Constructing inverse diagrams in (internal models of) HoTT

Josh Chen

j.w.w. Nicolai Kraus

University of Nottingham

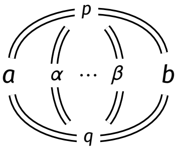
YaMCATS 29

15 Dec 2022

Background

In plain HoTT, all types A are ∞-groupoids.

- Objects are elements $a : A$
- $\hom(x, y)$ for n-cells x and y are iterated identity types



1-cells $\quad p, q : \hom(a, b) \equiv (a =_A b)$,

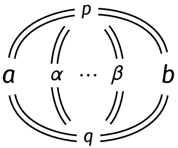
2-cells $\quad \alpha, \beta : \hom(p, q) \equiv (p =_{a=b} q)$,

etc.

Background

In plain HoTT, all types A are ∞-groupoids.

- Objects are elements $a : A$
- $\hom(x, y)$ for n-cells x and y are iterated identity types



1-cells $\quad p, q : \hom(a, b) \equiv (a =_A b)$,

2-cells $\quad \alpha, \beta : \hom(p, q) \equiv (p =_{a=b} q)$,

etc.

Question

How do we talk about $(\infty, 1)$-categories in plain homotopy type theory?

Background

In plain HoTT, all types A are ∞-groupoids.

- Objects are elements $a : A$
- $\hom(x, y)$ for n-cells x and y are iterated identity types

$$
\begin{tikzcd}
a \arrow[rr, bend left=60, equal, "p"{name=P}] \arrow[rr, bend right=60, equal, "q"'{name=Q}] & & b
\arrow[from=P, to=Q, phantom, "\alpha \ \cdots \ \beta"]
\end{tikzcd}
$$

1-cells $\quad p, q : \hom(a, b) \equiv (a =_A b)$,

2-cells $\quad \alpha, \beta : \hom(p, q) \equiv (p =_{a=b} q)$,

etc.

Question

How do we talk about $(\infty, 1)$-categories in plain homotopy type theory

…in a way that exploits HoTT's inherent higher categorical structure?

Simplicial objects in type theory?

Some models of $(\infty, 1)$-categories start with simplicial objects in some $\mathcal{C}$ ($= \mathit{Set}, \hat{\Delta}, \ldots$)
$\Longrightarrow$ Look for

1. a category $\mathcal{C}$ of type theoretic data +
2. a construction *defined in HoTT* that can *externally* be seen to give simplicial objects in $\mathcal{C}$.

Straightforward first try for (1): universe type $\mathcal{U}$ is a 1-category

- Objects: closed $\mathcal{U}$-small types
- $\mathrm{hom}(A, B) :=$ function type $A \longrightarrow B$

Might call $\mathcal{U}$-valued Δ-presheaves *simplicial types*.

Can we achieve (2)? What remains is to define $\mathcal{U}$-valued Δ-presheaves in HoTT.

Semisimplicial types

First, simplify by forgetting degeneracy maps: ask for the data of $\mathcal{U}$-valued Δ_+-presheaves, aka *semisimplicial types*.

Standard encoding of a Δ_+-presheaf $\mathcal{S}$ in $\mathcal{U}$:

$$
\begin{aligned}
& A_0 : \mathcal{U}, \quad A_1 : A_0 \to A_0 \to \mathcal{U}, \\
& A_2 : (x, y, z : A_0) \to A_1(x, y) \to A_1(x, z) \to A_1(y, z) \to \mathcal{U}, \\
& A_3 : (x, y, z, w : A_0) \to \\
& \qquad (e_{x,y} : A_1(x, y)) \to \cdots \to (e_{z,w} : A_1(z, w)) \to \\
& \qquad (f_{x,y,z} : A_2(x, y, z, e_{x,y}, e_{x,z}, e_{y,z})) \to \cdots \to (f_{y,z,w} : A_2(y, \ldots, e_{z,w})) \to \mathcal{U}, \\
& \ldots
\end{aligned}
$$

$\mathcal{S}_n$ is the *total space* of A_n. Face maps are given by projecting out subtuples.

Semisimplicial types

First, simplify by forgetting degeneracy maps: ask for the data of $\mathcal{U}$-valued Δ_+-presheaves, aka *semisimplicial types*.

