CS 243 - Assignment 1: Dataflow Analysis

Due: January 19, 3:00 pm

This homework contains a combination of written assignment and gradiance quizzes. For the radiance quizzes please refer to http://www.gradiance.com. For the written assignment, every student must hand in his or her homework. Please submit your homework in-class. SCPD students may submit their homework by e-mail via scpd-distribution@lists.stanford.edu or give your homework to the courier.

1. Is the following a meet operator? Please answer yes or no. If no, then please indicate which properties fail to hold. If yes, please sketch the semi-lattice.
   (a) Minimum value (on integers)
   (b) Product (on integers)
   (c) Addition (on integers)
   (d) Addition (on reals)
   (e) Product mod 2 (on the set {0, 1})
   (f) Addition mod 2 (on the set {0, 1})
   (g) The LCM (Lowest Common Multiple) function (on integers)
   (h) Mean/Average (on reals)
2. We say that a program point $p$ belongs to a live range of a definition $d \ u = x + y$ iff the value assigned to $u$ in definition $d$ may be accessed by using variable $u$ at point $p$, for some path in the flow graph starting at $p$. Consider for example the code fragment below:

The live range of the definition ‘$a = x + y$’ at point entrypoint($b_0$) is \{exitpoint($b_0$), entrypoint($b_1$), entrypoint($b_2$), entrypoint($b_3$)$\}$. Similarly, the live range of the definition ‘$a = a + 1$’ at point entrypoint($b_3$) is \{exitpoint($b_3$), entrypoint($b_4$)$\}$.

This concept of live ranges can be used to increase flexibility in register assignment. For example, the live ranges of the definitions of $a$ in $b_0$ and $b_3$ do not intersect at any points, so it may be possible to use same registers to hold these two values.

Describe an algorithm to find the live range of every definition in a program.

3. Design a data flow analysis to detect uninitialized variables. This can be used by the compiler to generate warnings to the programmer.
   - Warning I is issued on each use of a variable that may potentially be uninitialized.
   - Warning II is issued on each use of a variable that is definitely uninitialized

(a) What is the domain?
(b) What is the meet operator, top and bottom values?
(c) What is the direction of data flow?
(d) What is the initialization condition?
(e) What is the boundary condition?
(f) Describe the transfer function for a basic block.
4. This question asks you to think about how changes to the initial values in a data flow analysis can affect the result. Recall that an answer to a data flow problem is considered ‘safe’ if it is no bigger than the ideal solution. Suppose you are performing Live Variable Analysis but accidentally initialized IN(EXIT) to U.

Part 1.
(a) Will your algorithm give a safe answer for all flow graphs?
(b) If not, will it give a safe answer for some flow graphs? If it will, give an example.
(c) Will your algorithm give the MOP solution for all flow graphs?
(d) If not, will it give the MOP solution for some flow graphs? If it will, give an example.

Part 2.
Answer the same questions as part 1, but instead of initializing IN(EXIT) to U, what if IN(EXIT) = ∅ but IN[B] = ⊥ for all basic blocks B other than EXIT.