From Hedberg's Theorem to Truncation Elimination and Two-Level Type Theory

Types and Topology
Martín Escardó's 60th birthday

Nicolai Kraus

University of Nottingham

18 Dec 2025

Thank you, Martín!

How I met Martín:

- 2011: I started a PhD with Thorsten in Nottingham
  Topic: "Higher Dimensional Type Theory"
  Difficulty (for me): Not much work *in* HoTT

Thank you, Martín!

How I met Martín:

- 2011: I started a PhD with Thorsten in Nottingham
  Topic: "Higher Dimensional Type Theory"
  Difficulty (for me): Not much work *in* HoTT
- 2012: Martín wrote to me about my blog post *A direct proof of Hedberg's theorem*
  (hard to overstate how encouraging that was!)

Thank you, Martín!

How I met Martín:

- 2011: I started a PhD with Thorsten in Nottingham
  Topic: "Higher Dimensional Type Theory"
  Difficulty (for me): Not much work *in* HoTT
- 2012: Martín wrote to me about my blog post *A direct proof of Hedberg's theorem*
  (hard to overstate how encouraging that was!)
- Then: Martín shared his research questions with me, we had several visits, wrote two papers together, Martín wrote many (!) reference letters for me
  – thank you (again), Martín!

Thank you, Martín!

How I met Martín:

- 2011: I started a PhD with Thorsten in Nottingham
  Topic: "Higher Dimensional Type Theory"
  Difficulty (for me): Not much work *in* HoTT
- 2012: Martín wrote to me about my blog post *A direct proof of Hedberg's theorem*
  (hard to overstate how encouraging that was!)
- Then: Martín shared his research questions with me, we had several visits, wrote two papers together, Martín wrote many (!) reference letters for me
  – thank you (again), Martín!
- Disclaimer: All/most research in this talk is over 10 years old.

One of Martín's questions (from 2012)

Given $f : A \to B$ that is constant, i.e. $\mathsf{wconst}_f :\equiv \prod_{a_1, a_2 : A} f(a_1) = f(a_2)$, does it factor through $\|A\|$?

$$\begin{tikzcd}
A \arrow[rr, "f"] \arrow[d] & & B \\
\|A\| \arrow[urr, dotted] & &
\end{tikzcd}$$

One of Martín's questions (from 2012)

Given $f : A \to B$ that is constant, i.e. $\mathsf{wconst}_f :\equiv \prod_{a_1, a_2 : A} f(a_1) = f(a_2)$, does it factor through $\|A\|$?

$$\begin{tikzcd}
A \arrow[rr, "f"] \arrow[d] & & B \\
\|A\| \arrow[urr, dotted] & &
\end{tikzcd}$$

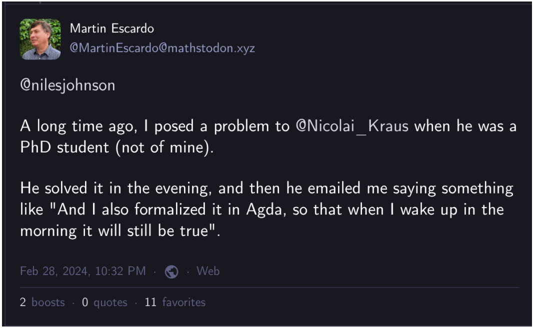
Not in general, but it works for endofunctions ($f : A \to A$) because $\mathsf{fix}\, f :\equiv \Sigma(a : A). f\ a = a$ is a proposition.

One of Martín's questions (from 2012)

Given $f : A \to B$ that is constant, i.e. $\mathsf{wconst}_f :\equiv \prod_{a_1,a_2:A} f(a_1) = f(a_2)$, does it factor through $\|A\|$?

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[d] & & B \\ \|A\| \arrow[urr, dotted] & & \end{tikzcd}$$

Not in general, but it works for endofunctions ($f : A \to A$) because $\mathsf{fix}\, f :\equiv \Sigma(a : A).f\ a = a$ is a proposition.



