Towards Higher Observational Type Theory

Thorsten Altenkirch[1], Ambrus Kaposi[2]*, and Michael Shulmann[3]†

[1] School of Computer Science, University of Nottingham, UK
psztxa@nottingham.ac.uk
[2] Eötvös Loránd University, Budapest, Hungary
{akaposi}@inf.elte.hu
[3] University of San Diego, USA
shulman@sandiego.edu

Abstract

Do we need to go cubical when presenting Homotopy Type Theory? The seminal paper by Cohen, Coquand, Huber and Mörtberg [4] presented important progress in giving not only a constructive semantics of Homotopy Type Theory (which was first achieved by Bezem, Coquand, and Huber [3]) but also in using this model to present a computationally well behaved Type Theory which lead to implementations like Cubical Agda [7] and RedTT. (Although the notions of "cubes" used in all these cases are slightly different). The basic idea here is to introduce a special type, called the interval, and model equality as paths, i.e. as functions from the interval to the given type. A composition operator which corresponds to Kan filling operations can then be defined by recursion over types and most importantly we can give a constructive interpretation of univalence exploiting the fact that the interval is tiny, i.e. exponentiation with it has a right adjoint. The latter property isn't easily transferable to other models, e.g. simplicial sets, leading to a mismatch with standard constructions in homotopy theory.

In the present work we attempt to find a different way to formulate a Type Theory with univalence, which we call Higher Observational Type Theory. The basic idea is to avoid the introduction of an interval but instead to provide rules how to calculate equality types for every type former. To achieve this we first introduce a calculus of logical relations, which forces us to consider context extensions called telescopes. We present this telescopic calculus using higher order abstract syntax, which means that all constructions should be viewed as taking place in the presheaf category over contexts and substitutions, i.e. all rules are relative to a context, all type formers are functorial, and all term formers are natural transformations.

1 Equality

We write telescopes as $\Delta\ \mathbf{Tel}$; these are extensions of a given context Γ. A telescopic substitution $\delta : \Delta$ is basically a substitution Γ to $\Gamma\Delta$ which is inverse to the projection. We introduce the usual judgements relative to a telescope $\Delta \vdash \mathcal{J}$ and they are closed under telescopic substitution: that is from the above we can derive $\mathcal{J}[\delta]$.

*Ambrus Kaposi was supported by the "Application Domain Specific Highly Reliable IT Solutions" project which has been implemented with support from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme, and by Bolyai Scholarship BO/00659/19/3.

†Michael Shulman was supported by the United States Air Force Office of Scientific Research under award number FA9550-21-1-0009.

We introduce a homogeneous equality on telescopes and a heterogenous equality on types:

$$\frac{\Delta\ \mathbf{Tel} \qquad \delta_0, \delta_1 : \Delta}{\mathrm{Id}_\Delta\, \delta_0\, \delta_1\ \mathbf{Tel}} \qquad \frac{\Delta \vdash A : \mathsf{U} \qquad \delta_2 : \mathrm{Id}_\Delta\, \delta_0\, \delta_1 \qquad a_0 : A[\delta_0] \qquad a_1 : A[\delta_1]}{\mathrm{Id}^{\delta_2}_{\Delta.A}\ a_0\ a_1 : \mathsf{U}}$$

Note that all these rules are in a presheaf setting, i.e. relative to a given context Γ. The telescope equality is composed of the type equality: $\mathrm{Id}_{\Delta\,A}\,(\delta_0\,a_0)\,(\delta_1\,a_1) \equiv (\delta_2 : \mathrm{Id}_\Delta\,\delta_0\,\delta_1)\,(\mathrm{Id}^{\delta_2}_{\Delta.A}\ a_0\ a_1)$. Every telescopic substitution and every term preserve equality:

$$\frac{\Delta \vdash A : \mathsf{U} \qquad \delta_0, \delta_1 : \Delta \qquad \delta_2 : \mathrm{Id}_\Delta\, \delta_0\, \delta_1 \qquad \Delta \vdash a : A}{\mathrm{ap}^{\delta_2}_{\Delta.a} : \mathrm{Id}^{\delta_2}_{\Delta.A}\, a[\delta_0]\, a[\delta_1]}$$

