An overview of game semantics for polymorphism

Guilhem Jaber[1] and Nikos Tzevelekos[2]

[1] ENS de Lyon, Université de Lyon, LIP (guilhem.jaber@ens-lyon.fr)
[2] Queen Mary University of London (nikos.tzevelekos@qmul.ac.uk)

Abstract. In this talk we aim to present an overview of game semantic models for polymorphic languages (extensions of System F), starting from the hypergame semantics of Hughes and ending in recent trace models based on names.

Game models for Parametric Polymorphism

The first game model for System F was introduced by Hughes [6,5], who introduced the idea of a *hypergame*: a game in which players can play arenas in place of moves. Hughes' model is intentional, in the sense that it is fully complete for $\beta\eta$-equivalence. Starting from that model, de Lataillade [4,3] characterised parametricity categorically via the notion of dinaturality [2]. In [1], Abramsky and Jagadeesan developed a model for System F to characterise genericity, as introduced by Longo, Milstead and Soloviev [13]. A type θ is said to be *generic* when two terms M_1, M_2 of type $\forall X.\theta'$ are considered equivalent as soon as $M_1\theta$ and $M_2\theta$ are equivalent. Their model contains several generic types. More recently, Laird [11] introduced a game model for System F augmented with mutable variables. His model approaches parametricity via *copycat links*: these are constrains, presented in terms of pointers and added to plays to impose parametric behaviour. Similar ideas were used by Laird in order to capture call-by-value polymorphism in [10].

In all of the above models the denotation of terms is built compositionally by induction on the structure of the term. In a different line of work, Jeffrey and Rathke [9] produced a fully abstract trace model for polymorphic π-calculus using LTSs that reduce open terms. This approach in effect builds game models *operationally*, and composition corresponds to proving the model sound. Lassen and Levy [12] introduced normal form bisimulations for a language with parametric polymorphism. These bisimulations are defined on LTSs in the tradition of Jeffrey and Rathke. In recent joint work, following the approaches of [9,11], we advocated modelling parametricity by use of *names*: an abstract type is treated by a receiving term simply as a name, without internal structure, and a similar treatment is given to values of abstract types. Using this approach, we built a trace model for a call-by-value language with ML-like references [7], and proposed a trace semantics for Strachey parametric polymorphism in System F [8].

References

1. S. Abramsky and R. Jagadeesan. A game semantics for generic polymorphism. *Annals of Pure and Applied Logic*, 133(1):3–37, 2005. Festschrift on the occasion of Helmut Schwichtenberg's 60th birthday.
2. E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott. Functorial polymorphism. *Theor. Comput. Sci.*, 70(1):35–64, Jan. 1990.

3. J. de Lataillade. *Quantification du second ordre en sémentique des jeux: application aux isomorphismes de types*. PhD thesis, Paris 7, 2007.
4. J. de Lataillade. Second-order type isomorphisms through game semantics. *Annals of Pure and Applied Logic*, 151(2-3):115–150, 2008.
5. D. Hughes. *Hypergame semantics: full completeness for system F*. PhD thesis, D. Phil. thesis, Oxford University, 2000.
6. D. J. D. Hughes. Games and definability for system f. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science*, LICS '97, pages 76–, Washington, DC, USA, 1997. IEEE Computer Society.
7. G. Jaber and N. Tzevelekos. Trace semantics for polymorphic references. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 585–594, 2016.
8. G. Jaber and N. Tzevelekos. A trace semantics for System F parametric polymorphism. In *Proceedings of FOSSACS'18 (to appear)*, 2018.
9. A. Jeffrey and J. Rathke. Full abstraction for polymorphic pi-calculus. *Theor. Comput. Sci.*, 390(2-3):171–196, 2008.
10. J. Laird. Game semantics for call-by-value polymorphism. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, pages 187–198, 2010.
11. J. Laird. Game semantics for a polymorphic programming language. *J. ACM*, 60(4):29:1–29:27, Sept. 2013.
12. S. B. Lassen and P. B. Levy. Typed normal form bisimulation for parametric polymorphism. In *Proceedings of the 2008 23rd Annual IEEE Symposium on Logic in Computer Science*, LICS '08, pages 341–352, Washington, DC, USA, 2008. IEEE Computer Society.
13. G. Longo, K. Milsted, and S. Soloviev. The genericity theorem and parametricity in the polymorphic lambda-calculus. *Theor. Comput. Sci.*, 121(1-2):323–349, Dec. 1993.