Homework 3 (Written):
Dataflow Analysis

17-355/17-665/17-819O: Program Analysis *(Spring 2019)*
Jonathan Aldrich
aldrich@cs.cmu.edu
course material developed with Claire Le Goues

Due: Thursday, February 7, 2019 (11:59 pm)

100 points total

Assignment Objectives:

- Precisely define an analysis using a lattice and flow functions.
- Simulate analysis execution on a program using the worklist algorithm.

Handin Instructions. Please submit your assignment on Canvas as a **PDF** by the due date. Name it **[your-andrew-id]-hw3.pdf**.

Integer sign analysis tracks whether each integer in the program is positive, negative, or zero. The results of this analysis can be used to optimize a program or to circumvent errors like using a negative index into an array. The analysis is broadly similar to the zero analysis discussed in class. For the purposes of this assignment, we will ignore the possibility of integer overflow (i.e. consider mathematical integers).

Question 1 (20 points).

> Design a ”precise” lattice for a single variable. Your lattice should track whether a value is less than zero, greater than zero, equal to zero, greater than or equal to zero, less than or equal to zero, non-zero, or unknown. Define the lattice by giving (a) the set of lattice elements and (b) the ordering relation between them, (c) the top element and (d) the bottom element.

Question 2 (10 points).

> Design a ”less precise” lattice for a single variable. This lattice should only track whether a value is less than zero, greater than zero, equal to zero, or unknown (which in this case will include cases like greater than or equal to zero). Define the lattice by giving (a) the set of lattice elements and (b) the ordering relation between them, (c) the top element and (d) the bottom element.

Question 3 (5 points).

What is the initial analysis information before the first statement of each function? Use the lattice from question 2, and justify your choice (more than one answer may be correct, so long as it is justified).

Question 4 (25 points).

Define a flow function for a multiplication of two variables assigned to a third variable, i.e. of the form $x := y * z$. Your flow function should be based on the second, simpler lattice. It should be as precise as possible given the analysis information available. You may define the flow function in any notation you like (e.g. mathematics, code, pseudo-code) as long as it is unambiguous.

Question 5 (10 points).

Assume you had an implementation of your sign analysis, as specified above in questions 2, 3, and 4. Explain how you would detect errors due to a negative array index. Specifically, assume a 3-address code operation of the form $x := y[z]$, and describe what condition on the analysis results just before such an operation would yield (a) a definite negative array index error and (b) a possible negative array index error (e.g. in cases where the analysis is too imprecise to tell if there is definitely an error).

Question 6 (30 points).

Simulate your analysis from questions 2, 3, and 4 on the following program using Kildall's worklist algorithm with the strongly-connected component and reverse post-order heuristics. You should produce a table as done in class. Your table should have a column for the program point, a column for tracking the worklist, and a column for the abstract value of each variable. Each row should track the value after the execution of the corresponding statement. The rows should show how the analysis executes, examining one statement at a time:

```
1 :   $x := 0$
2 :   $y := 5$
3 :   $z := -3$
4 :   if $w > 0$ goto 7
5 :   $w := x * z$
6 :   goto 8
7 :   $w := z * z$
8 :   if $w > 0$ goto 12
9 :   $z := y * z$
10 :  $y := w$
11 :  goto 8
12 :  $x := z * z$
```