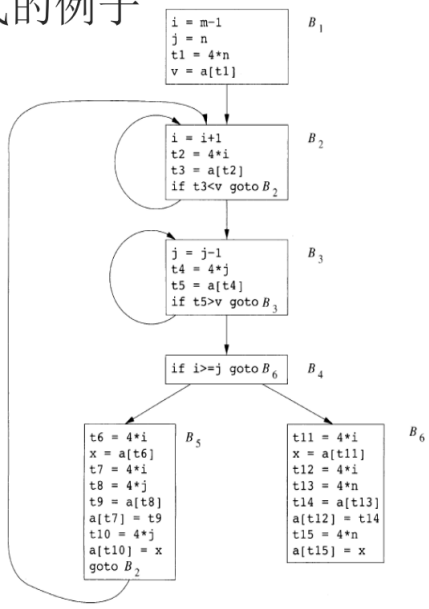
第9章复习笔记

消除公共子表达式

全局公共子表达式的例子

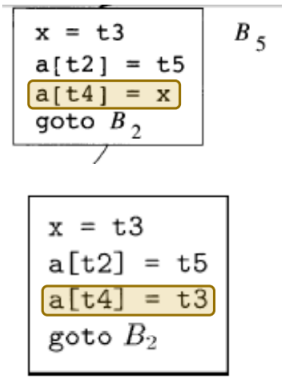
- 右图
  - 在B_2、B_3中计算了4*i和4*j
  - 到达B_5之前必然经过B_2、B_3
  - t2、t4在赋值之后没有被改变过，因此B_5中可直接使用它们
  - t4在替换t8之后，B_5：a[t8]和B_3：a[t4]又相同
- 同样：
  - B_5中赋给x的值和B_2中赋给t3的值相同
  - B_6中的a[t13]和B_1中的a[t1]不同，因为B_5中可能改变a的值



复制传播

复制传播的例子

- 右图显示了对B_5进行复制传播处理的情况
  - 可能消除所有对x的使用



死代码消除

- 如果一个变量在某个程序点上的值可能会在之后被使用，那么这个变量在这个点上活跃；否则这个变量就是死的，此时对这个变量的赋值就是没有用的死代码
- 死代码多半是因为前面的优化而形成的
- 比如，B_5中的x=t3就是死代码
- 消除后得到

x=t3
a[t2] = t5
a[t4] = t3
goto B_2

→

a[t2] = t5
a[t4] = t3
goto B_2

代码移动

- 循环不变表达式
  - 循环的同一次运行的不同迭代中，表达式的值不变
- 把循环不变表达式移动到循环入口之前计算可以提高效率
  - 循环入口：进入循环的跳转都以这个入口为目标
- while(i <= limit-2) …
  - 如果循环体不改变limit的值，可以在循环外计算limit - 2

```
t=limit-2
while(i<= t) …
```

