UNFOLDING FOLDS

MATTHEW WEAVER AND DIMITRIS TSEMENTZIS

A well-known problem in Homotopy Type Theory is that of constructing objects that seemingly require infinitely many coherence conditions in their definition. One solution to the problem is to introduce a two-level type theory that contains a strict propositional equality which allows one to replicate the usual set-theoretic definitions using functorial semantics. For example, there is the HTS of Voevodsky [Voe13], logic-enriched type theory in [PL15], and the two level system in [ACK17] and [ACK16].

We propose a different solution to the problem. Our approach is to augment type theory with a postulated interpretation $I\colon \mathbf{Sig} \to \mathbf{Type}$ where $\mathbf{Sig}$ is a type of type expressions (or codes) for a class of types in $\mathbf{Type}$ and where I can be thought of as picking out the correct type corresponding to that type expression. In the specific system we propose, $\mathbf{Sig}$ corresponds to the type expressions originating from signatures in FOLDS [Mak95] (i.e. of finite inverse categories). These type expressions correspond to nested Σ-types whose components are all dependent functions into $\mathbf{Type}$, or equivalently contexts consisting solely of (families of) types, each potentially indexed by the types appearing before it. To illustrate, consider the FOLDS signature given by the finite inverse category depicted by Figure 1 subject to the relation $di = ci$. The encoding of $\mathcal{L}_{\mathrm{rg}}$ corresponds to the data in type theory shown in Figure 2 which themselves can be packaged (given a universe) as the Σ-type

$$\Sigma\,(O\colon \mathbf{Type})\,(A\colon O \times O \to \mathbf{Type}), (\Sigma\,(x : O), A(x,x)) \to \mathbf{Type}$$

Although for any *given* FOLDS signature $\mathcal{L}\colon \mathbf{Sig}$ we can write down its corresponding Σ-type by hand, producing the correct Σ-type for *arbitrary* FOLDS signatures in $\mathbf{Sig}$ encounters precisely the challenge (well documented in [Shu14]) of having to express infinitely many coherence conditions. However, it is easy to determine what the correct Σ-type is externally, and so the key idea in this work is to simply add this external determination to type theory, in the form of a hard-coded interpretation function $I\colon \mathbf{Sig} \to \mathbf{Type}$, together with judgmental equalities defining the action of this function.

Let's take a more detailed look at our approach. A type of well-formed signatures $\mathbf{Sig}$ is defined as an inductive-inductive type (together with a definition of the families of contexts $\mathbf{Con}$, sorts $\mathbf{Sort}$ and substitutions $\mathbf{Sub}$) inspired by the representation of type theory demonstrated in [AK16]. This much can be done internally in a type theory supporting inductive-inductive definitions (or simply hard-coded into a simpler type theory, which is indeed the approach we take). The interpretation is then postulated as a function $I : \mathbf{Sig} \to \mathbf{Type}$ that sends each encoding of a FOLDS signature to its corresponding nested Σ-type. To define I we also need to define auxiliary interpretations

$$i_{\mathbf{Con}}\colon (\mathcal{L}\colon \mathbf{Sig}) \to \mathbf{Con}\,\mathcal{L} \to I\,\mathcal{L} \to \mathbf{Type}$$

$$i_{\mathbf{Sort}}\colon (\mathcal{L}\colon \mathbf{Sig}) \to (\Gamma\colon \mathbf{Con}\,\mathcal{L}) \to \mathbf{Sort}\,\mathcal{L}\,\Gamma \to (x : I\,\mathcal{L}) \to i_{\mathbf{Con}}\,\mathcal{L}\,\Gamma\,x \to \mathbf{Type}$$

$$i_{\mathbf{Sub}}\colon (\mathcal{L} : \mathbf{Sig}) \to (\Gamma, \Delta\colon \mathbf{Con}\,\mathcal{L}) \to \mathbf{Sub}\,\Gamma\,\Delta \to (x : I\,\mathcal{L}) \to i_{\mathbf{Con}}\,\mathcal{L}\,\Gamma\,x \to i_{\mathbf{Con}}\,\mathcal{L}\,\Delta\,x$$

Department of Computer Science, Princeton University
Department of Statistics and Biostatistics, Rutgers University
E-mail addresses: mzw@cs.princeton.edu, dt506@rci.rutgers.edu.

