

编译原理

循环优化

优化的级别

- ▶ 局部优化
- ▶ 循环优化
- ▶ 全局优化

优化的级别

- ▶ 局部优化
- ▶ 循环优化
 - ▶ 可能反复执行的代码序列
- ▶ 全局优化

循环优化的措施

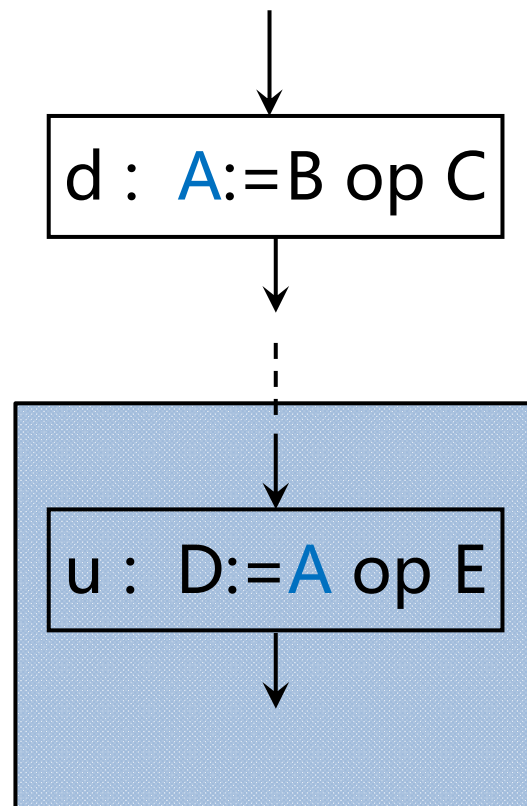
- ▶ 对循环中的代码，可以实行
 - ▶ 代码外提
 - ▶ 强度消弱
 - ▶ 删除归纳变量(变换循环控制条件)
 - ▶ 循环展开
 - ▶ 循环合并

编译原理

代码外提

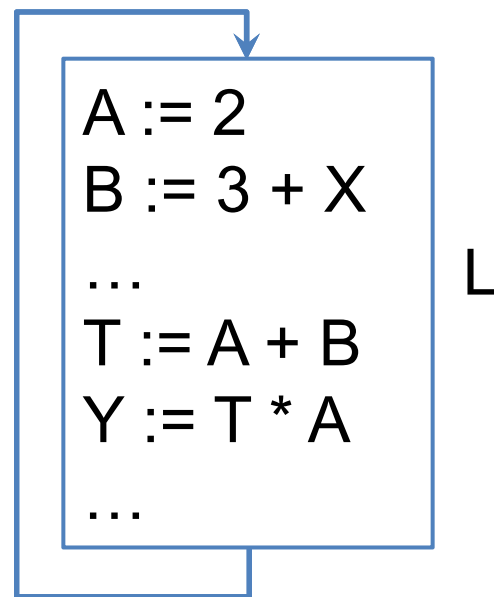
代码外提

- ▶ 所谓变量A在某点d的**定值到达**另一点u（或称变量A的定值点d到达另一点u），是指流图中从d**有一通路**到达u且该通路上没有A的其它定值
- ▶ **循环不变运算**: 对四元式 **$A := B \text{ op } C$** , 若B和C是常数，或者到达它们的B和C的定值点都在循环外



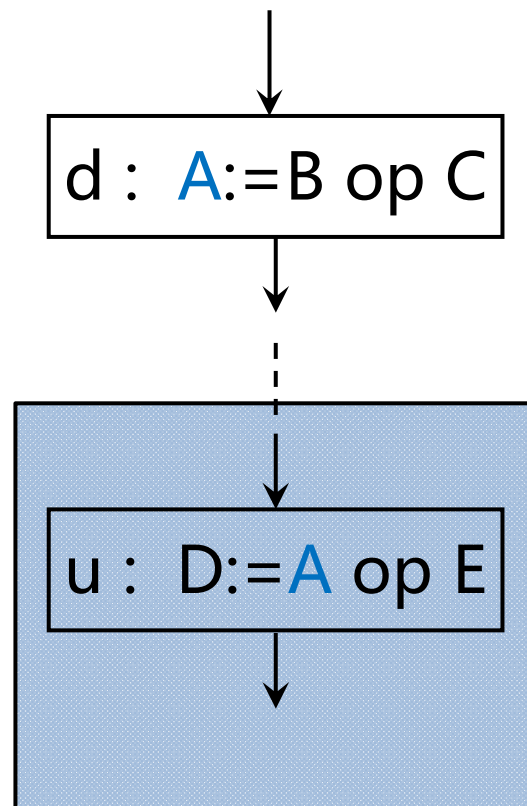
查找循环中不变运算的算法

1. 依次查看L中各基本块的每个四元式，如果它的每个运算对象或为常数，或者定值点在L外，则将此四元式标记为"不变运算"；
2. 重复第3步直至没有新的四元式被标记为"不变运算"为止；
3. 依次查看尚未被标记为"不变运算"的四元式，如果它的每个运算对象或为常数，或定值点在L之外，或只有一个到达-定值点且该点上的四元式已被标记为"不变运算"，则把被查看的四元式标记为"不变运算"。

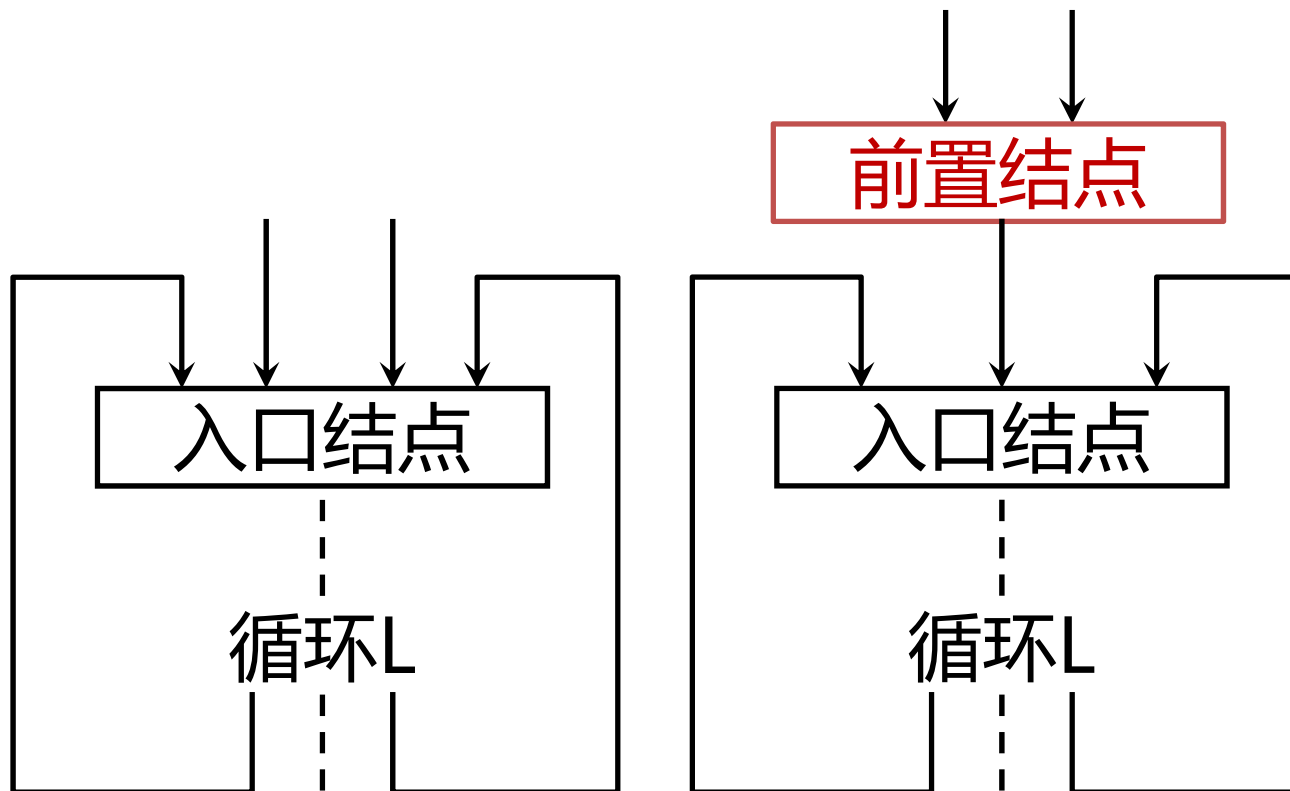


代码外提

- ▶ 所谓变量A在某点d的**定值到达**另一点u（或称变量A的定值点d到达另一点u），是指流图中从d**有一通路**到达u且该通路上没有A的其它定值
- ▶ **循环不变运算**: 对四元式 **$A := B \text{ op } C$** , 若B和C是常数，或者到达它们的B和C的定值点都在循环外
- ▶ 把循环不变运算提到循环体外

