2002 Paper 8 Question 7

Optimising Compilers

Consider the ML-like language given by abstract syntax

$$e ::= x \mid \lambda x.e \mid e_1\ e_2 \mid \mathit{ref}\ e \mid !e \mid e_1 := e_2 \mid \mathit{if}\ e_1\ \mathit{then}\ e_2\ \mathit{else}\ e_3$$

where x ranges over variable names. A previous compiler phase has already determined a type for each variable and has checked the program is well-typed, where types have syntax

$$\tau ::= \mathit{int} \mid \mathit{intref} \mid \tau \to \tau'.$$

Note that there is no polymorphism and moreover if-then-else uses an integer rather than a boolean for a condition and the result of an assignment is the *int* value assigned rather than a unit value.

(*a*) Give an *effect system* (also known as an annotated type system) in which we can derive judgements of the form

$$\Gamma \vdash e : t, F$$

where t is an extended form of τ and Γ is a set of assumptions of the form $x : t$. Effects F are subsets of $\{A, R, W\}$ representing that the side-effects of evaluating e may respectively include a *ref* allocation (A), a dereferencing read (R) or an update (W). [12 marks]

(*b*) Give types and effects for the following programs, commenting briefly on any issues or problems your scheme encounters and how they may be resolved. (Assume g represents a global variable of type *intref*.)

- $\mathit{if}\ !g\ \mathit{then}\ \mathit{ref}\ 1\ \mathit{else}\ g$
- $\lambda x.\mathit{if}\ !g\ \mathit{then}\ \mathit{ref}\ x\ \mathit{else}\ g$
- $\mathit{if}\ !g\ \mathit{then}\ \lambda x.\mathit{ref}\ x\ \mathit{else}\ \lambda x.g$ [8 marks]