归纳变量和强度消减

归纳变： 每次对x的赋值都使得x的值增加c，那么x就是归纳变量

把对x的赋值改成增量操作，可消减计算强度，提高效率

如果两个归纳变量步调一 致，还可以删除其中的某 一个

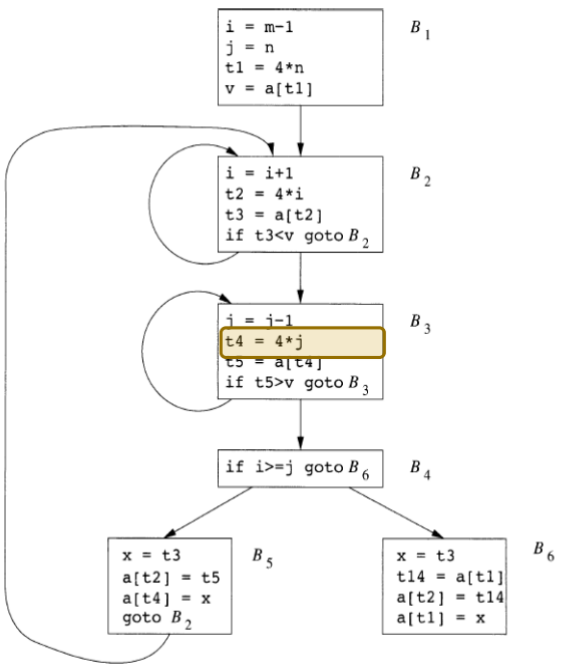


图 9-5 经过公共子表达式消除之后的 B_5 和 B_6

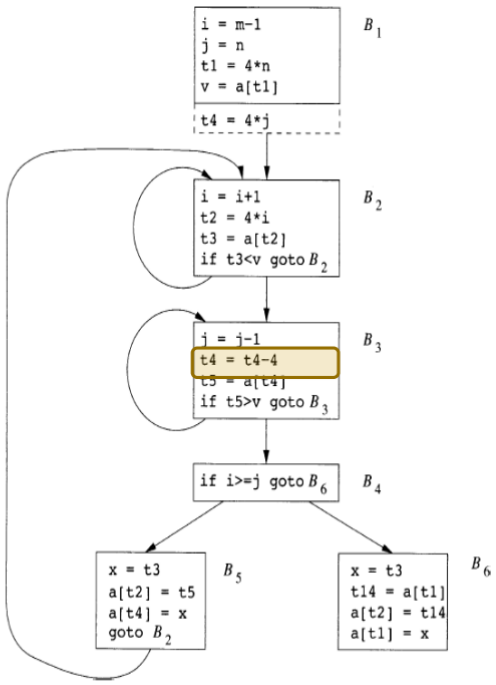


图 9-8 对基本块 B_3 中的 $4*j$ 应用强度消减优化

继续优化

- 对t2强度消减
- B_4中对i和j的测试可以替换为对t2, t4的测试

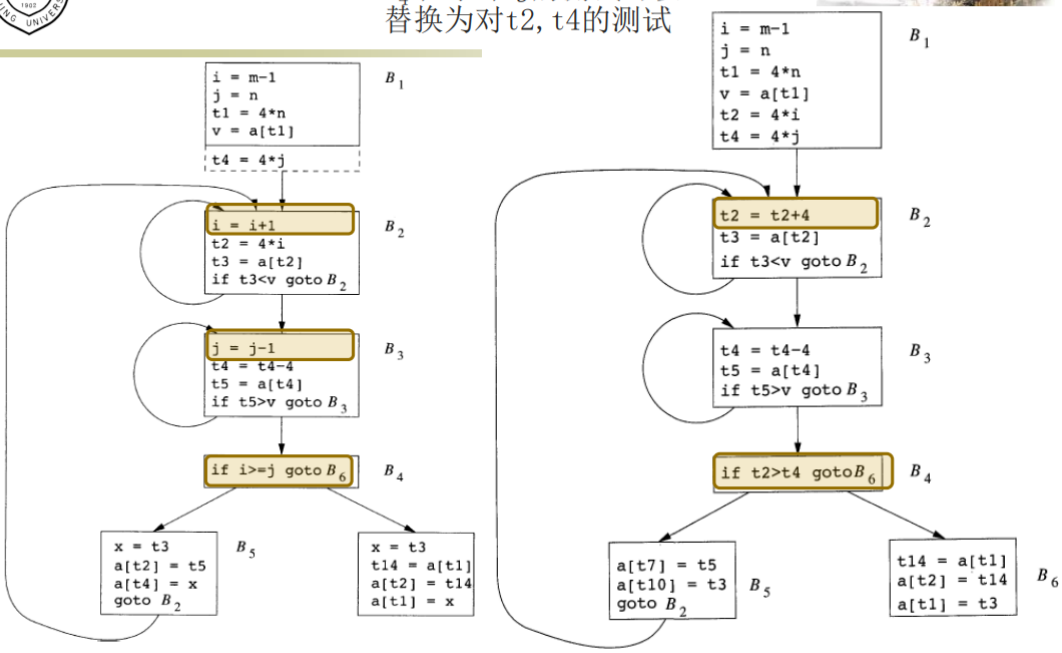


图 9-8　对基本块 B_3 中的 $4*j$ 应用强度消减优化

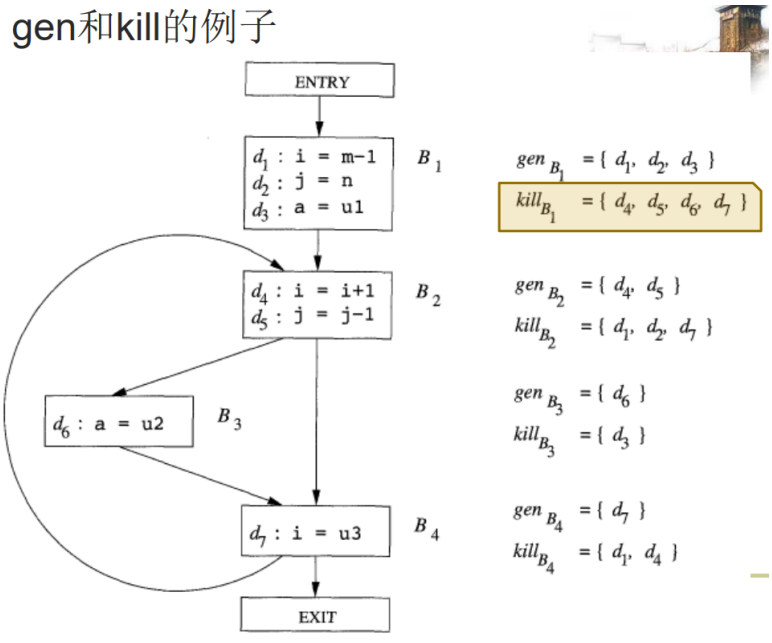
数据流分析