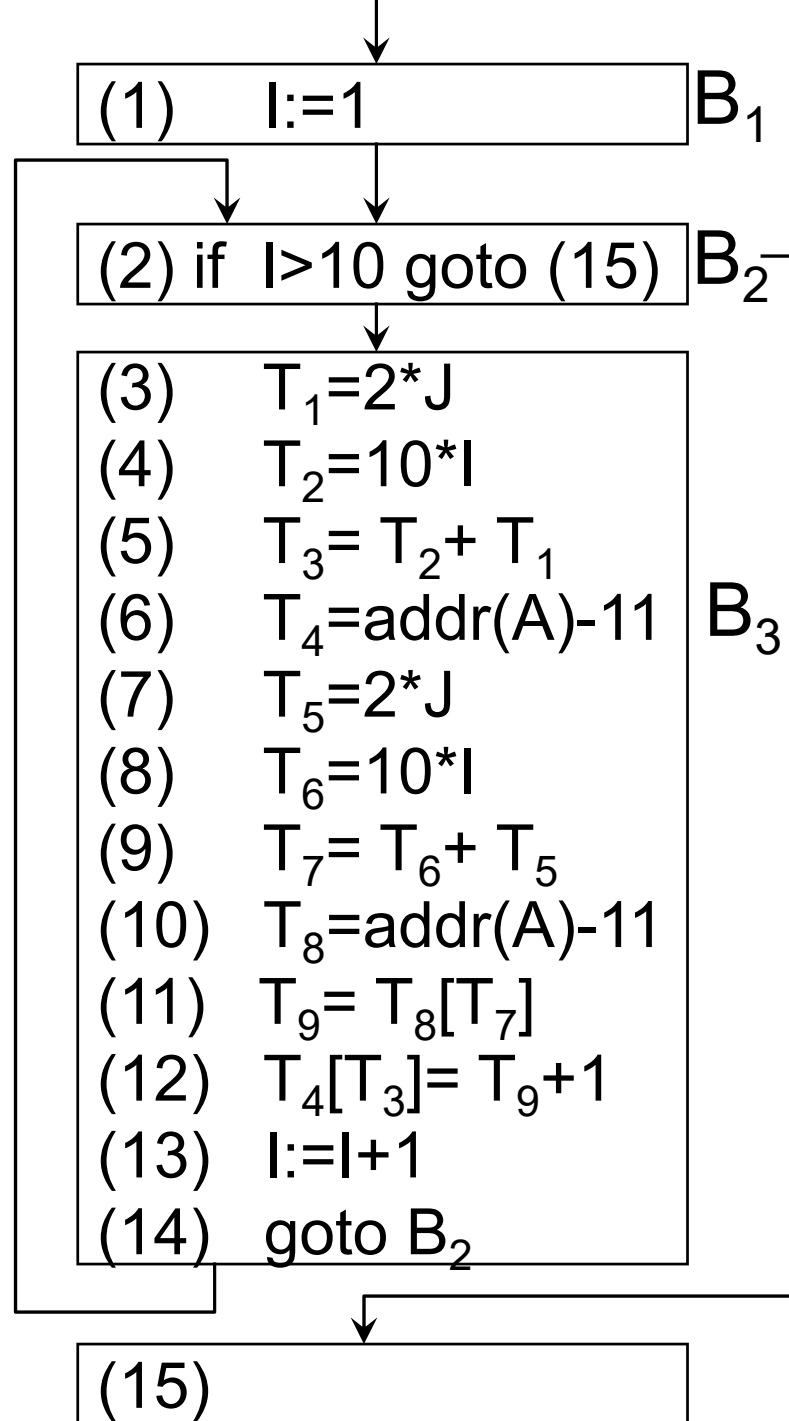


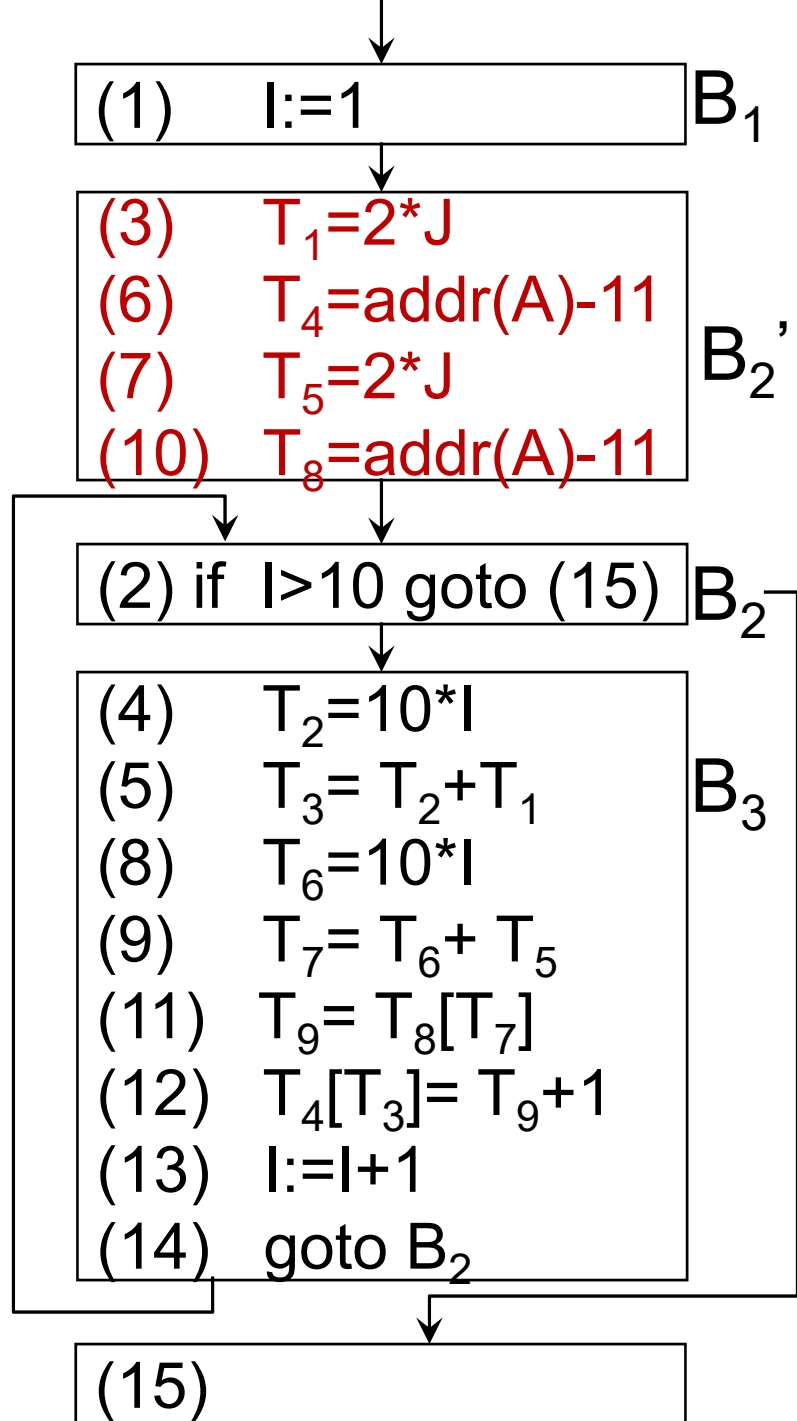
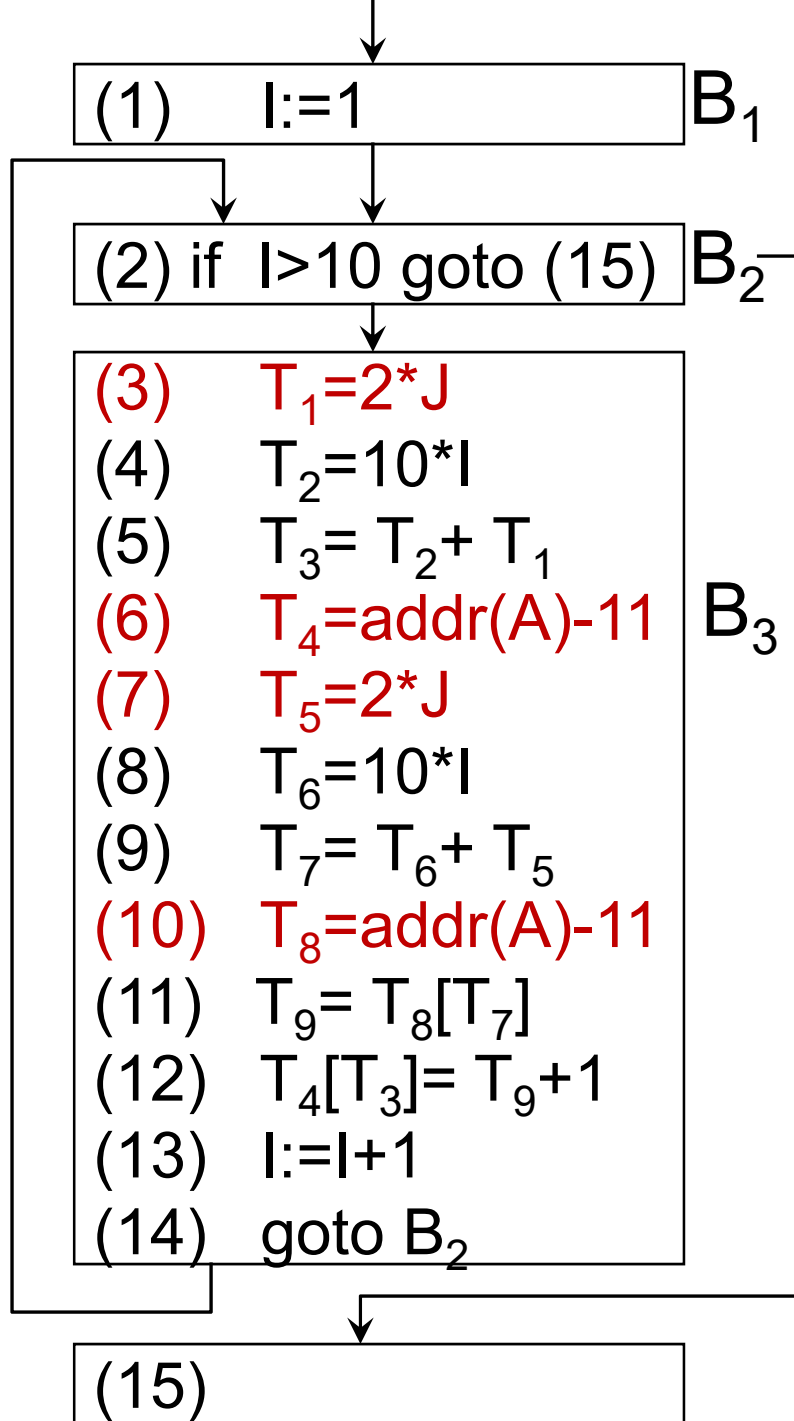
外提的位置



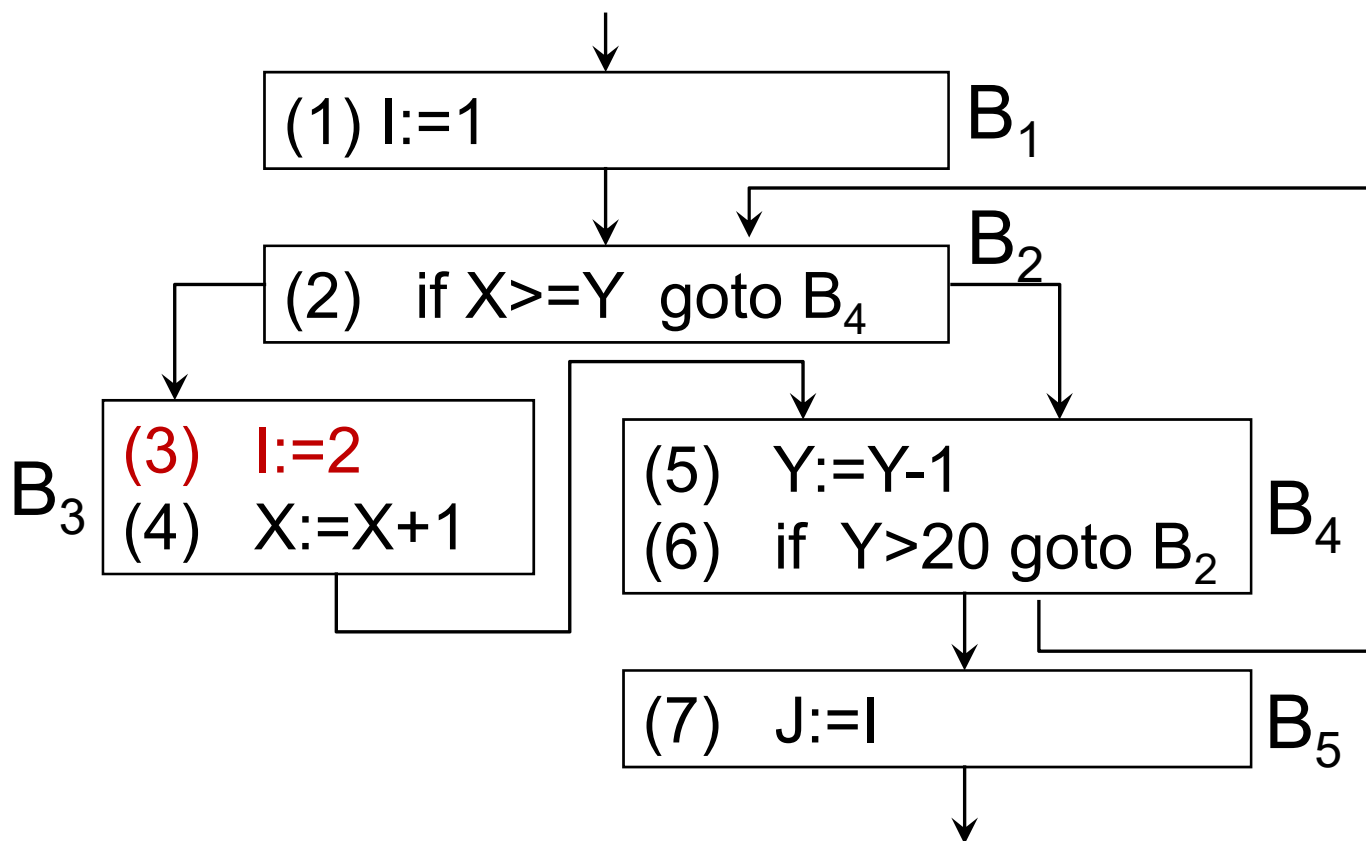
循环优化示例

```
for l:=1 to 10 do  
  A[l, 2*J] := A[l, 2*J] + 1
```





