Internal $\infty$-Categorical Models of Dependent Type Theory

Towards 2LTT Eating HoTT

Nicolai Kraus

LICS’21 (Rome/online), 29 June 2021
Why?
Goal: Define what a model of type theory is – in type theory!
(in particular: intended initial model ~ “syntax”)
Goal: Define what a model of type theory is
– in type theory!
(in particular: intended initial model ~ “syntax”)

Peter Dybjer, 2005: *Internal Type Theory* – Category with Families ("CwF")

Danielsson 2006
Chapman 2009
McBride 2010
Shulman 2014
Escardó-Xu 2014
K. 2015
Altenkirch-Kaposi 2016
Bucholtz 2017
Abel-Öhman-Vezzosi 2017
Ahrens-Lumsdaine-Voevodsky 2017/18
Brunerie-de Boer 2018–20
Lumsdaine-Mörtberg 2018–20
Kaposi-Kovács-K., 2019
Goal: Define what a model of type theory is – in type theory!
(in particular: intended initial model ~ “syntax”)

Peter Dybjer, 2005: *Internal Type Theory* – Category with Families (“CwF”)

Danielsson 2006
Chapman 2009
McBride 2010
Shulman 2014
Escardó-Xu 2014
K. 2015
Altenkirch-Kaposi 2016
Bucholtz 2017
Abel-Öhman-Vezzosi 2017
Ahrens-Lumsdaine-Voevodsky 2017/18
Brunerie-de Boer 2018–20
Lumsdaine-Mörtberg 2018–20
Kaposi-Kovács-K., 2019

```
record CwF : Set₁ where
  field
  Con : Set
  Sub : Con → Con → Set
  Ty : Con → Set
  Tm : (Γ : Con) → Ty Γ → Set

  • : Con
  _ₒₒ : (Γ : Con) → Ty Γ → Con

  -- (and so on)
```
Goal: Define what a model of type theory is – in type theory!
(in particular: intended initial model ~ “syntax”)

Peter Dybjer, 2005: *Internal Type Theory* – Category with Families (“CwF”)

Danielsson 2006
Chapman 2009
McBride 2010
Shulman 2014
Escardó-Xu 2014
K. 2015
Altenkirch-Kaposi 2016
Bucholtz 2017
Abel-Öhman-Vezzosi 2017
Ahrens-Lumsdaine-Voevodsky 2017/18
Brunerie-de Boer 2018–20
Lumsdaine-Mörtberg 2018–20
Kaposi-Kovács-K., 2019

```haskell
record CwF : Set₁ where
  field
    Con : Set
    Sub : Con → Con → Set
    Ty : Con → Set
    Tm : (Γ : Con) → Ty Γ → Set

  • : Con
  _⇒_ : (Γ : Con) → Ty Γ → Con
```

-- (and so on)
CwF definition as a Generalised Algebraic Theory

\[
\begin{align*}
\text{Con} & : \text{Type} \\
\text{Sub} & : \text{Con} \to \text{Con} \to \text{Type} \\
\_ \diamond \_ & : \text{Sub} \Theta \Delta \to \text{Sub} \Gamma \Theta \to \text{Sub} \Gamma \Delta \\
\text{assoc} & : (\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu) \\
\text{id} & : \text{Sub} \Gamma \Gamma \\
\text{idl}_\sigma & : \text{id} \diamond \sigma = \sigma \\
\text{idr}_\sigma & : \sigma \diamond \text{id} = \sigma \\
\_ & : \text{Con} \\
\epsilon & : \text{Sub} \Gamma \_ \\
\text{\_\_} & : \forall (\sigma : \text{Sub} \Gamma \_). \sigma = \epsilon \\
\text{Ty} & : \text{Con} \to \text{Type} \\
\_ [\_]^\text{T} & : \text{Ty} \Delta \to \text{Sub} \Gamma \Delta \to \text{Ty} \Gamma \\
[\text{id}]^\text{T} & : A[\text{id}]^\text{T} = A \\
[\text{\_\_}]^\text{T} & : A[\sigma \diamond \delta]^\text{T} = A[\sigma]^\text{T}[\delta]^\text{T} \\
\text{Tm} & : (\Gamma : \text{Con}) \to \text{Ty} \Gamma \to \text{Type} \\
\_ [\_]^\text{T} & : \text{Tm} \Delta A \to (\sigma : \text{Sub} \Gamma \Delta) \to \text{Tm} \Gamma (A[\sigma]^\text{T}) \\
[\text{id}]^\text{T} & : t[\text{id}]^\text{T} = t \\
[\_\_] & : t[\sigma \diamond \delta]^\text{T} = t[\sigma]^\text{T}[\delta]^\text{T}t \\
\text{\_ \_ \_} & : (\Gamma : \text{Con}) \to \text{Ty} \Gamma \to \text{Con} \\
\text{p} & : \text{Sub} (\Gamma \triangleright A) \Gamma \\
\text{q} & : \text{Tm} (\Gamma \triangleright A) (A[p]^\text{T}) \\
\_ \_ \_ \_ & : (\sigma : \text{Sub} \Gamma \Delta) \to \text{Tm} \Gamma (A[\sigma]^\text{T}) \to \text{Sub} \Gamma (\Delta \triangleright A) \\
\triangleright \beta_1 & : p \diamond (\sigma, t) = \sigma \\
\triangleright \beta_2 & : q[\sigma, t]^\text{T} = tt \\
\triangleright \eta & : (p, q) = \text{id} \\
\_ \_ \_ \_ \_ & : (\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^\text{T}t) \\
\text{(Good definition in a type theory with K/UIP)}
\end{align*}
\]
CwF definition as a Generalised Algebraic Theory