Standard encoding of a Δ_+-presheaf $\mathcal{S}$ in $\mathcal{U}$:

$$A_0 : \mathcal{U}, \quad \mathcal{S}_0 = A_0, \quad A_1 : A_0 \to A_0 \to \mathcal{U}, \quad \mathcal{S}_1 = (x, y : A_0) \times A_1(x, y)$$

$$A_2 : (x, y, z : A_0) \to A_1(x, y) \to A_1(x, z) \to A_1(y, z) \to \mathcal{U},$$

$$\mathcal{S}_2 = (x, y, z : A_0) \times (e_{x,y} : A_1(x, y)) \times (e_{x,z} : A_1(x, z)) \times (e_{y,z} : A_1(y, z)) \times A_2(x, y, z, e_{x,y}, e_{x,z}, e_{y,z}),$$

$$\begin{aligned} A_3 :\ & (x, y, z, w : A_0) \to \\ & (e_{x,y} : A_1(x, y)) \to \cdots \to (e_{z,w} : A_1(z, w)) \to \\ & (f_{x,y,z} : A_2(x, y, z, e_{x,y}, e_{x,z}, e_{y,z})) \to \cdots \to (f_{y,z,w} : A_2(y, \ldots, e_{z,w})) \to \mathcal{U}, \quad \ldots \end{aligned}$$

$\mathcal{S}_n$ is the *total space* of A_n. Face maps are given by projecting out subtuples.

Semisimplicial types

Some observations:

- The type of each A_n depends on $A_0, \ldots, A_{n-1}$.
- For given *fixed* n, can define the type of tuples $(A_0, \ldots, A_n)$, e.g. fixing $n = 2$,

```
record SST₂ : Type₁ where
  A₀ : Type₀
  A₁ : A₀ → A₀ → Type₀
  A₂ : (x y z : A₀) → A₁ x y → A₁ x z → A₁ y z → Type₀
```

Question

Define *in HoTT* a function $\mathrm{SST} : \mathbb{N} \longrightarrow \mathcal{U}^+$ so that $\mathrm{SST}(n)$ is the type of sequences $(A_0, \ldots, A_n)$?

Semisimplicial types

Some observations:

- The type of each A_n depends on $A_0, \ldots, A_{n-1}$.
- For given *fixed* n, can define the type of tuples $(A_0, \ldots, A_n)$, e.g. fixing $n = 2$,

```
record SST₂ : Type₁ where
  A₀ : Type₀
  A₁ : A₀ → A₀ → Type₀
  A₂ : (x y z : A₀) → A₁ x y → A₁ x z → A₁ y z → Type₀
```

Open Question — "Constructing semisimplicial types"

Define *in HoTT* a function $\mathsf{SST} : \mathbb{N} \longrightarrow \mathcal{U}^+$ so that $\mathsf{SST}(n)$ is the type of sequences $(A_0, \ldots, A_n)$?

More generally, is the type of semisimplicial types definable in (plain) HoTT?

Semisimplicial types

Some observations:

- The type of each A_n depends on $A_0, \ldots, A_{n-1}$.
- For given *fixed* n, can define the type of tuples $(A_0, \ldots, A_n)$, e.g. fixing $n = 2$,

```
record SST₂ : Type₁ where
  A₀ : Type₀
  A₁ : A₀ → A₀ → Type₀
  A₂ : (x y z : A₀) → A₁ x y → A₁ x z → A₁ y z → Type₀
```

Open Question — "Constructing semisimplicial types"

Define *in HoTT* a function $\mathrm{SST} : \mathbb{N} \longrightarrow \mathcal{U}^+$ so that $\mathrm{SST}(n)$ is the type of sequences $(A_0, \ldots, A_n)$?

More generally, is the type of semisimplicial types definable in (plain) HoTT?

Obstruction: coherence problem because equality in HoTT is structure, not property.

Semisimplicial types

Difficulty: haven't managed to internalize the matching objects of semisimplicial types.

- For nice enough $\mathcal{C}$, can construct "Reedy fibrant" $\mathcal{C}$-valued diagrams indexed by inverse $\mathcal{I}$.
- Construction by well founded induction, using certain limits—the *matching objects*—at each stage.
- Matching objects give a functor *M* from (a subcategory of) *CoSv*$(\mathcal{I})$ to $\mathcal{C}$.
- Coherence problem arises from failure of *M* to be strict for $\mathcal{C} = \mathcal{U}$.

Inverse diagrams in internal models of HoTT

Current work:

- Formulate models of type theory inside HoTT.
- Construct semisimplicial types and more general inverse diagrams in the model.

Control the height of the tower of coherence conditions by truncating the internal model.

Inverse diagrams in internal models of HoTT

Current work:

- Formulate models of type theory inside HoTT.
- Construct semisimplicial types and more general inverse diagrams in the model.

