Eliminating Higher Truncations via Constancy

Paolo Capriotti and Nicolai Kraus

University of Nottingham

Abstract

We show how to construct functions $\|A\|_n \to B$ if B is not an n-type.

In Homotopy Type Theory (HoTT), *truncations* constitute an important class of *higher inductive types*: for any type A and any integer $n \geq -1$, the n-truncation $\|A\|_n$ can be understood as a version of the type A where all homotopical structure above level n is collapsed. In general, a type without any nontrivial structure above level n is called an n-type.

When $n = -1$, truncations correspond to the *squashing* or *bracketing* [1] operator, of which they can be thought of as higher-dimensional generalisations. Truncations are usually presented as reflectors of the corresponding sub-$(\infty, 1)$-categories of n-types, resulting in an elimination principle which only allows n-types as targets. Given a function $f : A \to B$, we can construct a function $\|A\|_n \to B$ as long as B is an n-type.

If B happens to be an m-type for some $m > n$, then the eliminator cannot be applied directly. Therefore, in order to factor a function $f : A \to B$ through the truncation $\|A\|_n$, the usual approach is to construct an ad-hoc n-type P, and show, with the help of the eliminator, that f factors through P. However, it is not always clear how to construct such a type P.

We address this problem in vast generality by reducing the problem of factoring a function $f : A \to B$ through $\|A\|_n$ to that of proving a number of coherence conditions on f.

The simplest nontrivial special case of our construction is, for a given 0-truncated type B (a *set*), the equivalence

$$\left(\|A\|_{-1} \to B\right) \;\simeq\; \left(\Sigma_{f:A\to B}\, \forall a_1\, a_2.\, f(a_1) = f(a_2)\right). \tag{1}$$

This equivalence tells us that, in order to define $\|A\|_{-1} \to B$, we need to find $f : A \to B$ and a proof that f takes equal values for any pair of points in its domain. The latter can be understood as a weak form of constancy. Let us write C_1 for this condition:

$$C_1^{A;f} \;:\equiv\; \Pi_{a_1,a_2:A}\; f(a_1) = f(a_2).$$

Unfortunately, the equivalence (1) breaks down when B is anything other than a 0-type (a related explanation can be found in [3]). For example, if B is a 1-type, then, given a function $f : A \to B$, a term $c_1 : C_1^{A;f}$ is not sufficient to guarantee that f factors through $\|A\|_{-1}$. We need to impose an additional condition: c_1 should provide "coherent" equality proofs in B. More precisely, we require an inhabitant of the type

$$C_2^{A;f,c_1} \;:\equiv\; \Pi_{a_1,a_2,a_3:A}\; c_1(a_1,a_2) \cdot c_1(a_2,a_3) = c_1(a_1,a_3).$$

Indeed, we can then prove the following equivalence: for any type A and any 1-type B,

$$\left(\|A\|_{-1} \to B\right) \;\simeq\; \left(\Sigma_{f:A\to B}\, \Sigma_{c_1:C_1^{A;f}}\, C_2^{A;f,c_1}\right). \tag{2}$$

We can deal with higher truncations similarly. For example, factoring through the 0-truncations requires the same conditions, but this time they are imposed on ap_f rather than f directly. When B is a 1-type, we then obtain the equivalence:

$$\left(\|A\|_0 \to B\right) \;\simeq\; \left(\Sigma_{f:A\to B}\, \Pi_{a:A} C_1^{\Omega(A,a);\mathsf{ap}_f}\right). \tag{3}$$

It is not hard to imagine that the equivalences (1), (2) and (3) can be generalised both to any truncation level of B and to any truncation operator, by formulating appropriate coherence conditions C_n for all n, of which C_1 and C_2 are the first two examples.

For any fixed numbers $k, n \geq -1$ (technically, -2) and n-type B, we show how to construct a type in basic HoTT (using Σ, Π, Id only) which is equivalent to $\|A\|_k \to B$. Our proof of the equivalence, however, requires higher inductive types for $k \geq 0$.

We view these equivalences as *generalised universal properties* of the truncations. In the case $n = k$, they degenerate to the ordinary universal property of the truncation, as defined in [4, Lemma 7.3.3], $(\|A\|_n \to B) \;\simeq\; (A \to B)$.

Note that it is not immediately obvious how to even express the general result. It is not difficult to guess the conditions C_n for the first few values of n, but, although the pattern is intuitively clear, it is quite hard to capture it precisely.

Fortunately, Shulman's work on inverse diagrams [5] provides a powerful framework, which helps formulate and reason about towers of coherence conditions.

The idea to generalise the equivalences (1) and (2) to higher truncation levels of B (but still assuming $k = -1$ for now) is that the required coherence conditions may be regarded as a morphism between two *semi-simplicial types* [2]: the 0-coskeletal semi-simplicial type "generated" by A, and the *equality semi-simplicial type on B* (an explicit Reedy-fibrant resolution of B regarded as a constant presheaf). The statement of the general case ($k \geq -1$) can then be obtained by applying the above construction to ap_f^{k+1}.

We state and prove our result in any model of HoTT with ω^{op}-Reedy limits, without putting any restriction on B. However, if B is an n-type for some n, as is often the case for the target of an eliminator, the condition of existence of Reedy limits in the model can be dropped (intuitively, the infinite tower of conditions becomes finite).

In that case, we obtain a simplified formulation, which, for any fixed pair of natural numbers k and n, holds in *any* model of HoTT. In particular, this is true for the initial (syntactical) model, and the equivalences can be used when formalising mathematics in a proof assistant.

Note that our result can simply be regarded as a family of types and equivalences depending on the two indices k and n. The construction is uniform enough to be done mechanically, that is, one could write a program which takes k, n as inputs and generates the required types and proof terms. However, we believe that it is impossible to perform this construction internally for *variables* k, n. This is closely related to the difficulties encountered by several people when trying, for example, to formalise HoTT in itself; Shulman has recently given a thorough analysis of this phenomenon [6].

References

[1] Steven Awodey and Andrej Bauer. Propositions as [types]. *Journal of Logic and Computation*, 14(4):447–471, 2004.

[2] Hugo Herbelin. A dependently-typed construction of semi-simplicial types. 2014.

[3] Nicolai Kraus, Martín Escardó, Thierry Coquand, and Thorsten Altenkirch. Notions of anonymous existence in Martin-Löf type theory. 2014. In preparation.

[4] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. first edition, 2013. Available online at homotopytypetheory.org/book.

[5] Michael Shulman. Univalence for inverse diagrams and homotopy canonicity. *ArXiv e-prints*, March 2012.

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