Infinite Structures in Type Theory: Problems and Approaches

Thorsten Altenkirch, Paolo Capriotti and Nicolai Kraus

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

When exploring the possibilities of Homotopy Type Theory (HoTT), one quickly realises that many constructions come with coherence problems. This is already apparent when we try to define the concept of *monoid* in full generality: we begin with a type A of elements and a composition function $\circ$. Then we add the requirement that $\circ$ is associative, which can be expressed using the identity type. However, in many situations, associativity might not be enough: we need a form of coherence for the equalities produced by it (the "pentagon"). But these coherence proofs may be required to satisfy their own coherence condition, and this process continues indefinitely, leading to a tower of "coherence properties" which we currently cannot capture with a single definition within type theory. A possible solution is to restrict ourselves to types of some given (small) truncation level, and in the described example, it is justifiable that A should be a *set* (in the sense of HoTT). However, there are a number of interesting cases where either this is not possible or makes no sense. We can think of the following:

1. Defining semi-simplicial types (see e.g. [2]). More generally, representing Reedy fibrant diagrams [7] internally.
2. Given a powerful enough mutual induction principle for higher inductive types, it is possible to define the syntax of HoTT in itself. However, any attempts to define an interpretation function that maps syntactic types to outer types is stymied by the appearance of coherence problems, when mapping definitional equalities of syntactical elements to propositional equalities. A discussion can be found in [6].
3. The current formulation of category theory in HoTT assumes 0-truncated hom-sets [1]. This has the problem of excluding the "category" of types and functions, and many other interesting categorical structures, for which we would need to internalise a notion of $(\infty, 1)$-category.
4. If we want to construct functions $\|A\| \to B$, previous work [4] shows that we have (depending on the truncation level of B) to provide a tower of coherence conditions. In the current theory, we can only express finite parts, and we would like to be able to internalise and formalise the complete result.

All these questions are related and we expect that none of them is solvable in the theory that is considered in the standard reference [8]. Let us elaborate on the first, which we can express as follows: given a type expression E, is it possible to define a family of types $M : \mathbb{N} \to \mathsf{Type}$ such

that E can be given the type $(n : \mathbb{N}) \to M_n \to \mathsf{Type}$ and M satisfies[1]

$$\begin{aligned}
M_0 &\cong \mathsf{1} \\
M_1 &\cong E_0 \times E_0 \\
M_2 &\cong (x_0, x_1, x_2 : E_0) \times E_1(x_0, x_1) \times E_1(x_0, x_2) \times E_1(x_1, x_2) \\
M_3 &\cong (x_0, x_1, x_2, x_3 : E_0) \times (x_{01} : E_1(x_0, x_1)) \times (x_{02} : E_1(x_0, x_2)) \times (x_{03} : E_1(x_0, x_3)) \\
&\quad \times (x_{12} : E_1(x_1, x_2)) \times (x_{13} : E_1(x_1, x_3)) \times (x_{23} : E_1(x_2, x_3)) \\
&\quad \times E_2(x_0, x_1, x_2, x_{01}, x_{02}, x_{12}) \times E_2(x_0, x_1, x_3, x_{01}, x_{03}, x_{13}) \\
&\quad \times E_2(x_0, x_2, x_3, x_{02}, x_{03}, x_{23}) \times E_2(x_1, x_2, x_3, x_{12}, x_{13}, x_{23})
\end{aligned} \tag{1}$$

and so on? At first sight, it looks as if M should be relatively straightforward to define by induction on its index. However, a lot of time and effort has already been spent on this (see, for example, Herbelin's exposition [2] and Shulman's discussion [6]), and it seems that none of the proposed constructions work. It is not easy to pinpoint what goes wrong.

We could try the following: first, we define the category Δ_+ in type theory. It has nonempty finite sets as objects and strictly increasing functions as morphisms.Then, we can attempt to define a double-indexed family $M_n^{(k)}$ by induction on k, such that the sequence $M_n^{(k)}$ stabilises for $k \geq n$, and such that $M_n^{(k)}$ is a Δ_+-functor in n. Our intention is then to take $M_n :\equiv M_n^{(n)}$. We set $M_n^{(0)} :\equiv \mathsf{1}$. Then, inductively, we define

$$M_n^{(k+1)} :\equiv (x : M_n^{(k)}) \times ((\sigma : \Delta_+(k, n)) \to E_k(\sigma^\star x)), \tag{2}$$

where $\sigma^\star$ is the functorial action of $M^{(k)}$ (defined at the same time as $M_n^{(k)}$). Unfortunately, this does not type-check. If we (manually or by a script) start to write down the sequence $M_0, M_1, M_2, \ldots$, together with appropriate E_i, following the formula (2), then this does type-check.[2] However, as soon as the indices are variables of type $\mathbb{N}$ (instead of fixed numerals), the construction does not type-check because the functorial action of $M^{(k)}$ is not strict. This suggests that a judgmental η-law for natural numbers, or some form of equality reflection, would provide a solution. A similar observation, we believe, motivated Voevodsky to start the development of HTS.

The problem of constructing "Reedy-fibrant diagrams", or "infinite Σ-types", has been discussed in the community quite frequently (see, for example, Shulman's blog post and its discussion [6]), and several people have argued that it would be desirable to formulate a version of the HoTT theory in which this is possible. Motivations that are similar to ours have led to the development of Voevodsky's HTS [9], a 2-level system, and also to Hickey's *very dependent types* [3] that have been suggested in the context of NuPRL (and for which it is unclear whether they can be made sense of in HoTT).

Attempting to provide an extension of the "standard theory" to perform such constructions, we suggest a system with two different equalities. This is the same approach that is taken in the development of HTS. However, we hope to be able to avoid resorting to the reflection rule for strict equality, which makes typechecking undecidable. Currently, we believe that this is possible in such a way that we can use Agda to provide a nice implementation of our system.

[1] We write $(a : A) \times B(a)$ for $\Sigma(a : A).B(a)$ and (later) $(a : A) \to B(a)$ for $\Pi(a : A).B(a)$.

[2] An interesting observation is that, if the type theory has η for both Π- and Σ-types (as in Agda, but not in Coq), Δ_+ can be implemented in such a way that all categorical laws hold strictly [5]. This is a requirement for the claim that $M_1, M_2, \ldots$ type-check. It looks as if this strictness could be useful for the construction in general, but so far, we are not sure whether it is indeed relevant.

References

[1] B. Ahrens, K. Kapulkin, and M. Shulman. Univalent categories and the Rezk completion. *Math. Struct. in Comput. Sci.*, 25(5):1010–1039, 2015.

[2] H. Herbelin. A dependently-typed construction of semi-simplicial types. *Math. Struct. in Comput. Sci.*, 25(5):1116–1131, 2015. 2015.

[3] J. J. Hickey. Formal objects in type theory using very dependent types. In *Informal Proc. of Foundations of Object Oriented Languages 3*, 1996.

[4] N. Kraus. The general universal property of the propositional truncation. In *Proc. of TYPES '14*, *Leibniz Int. Proc. in Inform.*, Dagstuhl, to appear. arXiv:1411.2682

[5] N. Kraus. *Truncation Levels in Homotopy Type Theory.* PhD thesis, University of Nottingham, 2015.

[6] M. Shulman. Homotopy type theory should eat itself (but so far, it's too big to swallow). Blog post at homotopytypetheory.org, 3 March 2014.

[7] M. Shulman. Univalence for inverse diagrams and homotopy canonicity. *Math. Struct. in Comput. Sci.*, 25(5):1203–1277, 2015.

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

[9] V. Voevodsky. A simple type system with two identity types. Unpublished note, 2013.