Control the height of the tower of coherence conditions by truncating the internal model.

Goals:

- Investigate, inside HoTT, minimal models in which coherence issues arise.
- Determine minimal sufficient conditions for the model—and by extension, type theory—to support semisimplicial types.
- Develop constructions to test the theory of higher models of type theory.
- Bonus 🎁—provide main part of proof relating open problems in HoTT.

Technical outline

- Internal model: Categories with families
- Diagrams:
  1. The index categories we use
  2. Matching objects
  3. Constructing diagrams in internal CwFs

Categories with families

Common categorical model of type theory:

Definition

A *category with families* is a category Con together with

- a choice of terminal object $\mathbb{1} \in Con$
- $Ty : Con^{op} \longrightarrow Set$
- $Tm : \big(el(Ty)\big)^{op} \longrightarrow Set$
- For every $(\Gamma, A) \in el(Ty)$, a choice of terminal object in

$$el_{Con/\Gamma}\Big[Tm\big(dom(\cdot), Ty(\cdot)(A)\big)\Big].$$

In particular, have context extension $\Gamma \triangleright A$ and substitution on types $A[\sigma]$ and terms $a[\sigma]$.

Countable locally finite inverse categories

For a category $\mathcal{I}$, define $j \prec i$ iff there's a morphism $j \leftarrow i$.
$\mathcal{I}$ is *inverse* if $\prec$ is well founded.

Countable locally finite inverse categories

For a category $\mathcal{I}$, define $j \prec i$ iff there's a morphism $j \leftarrow i$.
$\mathcal{I}$ is *inverse* if $\prec$ is well founded.

Definition

An inverse category $\mathcal{I}$ is *countable and locally finite* if

1. there is $\#: Ob(\mathcal{I}) \cong \mathbb{N}$ such that $\#j < \#i$ whenever $j \prec i$,
2. for $i, j \in Ob(\mathcal{I})$, $\hom(i, j)$ is finite and totally ordered,
3. $\hom(i, i) \cong Fin(1)$ for all i.

Write $\mathrm{idx}: \hom(i, j) \cong Fin(|\hom(i, j)|)$ for the canonical order isomorphism.

Countable locally finite inverse categories

For a category $\mathcal{I}$, define $j \prec i$ iff there's a morphism $j \leftarrow i$.
$\mathcal{I}$ is *inverse* if $\prec$ is well founded.

Definition

An inverse category $\mathcal{I}$ is *countable and locally finite* if

1. there is $\# : Ob(\mathcal{I}) \cong \mathbb{N}$ such that $\#j < \#i$ whenever $j \prec i$,
2. for $i, j \in Ob(\mathcal{I})$, $\hom(i, j)$ is finite and totally ordered,
3. $\hom(i, i) \cong Fin(1)$ for all i.

Write $\mathrm{idx} : \hom(i, j) \cong Fin(|\hom(i, j)|)$ for the canonical order isomorphism.

Examples: Δ_+ (also $\square_+$, Ω_+)

We will refer to objects $i \in Ob(\mathcal{I})$ as natural numbers.

0 is always $\prec$-minimal.

Matching objects

Let $\mathcal{I}$ be inverse and $i \in Ob(\mathcal{I})$.

$$\begin{aligned} \mathcal{I}_{\prec i},\ \mathcal{I}_{\preceq i} &\quad \text{— full subcategories on objects } j \prec i \text{ and } j \preceq i\text{, resp.} \\ {}^{i/\!\!/}\mathcal{I} &\quad \text{— full subcategory on } Ob({}^{i/}\mathcal{I}) - \{id_i\}. \end{aligned}$$

The codomain forgetful functor U projects from ${}^{i/\!\!/}\mathcal{I}$ to $\mathcal{I}_{\prec i}$.

Matching objects

Let $\mathcal{I}$ be inverse and $i \in Ob(\mathcal{I})$.

$$\begin{aligned} \mathcal{I}_{\prec i},\ \mathcal{I}_{\preceq i} &\quad \text{— full subcategories on objects } j \prec i \text{ and } j \preceq i \text{, resp.} \\ {}^{i/\!\!/}\mathcal{I} &\quad \text{— full subcategory on } Ob({}^{i/}\mathcal{I}) - \{id_i\}. \end{aligned}$$

The codomain forgetful functor U projects from ${}^{i/\!\!/}\mathcal{I}$ to $\mathcal{I}_{\prec i}$.

