Internalized Parametricity via Lifting Universals

Aaron Stump

Boston College, Boston, Massachusetts, `aaron.stump@bc.edu`

Introduction. Internalizing the parametricity principle introduced by Reynolds has several well-known benefits for Type Theory [6], including derivation of induction principles for lambda-encoded data, and "free theorems" derived solely from types [8, 7]. Bernardy and Moulin identified a critical issue taking the (meta-level) relational interpretation $\llbracket - \rrbracket$ of a type containing an internalized use $\llbracket X \rrbracket$ of that interpretation [4]: there is a mismatch between the meta-level interpretation, which is indexed by variable renamings, and the internalized version, which it seems should not be. Bernardy and Moulin's solution, based on a symmetry property of relational interpretations, necessitates a pervasive change to the theory, where abstractions now bind hypercubes of variables. Altenkirch et al. follow this geometric approach, leading to a complex formal system [1]. Our goal is a simpler solution.

This abstract describes a constructive type theory $\llbracket \texttt{CC} \rrbracket$ (pronounced "lift CC") in progress, which extends the Calculus of Constructions (CC) with an internalized parametricity principle. $\llbracket \texttt{CC} \rrbracket$'s introduces a construct $\llbracket A \rrbracket_i^k$ ("lifting"), with $i \leq k$, where k is the arity of the relation. Definitional equality unfolds applications of this operator, as suggested also by Altenkirch et al. [1]. But $\llbracket \texttt{CC} \rrbracket$ avoids the mismatch identified by Bernardy and Moulin, by internalizing the meta-level renamings as part of a type form called a *lifting universal*, with syntax $\Pi\, x\langle \bar{x} \rangle : A\, .\, B$.

Syntax. The syntax of $\llbracket \texttt{CC} \rrbracket$ is shown in Figure 1. It can be seen as an extension of the Calculus of Constructions (CC), and uses the sorts $\star$ and $\Box$ as in [2]. Metavariable θ ranges over the binders. We have generalized forms for Π-types, λ-abstractions, and applications, as well as $\llbracket t \rrbracket_i^k$ for internalized interpretation. For $\llbracket t \rrbracket_i^k$, it is required that $i \leq k$.

Lifting universals. In general, the interpretation of a type T as a k-ary relation is indexed by $k+1$ variable renamings $\rho_0, \cdots, \rho_k$. Bernardy and Lasson realize these renamings by a fixed scheme for deriving the names of new variables $x_0, \cdots, x_{k-1}, \mathring{x}$ from a starting variable x [3]. Here, in contrast, we will consider the renamings explicitly. If $i < k$, then ρ_i maps each free variable x in T to a variable x_i used in speaking about the i'th term that the relation is relating. The renaming ρ_k is used to map $x : R$ to a variable x_k showing that the inputs $\bar{x}$ are related by the interpretation of the type R. For simplicity, we take x_k to be x, so ρ_k is the identity renaming and may be omitted. We write $\bar{x}^k$ for the vector $x_0, \cdots, x_{k-1}$.

$\llbracket \texttt{CC} \rrbracket$ augments the Π-binder from CC to quantify additionally over $\bar{x}^k$, with syntax $\Pi\, x\langle \bar{x} \rangle : R\, .\, S$. This combines the renamings $x \mapsto x_i$, for $i < k$, with accepting an input x that proves the $\bar{x}$ are related by the relational interpretation of R. The formation, introduction, and elimination

naturals	$\mathbb{N}$	$\ni$	i, j, k		
variables	*Var*	$\ni$	x, y, z, X, Y, Z		
binders	*Bnd*	$\ni$	θ	$::=$	$\lambda \mid \Pi$
sorts	*Srt*	$\ni$	s	$::=$	$\star \mid \Box$
terms	*Tm*	$\ni$	A, B, C, t, R, S, T	$::=$	$x \mid s \mid \theta\, x\langle \bar{x} \rangle : R\, .\, S \mid t'\ t\langle \bar{t} \rangle \mid \llbracket t \rrbracket_i^k$
contexts	*Ctx*	$\ni$	Γ	$::=$	$\cdot \mid x\langle \bar{x} \rangle : A, \Gamma$

