码划分为基本块，首先需要确定其入口语句，(1)中的三地址代码有 7 个入口语句，分别是(1)、(2)、(3)、(4)、(5)、(18)和(21)。然后根据入口语句可以将三地址代码划

分为 7 个基本块，分别是  $B_1=\{ (1) \}$ 、 $B_2=\{ (2) \}$ 、 $B_3=\{ (3) \}$ 、 $B_4=\{ (4) \}$ 、 $B_5=\{ (5\sim 17) \}$ 、 $B_6=\{ (18)\sim(20) \}$ 和  $B_7=\{ (21) \}$ 。

将控制流信息加入到基本块中，得到如图 5 所示的流图。

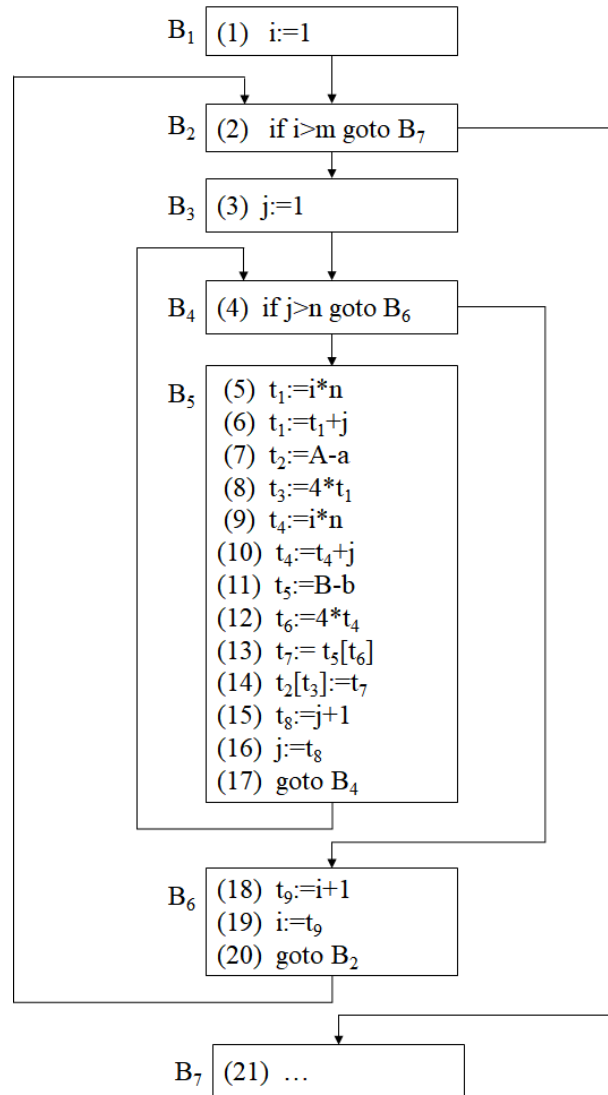


图 5 基本块及流图

(3) 对内循环代码进行所有可能的优化。

由图 5 可以看出，程序中的循环有： $L_1=\{ B_4, B_5 \}$  和  $L_2=\{ B_2, B_3, B_4, B_5, B_6 \}$ ，其中  $L_1$  是内循环。对于内循环  $L_1$  进行优化。

首先对  $B_5$  进行基本块优化。

删除公共子表达式。第(9)句中的  $i*n$  和第(5)句中的  $i*n$  是公共子表达式，第(9)句中的  $i*n$  是冗余的，删除之，另外，第(15)和(16)句可以替换为  $j:=j+1$ ，结果如图 6(a)所示。

复制传播。用  $t_1$  代替后面表达式中出现的  $t_4$ ，涉及到语句(10)和(12)，结果如图 6(b)所示。

删除公共子表达式。第(10)句中的  $t_1+j$  和第(6)句中的  $t_1+j$  是公共子表达式，第(12)句中的  $4*t_1$  和第(8)句中的  $4*t_1$  是公共子表达式，删除第(10)句和第(12)句中冗余的公共子表达式，结果如图 6(c)所示。

