Internal ∞-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 $\sim$ "syntax")

Goal: Define what a model of type theory is – in type theory!
(in particular: intended initial model $\sim$ "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
    _▸_ : (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

```
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

$$
\begin{array}{lcl}
\mathsf{Con} & : & \mathsf{Type} \\
\mathsf{Sub} & : & \mathsf{Con} \to \mathsf{Con} \to \mathsf{Type} \\
\_ \diamond \_ & : & \mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta \\
\mathsf{assoc} & : & (\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu) \\
\mathsf{id} & : & \mathsf{Sub}\,\Gamma\,\Gamma \\
\mathsf{idl}_\sigma & : & \mathsf{id} \diamond \sigma = \sigma \\
\mathsf{idr}_\sigma & : & \sigma \diamond \mathsf{id} = \sigma \\
\bullet & : & \mathsf{Con} \\
\epsilon & : & \mathsf{Sub}\,\Gamma\,\bullet \\
\bullet\eta & : & \forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon \\
\mathsf{Ty} & : & \mathsf{Con} \to \mathsf{Type} \\
\_[\_]^{\mathsf{T}} & : & \mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma \\
[\mathsf{id}]^{\mathsf{T}} & : & A[\mathsf{id}]^{\mathsf{T}} = A \\
[\diamond]^{\mathsf{T}} & : & A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}
\end{array}
$$

$$
\begin{array}{lcll}
\mathsf{Tm} & : & (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type} & \\
\_[\_]^{\mathsf{t}} & : & \mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) & \\
[\mathsf{id}]^{\mathsf{t}} & : & t[\mathsf{id}]^{\mathsf{t}} = t & \text{over } [\mathsf{id}]^{\mathsf{T}} \\
[\diamond]^{\mathsf{t}} & : & t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}} t & \text{over } [\diamond]^{\mathsf{T}} \\
\_ \triangleright \_ & : & (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con} & \\
\mathsf{p} & : & \mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma & \\
\mathsf{q} & : & \mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}]^{\mathsf{T}}) & \\
\_\,,\,\_ & : & (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) \to \mathsf{Sub}\,\Gamma\,(\Delta \triangleright A) & \\
\triangleright\beta_1 & : & \mathsf{p} \diamond (\sigma, t) = \sigma & \\
\triangleright\beta_2 & : & \mathsf{q}[\sigma, t]^{\mathsf{t}} = tt & \text{over } [\diamond]^{\mathsf{T}} \text{ and } \triangleright\beta_1 \\
\triangleright\eta & : & (\mathsf{p}, \mathsf{q}) = \mathsf{id} & \\
,\diamond & : & (\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}})t & \text{over } [\diamond]^{\mathsf{T}}
\end{array}
$$

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

CwF definition as a Generalised Algebraic Theory

Con	:	Type	
Sub	:	$\mathsf{Con} \to \mathsf{Con} \to \mathsf{Type}$	
$\_ \diamond \_$	:	$\mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$	
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$	
id	:	$\mathsf{Sub}\,\Gamma\,\Gamma$	category
idl_σ	:	$\mathsf{id} \diamond \sigma = \sigma$	
idr_σ	:	$\sigma \diamond \mathsf{id} = \sigma$	
$\bullet$	:	Con	
ϵ	:	$\mathsf{Sub}\,\Gamma\,\bullet$	
$\bullet\eta$	:	$\forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$	
Ty	:	$\mathsf{Con} \to \mathsf{Type}$	
$\_[\_]^{\mathsf{T}}$	:	$\mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$	
$[\mathsf{id}]^{\mathsf{T}}$	:	$A[\mathsf{id}]^{\mathsf{T}} = A$	
$[\diamond]^{\mathsf{T}}$	:	$A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}$	

Tm	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$	
$\_[\_]^{\mathsf{t}}$	:	$\mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}})$	
$[\mathsf{id}]^{\mathsf{t}}$	:	$t[\mathsf{id}]^{\mathsf{t}} = t$	over $[\mathsf{id}]^{\mathsf{T}}$
$[\diamond]^{\mathsf{t}}$	:	$t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}} t$	over $[\diamond]^{\mathsf{T}}$
$\_ \triangleright \_$	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}$	
p	:	$\mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma$	
q	:	$\mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}]^{\mathsf{T}})$	
$\_ , \_$	:	$(\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) \to \mathsf{Sub}\,\Gamma\,(\Delta \triangleright A)$	
$\triangleright\beta_1$	:	$\mathsf{p} \diamond (\sigma, t) = \sigma$	
$\triangleright\beta_2$	:	$\mathsf{q}[\sigma, t]^{\mathsf{t}} = tt$	over $[\diamond]^{\mathsf{T}}$ and $\triangleright\beta_1$
$\triangleright\eta$	:	$(\mathsf{p}, \mathsf{q}) = \mathsf{id}$	
$, \diamond$	:	$(\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}})t$	over $[\diamond]^{\mathsf{T}}$

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

CwF definition as a Generalised Algebraic Theory

Con	:	Type
Sub	:	Con $\to$ Con $\to$ Type
$\_ \diamond \_$	:	$\mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$