Definition

Let $i \in Ob(\mathcal{I})$ and $\mathcal{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. The *matching object* M_i of $\mathcal{D}$ is the limit

$$\lim_{{}^{i/\!\!/}\mathcal{I}} (\mathcal{D} \circ U).$$

Matching objects

Definition

Let $i \in Ob(\mathcal{I})$ and $\mathcal{D} : \mathcal{I}_{<i} \longrightarrow \mathcal{C}$. The *matching object* M_i of $\mathcal{D}$ is given by $M_i = \lim_{i /\!\!/ \mathcal{I}} (\mathcal{D} \circ U)$.

Matching objects

Definition

Let $i \in Ob(\mathcal{I})$ and $\mathscr{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. The *matching object* M_i of $\mathscr{D}$ is given by $M_i = \lim_{i /\!\!/ \mathcal{I}} (\mathscr{D} \circ U)$.

1. If i is $\prec$-minimal, ${}^{i /\!\!/}\mathcal{I}$ is empty and M_i is terminal.

Matching objects

Definition

Let $i \in Ob(\mathcal{I})$ and $\mathcal{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. The *matching object* M_i of $\mathcal{D}$ is given by $M_i = \lim_{i/\!\!/\mathcal{I}}(\mathcal{D} \circ U)$.

1. If i is $\prec$-minimal, ${}^{i/\!\!/}\mathcal{I}$ is empty and M_i is terminal.
2. Assume $\mathcal{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. To extend $\mathcal{D}$ to $\mathcal{I}_{\preceq i}$, simply give $\mathcal{D}_i \in \mathcal{C}$ and an $f : \mathcal{D}_i \longrightarrow M_i$.

Matching objects

Definition

Let $i \in Ob(\mathcal{I})$ and $\mathscr{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. The *matching object* M_i of $\mathscr{D}$ is given by $M_i = \lim_{i /\!\!/ \mathcal{I}} (\mathscr{D} \circ U)$.

1. If i is $\prec$-minimal, ${}^{i}/\!\!/\mathcal{I}$ is empty and M_i is terminal.
2. Assume $\mathscr{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. To extend $\mathscr{D}$ to $\mathcal{I}_{\preceq i}$, simply give $\mathscr{D}_i \in \mathcal{C}$ and an $f : \mathscr{D}_i \longrightarrow M_i$.

$\Longrightarrow$ Inductive construction of diagrams $\mathscr{D}$ (c.f. Makkai, Shulman)

Matching objects

Definition

Let $i \in Ob(\mathcal{I})$ and $\mathcal{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. The *matching object* M_i of $\mathcal{D}$ is given by $M_i = \lim_{i /\!/ \mathcal{I}} (\mathcal{D} \circ U)$.

1. If i is $\prec$-minimal, ${}^{i/\!/}\mathcal{I}$ is empty and M_i is terminal.
2. Assume $\mathcal{D} : \mathcal{I}_{\prec i} \longrightarrow \mathcal{C}$. To extend $\mathcal{D}$ to $\mathcal{I}_{\preceq i}$, simply give $\mathcal{D}_i \in \mathcal{C}$ and an $f : \mathcal{D}_i \longrightarrow M_i$.

$\Longrightarrow$ Inductive construction of diagrams $\mathcal{D}$ (c.f. Makkai, Shulman)

Remark

When $\mathcal{C} = \mathcal{U}^+$, giving $(\mathcal{D}_i, f)$ is equivalent to giving a morphism $A_i : M_i \longrightarrow \mathcal{U}$.

If M_i were to exist in $\mathcal{U}$, would get the components for semisimplicial types in the case $\mathcal{I} = \Delta_+^{op}$.

Refining M_i with linear cosieves

Definition

For $h < i \in \mathcal{I}$ and $t \leq |\hom(i, h)|$, define the *linear cosieve of shape* (i, h, t) by

$$S_{i,h,t} := \left(\bigcup_{k<h} \hom(i, k) \right) \cup \left\{ f \in \hom(i, h) \mid \operatorname{idx}(f) < t \right\}.$$

Define ${}^{i,h,t/}\mathcal{I}$ to be the full subcategory of ${}^{i/}\mathcal{I}$ on $S_{i,h,t}$.

Refining M_i with linear cosieves

Definition

For $h < i \in \mathcal{I}$ and $t \leq |\mathrm{hom}(i,h)|$, define the *linear cosieve of shape* (i,h,t) by

$$S_{i,h,t} := \left(\bigcup_{k<h} \mathrm{hom}(i,k) \right) \cup \left\{ f \in \mathrm{hom}(i,h) \mid \mathrm{idx}(f) < t \right\}.$$

Define ${}^{i,h,t/}\mathcal{I}$ to be the full subcategory of ${}^{i/}\mathcal{I}$ on $S_{i,h,t}$.

