Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 319 (2015) 67–82

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Presheaf Model of Parametric Type Theory

Jean-Philippe Bernardy Thierry Coquand Guilhem Moulin

Chalmers University of Technology and University of Gothenburg
{bernardy,coquand,mouling}@chalmers.se

Abstract

We extend Martin-Löf's Logical Framework with special constructions and typing rules providing internalized parametricity. Compared to previous similar proposals, this version comes with a denotational semantics which is a refinement of the standard presheaf semantics of dependent type theory. Further, this presheaf semantics is a refinement of the one used to interpret nominal sets with restrictions. The present calculus is a candidate for the core of a proof assistant with internalized parametricity.

Keywords: Parametricity, Presheaf semantics, Type theory

1 Introduction

Reynolds [17] proved a general *abstraction theorem* (sometimes called *parametricity theorem*) about polymorphic functions. His argument is about a set theoretic semantic. As he stated it, *the underlying idea is that the meanings of an expression in "related" environments will be "related" values*. For instance, he proves that if t_X is a term of type $X \to X$ and if we consider two sets A_0, A_1 and a relation $R \subseteq A_0 \times A_1$, then we have $R([t_X]_{X=A_0}(a_0), [t_X]_{X=A_1}(a_1))$ whenever $R(a_0, a_1)$, where $[t_X]_{X=A}$ denotes the meaning of the expression t_X where X is interpreted by the set A. As he noted, one can replace binary relations by n-ary relations in this statement, and in particular unary relations (predicates). In the latter case, the statement is the following: if A is a set and P is a predicate on A, then we have $P([t_X]_{X=A}(a))$ whenever $P(a)$ holds. Wadler [18] illustrates by many examples how this result is useful for reasoning about functional programs.

The argument and result of Reynolds are model-theoretic in nature. In the Logical Framework, it is possible to state such an abstraction result in a purely syntactical way. One states for example that if a function f has type $(A : U) \to A \to A$ — the type of the polymorphic identity — then $f\,A\,x$ is Leibniz-equal to x, *i.e.*, the following proposition holds:

$$(A : U) \to (P : A \to U) \to (x : A) \to P\,x \to P(f\,A\,x)$$

http://dx.doi.org/10.1016/j.entcs.2015.12.006

Indeed Bernardy et al. [9] prove such a result as a (syntactical) meta-theorem about type systems. However this result is not provable internally, *i.e.*, the following proposition is not provable:

$$(f : (A : U) \to A \to A) \to (A : U) \to (P : A \to U) \to (x : A) \to P\,x \to P(f\,A\,x) \quad (\star)$$

Therefore users relying on the parametricity conditions have postulated the parametricity axiom [3, 11, 16]. However, because postulates do not have computational interpretations, such parametricity conditions can only be used in computationally-irrelevant positions.

Instead, one would like to be able to rely on parametricity conditions within the theory itself. Several attempts have been made [6, 7] — or are currently developed [2] — for designing an extension of dependent type theory in which such an internal form of parametricity holds. We propose another such system here. Our technical contributions are as follows:

- We present an extension of Martin-Löf's Logical Framework (Section 2) which internalizes parametricity (as we show in Section Section 3) and can be seen as a simplification and generalization of the systems of Bernardy and Moulin [6, 7]. In particular, we have a special construction $(a,_i\, p)$ which pairs a term a with its parametricity proof p, as well as special projections to extract the proof. As we will show in Section 3.3, these new constructions enable us to prove the proposition (Equation $\star$) *internally.* (This is not possible with usual pairs and projections since the first projection does not commute with application.) The name i in the above construction is what we call a "color"; we want internalized parametricity not only for LF but also for the extended calculus, and as explained in [7], colors enable nested parametricity by keeping track of the different uses (this is analogous to building hypercubes and accessing their vertices as in [6]). However, unlike previous type theories with internalized parametricity [6, 7], the system presented here does not *compute* parametricity types: for instance, parametricity conditions are *isomorphic to* functions, rather than functions themselves. (As shown in Section 3, this does not appear to be an issue in practice.)
- We provide a *denotational* semantics, in the form of a presheaf model, for this type theory (Section 4). This model is a refinement of the presheaf semantics used to interpret nominal sets with restrictions [10, 15].

We conjecture that conversion and type-checking are decidable for this system.

2 Syntax

In this section we define the syntax and typing rules of our parametric type theory, as well as the equality judgment.

We assume a special symbol '0', and a countably infinite set $\mathbb{I}$ of other symbols, called *colors*. The metasyntactic variables $i, j, \ldots$ range over colors, while φ range over $\mathbb{I} \cup \{0\}$. We further assume a fixed function $\mathsf{fresh}(\cdot)$ such that $\mathsf{fresh}(I) \in \mathbb{I} \backslash I$