A Nominal Syntax for Internal Parametricity

Thorsten Altenkirch and Ambrus Kaposi

School of Computer Science, University of Nottingham, United Kingdom
{txa,auk}@cs.nott.ac.uk

Abstract

We define a type theory with internal parametricity together with an operational semantics. This is a first step towards the goal of defining cubical type theory where univalence is an admissible rule.

1 Introduction

Parametricity [6] is the principle that terms respect logical relations. Univalence [5] can be understood as a principle that terms respect extensional equality.

Two pairs are logically related if they are componentwise related. Two functions are logically related if they map related inputs to related outputs. Two types are logically related if there is any relation between them.

Two pairs are equal if they are componentwise equal. Two functions are equal if they map equal inputs to equal outputs. Two types are equal if there is a relation between them which is a graph of an equivalence.

The above correspondence suggests that if we have a type theory with internal parametricity, by replacing the definition of "relatedness" for the universe and adjusting the rest of theory, we get a theory with univalence. The current formulations of Homotopy Type Theory (eg. [5]) lack a computational understanding of univalence. Internal parametricity was given computational meaning by [3] and [2]. Our work can be seen as a reformulation of [3] in an explicit substitution calculus using dimension names instead of permutations which simplifies the presentation. One difference from [2] is that we don't decorate typing judgements with a list of dimensions.

We first extend the syntax of type theory with the metatheoretic proof that every term is parametric (section 2). However, this is not enough for internal parametricity, since the function which maps a term to its parametricity proof cannot be typed, because the domain and codomain live in different contexts. To solve this problem, we introduce a substitution from a context to the extended context of related elements (section 3). Finally, we discuss how to extend the theory to use equivalence instead of any relation for "relatedness" (section 4).

2 External parametricity

Parametricity says that logically related interpretations of a term are logically related. More precisely, given $\Gamma \vdash t : A$, $\rho, \rho' : \emptyset \Rightarrow \Gamma$ and a witness that ρ and ρ' are related in the logical relation generated by Γ, then $t[\rho]$ and $t[\rho']$ will be related in the logical relation generated by A. We formalize this by the following rule:

$$\frac{\Gamma \vdash t : A}{\Gamma^i \vdash t^i : t[0_{i\Gamma}] \sim_{A^i} t[1_{i\Gamma}]}$$

Γ^i contains two copies of Γ (ρ and ρ' above) and a logical relation between them. $\sim_{A^i}$ is the logical relation generated by A. It relates the two interpretations of the term t, in the two different copies of Γ, which are projected out by the substitutions $0_{i\Gamma}$, $1_{i\Gamma}$. t^i is the witness of this relation.

The context Γ^i contains two copies of Γ and witnesses that they are pointwise related:

$$(\Gamma.x : A)^i \equiv \Gamma^i.x_{i0} : A[0_{i\Gamma}].x_{i1} : A[1_{i\Gamma}].x_{i2} : x_{i0} \sim_{A^i} x_{i1}).$$

The operation $-^i$ on terms is defined by induction on the term structure, eg. for the universe:

$$A \sim_{\mathsf{U}^i} B \equiv A \to B \to \mathsf{U}.$$

3 Internal parametricity

To internally derive that every function of type $\Pi(A : \mathsf{U}).A \to A$ is identity, we would need a term t expressing parametricity for f assuming f:

$$\begin{aligned} f : \Pi(A : \mathsf{U}).A \to A \vdash t : & \Pi(A_{i0}, A_{i1} : \mathsf{U}, A_{i2} : A_{i0} \to A_{i1} \to \mathsf{U})\,. \\ & \Pi(x_{i0} : A_{i0}, x_{i1} : A_{i1}, x_{i2} : A_{i2}\, x_{i0}\, x_{i1})\,.\, A_{i2}\,(f\, A_{i0}\, x_{i0})\,(f\, A_{i1}\, x_{i1}). \end{aligned}$$

We could choose t to be f^i, but that lives in the context $(f : \Pi(A : \mathsf{U}).A \to A)^i$.

This motivates the definition of a substitution that takes us into the $-^i$-d context. We define this substitution mutually with the term former refl_i. The typing rules are as follows:

$$\frac{\Gamma \vdash}{\mathsf{R}_{i\Gamma} : \Gamma \Rightarrow \Gamma^i} \qquad \frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{refl}_i\, t : t \sim_{A^i[\mathsf{R}_{i\Gamma}]} t}$$

Now the above term t can be defined as $f^i[\mathsf{R}_i]$.

We give an operational semantics for the type theory which has the substitution R.

4 Equivalence

If we replace the definition of U^i by equivalence, i.e. $A \sim_{\mathsf{U}^i} B \equiv \Sigma(\sim : A \to B \to \mathsf{U}).\big(\Pi(x : A).\mathsf{isContr}(\Sigma(y : B).x \sim y)\big) \times \big(\Pi(y : B).\mathsf{isContr}(\Sigma(x : A).x \sim y)\big)$, we are forced to add Kan fillers for every type, as in [4]. This is ongoing work [1].

References

[1] T. Altenkirch and A. Kaposi. A syntax for cubical type theory. Unpublished draft, 2014.

[2] J.-P. Bernardy and G. Moulin. Type-theory in color. In *Proc. of 18th ACM SIGPLAN Int. Conf. on Functional Programming, ICFP '13*, pp. 61–72. ACM, 2013.

[3] J.-P. Bernardy and G. Moulin. A computational interpretation of parametricity. In *Proc. of 27th Ann. IEEE/ACM Symp. on Logic in Computer Science, LICS '12*, pp. 135–144. IEEE CS, 2012.

[4] M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. In R. Matthes and A. Schubert, eds., *Proc. of 19th Int. Conf. on Types for Proofs and Programs, TYPES 2013*, v. 26 of *Leibniz Int. Proc. in Inform.*, pp. 107–128. Dagstuhl, 2014.

[5] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* Institute for Advanced Study, 2013. `http://homotopytypetheory.org/book`

[6] J. C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, ed., *Proc. of IFIP 9th World Computer Congress, Information Processing '83*, pp. 513–523. North-Holland, 1983.