MPhil in Advanced Computer Science

Semantics of HOT Languages

Leader:	Prof. AM Pitts
Timing:	Lent
Prerequisites:	Set Theory for Computer Science
Structure:	16 Lectures

AIMS

The aim is to cover some advanced topics and techniques that address the question: "what does it mean for two programs in a higher-order typed (HOT) programming language to be *equal* and how do we prove such equalities?" The emphasis will be on operationally-based techniques, but since several of those have their origins in domain theory and denotational semantics, those topics will also be covered to some extent. This module assumes some familiarity with basic material on the semantics of programming languages (such as provided by CST IB Semantics of Programming Languages and CST II Denotational Semantics).

SYLLABUS

- Motivating examples for studying programming language semantics in general and equivalence of programs in particular. Operational and denotational notions of semantic equivalence. [2L]

- Operational semantics.

  Higher order recursive functions with local sate. A fragment of OCaml. Styles of operational semantics. Contextual equivalence. Proving ML contextual equivalence via logical relations. [6L]

- Denotational semantics.

  The need to solve recursive domain equations. Locally continuous functors of mixed variance on the category of pointed ω-cpos and strict continuous functions. The existence and uniqueness of minimal invariants. Freyd's characterization of recursively defined domains. Application: computational adequacy results via logical relations between denotational and operational semantics. [7L]

- Survey of recent techniques for proving contextual equivlance in HOT languages. [1L]

OBJECTIVES

On completion of this module, students should be in a position to start a PhD involving the semantics of higher-order typed programming languages.

COURSEWORK

None.

PRACTICAL WORK

None.

ASSESSMENT

A written test will be set. The final module mark will be in the form of a percentage.

RECOMMENDED READING

Lectures 1–9:
A. M. Pitts, *Operational Semantics and Program Equivalence.* In: G. Barthe, P. Dybjer and J. Saraiva (Eds), *Applied Semantics.* Lecture Notes in Computer Science, Tutorial, Volume 2395 (Springer-Verlag, 2002), pages 378-412.

Lectures: 10-13:
S. Abramsky and A. Jung, *Domain Theory.* Chapter 1 of S. Abramsky and D. M. Gabbay and T. S. E. Maibaum (eds), *Handbook of Logic in Computer Science, Volume 3. Semantic Structures* (Oxford University Press, 1994).

Lectures 14-16:
A. M. Pitts, *Typed Operational Reasoning.* In B. Pierce (ed), *Advanced Topics in Types and Programming Languages*, chapter 7, pages 245-289 (MIT Press, 2005).

[Last updated: 2010-03-02]