



## 第8章 代码优化

---

### 教学目标

1. 要求明确代码优化的目的和分类
2. 掌握基本块的划分方法，基本块内的三种优化方法
3. 掌握程序流图的构造方法，循环优化的三种优化方法

# 教学内容

- 8.1 局部优化
- 8.2 循环优化

## 8.1 代码优化

---

目的：提高目标代码运行效率

时间效率（减少运行时间）

空间效率（减少内存容量）

原则：不能改变原有程序语义

✓ 优化实质上是对代码进行等价变换，  
变换后代码结构不同但运行结果相同。

# 代码优化分类

☆ 从优化的层次，与机器是否有关

□与机器无关：与目标机无关，在中间代码上优化

□与机器有关：充分利用系统资源，（指令系统，寄存器）

★ 从优化涉及的范围

局部优化：在基本块内进行优化。

循环优化：对循环语句所生成的中间代码进行优化。

全局优化：跨越多个基本块的全局范围内的优化，复杂。

## 8.1 局部优化

---

✓ 在一个**基本块上**进行的优化

**基本块：**程序中一个**顺序执行**的语句序列，即一个程序段，它只有**一个入口**和**一个出口**，入口是第一条语句，出口是最后一条语句。

## 基本块划分方法

### 1.确定各个基本块的的入口语句(基本块的第一个语句)

- ① 语句序列的**第一个语句**是入口语句；
- ② 能由**条件转移语句或无条件转移语句**转到的语句是入口语句；
- ③ **紧跟在**条件转移语句或无条件转移语句后面的语句是入口语句。

## 基本块划分方法

2. 对于每个入口语句，构造其所属的基本块。它是以下三种情况之一：

- ① 该入口语句到**下一条入口语句**（不包括该入口语句）之间的语句序列；
- ② 该入口语句到**一条转移语句**（包括该转移语句）之间的语句序列；
- ③ 该入口语句到**一条停语句**（包括该停语句）之间的语句序列；

3. 凡**未被纳入**某一基本块的语句，都是程序中控制流程无法到达的语句，从而也是不会被执行到的语句，将其**删除**。



### 【例8.1】

(1) read X

(2) read Y

(3)  $R = X \bmod Y$

(4) if  $R = 0$  goto (8)

(5)  $X = Y$

(6)  $Y = R$

(7) goto (3)

(8) write Y

(9) halt

(1) read X

(2) read Y

(3)  $R = X \bmod Y$

(4) if  $R = 0$  goto (8)

(5)  $X = Y$

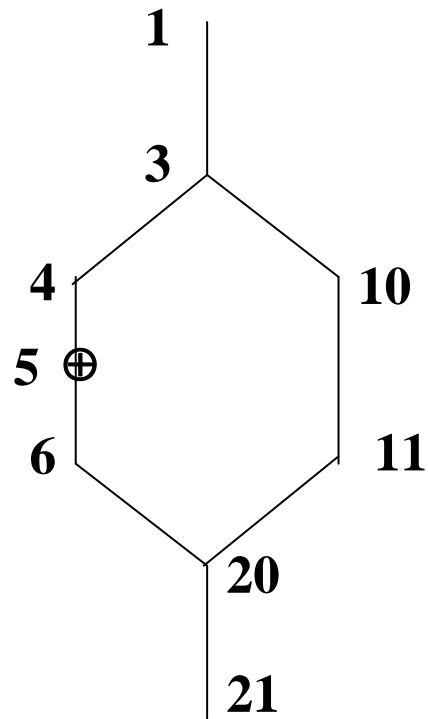
(6)  $Y = R$

(7) goto (3)

