

# A Denotational Semantics for a polymorphic Effects Systems

## *A Part III 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 language with polymorphism over effects.*

## **1 Introduction, approach and outcomes (500 words)**

*A denotational semantics of a programming language is an extremely useful tool for static program analysis. Since Scott and Strachey's original work in the 1970s, denotational models have been developed for increasingly complex languages, such as those that include recursion<sup>1</sup>, effects<sup>2</sup>, and polymorphic types<sup>3</sup>. Little to no work, however, has been published on the denotational semantics of polymorphic effects. Allowing such polymorphism is useful as it generalises effects and may lead to better reasoning about functions in an effectful language. (C.F. Theorems for Free<sup>4</sup> for a similar phenomenon with polymorphic types.)*

The approach I intend to take is to construct a toy lambda calculus based language with a graded monad to handle effects. After proving simple properties of the language, I shall construct a category theoretic description of an abstract model of this language in a manner similar to that of Moggi<sup>5</sup> and Andrew Pitts<sup>6</sup>. Once this is performed, I shall attempt to attach effect polymorphism to the abstract model of the language. In doing so I hope to characterise what constraints need to apply to such a model to achieve this, or on failure what a solution would look like.

Although a denotational semantics for polymorphic types is hard due to a manifestation of Russell's paradox, there is good reason to expect that the semantics of effects is easier to formalise due to effects not being self referential.

**Deliverable** The main deliverable of this project will be the presentation of a valid denotational model for the aforementioned language, along with proofs of the normal denotational semantics theorems (Soundness, Compositionality, Adequacy).

---

<sup>1</sup>Where?

<sup>2</sup>Where?

<sup>3</sup>Where?

<sup>4</sup><https://people.mpi-sws.org/~dreyer/tor/papers/wadler.pdf>

<sup>5</sup><https://core.ac.uk/download/pdf/21173011.pdf>

<sup>6</sup><https://www.cl.cam.ac.uk/teaching/1617/L108/catl-notes.pdf>

## 2 Workplan (500 words)

2 <sup>nd</sup> December - 15 <sup>th</sup> 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 <sup>th</sup> December - 29 <sup>th</sup> December	Prove simple properties of operational semantics without effect polymorphism. These shall include Type Preservation, Progress, Type Safety.
30 <sup>th</sup> December - 12 <sup>th</sup> 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. This step should be relatively simple as this work has been done before. This chunk also corresponds with the hand in for several modules of taught courses.
13 <sup>th</sup> January - 26 <sup>th</sup> January	Add effect polymorphism to the language and extend the proofs of simple operational properties to the new polymorphic language. This is the first section of the project to steer into novel territory.
27 <sup>th</sup> January - 9 <sup>th</sup> February	Extend denotational model to polymorphic language. I shall attempt to add morphisms between terms and their generalised equivalents ( $gen : \llbracket T \rrbracket \rightarrow \llbracket \forall \phi. T \rrbracket$ ) and between polymorphic terms and their specialised equivalents ( $spec : \llbracket \forall \phi. T \rrbracket \rightarrow \llbracket T[\epsilon/\phi] \rrbracket$ )
10 <sup>th</sup> February - 23 <sup>rd</sup> February	
24 <sup>th</sup> February - 9 <sup>th</sup> March	Continue extension of denotations, aiming to formalise and prove the standard properties of a denotational semantics (Soundness, Adequacy, equal denotations $\Rightarrow$ contextual equivalence). This is likely to be the hardest part of the project, and may require iteration with the previous step to refine the denotational model.
10 <sup>th</sup> March - 23 <sup>rd</sup> March	Extensions. This chunk is deliberately left vague as it not yet known what I might find in the previous step. Potential contents here might include instantiating the abstract model to model particular concrete languages with particular effects, finding deficiencies in categorical approaches which prevent the construction of such a denotational system, or further generalisation to include type polymorphism.
24 <sup>th</sup> March - 6 <sup>th</sup> April	Collation of results and optimisation of proofs for presentation
7 <sup>th</sup> April - 20 <sup>th</sup> April	Write dissertation, including iterations due to feedback.
21 <sup>st</sup> April - 4 <sup>th</sup> May	
5 <sup>th</sup> May - 18 <sup>th</sup> May	Contingency and hand in.
19 <sup>th</sup> May - 31 <sup>st</sup> May	