A Denotational Semantics for a polymorphic Effect 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 effects[1] and polymorphic types[2]. No work, however, has been published to date on the denotational semantics of polymorphic effect systems. Allowing such polymorphism may be useful to generalise over effects and open the door to better reasoning about functions in an effectful language. (Cf. Theorems for Free[3] 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 similar to Moggi's monadic metalanguage[4]. After proving simple properties of the language, such as the weakening and substitution lemmas, and theorems related to type safety, I shall construct a category theoretic description of an abstract semantics of this language in a manner similar to that of Moggi [5]. Once this is performed, I shall attempt to attach effect polymorphism to the abstract semantics of the language. In doing so I hope to describe what constraints need to apply to such a model to achieve this, or on failure what a solution would look like.

Although a denotional semantics for polymorphic types is hard due to a manifestation of Russell's paradox[6], 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 the presentation of a sound denotational semantics for the aforementioned language, along with proofs of the normal denotational semantics theorems. These theorems are soundness, compositionality, and potentially adequacy for individual concrete cases).

[1]Moggi, Notions of computation and monads: https://core.ac.uk/download/pdf/21173011.pdf
[2]Jacobs, Categorical Logic and Type Theory, Chapter 8
[3]https://people.mpi-sws.org/ dreyer/tor/papers/wadler.pdf
[4]Moggi, Notions of computation and monads: https://core.ac.uk/download/pdf/21173011.pdf
[5]https://core.ac.uk/download/pdf/21173011.pdf
[6]"Polymorphism is not set-theoretic" by Reynolds. https://hal.inria.fr/inria-00076261/document

2 Workplan (500 words)

2nd December - 15th 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.
16th December - 29th December	Prove simple properties of operational semantics without effect polymorphism. These shall include the Weakening and Substitution Lemmas, Type Preservation, Progress, Type Safety.
30th December - 12th 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.
13th January - 26th 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.
27th January - 9th February	Extend denotational model to the polymorphic language.
10th February - 23rd February	
24th February - 9th 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.
10th March - 23rd 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 intantiating the abstract model to model particular concrete languages with particular effects, proving adequacy for a particular concrete model and language, finding deficiencies in categorical approaches which prevent the construction of such a denotational system, or further generalisation to include type polymorphism.
24th March - 6th April	Collation of results and optimisation of proofs for presentation
7th April - 20th April	Write dissertation, including iterations due to feedback.
21st April - 4th May	
5th May - 18th May	Contingency and hand in.
19th May - 31st May	