2007 Paper 8 Question 8

Optimising Compilers

(a) Summarise the core ideas of *Common Sub-Expression Elimination* (CSE) optimisation based on dataflow analysis. Your explanation should include dataflow equations (including auxiliary functions used) and sketches of how (i) to compute dataflow solutions and (ii) these solutions drive the transformation itself. [8 marks]

(b) Consider P, the following SSA-form intermediate code, resulting from CSE (treat `READ A` as an atomic instruction writing a run-time-determined value to `A`):

```
1: READ A
2: READ B
   ...
6: Z := A+B
7: C := Z
8: D := Z
9: PRINT C+D
```

(i) Instruction 6 was inserted by CSE. Leaving the other instructions alone, where else might it have been reasonably inserted? Indicate any conditions your answer places on the unspecified instructions 3 to 5, recalling that P is in SSA form. [3 marks]

(ii) Registers are often in short supply. One heuristic for positioning instruction 6 is to minimise the sum of the lengths (in lines) of all live ranges of program variables (here `A`, `B`, `C`, `D`, `Z`). Calculate the values of this heuristic for the given program, and also for program Q obtained by following P with `10: PRINT A-B`. (For this purpose, it is sufficient to assume instructions 3 to 5 are simple "no-operations" reading and writing no variables.) [4 marks]

(iii) Now calculate the minimal values of this heuristic based on swapping instruction 6 with one of instructions 3 to 5 for the two programs. Hence, or otherwise, give a one-sentence refinement to your answer in part (a). [5 marks]