# Towards Simplicial Complexes in Homotopy Type Theory

Maximilian Doré and Samson Abramsky

University of Oxford, United Kingdom
{maximilian.dore,samson.abramsky}@cs.ox.ac.uk

One natural application of homotopy type theory (HoTT) is the verification of computational topology. For example, HoTT seems very appropriate to formulate algorithms in Discrete Morse theory (DMT) [5]: DMT is an adaption of Morse theory for simplicial complexes and provides a rich apparatus to study topological spaces without introducing any notion of continuity. Of central interest in DMT is the reduction of bigger complexes to simpler ones under preservation of homotopy type, which should be slickly presentable in HoTT since homotopy equivalence is its central notion of equality.

Unfortunately, there is no satisfactory account of simplicial complexes in HoTT at hand, and also more general simplicial structures elude a definition. The importance of defining simplicial structures has been appreciated since the birth of HoTT since the first model of HoTT was given in simplicial sets, and an internal definition of simplicial sets would allow for reasoning about HoTT in HoTT itself. An important step in this direction is considered the construction of *semi-simplicial types*, which are, roughly speaking, a definition of $\Delta$-sets [6] in type theory. It has not been possible to carry out this definition in classical HoTT of [13]. A variant of type theory that allows for defining semi-simplicial types are *two-level type theories* [14, 2] (2LTT), in which an external, extensional type theory manages the coherences in the inner, intensional type theory. However, 2LTT are intricate to implement and work with. For the verification of computational topology we need a practical type theory with good computational properties.

Below, we first present a naïve approach to simplicial structures in HoTT: We develop an elementary construction of simplicial complexes in dependent type theory in which simplicial complexes are simply lists of vectors of vertices. For lower-dimensional simplicial complexes, we construct from the list description a CW complex with the adequate homotopy type.

This approach might appear unwieldy, but it underlines that the failure of defining simplicial structures in HoTT is not due to a lack of expressivity. Indeed, when looking at the various attempts to define semi-simplicial types, the opposite seems to be the case: The problem is that there is *too much* structure present in intensional type theory (ITT). The iterated identity type exhibits non-trivial behaviour, even if the structure under consideration is only supposed to be truth-apt or a simple collection of objects. Our second approach presents a solution to this problem by introducing a variant of ITT which we call *finite-dimensional type theory* (FTT). In FTT, every type comes with a fixed dimension, and all iterated identity types above this dimension are collapsed by a new eliminator called Axiom $L$, a generalization of Streicher's Axiom $K$ [12]. To our knowledge, FTT is the first attempt to introduce extensionality into ITT by incorporating dimensions on the level of judgments. We have implemented an experimental version of FTT in Agda and work is in progress to formalize semi-simplicial types.

If FTT keeps up to our hopes, we want to unite it with the recent advances in cubical type theory, thereby yielding a univalent type theory with a productive handling of extensionality. In this type theory, we want to formalize recent results of computational mathematics to advance the dissemination of formal reasoning in the mathematical mainstream. We believe that researchers in computational mathematics would be most inclined to adopt constructive formal reasoning if this has been shown to be viable — after all, the promise of intuitionistic type theory was to unify proving and programming!

## Abstract simplicial complexes and their homotopy types

We present a slim definition of ordered abstract simplicial complexes in dependent type theory: A simplicial complex of dimension $n$ is a dependent type that contains for each $i \leq n$ a list of faces of dimension $i$. A face of dimension $i$ is a vector of $i$ natural numbers, which stand for the vertices of the complex. The closure property of simplicial complexes, which requires that all non-empty subsets of a face are also in the complex, is formulated as a dependent function type. This representation is independent of the types-as-spaces interpretation and could be carried out in any dependent type theory.

In order to reason about the homotopy type of simplicial complexes, we utilize a construction of CW complexes introduced by [3], in which cells are attached iteratively by homotopy pushouts. We port their formalization to Cubical Agda, which greatly reduces the required code due to the direct access to higher inductive types (HIT). We prove some basic results of CW complexes such as the equivalence to certain HIT representations of CW complexes.