\[
\begin{align*}
\text{Con} & : \text{Type} \\
\text{Sub} & : \text{Con} \to \text{Con} \to \text{Type} \\
\_ \triangleleft \_ & : \text{Sub} \Theta \Delta \to \text{Sub} \Gamma \Theta \to \text{Sub} \Gamma \Delta \\
\text{assoc} & : (\sigma \triangleleft \delta) \triangleleft \nu = \sigma \triangleleft (\delta \triangleleft \nu) \\
\text{id} & : \text{Sub} \Gamma \Gamma \\
\text{idl}_\sigma & : \text{id} \triangleleft \sigma = \sigma \\
\text{idr}_\sigma & : \sigma \triangleleft \text{id} = \sigma
\end{align*}
\]

\[
\begin{align*}
\text{Ty} & : \text{Con} \to \text{Type} \\
\_ \downarrow \_ & : \text{Ty} \Delta \to \text{Sub} \Gamma \Delta \to \text{Ty} \Gamma \\
\text{id}^\text{T} & : A [\text{id}]^\text{T} = A \\
\circ & : (\sigma, t) \circ \nu = (\sigma \circ \nu, t [\nu]^\text{T}) t
\end{align*}
\]

\[
\begin{align*}
\text{Tm} & : (\Gamma : \text{Con}) \to \text{Ty} \Gamma \to \text{Type} \\
\_ [\_]^\text{T} & : \text{Tm} \Delta A \to (\sigma : \text{Sub} \Gamma \Delta) \to \text{Tm} \Gamma (A [\sigma]^\text{T}) \\
\text{id}^\text{T} & : t [\text{id}]^\text{T} = t \\
\circ^\text{T} & : t [\sigma \circ \delta]^\text{T} = t [\sigma]^\text{T} [\delta]^\text{T} t \\
\_ \triangleright \_ & : (\Gamma : \text{Con}) \to \text{Ty} \Gamma \to \text{Con} \\
p & : \text{Sub} (\Gamma \triangleright A) \Gamma \\
q & : \text{Tm} (\Gamma \triangleright A) (A [p]^\text{T}) \\
\triangleright \beta_1 & : p \triangleright (\sigma, t) = \sigma \\
\triangleright \beta_2 & : q [\sigma, t]^\text{T} = tt \\
\triangleright \eta & : (p, q) = \text{id}
\end{align*}
\]

(Good definition in a type theory with K/UIP)
CwF definition as a Generalised Algebraic Theory

Con : Type
Sub : Con → Con → Type
_ ◦ _ : Sub Θ Δ → Sub Γ Θ → Sub Γ Δ
assoc : (σ ◦ δ) ◦ ν = σ ◦ (δ ◦ ν)
id : Sub Γ Γ
idlσ : id ◦ σ = σ
idrσ : σ ◦ id = σ

Ty : Con → Type
_ [ _ ]^T : Ty Δ → Sub Γ Δ → Ty Γ

Tm : (Γ : Con) → Ty Γ → Type
_ [ _ ]^T : Tm Δ A → (σ : Sub Γ Δ) → Tm Γ (A[σ]^T)
[id]^T : t[id]^T = t

Good definition in a type theory with K/UIP

See e.g.
- Altenkirch and Kaposi, Type Theory in Type Theory using Quotient Inductive Types, 2016
- Kaposi, Huber, and Sattler, Gluing for Type Theory, 2019

category

terminal

object
CwF definition as a Generalised Algebraic Theory

\[\text{Con} : \text{Type}\]
\[\text{Sub} : \text{Con} \to \text{Con} \to \text{Type}\]
\[\_ \triangleleft \_ : \text{Sub} \Theta \Delta \to \text{Sub} \Gamma \Theta \to \text{Sub} \Gamma \Delta\]
\[\text{assoc} : (\sigma \triangleleft \delta) \triangleleft \nu = \sigma \triangleleft (\delta \triangleleft \nu)\]
\[\text{id} : \text{Sub} \Gamma \Gamma\]
\[\text{idl}_\sigma : \text{id} \triangleleft \sigma = \sigma\]
\[\text{idr}_\sigma : \sigma \triangleleft \text{id} = \sigma\]

\[\_ \triangleright \_ : (\Gamma : \text{Con}) \to \text{Ty} \Gamma \to \text{Con}\]
\[p : \text{Sub} (\Gamma \triangleright A) \Gamma\]
\[q : \text{Tm} (\Gamma \triangleright A) (A[p]^T)\]

\[\_ , \_ : (\sigma : \text{Sub} \Gamma \Delta) \to \text{Tm} \Gamma (A[\sigma]^T) \to \text{Sub} \Gamma (\Delta \triangleright A)\]
\[\triangleright \beta_1 : p \triangleleft (\sigma, t) = \sigma\]
\[\triangleright \beta_2 : q[\sigma, t]^T = tt\]
\[\triangleright \eta : (p, q) = \text{id}\]

\[\_ \triangleright \_ : (\sigma, t) \triangleright \nu = (\sigma \triangleright \nu, t[\nu]^T)t\]

\[\text{Tm} : (\Gamma : \text{Con}) \to \text{Ty} \Gamma \to \text{Type}\]
\[\_ [\_]^T : \text{Tm} \Delta A \to (\sigma : \text{Sub} \Gamma \Delta) \to \text{Tm} \Gamma (A[\sigma]^T)\]
\[\text{[id]}^T : t[\text{id}]^T = t\]
\[\text{[\_]}^T : t[\sigma \triangleleft \delta]^T = t[\sigma]^T[\delta]^T t\]

