Semisimplicial Types in Internal CwFs

Joshua Chen and Nicolai Kraus

TYPES 2021

Semisimplicial Types

A *semisimplicial type* is a semisimplicial object in "the" category of types.

Presented by:

$$\begin{aligned}
A_0 &: \mathsf{Type} \\
A_1 &: A_0 \times A_0 \to \mathsf{Type} \\
A_2 &: \left(\sum_{x,\, y,\, z : A_0} A_1(x, y) \times A_1(x, z) \times A_1(y, z)\right) \to \mathsf{Type} \\
&\ \ \vdots \\
A_n &: \partial\Delta^n(A_0, \ldots, A_{n-1}) \to \mathsf{Type} \\
&\ \ \vdots
\end{aligned}$$

Semisimplicial Types

Why? [Kra18]

- Building block for homotopical/higher categorical structures. [KS17; CK17; Kra21]
- Example of higher coherence we don't know how to internalize.

  This is okay:

$$\mathsf{SST}(n) \coloneqq \text{type of tuples } (A_0, \ldots, A_n)$$

[Bru; Kra14]

Semisimplicial Types

Why? [Kra18]

- Building block for homotopical/higher categorical structures. [KS17; CK17; Kra21]
- Example of higher coherence we don't know how to internalize.

  This is okay:

  $$\mathsf{SST}(n) :\equiv \text{type of tuples } (A_0, \ldots, A_n)$$

  [Bru; Kra14]

Open question

Define

$$\mathsf{SST} : \mathbb{N} \to \mathsf{Type}_1$$

in HoTT. *Can we construct semisimplicial types?* [Her15; ACK16]

HoTT in HoTT

Open question

Define the syntax of HoTT in HoTT together with interpretations into itself. *Can HoTT eat itself?* [Shu14; EX14; Buc17]

HoTT in HoTT

Open question

Define the syntax of HoTT in HoTT together with interpretations into itself. *Can HoTT eat itself?* [Shu14; EX14; Buc17]

Theorem (expected)

If HoTT can eat itself then we can construct semisimplicial types.

Set-Based Internal CwFs

An *internal CwF* $\mathcal{C}$ is a category with families [Dyb95] internal to HoTT. Consists of

$$\begin{aligned}
\mathsf{Con} &: \mathsf{Type} \\
\mathsf{Sub} &: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Type} \\
\mathsf{Ty} &: \mathsf{Con} \to \mathsf{Type} \\
\mathsf{Tm} &: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type} \\
&\vdots
\end{aligned}$$

$\mathcal{C}$ is *set-based* if Con, Sub, Ty, Tm are h-sets.

SST in Internal CwFs

Current work in progress

Construct semisimplicial types in set-based internal CwFs $\mathcal{C}$ with $\hat{\Pi}$, $\hat{\Sigma}$ and U (and $\mathbb{1}$?).
i.e. Define

$$\mathsf{SST}_{\mathcal{C}} : \mathbb{N} \to \mathsf{Con}$$

where $\mathsf{SST}_{\mathcal{C}}\, n$ is the context

$$A_0 : U,\ \ A_1 : A_0 \mathbin{\hat{\times}} A_0 \mathbin{\hat{\to}} U,\ \ \ldots,\ \ A_n : \partial\Delta^n \mathbin{\hat{\to}} U$$

Technicalities

Sketch of construction:

Assume set-based internal CwF $\mathcal{C} = (\mathsf{Con}, \mathsf{Sub}, \mathsf{Ty}, \mathsf{Tm}, \hat{\Pi}, \hat{\Sigma}, U, \ldots)$. Define

$$\mathsf{SST} : \mathbb{N} \to \mathsf{Con} \quad \ldots \; \mathsf{SST}_{k+2} \coloneqq \mathsf{SST}_{k+1}, \underbrace{\mathsf{Sk}_{(k+2,\, \ldots)} \mathrel{\hat{\to}} U}_{\text{Fillers}}$$

Technicalities

Sketch of construction:

