

Lab4.2 实验报告

阿非提

PB20111633

实验要求

- 阅读论文理解算法实现优化代码的过程
- 使用论文中的算法实现
 - 对冗余指令的检测与消除
 - 对于 call 指令冗余的检测与消除
 - 常量传播
 - value phi function 的冗余的检测与消除

实验难点

- 论文中的算法在具体实现中仍有很多不清楚，有疑问的地方。如论文中没有关于算法收敛的说明。需要学生自己动手实验找出相对方法。
- 算法收敛问题主要在不同迭代中的两个等价的等价类判别。

实验设计

- detectEquivalences :
 - 多所有全局变量，函数参数创建等价类。这些等价类可放置暂时的变量temp中，以便在后续传播到每个pout中
 - 对函数第一块（entry block）进行一次迭代初始化。
 - 对函数中的每个基本块进行遍历，对基本块中的每个非Phi指令调用transferFunction，找出该基本块的后继中所有的Phi，将其转化成copy statement调用transferFunction。
 - 等pout不再变化时停止迭代。
- transferFunction :
 - 忽略优化无关指令（通过调用is_void（）进行判断）
 - 对Phi 指令，将其左值添加到右值所在的等价类中。
 - 对其他指令，调用 valueExpr 和 valueExpr 找出该指令的 value expression 和 phi value，当且仅当在pout找不到相应地等价类时创建新的等价类，否则将其加入的已有的等价类中。
- valueExpr:
 - 找出指令所对应的 value expression
 - 通过指令分类对不同类型的指令区别做处理

- 从pout中找出指令操作数（operand）所在的等价类，取出value number（cc中的index）
- valuePhiFunc :
 - 判断参数ve的类型是否为 $\phi(a, b) + \phi(c, d)$ 的形式，如不是返回nullptr
 - 调用getVN 找出 value expression 为 $a+c$, $b+d$ 的等价类，如找不到递归寻找。
 - 当且仅当找出 v_i, v_j 时，返回 $\phi(v_i, v_j)$ 否则返回nullptr
- getVN
 - 从pout中找出ve所在的等价类，如找到返回其value number（cc中的index），否则返回null
- join
 - 对P1, P2中的每个cc的组和调用 intersect，当intersect 返回值非空时将其加入到P中。
- Intersect
 - 如index_ 为 0 说明为顶元 返回另一个cc即可。
 - 对两个cc的members 做交集运算，当交集为空时返回nullptr，否则继续算法。
 - 由于每个cc中的leader 在创建新的cc时才会赋值，所以可以根据leader判断是否为同一个等价类，当且仅当不是同一个等价类时生成value phi。

实验总结

通过本次实验，我体会到将理论算法具体实现时的不易。