(Good definition in a type theory with K/UIP)
CwF definition as a Generalised Algebraic Theory

\[ \text{Ty} : \text{Con} \rightarrow \text{Type} \]
\[ _\wedge [\_]^T : \text{Ty} \Delta \rightarrow \text{Sub} \Gamma \Delta \rightarrow \text{Ty} \Gamma \]
\[ [\text{id}]^T : A[\text{id}]^T = A \]
\[ [\text{\wedge}]^T : A[\text{\wedge}]^T = A[\text{\wedge}]^T[\text{\wedge}]^T \]

\[ \text{Tm} : (\Gamma : \text{Con}) \rightarrow \text{Ty} \Gamma \rightarrow \text{Type} \]
\[ _\wedge [\_]^T : \text{Tm} \Delta A \rightarrow (\sigma : \text{Sub} \Gamma \Delta) \rightarrow \text{Tm} \Gamma (A[\sigma]^T) \]
\[ [\text{id}]^T : t[\text{id}]^T = t \quad \text{over } [\text{id}]^T \]
\[ [\text{\wedge}]^T : t[\sigma \text{\wedge} \delta]^T = t[\sigma]^T[\delta]^T t \quad \text{over } [\text{\wedge}]^T \]

\[ _\wedge _ : (\Gamma : \text{Con}) \rightarrow \text{Ty} \Gamma \rightarrow \text{Con} \]
\[ \text{p} : \text{Sub} (\Gamma \wedge A) \Gamma \]
\[ \text{q} : \text{Tm} (\Gamma \wedge A) (A[p]^T) \]
\[ _\wedge , _ : (\sigma : \text{Sub} \Gamma \Delta) \rightarrow \text{Tm} \Gamma (A[\sigma]^T) \rightarrow \text{Sub} \Gamma (\Delta \wedge A) \]
\[ \triangleright \beta_1 : p \wedge (\sigma, t) = \sigma \quad \text{over } [\text{\wedge}]^T \text{ and } \triangleright \beta_1 \]
\[ \triangleright \beta_2 : q[\sigma, t]^T = tt \quad \text{over } [\text{\wedge}]^T \text{ and } \triangleright \beta_1 \]
\[ \triangleright \eta : (p, q) = \text{id} \]
\[ \triangleright \wedge : (\sigma, t) \wedge \nu = (\sigma \wedge \nu, t[\nu]^t) t \quad \text{over } [\text{\wedge}]^T \]

(Good definition in a type theory with K/UIP)

See e.g.:
• Altenkirch and Kaposi, Type Theory in Type Theory using Quotient Inductive Types, 2016
• Kaposi, Huber, and Sattler, Gluing for Type Theory, 2019
CwF definition as a Generalised Algebraic Theory

Con : Type
Sub : Con → Con → Type
_ ⊘ _ : Sub Θ Δ → Sub Γ Θ → Sub Γ Δ
assoc : (σ ⊘ δ) ⊘ ν = σ ⊘ (δ ⊘ ν)
id : Sub Γ Γ
idl : id ⊘ σ = σ
idr : σ ⊘ id = σ

Ty : Con → Type
_[_]T : Ty ∆ → Sub Γ ∆ → Ty Γ

Tm : (Γ : Con) → Ty Γ → Type
[id]T : t[id]T = t
[⊕]T : t[σ ⊘ δ]T = t[σ]T[δ]Tt over [⊕]T

_G_ : (Γ : Con) → Ty Γ → Con
p : Sub (G > A) Γ
q : Tm (G > A) (A[p]T)
_ > _ : (σ : Sub Γ Δ) → Tm Γ (A[σ]T) → Sub Γ (Δ > A)
β1 : p ⊘ (σ, t) = σ
β2 : q[σ, t]T = tt over [⊕]T and β1
η : (p, q) = id
⊕ : (σ, t) ⊘ ν = (σ ⊘ ν, t[ν]T)t over [⊕]T

(Good definition in a type theory with K/UIP)
CwF definition as a Generalised Algebraic Theory

\[\text{Con} : \text{Type} \quad \text{Tm} : (\Gamma : \text{Con}) \rightarrow \text{Ty} \Gamma \rightarrow \text{Type}\]

\[\text{Sub} : \text{Con} \rightarrow \text{Con} \rightarrow \text{Type} \quad \_ \ [\_]^\dagger : \text{Tm} \Delta A \rightarrow (\sigma : \text{Sub} \Gamma \Delta) \rightarrow \text{Tm} \Gamma (A[\sigma]^T)\]

\[\_ \diamond \_ : \text{Sub} \Theta \Delta \rightarrow \text{Sub} \Gamma \Theta \rightarrow \text{Sub} \Gamma \Delta \quad [\text{id}]^t : t[id]^t = t \quad \text{over } [\text{id}]^T\]

\[\text{assoc} : (\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu) \quad [\diamond]^t : t[\sigma \diamond \delta]^t = t[\sigma]^t[\delta]^tt \quad \text{over } [\diamond]^T\]

\[\text{id} : \text{Sub} \Gamma \Delta \quad \text{idl}_\sigma : \text{id} \diamond \sigma \quad \text{idr}_\sigma : \sigma \diamond \text{id} \]

\[\varepsilon : \text{Con} \quad \eta : \text{Sub} \Gamma \Delta \quad \eta^\Delta : \forall (\sigma : \text{Sub} \Gamma \Delta) \quad \varepsilon = \eta\]

