17-654/17-754 Homework #1 ADDENDUM

Out Jan. 18

Due Jan. 24, 11:59PM

Problem #1

No Changes

Problem #2

Q: What's going on with the after statement results? Why is there a $3_{(true)}$ and a $3_{(false)}$?

A: I'm so glad you asked! What's happening is that we're being clever. We've observed that the combination of a branching condition and the taken vs. not-taken path can often tell us something useful about the sign of the variable being tested. Thus, for example, when the while-loop test is **`[a > 0]`**, we know that **`a`** must be greater than 0 on entry to the loop body. Conversely, we know that **`a`** must be less than or equal to 0 when the loop exit path is taken.

Here's how you'll handle this for your transfer functions:

1. You'll write three transfer functions for Sign Analysis. They will be:
   a. $SA_{entry}(l)$ which will define the before values in terms of the appropriate exit values
   b. $SA_F(l)$ which will define the after values for false branches of if and while statements. It should be defined in terms of $SA_{entry}(l)$ and of the $\mathbf{Gen_{SA,F}(B^l)}$ and $\mathbf{Kill_{SA,F}(B^l)}$ functions you will also define.
   c. $SA_T(l)$ which will define the after values for true branches of if and while and for the fall-through of all other statements. It should be defined in terms of $SA_{entry}(l)$ and of the $\mathbf{Gen_{SA,T}(B^l)}$ and $\mathbf{Kill_{SA,T}(B^l)}$ functions you will also define.
2. Some of the **`Gen`** and **`Kill`** functions you write will need to be bipartite. By this I mean that they will take the form:

   $\mathbf{Kill_{SA,F}(B^l)}$ **`= <<something>>  when(<<some condition>>)`**
   `<<somethingelse>>  otherwise`

3. You may find it easier to combine the smarts for the if and while tests into the statement gen and kill rather than the expression gen and kill. If so, you need only handle a single level of comparison to a single variable. You are not expected to write the fully general transfer functions that could do right by

more complex Boolean expressions like

```
((a>0) && (b=1) || (c< -1))
```

.[1]

See slide 32 of Thursday's lecture for notation details. You can find the slides on the class web site.[2]

Hint: Compare the results you get with your transfer functions to the charts given in the homework. We gave you the example results for a reason...

[1] This is an important simplification. Pay attention to this hint!

[2] Yes, I know we didn't get that far. That's why this Addendum to the homework.