Type Refinements for the Working Class

Jon Sterling and Darin Morrison

There are two conflicting views which bedevil any discussion of the nature of type theory. First, there is the notion of type theory as an extension or generalization of universal algebra to support interdependency of sorts and operations, possibly subject to an arbitrary equational theory [1, 3]; we will call this *formal type theory*. Typing, in such a setting, is a mere matter of grammar and is nearly always decidable. In hindsight, we may observe that this is the sort of type theory which Martin-Löf first proposed in 1972 [8], even if we will admit that this was not the intention at the time. A model for such a type theory is usually given by interpreting the types or sorts as presheaves or sheaves over contexts of hypotheses, and as such, a proof theoretic interpretation of the hypothetical judgment is inevitable.

Secondly, there is the view of type theory as semi-formal theory of constructions for the Brouwer-Heyting-Kolmogorov interpretation of intuitionistic mathematical language, which we will call *behavioral* or *semantic type theory*. The most widely known development of this program is Martin-Löf's 1979 "extensional" type theory [9, 10], but we must give priority to Dana Scott for inventing this line of research in 1970 with his prophetic report, *Constructive Validity* [11]. More recently, behavioral type theory has been developed much further in the Nuprl family [2] of proof assistants, including MetaPRL [7] and JonPRL [12].

Martin-Löf's key innovation was the commitment to pervasive functionality (extensionality) as part of the *definitions* of the judgments and the types, in contrast to the state of affairs in formal type theory where functionality is a metatheorem which may or may not hold depending on the particular equational theory which has been imposed. Furthermore, models for behavioral type theory interpret the types as partial equivalence relations on only closed terms, and the meaning of the hypothetical judgment is defined separately and uniformly in the logical relations style.

Our position is that these views of type theory are not in conflict, but rather merely describe two distinct layers in a single, harmonious system. From this perspective, formal type theories can do little more than negotiate matters of grammar, and therefore may serve as a syntactic (linguistic) framework for mathematical language, being responsible for the management of variable binding and substitution. On the other hand, the meaning of mathematical statements shall be specified *behaviorally* in the semantic type theory.

The types of the semantic theory can then be said to *refine* the types of the syntactic theory [5, 6, 4], both by placing restrictions on membership and by coarsening equivalence.

Thus far, all developments of behavioral type theory have been built on a *unityped* syntactic framework, and so the relation to type refinements has been difficult to see. In this paper, we contribute a full theory of behavioral refinements over multi-sorted abstract binding trees, a simple formal type theory [4, 13]; this hybrid system allows the deployment of a Nuprl-style type theory over any signature of sorts and operators.

1 Abstract Binding Trees and Symbols

See [13] for the development of abstract binding trees with symbols.

give a brief description of the framework, and present its rules.

2 The Ambient Logical Framework

In this paper, we hint at the *modes* of judgments and assertions [4] using colors, marking inputs with `blue` and outputs with `red`. As a rule of thumb, inputs are things which are supplied when checking the correctness of a judgment, and outputs are things which are synthesized in the process.

Then, for any abt signature $\Sigma \equiv \langle \mathcal{S}, \mathcal{O} \rangle$, we can deploy a generic logical framework. Recall from [13] that **SCtx** is the category of symbol-contexts and their injective, sort-preserving renamings. Then, fix a small category $\mathcal{W}$ of worlds along with a map $\pi_{\mathcal{W}} : \mathcal{W} \to \mathbf{SCtx}$ such that $\pi_{\mathcal{W}}$ is a Grothendieck opfibration. Then, a *judgment* shall be interpreted as an object in the presheaf category $\mathbb{J} \triangleq \mathbf{Set}^{\mathcal{W}}$; in other words, a judgment is identified with the intensional set of its renaming-sensitive derivations.

