



School of Computer Science & Technology
Harbin Institute of Technology



第十章 代码优化

重点： 代码优化的任务，局部优化、循环优化、
全局优化的基本方法。

难点： 控制流分析，数据流分析。





第10章 代码优化

10.1 优化的种类

10.2 控制流分析

10.3 数据流分析

10.4 局部优化

10.5 循环优化

10.6 全局优化

10.7 本章小结



第10章 代码优化

- 代码优化就是为了提高目标程序的效率，对程序进行**等价变换**，亦即在保持功能不变的前提下，对源程序进行合理的变换，使得目标代码具有更高的时间效率和/或空间效率。
- 空间效率和时间效率有时是矛盾的，有时不能兼顾。



10.1 优化的种类

■ 机器相关性

- 机器相关优化：寄存器优化，多处理器优化，特殊指令优化，无用指令消除等。
- 机器无关优化

■ 优化范围

- 局部优化：单个基本块范围内的优化，常量合并优化，公共子表达式删除，计算强度削弱和无用代码删除。
- 全局优化：主要是基于循环的优化：循环不变优化，归纳变量删除，计算强度削减。

■ 优化语言级

- 优化语言级：针对中间代码，针对机器语言。



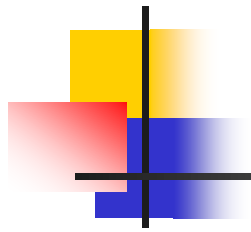
10.1.1 公共子表达式删除

- 如果表达式 E 在某次出现之前已经被计算过，并且 E 中变量的值从那次计算到本次出现之间未被改变过，则 E 的本次出现称为公共子表达式
- 用先前的计算结果替换公共子表达式的本次出现称为公共子表达式删除



10.1.2 复制传播

- 形如 $f := g$ 的赋值语句叫做复制语句
- 优化过程中会大量引入复制
- 复制传播变换的思想是在复制语句 $f := g$ 之后尽可能用 g 代替 f
- 复制传播变换本身并不是优化，但它给其它优化带来机会



10.1.3 无用代码删除

- 无用代码是指计算结果以后不被引用的语句
- 一些优化变换可能会引入无用代码



10.1.3 无用代码删除

■ 无用代码的危害

- **事件：**2012年8月1日纽约股票交易所，Knight Capital Group最新的高速算法在45分钟内自动完成了数百万次未经授权股票交易，导致股市发生剧烈震荡
- **原因：**公司在服务器上部署更新，工程师手工更新了服务器，其中一次部署失败使旧版本保留运行，重用了一个旧标识，但是该停用标识已更改了用途，用于表示另一个意义
- **虽然该标识在过去的八年都没有被使用，旧版本代码依然依赖于这个旧标，直到无用状态的代码由于标识值的更新而被唤醒了**



10.1.4 代码外提

- **结果独立于循环执行次数的表达式称为循环不变计算。**
- **如果将循环不变计算从循环中移出到循环的前面，将会减少循环执行的代码总数，大大提高代码的执行效率。这种与循环有关的优化方法称为代码外提。**



10.1.5 强度削弱

- 实现同样的运算可以有多种方式。用计算较快的运算代替较慢的运算。

- x^2 变成 $x * x$ 。

- $2 * x$ 或 $2.0 * x$ 变成 $x + x$

- $x/2$ 变成 $x * 0.5$

- $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ 变成
 $((\dots(a_n x + a_{n-1})x + a_{n-2})\dots)x + a_1)x + a_0$



10.2 控制流分析

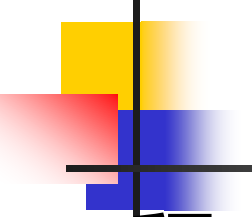
- 为了对程序进行优化，尤其是对**循环**进行优化，必须首先分析程序中的控制流程，以便找出程序中的循环结构，这是控制流分析的主要任务。
- 为此，首先需要将程序划分为基本块集合并转换成流图，然后再从流图中找出循环。



10.2.1 基本块

- **基本块(basic block)是一个连续的语句序列，控制流从它的开始进入，并从它的末尾离开，中间不存在中断或分支(末尾除外)。下面的三地址码序列就形成了一个基本块：**

- $t1 := a * a$
- $t2 := a * b$
- $t3 := 2 * t2$
- $t4 := t1 + t3$
- $t5 := b * b$
- $t6 := t4 + t5$



10.2.2 流图

- 程序的控制流信息可以用流图表示，流图是一个节点为基本块的有向图。
- 以程序的第一个语句作为入口语句的节点称为**初始节点**。
- 如果在某个执行序列中 B_2 跟随在 B_1 之后，则从 B_1 到 B_2 有一条有向边。
- 如果从 B_1 的最后一条语句有条件或无条件转移到 B_2 的第一个语句；或者按程序正文的次序 B_2 紧跟在 B_1 之后，并且 B_1 不是结束于无条件转移，则称 B_1 是 B_2 的**前驱**，而 B_2 是 B_1 的**后继**。



10.4 局部优化

- **基本块的功能实际上就是计算一组表达式，这些表达式是在基本块出口活跃的变量的值。如果两个基本块计算一组同样的表达式，则称它们是等价的。**
- **可以对基本块应用很多变换而不改变它所计算的表达式集合，许多这样的变换对改进最终由某基本块生成的代码的质量很有用。**
- **利用基本块的dag表示可以实现一些常用的对基本块的变换。**



利用dag进行的基本块变换

- (1) 局部公共子表达式删除。**
- (2) 无用代码删除。**
- (3) 交换两个独立的相邻语句的次序，以便减少某个临时值需要保存在寄存器中的时间。**
- (4) 使用代数规则重新排列三地址码的运算对象的顺序，以便简化计算过程。**



10.5 循环优化

- **循环不变计算的检测**
- **代码外提**
- **归纳变量删除和强度削弱**



10.6 全局优化

- 全局公共子表达式的删除
- 复制传播



本章小结

- **代码优化就是对程序进行等价变换，以提高目标程序的效率，通常只对中间代码进行优化。通常包括控制流分析、数据流分析和变换三部分。**
- **以程序的基本块为基础，基本块内的优化叫局部优化，跨基本块的优化为全局优化，循环优化是针对循环进行的优化，是全局优化的一部分。**
- **公共子表达式的删除、复制传播、无用代码删除、代码外提、强度削弱和归纳变量删除等都是常用的针对局部或者全局的代码优化方法。**