外提的条件

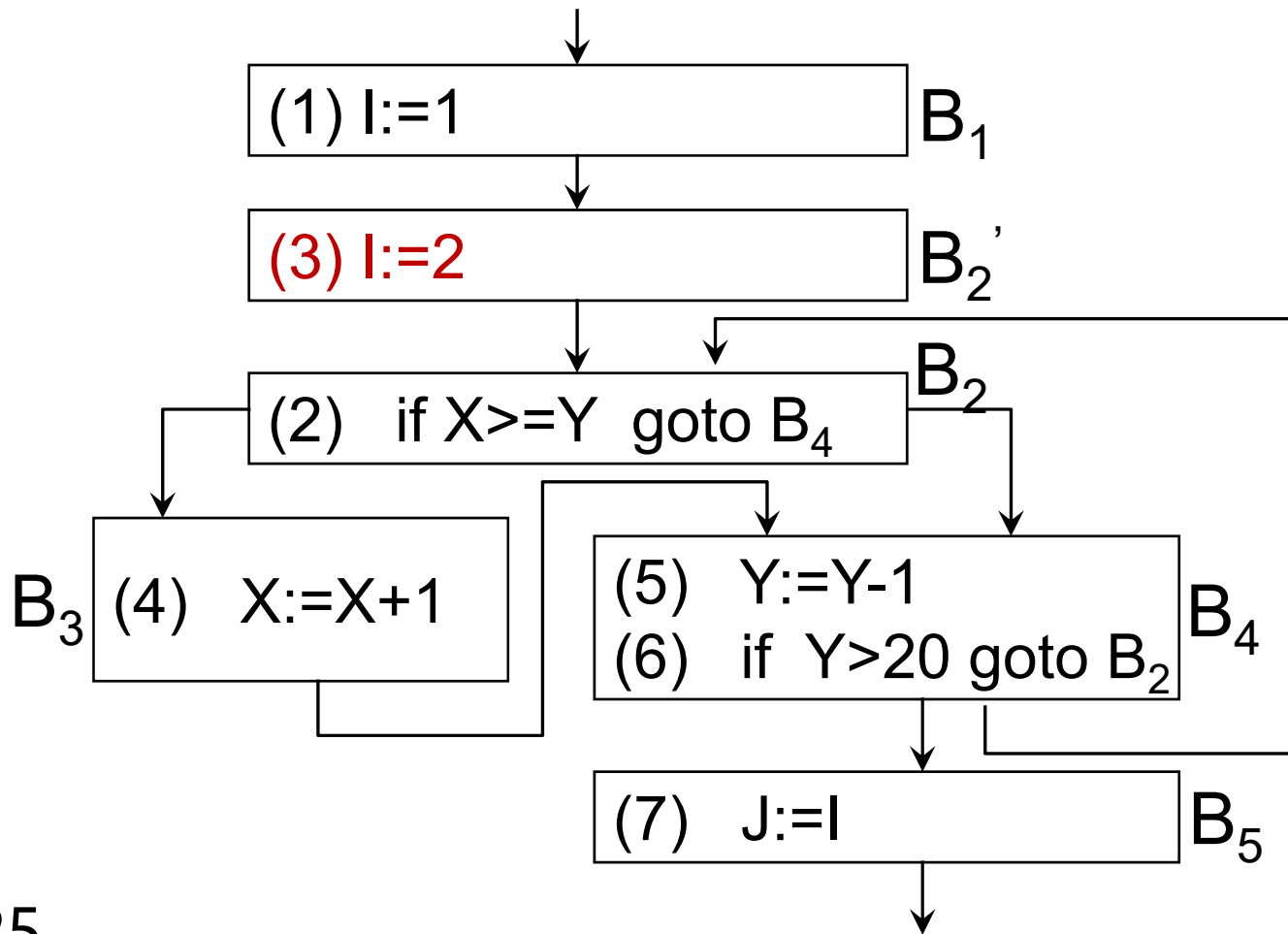


X=30, Y=25

$B_1 \rightarrow B_2 \rightarrow B_4 \rightarrow \dots \rightarrow B_2 \rightarrow B_4 \rightarrow B_5$

I=1, J=1

外提的条件



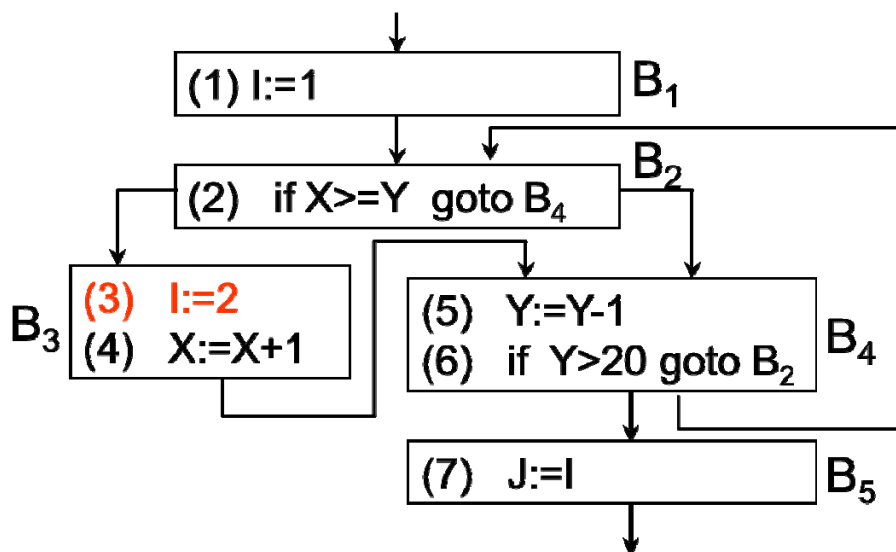
X=30, Y=25

$B_1 \rightarrow B_2' \rightarrow B_2 \rightarrow B_4 \rightarrow \dots \rightarrow B_2 \rightarrow B_4 \rightarrow B_5$