Assume set-based internal CwF $\mathcal{C} = (\mathsf{Con}, \mathsf{Sub}, \mathsf{Ty}, \mathsf{Tm}, \hat{\Pi}, \hat{\Sigma}, U, \ldots)$. Define

$$\mathsf{SST} : \mathbb{N} \to \mathsf{Con} \quad \ldots \ \mathsf{SST}_{k+2} :\equiv \mathsf{SST}_{k+1}, \underbrace{\mathsf{Sk}_{(k+2, \ldots)} \hat{\to} U}_{\mathsf{Fillers}}$$

mutually with

$\mathsf{Sk} : (b, h, t : \mathbb{N}) \to \mathsf{Ty}\,(\mathsf{SST}_{h+1})$

Technicalities

Sketch of construction:

Assume set-based internal CwF $\mathcal{C} = (\mathsf{Con}, \mathsf{Sub}, \mathsf{Ty}, \mathsf{Tm}, \hat{\Pi}, \hat{\Sigma}, U, \ldots)$. Define

$$\mathsf{SST} : \mathbb{N} \to \mathsf{Con} \quad \ldots \; \mathsf{SST}_{k+2} :\equiv \mathsf{SST}_{k+1}, \underbrace{\mathsf{Sk}_{(k+2,\,\ldots)} \mathbin{\hat{\to}} U}_{\text{Fillers}}$$

mutually with

$$\mathsf{Sk} : (b, h, t : \mathbb{N}) \to \mathsf{Ty}\,(\mathsf{SST}_{h+1}) \quad \ldots \; \mathsf{Sk}_{(b,\,h,\,t+1)} :\equiv \widehat{\sum}_{x : \,\mathsf{Sk}_{(b,h,t)}} \underbrace{A_{h+1}\,(\mathsf{Skm}\,(\ldots, x))}_{\text{Fill the } (t+1)^{\text{th}} \text{ face}}$$

Technicalities

Sketch of construction:

Assume set-based internal CwF $\mathcal{C} = (\mathsf{Con}, \mathsf{Sub}, \mathsf{Ty}, \mathsf{Tm}, \hat{\Pi}, \hat{\Sigma}, U, \dots)$. Define

$$\mathsf{SST} : \mathbb{N} \to \mathsf{Con} \qquad \dots \; \mathsf{SST}_{k+2} :\equiv \mathsf{SST}_{k+1}, \underbrace{\mathsf{Sk}_{(k+2, \dots)} \hat{\to} U}_{\text{Fillers}}$$

mutually with

$$\mathsf{Sk} : (b, h, t : \mathbb{N}) \to \mathsf{Ty}\,(\mathsf{SST}_{h+1}) \qquad \dots \; \mathsf{Sk}_{(b,\,h,\,t+1)} :\equiv \widehat{\sum}_{x :\, \mathsf{Sk}_{(b,h,t)}} \underbrace{A_{h+1}\,(\mathsf{Skm}\,(\dots, x))}_{\text{Fill the } (t+1)^{\text{th}} \text{ face}}$$

and

$$\begin{aligned} \mathsf{Skm} : \{b, h, t, k : \mathbb{N}\} \\ &\to (f : \Delta_{+}(k, b)) \\ &\to \mathsf{Tm}\left(\mathsf{Sk}_{(b,h,t)}\right) \to \mathsf{Tm}\left(\mathsf{Sk}_{(b,h,t) \cap f}\right) \end{aligned}$$

SST in Internal CwFs

Theorem (expected)

If HoTT can eat itself then we can construct semisimplicial types.

Proof.

An interpretation of the syntax of (HoTT + a universe) as an internal set-based CwF $\mathcal{C}$ gives

$$\mathsf{HoTT\text{-}Con} : \mathsf{Con} \to \mathsf{Type}_1 .$$

$\square$

SST in Internal CwFs

Theorem (expected)

If HoTT can eat itself then we can construct semisimplicial types.

Proof.