Definition 2.1 (General Judgment)**.** For a metavariable context Ω *mctx*, we can define the presheaf of its substitutions $\Box\Omega : \mathbf{Set}^{\mathbf{SCtx}}$ as follows:

$$\Box\Omega\,(\Upsilon) \triangleq \prod_{(m:\nu)\in\Omega} \{E \mid \cdot \triangleright \Upsilon \parallel \cdot \vdash E : \nu\}$$

For any diagram $\mathcal{J} : \mathbb{J}^{\Box\Omega(\Upsilon)}$, we can define the general judgment $|_\Omega\ \mathcal{J}$ as follows, where $\mathbf{y}(-) : \mathcal{W} \to \mathbb{J}$ is the Yoneda Embedding:

$$(|_\Omega\ \mathcal{J})\,(w) \triangleq \mathbb{J}[\mathbf{y}(w), \mathtt{Lan}_{\mathbb{1}}\mathcal{J}]$$

Intuitively, the general judgment $|_\Omega\ \mathcal{J}$ is evident when the family of judgments $\mathcal{J}$ shall forevermore have a section (i.e. for any substitution $\omega \in \Box\Omega(\Upsilon)$, $\mathcal{J}(\omega)$ shall be evident). Recall that metavariables have *valences* rather than sorts; as such, the general judgment allows us to quantify over terms of higher type. However, for clarity we will often write $|_{m:\sigma}\ \mathcal{J}$ rather than $|_{m:\{\cdot\}[\cdot],\,\sigma}\ \mathcal{J}$; likewise, in the body of $\mathcal{J}$ we will simply write m rather than the more proper $m\{\cdot\}(\cdot)$.

Definition 2.2 (Hypothetical Judgment)**.** For two judgments $\mathcal{J}_1$ and $\mathcal{J}_2$, we can form the hypothetical judgment $\mathcal{J}_2 \ (\mathcal{J}_1)$, which expresses the semantic consequence of $\mathcal{J}_2$ from $\mathcal{J}_1$:

$$\begin{aligned}(\mathcal{J}_2 \ (\mathcal{J}_1))\,(w) &\triangleq \left(\mathcal{J}_2{}^{\mathcal{J}_1}\right)(w) \\ &\cong \mathbb{J}[\mathbf{y}(w) \times \mathcal{J}_1, \mathcal{J}_2]\end{aligned}$$

In other words, the hypothetical judgment $\mathcal{J}_2 \ (\mathcal{J}_1)$ is evident when there shall forevermore be an effective transformation of evidence for the antecedent into evidence for the consequent. It is important to keep in mind that this is a *semantic* consequence and therefore expresses admissibilities, in contrast to the *logical* consequence which expresses derivabilities [4]. We will not have any need for the latter.

After this section, we will define a judgment either by rules, or by an informal semantical explanation, in both cases leaving its intensional character implicit. A judgment is said to be *correct* or *valid* just in case it is globally inhabited.

3 Behavioral Refinements

Fixing a signature $\Sigma \equiv \langle \mathcal{S}, \mathcal{O} \rangle$ in the abt framework, we will define the notion of behavioral refinement by propounding several judgments and their semantical explanations.

3.1 Parametric Refinement

We will require two judgments, defined mutually: $\Phi \sqsubset^\star \Upsilon$, which means that Φ is a symbol context which refines the symbol context Υ (presupposing Υ *sctx*), and parametric refinement $\Phi \parallel \phi \sqsubset \tau$ (which shall presuppose $\Phi \sqsubset^\star \Upsilon$ and τ *sort*).

The first judgment we define inductively:

$$\frac{}{\cdot \sqsubset^\star \cdot}\ \sqsubset^\star_{\mathtt{nil}} \qquad \frac{\Phi \sqsubset^\star \Upsilon \quad \Phi \parallel \phi \sqsubset \tau}{\Phi, u : \phi \sqsubset^\star \Upsilon, u : \tau}\ \sqsubset^\star_{\mathtt{snoc}}$$

The second judgment is defined coinductively via the following meaning explanation:

Definition 3.1 (Parametric Refinement)**.** To know $\Phi \parallel \phi \sqsubset \tau$ (presupposing $\Phi \sqsubset^\star \Upsilon$ and τ *sort*) is to know, for any renaming $\rho : \Upsilon \hookrightarrow \Upsilon'$, and for any M, N, what it means for M and N to be identified under ϕ, supposing $\cdot \triangleright \Upsilon' \parallel \cdot \vdash M : \tau$ and $\cdot \triangleright \Upsilon' \parallel \cdot \vdash N : \tau$, requiring that if $M \sim^\rho_\phi N$ then $N \sim^\rho_\phi M$, and if $M \sim^\rho_\phi N$ and $N \sim^\rho_\phi O$, then $M \sim^\rho_\phi O$.