复制传播。用  $t_1$  代替后面表达式中出现的  $t_4$ （没有），用  $t_3$  代替后面表达式中出现的  $t_6$ ，涉及到语句(13)，结果如图 6(d)所示。

删除死代码。由于语句(9)、(10)和(12)为  $t_4$  和  $t_6$  赋的值并没有被引用，所以他们是死代码，删除后，结果如图 6(e)所示。

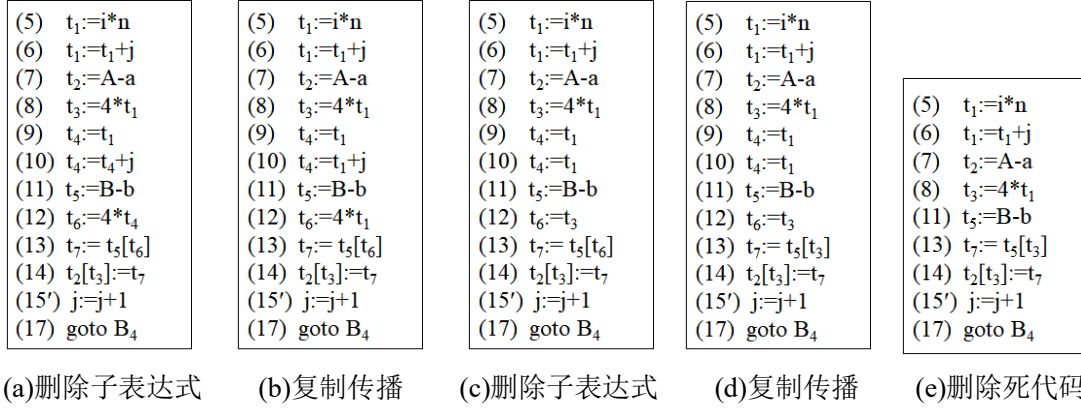


图 6 基本块  $B_5$  的优化过程和结果

基本块  $B_5$  优化后，程序流程图如图 7 所示。

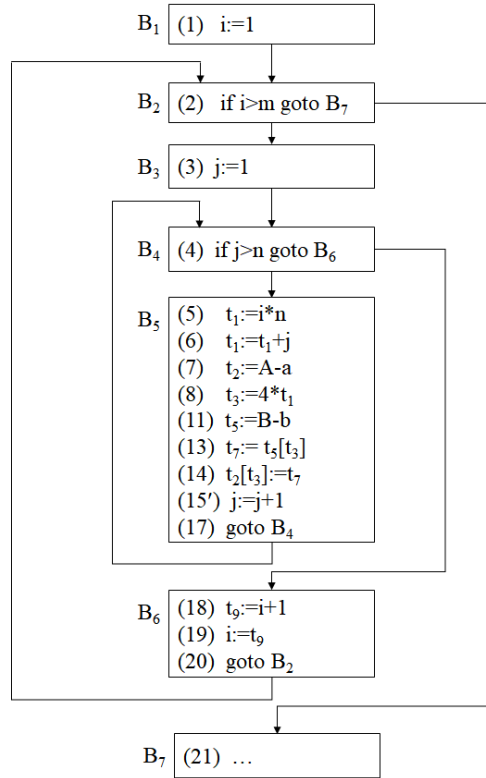


图 7 基本块  $B_5$  优化后的流图

下面对由  $B_4$  和  $B_5$  构成的内循环进行优化。

代码外提。对于内循环而言， $i$ 、 $n$ 、 $A$ 、 $a$ 、 $B$ 、 $b$  均是循环不变量，故可以将语句(5)、(7)和(11)提到循环之前，放在基本块  $B_3$  中。语句(5)外提后，语句(6)  $t_1 := t_1 + j$  右边表达式中的  $t_1$  应该是由语句(5)定值的内循环不变量，为做区分，用  $t_1'$  表示。代码外提之后的基本块  $B_3$  和  $B_5$  如图 8(a)所示。

