P9Q7
AM

Optimising Compilation `optcomp9.tex` 2001

Sketch an algorithm for *instruction scheduling.* Be careful to specify the nature of your initial and final data-structures; also give the worst-case running time of the algorithm and a program which causes this worst-case running time to be achieved. [10 marks]

Explain the conflicts between register allocation, common sub-expression elimination and instruction scheduling on a machine with a register use should be minimised (for example to reduce procedure entry save/restore code). [5 marks]

Describe Static Single Assignment (SSA) form and explain how it can help to produce better results in both compilation and decompilation. [5 marks]

Solution Notes

1. Bookwork (notes). 2. CSE temps can cause additional spilling – arrange that small CSE temps are spilt by undoing the CSE optimisation. Similarly "ld r0; st r0; ld r0; st r0" can be scheduled into faster "ld r0; ld r1; st r0; st r1" but only at the cost of using more register. 3. It reduces a single user variable into separate variables for each live range. For compilation is can reduce the number of registers required; for decompilation is can increase the number of variable (instead of just using r0..rn as variables).