\[\text{Ty} : \text{Con} \rightarrow \text{Type} \quad \eta^\Delta : (p, q) = \text{id} \quad \text{over } [\diamond]^T\]

\[\_ [\_]^T : \text{Ty} \Delta \rightarrow \text{Sub} \Gamma \Delta \rightarrow \text{Ty} \Gamma \quad (\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^t)t \quad \text{over } [\diamond]^T\]

\[\text{id}^T : A[id]^T = A \quad [\text{id}]^T : A[\sigma]^T[\delta]^T \quad \text{over } [\diamond]^T\]

\[\text{p} : \text{Sub} \Delta \quad \text{q} : \text{Tm} (A \rightarrow (A \rightarrow \text{Ty})) \quad \text{and } \|\text{p}\text{\|^T} = (\text{id}, A[\text{id}]^T)\]

\[\text{⌘} : (p, q) = \text{id} \quad \text{over } [\diamond]^T\]

\[\text{q}[^\dagger] : q[\sigma, t]^t = tt \quad \text{over } [\diamond]^T\]

(See e.g.:)

- Altenkirch and Kaposi, *Type Theory in Type Theory using Quotient Inductive Types*, 2016
- Kaposi, Huber, and Sattler, *Gluing for Type Theory*, 2019

(Good definition in a type theory with K/UIP)
First example: the syntax / (intended) initial CwF

Possible implementation:

(I) via raw syntax

- possibly ill-typed expressions plus wellformedness predicates

⇒ Initial by the Initiality Theorem
  (Brunerie, de Boer, Lumsdaine, Mörtberg 2019–20).

(II) via a quotient inductive-inductive type (Altenkirch-Kaposi 2016)

- mutually defined inductive families Con, Sub, Ty, Tm
- a constructor for every component of the previous

⇒ Initial by construction.
First example: the syntax / (intended) initial CwF

Possible implementation:

(I) via raw syntax
   • possibly ill-typed expressions plus wellformedness predicates
   ⇒ Initial by the Initiality Theorem
      (Brunerie, de Boer, Lumsdaine, Mörtberg 2019–20).

(II) via a quotient inductive-inductive type (Altenkirch-Kaposi 2016)
   • mutually defined inductive families Con, Sub, Ty, Tm
   • a constructor for every component of the previous
   ⇒ Initial by construction.
First example: the syntax / (intended) initial CwF

Possible implementation:

(I) via raw syntax

• possibly ill-typed expressions plus wellformedness predicates

⇒ Initial by the Initiality Theorem
(Brunerie, de Boer, Lumsdaine, Mörtberg 2019–20).

(II) via a quotient inductive-inductive type (Altenkirch-Kaposi 2016)

• mutually defined inductive families Con, Sub, Ty, Tm
• a constructor for every component of the previous

⇒ Initial by construction.
Second example of a CwF: “Standard Model”, a.k.a. the universe with the obvious structure

- Con is the universe $U$
- Sub $\Gamma \Delta$ is the function type $(\Gamma \rightarrow \Delta)$
- Ty $\Gamma$ is given as $(\Gamma \rightarrow U)$
- Tm $\Gamma A$ is given as $\Pi(x : \Gamma). (Ax)$
- all operations are canonical
- all equations hold judgmentally (assuming enough $\eta$-laws)
Second example of a CwF: \textbf{“Standard Model”}, a.k.a. the universe with the obvious structure

- \textbf{Con} is the universe $\mathcal{U}$
- \textbf{Sub $\Gamma \Delta$} is the function type $(\Gamma \rightarrow \Delta)$
- \textbf{Ty $\Gamma$} is given as $(\Gamma \rightarrow \mathcal{U})$
- \textbf{Tm $\Gamma A$} is given as $\Pi(x : \Gamma).(A x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough $\eta$-laws)
Second example of a CwF: “Standard Model”, a.k.a. the universe with the obvious structure

- \text{Con} is the universe \mathcal{U}
- \text{Sub } \Gamma \Delta \text{ is the function type } (\Gamma \to \Delta)
- \text{Ty } \Gamma \text{ is given as } (\Gamma \to \mathcal{U})
- \text{Tm } \Gamma A \text{ is given as } \Pi(x : \Gamma). (A x)
- all operations are canonical
- all equations hold judgmentally (assuming enough } \eta\text{-laws)
Second example of a CwF: “Standard Model”, a.k.a. the universe with the obvious structure

- \textbf{Con} is the universe \( U \)
- \textbf{Sub} \( \Gamma \Delta \) is the function type \( (\Gamma \to \Delta) \)
- \textbf{Ty} \( \Gamma \) is given as \( (\Gamma \to U) \)
- \textbf{Tm} \( \Gamma A \) is given as \( \Pi(x : \Gamma). (A x) \)
- all operations are canonical
- all equations hold judgmentally (assuming enough \( \eta \)-laws)
Second example of a CwF: "Standard Model", a.k.a. the universe with the obvious structure

- \( \text{Con} \) is the universe \( \mathcal{U} \)
- \( \text{Sub} \Gamma \Delta \) is the function type \( (\Gamma \rightarrow \Delta) \)
- \( \text{Ty} \Gamma \) is given as \( (\Gamma \rightarrow \mathcal{U}) \)
- \( \text{Tm} \Gamma A \) is given as \( \Pi(x : \Gamma). (A x) \)
- all operations are canonical
- all equations hold judgmentally (assuming enough \( \eta \)-laws)
Second example of a CwF: “**Standard Model**”, a.k.a. the universe with the obvious structure

- $\text{Con}$ is the universe $\mathcal{U}$
- $\text{Sub} \Gamma \Delta$ is the function type $(\Gamma \to \Delta)$
- $\text{Ty} \Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\text{Tm} \Gamma A$ is given as $\Pi(x : \Gamma). (A x)$

- all operations are canonical
- all equations hold judgmentally (assuming enough $\eta$-laws)

*e.g. in Agda*
The trouble with(out) UIP

Recall: **UIP** (uniqueness of identity proofs) a.k.a. **Axiom K** says:

\[ \Pi(x \ y : A).\Pi(p \ q : x = y). (p = q) \]

The above definition of a CwF works assuming this axiom!

