# CS 380C: Advanced Topics in Compilers
# Problem Set on Dataflow Analysis and Control Dependence
# Due: March 10th

March 7, 2016

**Late submission policy:** Submission can be at the most 2 days late. There will be a 10% penalty for each day after the due date (cumulative).

This is a written assignment. So, you do not need to write code.

1. (Dataflow Analysis) In this problem, you will use dataflow anaysis to perform partial redundancy elimination (PRE) in a small program.

   There are many algorithms for PRE. One of nicest ones is called "Lazy Code Motion" and it was described in an award-winning paper by Jens Knoop's group in PLDI 1992. A few months after Knoop et al's paper was published, Dreschler and Stadl published a simplified version of the algorithm in SIGPLAN Notices. We will use this simplified version of the algorithm, which we call DS93, in this assignment.

   (a) Read the lecture slides from Cooper and Torczon to get a sense of the DS93 algorithm. Then read the Lazy Code Motion paper and the DS93 paper.

   (b) Slide 16 of the Torczon and Cooper presentation has an example for the DS93 algorithm. Ignore basic blocks 1,2,3,4 in this example, and assume that block 5 is START.

   Write down the DS93 dataflow equations for this simpler program. Use the same notation as the paper, so your equations for basic block b should be of the form ANTIN(b) = ... and ANTOUT(b) = ...

   (c) Using any of the techniques described in class, solve these dataflow equations. You can use elimination, iteration, or both. You do not need to show the intermediate steps. Just show us the solution.

   (d) Verify that DS93 would transform the program as shown on slide 17.

2. (Control Dependence) Java Nagila is taking CS 380C and she has come up with this definition of control dependence.

   "A node n is control-dependent on a node p if there is a path in the control-flow graph from p to n that does not contain ipdom(p) (the immediate postdominator of p)."

   Is this definition equivalent to the standard one? If so, prove it. If not, give a counterexample.

3. (Open Problem) In class, we saw that dominance and reaching definitions can be computed using dataflow equations. Is there a way to compute (minimal) phi-placement as the solution of a dataflow system? Assume that some basic blocks (including START) have definitions of a given variable and set up a dataflow system that gives you phi-placement for that variable. It seems like this should be a forward-flow problem.

   I don't know how to do this but if you can figure it out, you can publish it as a research note and professors who need to teach SSA will be very grateful to you.