An interpretation of the syntax of (HoTT + a universe) as an internal set-based CwF $\mathcal{C}$ gives

$$\mathsf{HoTT\text{-}Con} : \mathsf{Con} \to \mathsf{Type}_1.$$

Then we have

$$\mathsf{SST} :\equiv \mathsf{HoTT\text{-}Con} \circ \mathsf{SST}_{\mathcal{C}} : \mathbb{N} \to \mathsf{Type}_1.$$

$\square$

References I

[ACK16] Thorsten Altenkirch, Paolo Capriotti and Nicolai Kraus. "Extending Homotopy Type Theory with Strict Equality". In: *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*. Ed. by Jean-Marc Talbot and Laurent Regnier. Vol. 62. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 21:1–21:17. ISBN: 978-3-95977-022-4. DOI: 10.4230/LIPIcs.CSL.2016.21. URL: http://drops.dagstuhl.de/opus/volltexte/2016/6561.

[Bru] Guillaume Brunerie. *A Python script generating Agda code for semi-simplicial types, truncated at any given level*. Accessed 23 Apr 2021. URL: https://guillaumebrunerie.github.io/other/semisimplicial.py.

[Buc17] Ulrik Buchholtz. *Formalizing type theory in type theory using nominal techniques*. Invited talk at the Workshop on Homotopy Type Theory/Univalent Foundations. Accessed 9 Jun 2021. 2017. URL: https://hott-uf.github.io/2017/slides/Buchholtz.pdf.

[CK17] Paolo Capriotti and Nicolai Kraus. "Univalent Higher Categories via Complete Semi-Segal Types". In: *Proceedings of the ACM on Programming Languages* 2 (2017). Full version available at https://arxiv.org/abs/1707.03693, 44:1–44:29. ISSN: 2475-1421. DOI: 10.1145/3158132. URL: http://doi.acm.org/10.1145/3158132.

[Dyb95] Peter Dybjer. "Internal type theory". In: *Types for Proofs and Programs (TYPES)*. Ed. by Stefano Berardi and Mario Coppo. Vol. 1158. Lecture Notes in Computer Science. Springer-Verlag, 1995, pp. 120–134. ISBN: 978-3-540-70722-6. DOI: 10.1007/3-540-61780-9_66.

References II

[EX14] Martín Hötzel Escardó and Chuangjie Xu. *Type theory eating itself?* Accessed 9 Jun 2021. 2014. URL: https://www.cs.bham.ac.uk/~mhe/TT-perhaps-eating-itself/TT-perhaps-eating-itself.html.

[Her15] Hugo Herbelin. "A dependently-typed construction of semi-simplicial types". In: *Mathematical Structures in Computer Science* 25.5 (2015), pp. 1116–1131. DOI: 10.1017/S0960129514000528.

[Kra14] Nicolai Kraus. *A Haskell script to generate the type of n-truncated semi-simplicial types*. Available at https://nicolaikraus.github.io/docs/generateSemiSimp.hs. Accessed 23 Apr 2021. 2014.

[Kra18] Nicolai Kraus. *On the Role of Semisimplicial Types*. Extended abstract presented at the 24th International Conference on Types for Proofs and Programs. Accessed 9 Jun 2021. 2018. URL: https://nicolaikraus.github.io/docs/on_semisimplicial_types.pdf.

[Kra21] Nicolai Kraus. *Internal ∞-Categorical Models of Dependent Type Theory: Towards 2LTT Eating HoTT*. To appear, 36th Annual Symposium on Logic in Computer Science. 2021. arXiv: 2009.01883 [cs.LO].

[KS17] Nicolai Kraus and Christian Sattler. *Space-Valued Diagrams, Type-Theoretically (Extended Abstract)*. 2017. arXiv: 1704.04543 [math.LO]. URL: https://arxiv.org/abs/1704.04543v1.

References III

[Shu14] Michael Shulman. *Homotopy Type Theory should eat itself (but so far, it's too big to swallow)*. Blog post accessed 7 Jun 2021. 3 Mar. 2014. URL: https://homotopytypetheory.org/2014/03/03/hott-should-eat-itself.