We then construct a function that takes any simplicial complex of dimension smaller than 3 to a CW complex with a corresponding homotopy type: The points of the simplicial complex remain discrete 0-cells, the lines are mapped to 1-cells, and the triangles are mapped to squares with two vertices identified. The combinatorial description ascribes which vertices of these standard simplices need to be identified. This allows us to reason about the homotopy type of simplicial complexes with the built-in path arithmetic of Cubical Agda.

The code of this section can be found online: `https://github.com/maxdore/cubical`

## Semi-simplicial types in finite-dimensional type theory

In the construction described above, we have to enforce the closure property of simplicial complexes explicitly. This makes further operations on the complexes cumbersome to implement, which would be avoided if the closure property is ensured intrinsically. Famously, it appears to be impossible to do this in HoTT. More precisely, the construction of Reedy fibrant $n$-truncated semi-simplicial types could not be carried out inside HoTT despite many efforts [7, 1, 10]. The main obstacle is that internally defined categories in HoTT do not preserve certain functor laws judgmentally, which is symptomatic of a bigger problem: Oftentimes, the equality proofs of a type beyond a certain dimension are of no interest in further constructions and only complicate formalizations.

We propose the following variation of ITT to solve this "problem of coherences": In *finite-dimensional type theory*, the identity type of any type collapses at a specific level. The basic judgment of ITT, "$A$ is a type", is refined to "$A$ is an $n$-type", denoted $A\ \mathsf{Type}^n$. We call $n$ the *dimension* of $A$. Intuitively, a 0-type has at most one inhabitant and represents a proposition, a 1-type is a set with different inhabitants, but only unique identity proofs, a 2-type has a 1-groupoidal structure, and so on. The rules of ITT are refined by taking into account the dimension. For example, the dependent function type takes a dimension as parameter, e.g., $\Pi^1(x : A)(B(x))$ can be thought of as the "hom-set" of dependent morphisms between $A$ and $B$, were $A\ \mathsf{Type}^i$ and $B\ \mathsf{Type}^j$ for arbitrary dimensions $i, j$.

The introduction rule of the identity type is changed crucially to this:

$$\frac{\Gamma \vdash A :\ \mathsf{Type}^n \quad \Gamma \vdash a : A \quad \Gamma \vdash b : A}{\Gamma \vdash \mathsf{Id}_A(a, b) :\ \mathsf{Type}^{\mathsf{pred}(n)}}\ \textsc{Id-Intro}$$

The predecessor function $\mathsf{pred}$ sends $\mathsf{suc}(n)$ to $n$ and 0 to 0. Hence, Id-Intro establishes that for $A\ \mathsf{Type}^n$, all identity types on $A$ from the $n$-th iteration on have dimension 0. For example, for $A$ $\mathsf{Type}^2$ we have $\mathsf{Id}_A(a,b)$ $\mathsf{Type}^1$ and $\mathsf{Id}_{\mathsf{Id}_A(a,b)}(p,q)$ $\mathsf{Type}^0$ for $a,b : A$ and $p,q : \mathsf{Id}_A(a,b)$. The elimination and computation rule for identity types remain standard.

The collapse of 0-types is achieved by adding a new eliminator, which we call *Axiom L*. Axiom $L$ can be thought of as a generalization of Streicher's Axiom $K$ to all 0-types (which might or might not be identity types). Axiom $L$ and its computation principle are the following:

$$\frac{\Gamma \vdash A\ \mathsf{Type}^0 \quad \Gamma, x : A \vdash C\ \mathsf{Type}^n \quad \Gamma \vdash a : A \quad \Gamma \vdash c : C[a/x] \quad \Gamma \vdash b : A}{\Gamma \vdash L(A, C, a, c, b) : C[b/x]}\ L$$

$$\frac{\Gamma \vdash A\ \mathsf{Type}^0 \quad \Gamma, x : A \vdash C\ \mathsf{Type}^n \quad \Gamma \vdash a : A \quad \Gamma \vdash c : C[a/x]}{\Gamma \vdash L(A, C, a, c, a) \equiv c : C[a/x]}\ L\text{-}\textsc{Comp}$$

Intuitively, $L$ states that if we want to construct an inhabitant of a type $C$ depending on a 0-type $A$, we only need to prove that $C$ is inhabited for some $a : A$, and then we get an inhabitant of $C$ for any $b : A$. The computation rule ensures that if we give the witness to $L$ for which we have already constructed a proof, we retain the original proof.

