A Denotational Semantics for a polymorphic Effects Systems

A PartIII project proposal

A. J. Taylor (*at736*), St John's College

Project Supervisor: Prof Alan Mycroft

Abstract

A category theoretic approach to build a graded monad based denotational semantics for a polymorphic effects system.

1 Introduction, approach and outcomes (500 words)

Provide an introduction to your project or essay. In particular, try to motivate the work and explain the relevant context (general background, as well as sufficient detail about any related work).

Modelling effects of a program is crucial in modern optimising compiler design. It allows statements to be reordered or pruned and simplification of program structure.

Denotational semantics allows "full program" analysis

There already exist denotational semantics for non-polymorphic effects systems using graded monads, but this may be improved by the use of polymorphism (c.f "Theorems for free" in polymorphic type systems)

Although semantics for polymorphic types is hard (russell's paradox) it is likely to be easier for effects as there is a lack of self recursion.

Deliverable

2 Workplan (500 words)

2^{nd}December - 15^{th}December	Construct a simple graded-monadic lambda calculus based language with a type system and operational semantics. This language shall be designed such that effect polymorphism can be appended onto the core in an easy and intuitive way. I expect that I shall take the route of having an explicit graded monad in the language, and polymorphism shall be added in a similar fashion to the polymorphic lambda calculus with explicit generalisation and specialisation terms.
16^{th}December - 29^{th}December	Prove simple properties of operational semantics without effect polymorphism. These shall include Type Preservation, Progress, Type Safety.
30^{th}December - 12^{th}January	Characterise an abstract model for the language in category theory using cartesian closed categories. This shall be performed in a similar fashion to Andrew Pitts' example for STLC and the original paper by E. Moggi.
13^{th}January - 26^{th}January	Add effect polymorphism to the language and extend the proofs of simple operational properties to the new polymorphic language.
27^{th}January - 9^{th}February	Extend denotational model to polymorphic language. I shall attempt to add morphisms between terms and their generalised equivalents ($gen : [\![T]\!] \to [\![\forall\phi.T]\!]$) and between polymorphic terms and their specialised equivaluents ($spec : [\![\forall\phi.T]\!] \to [\![T[\epsilon/\phi]]\!]$)
10^{th}February - 23^{rd}February	
24^{th}February - 9^{th}March	Continue extension of denotations, aiming to formalise and prove the standard properties of a denotational semantics (Soundness, Adequacy, equal denotations $\Rightarrow$ contextual equivalence
10^{th}March - 23^{rd}March	Extensions.
24^{th}March - 6^{th}April	Collation of results.
7^{th}April - 20^{th}April	Write dissertation.
21^{st}April - 4^{th}May	
5^{th}May - 18^{th}May	Contingency and hand in.
19^{th}May - 31^{st}May	