Get the following filtration of ${}^{i/\!/}\mathcal{I}$:

$$\varnothing = {}^{i,0,0/}\mathcal{I} \hookrightarrow {}^{i,0,1/}\mathcal{I} \hookrightarrow \cdots \hookrightarrow {}^{i,h,|\mathrm{hom}(i,h)|/}\mathcal{I} = {}^{i,h+1,0/}\mathcal{I} \hookrightarrow \cdots$$
$$\hookrightarrow {}^{i,i-1,|\mathrm{hom}(i,i-1)|/}\mathcal{I} = {}^{i/\!/}\mathcal{I}$$

Computing matching objects

From

$$\varnothing = {}^{i,0,0/}\mathcal{I} \hookrightarrow {}^{i,0,1/}\mathcal{I} \hookrightarrow \cdots \hookrightarrow {}^{i,h,|\hom(i,h)|/}\mathcal{I} = {}^{i,h+1,0/}\mathcal{I} \hookrightarrow \cdots$$
$$\hookrightarrow {}^{i,i-1,|\hom(i,i-1)|/}\mathcal{I} = {}^{i/\!\!/}\mathcal{I}$$

Computing matching objects

From

$$\varnothing = {}^{i,0,0/}\mathcal{I} \hookrightarrow {}^{i,0,1/}\mathcal{I} \hookrightarrow \cdots \hookrightarrow {}^{i,h,|\hom(i,h)|/}\mathcal{I} = {}^{i,h+1,0/}\mathcal{I} \hookrightarrow \cdots$$
$$\hookrightarrow {}^{i,i-1,|\hom(i,i-1)|/}\mathcal{I} = {}^{i/\!\!/}\mathcal{I}$$

we will recursively compute a sequence of *partial matching objects*

$$\mathbb{1} = M_{i,0,0} \rightsquigarrow M_{i,0,1} \rightsquigarrow \cdots \rightsquigarrow M_{i,h,|\hom(i,h)|} = M_{i,h+1,0} \rightsquigarrow \cdots$$
$$\rightsquigarrow M_{i,i-1,|\hom(i,i-1)|} = M_i,$$

where $M_{i,h,t} \approx \lim_{{}^{i,h,t/\!\!/}\mathcal{I}}(\mathscr{D} \circ U)$.

Constructing diagrams in internal CwFs

From now on,

- Take $\mathcal{C} = \mathit{Con}$ of an internal CwF equipped with Π-types and a universe type V
- Assume $\mathcal{I}$ to be inverse, countable and locally finite (for intuition, take $\mathcal{I} = \Delta_+^{op}$)
- Work in HoTT (informally)

  Note: Categorical terms will still have the HoTT equivalent of their usual meanings.

Constructing diagrams in internal CwFs

From now on,

- Take $\mathcal{C} = \mathit{Con}$ of an internal CwF equipped with Π-types and a universe type V
- Assume $\mathcal{I}$ to be inverse, countable and locally finite (for intuition, take $\mathcal{I} = \Delta_+^{op}$)
- Work in HoTT (informally)

  Note: Categorical terms will still have the HoTT equivalent of their usual meanings.

Warning: Actual construction is a large mutually recursive definition with seven main components, formalized in Agda for precision.

This talk: main ideas for key components.

Constructing diagrams in internal CwFs

"Main" component $SCT : \mathbb{N} \longrightarrow Con$.

$$\begin{aligned} SCT(0) &:\equiv \mathbb{1} \\ SCT(1) &:\equiv SCT(0) \triangleright V \\ SCT(n+1) &:\equiv SCT(n) \triangleright \Pi^{*}_{n,(n,n-1,|\mathrm{hom}(n,n-1)|)} V \end{aligned}$$

where

- $\Pi^{*}_{n,(i,h,t)} : Ty(M_{n,(i,h,t)}) \longrightarrow Ty(SCT(n))$ is a HoTT function.