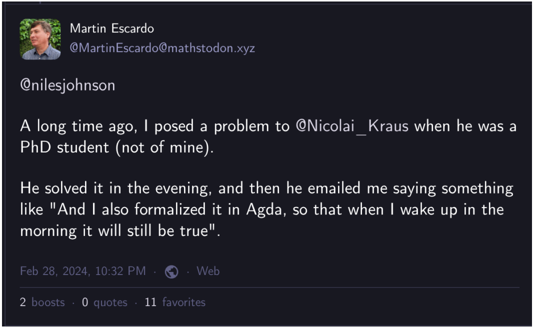
(I had an ugly path algebra proof.)

One of Martín's questions (from 2012)

Given $f : A \to B$ that is constant, i.e. $\mathsf{wconst}_f :\equiv \Pi_{a_1,a_2:A}\ f(a_1) = f(a_2)$, does it factor through $\|A\|$?

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[d] & & B \\ \|A\| \arrow[urr, dotted] & & \end{tikzcd}$$

Not in general, but it works for endofunctions ($f : A \to A$) because $\mathsf{fix}\, f :\equiv \Sigma(a : A). f\ a = a$ is a proposition.


Martin Escardo
@MartinEscardo@mathstodon.xyz

@nilesjohnson

A long time ago, I posed a problem to @Nicolai_Kraus when he was a PhD student (not of mine).

He solved it in the evening, and then he emailed me saying something like "And I also formalized it in Agda, so that when I wake up in the morning it will still be true".

Feb 28, 2024, 10:32 PM · Web

2 boosts · 0 quotes · 11 favorites

(I had an ugly path algebra proof.)

Christian Sattler, some years later: Assume $(a_0, p_0) : \mathsf{fix}\, f$. Then $\mathsf{fix}\, f \simeq \Sigma(a : A). f\ a_0 = a$, which is contractible. $\square$

Another case that works (Martín, 2012)

Theorem. If B is a set, then the canonical map

$$(\|A\| \to B) \quad \longrightarrow \quad \Sigma\,(f : A \to B)\,.\,\mathsf{wconst}_f$$

is an equivalence.

This is simple and (now) contained in most/all libraries.

Another case that works (Martín, 2012)

Theorem. If B is a set, then the canonical map

$$(\|A\| \to B) \quad \longrightarrow \quad \Sigma\,(f : A \to B)\,.\,\mathsf{wconst}_f$$

is an equivalence.

This is simple and (now) contained in most/all libraries.
Unnecessarily complicated argument (not the original):

Lemma 1. Assume that B is a set and $a : A$. Then,
$B \quad \longrightarrow \quad \Sigma\,(f : A \to B)\,.\,\mathsf{wconst}_f$
is an equivalence. $\square$

Lemma 2. If X implies that $Y \xrightarrow{e} Z$ is an equivalence, then $(X \to Y) \xrightarrow{e \circ -} (X \to Z)$ is an equivalence. $\square$

Proof of the theorem: In Lemma 1, weaken the assumption to $\|A\|$. Apply Lemma 2 with $X :\equiv \|A\|$ and observe that

$$(\|A\| \to \Sigma\,(f : A \to B)\,.\,\mathsf{wconst}_f) \simeq (\Sigma\,(f : A \to B)\,.\,\mathsf{wconst}_f).$$

$\square$

Generalizations (2014)

Theorem. If B is a set, then the canonical map

$$(\|A\| \to B) \quad \longrightarrow \quad \Sigma\,(f : A \to B)\,.\,\mathsf{wconst}_f$$

is an equivalence.

Goal: Weaken the assumption on B.