作业里有题目，但是例题里没有。

看看作业题和课件即可，这类题比较难做，应该不会考。

如果考的话，我觉得不会考迭代的过程，最多求求以下几个函数。

gen和kill



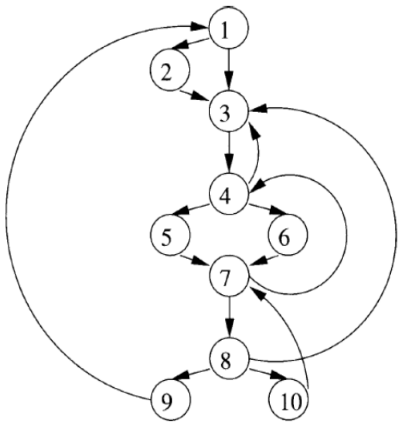
def和use

- use_B，可能在B中先于定值被使用（GEN）
- def_B，在B中一定先于使用被定值（KILL）

流图中的循环

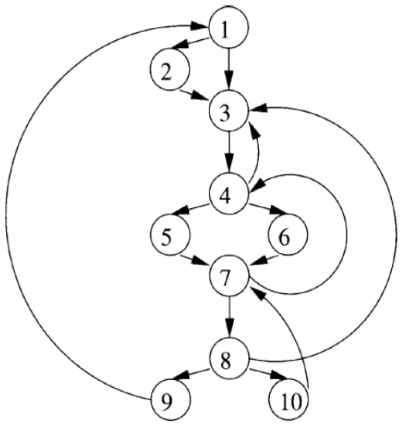
支配结点

- 如果每一条从入口结点到达n的路径都经过d，我们就说d支配(dominate)n，记为d dom n
- 右图：
  - 2只支配自己
  - 3支配除了1，2之外的其它所有结点
  - 4支配1、2、3之外的其它结点
  - 5、6只支配自身
  - 7支配7,8,9,10
  - 8支配8,9,10
  - 9，10只支配自身



