2006 Paper 9 Question 8

Optimising Compilers

Consider the ML-like language given by abstract syntax

$$e ::= x \mid n \mid \lambda x.e \mid e_1\ e_2 \mid \mathit{if}\ e_1\ \mathit{then}\ e_2\ \mathit{else}\ e_3 \mid \mathit{store}\ e\ \mathit{in}\ r \mid \mathit{load}\ e\ \mathit{from}\ r$$

where x ranges over variable names, n over integer constants, and r over global names for disjoint areas of memory known as *regions*. This language allows values to be stored inside regions: $\mathit{store}\ e\ \mathit{in}\ r$ writes the value of e at some newly allocated memory location within region r and returns a pointer to this new location; the complementary operation $\mathit{load}\ e\ \mathit{from}\ r$ reads the value which e points to (provided e is indeed a pointer into region r, otherwise the operation fails without accessing r).

Types have syntax

$$\tau ::= \mathit{int} \mid \tau \to \tau \mid *\tau\ \mathit{in}\ r$$

where $*\tau\ \mathit{in}\ r$ is the type of a pointer to a τ-typed value stored in region r. Note that there is no polymorphism and that *if-then-else* uses an integer (rather than boolean) condition.

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

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

where t is an extended form of τ and Γ is a set of assumptions of the form $x : t$. Effects φ are sets of region names representing the regions which e may need to access (i.e. write into or read from) during its execution.

[12 marks]

(b) Give types and effects for the following expressions, commenting briefly on any problems your scheme encounters and how they may be resolved. (Assume that r and s are region names, x is a variable of type $*\mathit{int}\ \mathit{in}\ r$, and p is a variable of type $*\mathit{int}\ \mathit{in}\ s$.)

(i) $\mathit{if}\ \ \mathit{load}\ x\ \mathit{from}\ r\ \ \mathit{then}\ \ \mathit{store}\ 42\ \mathit{in}\ s\ \ \mathit{else}\ \ p$ [2 marks]

(ii) $\lambda y.\ \mathit{if}\ \ \mathit{load}\ x\ \mathit{from}\ r\ \ \mathit{then}\ \ \mathit{store}\ y\ \mathit{in}\ s\ \ \mathit{else}\ \ p$ [2 marks]

(iii) $\mathit{if}\ \ \mathit{load}\ x\ \mathit{from}\ r\ \ \mathit{then}\ \ \lambda y.\ \mathit{store}\ y\ \mathit{in}\ s\ \ \mathit{else}\ \ \lambda y.\ p$ [4 marks]