Constructing diagrams in internal CwFs

"Main" component $SCT : \mathbb{N} \longrightarrow Con$.

$$\begin{aligned} SCT(0) &:\equiv \mathbb{1} \\ SCT(1) &:\equiv SCT(0) \triangleright V \\ SCT(n+1) &:\equiv SCT(n) \triangleright \Pi^{*}_{n,(n,n-1,|\hom(n,n-1)|)} V \end{aligned}$$

where

- $\Pi^{*}_{n,(i,h,t)} : Ty(M_{n,(i,h,t)}) \longrightarrow Ty(SCT(n))$ is a HoTT function.
- $M_{n,(i,h,t)} : Con$ is the context $SCT(n)$ extended with a telescope of components of the *(i,h,t)-partial matching object.*

  e.g. for $\mathcal{I} = \Delta_{+}^{op}$,

$$M_{n,(1,0,2)} \equiv SCT(n) \triangleright A_0 \triangleright A_0$$

Constructing diagrams in internal CwFs

"Main" component $SCT : \mathbb{N} \longrightarrow Con$.

$$\begin{aligned} SCT(0) &:\equiv \mathbb{1} \\ SCT(1) &:\equiv SCT(0) \triangleright V \\ SCT(n+1) &:\equiv SCT(n) \triangleright \Pi^*_{n,(n,n-1,|\hom(n,n-1)|)} V \end{aligned}$$

where

- $\Pi^*_{n,(i,h,t)} : Ty(M_{n,(i,h,t)}) \longrightarrow Ty(SCT(n))$ is a HoTT function.
- $M_{n,(i,h,t)} : Con$ is the context $SCT(n)$ extended with a telescope of components of the *(i,h,t)-partial matching object.*
  e.g. for $\mathcal{I} = \Delta_+^{op}$,
  $$M_{n,(1,0,2)} \equiv SCT(n) \triangleright A_0 \triangleright A_0$$
- $\Pi^*_{n,(i,h,t)}$ iteratedly applies the isomorphism $Ty(\Gamma \triangleright A) \cong Ty(\Gamma)$ given by Π-introduction.
  e.g. for $\mathcal{I} = \Delta_+^{op}$,
  $$\Pi^*_{n,(1,0,2)} V \;\equiv\; \Pi^*_{n,(1,0,1)}(\Pi_{A_0} V) \;\equiv\; \Pi_{A_0}\Pi_{A_0} V$$

Constructing diagrams in internal CwFs

From earlier: want to recursively compute partial matching objects $M_{i,h,t}$

$$\begin{aligned}\mathbb{1} = M_{i,0,0} \rightsquigarrow M_{i,0,1} \rightsquigarrow \cdots \rightsquigarrow M_{i,h,|\mathrm{hom}(i,h)|} = M_{i,h+1,0} \rightsquigarrow \cdots \\ \rightsquigarrow M_{i,i-1,|\mathrm{hom}(i,i-1)|} = M_i\end{aligned}$$

For technical reasons, also index over n.

Constructing diagrams in internal CwFs

From earlier: want to recursively compute partial matching objects $M_{i,h,t}$

$$\mathbb{1} = M_{i,0,0} \rightsquigarrow M_{i,0,1} \rightsquigarrow \cdots \rightsquigarrow M_{i,h,|\hom(i,h)|} = M_{i,h+1,0} \rightsquigarrow \cdots$$
$$\rightsquigarrow M_{i,i-1,|\hom(i,i-1)|} = M_i$$

For technical reasons, also index over n.
First two cases easy:

$$\begin{aligned} M_{n,(i,0,0)} &:\equiv SCT(n), \\ M_{n,(i,h+1,0)} &:\equiv M_{n,(i,h,|\hom(i,h)|)}. \end{aligned}$$

Computing matching objects

Final case $M_{n,(i,h,t+1)}$ requires some machinery.

Computing matching objects

Final case $M_{n,(i,h,t+1)}$ requires some machinery.

Definition

Let S be a cosieve under i in $\mathcal{I}$, and $f \in \hom(i, j)$.
The *restriction* $(S \cdot f)$ *of* S *along* f is the cosieve under j given by

$$S \cdot f := \left\{\alpha : j \longrightarrow k \mid k \in Ob(I),\ \alpha \circ f \in S\right\}.$$

Computing matching objects

Final case $M_{n,(i,h,t+1)}$ requires some machinery.

Definition

Let S be a cosieve under i in $\mathcal{I}$, and $f \in \mathrm{hom}(i,j)$.
The *restriction* $(S \cdot f)$ *of* S *along* f is the cosieve under j given by

$$S \cdot f := \left\{\alpha : j \longrightarrow k \mid k \in Ob(I),\ \alpha \circ f \in S\right\}.$$

Definition

A countable and locally finite inverse $\mathcal{I}$ is *well oriented* if for all $f \in \mathrm{hom}(x,y)$ and $g, h \in \mathrm{hom}(y,z)$,

$$g < h \implies g \circ f \leq h \circ f.$$

Examples: Δ_+, $\square_+$ (Ω_+?...)