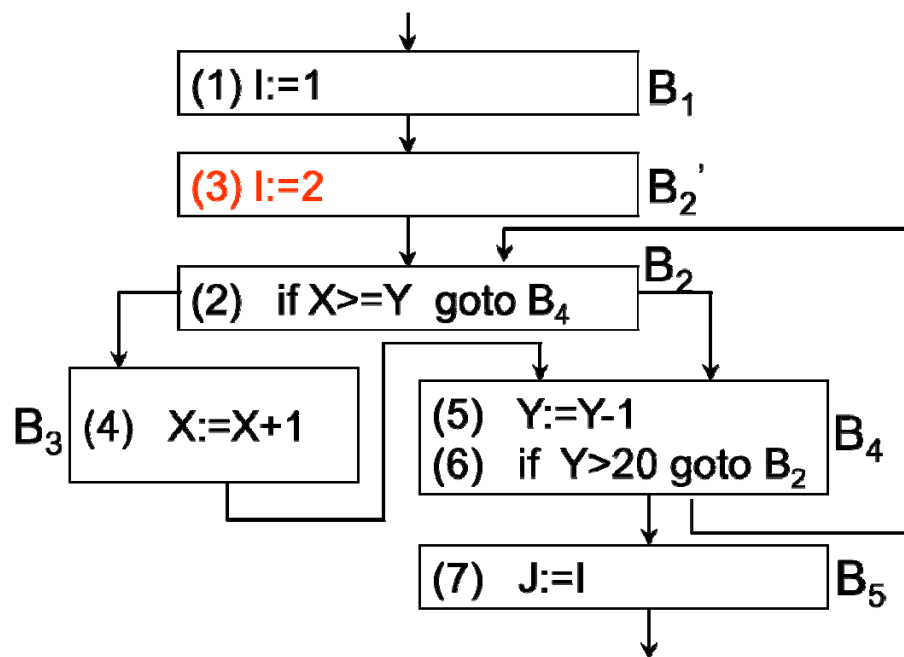
I=2, J=2

外提的条件

- ▶ 不变运算所在的结点是L所有出口结点的必经结点

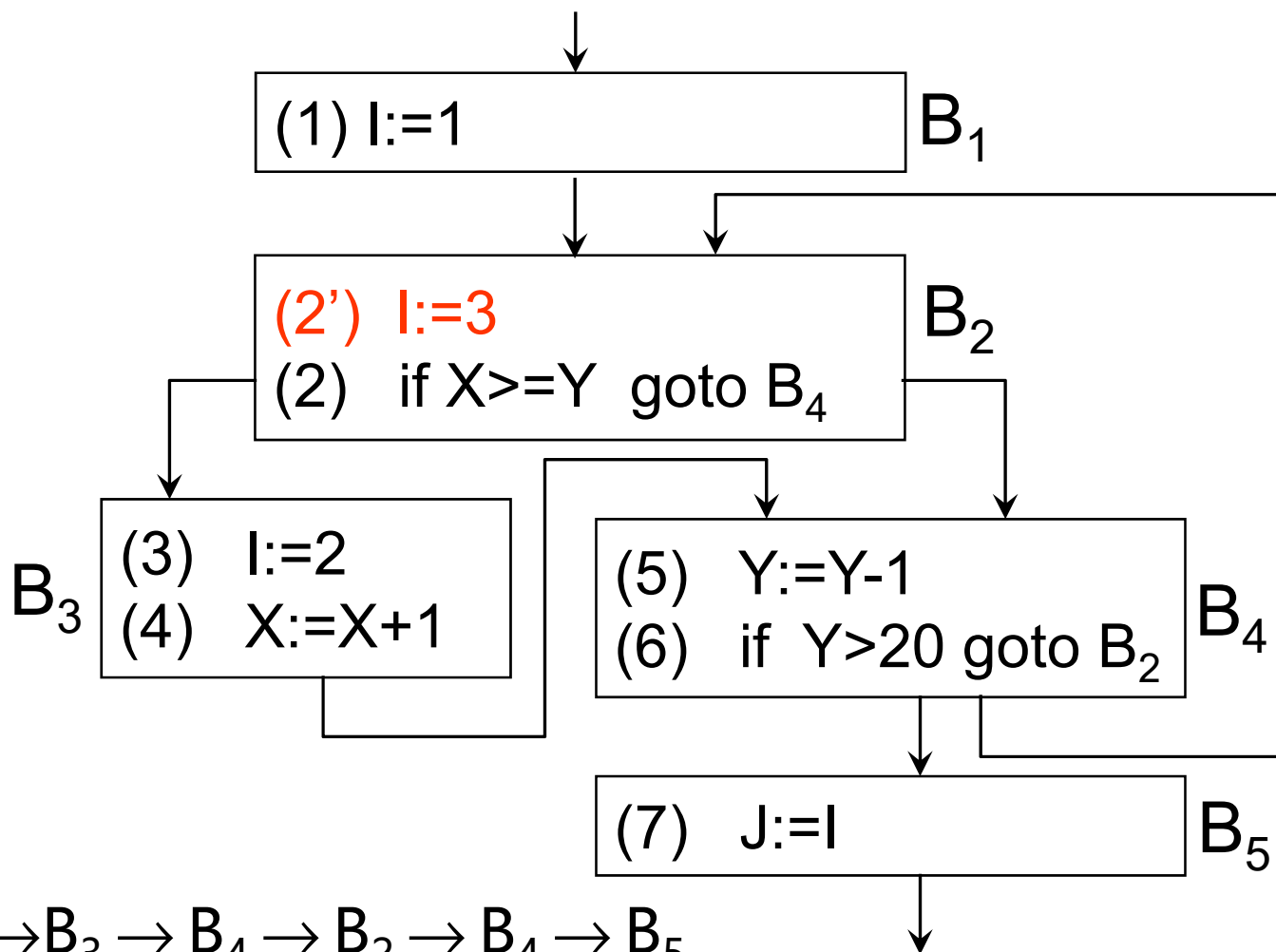


$I=1, J=1$



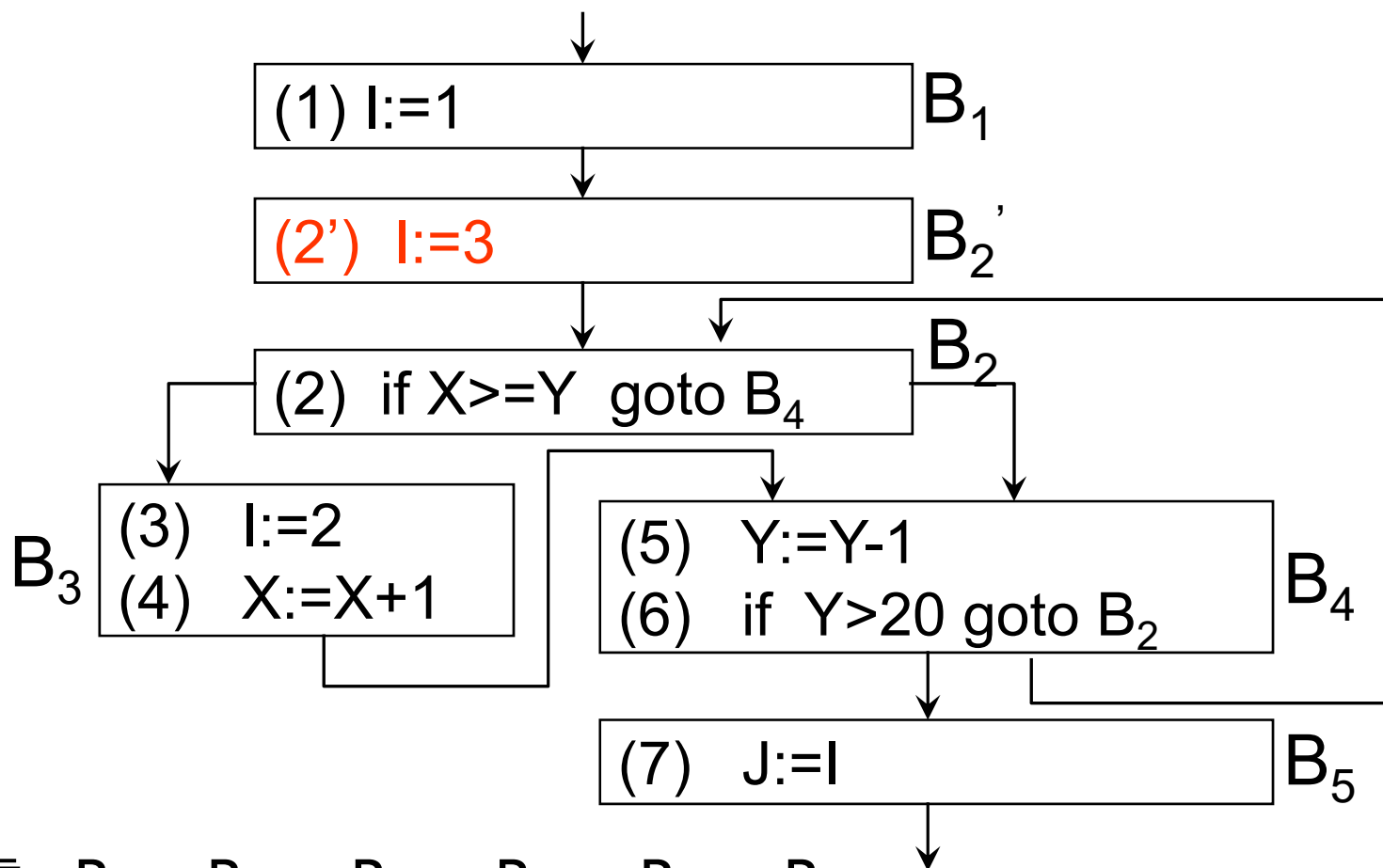
$I=2, J=2$

外提的条件



考虑: $B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_2 \rightarrow B_4 \rightarrow B_5$
 $I=3, J=3$

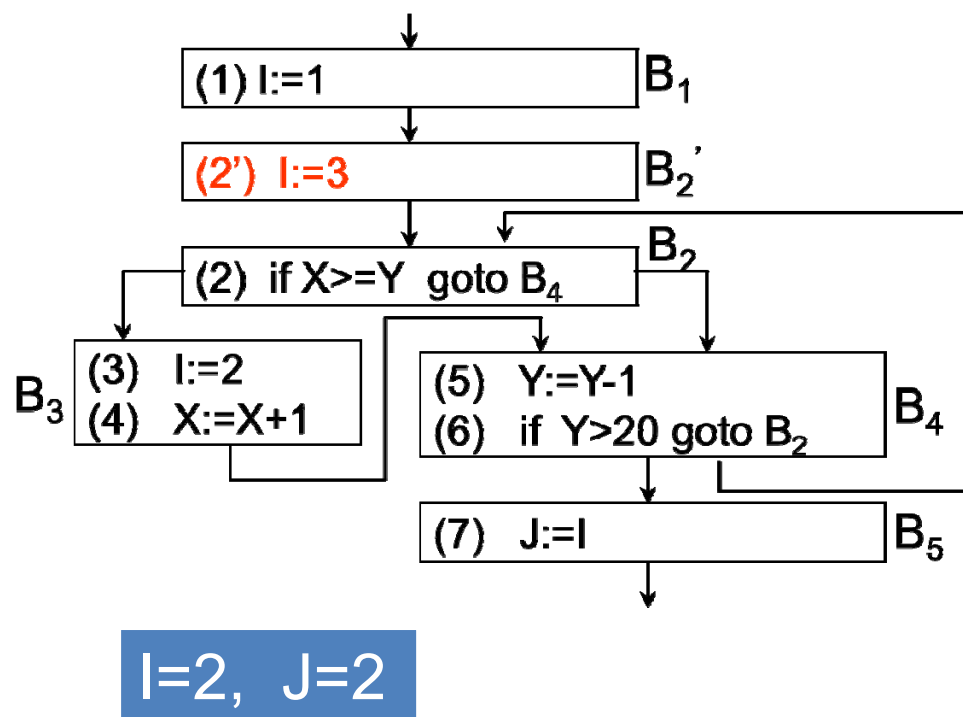
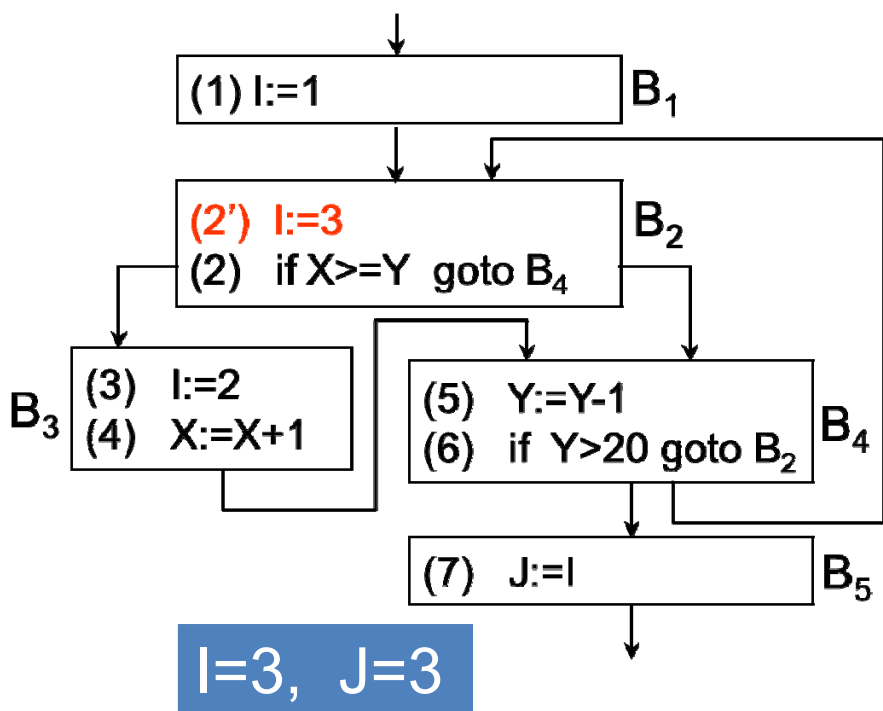
外提的条件



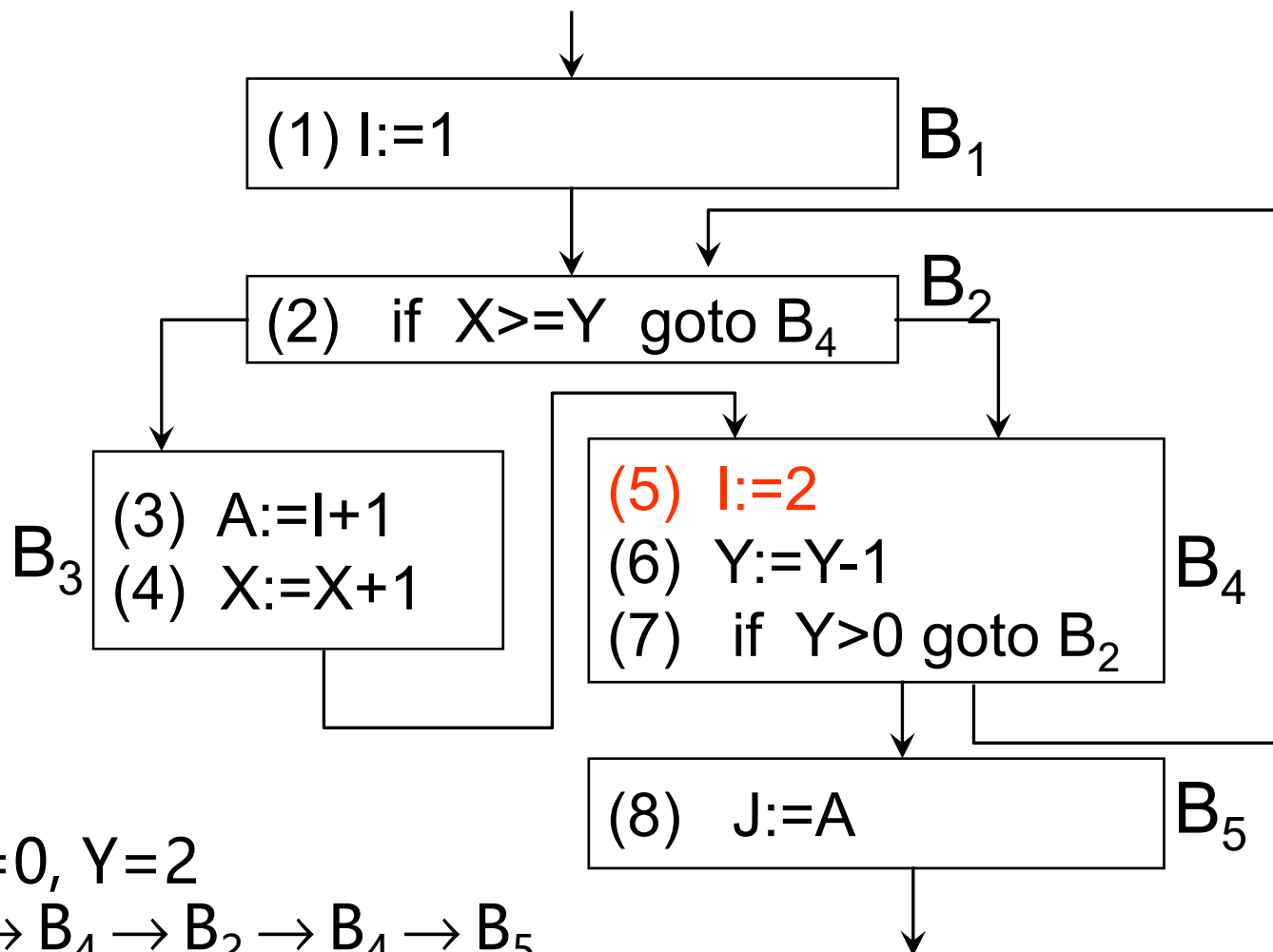
考虑: $B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_2 \rightarrow B_4 \rightarrow B_5$
 $l=2, J=2$

