

# Homework # 4

## due February 25

### 1 Reading

Finish reading Chapter 6 and then read Section 2.2

### 2 Problems

Do problem 2.3 on page 133:

- Define GEN and KILL sets.
- Is this forward or backwards?
- Does this use union or intersection?

### 3 Programming

Create a flow-graph for a program and perform the four analyses from Section 2.1 on it. Use a worklist and count the number of iterations required. Try both stack and queue worklists. You will *not* need the array lattice classes from the previous homework. We will ignore lattices for this homework.

Here is a suggested design: implement the following classes with the suggested public methods:

`FlowGraph` `getSuccessors`, `getPredecessors`, `getEntries`, `getExits`, `isEntry`, `isExit`, `isEdge`, `addEntry`, `addExit`, `addEdge`. The nodes should be WHILE program statements. You should also have ways to create a flow-graph from  $S^*$  and to reverse a flow-graph (for use in backwards analysis).

`Expressions` `canonicalize`, `usesVar`. Code to keep a canonical node for each identical arithmetic expression, and to test for “free variables.” Needed for available expressions and very busy expressions analyses.

`GenKillAnalysis` An interface with `getGen`, `getKill`, `getInitial`, `isIntersection`, `isReverse`. It should have four implementations: one for each analysis.

`Worklist` An interface with `add`, `hasNext`, `next`. It should have two implementations: one FIFO and one LIFO.

`PerformGenKill` A class that takes a flow-graph, a gen-kill analysis and a worklist and a method `run` which performs the analysis. For now, it can log the number of iterations to standard output. Another method should be provided to get the results before and after any node in the graph.

You *should* use the `ImmutableSet` class.

### 4 Submission

Please turn in the first part of your homework on paper at the beginning of lecture. The second part should be left in your AFS volume with an explanatory README file.