Overview of results EPSRC-grant "The eXplicit Substitution Linear Abstract Machine"

June 2000

The goal of the project was to use recent advances in the foundations of functional programming to design and implement *linear* abstract machines, whose correctness could be proved clearly and readily. The project developed both the theory and implementation of linear abstract machines, using and refining the basic tools of (i) *categorical semantics* for linear lambda-calculi, and (ii) the technique of *explicit substitutions.*

Linear functional programming, with its promise of efficient memory management through avoiding garbage collection for variables used in a linear (once-only) way, has been a tantalising but remarkably elusive goal. The project succeeded in removing many conceptual and technological obstacles to this goal, as well as providing novel insights into other areas of functional programming such as computation in stages. The results of the project have received wide international dissemination, with 18 journal and conference papers and three workshops organised as a direct result of the project, plus further more indirectly related publications.

The final design of the linear abstract machine differed significantly from that envisaged in the project proposal, which was implemented and tested in its early stages. While linearity ensures that a variable only occurs once, it does not ensure that the storage space allocated to the variable is used only once. This realization posed a major barrier to our initial aim of using an update-in-place mechanism in the abstract machine. After several attempts at using partial evaluation, a solution was found by devising a formalism that separates occurrence from usage: usage information about memory locations is linear, but individual variables are not. This theoretical advance came too late in the project to allow the implementation and testing of a revised abstract machine.

Other respects in which the project met and exceeded its objectives include:

(i) Obtaining a generic criterion for termination of explicit substitutions. This answered a long-standing open problem by the transfer of the reducibility method to calculi with explicit substitutions.

(ii) Defining a linear lambda-calculus with explicit substitutions with all the usual meta-theoretic properties together with a categorical semantics.

(iii) A more general investigation of linear lambda-calculi and their categorical models. We analysed the semantics of alternative versions of linear λ-calculi. We established that apart from soundness and completeness of the categorical models we need to describe their internal languages to be able to explain the relationship between the models of these λ-calculi.

(iv) Linear recursion: We defined a calculus with a linear natural number object and established a bound for the number of reductions possible in this calculus.