**Decaf PA4实验报告**

电53 魏宇轩 2015011942

**1. 现有框架下DU链求解的实现**

**1.1 DU链求解算法**

将活跃变量求解算法扩展得到DU链求解算法。

**(1) 符号定义**

令B为一个基本块的标号，为一条TAC语句，A为一个变量；

令为的集合，其中是B中某点，且B中在前面没有的定值点；

令为的集合，其中是不属于B的某点，引用了变量且在中被定值；

令为的集合，表示一个基本块入口处的DU链；

令为的集合，表示一个基本块出口处的DU链；

令为的集合，表示一个基本块内所有引用的定值点-变量集合；

令为的集合，表示一个基本块外所有引用的定值点-变量集合；

令为的集合，表示基本块定值的所有变量。

**(2) 基本块的DU链求解算法**

DU链方程同活跃变量方程：

DU链方程同样可以用迭代算法求解。

**(3) TAC语句的DU链求解算法**

从基本块的出口语句开始，

(a) 如果语句是变量的定值点，从DUliveOut中取出所有变量A的引用点作为的DUChain，并从DUliveOut中删除所有A的引用点。如果引用了变量C，将加入DUliveOut;

(b) 如果语句不是任何变量的定值点且引用了变量C，将加入DUliveOut;

转入上一条语句的分析。

**(4) 与的初始化算法**

从基本块B的入口语句开始，如果语句引用了没有在B中被定值的变量，则将加入；

从基本块B的入口语句开始，如果语句引用了变量，则将加入;

从基本块B的入口语句开始，如果语句定值了变量，则将变量A加入dudef中；

计算得到每个基本块的，计算每个基本块的，再根据得到.

**1.2 代码改动**

(1) 修改Temp.java, 为Temp类增加int lastDefBB属性

用lastDefBB属性标记一个变量上一次被定义的基本块，用于和的计算。

(2) 修改BasicBlock.java，为BasicBlock类增加属性Map<Temp, Set<Pair>> usePoints, Map<Temp, Set<Pair>> innerUsePoints, Set<Temp> dudef, Set<Pair> DUdef, Set<Pair> DUliveUse, Set<Pair> DUliveIn, Set<Pair> DUliveOut

各属性的意义见1.1.1.

(3) 修改BasicBlock.java的computeDefAndLiveUse方法

增加和的计算，算法见1.1.4.

(4) 修改BasicBlock.java的analyzeLiveness方法

增加的修改和的计算，算法见1.1.3.

(5) 修改BasicBlock.java，增加computeDUdef方法

通过计算，算法见1.1.4.

(6) 修改FlowGraph.java，增加computeUsePoints方法

通过每个BasicBlock的属性计算属性，算法见1.1.4.

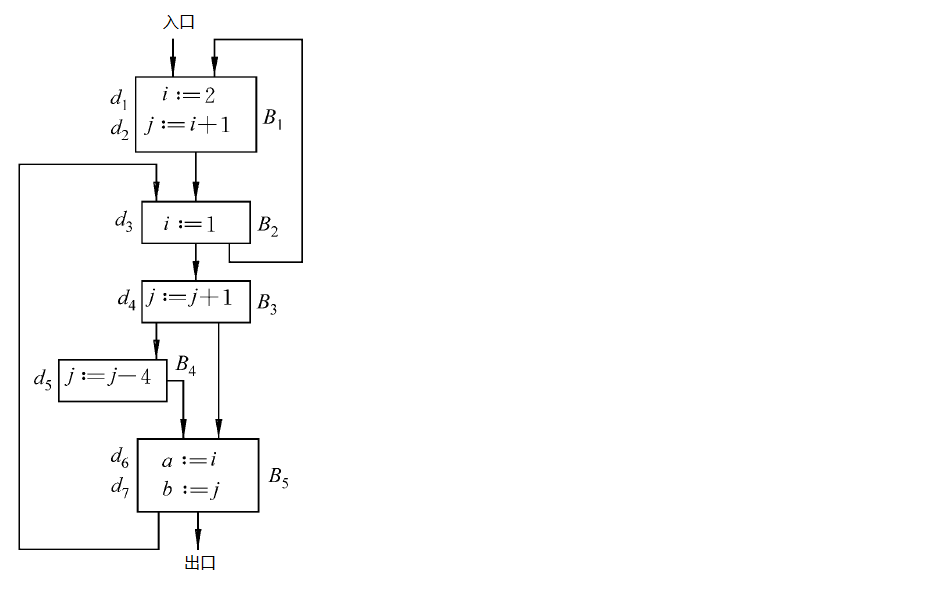
(7) 修改FlowGraph.java的analyzeLiveness方法

依次调用BasicBlock.computeDefAndLiveUse方法、FlowGraph.computeUsePoints方法、BasicBlock.computeDUdef方法初始化和。

增加DU链方程的计算过程。

**2. 以TestCases/S4/t0.decaf为例，分析TAC序列与DU链信息**

t0.decaf的数据流图如下图所示：



(1) 初始化和

(2) 迭代计算DUliveOut

以下用表示，其中是变量的定值点。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

(3) 求每条定值语句的DU链

综上，