We can make the correspondence between Axiom $L$ and Axiom $K$ precise as follows: Uniqueness of identity proofs, which is equivalent to Axiom $K$, can be derived from $L$ for all 1-types. Furthemore, the elimination and computation rule for the unit type $\top$ follow from the statement that $\top$ $\mathsf{Type}^0$ and Axiom $L$, and do not need to be added to the system.

To sum up, FTT refines ITT as follows: Every type intrinsically has a dimension, and all identity types above this dimension are trivial. This gives us a way to impose proof-irrelevant coherences to our constructions in type theory. For example, we can construct semi-simplicial types as in the formalization accompanying [1], but without any postulates: By stating that semi-simplicial types and their skeletons are 1-types, the skeleton functor is evidently coherent.

We have implemented an experimental version of FTT: https://github.com/maxdore/FTT

## Outlook

We have presented the first account of finite-dimensional type theory, which is a manageable variant of ITT that allows for defining semi-simplicial types. Some fundamental results for FTT are imperative: We have to show consistency and canonicity for the theory. The easiness with which we could implement a first version of it and the similarity of Axiom $L$ to Streicher's Axiom $K$ make us hopeful that FTT has good computational properties.

Extensionality in ITT is of course a well-researched topic, for which the standard approach is a *truncation* operation that erases computational content. Truncation has been extensively researched, e.g., in [8], which will likely shed light onto FTT. Of particular interest is if intricacies of truncation such as "Kraus' magic trick" [9, Section 6.4] are bypassed with FTT.

Next, it will be exciting to relate FTT to cubical type theories [4], in which univalence holds constructively. In [11] it is shown that univalence restricted to $(n-1)$-types is consistent with a type theory in which all types are $n$-truncated, which might lead the way to incorporate partial forms of univalence in FTT. Introducing higher inductive types in FTT will also be interesting: It is an open problem to define the $n$-sphere as a general HIT parametrized by $n$, having dimensions on a judgmental level might make this feasible.

We expect that an extension of FTT that allows for types without coherences is consistent: By adding $\infty$ as a possible dimension, Axiom $L$ is disabled for any $A$ $\mathsf{Type}^\infty$, thereby allowing for weak $\omega$-groupoidal types in the spirit of classical HoTT. We expect that FTT with an $\infty$-dimension is, if well-behaved, conservative over HoTT.

# References


[1] Thorsten Altenkirch, Paolo Capriotti, and Nicolai Kraus. Extending Homotopy Type Theory with Strict Equality. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *LIPIcs*, pages 21:1–21:17. Schloss Dagstuhl, 2016.

[2] Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications. *arXiv*, 2017. arXiv:1705.03307.

[3] Ulrik Buchholtz and Kuen-Bang Hou (Favonia). Cellular cohomology in homotopy type theory. In *Proc. of the 33rd ACM/IEEE Symposium on Logic in Computer Science*, pages 521–529, 2018.

[4] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, volume 69 of *LIPIc*, pages 5:1–5:34. Schloss Dagstuhl, 2018.

[5] Robin Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90–145, 1998.

[6] Allen Hatcher. *Algebraic topology.* 2005.

[7] Hugo Herbelin. A dependently-typed construction of semi-simplicial types. *Mathematical Structures in Computer Science*, 25(5):1116–1131, 2015.

[8] Nicolai Kraus. The General Universal Property of the Propositional Truncation. In Hugo Herbelin, Pierre Letouzey, and Matthieu Sozeau, editors, *20th International Conference on Types for Proofs and Programs (TYPES 2014)*, volume 39 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 111–145. Schloss Dagstuhl, 2015.

[9] Nicolai Kraus. *Truncation levels in homotopy type theory.* PhD thesis, University of Nottingham, 2015.

[10] Nicolai Kraus. On the role of semisimplicial types. *TYPES 2018*, 2018.

[11] Christian Sattler and Andrea Vezzosi. Partial univalence in n-truncated type theory. *arXiv*, 2020. arXiv:2005.00260.

[12] Thomas Streicher. Investigations into intensional type theory. *Habilitiation Thesis, Ludwig Maximilian Universität*, 1993.

[13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

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