What if UIP is not assumed (or even inconsistent, e.g. in homotopy type theory)?

Two obvious approaches:

(I) Ignore it: Do everything as before.

or

(II) Make up for it: Assume that Con, Sub, Ty, Tm are families of h-sets.
The trouble with(out) UIP

Recall: **UIP** (*uniqueness of identity proofs*) a.k.a. **Axiom K** says:

$$\Pi(x \, y : A).\Pi(p \, q : x = y).(p = q)$$

The above definition of a CwF works assuming this axiom!

What if UIP is not assumed (or even inconsistent, e.g. in homotopy type theory)?

Two obvious approaches:

(I) Ignore it: Do everything as before.

or

(II) Make up for it: Assume that Con, Sub, Ty, Tm are families of h-sets.
The trouble with(out) UIP

Recall: **UIP** *(uniqueness of identity proofs)* a.k.a. **Axiom K** says:

\[ \Pi(x, y : A).\Pi(p, q : x = y).(p = q) \]

The above definition of a CwF works assuming this axiom!

What if UIP is not assumed (or even inconsistent, e.g. in homotopy type theory)?

Two obvious approaches:

(I) Ignore it: Do everything as before.

or

(II) Make up for it: Assume that Con, Sub, Ty, Tm are families of h-sets.
No UIP: problems of the obvious approaches

(I) Ignore the absence of UIP: Do everything as before.

But then:

\[ \text{idl}_{\sigma} : \text{id} \diamond \sigma = \sigma \]
\[ \text{idr}_{\sigma} : \sigma \diamond \text{id} = \sigma \]

Initial model (w/ base types) does \textbf{not} satisfy \text{idl}_{\text{id}} = \text{idr}_{\text{id}}.

\[ \Rightarrow \] Initial model is \textbf{not} based on h-sets & does \textbf{not} have decidable equality.

\[ \Rightarrow \] The “syntax” (first example) is not initial.

(II) Bake UIP into the definition of CWF: Require Con etc. to be h-sets.

Typical “HoTT solution”.

But: The universe is not an h-set.

\[ \Rightarrow \] The “standard model” (second) fails.
No UIP: problems of the obvious approaches

(I) Ignore the absence of UIP: Do everything as before.

But then: \[ \text{idl}_\sigma : \text{id} \bowtie \sigma = \sigma \]
\[ \text{idr}_\sigma : \sigma \bowtie \text{id} = \sigma \]

Initial model (w/ base types) does not satisfy \[ \text{idl}_{\text{id}} = \text{idr}_{\text{id}}. \]
⇒ Initial model is not based on h-sets & does not have decidable equality.
⇒ The “syntax” (first example) is not initial.

(II) Bake UIP into the definition of CWF: Require Con etc. to be h-sets.

Typical “HoTT solution”.
But: The universe is not an h-set.
⇒ The “standard model” (second) fails.
No UIP: problems of the obvious approaches

(I) Ignore the absence of UIP: Do everything as before.

But then: \[ \text{idl}_\sigma : \text{id} \diamond \sigma = \sigma \]
\[ \text{idr}_\sigma : \sigma \diamond \text{id} = \sigma \]

Initial model (w/ base types) does not satisfy \[ \text{idl}_\text{id} = \text{idr}_\text{id} \].
⇒ Initial model is not based on h-sets & does not have decidable equality.
⇒ The “syntax” (first example) is not initial.

(II) Bake UIP into the definition of CWF: Require Con etc. to be h-sets.

Typical “HoTT solution”.

But: The universe is not an h-set.
⇒ The “standard model” (second) fails.
Why we really want both examples (syntax and standard model)

**Shulman 2014:**
Is the $n^{\text{th}}$ universe a model of HoTT with $(n-1)$ universes?
I.e.: Can we define the syntax and *interpret* it in $U_n$?

Work by: Escardó-Xu, K., Bucholtz, Lumsdaine, Kaposi-Kovaćs, Altenkirch, …

**However:** Even the simplest\(^1\) version of this is still open!
\(^1\) (where the core problem occurs)

The two examples would give us:

| Syntax (raw) | initiality theorem | Syntax (as QIIT) | by initiality | universe $U$ (standard model) |
Why we really want both examples (syntax and standard model)

**Shulman 2014:**
Is the \(n^{\text{th}}\) universe a model of HoTT with \((n-1)\) universes?
I.e.: Can we define the syntax and **interpret** it in \(U_n\)?

Work by: Escardó-Xu, K., Bucholtz, Lumsdaine, Kaposi-Kovačs, Altenkirch, …

**However:** Even the simplest\(^1\) version of this is still open!
\(^1\) (where the core problem occurs)

The two examples would give us:

- Syntax (raw) \(\xleftarrow{\text{initiality theorem}}\) Syntax (as QIIT)
- Syntax (as QIIT) \(\xrightarrow{\text{by initiality}}\) universe \(\mathcal{U}\) (standard model)
Why we really want both examples (syntax and standard model)

Shulman 2014:
Is the $n^{th}$ universe a model of HoTT with (n-1) universes?
I.e.: Can we define the syntax and interpret it in $U_n$?