Theorem. If B is a 1-type, then $\|A\| \to B$ is equivalent to the following data:

$$f : A \to B$$

$$c : \mathsf{wconst}_f :\equiv \textstyle\prod_{a_1, a_2 : A} \; f(a_1) = f(a_2)$$

$$d : \mathsf{coh}_{f,c} :\equiv \textstyle\prod_{a_1 a_2 a_3 : A} \; c(a_1, a_2) \cdot c(a_2, a_3) = c(a_1, a_3).$$

Proof. Show that $\mathfrak{a}_0 : A$ implies that

$$B \to \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$$

is an equivalence, then same as above.

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

B

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$\underbrace{\Sigma\,(f_1 : B)\,.}_{\mathbf{1}}$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$\Sigma\,(f_1 : B)\,.$$
$$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \Pi_{a:A}\ f(a) = f_1)\,.$$
$$\mathbf{1}$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$
\begin{aligned}
&\Sigma\,(f_1 : B)\,.\\
&\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \textstyle\prod_{a:A}\; f(a) = f_1)\,.\\
&\Sigma\,(c : \mathsf{wconst}_f)\,.\,\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2 : A}\; c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))\,.\\
&\quad \mathbf{1}
\end{aligned}
$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$\Sigma\,(f_1 : B)\,.$$

$$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \textstyle\prod_{a:A}\; f(a) = f_1)\,.$$

$$\Sigma\,(c : \mathsf{wconst}_f)\,.\,\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2 : A}\; c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))\,.$$

$$\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.\,\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0) \cdot c_1(\mathfrak{a}_0) = c_2)\,.$$

$$\mathbf{1}$$

Proof of $\quad A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$
\begin{aligned}
&\Sigma\,(f_1 : B)\,.\\
&\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \textstyle\prod_{a:A}\ f(a) = f_1)\,.\\
&\Sigma\,(c : \mathsf{wconst}_f)\,.\,\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2 : A}\ c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))\,.\\
&\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.\,\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0) \cdot c_1(\mathfrak{a}_0) = c_2)\,.\\
&\color{red}{\Sigma\,(d : \mathsf{coh}_{f,c})\,.}\\
&\qquad \mathbf{1}
\end{aligned}
$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$
\begin{aligned}
&\Sigma\,(f_1 : B)\,.\\
&\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \textstyle\prod_{a:A}\ f(a) = f_1)\,.\\
&\Sigma\,(c : \mathsf{wconst}_f)\,.\,\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2 : A}\ c(a_1, a_2)\cdot c_1(a_2) = c_1(a_1))\,.\\
&\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.\,\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0)\cdot c_1(\mathfrak{a}_0) = c_2)\,.\\
&\Sigma\,(d : \mathsf{coh}_{f,c})\,.\\
&\Sigma\,(d_2 : \textstyle\prod_{a:A}\ c(\mathfrak{a}_0, a)\cdot c_1(a) = c_2)\,.\\
&\quad \mathbf{1}
\end{aligned}
$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$\Sigma\,(f_1 : B)\,.$$
$$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \textstyle\prod_{a:A}\ f(a) = f_1)\,.$$
$$\Sigma\,(c : \mathsf{wconst}_f)\,.\,\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2:A}\ c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))\,.$$
$$\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.\,\cancel{\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0) \cdot c_1(\mathfrak{a}_0) = c_2)\,.}$$
$$\Sigma\,(d : \mathsf{coh}_{f,c})\,.$$
$$\Sigma\,(d_2 : \textstyle\prod_{a:A}\ c(\mathfrak{a}_0, a) \cdot c_1(a) = c_2)\,.$$
$$\mathbf{1}$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$
\begin{aligned}
&\Sigma\,(f_1 : B)\,.\\
&\Sigma\,(f : A \to B)\,.\,\Sigma\,(c_1 : \textstyle\prod_{a:A}\ f(a) = f_1)\,.\\
&\Sigma\,(c : \mathsf{wconst}_f)\,.\,\cancel{\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2 : A}\ c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))\,.}\\
&\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.\,\cancel{\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0) \cdot c_1(\mathfrak{a}_0) = c_2)\,.}\\
&\Sigma\,(d : \mathsf{coh}_{f,c})\,.\\
&\Sigma\,(d_2 : \textstyle\prod_{a:A}\ c(\mathfrak{a}_0, a) \cdot c_1(a) = c_2)\,.\\
&\quad \mathbf{1}
\end{aligned}
$$