In other words, to know $\Phi \parallel \phi \sqsubset \tau$ (presupposing $\Phi \sqsubset^\star \Upsilon$) is to know, for any renaming of Υ to Υ', which partial equivalence relation on the closed τ-sorted Υ'-terms ϕ denotes. At this point, it should be remarked that this is very similar to the semantical explanation

of typehood given in [9], except that we have generalized it to a multi-sorted setting, and that we have fibred the entire apparatus over collections Υ of symbols.

define maps of refined contexts, then change weakening lemma to monotonicity, which is stronger and more useful.

Theorem 3.2 (Weakening). *If* $\Phi \parallel \phi \sqsubset \tau$, *then* $\Phi, u : \psi \parallel \phi \sqsubset \tau$ *(supposing* $u \notin |\Phi|$ *and* $\Phi \parallel \psi \sqsubset \sigma$*).*

Proof. This follows trivially from the rule $\sqsubset^\star_{\mathsf{snoc}}$ and the canonical renaming $\Upsilon \hookrightarrow \Upsilon, u : \sigma$. □

When we have $\cdot \parallel \phi \sqsubset \tau$, we say that ϕ *globally* refines τ and may simply write $\phi \sqsubset \tau$. By weakening, if $\phi \sqsubset \tau$ then for all Φ, we have $\Phi \parallel \phi \sqsubset \tau$ supposing $\Phi \sqsubset^\star \Upsilon$.

3.1.1 Equality of parametric refinement

Two refinements are equal when they denote the same PER.That is, will define the refinement equality judgment $\Phi \parallel \phi = \psi \sqsubset \tau$ as follows, presupposing $\Phi \sqsubset^\star \Upsilon$, $\Phi \parallel \phi \sqsubset \tau$ and $\Phi \parallel \psi \sqsubset \tau$:

Definition 3.3. To know $\Phi \parallel \phi = \psi \sqsubset \tau$ is to know, for any substitution $\rho : \Upsilon \hookrightarrow \Upsilon'$, and for any M, N such that $\cdot \triangleright \Upsilon' \parallel \cdot \vdash M : \tau$ and $\cdot \triangleright \Upsilon' \parallel \cdot \vdash N : \tau$, $M \sim^\rho_\phi N$ if and only if $M \sim^\rho_\psi N$.

3.2 Parametric Equality

The primary judgment concerning refinements is the parametric equality, $\Phi \parallel M = N \in \phi$, which presupposes $\Phi \sqsubset^\star \Upsilon$, $\Phi \parallel \phi \sqsubset \tau$, $\cdot \triangleright \Upsilon \parallel \cdot \vdash M : \tau$ and $\cdot \triangleright \Upsilon \parallel \cdot \vdash N : \tau$. The meaning of this judgment is that M and N are identified by ϕ at the identity renaming $1 : \Upsilon \hookrightarrow \Upsilon$:

$$\frac{M \sim^1_\phi N}{\Phi \parallel M = N \in \phi}$$

3.3 Functional Refinement and Equality

Next, we define functional refinement $\Phi \parallel \Psi \vDash \phi = \psi \sqsubset \tau$ in terms of parametric refinement, simultaneously with parametric context refinement $\Phi \parallel \Psi \sqsubset^\star \Gamma$ and functional equality $\Phi \parallel \Psi \vDash M = N \in \phi$. We will write $\Phi \parallel \Psi \vDash \phi \sqsubset \tau$ as a shorthand for $\Phi \parallel \Psi \vDash \phi = \phi \sqsubset \tau$.

Properly express the *syntactic* presuppositions of the sequent judgments so that the use of hypothetical and general judgment is well-formed.