Partial matching object as functor

Lemma

In a well oriented inverse category, the restriction of a linear cosieve $S_{i,h,t}$ along any $f \in \hom(i,j)$ is a linear cosieve.

$$S_{i,h,t} \xrightarrow{f} S \cdot f = S_{j,h',t'}$$

Thus linear cosieves organize into a full subcategory $\mathit{LCoSv}(\mathcal{I})$ of $\mathit{CoSv}(\mathcal{I})$.

Partial matching object as functor

Lemma

In a well oriented inverse category, the restriction of a linear cosieve $S_{i,h,t}$ along any $f \in \hom(i,j)$ is a linear cosieve.

$$S_{i,h,t} \xrightarrow{f} S \cdot f = S_{j,h',t'}$$

Thus linear cosieves organize into a full subcategory $LCoSv(\mathcal{I})$ of $CoSv(\mathcal{I})$.

Key Idea

View partial matching objects as the object part of a weak functorial action $LCoSv(\mathcal{I}) \longrightarrow Con$, and simultaneously define the action on morphisms

$$\vec{M}_{n,(i,h,t)}(f) : Sub(M_{n,(i,h,t)}\ ,\ M_{n,(i,h,t)\cdot f})$$

(definition omitted in this talk)

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h\left[\vec{M}_{n,(i,h,t)}(\bar{t})\right]$$

where

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h \left[\vec{M}_{n,(i,h,t)}\,(\bar{t})\right]$$

where

- A_h is constructed earlier (since $h < i$) by another component of the mutually recursive definition

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h\left[\vec{M}_{n,(i,h,t)}(\bar{t})\right]$$

where

- A_h is constructed earlier (since $h < i$) by another component of the mutually recursive definition
- A_h is an open term of type V in context $M_{n,(h,h-1,|\mathrm{hom}(h,h-1)|)}$

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h \left[\vec{M}_{n,(i,h,t)}(\bar{t})\right]$$

where

- A_h is constructed earlier (since $h < i$) by another component of the mutually recursive definition
- A_h is an open term of type V in context $M_{n,(h,h-1,|\mathrm{hom}(h,h-1)|)}$
- $\bar{t} \in \mathrm{hom}(i,h)$ is the morphism for which $\mathrm{idx}(\bar{t}) = t$

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h\left[\vec{M}_{n,(i,h,t)}(\bar{t})\right]$$

where

- A_h is constructed earlier (since $h < i$) by another component of the mutually recursive definition
- A_h is an open term of type V in context $M_{n,(h,h-1,|\mathrm{hom}(h,h-1)|)}$
- $\bar{t} \in \mathrm{hom}(i,h)$ is the morphism for which $\mathrm{idx}(\bar{t}) = t$
- $\vec{M}_{n,(i,h,t)}(\bar{t})$ is a substitution from $M_{n,(i,h,t)}$ to $M_{n,(i,h,t)\cdot\bar{t}}$

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h\left[\vec{M}_{n,(i,h,t)}(\bar{t})\right]$$

where

- A_h is constructed earlier (since $h < i$) by another component of the mutually recursive definition
- A_h is an open term of type V in context $M_{n,(h,h-1,|\mathrm{hom}(h,h-1)|)}$
- $\bar{t} \in \mathrm{hom}(i,h)$ is the morphism for which $\mathrm{idx}(\bar{t}) = t$
- $\vec{M}_{n,(i,h,t)}(\bar{t})$ is a substitution from $M_{n,(i,h,t)}$ to $M_{n,(i,h,t)\cdot\bar{t}}$

Constructing diagrams in internal CwFs

Now we can define

$$M_{n,(i,h,t+1)} :\equiv M_{n,(i,h,t)} \triangleright A_h \left[\vec{M}_{n,(i,h,t)}(\bar{t})\right]$$

where

- A_h is constructed earlier (since $h < i$) by another component of the mutually recursive definition
- A_h is an open term of type V in context $M_{n,(h,h-1,|\mathrm{hom}(h,h-1)|)}$
- $\bar{t} \in \mathrm{hom}(i,h)$ is the morphism for which $\mathrm{idx}(\bar{t}) = t$
- $\vec{M}_{n,(i,h,t)}(\bar{t})$ is a substitution from $M_{n,(i,h,t)}$ to $M_{n,(i,h,t)\cdot\bar{t}}$

Lemma

Let $\mathcal{I}$ be well oriented, $S_{i,h,t}$ be a linear sieve, $f \in \mathrm{hom}(i,j)$ and $j \leq h$. Then

$$S_{i,h,t} \cdot f = S_{j,j-1,|\mathrm{hom}(j,j-1)|}.$$

Constructing diagrams in internal CwFs

Summary:

- Have $SCT : \mathbb{N} \longrightarrow Con$.
  When $\mathcal{I} = \Delta_{+}^{op}$, gives the components of semisimplicial types in the internal CwF.