Figure 1: Syntax for $\llbracket \texttt{CC} \rrbracket$

$$\frac{\Gamma \vdash A : s_1 \quad \Gamma, x\langle \bar{x} \rangle : A \vdash B : s_2}{\Gamma \vdash \Pi\, x\langle \bar{x} \rangle : A \,.\, B : s_2} \qquad \frac{\Gamma \vdash \Pi\, x\langle \bar{x} \rangle : A \,.\, B : s \quad \Gamma, x\langle \bar{x} \rangle : A \vdash t : B}{\Gamma \vdash \lambda\, x\langle \bar{x} \rangle : A \,.\, t : \Pi\, x\langle \bar{x} \rangle : A \,.\, B}$$

$$\frac{\Gamma \vdash t' : \Pi\, x\langle \bar{x}^k \rangle : A \,.\, C \quad \Gamma \vdash t\langle \bar{t} \rangle : A}{\Gamma \vdash t'\ t\langle \bar{t}^k \rangle : [t\langle \bar{t} \rangle / x]C} \qquad \frac{(\forall\ i < k.\ \Gamma \vdash t_i : \llbracket A \rrbracket_i^k) \quad \Gamma \vdash t : \llbracket A \rrbracket_k^k\ \bar{t}}{\Gamma \vdash t\langle \bar{t}^k \rangle : A}$$

Figure 2: Typing rules for lifting universals, along with helper judgement $\Gamma \vdash t\langle \bar{t} \rangle : A$

$$\begin{array}{lcll}
[t\langle \bar{t} \rangle / x]\llbracket x \rrbracket_i^k & = & t_i & i < k \\
[t\langle \bar{t} \rangle / x]\llbracket x \rrbracket_k^k & = & t & \\
[t\langle \bar{t}^k \rangle / x]\llbracket y \rrbracket_i^n & = & \llbracket y \rrbracket_i^n & n \neq k \vee y \neq x \\
[t\langle \bar{t} \rangle / x]x & = & t & \\
[t\langle \bar{t} \rangle / x]y & = & y & x \neq y \\
[t\langle \bar{t} \rangle / x]\star & = & \star & \\
[t\langle \bar{t} \rangle / x]\theta\, y\langle \bar{y} \rangle : R \,.\, S & = & \theta\, y\langle \bar{y} \rangle : [t\langle \bar{t} \rangle / x]R \,.\, [t\langle \bar{t} \rangle / x]S & \\
[t\langle \bar{t} \rangle / x](s'\ s\langle \bar{s} \rangle) & = & [t\langle \bar{t} \rangle / x]s'\ [t\langle \bar{t} \rangle / x]s\langle [t\langle \bar{t} \rangle / x]\bar{s} \rangle &
\end{array}$$

Figure 3: Applying a substitution $[t\langle \bar{t} \rangle / x]$ to a term

rules are shown in Figure 2. When $k = 0$, these rules are isomorphic to the usual ones from CC for Π-types, and we use the usual syntax $\Pi\, x : A \,.\, B$ for $\Pi\, x\langle\, \rangle : A \,.\, B$. Similarly, $t'\ t$ abbreviates $t'\ t\langle\, \rangle$, and $\lambda\, x : A \,.\, B$ abbreviates $\lambda\, x\langle\, \rangle : A \,.\, B$. Figure 3 defines substitution. The critical idea is to respect the lifting operator: when substituting into $\llbracket x \rrbracket_i^k$, we choose the i'th term from the vector $\bar{t}$ (first equation of Figure 3).

Typing liftings. Figure 4 gives the last typing rules for $\llbracket \mathtt{CC} \rrbracket$, which are those for liftings, as well as the axiom $\star : \square$ and the conversion rule. Parametricity is expressed in rule π. The rules *vr* and *vp* show how assumptions of the form $x\langle \bar{x} \rangle : A$ contribute to typing: x is a proof that $\bar{x}$ is in the relational interpretation of A, and each x_i has type $\llbracket A \rrbracket_i^k$, where the lifting is needed to interpret variables y in A introduced with similar assumptions $y\langle \bar{y} \rangle : B$.

Conversion. $\llbracket \mathtt{CC} \rrbracket$ uses definitional equality to simplify uses of $\llbracket - \rrbracket$. The critical idea is to use a contextual definitional equality of the form $\Gamma \vdash A \simeq B$, and to add an assumption $x\langle \bar{x} \rangle : A$ to the context in the case of a lifting universal. This information may then be used to reduce $\llbracket x \rrbracket_i^k$, as expressed in rule L of Figure 5, which says that if we find an assumption $x\langle \bar{x} \rangle : A$ in Γ where the length of $\bar{x}$ is k_j for some j, then we can replace x with x_{i_j}, while retaining the other liftings. (Here, i_j is the requested position from the $\llbracket x \rrbracket_{\bar{i}}^{\bar{k}}$ notation.) But this replacement is only allowed for positional liftings, where $i_j < k_j$. This replacement can be justified as a permutation of a positional lifting with other liftings. But such permutation

$$\frac{\Gamma \vdash A : B \quad i < k}{\Gamma \vdash \llbracket A \rrbracket_i^k : \llbracket B \rrbracket_i^k} \qquad \frac{\Gamma \vdash A : B}{\Gamma \vdash \llbracket A \rrbracket_k^k : \llbracket B \rrbracket_k^k\ \llbracket A \rrbracket_0^k \cdots \llbracket A \rrbracket_{k-1}^k}\ \pi \qquad \frac{}{\Gamma \vdash \star : \square}$$

$$\frac{x\langle \bar{x}^k \rangle : A \in \Gamma}{\Gamma \vdash x : \llbracket A \rrbracket_k^k\ \bar{x}}\ vr \qquad \frac{x\langle \bar{x}^k \rangle : A \in \Gamma \quad i < k}{\Gamma \vdash x_i : \llbracket A \rrbracket_i^k}\ vp \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash A \simeq B}{\Gamma \vdash t : B}$$