削弱计算强度。由语句(6')可知，每次循环， $t_1$  和  $j$  同步加 1，故语句(6')可以改写为  $t_1 := t_1 + 1$ ，而把为  $t_1$  赋初值的语句  $t_1 := t_1 + j$  放在循环之前，即基本块  $B_3$  中，而语句  $t_1 := t_1 + 1$  需要调整到对  $t_1$  引用之后，可以放在语句(15')之后。由语句(8)可知，每次循环， $t_3$  和  $t_1$  同步增加， $t_1$  加 1， $t_3$

加 4，故语句(8)可以改写为  $t_3:=t_3+4$ ，而把为  $t_3$  赋初值的语句  $t_3:=4*t_3$  放在循环之前，即基本块  $B_3$  中，而语句  $t_3:=t_3+4$  需要调整到对  $t_3$  引用之后，即放在语句(14)之后。削弱计算强度后的基本块  $B_3$  和  $B_5$  如图 8(b)所示。

删除归纳变量。对于内循环而言， $j$  是基本归纳变量， $t_1$  和  $t_3$  是  $j$  的同族归纳变量。由于  $t_1:=t_1'+j$ ， $t_3:=4*t_1$ ，故  $j>n$  等价于  $t_3>4*(t_1'+n)$ ，所以，在进入内循环之前，在基本块  $B_3$  中先计算出  $4*(t_1'+n)$  的值  $T$ ，即增加语句  $T:=t_1'+n$  和  $T:=4*T$ ，并且用  $t_3>T$  代替基本块  $B_4$  中的条件表达式。这样，内循环中的归纳变量  $j$  和  $t_1$  就不需要了，可以删除，即删除语句(15')和(6')，得到如图 8(c)所示的基本块  $B_3$ 、 $B_4$  和  $B_5$ 。