Proof of $\quad A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$
\begin{aligned}
&\Sigma\,(f_1 : B)\,. \\
&\Sigma\,(f : A \to B)\,.\,\cancel{\Sigma\,(c_1 : \textstyle\prod_{a:A}\ f(a) = f_1)}\,. \\
&\Sigma\,(c : \mathsf{wconst}_f)\,.\,\cancel{\Sigma\,(d_1 : \textstyle\prod_{a_1 a_2:A}\ c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))}\,. \\
&\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.\,\cancel{\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0) \cdot c_1(\mathfrak{a}_0) = c_2)}\,. \\
&\Sigma\,(d : \mathsf{coh}_{f,c})\,. \\
&\cancel{\Sigma\,(d_2 : \textstyle\prod_{a:A}\ c(\mathfrak{a}_0, a) \cdot c_1(a) = c_2)}\,. \\
&\quad \mathbf{1}
\end{aligned}
$$

Proof of $A \longrightarrow B \simeq \Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$

Assume $\mathfrak{a}_0 : A$. We transform B into
$\Sigma\,(f : A \to B)\,.\,\Sigma\,(c : \mathsf{wconst}_f)\,.\,\mathsf{coh}_{f,c}$
by adding and removing contractible components:

$$
\begin{aligned}
&\cancel{\Sigma\,(f_1 : B)\,.}\\
&\Sigma\,(f : A \to B)\,.\,\cancel{\Sigma\,(c_1 : \Pi_{a:A}\ f(a) = f_1)\,.}\\
&\Sigma\,(c : \mathsf{wconst}_f)\,.\,\cancel{\Sigma\,(d_1 : \Pi_{a_1 a_2 : A}\ c(a_1, a_2) \cdot c_1(a_2) = c_1(a_1))\,.}\\
&\cancel{\Sigma\,(c_2 : f(\mathfrak{a}_0) = f_1)\,.}\,\cancel{\Sigma\,(d_3 : c(\mathfrak{a}_0, \mathfrak{a}_0) \cdot c_1(\mathfrak{a}_0) = c_2)\,.}\\
&\Sigma\,(d : \mathsf{coh}_{f,c})\,.\\
&\cancel{\Sigma\,(d_2 : \Pi_{a:A}\ c(\mathfrak{a}_0, a) \cdot c_1(a) = c_2)\,.}\\
&\quad \mathbf{1}
\end{aligned}
$$

General case

- This "expanding and contracting" strategy can be done at any level, with minimal assumptions on the type theory – it only needs $\mathbf{1}, \Sigma, \Pi, \mathsf{Id}$ with function extensionality, $\|-\|$.

General case

- This "expanding and contracting" strategy can be done at any level, with minimal assumptions on the type theory – it only needs $\mathbf{1}, \Sigma, \Pi, \mathsf{Id}$ with function extensionality, $\|-\|$.

- Main result: In a type theory with Shulman's Reedy ω^{op}-limits (a.k.a. infinite Σ-types), the type $\|A\| \to B$ is equivalent to the type of *coherently constant* functions $A \to B$. (cf. Lurie, HTT 6.2.3.4)

General case

- This "expanding and contracting" strategy can be done at any level, with minimal assumptions on the type theory – it only needs $\mathbf{1}, \Sigma, \Pi, \mathsf{Id}$ with function extensionality, $\|{-}\|$.

- Main result: In a type theory with Shulman's Reedy ω^{op}-limits (a.k.a. infinite Σ-types), the type $\|A\| \to B$ is equivalent to the type of *coherently constant* functions $A \to B$. (cf. Lurie, HTT 6.2.3.4)

- Main result for HoTT: Fix $n \in \{0, 1, 2, \ldots\}$. If B is an n-type, then $\|A\| \to B$ is equivalent to constant functions $A \to B$ with n levels of coherence.