外提的条件

- ▶ A在循环中其他地方未再定值,才能把循环不变运算 $A:=B \text{ op } C$ 外提



外提的条件

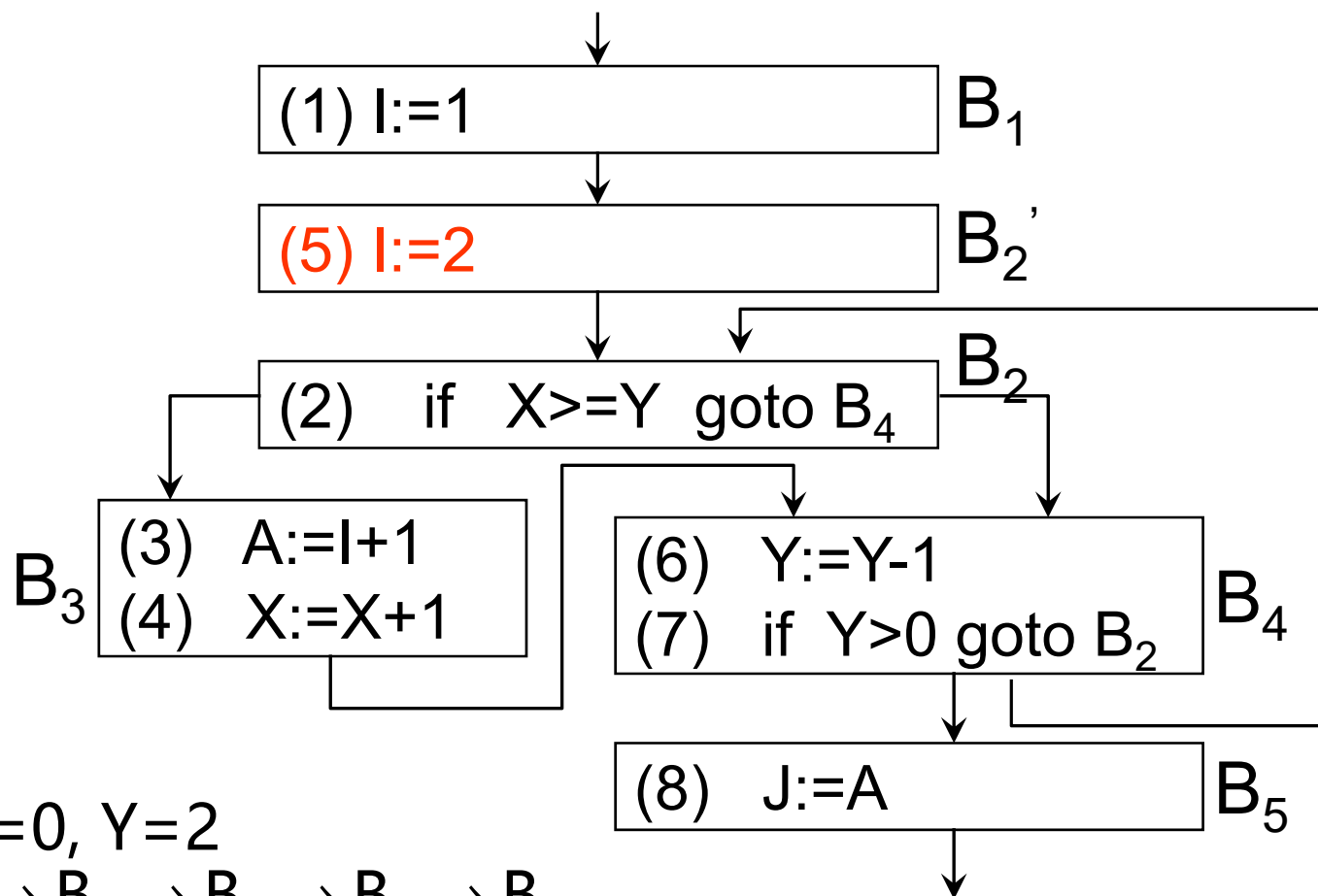


考虑: $X=0, Y=2$

$B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_2 \rightarrow B_4 \rightarrow B_5$

$A=2, J=2$

外提的条件



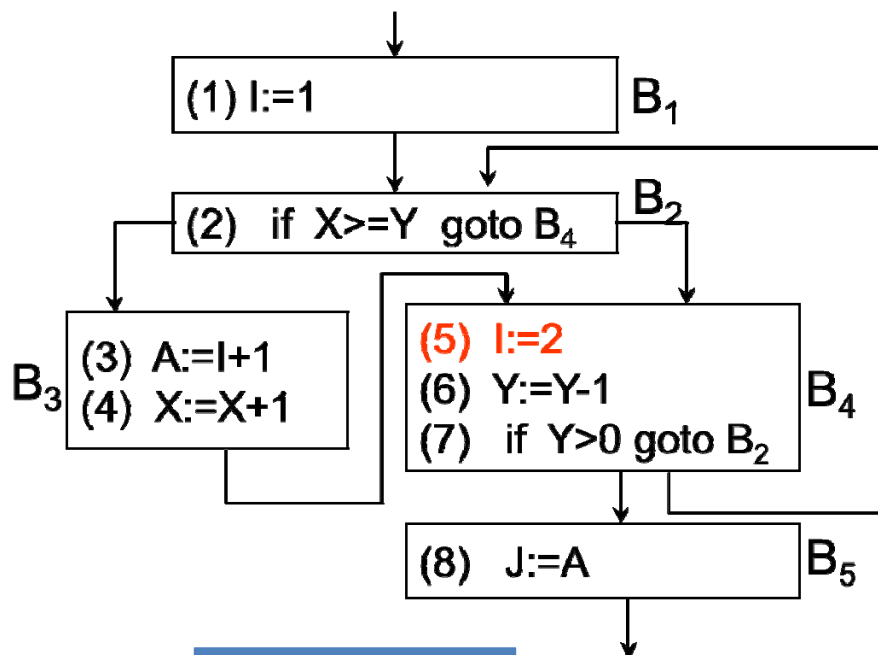
考虑: $X=0, Y=2$

$B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_2 \rightarrow B_4 \rightarrow B_5$

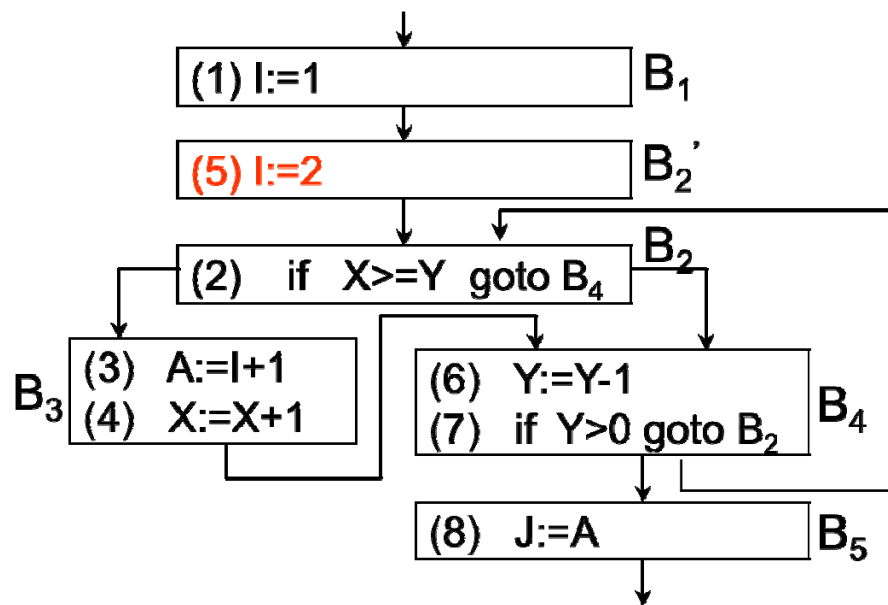
$A=3, J=3$

外提的条件

- ▶ 循环中所有A的引用点只有S中的A的定值才能到达



A=2, J=2



A=3, J=3

外提的条件

► 四元式 $S(A:=B \text{ OP } C)$ 外提的条件

- (i) S所在的结点是L所有出口结点的必经结点;
- (ii) A在L中其他地方未再定值;
- (iii) L中所有A的引用点只有S中的A的定值才能到达。

或者

- (i) A在离开L后不再是活跃的;
- (ii) A在L中其他地方未再定值;
- (iii) L中所有A的引用点只有S中的A的定值才能到达。

代码外提算法

1. 求出L的所有不变运算
2. 对每个不变运算 $s: A := B \text{ op } C$ 或 $A := \text{op } B$ 或 $A := B$ 检查是否满足条件(1)或(2)

条件(1)

- (i) S所在的结点是L所有出口结点的必经结点;
- (ii) A在L中其他地方未再定值;
- (iii) L中所有A的引用点只有S中的A的定值才能到达。

条件(2)

- (i) A在离开L后不再是活跃的;
- (ii) A在L中其他地方未再定值;
- (iii) L中所有A的引用点只有S中的A的定值才能到达。

3. 按步骤1所找出不变运算的次序, 依次把符合步骤2的条件(1)或(2)的不变运算s外提到L的前置结点中。但是, 如果s的运算对象(B或C)是在L中定值的, 那么, 只有当这些定值四元式都已外提到前置结点中时, 才能把s也外提到前置结点中。

