University of Leicester
CO7206: Quiz #1 Model Answer (8%)
Oct 23rd, 2003

Solve the three following questions (each question equals 3%). The maximum mark a student can get is 8%). You can use your printed notes or the course web page. You **cannot** use any other resources or cooperate with other students in solving the quiz. Duration is 40 min.

1. For the program below (Taken from Figure 28.14 in SE by Sommerville, 6th Edition):

- Translate this unstructured routine to a structured one and eliminate all goto statements
- Suggest meaningful names for the variables in the program to enhance its maintainability (this would require that you relatively understand what it does)

```
Routine BS (K, T, S, L)
1    B := 1
2    NXT: if S >= B goto CON
3    L := -1
4    goto STP
5    CON: L:= INTEGER (B/S)
6    L := INTEGER ((B+S) / 2)
7    if T(L) = K then return
8    if T(L) > K then goto GRT
9    B:= L + 1
10   goto NXT
11   GRT: S:= L - 1
12   goto NXT
13   STP: end
```

One possible solution to this problem is:

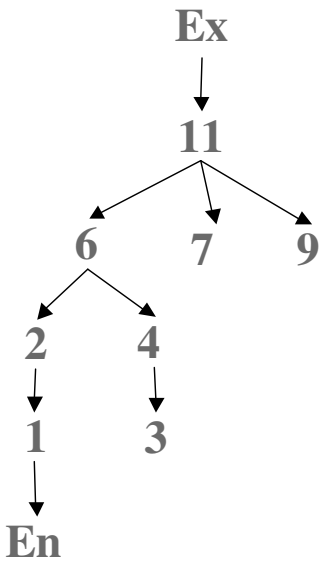
```
Routine BS (K, T, S, L)
1    B := 1
2    while (S >= B)
5         L := INTEGER (B/S)
6         L := INTEGER ((B+S) / 2)
7         if T(L) = K then return
8         if T(L) > K then S := L - 1
8         else B:= L + 1
10   end while
11   L := - 1
end
```

Variable names: BS → BinarySearch, K → Key, T → Table, S → Size, L →Lookup, B → Base

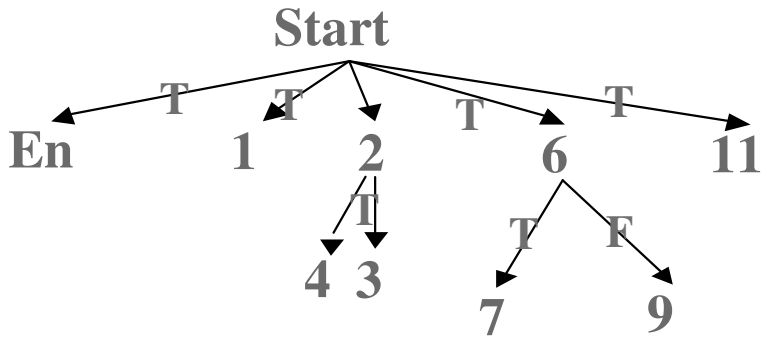
2. Construct the post dominance tree and the control dependence graph for this Ada program (skip any unnecessary statements, e.g., st 5, 8, 10)

```
function P return INTEGER is
begin
     X,Y:INTEGER
1    READ(X);READ(Y);
2    while (X>10) loop
3         X:=X-10;
4         exit when  X=10;
5    end loop;
6    if (Y<20 and then X mod 2 = 0) then
7         Y:=Y+20;
8    else
9         Y:=Y-20;
10   end if;
11   return 2*X+Y;
end P;
```

Post Dominance Tree



Control Dependence Graph



3. Do reaching definitions analysis for this program (at least 3 iterations) and calculate the DU and UD chains. Based on the findings of this analysis, suggest ways of simplifying the program, if possible (e.g. eliminating unneeded statements or simplifying some statements). Justify your suggestions with the outcomes of the analysis.

```
1. read (n,c1)
2. b = 5
3. for i := 1 to n do
4.       a := 2
5.            if c1==1 then
6.                a := 4
7.            else
8.                a := 6
9. z := a + b
10.write (z)
```

A detailed solution is given below. More abstract versions are acceptable too. For example, maximal blocks can be used, with the block one containing definitions 1:n, 1:c1 and 2:b.

$$OUT = GEN \cup (IN - KILL)$$

St	Gen	Kill	Out0	In1	Out1	In2	Out2
1	1:n, 1:c1		1:n,1:c1	–	1:n,1:c1		1:n,1:c1
2	2:b		2:b	1:n, 1:c1	1:n,1:c1, 2:b	1:n,1:c1	1:n,1:c1 ,2:b
3	3:i		3:i	2:b, 6:a, 8:a	2:b,3:i, 6:a,8:a	1:n,1:c, 2:b,6:a, 8:a	1:n,1:c, 2:b,3:i, 6:a,8:a
4	4:a	6:a,8:a	4:a	3:i	3:i,4:a	2:b,3:i, 6:a,8:a	2:b,3:i, 4:a
5				4:a	4:a	3:i,4:a	3:i,4:a
6	6:a	4:a,8:a	6:a		6:a	4:a	6:a
8	8:a	4:a,6:a	8:a		8:a	4:a	8:a
9	9:z		9:z	3:i, 6:a, 8:a	3:i,6:a,8 :a,9:z	2:b,3:i, 6:a,8:a	2:b,3:i, 6:a,8:a, 9:z
10				9:z	9:z	3:i,6:a, 8:a,9:z	3:i,6:a, 8:a,9:z

A possible suggestion to simplify the program is to eliminate statement 4 since this definition of `a` does not reach any use of `a`.