Parametric context refinement $\Phi \parallel \Psi \sqsubset^\star \Gamma$ shall be defined as follows, presupposing $\Phi \sqsubset^\star \Upsilon$ and Γ *vctx*:

$$\frac{}{\Phi \parallel \cdot \sqsubset^\star \cdot}\ \parallel\!\sqsubset^\star_{\mathsf{nil}} \qquad \frac{\Phi \parallel \Psi \sqsubset^\star \Gamma \quad \Phi \parallel \Psi \vDash \phi \sqsubset \tau}{\Phi \parallel \Psi, x : \phi \sqsubset^\star \Gamma, x : \tau}\ \parallel\!\sqsubset^\star_{\mathsf{snoc}}$$

Functional refinement $\Phi \parallel \Psi \vDash \phi = \psi \sqsubset \tau$ presupposes $\Phi \parallel \Psi \sqsubset^\star \Gamma$ and τ *sort*, and is explained by induction on the evidence for its first presupposition.

Case $\dfrac{}{\Phi \parallel \cdot \sqsubset^\star \cdot}\ \parallel\!\sqsubset^\star_{\mathsf{nil}}$.

$$\frac{\Phi \parallel \phi = \psi \sqsubset \tau}{\Phi \parallel \cdot \vDash \phi = \psi \sqsubset \tau}$$

Case $\dfrac{\Phi \parallel \Psi \sqsubset^\star \Gamma \quad \Phi \parallel \Psi \vDash \chi \sqsubset \sigma}{\Phi \parallel \Psi, x : \chi \sqsubset^\star \Gamma, x : \sigma}\ \parallel\!\sqsubset^\star_{\mathsf{snoc}}$.

$$\frac{\mid_{y,z:\sigma} \Phi \parallel \Psi \vDash [y \,/\, x]\,\phi = [z \,/\, x]\,\psi \sqsubset \tau \ \ (\Phi \parallel \Psi \vDash y = z \in \chi)}{\Phi \parallel \Psi, x : \chi \vDash \phi = \psi \sqsubset \tau}$$

References

[1] J. Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209 – 243, 1986.

[2] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

[3] P. Dybjer. Internal type theory. In S. Berardi and M. Coppo, editors, *Types for Proofs and Programs*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer Berlin Heidelberg, 1996.

[4] R. Harper. *Practical Foundations for Programming Languages*. Cambridge University Press, New York, NY, USA, 2016.

[5] R. Harper and R. Davies. Refining Objects (Preliminary Summary). `https://www.cs.cmu.edu/~rwh/papers/lc60/lc60.pdf`, 2014.

[6] R. Harper and W. Duff. Type Refinements in an Open World. `https://www.cs.cmu.edu/~rwh/papers/trow/summary.pdf`, 2015.

[7] J. Hickey, A. Nogin, R. L. Constable, B. E. Aydemir, E. Barzilay, Y. Bryukhov, R. Eaton, A. Granicz, A. Kopylov, C. Kreitz, V. N. Krupski, L. Lorigo, S. Schmitt, C. Witty, and X. Yu. MetaPRL – a modular logical environment. In D. Basin and B. Wolff, editors, *Theorem Proving in Higher Order Logics*, volume 2758 of *Lecture Notes in Computer Science*, pages 287–303. Springer Berlin Heidelberg, 2003.

[8] P. Martin-Löf. An intuitionistic theory of types, 1972.

[9] P. Martin-Löf. Constructive mathematics and computer programming. In L. J. Cohen, J. Łoś, H. Pfeiffer, and K.-P. Podewski, editors, *Logic, Methodology and Philosophy of Science VI, Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science, Hannover 1979*, volume 104 of *Studies in Logic and the Foundations of Mathematics*, pages 153–175. North-Holland, 1982.

[10] P. Martin-Löf and G. Sambin. *Intuitionistic type theory*. Studies in proof theory. Bibliopolis, Napoli, 1984.

[11] D. Scott. Constructive validity. In M. Laudet, D. Lacombe, L. Nolin, and M. Schtzenberger, editors, *Symposium on Automatic Demonstration*, volume 125 of *Lecture Notes in Mathematics*, pages 237–275. Springer Berlin Heidelberg, 1970.

[12] J. Sterling, D. Gratzer, and V. Rahli. JonPRL. `http://www.jonprl.org/`, 2015.

[13] J. Sterling and D. Morrison. Syntax and Semantics of Abstract Binding Trees. 2015.