Eliminating Higher Truncations via Constancy

Paolo Capriotti
joint work with Nicolai Kraus

Functional Programming Lab
University of Nottingham

TYPES, 13th May 2014

Truncation levels

Types in HoTT are organised in a hierarchy

- level -1 (*propositions*): types with trivial equality

$$\mathrm{isprop}(A) :\equiv \Pi(x, y : A).\ x =_A y$$

- level 0 (*sets*): types whose equality is a proposition

$$\mathrm{isset}(A) :\equiv \Pi(x, y : A).\ \mathrm{isprop}(x =_A y)$$

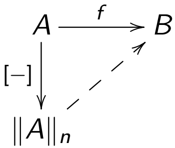
- ...
- level n (*n-types*): types whose equality is in level $n-1$

$$\mathrm{islevel}_n(X) :\equiv \Pi(x, y : A).\ \mathrm{islevel}_{n-1}(x =_A y)$$

Truncations in HoTT

Given any type A, the n-truncation of A gives us:

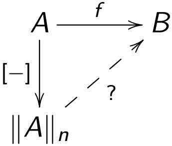
- an n-type $\|A\|_n$
- a "projection" function $[-] : A \to \|A\|_n$
- a *universal property/eliminator*: given any n-type B and a function $f : A \to B$, we can factor f through $[-]$



and the factorisation is unique up to homotopy

Eliminating to higher types

What if B is not an n-type?

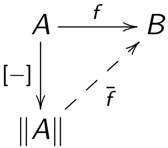


The eliminator doesn't help us.

Eliminating to a set

Let's focus on the -1 truncation (denoted $\| - \|$).

Consider this diagram again:



If $\bar{f}$ does exist, then we have, for any $x, y : A$:

$$f(x) = \bar{f}[x] = \bar{f}[y] = f(y)$$

so f is "constant".

Constancy

We define:

$$\mathsf{const}_0(f) :\equiv \Pi(x, y : A).\ f(x) =_B f(y)$$

So $\mathsf{const}_0(f)$ is a *necessary* condition for $f : A \to B$ to factor through $\|A\|$.

If B is a set, $\mathsf{const}_0(f)$ is also *sufficient*.

Factoring 0-constant functions

Define a 0-truncated higher inductive type P given by:

- $h : A \to P$
- $\Pi(x, y : A).\ h(x) =_P h(y)$
- $\text{isset}(P)$

Clearly f factors through h by construction, so we only need to show that P is a proposition.

For that, we assume $x_0 : A$, and show that

$$\Pi(p : P).\ [x_0] =_P p$$

using the eliminator of P.

This proves $A \to \text{contr}(P)$, from which it easily follows that P is a proposition.

Notions of higher constancy

What if B is not a set?

Given:

- A 1-type B
- $f : A \to B$
- A 0-constancy proof: $c_1 : \mathrm{const}_0(f)$

in order for f to factor through $\|A\|$ we need the following extra condition:

$$c_2 : \Pi(x_1, x_2, x_3 : A).\ c_1(x_1, x_2) \cdot c_1(x_2, x_3) = c_1(x_1, x_3)$$

In general, if B is an n-type, we need a tower of conditions $c_1, \ldots, c_{n+1}$ involving higher paths.

Can we express this tower uniformly in n?

Constancy conditions as maps of simplices

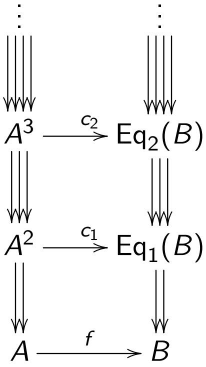
- A function $f : A \to B$ maps points of A to *points* of B
- A term like c_1 maps pairs of points of A to *paths* in B
- A term like c_2 maps triples of points of A to *triangles* in B
- ...

In general, the n-th condition c_n should give a mapping from A^{n+1} to a type $\mathrm{Eq}_n(B)$ of n-simplices in B, compatible with the previous conditions.

Constancy as a semi-simplicial map

We can give $\mathrm{Eq}(B)$ the structure of a (Reedy fibrant) *semi-simplicial* type.

The tower of conditions $c_1, \ldots, c_n$ then becomes a map of semi-simplicial types:



Conclusion

- we give an elimination property of truncations that can be applied to all types
- the formulation and the proof of this result are carried out *externally*
- for fixed values of n and m, the eliminator of the m-truncation into n-types can be expressed internally and used e.g. in formalised proofs