Programming an institution in Haskell

Marie Farrell
Principles of Programming Research Group, Department of Computer Science,
Maynooth University

Supervisors: Dr. Rosemary Monahan & Dr. James Power

mfarrell@cs.nuim.ie

Abstract:

A common problem in the field of Software Verification is the sheer abundance of formalisms, because of this, the software engineer is faced with the daunting task of deciphering which formalism best suits a particular project. This task is complicated by the fact that it may be necessary to verify properties of the system that no formalism facilitates in isolation. Therefore the developer must utilise multiple formalisms. This can result in the system being modelled numerous times or, more frequently, the system is decomposed to allow subcomponents thereof be verified in separate formalisms. Often, the same proofs will need to be carried out multiple times. The task is further complicated by the use of refinement which facilitates the modelling of a system at different levels of abstraction. The aim of this PhD is to provide a framework within which proofs and refinement steps can be shared across formalisms and our thesis is that the theory of institutions can provide such a framework.

The theory of institutions observes that once the syntax and semantics of a formal system have been defined in a uniform way, using some basic constructs from category theory, then a set of specification building operators can be defined that allow you to write, modularise and build up specifications that can be defined in a formalism-independent manner. Event-B is a formal specification language that enables the user to prove safety properties of a specification. It uses set theory as notation and facilitates the modelling of systems at different levels of abstraction through the verifiable process of refinement. Our current goal is to represent the Event-B formalism in terms of institutions and thus provide modularisation constructs which will increase the scalability of Event-B for use in larger projects. Another benefit of this approach is the increased interoperabilty of Event-B via institution comorphisms which will allow aspects of the system be specified in different formalisms and included in the final Event-B specification.

We have devised our institution for Event-B (EVT) as somewhat of an extension to the institution for first order predicate logic with equality (FOPEQ). This talk will provide an introduction to FOPEQ and its implementation as Haskell code.

This project is funded by the Irish Research Council and supported by the John & Pat Hume Scholarship.