University of Toronto - Department of Computer Science
CSC410, Fall 2016
Homework 4 - Updated 7 November

This homework is worth 4% of your final mark

Due: Sunday, 13 November 2016 at 23:59

Instructions:

The homework is to be done individually. You can write the answers on a sheet of paper or using an editor. Handwritten answers must be legible; otherwise, it will affect the mark. Write clearly which problem you are answering.

To submit, either scan your solutions or save your document as a PDF file and upload it to MarkUs.

Note that while all of the material is relevant for the final, only some problems may be marked.

Questions? Ask them on Piazza (folder hw4).

At the beginning of the homework, include and complete the following header :

==========

CSC410, Fall 2016 - Homework 4

Name: _____

Student Number: _____

Lecture: o Monday o Tuesday

I am the sole author of this homework. Signature: _____

==========

Problem 1 (Dataflow analysis - Constant propagation)

```
void problem1(){
   1. w = 820
   2. y = 410
   3. while (y > 400) {
   4.    if (y == 410) {
   5.       z = 1000;
```

```
   6.     } else {
   7.        z = w+5;
   8.     }
   9.     y = 400;
   10.   }
   11.   return;
}
```

1. [2 marks] Draw the control flow graph representing `problem1`.
2. [2 marks] According to the CFG, which statements require calculating the intersection of the facts of two (or more) other statements?
3. [6 marks] Apply constant propagation analysis to determine for each statement which variables are known to be constant and which are not.
   For each statement fill the following table showing how the information is propagated:

stmt	in	out	If union or intersection, write here which statements are the input to the result
1	?	?	?

4. [1 mark] Is it a forward or backward flow analysis? All-paths analysis or one-path analysis?

Problem 2 (Dataflow coverage)

For the following variation of the function in Problem 1:

```
   1. void problem2(int y){
   2. w = 820
   3. while (y > 400) {
   4.     if (y == 410) {
   5.        z = 1000;
   6.     } else {
   7.        z = w+5;
   8.     }
   9.     y -= 1;
   10.   }
   11.   return;
}
```

1. [3 marks] List test cases (sequences of statements that represent a feasible path) that allow 100% “All-Defs” coverage. (Indicate which defs are covered by each test)
2. [3 marks] List test cases that allow 100% “All-Uses” coverage (Indicate which uses are covered by each test)
3. [5 marks] List test cases that allow 100% “All-DU-Paths” coverage (Indicate which du paths are covered by each test)
4. [2 marks] List test cases that allow 100% “All-C-Uses, Some-P-Uses” coverage (Indicate whether it’s all c-uses, or some p-use and how each test covers each of them)
5. [3 marks] List test cases that allow 100% “All-P-Uses, Some-C-Uses” coverage (Indicate whether it’s all p-uses, or some c-use and how each test covers each of them)

Problem 3 (Dataflow analysis - Live variables)

```
void problem3(){
   1. w = x + y + z
   2. if (y > 10){
   3.    w = x-y
   4. } else {
   5.    w = 2*x
   6. }
   7. y+=w;
   8. return;
}
```

1. [2 marks] Draw the control flow graph representing `problem3`.
2. [2 marks] According to the CFG, which statements require calculating the union of the facts of two (or more) other statements?
3. [6 marks] Apply live variable analysis to determine which variable definitions are live at each statement. To do so, fill in a table as the one shown in Problem 1, that allows clearly seeing how the analysis works.
4. [1 mark] Is it a forward or backward flow analysis? All-paths analysis or one-path analysis?