

# MPhil in Advanced Computer Science

## Advanced Topics in Programming Languages

- Lecturer: Dr Peter Sewell
- 16 one-hour meetings (8 in Michaelmas, 8 in Lent)

This module provides an introduction to a variety of advanced topics in programming-language and semantics research. Participants are expected to present 30-minute talks based on research papers, and to engage in critical discussion. The module is also intended to foster critical reading of the research literature and skill in research presentation.

Papers will be assigned by the lecturer, after discussion, from a list chosen to cover selected themes, with several papers on each theme. A sample list is below, though it will change from year to year, and be flexible in response to suggestions from the students.

Assessment will be by ticks for presentations and a graded essay discussing one or more of the presented papers.

### Sample Papers

#### 1. Low-Level Semantics

- (a) Either *From system  $F$  to typed assembly language*, Greg Morrisett, David Walker, Karl Crary, Neal Glew, TOPLAS 1999; or *Typed Assembly Language*, Greg Morrisett, ATTAPL.
- (b) *Proof-Carrying Code*, George Necula, ATTAPL.
- (c) *Effect Types and Region-Based Memory Management*, Fritz Henglein, Henning Makholm, Henning Niss, ATTAPL.
- (d) *Safe manual memory management in Cyclone*, Nikhil Swamy, Michael W. Hicks, Greg Morrisett, Dan Grossman, Trevor Jim, Sci. Comput. Program. 62(2): 122-144 (2006)
- (e) *A formally verified compiler back-end*, Xavier Leroy, draft, 2008
- (f) *Formalizing and Verifying Semantic Type Soundness for a Simple Compiler*, N. Benton and U. Zarfaty, PPDP 2007

#### 2. High-level Concurrency

- (a) *Composable memory transactions*, Tim Harris, Simon Marlow, Simon Peyton-Jones, Maurice Herlihy, PPDP 05
- (b) Another Transactions paper - perhaps *High-Level Small-Step Operational Semantics for Transactions*, Katherine F. Moore, Dan Grossman, POPL 2008

- (c) CML. Perhaps from *The Essence of Concurrent ML*. Prakash Panangaden and John Reppy. In Flemming Nielson, editor, *ML with Concurrency*, Chapter 1. Springer-Verlag, 1997.
- (d) Pict and Join Calculus. *Pict: A Programming Language Based on the Pi-Calculus*, Benjamin C. Pierce and David N. Turner. In *Proof, Language and Interaction: Essays in Honour of Robin Milner*, pages 455-494. MIT Press, 2000; *A Calculus of Mobile Agents*, Cédric Fournet, Georges Gonthier, Jean-Jacques Lévy, Luc Maranget, Didier Rémy, CONCUR 1996

### 3. Fancy Types

- (a) *Dependent Types*, David Aspinall and Martin Hofmann, ATTAPL
- (b) Type inference. Either *The Essence of ML Type Inference*, François Potter, Didier Rémy, ATTAPL, or one of the papers on Bidirectional Type Checking
- (c) ML modules. *Manifest types, modules, and separate compilation*, Xavier Leroy, POPL 1994, and *A type-theoretic approach to higher-order modules with sharing*, Robert Harper and Mark Lillibridge, POPL 1994.
- (d) GADTs. Perhaps *Simple unification-based type inference for GADTs*, Simon L. Peyton Jones, Dimitrios Vytiniotis, Stephanie Weirich, and Geoffrey Washburn, ICFP 2006
- (e)  $F\omega_{<}$ . The relevant chapters from *Types and Programming Languages*, Pierce.
- (f) Scala. Perhaps *A Core Calculus for Scala Type Checking*, Vincent Cremet, François Garillot, Serguei Lenglet, Martin Odersky. MFCS 06.

### Preparatory reading

Some familiarity with operational semantics and type systems will be useful, e.g. from the first 15 chapters of [TAPL] and/or the notes for the Computer Laboratory courses on Semantics of Programming Languages, Foundations of Functional Programming, and Types.

### References

- TAPL *Types and Programming Languages*. MIT Press. Benjamin C. Pierce.
- ATTAPL *Advanced Topics in Types and Programming Languages*. MIT Press. Edited by Benjamin C. Pierce.