Extending Bifibrational Models of System F to Effects

presented by Clément Aubert – Department of Computer Science

Abstract of proposed research

Application for Appalachian State University's Research Development Travel Grant

Background The polymorphic lambda calculuus (known as System F) is an elegant, complete, and simple abstraction at the core of functional programming languages like `ML` or `Haskell`. It is used by major companies as Jane Street or Facebook [1], as well as by certified compilers [2].

The study of System F's mathematical interpretation abstracts away syntactical subtleties and reveals its computational nature. In particular, categorical semantics for System F provide deep and important insights into *parametric polymorphism* [3], which restricts the quantification representing polymorphism in System F to "well-behaved", or "generic", polymorphism.

A New and Timely Challenge A growing number of programming languages (including `Haskell`, with its built-in monads) implement effects. Effects allow programs to dynamically take users' actions into account. Different extensions of System F (such as λ_c [4] and PE [5]) sucessfully incorporate effects into System F.

Implementing, handling, and understanding effects is an active topic of research, as witnessed, for instance, by the International Dagstuhl Seminar 16112 that I recently attended. The interaction of polymorphism with effects is not yet completly understood: for example, how to tackle quantification over effects is completely unknown.

Modularly Extending Previous Models A persistent line of work has given birth to a model of System F [6] based on the categorical notion of bifibration. This bifibrational model provides a novel perspective on parametric interpretations of System F by refining the canonical interpretation and giving a new categorical understanding. On the other hand, semantic models of effects (e.g., in terms of monads, Kleisli categories, and Eilenberg-Moore categories) are stable and well-understood.

However, combining bifibrational model of System F and effects remains an open problem. This project therefore aims to construct a bifibrational model of System F's parametric polymorphism that is robust enough to accomodate effects as well. By doing so, this project will:

- develop techniques for assessing different syntactical representations of parametric polymorphism with effects;
- give rise to bifibrational treatment of relational parametricity and effects;
- investigate the aformentioned problem of quantification over effects.

The PI will benefit from Dr. Johann's expertise to write proposals and conduct research.

References

[1] Wikipedia. *OCaml — Wikipedia, The Free Encyclopedia*. url: `https://en.wikipedia.org/wiki/OCaml#Commercial_users_of_OCaml` (visited on 03/21/2016).

[2] L.-J. Guillemette. "A Type-Preserving Compiler from System F to Typed Assembly Language". PhD thesis. Université de Montréal, 2009.

[3] J. C. Reynolds. "Types, Abstraction and Parametric Polymorphism". In: *IFIP Congress*. 1983, pp. 513–523.

[4] K. Asada. "Extensional Universal Types for Call-by-Value". In: *APLAS*. Ed. by G. Ramalingam. Vol. 5356. LNCS. Springer, 2008, pp. 122–137. doi: 10.1007/978-3-540-89330-1_9.

[5] R. E. Møgelberg and A. Simpson. "A Logic for Parametric Polymorphism with Effects". In: *TYPES 2007*. Ed. by M. Miculan, I. Scagnetto, and F. Honsell. Vol. 4941. LNCS. Springer, 2007, pp. 142–156. doi: 10.1007/978-3-540-68103-8_10.

[6] N. Ghani et al. "Bifibrational functorial semantics of parametric polymorphism". In: *MFPS XXXI*. Ed. by D. R. Ghica. Vol. 319. ENTCS. Elsevier, 2015, pp. 165–181. doi: 10.1016/j.entcs.2015.12.011.