Figure 4: Typing rules for liftings, plus additional standard rules

$$\frac{x \notin FV(B)}{\Gamma \vdash \Pi\, x\langle \bar{x}^k \rangle : A \,.\, B \simeq \Pi\, \bar{x} : \llbracket A \rrbracket^k \,.\, \Pi\, x : \llbracket A \rrbracket_k^k\ \bar{x} \,.\, B} \qquad \frac{x\langle \bar{x}^{k_j} \rangle : A \in \Gamma \quad i_j < k_j}{\Gamma \vdash \llbracket x \rrbracket_{\bar{i}}^{\bar{k}} \simeq \llbracket x_{i_j} \rrbracket_{\bar{i} \backslash i_j}^{\bar{k} \backslash k_j}}\ L$$

$$\frac{S \equiv \Pi\, X : A \,.\, B}{\Gamma \vdash \llbracket S \rrbracket_k^k \simeq \lambda\, \bar{Y}^k : S \,.\, \Pi\, X\langle \bar{X}^k \rangle : A \,.\, \llbracket B \rrbracket_k^k\ (\bar{Y}\ \bar{X})} \qquad \frac{}{\Gamma \vdash \llbracket A\ B \rrbracket_k^k \simeq \llbracket A \rrbracket_k^k\ \llbracket B \rrbracket_0^k \cdots \llbracket B \rrbracket_{k-1}^k}$$

$$\frac{\Gamma \vdash A \simeq A' \quad \Gamma, x\langle \bar{x} \rangle : A \vdash B \simeq B'}{\Gamma \vdash \theta\, x\langle \bar{x} \rangle : A \,.\, B \simeq \theta\, x\langle \bar{x} \rangle : A' \,.\, B'} \qquad \frac{i < k}{\Gamma \vdash \llbracket \star \rrbracket_i^k \simeq \star} \qquad \frac{}{\Gamma \vdash \llbracket \star \rrbracket_k^k \simeq \lambda\, \bar{Y}^k : \star \,.\, \bar{Y} \to \star}$$

Figure 5: Selected rules for definitional equality.

does not make sense for relational liftings ($\llbracket A \rrbracket_k^k$), where arity-$k$ and arity-j relational liftings result in different arity relations, and hence could not be permuted. The first rule of Figure 5 makes $\Pi\, x\langle \bar{x} \rangle : A \,.\, B$ an abbreviation for the nested Π-type one would expect from the relational interpretation of $\Pi\, x : A \,.\, B$, as long as x has been completely eliminated from the body. The middle row of Figure 5 expresses the relational semantics of Π-types using lifting universals, and applications using positional liftings ($0 \leq i < k$) and the relational lifting ($i = k$). Positional lifting behaves homomorphically with respect to the constructs of CC (rules omitted).

Example: iterated internal parametricity. Internalized unary parametricity is expressed as $\Pi\, A : \star \,.\, \Pi\, a : \llbracket A \rrbracket_0^1 \,.\, \llbracket A \rrbracket_1^1\ a$ (abbreviate this $\mathcal{T}$). The type given for a is as required by the type of $\llbracket A \rrbracket_1^1$, based on rule π. Let us calculate $\llbracket \mathcal{T} \rrbracket_2^2$.

$$\begin{array}{ll}
\llbracket \mathcal{T} \rrbracket_2^2 & \simeq \\
\lambda\, \bar{X}^2 : \mathcal{T} \,.\, \Pi\, A\langle \bar{A}^3 \rangle : \star \,.\, \Pi\, a\langle \bar{a}^3 \rangle : \llbracket A \rrbracket_0^1 \,.\, \llbracket \llbracket A \rrbracket_1^1\ a \rrbracket_2^2\ (\bar{X}\ \bar{A}\ \bar{a}) & \simeq \\
\lambda\, \bar{X}^2 : \mathcal{T} \,.\, \Pi\, A\langle \bar{A}^3 \rangle : \star \,.\, \Pi\, a\langle \bar{a}^3 \rangle : \llbracket A \rrbracket_0^1 \,.\, \llbracket \llbracket A \rrbracket_1^1 \rrbracket_2^2\ \llbracket a \rrbracket_0^2\ \llbracket a \rrbracket_1^2\ (\bar{X}\ \bar{A}\ \bar{a}) & \simeq \\
\lambda\, \bar{X}^2 : \mathcal{T} \,.\, \Pi\, A\langle \bar{A}^3 \rangle : \star \,.\, \Pi\, a\langle \bar{a}^3 \rangle : \llbracket A \rrbracket_0^1 \,.\, \llbracket \llbracket A \rrbracket_1^1 \rrbracket_2^2\ a_0\ a_1\ (X_0\ A_0\ a_0)\ (X_1\ A_1\ a_1) &
\end{array}$$