Now homogeneous equality and reflexivity arise as special cases:

$$\frac{A : \mathsf{U} \qquad a_0, a_1 : A}{\begin{array}{c}\mathrm{Id}_A\, a_0\, a_1 : \mathsf{U} \\ \mathrm{Id}_A\, a_0\, a_1 \equiv \mathrm{Id}^{()}_{\bullet.A}\, a_0\, a_1\end{array}} \qquad \frac{A : \mathsf{U} \qquad a : A}{\begin{array}{c}\mathrm{refl}_A\, a : \mathrm{Id}_A\, a\, a : \mathsf{U} \\ \mathrm{refl}_A\, a \equiv \mathrm{ap}^{()}_{\bullet.a}\end{array}}$$

2 Univalence

We define equality in the universe to be a relational characterisation of equivalence such as that in [6, Exercise 4.2]. A proof relevant relation (or "correspondence") $R : A_0 \to A_1 \to \mathsf{U}$ is an equivalence (or "one-to-one correspondence"), written $\mathrm{Eq}\,R$, if it is a function in both directions. It is a function from left to right if $\Sigma a_1 : A_1.R\,a_0\,a_1$ is contractible for all $a_0 : A_0$.

$$\frac{\delta_0, \delta_1 : \Delta \qquad \delta_2 : \mathrm{Id}_\Delta\, \delta_0\, \delta_1 \qquad A_0, A_1 : \mathsf{U}}{\mathrm{Id}^{\delta_2}_{\Delta.\mathsf{U}}\, A_0\, A_1 \equiv \Sigma(R : A_0 \to A_1 \to \mathsf{U})\, \mathrm{Eq}\, R}$$

Now equality is just given by the relation we obtain from type equality. That means we can actually compute equality using ap:

$$\frac{\Delta \vdash A : \mathsf{U} \qquad \delta_0, \delta_1 : \Delta \qquad \delta_2 : \mathrm{Id}_\Delta\, \delta_0\, \delta_1 \qquad a_0 : A[\delta_0] \qquad a_1 : A[\delta_1]}{\mathrm{Id}^{\delta_2}_{\Delta.A}\ a_0\ a_1 \equiv \pi_1\, \mathrm{ap}^{\delta_2}_{\Delta.A}\, a_0\, a_1}$$

That means that the only thing we need to define is ap. Given this definition of equality in the universe we can define weak path induction (that is without the definitional β-equality).

In particular we need to show that the universe is closed under the standard type formers, i.e. that they preserve equivalences. The lifting of the relation to type formers is the standard lifting of logical relations.

3 Progress so far

This is work in progress and hence incomplete and subject to change. It is to some degree inspired by earlier work by the first two authors [1]. A more concise definition of the calculus can be given in terms of Categories with Families (CwF) in a presheaf category. We note that the telescope calculus gives rise to a CwF in the presheaf category and that the congruence rules we give arise from the notion of logical relations as given by [2]. While our definition of the universe seems sound, it is not clear whether this approach gives rise to a straightforward decision procedure. Another line of research is to show that some cubical set model can be presented in a way to satisfy the definitional equalities of our calculus; and more generally that it can be interpreted in model categories for all higher toposes, as was shown for non-computational Homotopy Type Theory in [5].

References

[1] Thorsten Altenkirch and Ambrus Kaposi. Towards a cubical type theory without an interval. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18-21, 2015, Tallinn, Estonia*, volume 69 of *LIPIcs*, pages 3:1–3:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[2] 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.

[3] Marc Bezem, Thierry Coquand, and Simon Huber. A model of type theory in cubical sets. In *19th International Conference on Types for Proofs and Programs*, volume 26 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 107–128. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2014.

[4] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. *FLAP*, 4(10):3127–3170, 2017.

[5] Michael Shulman. All $(\infty, 1)$-toposes have strict univalent universes. arXiv:1904.07004, 2019.

[6] Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. http://homotopytypetheory.org/book/, first edition, 2013.

[7] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. *J. Funct. Program.*, 31:e8, 2021.