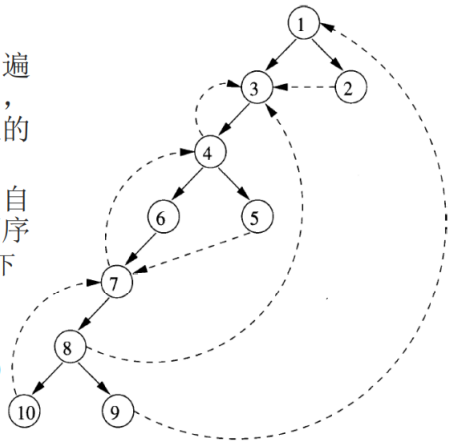
支配节点树

- 支配结点树可以表示支配关系
  - 根结点：入口结点
  - 每个结点d支配且只支配树中的后代结点
- 直接支配结点
  - 从入口结点到达n的任何路径（不含n）中，它是路径中最后一个支配n的结点
  - 前面的例子：1直接支配3，3直接支配4
  - n的直接支配结点m具有如下性质：如果d dom n，那么d dom m



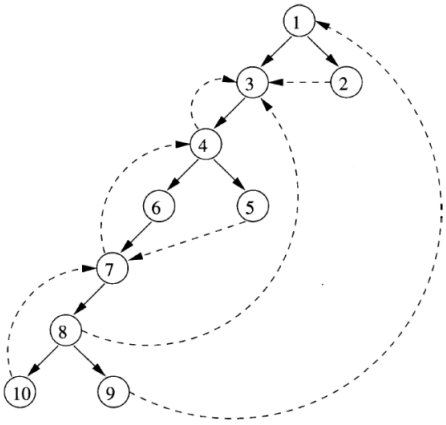
深度优先排序

- 深度优先排序
  - 先访问一个结点，然后遍历该结点的最右子结点，再遍历这个子结点左边的子结点，依此类推
  - 具体访问时，我们可以自己设定各个子结点的顺序
    - 哪个是最右的，哪个是下一个子结点等
- 例子见右图：
  - 1,2,3,4,5,6,7,8,9,10



图的深度

- 一个流图，相对于一棵DFST的深度
  - 各条无环路径上后退边数中的最大值
  - 这个深度不会大于直观上所说的流图中的循环嵌套深度。
- 对于可归约的流图，我们可以使用“回边”来定义，而且可以说是“流图的深度”
- 右边的流图深度为3
  - 10→7→4→3



回边和可归约性

- 前进边
  - 从结点m到达m在DFST树中的一个真后代的边
  - DFST中的所有边都是前进边
- 后退边
  - 从m到达m在DFST树中的某个祖先的边
- 交叉边
  - 边的src和dest都不是对方的祖先
  - 一个结点的子结点按照它们被加入到树中的顺序从左到右排列，那么所有的交叉边都是从右到左的
- 回边
  - 边a→b，头b支配了尾a
  - 每条回边都是后退边，但不是所有后退边都是回边
- 如果一个流图的任何优先生成树中的所有后退边都是回边，那么该流图就是可归约的
  - 可归约流图的DFST的后退边集合就是回边集合
  - 不可归约流图的DFST中可能有一些后退边不是回边，但是所有的回边仍然都是后退边
- 实践中出现的流图基本都是可归约的

自然循环

- 自然循环的性质
  - 有一个唯一的入口结点（循环头 header）。这个结点支配循环中的所有结点
  - 必然存在进入循环头的回边
- 自然循环的定义
  - 给定回边n→d的自然循环是d加上不经过d就能够到达n的结点的集合
  - d是这个循环的头

d加上不经过d就能够到达n的结点的集合，所以只需要检查其他节点是否有一条不经过d就可以到达n的路径即可；同时，当循环头（d）相同时，自然循环的节点也是相同的。