PA5实验报告

潘庆霖 2016011388

1、基本块中两个节点连边的条件

- 分析：在任意tac语句中，如果需要对一个标号进行定值，则可能需要分配寄存器。这时，就需要注意不能够影响已经分配的寄存器。即，不能和某些在这个块中已经定值过，且接下去还要使用的标号，发生寄存器分配冲突。这一情况在相干图中就体现为连边。
- 所以，对于当前基本块中的第n个tac语句来说，如果当中的某一标号，不妨假设为Tn，需要进行定值。则在相干图中和Tn需要连线的标号的集合可以由下列公式推导得到：

$$(.\sum_{i=0}^{n-1} Def(tac_i)).\bigcap(liveOut(tac_n))$$

- 因此，在具体的实现中，可以仅维护当前块内定值过并且在本tac之后依然活跃的标号的集合tacLive，动态维护这一集合，需要进行标号定值的时候，对这一集合中的每一个元素，都与当前要定值的标号连线即可。

2、如何在现有框架中实现完整的干涉图染色

- Mips.java中，需要修改：
  - 由逐个基本块分配改为逐个流图分配。
- GraphColorRegisterAllocator.java中，需要修改alloc函数：
  - 去掉saveLiveOutForBB的调用。
  - 首先，实例化一个InferenceGraph对象graph，传入当前的流图。
  - 调用graph的alloc函数进行寄存器分配。
  - 再对所有tac中的temp进行实际的寄存器绑定。
- 在IngerenceGraph对象的alloc函数中，需要实现：
  - 调用makeNodes，将传入的流图中各个基本块中所有的点都加入到干涉图中。
  - 遍历每一个基本块，调用makeEdges进行连边。makeEdges的伪代码描述如下：

```
tacLive.clear();
tacLive.addAll(BB.liveIn);
for tac in BB:
    for temp in tac:
        if temp is used:
            if temp is not in tac.liveOut:
                delete temp from tacLive;
        if temp is defined:
            for t in tacLive:
                add edge between t and temp;
```

关于tacLive的定义，修改为：在当前流图中，本条tac之前被定值过并且到达本条tac时依然活跃的标号所组成的集合。