Exercise 1

Jan 17, 2008

1. (10pts) Categorize the following program analysis problems based on their nature of approximation (may vs. must analysis), precision (flow/path and context sensitivity) and scope (local, regional, global, and whole program). Note that each analysis problem may be solved using multiple choices and you should try to list all the valid options.

   (a) For each variable reference v in a program, the statement that last defines the value of v.

   (b) For each pair of array references, $ar1$ and $ar2$, that refer to the same array and are inside a common loop, the difference in loop iterations that $ar1$ and $ar2$ would refer to the same memory location.

   (c) For each pair of distinct variable references in a program, whether or not a common memory location may be reached from both references (i.e., whether or not the two references could be aliased).

   (d) For each pair of distinct variable references in a program, whether or not a common memory location can always be reached from both references (i.e., whether or not the two references are always aliased).

   (e) For each pointer variable dereference in a program, whether a memory error could be caused by the pointer dereference (i.e., whether the pointer variable could point to an invalid memory location).

2. (10pts) Given the following control-flow graph,

```
                  I: [c := 100]1
                     [a := 0]2
                     [b := 0]3
                     [i := 0]4
      ---------------      |
     |               V     V
     |            L: [a := a + 2]5
     |               [b := a + b]6
     |               [if  (50 < i) goto E]7
     |              /                  \
     |             /                    \
     |            V                      V
     |   B1: [c := c / 2]8          E: [return b]12
     |       [if (c < a) goto B]9
     |          |           |
     |          V           |
     |   B2: [c:=c+a]10      |
     |          \           |
     |           V          V
     |           B: [i := i + 1]11
     |               goto L
      ------------------|
```

Use the data-flow analysis algorithm to compute the reaching definitions for each basic
block.