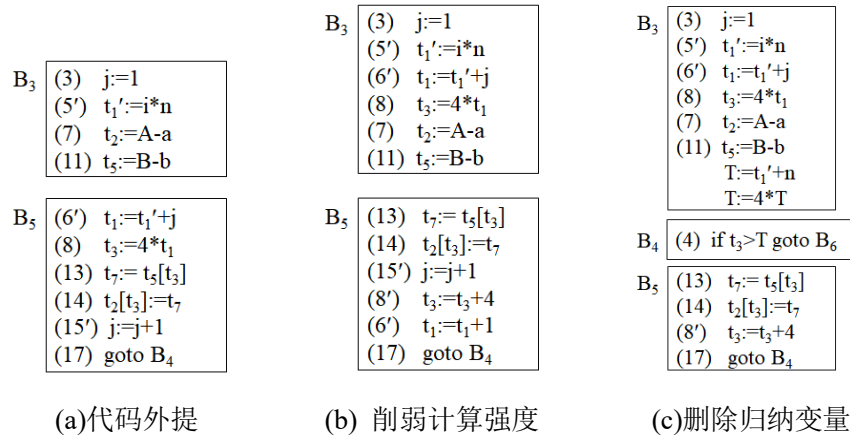


图 8 内循环优化过程和结果

内循环优化后，程序流程图如图 9 所示。

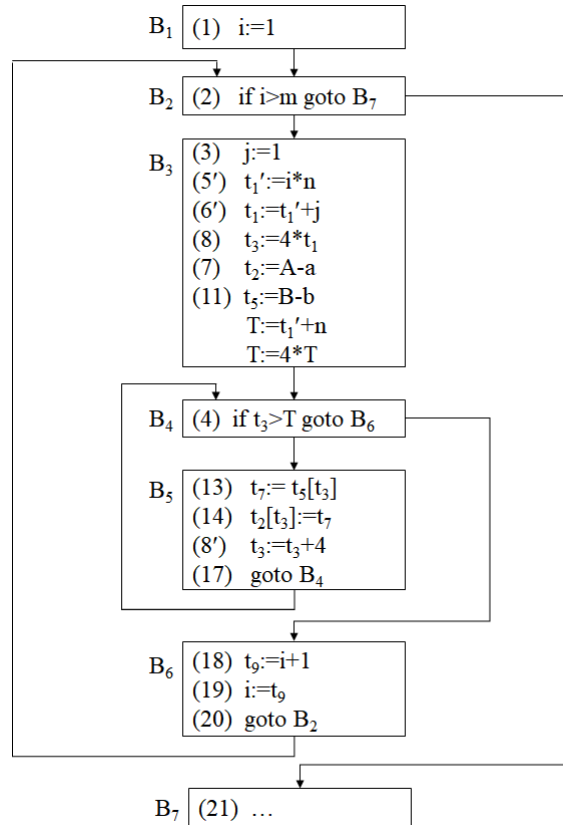


图 9 内循环优化后的流图

进一步，可以对外循环进行优化。

由于在外循环中，(7)  $t_2:=A-a$  和(11)  $t_5:=B-b$  也是循环不变量，故可以将他们提到外循环之前，放到基本块  $B_1$  中。

削弱计算强度。对基本块  $B_3$  进行计算强度削弱。由于  $t_1':=i*n$ ，每次循环， $i$  的值加 1， $t_1'$  的值加  $n$ ，故可以用  $t_1':=t_1'+n$  代替  $t_1':=i*n$ ，需要将为  $t_1'$  赋初值的语句  $t_1':=i*n$  放在循环之前，即  $B_1$  中，把语句  $t_1':=t_1'+n$  放在引用  $t_1'$  的所有语句之后，为确保无误，可以放在循环变量调节语句之处，即放在基本块  $B_6$  中语句(20)之前。

删除归纳变量。外循环的基本归纳变量是  $i$ ，由于  $t_1':=i*n$ ，所以  $t_1'$  和  $i$  是同族归纳变量。 $i>m$  等价于  $t_1'>m*n$ ，可以在进入循环之前，先计算出  $m*n$  的值，即在基本块  $B_1$  中增加语句  $W:=m*n$ ，将基本块  $B_2$  中的条件表达式替换为  $t_1'>W$ ，这样，对循环而言，变量  $i$  不再需要，可以删除，即删除基本块  $B_6$  中的语句(18)和(19)。

此时，基本块  $B_1$ 、 $B_2$ 、 $B_3$  和  $B_6$  如图 10 所示。

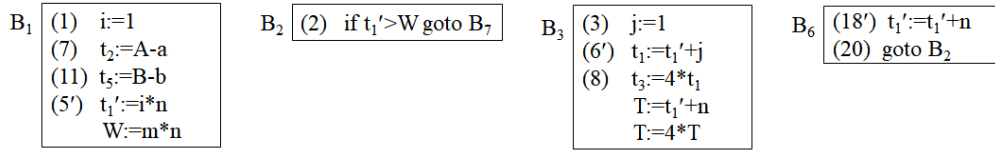


图 10 优化后的基本块  $B_1$ 、 $B_2$ 、 $B_3$  和  $B_6$

再对基本块  $B_1$  和  $B_3$  进行基本块优化，对基本块  $B_1$  依次进行常数传播和常数合并、删除死代码优化，结果如图 11(a)所示，其中常数  $C$  是  $m*n$  进行常数合并的结果。对基本块  $B_3$  依次进行常数传播和删除死代码优化，结果如图 11(b)所示。



(a) 优化后的基本块  $B_1$

(b) 优化后的基本块  $B_3$

图 11 优化后的基本块  $B_1$  和  $B_3$

外循环优化后，程序流图如图 12 所示。

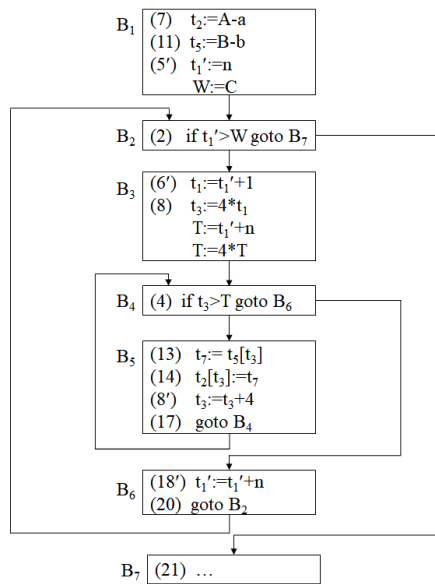


图 12 外循环优化后的流图