Optimising Compilers 2005 – Paper 8 Question 7 (AM)
Solution Notes

[This question is about part A of the course, dataflow equations and register colouring]

(*a*)

$$R(n) = \bigcup_{p \in pred(n)} (R(p) \setminus ref(p) \cup def(p))$$

Initialise $R[n]$ to empty (unsafe, but results in most useful safe solution); Iterate

$$\texttt{for i = 1 to N do } R[i] := \bigcup_{p \in pred(i)} (R[p] \setminus ref(p) \cup def(p))$$

while changes occur during any iteration. Report a write-write anomaly at node i if $(R(i) \setminus ref(i)) \cap def(i) \neq 0$.

(*ii*) Address taken: need to treat possible definition as def, but possible ref as like a no-op. Global variables are trickier – they may be address taken almost anywhere else, even if not, we must assume any proc. call may read or write them, perhaps even maintaining records of what external calls may do, like an effect system.

(*b*) [Note for supervisors: not obviously even possible if we additionally require SSA form.] There are shorter answers (e.g. 4 assignments, but this one is clear and is not trivially optimisable).

```
void f()
{    int a,b,c,d;
     a = g(11); b = g(12); h(a+b);
     b = g(21); c = g(22); h(b+c);
     c = g(31); d = g(32); h(c+d);
     d = g(41); a = g(42); h(d+a);
}
```

(*ii*)

$$\begin{aligned} c(k) &= 0 \text{ if } k = 0 \text{ (not really expected!)} \\ c(k) &= 1 \text{ if } k = 1 \\ c(k) &= 2 \text{ if } k \textit{ even} \\ c(k) &= 3 \text{ if } k \textit{ odd} \end{aligned}$$