Constructing diagrams in internal CwFs

Summary:

- Have $SCT : \mathbb{N} \longrightarrow Con$.
  When $\mathcal{I} = \Delta_+^{op}$, gives the components of semisimplicial types in the internal CwF.
- Goes via a large mutually recursive definition, with all components very closely intertwined.

Constructing diagrams in internal CwFs

Elided in this talk:

- Functoriality and coherence witnesses for M and $\vec{M}$.

Constructing diagrams in internal CwFs

Elided in this talk:

- Functoriality and coherence witnesses for M and $\vec{M}$.
- An extra well definedness proof: equivalent shapes of linear cosieves

$$(i, h + 1, 0) \sim (i, h, |\hom(i, h)|)$$

should give rise to equal partial matching contexts

$$M_{n,(i,h+1,0)} = M_{n,(i,h,|\hom(i,h)|)}.$$

Constructing diagrams in internal CwFs

Elided in this talk:

- Functoriality and coherence witnesses for M and $\vec{M}$.
- An extra well definedness proof: equivalent shapes of linear cosieves

$$(i, h+1, 0) \sim (i, h, |\mathrm{hom}(i, h)|)$$

  should give rise to equal partial matching contexts

$$M_{n,(i,h+1,0)} = M_{n,(i,h,|\mathrm{hom}(i,h)|)}.$$

- Dealing with explicit weakenings of the internal CwF.

Coda—Relating open problems

Open Question "Does HoTT interpret itself?"

Define a type *Syn* encoding the syntax of HoTT, plus interpretation function

$$[\![\cdot]\!] : \mathit{Syn} \to \mathcal{U}$$

sending syntax to their canonical interpretations (context expressions to nested Σ-types, type expressions to type families, etc.)?

Coda—Relating open problems

Open Question "Does HoTT interpret itself?"

Define a type *Syn* encoding the syntax of HoTT, plus interpretation function

$$[\![\cdot]\!] : \mathit{Syn} \to \mathcal{U}$$

sending syntax to their canonical interpretations (context expressions to nested Σ-types, type expressions to type families, etc.)?

Equivalently, find a notion of "model of type theory" such that

1. The syntax is initial, and
2. The "standard model" given by a universe type is an instance?

Coda—Relating open problems

In particular, a positive answer would include the data of a morphism

$$[\![\cdot]\!] \colon \mathit{Con}_{\mathit{Syn}} \to \mathcal{U}.$$

In particular, a positive answer would include the data of a morphism

$$[\![\cdot]\!] : \mathit{Con}_{\mathit{Syn}} \to \mathcal{U}.$$

Our construction gives a function $\mathit{SST} : \mathbb{N} \to \mathit{Con}$ for any CwF, in particular for the syntax *Syn*.
$\Longrightarrow$ Just precompose with *SST* to get

$$[\![\cdot]\!] \circ \mathit{SST} : \mathbb{N} \to \mathcal{U}!$$

Coda—Relating open problems

In particular, a positive answer would include the data of a morphism

$$[\![\cdot]\!] : Con_{Syn} \longrightarrow \mathcal{U}.$$

Our construction gives a function $SST : \mathbb{N} \longrightarrow Con$ for any CwF, in particular for the syntax Syn.
$\Longrightarrow$ Just precompose with SST to get

$$[\![\cdot]\!] \circ SST : \mathbb{N} \longrightarrow \mathcal{U}!$$

Lemma

If HoTT interprets itself, then semisimplicial types are definable in HoTT.

(conjectured by Shulman)

Coda—Relating open problems

In particular, a positive answer would include the data of a morphism

$$[\![\cdot]\!] : Con_{Syn} \longrightarrow \mathcal{U}.$$

Our construction gives a function $SST : \mathbb{N} \longrightarrow Con$ for any CwF, in particular for the syntax *Syn*.
$\Longrightarrow$ Just precompose with *SST* to get

$$[\![\cdot]\!] \circ SST : \mathbb{N} \longrightarrow \mathcal{U}!$$

Lemma

If HoTT interprets itself, then semisimplicial types are definable in HoTT.

(conjectured by Shulman)

Thanks!