编译原理

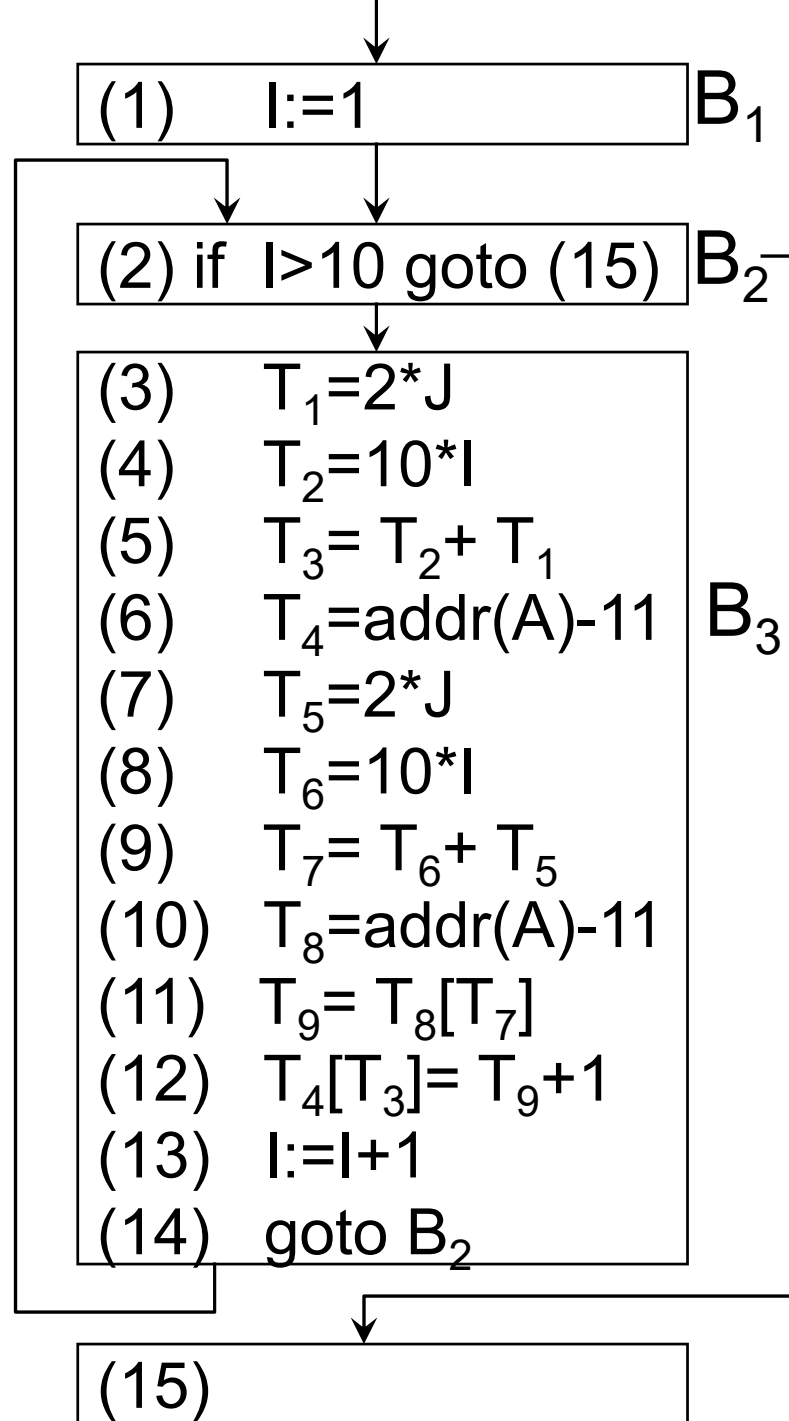
强度消弱

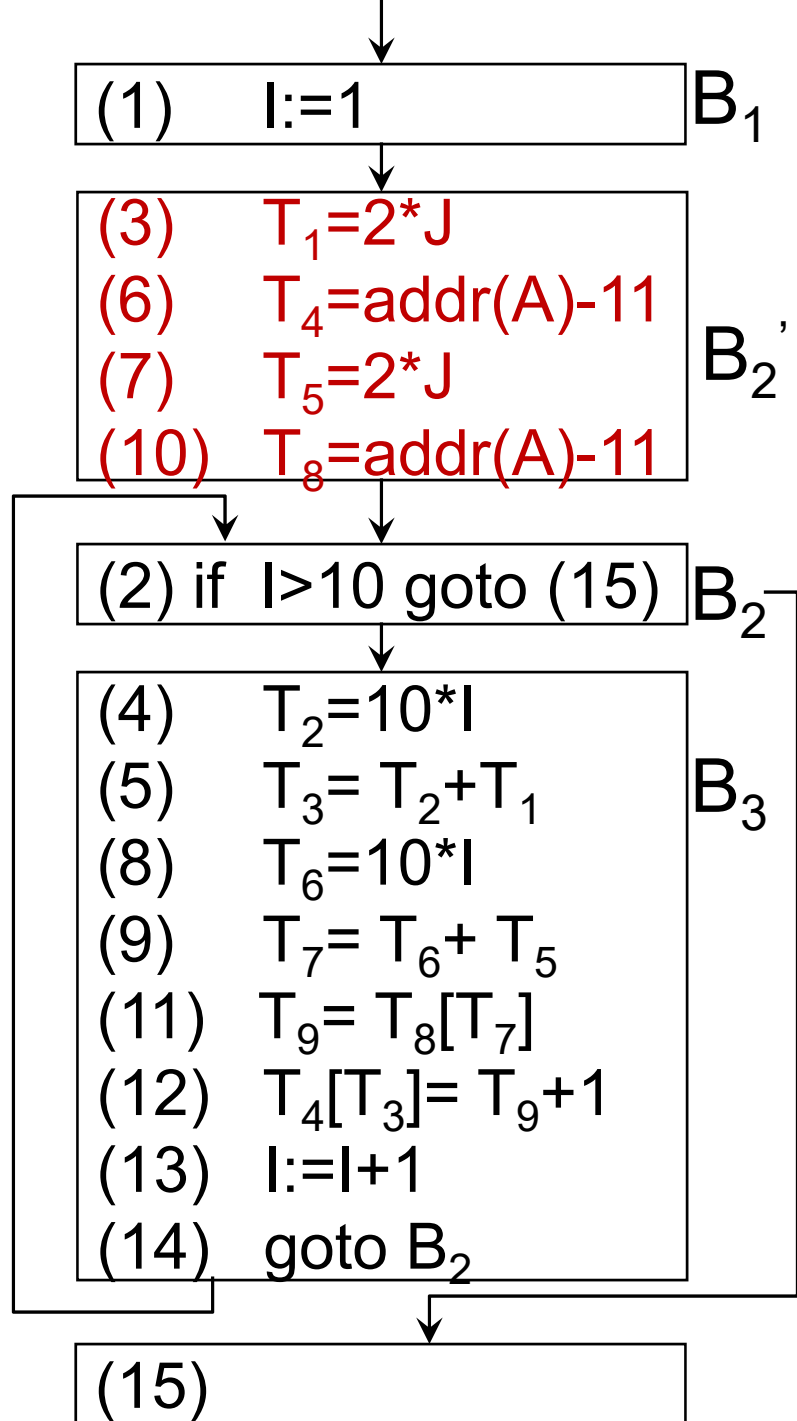
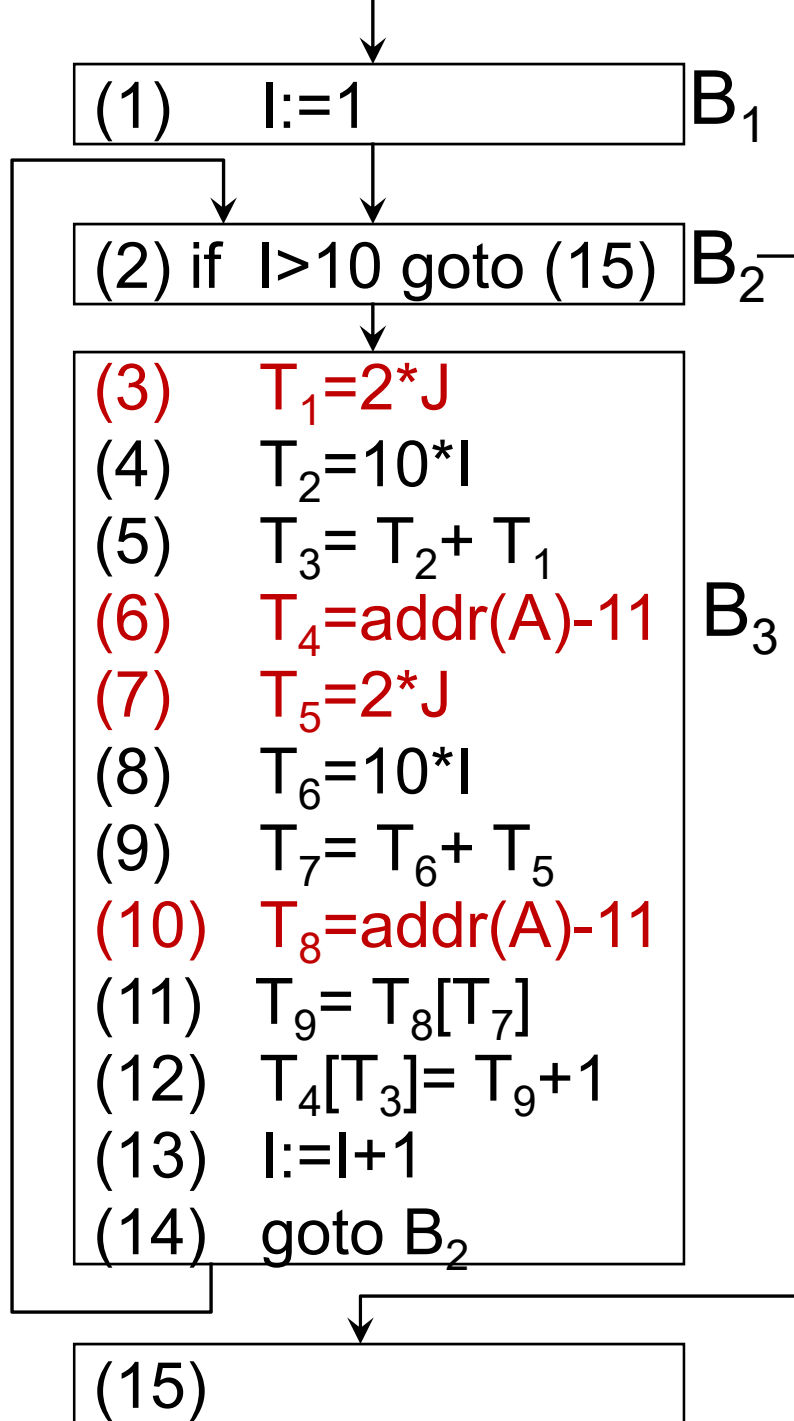
强度消弱

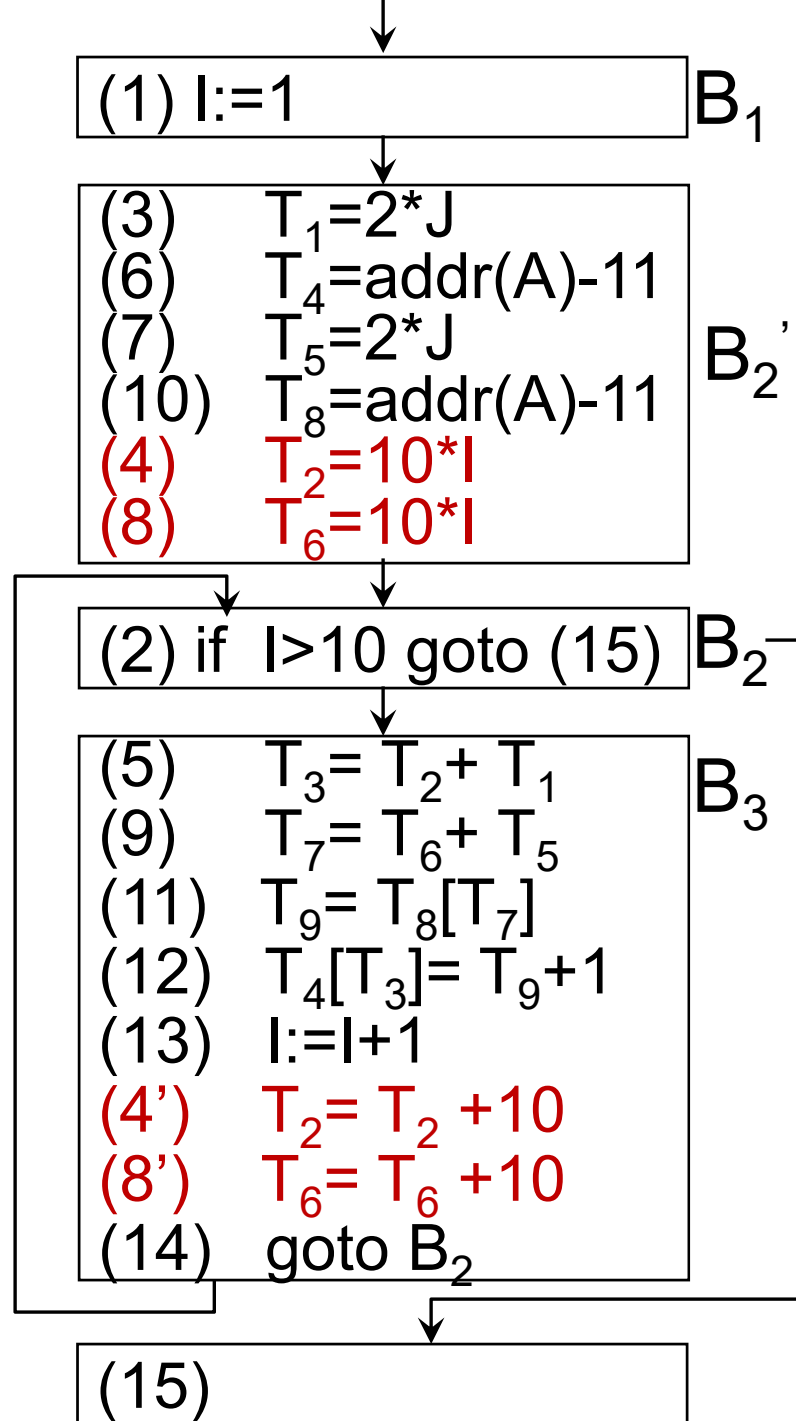
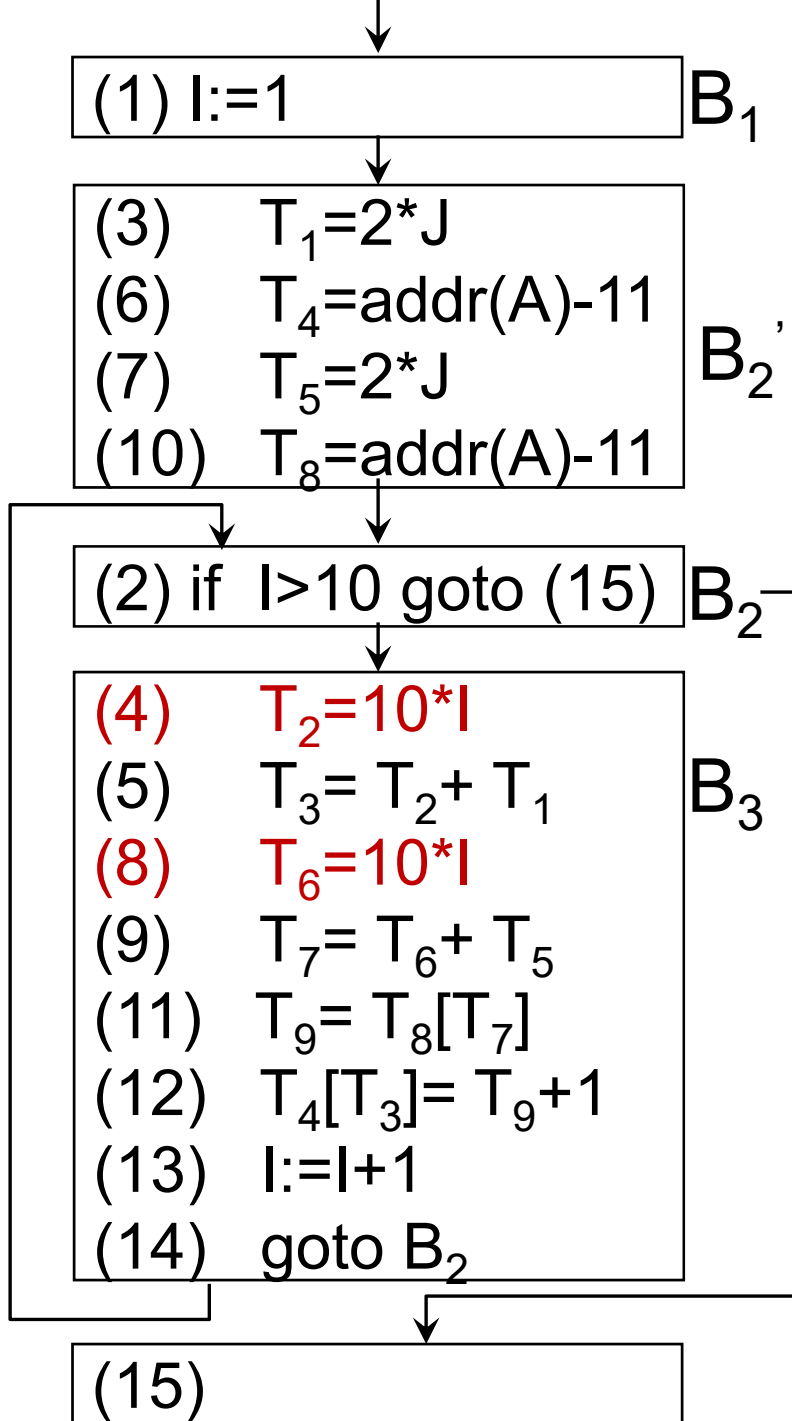
- ▶ 把程序中执行时间较长的运算转换为执行时间较短的运算