(8) write Y

(9) halt

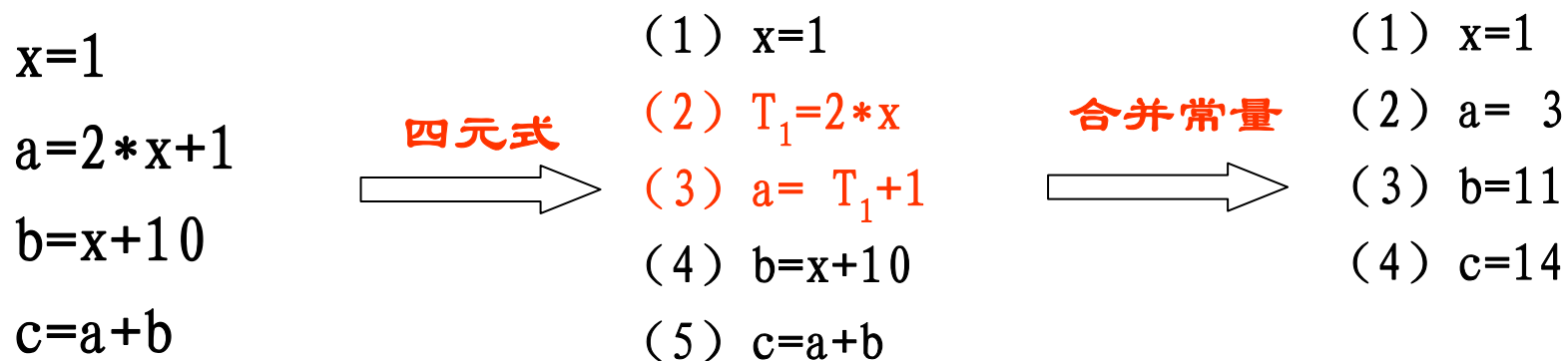
练习:



- |                            |   |            |
|----------------------------|---|------------|
| 1. <b>FACTOR=2</b>         | } | <b>基本块</b> |
| 2. <b>EXP 1=...</b>        |   |            |
| 3. <b>IF ( )GOTO 10</b>    |   |            |
| 4. <b>BASE=2.0</b>         | } | <b>基本块</b> |
| 5. <b>FACTOR=FACTOR**2</b> |   |            |
| 6. <b>GOTO 20</b>          |   |            |
| 10. <b>BASE=...</b>        | } | <b>基本块</b> |
| 11. <b>FACTOR...</b>       |   |            |
| 20. <b>Q=</b>              | } | <b>基本块</b> |
| 21. <b>RETURN</b>          |   |            |

## 基本块内的优化方法

### ★ 1. 合并常量：编译时就计算表达式中的常量运算



## 基本块内的优化方法

### ★ 2. 删除公共子表达式

$x = a * b + c$

$y = a * b + x$

$z = a * b + y$

四元式



(1)  $T_1 = a * b$

(2)  $x = T_1 + c$

(3)  $T_2 = a * b$

(4)  $y = T_2 + x$

(5)  $T_3 = a * b$

(6)  $z = T_3 + y$

删除公共  
子表达式



(1)  $T_1 = a * b$

(2)  $x = T_1 + c$

(3)  $y = T_1 + x$

(4)  $z = T_1 + y$

## 基本块内的优化方法

### ☆ 3. 删除无用赋值

(1)  $x=1$

(2)  $x=10+a$

(3)  $y=x+b$



(1)  $x=10+a$

(2)  $y=x+b$

## 8.2 循环优化

---

✓ 一个**程序流图**是具有唯一首结点的有向图。

**首结点**：从它开始到有向图中任何结点都有一条通路的结点。

**程序流图（控制流程图或流图）**：

**三元组** $G=(N, E, n_0)$

**N**：结点（基本块）集

**E**：边集

$n_0$ ：首结点。

(1) read X

(2) read Y

(3)  $R = X \bmod Y$

(4) if  $R = 0$  goto (8)

