Optimising Compilers

An expression is very busy at a program point \( n \) if, no matter what path is taken from \( n \), some occurrence of the expression is always evaluated before any of the variables appearing in it are redefined. A transformation using Very Busy Expression (VBE) analysis is to evaluate the expression at \( n \) and store its value for later use.

(a) Give dataflow equations for, and a pseudo-code algorithm to calculate, VBE for a program in flowgraph form. State whether your dataflow equations are forwards or backwards. Sketch the above transformation which exploits VBE in more detail. [11 marks]

(b) Show how to calculate the call graph of a program, and explain the safety property your call graph should have. (You should relate the call graph you define to possible execution behaviour.) Detail how you handle procedure-valued variables, and state whether it is possible to improve on the technique you have chosen for such variables. [6 marks]

(c) Argue how feasible it is to calculate the call graph for a Java program, considering carefully the case of inheritance and use of the final keyword. [3 marks]