

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, 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)

November	Plan + read
December	Construct a simple monadic lambda calculus based language with a type system and operational semantics, prove simple properties of operational semantics
January	Characterise an abstract model for the language
February	Add polymorphism over effects to this general model
March	Contingency
April	Collate results
May	write dissertation