(5)  $X = Y$

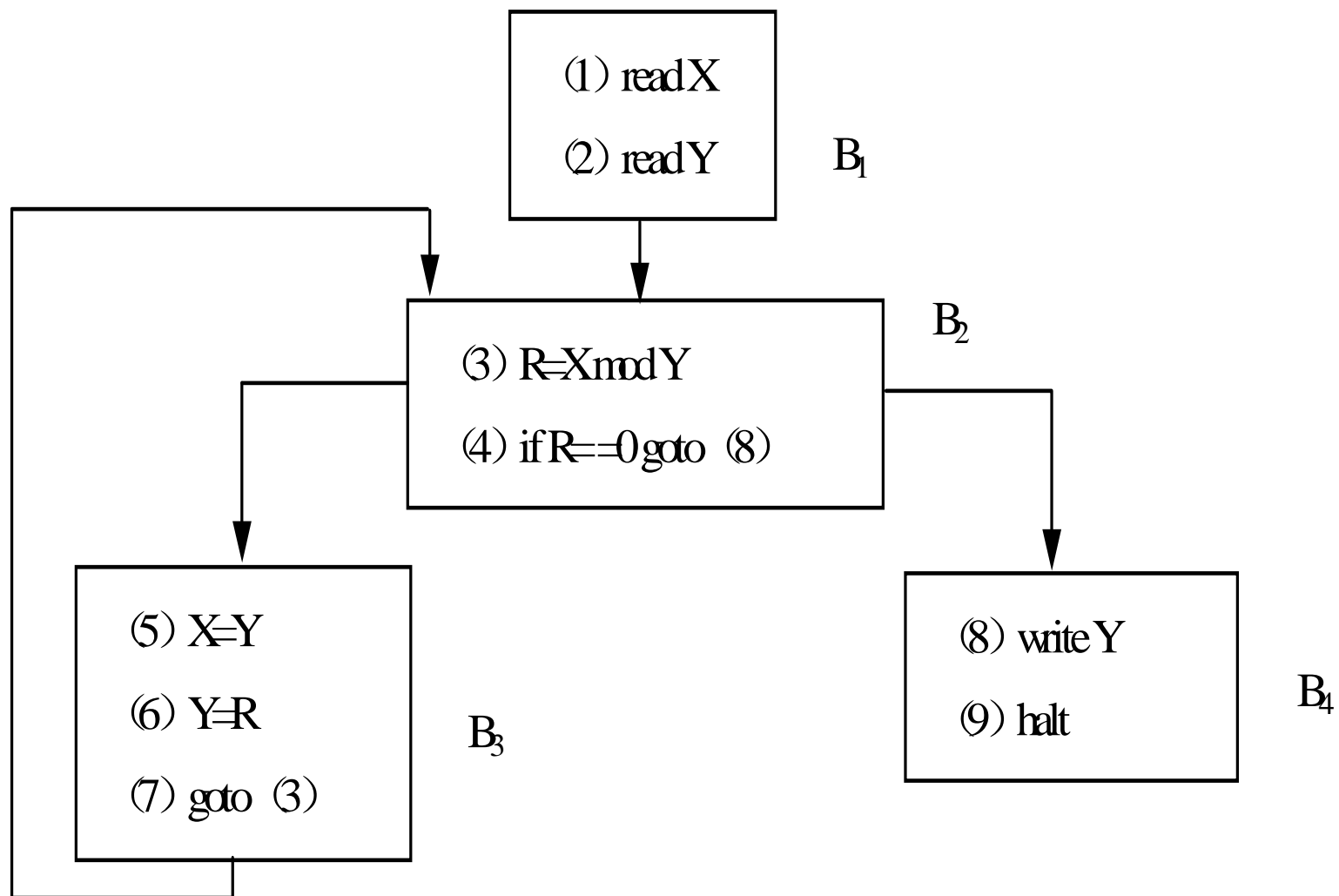
(6)  $Y = R$

(7) goto (3)

(8) write Y

(9) halt







## 循环优化方法

- ★ 1. **代码外提**：将循环中的**不变运算**提到循环前面，  
不变运算是指其运算结果不受循环影响的表达式。

循环中的不变运算均可提到循环外 ❌

# 循环优化方法

☆ 2. 强度削弱：把程序中执行时间较长的运算替换为执行时间较短的运算。

如：  $x**2$        $\rightarrow x*x$

$x/5$        $\rightarrow x*0.2$

$x*2, x*4$        $\rightarrow$  左移运算

$x/2, x/4$        $\rightarrow$  右移运算

## 循环优化方法

### ★ 3. 删除归纳变量

➤ 如果循环中变量I只有唯一的形如 $I = I \pm C$ 的赋值，其中C为循环不变量，则称I为**基本归纳变量**。

➤ 如果I为循环中的基本归纳变量，循环中的另一变量J可以表示为I的线性函数形式：

$$J = C1 * I \pm C2$$

C1和C2是循环不变量，则称J为循环中与I同族的**归纳变量**。

。

一个基本归纳变量一定是归纳变量。



# 小结

- 明确代码优化的**目的和分类**
- 掌握**基本块的划分方法**，基本块内的三种优化方法：**合并常量、删除公共子表达式和删除无用赋值**
- 掌握**程序流图的构造方法**，循环优化的三种优化方法：**代码外提、强度削弱和删除归纳变量**