Call the last type above Q, and let us see how its body is typable. For readability, write $\mathcal{A}_i$ for $\llbracket A \rrbracket_i^1$. $\mathcal{A}_i$ does not equal A_i, as the length of $\bar{A}$ in the context is 3, which does not match the arity 1 of this lifting. The type of $\llbracket \llbracket A \rrbracket_1^1 \rrbracket_2^2$ is the following, again followed by several definitionally equal types:

$$\begin{array}{ll}
\llbracket \llbracket \star \rrbracket_1^1\ \mathcal{A}_0 \rrbracket_2^2\ \llbracket \mathcal{A}_1 \rrbracket_0^2\ \llbracket \mathcal{A}_1 \rrbracket_1^2 & \simeq \\
\llbracket \mathcal{A}_0 \to \star \rrbracket_2^2\ \llbracket \mathcal{A}_1 \rrbracket_0^2\ \llbracket \mathcal{A}_1 \rrbracket_1^2 & \simeq \\
\Pi\, x\langle \bar{x}^3 \rangle : \mathcal{A}_0 \,.\, \llbracket \mathcal{A}_1 \rrbracket_0^2\ x_0 \to \llbracket \mathcal{A}_1 \rrbracket_1^2\ x_1 \to \star &
\end{array}$$

To type the body of Q, we use rule L to simplify a positional lifting when it is nested in another lifting. For one example, using L, we can prove that the type of $(X_0\ A_0\ a_0)$, which is $\llbracket A_0 \rrbracket_1^1\ a_0$, equals $\llbracket \llbracket A \rrbracket_1^1 \rrbracket_0^2\ a_0$. Since we have $A(\bar{A}^3) : \star$ in the context, the positional lifting $\llbracket \llbracket A \rrbracket_1^1 \rrbracket_0^2$ is equal to $\llbracket A_0 \rrbracket_1^1$, replacing A with A_0. This holds similarly for the types of $\bar{a}$ and $(X_1\ A_1\ a_1)$, allowing the body of Q to be typed.

Towards normalization. Developing a semantics for $\llbracket \mathsf{CC} \rrbracket$ seems challenging, because of liftings. One would need to define a relational semantics that can be iteratively applied, so one could apply the semantics to an object already in the semantic domain. Instead of this path, I propose to use the Girard projection to reduce normalization of $\llbracket \mathsf{CC} \rrbracket$ to normalization of F_ω, as in [2]. It then becomes plausible to consider internalizing Girard projection as a type construct, as in [5], which would allow new definitional equalities as discussed in [8].

Acknowledgments. Thanks to the anonymous reviewers for helpful comments.

References

[1] Thorsten Altenkirch, Yorgo Chamoun, Ambrus Kaposi, and Michael Shulman. Internal parametricity, without an interval. *Proc. ACM Program. Lang.*, 8(POPL):2340–2369, 2024.

[2] H. P. Barendregt. Lambda calculi with types. In *Handbook of Logic in Computer Science*. Oxford University Press, 12 1992.

[3] Jean-Philippe Bernardy and Marc Lasson. Realizability and parametricity in pure type systems. In Martin Hofmann, editor, *Foundations of Software Science and Computational Structures - 14th International Conference, FOSSACS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, volume 6604 of *Lecture Notes in Computer Science*, pages 108–122. Springer, 2011.

[4] Jean-Philippe Bernardy and Guilhem Moulin. A computational interpretation of parametricity. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 135–144. IEEE Computer Society, 2012.

[5] Jean-Philippe Bernardy and Guilhem Moulin. Type-theory in color. In *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming*, ICFP '13, page 61–72, New York, NY, USA, 2013. Association for Computing Machinery.

[6] John C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, France, September 19-23, 1983*, pages 513–523. North-Holland/IFIP, 1983.

[7] Philip Wadler. Theorems for free! In Joseph E. Stoy, editor, *Proceedings of the fourth international conference on Functional programming languages and computer architecture, FPCA 1989, London, UK, September 11-13, 1989*, pages 347–359. ACM, 1989.

[8] Philip Wadler. The girard-reynolds isomorphism (second edition). *Theor. Comput. Sci.*, 375(1-3):201–226, 2007.