CS4120/4121/5120/5121—Spring 2016
Homework 4
Program Analysis
Due: Sunday, April 24, 11:59PM

0 Updates

- 4/20: Corrected a typo in Problem 2(c). Line 6 is supposed to be an assignment to a memory location.
- 4/19: Removed references to a "preorder". (A preorder is just a binary relation that is reflexive and transitive, but unlike a partial order, not necessarily antisymmetric.)
- 4/19: The lattice of Problem 1 got turned upside down. It has been turned right-side up again.
- 4/19: Clarification: actually, all problems are required of all students. There is no 5120-only problem.

1 Instructions

1.1 Partners

You may work alone or with *one* partner on this assignment. But remember that the course staff is happy to help with problems you run into. Use Piazza for questions, attend office hours, or set up meetings with any course staff member for help.

1.2 Homework structure

All problems are required of all students.

1.3 Tips

You may find the Dot and Graphviz packages helpful for drawing graphs. You can get these packages for multiple OSes from the Graphviz download page.

2 Problems

1. Bounding intervals

We want to design a dataflow analysis that computes conservative intervals bounding the values of all integer variables. An analysis along these lines could be used for eliminating bounds checks, for example. We extend the set $\mathbb{Z}$ of integers with plus and minus infinity: $\mathbb{Z}^* = \mathbb{Z} \cup \{-\infty, +\infty\}$, such that $-\infty < n$ and $n < +\infty$ for any integer n. We then define a lower semilattice over the set $L = \{[a, b] \mid a, b \in \mathbb{Z}^* \wedge a \leq b\} \cup \{\top\}$, with an ordering relation $\sqsubseteq$ and a meet operator $\sqcap$.

(a) Explain what the element $\top$ represents and why we need it for this analysis. Define the ordering and the meet operator for elements in this lattice (including $\top$).

(b) Using this lattice to compute ranges of variables will fail in general. Explain why.

(c) To solve the problems from part (b), for a given constant k, we define a lattice

$$L_k = \{[a, b] \mid a, b \in \{-\infty, -k, ..., k, +\infty\} \wedge a \leq b\} \cup \{\top\}$$

with the same ordering as before, and build a dataflow analysis that computes ranges in L_k. Show the flow functions for the following nodes:

- assignment to a constant `x = c`
- addition `x = y + z`
- multiplication `x = y * z`

(d) This analysis is an example of where it makes sense to propagate different information along different out edges from a node. Give a suitable flow function for each of the out edges of a comparison $x < c$, and show that with this analysis we can derive a bound $[10, 10]$ for variable `i` at the exit from the following example:

```
i = 0;
while (i < 10) {
    i = i + 1
}
```

2. Dataflow analysis: Defending against zombies

Let us define "undead" code as code that depends on a variable that is *always* uninitialized. When such undead code is removed, additional program regions may become undead due to the disappearance of variable declarations. The goal of this exercise is to remove all undead code from a function using only a single analysis pass.

(a) Design a dataflow analysis that can be used for cascading undead-code removal. Describe its ordering, the meet operator, the top element, as well as the flow function. Where necessary, be conservative. You only need to specify the flow function for assignments $x = expr$.

(b) Show that the flow functions you defined are monotonic, and either show that they are distributive or construct a counterexample.

(c) Show that one run of your analysis leads to the removal of the following grayed-out undead code (remember that meets are used at merge points in the CFG):

```
a = 1
if (f(a) > 0) {
  c = c+1
  d = 5
}
[d] = a+c
g = a+d
```

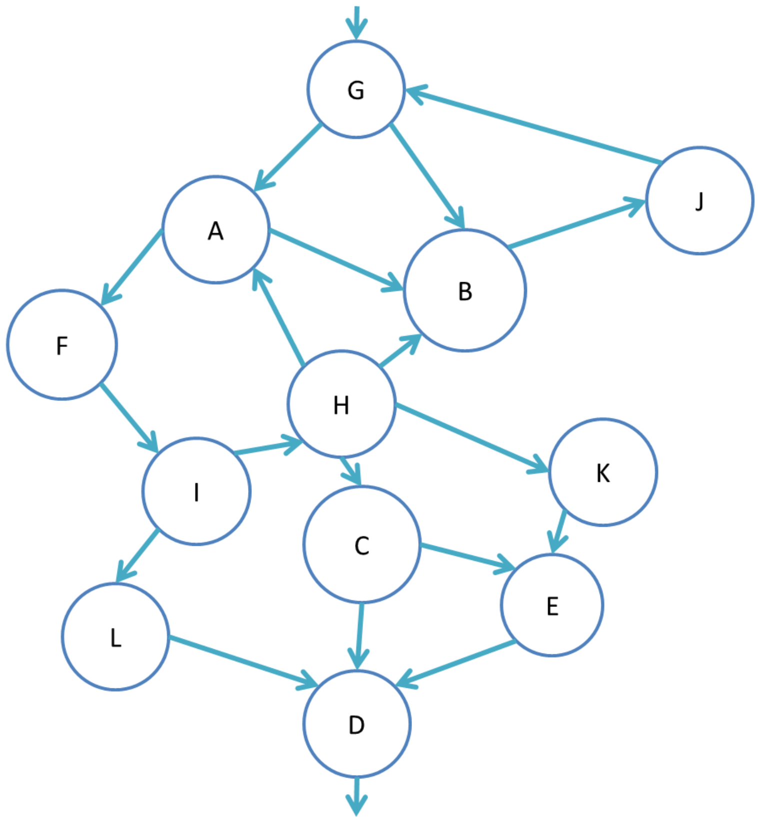

Figure 1: The control-flow graph for Problem 3

3. Control-flow analysis

For the control-flow graph in Figure 1, give the dominator tree, with back edges added as dashed edges. Identify the loops and the control tree, and for each loop indicate its set of nodes, its header node, and its exit edges.

3 Submission

Submit your solution as a PDF file on CMS. This file should contain your name, your NetID, all known issues you have with your solution, and the names of anyone with whom you have discussed the homework.