Work by: Escardó-Xu, K., Bucholtz, Lumsdaine, Kaposi-Kovačs, Altenkirch, …

However: Even the simplest\(^1\) version of this is still open!
\(^1\) (where the core problem occurs)

The two examples would give us:

Syntax (raw) \rightarrow initiality theorem \rightarrow Syntax (as QIIT) \rightarrow by initiality \rightarrow universe $U$ (standard model)
Back to the definition from slide 4:

Con : Type
Sub : Con → Con → Type
_ : Sub Θ Δ → Sub Γ Θ → Sub Γ Δ
assoc : (σ ⊙ δ) ⊗ ν = σ ⊙ (δ ⊗ ν)

Subf : Sub Γ Γ

idlσ : id ⊙ σ = σ
idrσ : σ ⊗ id = σ

• : Con
ε : Sub Γ •
•η : ∀(σ : Sub Γ •). σ = ε

Ty : Con → Type
_ : Ty Δ → Sub Γ Δ → Ty Γ

Tm : (Γ : Con) → Ty Γ → Type

_ [_]t : Tm Δ A → (σ : Sub Γ Δ) → Tm Γ (A[σ]T)
[id]t : t[id]t = t
[σ]t : t[σ ⊗ δ]t = t[σ][δ]tt

idl.idr = idr.idl

p : Sub (Γ > A) Γ
q : Tm (Γ > A) (A[p]T)

▷β1 : p ⊗ (σ, t) = σ
▷β2 : q[σ, t]t = tt
▷η : (p, q) = id

, ⊙ : (σ, t) ⊗ ν = (σ ⊗ ν, t[ν]t)

Goal: Make this coherent! E.g. we really need idl.idr = idr.idl.

Brutal method: Require h-sets everywhere (too restrictive).

Proposed method: Use higher categories \( \Rightarrow (\infty, 1)\)-CwF's.
How?
As discussed above: A 1-CwF consists of
- a category $\mathcal{C}$ of contexts and substitutions
- a presheaf of types
- another functor for terms
- a context extension operation.

We need to $\infty$-categorify everything.
As discussed above: A 1-CwF consists of

- a category $\mathcal{C}$ of contexts and substitutions
- a presheaf of types
- another functor for terms
- a context extension operation.

We need to $\infty$-categorify everything.

**Problem:** Unknown how to define ("half-synthetic") $\infty$-categories in type theory.
As discussed above: A 1-CwF consists of
- a category $C$ of contexts and substitutions
- a presheaf of types
- another functor for terms
- a context extension operation.

We need to $\infty$-categorify everything.

**Problem:** Unknown how to define (“half-synthetic”) $\infty$-categories in type theory.

**Therefore:** Work in extension of type theory, but which?
- 2LTT (*two-level type theory*, successor of Voevodsky’s *HTS*)
- Riehl-Shulman’17 type theory
- Allioux–Finster–Sozeau’21 extension
- …?
As discussed above: A 1-CwF consists of
- a category $\mathcal{C}$ of contexts and substitutions
- a presheaf of types
- another functor for terms
- a context extension operation.

We need to $\infty$-categorify everything.

**Problem:** Unknown how to define (“half-synthetic”) $\infty$-categories in type theory.

**Therefore:** Work in extension of type theory, but which?
- 2LTT (*two-level type theory*, successor of Voevodsky’s *HTS*)
- Riehl-Shulman’17 type theory
- Allioux–Finster–Sozeau’21 extension
- ?
As discussed above: A 1-CwF consists of
- a category $C$ of contexts and substitutions
- a presheaf of types
- another functor for terms
- a context extension operation.

We need to $\infty$-categorify everything.

**Problem:** Unknown how to define (“half-synthetic”) $\infty$-categories in type theory.

**Therefore:** Work in extension of type theory, but which?
- 2LTT (*two-level type theory*, successor of Voevodsky’s *HTS*)
- Riehl-Shulman’17 type theory
- Allioux–Finster–Sozeau’21 extension
- ?
What is an \(\infty\)-category? Model used: Rezk's Segal spaces.

Strategy:

(1) Start with a semisimplicial type ("basic data")

(2) Add Segal condition (\(\Rightarrow\) \(\infty\)-semicategory)

(3) Add identities (\(\Rightarrow\) \(\infty\)-category)
What is an $\infty$-category? Model used: Rezk’s Segal spaces.

Strategy:

1. Start with a semisimplicial type ("basic data")
2. Add Segal condition ($\Rightarrow \infty$-semicategory)
3. Add identities ($\Rightarrow \infty$-category)
What is an $\infty$-category? Model used: Rezk’s Segal spaces.

Strategy:

(1) Start with a semisimplicial type (“basic data”)

(2) Add Segal condition ($\Rightarrow \infty$-semicategory)

(3) Add identities ($\Rightarrow \infty$-category)
What is an $\infty$-category? Model used: Rezk’s Segal spaces.

Strategy:

(1) Start with a *semisimplicial type* ("basic data")

(2) Add Segal condition ($\Rightarrow \infty$-semicategory)

(3) Add identities ($\Rightarrow \infty$-category)
What is an $\infty$-category? Model used: Rezk’s Segal spaces.