Coherent constancy

A coherently constant function $A \to B$ is a map between the semisimplicial diagrams $\mathsf{cosk}_0(\mathsf{const}\,A)$ and $\mathsf{const}\,B$.

$$
\begin{tikzcd}
\cdots & \cdots \\
A \times A \times A \arrow[r, dashed, "\mathsf{coh}_{f,c}"] \arrow[d, shift left=2] \arrow[d] \arrow[d, shift right=2] & \begin{array}{c} \Sigma\,(b_1, b_2, b_3 : B)\,. \\ \Sigma\,(p_{12} : b_1 = b_2)\,. \\ \Sigma\,(p_{23} : b_2 = b_3)\,. \\ \Sigma\,(p_{13} : b_1 = b_3)\,. \\ p_{12} \cdot p_{23} = p_{13} \end{array} \arrow[d, shift left=2] \arrow[d] \arrow[d, shift right=2] \\
A \times A \arrow[r, dashed, "c\,:\,\mathsf{wconst}_f"] \arrow[d, shift left] \arrow[d, shift right] & \Sigma\,(b_1, b_2 : B)\,.\, b_1 = b_2 \arrow[d, shift left] \arrow[d, shift right] \\
A \arrow[r, dashed, "f"] & B
\end{tikzcd}
$$

Two-Level Type Theory

Main result for HoTT: Fix $n \in \{0, 1, 2, \ldots\}$. If B is an n-type, then $\|A\| \to B$ is equivalent to constant functions $A \to B$ with n levels of coherence.

- For every *externally fixed n*, this is a theorem in HoTT. Internalising seems to require us to define semisimplicial types — one of the big open problems of HoTT.

Two-Level Type Theory

Main result for HoTT: Fix $n \in \{0, 1, 2, \ldots\}$. If B is an n-type, then $\|A\| \to B$ is equivalent to constant functions $A \to B$ with n levels of coherence.

• For every *externally fixed n*, this is a theorem in HoTT. Internalising seems to require us to define semisimplicial types — one of the big open problems of HoTT.

• In other words: We can write a program $\mathbb{N} \to$ `Agda`, but not a direct Agda formalisation. To remedy this, we could use Voevodsky's HTS – but that's not HoTT!

Two-Level Type Theory

> Main result for HoTT: Fix $n \in \{0, 1, 2, \ldots\}$. If B is an n-type, then $\|A\| \to B$ is equivalent to constant functions $A \to B$ with n levels of coherence.

- For every *externally fixed n*, this is a theorem in HoTT. Internalising seems to require us to define semisimplicial types — one of the big open problems of HoTT.

- In other words: We can write a program $\mathbb{N} \to$ `Agda`, but not a direct Agda formalisation. To remedy this, we could use Voevodsky's HTS – but that's not HoTT!

- What convinced me: Capriotti's argument(*) that 2LTT (two-level type theory) is conservative over HoTT (2016), which means we can extract HoTT proofs.

  (*) Improved by Kovács (2022), Bocquet (2025)

Two-Level Type Theory

Main result for HoTT: Fix $n \in \{0, 1, 2, \ldots\}$. If B is an n-type, then $\|A\| \to B$ is equivalent to constant functions $A \to B$ with n levels of coherence.

• For every *externally fixed n*, this is a theorem in HoTT. Internalising seems to require us to define semisimplicial types — one of the big open problems of HoTT.

• In other words: We can write a program $\mathbb{N} \to \texttt{Agda}$, but not a direct Agda formalisation. To remedy this, we could use Voevodsky's HTS – but that's not HoTT!

• What convinced me: Capriotti's argument(*) that 2LTT (two-level type theory) is conservative over HoTT (2016), which means we can extract HoTT proofs.

(*) Improved by Kovács (2022), Bocquet (2025)

Happy birthday, Martín!