6th Wessex Theory Seminar

The sixth meeting of the Wessex Theory Seminar returned to Southampton on Friday 19th February 2010. Talks took place in the seminar room on the 4th floor, Mountbatten building, Highfield campus, Southampton University. The event was fully catered with lunch and coffee.

Programme

February, 19th, Southampton:

10:15-10:45 Welcome coffee

Morning session

10:45-11:30 Paulo Oliva: Selection Functions in Proof Theory .

11:30-12:30 Max Kanovich: How and Why Purely Propositional Separation Logic is undecidable .

12:30-1:30 Lunch

Afternoon session

1:30-2:15 Dirk Pattinson: Global Caching for Coalgebraic Description Logics .

2:15-2:50 Mark Wheelhouse: High Level Program Reasoning .

2:50-3:15 Afternoon tea

Final session

3:15-4:15 Andrzej Murawski: Full Abstraction for Reduced ML .

4:15-5:00 Kohei Honda: The pi-calculus in the Real World .

Attendance

From Southampton:

- Julian Rathke
- Pawel Sobocinski
- Ross Horne
- Sardaouna Hamadou
- Corina Cirstea
- Issam Maamria
- Toby Wilkinson
- Gabrielle Anderson
- Ehab ElSalamouny
- Issam Souliah
- Tope Omitola

From Imperial:

- Dirk Pattinson
- Mark Wheelhouse
- James Brotherston
- Faris Abou-Saleh

From Queen Mary:

- Paulo Oliva
- Max Kanovich
- Kohei Honda
- Rob Arthan
- Gilda Ferreira

From Bath:

- Guy McCusker

From Oxford:

- Andrzej Murawski

From Swansea:

- Fredrik Nordvall Forsberg

Abstracts

Paulo Oliva

Selection Functions in Proof Theory

By "selection function" we mean an operation that selects a witness to the non-emptyness of any given property/set (if that property/set is indeed non-empty). Hilbert's epsilon terms provide one example of the role played by selection functions in proof theory. In this talk I will show that, when generalised notions of predicate and quantifier are considered, such selection functions have in fact a ubiquitous role in the computational interpretation of proofs in arithmetic and analysis. (Joint work with Martin Escardo)

Max Kanovich

How and Why Purely Propositional Separation Logic is undecidable.

Separation logic has proven as an adequate formalism for the analysis of programs that manipulate memory (in the form of pointers, heaps, stacks, etc.). In addition to the standard propositional connectives, the most important feature of separation logic is its *separating conjunction* (A*B) which holds for a heap h iff h can be split into *separate* h1 and h2 so that A holds for h1 and B holds for h2.

Here our main focus is on the logic for *concrete* heap-like models of practical interest.

Via an encoding of Minsky machines, we show that:

- It is undecidable whether a purely propositional formula A is valid in the class of all (total) separation models.
- It is undecidable whether a purely propositional formula A is valid in the class of all separation models with indivisible units.
- Furthermore, whatever *concrete* heap-like model H we take, it is undecidable whether a purely propositional formula A is valid in this model H.

On top of that, our undecidability results for *concrete* heap-like models give new insights into the nature of decidable fragments of separation logic such as those given by Calcagno-Yang-O'Hearn(2001) and Berdine-Calcagno-O'Hearn(2004), as well as imposing boundaries on decidability.

E.g., we can deduce that to obtain decidability in a heap-like model, one should either give up infinite interpretations for atomic propositions (as in Calcagno-Yang-O'Hearn) or restrict the formula language (as in Berdine-Calcagno-O'Hearn).

We prove undecidability for a number of natural propositional systems developed on the road towards axiomatization of separation logic, such as

- Boolean BI,
- Boolean BI enriched with a restricted *-weakening,
- Classical BI,
- Classical BI enriched with the restricted *-weakening.

Notice that adding the unrestricted *-weakening to Boolean BI (as well as to Classical BI) forces a collapse into the standard Boolean logic, which is decidable.

To add a new exhibit to the Undecidability Zoo, we show the simplest undecidable purely propositional system, we call it Minimal BI, which employs only two conjunctions, that is * and &, and their two adjoint implications, respectively. (Neither negation nor `false' should be blamed for its undecidability)

(See also http://www.doc.ic.ac.uk/research/technicalreports/2010/)

This is joint work with James Brotherston (Imperial)

Dirk Pattinson

Global Caching for Coalgebraic Description Logics
Coalgebraic description logics offer a common semantic umbrella for extensions of description logics with reasoning principles outside relational semantics, e.g. quantitative uncertainty, non-monotonic conditionals, or coalitional power. Specifically, we work in coalgebraic logic with global assumptions (i.e. a general TBox), nominals, and satisfaction operators, and prove soundness and completeness of an associated tableau algorithm of optimal complexity EXPTIME. The algorithm is based on global caching, which raises hopes of practically feasible implementation. Instantiation of this result to concrete logics yields new algorithms in all cases including standard relational hybrid logic.

This is joint work with Rajeev Gore (ANU), Clemens Kupke (Imperial) and Lutz Schroeder (DFKI Bremen).

Mark Wheelhouse

High Level Program Reasoning

O'Hearn, Reynolds and Yang introduced Separation Logic to provide modular reasoning about simple, mutable data structures in memory. They were able to construct small specifications of programs by reasoning about the local parts of memory accessed by programs (their footprints). They have used their Low-Level reasoning to notable success verifying memory safety properties of C programs and device drivers.

Gardener, Calcagno and Zarfaty have generalised this work, introducing Context Logic to reason about more complex, High-Level, data structures in order to reason at the level of the program abstraction. In particular, with Smith and Wheelhouse, they have developed a formal, compositional specification for the Document Object Model (DOM) a W3C XML update library. However, whilst keeping to the spirit of local reasoning, they were not able to retain small specifications for all of their update commands.

We have since introduced Segment Logic, which provides a more fine-grained analysis of data structures and yields small specifications for all of our update commands. As well as being aesthetically pleasing, small specifications are essential for reasoning about concurrent update programs.

In this survey talk we shall take a look at the progression of program verification from Low-Lewel to High-Level reasoning. We will begin with a quick look at Separation Logic before introducing Context Logic and showing how this enables us to specify the High-Level DOM commands without needing to know how they are implemented at the Low-Level. We shall then see how these specifications are unsuitable for reasoning about a concurrent XML update language, and look at how Segment Logic can be used to provide a more fine-grained specification of these commands that can be used in a concurrent setting.

Andrzej Murawski

Full Abstraction for Reduced ML

We present the first effectively presentable fully abstract model for Stark's Reduced ML, the paradigmatic higher-order programming language combining call-by-value evaluation and integer-valued references.

The model is constructed using techniques of nominal game semantics. Its distinctive feature is the presence of carefully restricted information about the store in plays, combined with conditions concerning the participants' ability to distinguish reference names. This leads to an explicit characterization of program equivalence.

This is joint work with Nikos Tzevelekos.

Kohei Honda

The pi-calculus in the Real World

Along with technical stuff, I hope I can discuss why my industry colleagues have come to like types, logics and bisimulations.