17-654/17-754 Homework #1

Out Jan. 18

Due Jan. 24, 11:59PM

Problem #1

Consider the following program:

```
[b := 0]1
[e := false]2
while [a > 0]3 do
     if [(a & 1) = 0]4 then
          [b := b + 1]5
     else
          [e := true]6

     [a := a / 2]7
     [e := false]8
     [b := b * 2]9
if [a < 0]10 then
     [b := -1]11
else
     [skip]12
[skip]13
```

(5 points) Draw a flow-graph for this program. Clearly identify which instructions belong in which nodes.

(0 points) Just for grins (not for credit): What does this program do? Assume that "&" is bitwise-and.

Problem #2

Sign analysis attempts to determine whether the value is positive, negative or zero. The results of this analysis have a variety of uses in optimizing compilers. Although its operation is broadly similar to that of Reaching Definitions, there are a few key differences:

1. Instead of propagating the positions of assignments, we instead propagate tuples of (var, value-set) where the value-sets are drawn from{false, true, -, 0, +}[1], and we use ? as the unknown value. Note that ? is really an abbreviation for the set {false, true, -, 0, +}.
2. We don't care which assignment a value comes from.

---

[1] Note that this definition combines Integer Sign analysis with Boolean constant propagation. This is not a necessary property for sign analysis, but makes the problem a little more interesting.

3. For the purpose of this problem, we will ignore the possibility of integer overflow.

Sign Analysis applied to the program from problem one yields:

Before results

before stmt #	a	b	e
1	?	?	?
2	?	{0}	?
3	?	{0,+}	{false}
4	{+}	{0,+}	{false}
5	{+}	{0,+}	{false}
6	{+}	{0,+}	{false}
7	{+}	{0,+}	{false, true}
8	{0,+}	{0,+}	{false, true}
9	{0,+}	{0,+}	{false}
10	{-,0}	{0,+}	{false}
11	{-}	{0,+}	{false}
12	{0}	{0,+}	{false}
13	{-,0}	{-,0,+}	{false}

After results

after stmt #	a	b	e
1	?	{0}	?
2	?	{0}	false
3(true)	{+}	{0,+}	{false}
3(false)	{-,0}	{0,+}	{false}
4(true)	{+}	{0,+}	{false}
4(false)	{+}	{0,+}	{false}
5	{+}	{+}	{false}
6	{+}	{0,+}	true
7	{0,+}	{0,+}	{false, true}
8	{0,+}	{0,+}	false
9	{0,+}	{0,+}	false
10(true)	{-}	{0,+}	{false}
10(false)	{0}	{0,+}	{false}
11	{-}	{-}	{false}
12	{0}	{0,+}	{false}
13	{-,0}	{-,0,+}	{false}

For problem #2:

1. (10 points) Consider the analysis results for label 7 in the example program above. Give a different possible analysis result for label 7 (both before and after) that are:
    a) precise but unsafe
    b) safe but imprecise
2. (30 points) Come up with transfer functions for the analysis. State the function in terms of gen and kill sets. Note that you must consider arithmetic expressions and Boolean expressions as well as statements. Give transfer functions for each statement, plus the Boolean-valued relational operators on Integers (<, <=, and = will do fine, no need to do them all) and integer * and -.
3. (20 points) Simulate analysis on the program from Problem 1 using chaotic iteration until the values reach a fixed point. Use the format from lecture to show your work
4. (25 points) Prove that your transfer functions for assignment and <= are monotone.

5. (5 points) Write simple example code where your analysis yields an approximate (imprecise) result. By this we mean a result for which a human can easily see the correct answer, but for which the analysis yields a less precise answer.
6. (5 points) Explain briefly why your analysis will always be approximate, no matter how "smart" you make it