A Type Theory for Comprehensive Parametric Polymorphism

Neil Ghani[1], Fredrik Nordvall Forsberg[1], and Alex Simpson[2]

[1] Department of Computer and Information Sciences, University of Strathclyde, UK
[2] Faculty of Mathematics and Physics, University of Ljubljana, Slovenia

Introduction

A polymorphic program is *parametric* if it applies the same uniform algorithm at all instantiations of its type parameters [14]. Reynolds [12] proposed *relational parametricity* as a mathematical model of parametric polymorphism. Relational parametricity is a powerful mathematical tool with many useful consequences [7, 15]. The polymorphic lambda-calculus $\lambda\mathbf{2}$ [6, 11] serves as a model type theory for (impredicative) polymorphism. Taken separately, the categorical structures needed to model $\lambda\mathbf{2}$ and relational parametricity are well-known ($\lambda\mathbf{2}$ *fibrations* [8, 13] and *parametricity graphs* [3, 4], respectively). However, simply combining these two notions results in a structure that enjoys the expected properties of parametricity only in the special case that the underlying category is *well-pointed*. Since well-pointedness rules out many categories of interest in semantics (e.g., functor categories) this limits the generality of the theory.

Existing solutions (e.g. Birkedal and Møgelberg [2]) overcome this restriction by adding significant additional structure to models (enough to model the full logic of Plotkin and Abadi [10]). We give instead a minimal solution, where we retain the original idea of combining reflexive graph categories with category-theoretic models of $\lambda\mathbf{2}$. We implement this in a perhaps unexpected way: *we modify the notion of* $\lambda\mathbf{2}$ *model* by asking for $\lambda\mathbf{2}$ fibrations to additionally satisfy Lawvere's *comprehension* property [9]. This way, we can interpret a combined context containing both type and term variables interspersed (similar to how contexts need to be handled in dependent type theories). We then combine such *comprehensive* $\lambda\mathbf{2}$ *fibrations* with parametricity-graph structure in order to also model relational parametricity. We call the resulting structures *comprehensive* $\lambda\mathbf{2}$ *parametricity graphs*.

A type theory for reasoning about parametricity

The main focus of this presentation is a type theory, $\lambda\mathbf{2R}$, which has a sound and complete interpretation in comprehensive $\lambda\mathbf{2}$ parametricity graphs, and which can be used to show that our models enjoy that the expected consequences of parametricity. This type theory is similar in many respects to System R of Abadi, Cardelli and Curien [1] and System P of Dunphy [3]. In addition to the standard judgements of System F, we add three new judgements: $\boxed{\Theta\ \mathsf{rctxt}}$ says that Θ is a well-defined *relational context*; $\boxed{\Theta \vdash A_1 R A_2\ \mathsf{rel}}$ says that R is a *relation* between types A_1 and A_2, in relational context Θ; and $\boxed{\Theta \vdash (t_1 : A_1) R (t_2 : A_2)}$ is a *relatedness judgement*, asserting that $t_1 : A_1$ is related to $t_2 : A_2$ by the relation R.

Importantly, both type and term variables on the left-hand side of relations are always manipulated separately from variables on the right. As a consequence, it does not make sense to talk about equality relations in the type theory, since there is no way to ensure that a judgement $\Theta \vdash (x : A) R (x : A)$ refers to "the same" x on the left and right hand side. For each $\Gamma \vdash A\ \mathsf{type}$, we instead define a *pseudo-identity relation* $\langle\Gamma\rangle \vdash A \langle A \rangle A\ \mathsf{rel}$ — the canonical relational interpretation of A. However, because of the changed context $\langle\Gamma\rangle$ (obtained by taking the relational interpretation of Γ pointwise), $\langle A\rangle$ is not a proper identity relation in general — in fact, for open types B, the relation $\langle B\rangle$ is not even a homogeneous. The more subtle pseudo-identity property of $\langle A\rangle$ is instead given by the *parametricity rule*, which states that if $\langle\Gamma\rangle \vdash (s : A)\langle A\rangle(t : A)$, then $\Gamma \vdash s = t : A$. This rule is sound because of the parametricity-graph structure present in our models.

Graphs of functions are ubiquitous in standard arguments involving relational parametricity. Since we in general lack identity relations, we also lack graph relations, but we identify two forms of *pseudograph* relations, whose subtle interrelationship allows us to establish the consequences we need. One kind of pseudograph relation is immediately definable using the *fibrational* structure built into the notion of parametricity graph. The other type requires *opfibrational* structure. We use an impredicative encoding to show that opfibrational structure is definable in $\lambda\mathbf{2R}$. Finally, we can replay the usual proofs — but with graph relations replaced by pseudographs — showing that comprehensive $\lambda\mathbf{2}$ parametricity graphs enjoy the familiar consequences of parametricity.

Based on published work This is based on our recent paper published in FoSSaCS 2016 [5].

References

[1] Martín Abadi, Luca Cardelli, and Pierre-Louis Curien. Formal parametric polymorphism. *Theoretical Computer Science*, 121(1&2):9–58, 1993.

[2] Lars Birkedal and Rasmus E. Møgelberg. Categorical models for Abadi and Plotkin's logic for parametricity. *Mathematical Structures in Computer Science*, 15:709–772, 2005.

[3] Brian Dunphy. *Parametricity as a notion of uniformity in reflexive graphs*. PhD thesis, University of Illinois, 2002.

[4] Brian Dunphy and Uday Reddy. Parametric limits. In *LICS*, pages 242–251, 2004.

[5] Neil Ghani, Fredrik Nordvall Forsberg, and Alex Simpson. Comprehensive parametric polymorphism: categorical models and type theory. In B. Jacobs and C. Löding, editors, *FoSSaCS 2016*, volume 9634 of *LNCS*. Springer, 2016.

[6] Jean-Yves Girard. *Interprétation fonctionelle et élimination des coupures dans l'arithmétique d'ordre supérieur*. PhD thesis, University of Paris VII, 1972.

[7] Claudio Hermida, Uday Reddy, and Edmund Robinson. Logical relations and parametricity — a Reynolds programme for category theory and programming languages. *ENTCS*, 303:149–180, 2014.

[8] Bart Jacobs. *Categorical Logic and Type Theory*. Elsevier, 1999.

[9] F William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. *Applications of Categorical Algebra*, 17:1–14, 1970.

[10] Gordon Plotkin and Martín Abadi. A logic for parametric polymorphism. In Marc Bezem and Jan Friso Groote, editors, *TLCA*, volume 664 of *LNCS*, pages 361–375. Springer, 1993.

[11] John Reynolds. Towards a theory of type structure. In B. Robinet, editor, *Programming Symposium*, volume 19 of *LNCS*, pages 408–425. Springer, 1974.

[12] John Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing*, pages 513–523, 1983.

[13] Robert A.G Seely. Categorical semantics for higher order polymorphic lambda calculus. *Journal of Symbolic Logic*, pages 969–989, 1987.

[14] Christopher Strachey. Fundamental concepts in programming languages. *Higher Order Symbolic Computation*, 13(1-2):11–49, 2000.

[15] Philip Wadler. Theorems for free! In Joseph E. Stoy, editor, *FPCA*, pages 347–359. ACM, 1989.