CS352 - Compilers: Principles and Practice
Homework 3 - One Reg, Two Reg, Red Reg, Blue Reg
Due: 2004 December 09, beginning of lecture

Consider this simple code fragment:

```
L0:   a  = r1
      b  = r3
      c  = r4
      r1 = a + 5
      call func1
      d  = r1
L1:
      r1 = d
      r2 = 0
      call func2
      e  = d + r1
      r1 = e
      if e < 10 goto L1
      r4 = c
      r3 = b
      return  (r1,r3,r4 live out)
```

The code has been compiled for a target architecture with four (4) registers:

- $r1$ is a (caller-save) argument/result register,
- $r2$ is a caller-save argument register.
- $r3$ and $r4$ are callee-save temporaries.

1. Build the control flow graph for this program fragment, and perform liveness analysis on it. Show the *in* and *out* sets for each node in the graph. (There should be one node for each line of code.)
2. Construct the interference graph for the program fragment, using solid lines to show interference, and dotted lines to show *MOVE* relations.
3. Show the steps for register allocation (graph coloring with conservative coalescing) for this program in detail, as described in the notes and pages 229–232 of the textbook. When choosing nodes for potential spills, use the spill priority heuristic $(number\ of\ defs + number\ of\ uses)/degree$. Note which criteria you use for conservative coalescing.
4. Show the final program after register allocation, with redundant moves removed.