COMP 621 Analyses and Transformations - Assignment # 3
*****Please answer Q5 plus any three of the remaining questions (total of four questions)*****
Interprocedural Analysis, Register Allocation, Points-to Analyses and Special Topics
Due Date: Official deadline, Friday, April 11, automatic extension to April 18 if you prefer.

1 Q1: Intra- and Inter-procedural Analysis

Consider the following program:

```
int x,y;
main()
{
  int i,j;
  x = 13;
  y = x + 20;
  i = x + 1;
  /*-- Point A --*/
  foo(i);
  j = y + 3;
  /*-- Point B --*/
  x = 10;
  bar(i);
  j = x + 3;
  /*-- Point C --*/
}
```

```
void foo(int u)
{
  u = u * 2;
  x = u + 1;
  /*-- Point D --*/
  leaf(u);
}

void leaf(int c)
{ int t;
  t = 10;
  write(c+t);
  /* -- Point E --*/
}
```

```
void bar(int b)
{ int t;
  b = b * 2;
  y = b + 2;
  if (x != 10)
    foo(b);
  else
    { t = b+13;
      foo(t);
    }
  /*--Point F --*/
}
```

For the above program, give the constant propagation information at the labelled program points (A through F), using 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.

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). Use this summary information to get sharper constant information at program points A, B and C.

c) Full-blown Approach: 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 a call string (i.e. the complete calling path).

2 Q2:Register Allocation

Use the following example program.

```
{  a = read();
   b = read();
   if (a > 10)
     { i = a;
       sum = 0;
       while (i != 0)
         { sum = sum + i;
           i = i - 1;
         }
       write(sum);
     }
   else
    { i = b;
      prod = 1;
      while (i >= 0)
         { prod = prod * i;
           k = a - b;
           i = i - k;
         }
      write(prod);
    }
}
```

(a) Draw the register interference graph for the example program. Assume that the underlying architecture can use the same register as both a source and destination (i.e. instructions like `MOV R1, R1` and `ADD R1, R2, R1` are allowed). Based on this interference graph, give a minimal register assignment to the program (you may just indicate which register number is assigned to each program variable).

(b) Is there any program transformation that could be used to reduce register pressure in this program? If so, indicate the transformation.

3 Q3: Points-to Analysis

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 = &x;
      qq = &p;
      /* --- POINT A --- */
    }
  else
    { p = &x;
      q = &x;
      qq = &p;
      /* --- POINT B ---*/
     }
  pp = qq;
  /* --- POINT C --- */
  *pp = q;
  /* --- POINT D --- */
  S: *q = 512;
  z = *p;
  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-senstive 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?

4 Q4: Special Topics

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

5 Q5: Your Special Topic

(a) State your special topic question.

(b) Briefly justify why you chose that question.

(c) Give the answer that you are expecting.