1998 Paper 9 Question 7

Optimising Compilers

Consider the programming language with terms e having abstract syntax:

$$e ::= x \mid c \mid \lambda x.e \mid e_1 e_2 \mid \ let\ x = e_1 \ \ in\ e_2$$

where x ranges over a set of identifiers and c over a set of integer constants. For the rest of the question, your answers can be illustrated by reference to the program p:

$$\lambda z.let\ id = \lambda x.x\ in\ id\ id\ 7$$

State how to label terms in p uniquely so that a subterm occurring repeatedly in a term has different labels. [4 marks]

Show how such terms may be seen as a family of flowgraphs, one for each λ (you may find it useful to consider the above labelling as providing a unique function name for anonymous λ-abstractions). [4 marks]

Define the *call graph* of such a family of flowgraphs, stating clearly how indirect calls are treated. [4 marks]

Describe how to associate a flow-variable with each labelled node of a term such as p and to derive equations which can improve the above treatment of indirect calls to get a better approximation of the edges in the call graph. [8 marks]

[Hint: you may find it useful to recall the shorthand of $(\gamma \mapsto \delta) \supseteq \beta$ as representing the compound constraint that

$$\text{whenever } (\lambda x^j.e^k)^i \in \beta \text{ we have } \alpha_j \supseteq \gamma \wedge \delta \supseteq \alpha_k$$

where α_r is the flow variable associated with the node labelled r.]