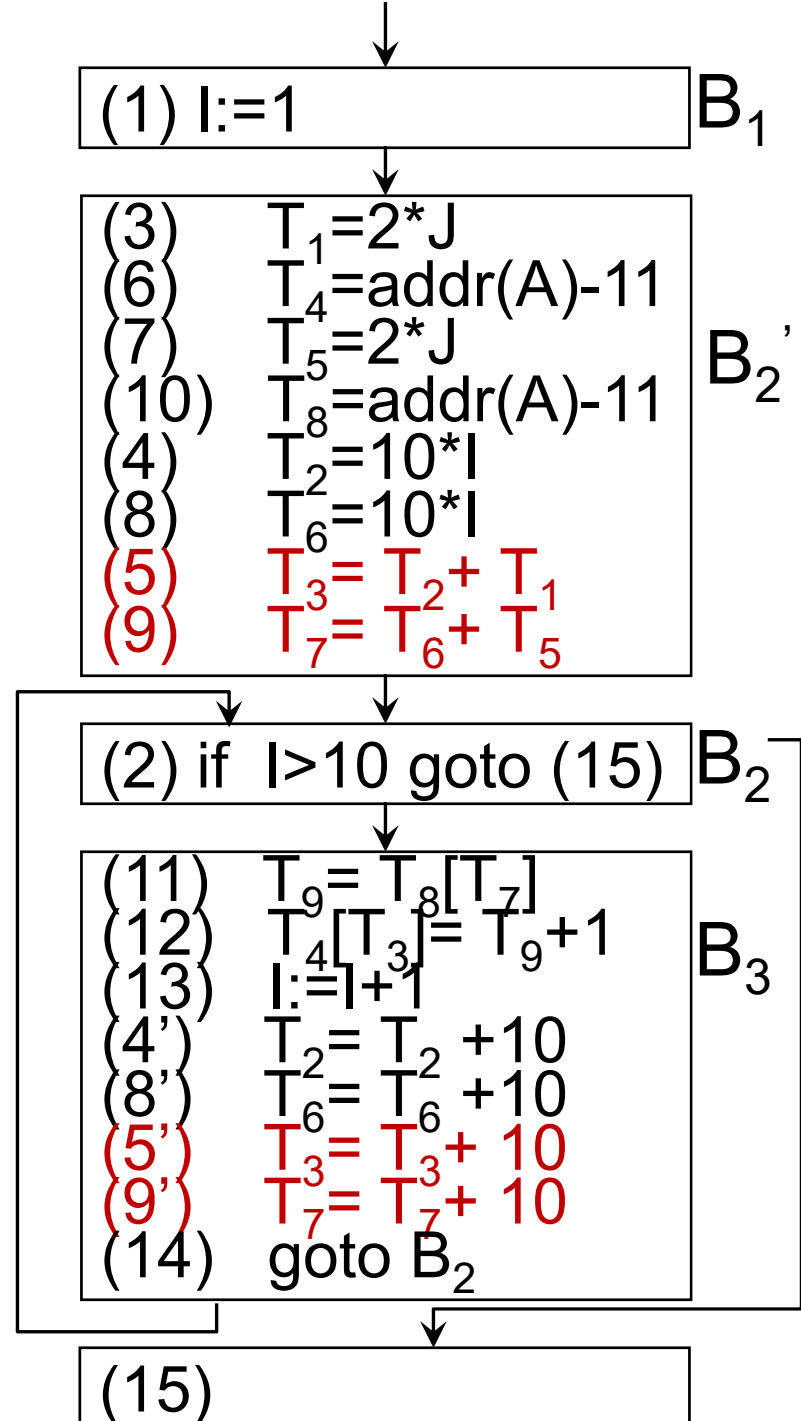
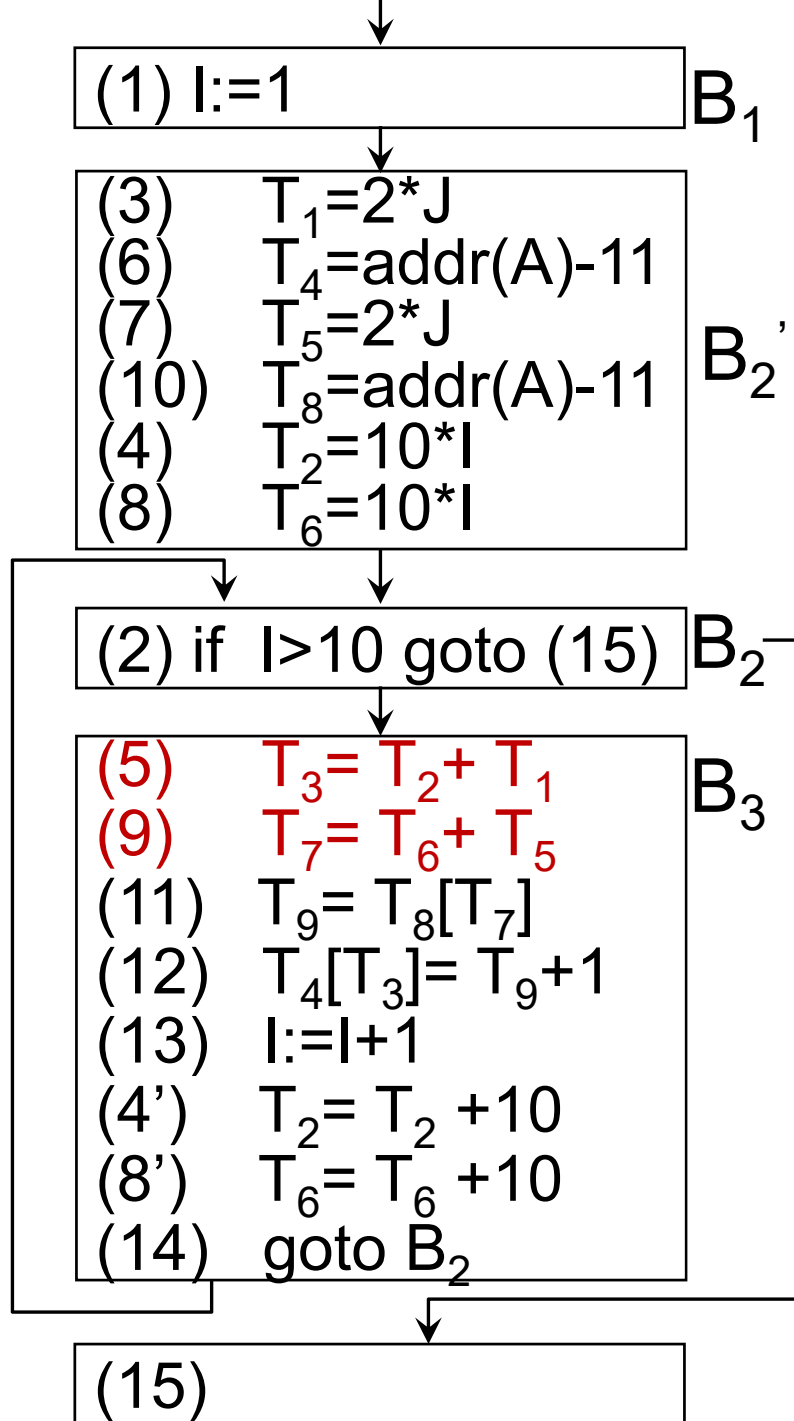
循环优化示例

```
for l:=1 to 10 do  
  A[l, 2*J] := A[l, 2*J] + 1
```







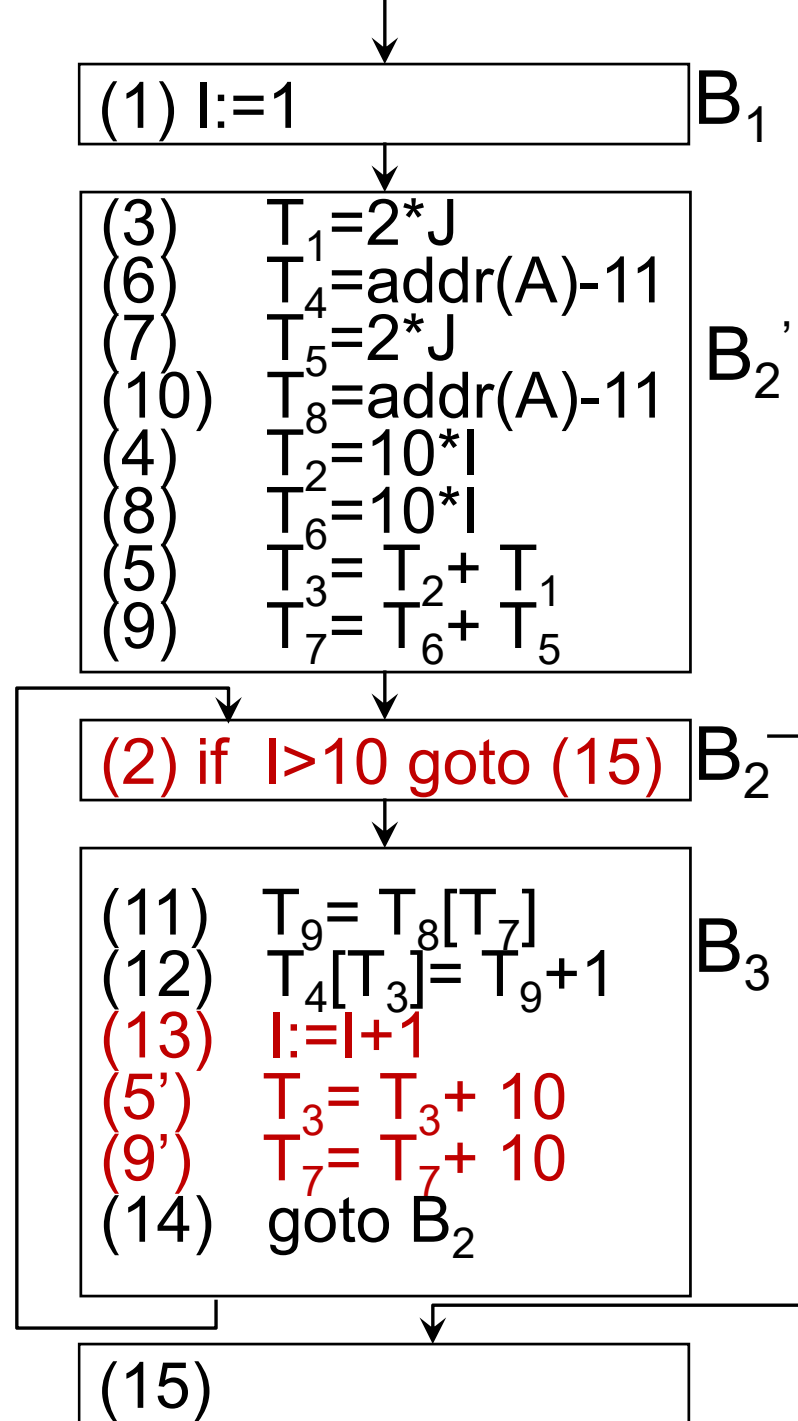
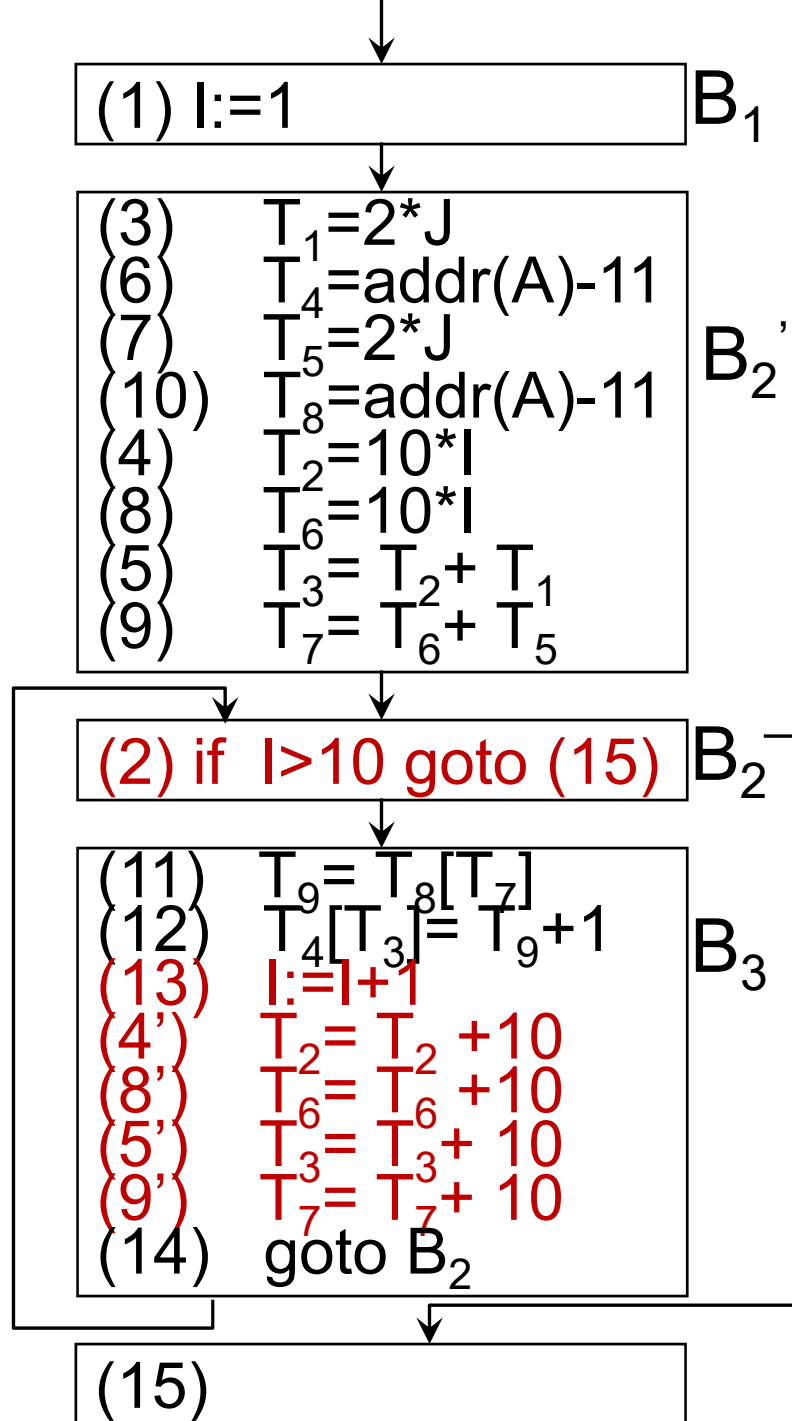