Strategy:

(1) Start with a semisimplicial type ("basic data")

(2) Add Segal condition ($\Rightarrow \infty$-semicategory)

(3) Add identities ($\Rightarrow \infty$-category)
(1) Recall: semisimplicial type up to dimension 2 is tuple \((A_0, A_1, A_2)\) where

\[
A_0 : \text{Type} \\
A_1 : A_0 \to A_0 \to \text{Type} \\
A_2 : \Pi\{x y z : A_0\}. (A_1 x y) \to (A_1 y z) \to (A_1 x z) \to \text{Type}
\]
(1) Recall: semisimplicial type up to dimension 2 is tuple \((A_0, A_1, A_2)\) where

\[
\begin{align*}
A_0 : & \text{ Type} \\
A_1 : & A_0 \to A_0 \to \text{ Type} \\
A_2 : & \Pi\{x y z : A_0\}. (A_1 x y) \to (A_1 y z) \to (A_1 x z) \to \text{ Type}
\end{align*}
\]
(1) Recall: semisimplicial type up to dimension 2 is tuple \((A_0, A_1, A_2)\) where

\[A_0\]: Type

\[A_1\]: \(A_0 \rightarrow A_0 \rightarrow\) Type

\[A_2\]: \(\Pi\{x y z : A_0\}. (A_1 x y) \rightarrow (A_1 y z) \rightarrow (A_1 x z) \rightarrow\) Type
(1) Recall: semisimplicial type up to dimension 2 is tuple \((A_0, A_1, A_2)\) where

\[
\begin{align*}
A_0 &: \text{Type} \\
A_1 &: A_0 \to A_0 \to \text{Type} \\
A_2 &: \prod \{x y z : A_0\}. (A_1 x y) \to (A_1 y z) \to (A_1 x z) \to \text{Type}
\end{align*}
\]
(2) Adding the Segal condition

**Semicategory (beginning)**

Ob : Type  
Hom : Ob → Ob → Type  
_ ◦ _ : \{x y z : Ob\} → (Hom y z)  
   → (Hom x y) → (Hom x z)

**Semisimplicial type (beginning)**

A_0 : Type  
A_1 : A_0 → A_0 → Type  
A_2 : \{x y z : A_0\} → (A_1 y z)  
   → (A_1 x y) → (A_1 x z) → Type

Lemma: For \(X : \text{Type}\), we have \(X \simeq \Sigma(P : X → \text{Type}).\)
   \(\text{isContr}(\Sigma(x : X).P x)\).
(2) Adding the Segal condition

Semicategory (beginning)  Semisimplicial type (beginning)

\[
\begin{align*}
\text{Ob} : & \quad \text{Type} \\
\text{Hom} : & \quad \text{Ob} \to \text{Ob} \to \text{Type} \\
_\circ_ : & \quad \{x y z : \text{Ob}\} \to (\text{Hom} y z) \\
& \quad \to (\text{Hom} x y) \to (\text{Hom} x z)
\end{align*}
\]

\[
\begin{align*}
A_0 : & \quad \text{Type} \\
A_1 : & \quad A_0 \to A_0 \to \text{Type} \\
A_2 : & \quad \{x y z : A_0\} \to (A_1 y z) \\
& \quad \to (A_1 x y) \to (A_1 x z) \to \text{Type}
\end{align*}
\]

Lemma: For \(X : \text{Type}\), we have \(X \simeq \Sigma(P : X \to \text{Type}).\) 
\(\text{isContr}(\Sigma(x : X).P x).\)
(2) Adding the Segal condition

Semicategory (beginning)  Semisimplicial type (beginning)

\[
\begin{align*}
\text{Ob} & : \text{Type} \\
\text{Hom} & : \text{Ob} \to \text{Ob} \to \text{Type} \\
\_ \circ \_ & : \{x \, y \, z : \text{Ob}\} \to (\text{Hom} \, y \, z) \to (\text{Hom} \, x \, y) \to (\text{Hom} \, x \, z)
\end{align*}
\]

\[
\begin{align*}
\text{A}_0 & : \text{Type} \\
\text{A}_1 & : \text{A}_0 \to \text{A}_0 \to \text{Type} \\
\text{A}_2 & : \{x \, y \, z : \text{A}_0\} \to (\text{A}_1 \, y \, z) \to (\text{A}_1 \, x \, y) \to (\text{A}_1 \, x \, z) \to \text{Type}
\end{align*}
\]

Lemma: For \( X : \text{Type} \), we have \( X \simeq \Sigma(P : X \to \text{Type}).\) isContr\((\Sigma(x : X).P \, x)\).
(2) Adding the Segal condition

Semicategory (beginning)  Semisimplicial type (beginning)

\[\text{Ob} : \text{Type} \quad \text{Hom} : \text{Ob} \to \text{Ob} \to \text{Type} \quad A_0 : \text{Type} \quad A_1 : A_0 \to A_0 \to \text{Type}\]

\[\_ \circ \_ : \{x y z : \text{Ob}\} \to (\text{Hom} y z) \quad A_2 : \{x y z : A_0\} \to (A_1 y z) \quad \to (\text{Hom} x y) \to (\text{Hom} x z) \quad \to (A_1 x y) \to (A_1 x z) \to \text{Type}\]

Lemma: For \(X : \text{Type}\), we have \(X \simeq \Sigma(P : X \to \text{Type}).\)

\[\text{isContr}(\Sigma(x : X).P x).\]
(2) Adding the Segal condition

Semicategory (beginning)

\[ \text{Ob} : \text{Type} \quad \text{Hom} : \text{Ob} \to \text{Ob} \to \text{Type} \]

Semisimplicial type (beginning)

\[ A_0 : \text{Type} \]
\[ A_1 : A_0 \to A_0 \to \text{Type} \]
\[ A_2 : \{x y z : \text{Ob}\} \to (\text{Hom} y z) \]
\[ \to (\text{Hom} x y) \to (\text{Hom} x z) \]

Lemma: For \( X : \text{Type} \), we have \( X \) is equivalent to
\[ \Sigma (P : X \to \text{Type}) \]
\[ \text{isContr}(\Sigma(x : X).P x). \]
(3) Add identities/degeneracies

In previous work: *Completeness* (Lurie/Harpaz/Capriotti) corresponding to univalent identities (cf. Capriotti-Kraus 2018).