Date: July 23, 2017.

$$\mathcal{L}_{\mathrm{rg}} =_{\mathrm{df}} \begin{tikzcd} I \arrow[d, "i"'] \\ A \arrow[d, bend right, "c"'] \arrow[d, bend left, "d"] \\ O \end{tikzcd}$$

FIGURE 1. an example FIC

$$\begin{aligned} &O \colon \mathbf{Type} \\ &A \colon O \times O \to \mathbf{Type} \\ &I \colon \Sigma\,(x \colon O),\ A(x,x) \to \mathbf{Type} \end{aligned}$$

FIGURE 2. the type information encoded in $\mathcal{L}_{\mathrm{rg}}$

together with judgmental equalities defining the action of these functions. We call the resulting type theory **TT+I**. To minimize the complexity of our initial approach, we define **TT+I** as the Calculus of Constructions with Σ-types along with the inductive-inductive definition of **Sig** and the interpretation I, but throughout this project we are keeping all definitions and proofs as theory-agnostic as possible; in particular we want this result to easily port to a univalent type theory with nontrivial higher equalities. In summary, our contributions are the following:

- An inductive-inductive-recursive definition (formalized in Agda) of the well-formed FOLDS signatures **Sig**, together with an external ZF-proof (in [TW]) that the definition does indeed capture exactly the finite inverse categories.
- An internalization of the above definition into a new type theory **TT+I** that also contains an interpretation function $I \colon \mathbf{Sig} \to \mathbf{Type}$.

As stated, our ultimate motivation in developing **TT+I** is to produce a mechanism through which objects involving infinite chains of coherence conditions can be defined (cf. [ACK15, Shu14]). The way this would work is that certain constructions would pick out the desired encoding from within **Sig** and these constructions would be reified into **Type** by composing with the interpretation I. For example, with a mutually inductive definition we anticipate being able to define $sst \colon \mathbb{N} \to \mathbf{Sig}$ where $sst(n)$ is the *encoding* of the signature for the n-truncated semi-simplicial types. By then composing with I we can define the type of semi-simplicial types SST as a limit (in the sense of [ACS15]) in **Type**. The reason why our approach works where the usual approaches fail is that we are picking out *codes* for the n-truncated semi-simplicial types rather than types themselves. Thus, the equalities that do not hold in the attempted definitions (e.g. in [Her15]) in our case do hold. Since our type of signatures **Sig** is an h-set and thus only has trivial proofs of equality, by using our approach we avoid requiring anything about the structure of equality in **Type**. In summary, the philosophy of **TT+I** is to side-step the coherence issues by having a way to carry out operations on set-level encodings of types rather than on types themselves.

Although we have defined **TT+I** on paper, its implementation is very much a work in progress. We also intend to prove a number of properties about the theory and operational semantics including type preservation and strong normalization, as well as consider developing a more traditional categorical semantics. Upon completing the work relevant to this current iteration of the theory we plan to extend **Sig** along the lines suggested in [Tse16] beyond plain FOLDS to include identity types, transport functions etc. thus increasing the expressive power and applicability of the system.

References

[ACK15] Thorsten Altenkirch, Paolo Capriotti, and Nicolai Kraus, *Infinite structures in type theory: Problems and approaches*, TYPES, Tallinn, Estonia **20** (2015).

[ACK16] ———, *Extending homotopy type theory with strict equality*, 25th EACSL Annual Conference on Computer Science Logic **21** (2016), 1–17.

[ACK17] Danil Annenkov, Paolo Capriotti, and Nicolai Kraus, *Two-level type theory and applications*, arXiv preprint arXiv:1705.03307 (2017).

[ACS15] Benedikt Ahrens, Paolo Capriotti, and Régis Spadotti, *Non-wellfounded trees in homotopy type theory*, arXiv preprint arXiv:1504.02949 (2015).

[AK16] Thorsten Altenkirch and Ambrus Kaposi, *Type theory in type theory using quotient inductive types*, ACM SIGPLAN Notices **51** (2016), no. 1, 18–29.

[Her15] Hugo Herbelin, *A dependently-typed construction of semi-simplicial types*, Mathematical Structures in Computer Science **25** (2015), no. 5, 1116–1131.

[Mak95] M. Makkai, *First order logic with dependent sorts with applications to category theory*, 1995. `http://www.math.mcgill.ca/makkai/folds/foldsinpdf/FOLDS.pdf`.

[PL15] Fedor Part and Zhaohui Luo, *Semi-simplicial types in logic-enriched homotopy type theory*, arXiv preprint arXiv:1506.04998 (2015).

[Shu14] M. Shulman, *Homotopy type theory should eat itself (but so far, it's too big to swallow).: Homotopy type theory blog post*, 2014.

[Tse16] Dimitris Tsementzis, *Homotopy model theory i: Syntax and semantics*, arXiv preprint arXiv:1603.03092 (2016).

[TW] Dimitris Tsementzis and Matthew Weaver, *Finite Inverse Categories as Dependently Typed Signatures.* In Preparation.

[Voe13] Vladimir Voevodsky, *A simple type system with two identity types*, Unpublished note (2013).