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，s为一条 TAC 语句，A 为一个变量；

令DUliveUse[B]为(s, A)的集合，其中s是 B 中某点，且 B 中在s前面没有A的定值点；

令DUdef[B]为(s, A)的集合，其中s是不属于 B 的某点，s引用了变量A且A在B中被定值；

令DUliveIn[B]为(s, A)的集合，表示一个基本块入口处的 DU 链；

令DUliveOut[B]为(s, A)的集合，表示一个基本块出口处的 DU 链；

令innerUsePoints[B]为(s, A)的集合，表示一个基本块内所有引用的定值点-变量集合；

令usePoints[B]为(s, A)的集合，表示一个基本块外所有引用的定值点-变量集合；

令dudef[B]为A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：

$$\text{DUliveIn[B]} = \text{DUliveUse[B]} \cup (\text{DUliveOut[B]} - \text{DUdef[B]})$$

$$\text{DUliveOut[B]} = \cup DULiveIn[b], b \in next[B]$$

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s是变量A的定值点，从 DUliveOut 中取出所有变量 A 的引用点作为s的 DUChain，并从 DUliveOut 中删除所有 A 的引用点。如果s引用了变量 C，将(s, C)加入 DUliveOut;

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

转入上一条语句的分析。

(4) DUliveUse[B]与DUdef[B]的初始化算法

从基本块 B 的入口语句开始，如果语句s引用了没有在 B 中被定值的变量A，则将(s, A)加入DUliveUse[B]；

从基本块 B 的入口语句开始，如果语句s引用了变量A，则将(s, A)加入innerUsePoints[B];

从基本块 B 的入口语句开始，如果语句s定值了变量A，则将变量 A 加入 dudef 中；

计算得到每个基本块的innerUsePoints，计算每个基本块的usePoints，再根据dudef得到 DUdef.

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DUliveUse和dudef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innerUsePoints, dudef和DUliveUse的计算，算法见 1.1.4.

(4) 修改 BasicBlock.java 的 analyzeLiveness 方法

增加DUliveOut的修改和DUChain的计算，算法见 1.1.3.

(5) 修改 BasicBlock.java，增加 computeDUdef 方法

通过usePoints, deduf计算DUdef，算法见 1.1.4.

(6) 修改 FlowGraph.java，增加 computeUsePoints 方法

通过每个 BasicBlock 的innerUsePoints属性计算usePoints属性，算法见 1.1.4.

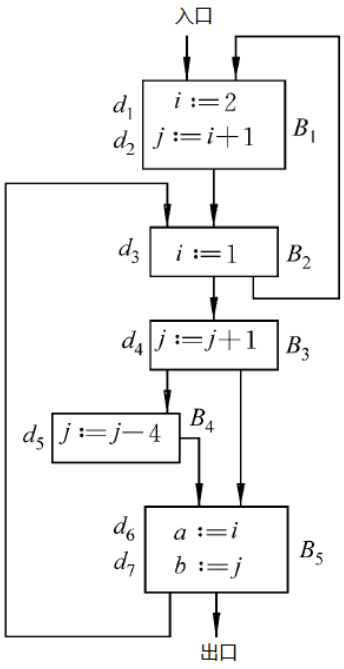
(7) 修改 FlowGraph.java 的 analyzeLiveness 方法

依次调用 BasicBlock.computeDefAndLiveUse 方法、FlowGraph.computeUsePoints 方法、BasicBlock.computeDUdef 方法初始化DUliveUse和DUdef。

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DUliveUse和DUdef

$$\mathrm{DUliveUse}[\mathrm{B_1}] = \{(\mathrm{d_2}, i)\}, DUdef[B_1] = \{(d_4, j), (d_5, j), (d_7, j)\}$$

$$\mathrm{DUliveUse}[\mathrm{B_2}] = \Phi, \mathrm{DUdef}[\mathrm{B_2}] = \{(\mathrm{d_2}, \mathrm{i}), (\mathrm{d_6}, \mathrm{i})\}$$

$$\mathrm{DUliveUse}[\mathrm{B_3}] = \{(\mathrm{d_4}, j)\}, DUdef[B_3] = \{(d_5, j), (d_7, j)\}$$

$$\mathrm{DUliveUse}[\mathrm{B_4}] = \{(\mathrm{d_5}, \mathrm{j})\}, \mathrm{DUdef}[\mathrm{B_4}] = \{(\mathrm{d_4}, \mathrm{j}), (\mathrm{d_7}, \mathrm{j})\}$$

$$\mathrm{DUliveUse}[\mathrm{B_5}] = \{(\mathrm{d_6}, \mathrm{i}), (\mathrm{d_7}, \mathrm{j})\}, \mathrm{DUdef}[\mathrm{B_5}] = \Phi$$

(2) 迭代计算 DUliveOut

以下用kj表示$(\mathrm{d_k}, j)$，其中$\mathrm{d_k}$是变量j的定值点。

	DUliveUse	DUdef	DUliveIn	DUliveOut
$\mathrm{B_1}$	Φ	4j, 5j, 7j	Φ	4j
$\mathrm{B_2}$	Φ	2i, 6i	4j	4j, 6i
$\mathrm{B_3}$	4j	5j, 7j	4j, 6i	4j, 5j, 6i, 7j

B_4	5j	4j, 7j	5j, 6i	4j, 6i, 7j
B_5	6i, 7j	Φ	4j, 6i, 7j	4j

(3) 求每条定值语句的 DU 链

$$DU(d_5, j) = DUliveOut[B_4] \cap j = \{4,7\}$$
$$DU(d_4, j) = DUliveOut[B_3] \cap j = \{4,5,7\}$$
$$DU(d_3, i) = DUliveOut[B_2] \cap i = \{6\}$$
$$DU(d_2, j) = DUliveOut[B_1] \cap j = \{4\}$$
$$DU(d_1, i) = DUliveOut[B_1] \cup \{2i\} \cap i = \{2\}$$

综上，

$$DU(d_1, i) = \{d_2\}$$
$$DU(d_2, j) = \{d_4\}$$
$$DU(d_3, i) = \{d_6\}$$
$$DU(d_4, j) = \{d_4, d_5, d_7\}$$
$$DU(d_5, j) = \{d_4, d_7\}$$