Here: We don’t want built-in univalence. Instead:

Def: A line $f : A_1 x x$ is a *good identity* if it is an *idempotent equivalence*.

Def: $f$ is *idempotent* if $A_2 f f f$.

Def: $f$ is an *equivalence* if pre- and post-composition with $f$ is.
(3) Add identities/degeneracies

In previous work: Completeness (Lurie/Harpaz/Capriotti) corresponding to univalent identities (cf. Capriotti-Kraus 2018).

Here: We don’t want built-in univalence. Instead:

Def: A line \( f : A_1 x x \) is a **good identity** if it is an **idempotent equivalence**.

Def: \( f \) is **idempotent** if \( A_2 f f f \).

Def: \( f \) is an **equivalence** if pre- and post-composition with \( f \) is.
(3) Add identities/degeneracies

In previous work: Completeness (Lurie/Harpaz/Capriotti) corresponding to univalent identities (cf. Capriotti-Kraus 2018).

Here: We don’t want built-in univalence. Instead:

**Def:** A line $f : A_1 x x$ is a **good identity** if it is an **idempotent equivalence**.

**Def:** $f$ is **idempotent** if $A_2 f f f$.

**Def:** $f$ is an **equivalence** if pre- and post-composition with $f$ is.
Definition: A semicategory (higher semicategory, semi-Segal type) has a **good identity structure** if every object (point) is equipped with an *idempotent equivalence*.

Theorem: “Having a good identity structure”:
- is a propositional property; and
- generates all degeneracies; and
- is interderivable with a “standard” identity structure (id with idl and idr).

Definition: An **∞-category** is a semisimplicial type which satisfies the Segal condition and has a good identity structure.

(Extending ∞-categories to ∞-CwF’s is not done in this talk.)
Definition: A semicategory (higher semicategory, semi-Segal type) has a **good identity structure** if every object (point) is equipped with an *idempotent equivalence*.

Theorem: “Having a good identity structure”: 
- is a propositional property; and 
- generates all degeneracies; and 
- is interderivable with a “standard” identity structure (id with idl and idr).

Definition: An $\infty$-category is a semisimplicial type which satisfies the Segal condition and has a good identity structure.

(Extending $\infty$-categories to $\infty$-CwF’s is not done in this talk.)
Definition: A semicategory (higher semicategory, semi-Segal type) has a **good identity structure** if every object (point) is equipped with an *idempotent equivalence*.

Theorem: “Having a good identity structure”: 
- is a propositional property; and 
- generates all degeneracies; and 
- is interderivable with a “standard” identity structure (id with idl and idr).

Definition: An ∞-*category* is a semisimplicial type which satisfies the Segal condition and has a good identity structure.

(Extending ∞-categories to ∞-CwF’s is not done in this talk.)
RESULTS
Done:

- Every h-set-based 1-CwF is an $\infty$-CwF
  \[ \Rightarrow \text{the syntax is an } \infty\text{-CwF} \]
- Every “wild” 1-CwF, where equations hold strictly, is an $\infty$-CwF
  \[ \Rightarrow \text{standard model (universe) is an } \infty\text{-CwF} \]
- Other constructions, e.g. slice $\infty$-CwF ("working with assumptions")

Main unsolved problem:

- Is the syntax initial?

And: How about other settings (not 2LTT)?
Done:

- Every h-set-based 1-CwF is an $\infty$-CwF
  $\Rightarrow$ the syntax is an $\infty$-CwF
- Every “wild” 1-CwF, where equations hold strictly, is an $\infty$-CwF
  $\Rightarrow$ standard model (universe) is an $\infty$-CwF
- Other constructions, e.g. slice $\infty$-CwF (“working with assumptions”)

Main unsolved problem:

- Is the syntax initial?

And: How about other settings (not 2LTT)?
Done:

- Every h-set-based 1-CwF is an $\infty$-CwF
  $\Rightarrow$ the syntax is an $\infty$-CwF
- Every “wild” 1-CwF, where equations hold strictly, is an $\infty$-CwF
  $\Rightarrow$ standard model (universe) is an $\infty$-CwF
- Other constructions, e.g. slice $\infty$-CwF (“working with assumptions”)

Main unsolved problem:

- Is the syntax initial?

And: How about other settings (not 2LTT)?
Done:

- Every h-set-based 1-CwF is an $\infty$-CwF
  - $\Rightarrow$ the syntax is an $\infty$-CwF
- Every “wild” 1-CwF, where equations hold strictly, is an $\infty$-CwF
  - $\Rightarrow$ standard model (universe) is an $\infty$-CwF
- Other constructions, e.g. slice $\infty$-CwF (“working with assumptions”)

Main unsolved problem:

- Is the syntax initial?

And: How about other settings (not 2LTT)?
THANKS!