3rd Wessex Theory Seminar

The third meeting of the Wessex Theory Seminar took place on Tuesday 3rd and Wednesday 4th March, co-located between Bath (3rd March) and Swansea (4th March).

Lectures on the 3rd were held in Rooms 1WN 3.12 (11:15 - 3:05), 8W 2.21 (3:15-4:05), and 8W 2.20 (4:15-5:05) of the University of Bath's Claverton (Bath) campus; lectures on the 4th were given in the Board Room (Far-314), Faraday Building, Swansea University.

Lunch and coffee were provided.

Talks were expected to last 35-40 minutes, giving time for questions and change-over. The schedule of the two days was as follows, with abstracts following below.

Programme

March, 3rd, Bath:

11:15 Makoto Takeyama: Mini-TT

12:00 John Longley: Eriskay: a Programming Language based on Game Semantics

1:00 Lunch

2:00 Yoshiki Kinoshita: Overview of DEOS dependability standard project

2:45 Coffee

3:15 Anton Setzer: Coalgebras and Codata in Agda

4:15 Martin Churchill: A Concrete Representation of Observational Equivalence for PCF

March, 4th, Swansea:

12:15 Yoshiki Kinoshita: Introduction to AIST, CVS and CFV

1:00 Lunch

2:00 Makoto Takeyama: Model-based Testing of System LSI using Agda

2:45 Coffee

3:00 John Power: Towards a Geometric Foundation for Game Semantics

3:45 Coffee

4:00 Yoshiki Kinoshita: Applications of Agda

Attendance

From AIST (Japan):

Yoshiki Kinoshita
Makoto Takeyama

From Bath:

Ana Carolina Martins Abbud
Martin Churchill
Adam Gundry
Dalia Khader
Jim Laird
Guy McCusker
John Power

From Edinburgh:

John Longley

From Southampton:

Ross Horne

From Swansea:

Arnold Beckmann
Ulrich Berger
Jens Blanck
Min Chuang
Matthew Gwynne
Roger Hindley
Phil James
Karim Kanso
Oliver Kullmann
Ebrahim Larijani
Faron Moller
Peter Mosses
Mark New
Liam O'Reilly
Markus Roggenbach
Monika Seisenberger
Anton Setzer

Abstracts

Makoto Takeyama: Mini-TT (Joint with Thierry Coquand, Bengt Nordstrom and Yoshiki Kinoshita)

We present Mini-TT, a small functional language with dependent types. Mini-TT is a step towards a simple and definitive core language for the proof-assistant Agda based on versions ofMartin-L¿of Type Theory. It is difficult to give a semantics directly to the full Agda language with advanced features such as synthesis of implicit arguments. Mini-TT will be used as the target of a translation from the full language, with which the Agda language will be specified and the implementation verified. Mini-TT contains data types, mutual recursive / inductive definitions and a universe of small types. The syntax, semantics and type system is specified in such a way that the implementation of a parser, interpreter and type checker is straightforward (around 400 lines in Haskell).

John Longley (joint work with Nicholas Wolverson): Eriskay: a Programming Language based on Game Semantics

I will describe an ongoing project to design a class-based object oriented language based around ideas from game semantics. Part of our goal is to create a powerful modern programming language whose clean semantic basis renders it amenable to work in program verification; however, we argue that our semantically inspired approach also yields benefits of more immediate relevance to programmers, such as expressive new language constructs and novel type systems for enforcing security properties of the language.

Our work is based on a simple game model due to Lamarche, when endowed with a suitable linear exponential, suffices for modelling stateful objects, higher order functions, coroutining, recursive types, polymorphism with subtyping, a class system with inheritance and dynamic binding, and even ? seen in a certain light ? such features as fresh name generation and higher-order store. I will explain in general terms how this model may be used to guide the design of a language, and will then focus on three specific areas where our approach appears to offer something new:

1. Linear types and coroutining operators.
2. Static control of the use of higher-order store, and how this helps with the encapsulation of computational effects such as exceptions.
3. Higher order programming with classes, including a full abstraction result for class implementations.

Yoshiki Kinishita: Overview of DEOS standardisation project

Our project "User Oriented Dependability" started in October 2008, as one of the project of the programme DEOS (Dependable Embedded Operating System), which is conducted under CREST scheme of JST (Japan's Agency for Science and Technology). We outline our plan of the project in four items: (1) clarification of the concept of user oriented dependability, (2) standardisation of the concept clarified in (1), (3) providing guideline for conformance evaluation and (4) providing guideline for development of dependable system lifecylces.

Anton Setzer: Coalgebras and Codata in Agda

Weakly final coalgebras have recently been added to Agda on the basis of the concept of codata. We discuss the problems of this representation, and suitable variants based on the categorical concept of a coalgebra. We investigate how to obtain most of the benefits of the codata type while keeping the conceptual clarity of categorical coalgebras.

We then give examples of how to use coinduction for carrying out proofs on coalgebras in the context of bisimulation, which are more straight forward than using the usual approaches based on writing bisimulation relations.

Martin Churchill (joint work with Jim Laird and Guy McCusker): A Concrete Representation of Observational Equivalence for PCF

The full abstraction result for PCF using game semantics requires one to identify all innocent strategies that are "innocently indistinguishable". This involves a quantification over all innocent tests, cf. quantification over all innocent contexts. Here we present a concrete representation of innocent strategies (and hence PCF terms) that equates precisely the terms that are observationally equivalent without any need for such quotienting.

Yoshiki Kinoshita: Introduction to AIST, CVS and CFV

I will overview our affiliation AIST (National Institute of Advanced Industrial Science and Technology) and its research center CVS (Research Center for Verification and Semantics), as well as recently started facility CFV (Collaborative Facilities for Verification). CVS is the research

center where we work and it has 39 members in total as of today, including support staffs. CFV are the facilities we are building in AIST. It consists of a cluster computer designed for symbolic computing such as model checking and automatic theorem proving, as well as human resources including supporting body run by 'Kansai Economic Federation,' of which NTT West, Panasonic, Sharp, Kyocera, Omron are active member.

Makoto Takeyama: Model-based testing of System LSI using Agda (Joint with CVS team and Renesas team)

This is a joint project with Renesas Technology Corp., a major Japanese LSI manufacturer. The design process of a CPU traditionally starts with an informal specification in a combination of a natural language and some pseudo programming language, which is then used to generate items for verification of lower-level detailed designs. Engineers generating those verification items are facing the usual problems of informality. Ambiguities, implicit assumptions, inconsistencies, etc. lead to wrong verification items or critical omission. The generation is basically a manual process that is not only error prone but also very costly. We report on how these problems are alleviated through the formalisation and automation developed in our project, as well as our experience in introducing a Formal Method to an industrial environment where people were new to that kind of FM.

John Power (joint work with Guy McCusker): Towards a Geometric Foundation for Game Semantics

Geometry has long played a foundational role in the description of categories with variants of monoidal structure. For instance, the free braided monoidal category on 1 is given by the category of braids, i.e., the category of downward-pointing strings in 3-space, and the free tortile category on 1 is given by a category of ribbons. Researchers in game semantics also draw pictures to describe maps in categories of games, but their formalism for categories of games has been combinatorial. So, in the spirit of the development of braided monoidal categories, we refine the combinatorial foundation for game semantics by giving a geometric one that more closely resembles blackboard practice. Specifically, we give a geometric account of the categories of schedules and O-heaps.

Yoshiki Kinoshita: Applications of Agda

This is a very informal introduction to on-going applications of Agda, done as research projects in CVS. I can show some code if necessary, although I cannot go into detail as I myself is not writing them.