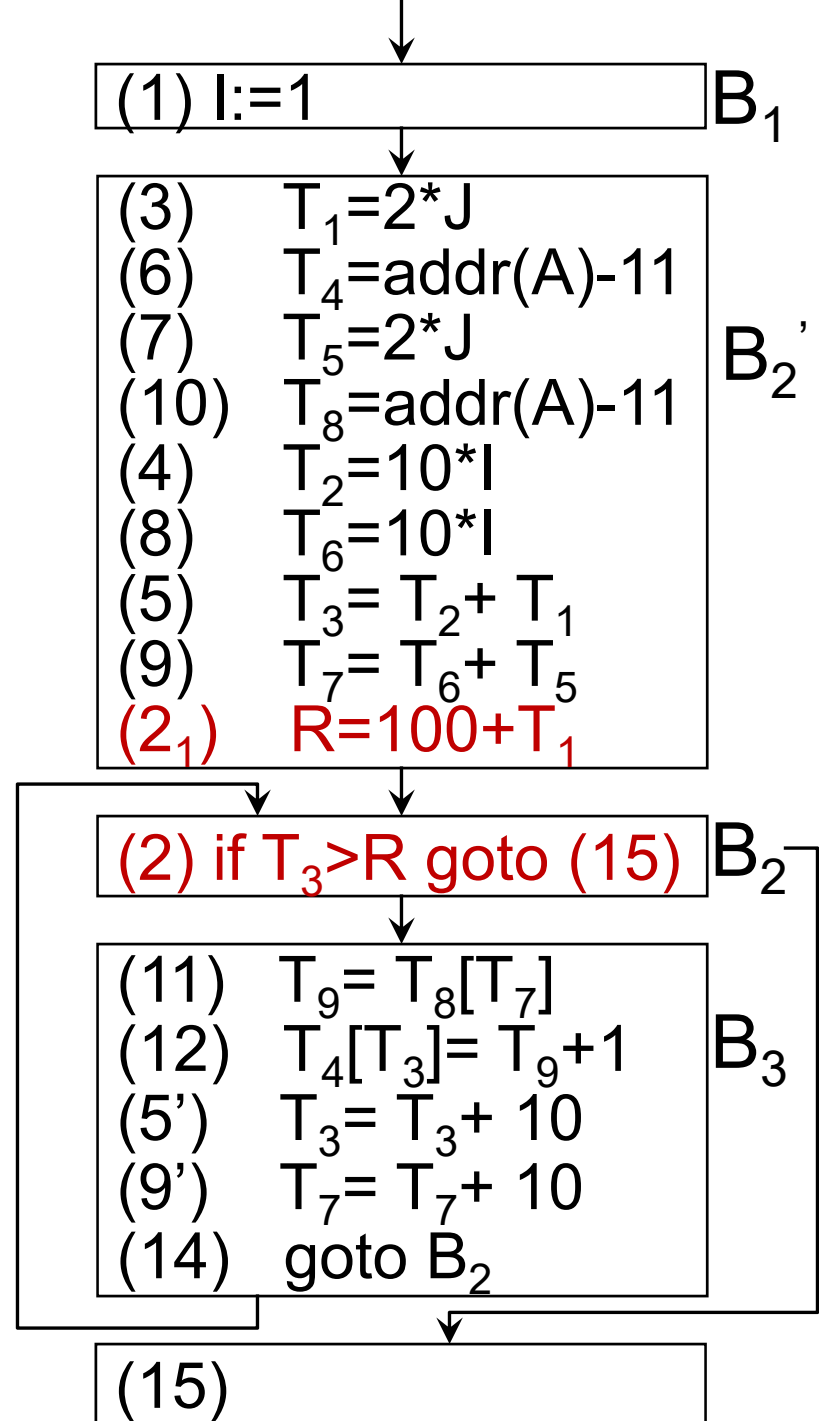
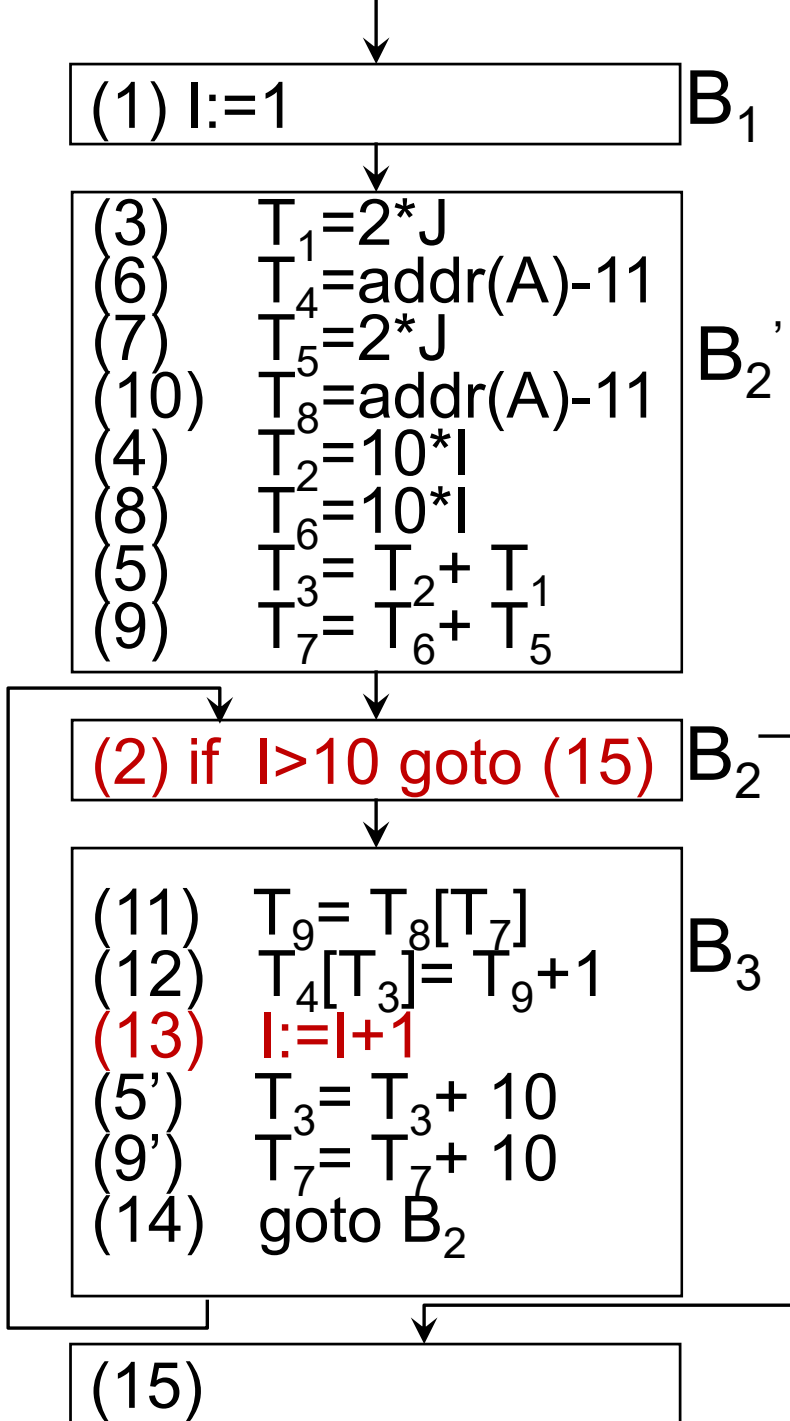
强度消弱

- ▶ 强度消弱通常是针对循环控制变量有线性关系的变量赋值进行
- ▶ 经过强度消弱后，循环中可能出现一些新的无用赋值
- ▶ 对于消弱下标变量地址计算的强度非常有效

删除归纳变量

- ▶ 如果循环中对变量 I 只有唯一的形如 $I := I \pm C$ 的赋值，且其中 C 为循环不变量，则称 I 为循环中的基本归纳变量
- ▶ 如果 I 是循环中一基本归纳变量， J 在循环中的定值总是可化归为 I 的同一线性函数，也即 $J = C_1 * I \pm C_2$ ，其中 C_1 和 C_2 都是循环不变量，则称 J 是归纳变量，并称它与 I 同族。基本归纳变量也是归纳变量。





删除归纳变量

- ▶ 删除归纳变量在强度削弱以后进行
- ▶ 强度削弱和删除归纳变量的统一算法框架

1. 利用循环不变运算信息，找出循环中所有基本归纳变量。
2. 找出所有其它归纳变量A，并找出A与已知基本归纳变量X的同族线性函数关系 $F_A(X)$ 。
3. 对2中找出的每一归纳变量A，进行强度削弱。
4. 删除对归纳变量的无用赋值。
5. 删除基本归纳变量。如果基本归纳变量B在循环出口之后不是活跃的，并且在循环中，除在其自身的递归赋值中被引用外，只在形如 **if B rop Y goto L** 的语句中被引用，则可选取一与B同族的归纳变量M来替换B进行条件控制。最后删除循环中对B的递归赋值的代码。

小结

- ▶ 代码外提
 - ▶ 循环不变运算，外提条件
- ▶ 强度消弱
- ▶ 删除归纳变量