

COMP 621 Analyses and Transformations - Assignment # 3
 Interprocedural Analysis, Points-to Analyses and Special Topics
 Due Date: December 7th, 2015

1 Q1: Intra- and Inter-procedural Analysis (25 pts)

Consider the following program, with the usual C semantics (pass-by-value function calls, etc.):

```

01: int x,y;
02: main()
03: { int i,j;
04:   y = 7;
05:   x = y + 20;
06:   i = x + 1;
07:   /*-- Point A --*/
08:   foo(i);
09:   j = y + 3;
10:   /*-- Point B --*/
11:   x = 2;
12:   bar(i);
13:   j = 3*x;
14:   /*-- Point C --*/
15: }

16: void foo(int u)
17: { u = u * 2;
18:   x = u + 1;
19:   /*-- Point D --*/
20:   leaf(u);
21: }

22: void leaf(int c)
23: { int z;
24:   z = 6;
25:   write(c+z);
26:   /* -- Point E --*/
27: }

28: void bar(int b)
29: { int z;
30:   y = b + 2;
31:   if (x != 2)
32:     foo(b);
33:   else
34:     { z = b+13;
35:       foo(z);
36:     }
37:   /*--Point F --*/
38: }

```

For the above program, give the constant propagation information at the labelled program points (A through F), knowing the analysis both propagates constant values when it is safe to do so and evaluates constant expressions. Use the following different strategies to estimate the effects of procedure calls. You only need to give the results for the program points mentioned in each section below. Give your answer for constant values as a set of variable-value pairs (ex: $\{(x,1),(y,2)\}$) for each program point.

- a) Conservative Approach: Assume nothing is known about the called procedures and provide the information at program points A, B and C. Clearly state what assumptions you are making for your conservative approach.
- b) Summary Approach: Prepare a summary of each procedure as to which variables can be potentially modified by any call to the procedure (including transitive calls) as a side-effect. For each function, provide the set of variables that can potentially be modified (ex: $\{x,y\}$). Use this summary information to get sharper constant information at program points A, B and C.
- c) Full-blown Approach: (1) Precisely estimate the effect of a procedure call, by visiting and analyzing the body of the called function for each call. Provide the constant information at program points A, B, C, D and E. You should tag each piece of dataflow information with the calling path. For example, supposing 'x' has value '1' in 'foo' at point D when called from main at line 8, the answer would be $\{('main-08/foo', x, 1)\}$.

2 Q2: Points-to Analysis (25 pts)

Consider the following program:

```

void foo(int x, int y)
{ int z, *p, *q, **pp, **qq;
  z = x + y;
  if (z > 10)
  { p = &y;
    q = &z;
    pp = &p;
    /* --- POINT A --- */
  }
  else
  { p = &z;
    q = &z;
    pp = &p;
    /* --- POINT B ---*/
  }
  qq = pp;
  /* --- POINT C --- */
  *qq = q;
  /* --- POINT D --- */
  S: *p = 128;
  x = *q;
  write(x);
  write(y);
  write(z);
}

```

- (a) What does this program write for the calls `foo(3,4)` and `foo(5,6)`?
- (b) Give the flow-sensitive points-to sets that would be computed at the named points (Points A through D).
- (c) Using the points-to sets from (b), what program transformation, if any, could be performed at program point S?
- (d) Show the points-to graph that would be computed by a flow-insensitive points-to analysis for function `foo`.
- (e) Using the points-to graph from (d), what program transformation, if any, could be performed at program point S?

3 Q3: Special Topics (20 pts)

Answer any long question from a special top presentation (other than your presentation).

4 Q4: Your Special Topic (30 pts)

- (a) State your special topic question.
- (b) Briefly justify why you chose that question.
- (c) Give the answer that you are expecting, along with explanations to help the TA. Give the complete answer if you gave one option, or an outline for each option if you gave several options. Your own grade for this question will be influenced by how easy it is to grade other student's answers given yours. A complete, precise answer with highlighted important pedagogical points make it easier to evaluate other student's answers.