id	:	$\mathsf{Sub}\,\Gamma\,\Gamma$
idl_σ	:	$\mathsf{id} \diamond \sigma = \sigma$
idr_σ	:	$\sigma \diamond \mathsf{id} = \sigma$

category

$\bullet$	:	Con
ϵ	:	$\mathsf{Sub}\,\Gamma\,\bullet$
$\bullet\eta$	:	$\forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$

terminal object

Ty	:	Con $\to$ Type
$\_[\_]^{\mathsf{T}}$	:	$\mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$
$[\mathsf{id}]^{\mathsf{T}}$	:	$A[\mathsf{id}]^{\mathsf{T}} = A$
$[\diamond]^{\mathsf{T}}$	:	$A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}$

Tm	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$	
$\_[\_]^{\mathsf{t}}$	:	$\mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}})$	
$[\mathsf{id}]^{\mathsf{t}}$	:	$t[\mathsf{id}]^{\mathsf{t}} = t$	over $[\mathsf{id}]^{\mathsf{T}}$
$[\diamond]^{\mathsf{t}}$	:	$t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}}t$	over $[\diamond]^{\mathsf{T}}$
$\_ \triangleright \_$	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}$	
p	:	$\mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma$	
q	:	$\mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}]^{\mathsf{T}})$	
$\_ , \_$	:	$(\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) \to \mathsf{Sub}\,\Gamma\,(\Delta \triangleright A)$	
$\triangleright\beta_1$	:	$\mathsf{p} \diamond (\sigma, t) = \sigma$	
$\triangleright\beta_2$	:	$\mathsf{q}[\sigma, t]^{\mathsf{t}} = tt$	over $[\diamond]^{\mathsf{T}}$ and $\triangleright\beta_1$
$\triangleright\eta$	:	$(\mathsf{p}, \mathsf{q}) = \mathsf{id}$	
$,\diamond$	:	$(\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}})t$	over $[\diamond]^{\mathsf{T}}$

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

CwF definition as a Generalised Algebraic Theory

Con	:	Type
Sub	:	Con → Con → Type
$\_ \diamond \_$	:	$\mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$
id	:	$\mathsf{Sub}\,\Gamma\,\Gamma$
idl_σ	:	$\mathsf{id} \diamond \sigma = \sigma$
idr_σ	:	$\sigma \diamond \mathsf{id} = \sigma$

category

$\bullet$	:	Con
ϵ	:	$\mathsf{Sub}\,\Gamma\,\bullet$
$\bullet\eta$	:	$\forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$

terminal object

Ty	:	Con → Type
$\_[\_]^{\mathsf{T}}$	:	$\mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$
$[\mathsf{id}]^{\mathsf{T}}$	:	$A[\mathsf{id}]^{\mathsf{T}} = A$
$[\diamond]^{\mathsf{T}}$	:	$A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}$

presheaf

Tm	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$	
$\_[\_]^{\mathsf{t}}$	:	$\mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}})$	
$[\mathsf{id}]^{\mathsf{t}}$	:	$t[\mathsf{id}]^{\mathsf{t}} = t$	over $[\mathsf{id}]^{\mathsf{T}}$
$[\diamond]^{\mathsf{t}}$	:	$t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}} t$	over $[\diamond]^{\mathsf{T}}$
$\_ \triangleright \_$	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}$	
p	:	$\mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma$	
q	:	$\mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}]^{\mathsf{T}})$	
$\_ , \_$	:	$(\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) \to \mathsf{Sub}\,\Gamma\,(\Delta \triangleright A)$	
$\triangleright\beta_1$	:	$\mathsf{p} \diamond (\sigma, t) = \sigma$	
$\triangleright\beta_2$	:	$\mathsf{q}[\sigma, t]^{\mathsf{t}} = tt$	over $[\diamond]^{\mathsf{T}}$ and $\triangleright\beta_1$
$\triangleright\eta$	:	$(\mathsf{p}, \mathsf{q}) = \mathsf{id}$	
$, \diamond$	:	$(\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}})t$	over $[\diamond]^{\mathsf{T}}$

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

CwF definition as a Generalised Algebraic Theory

$\mathsf{Con} : \mathsf{Type}$

$\mathsf{Sub} : \mathsf{Con} \to \mathsf{Con} \to \mathsf{Type}$

$\_\diamond\_ : \mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$

$\mathsf{assoc} : (\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$

$\mathsf{id} : \mathsf{Sub}\,\Gamma\,\Gamma$

$\mathsf{idl}_\sigma : \mathsf{id} \diamond \sigma = \sigma$

$\mathsf{idr}_\sigma : \sigma \diamond \mathsf{id} = \sigma$

category

$\bullet : \mathsf{Con}$

$\epsilon : \mathsf{Sub}\,\Gamma\,\bullet$

$\bullet\eta : \forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$

terminal object

$\mathsf{Ty} : \mathsf{Con} \to \mathsf{Type}$

$\_[\_]^{\mathsf{T}} : \mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$

$[\mathsf{id}]^{\mathsf{T}} : A[\mathsf{id}]^{\mathsf{T}} = A$

$[\diamond]^{\mathsf{T}} : A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}$

presheaf

$\mathsf{Tm} : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$

$\_[\_]^{\mathsf{t}} : \mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}})$

$[\mathsf{id}]^{\mathsf{t}} : t[\mathsf{id}]^{\mathsf{t}} = t$ over $[\mathsf{id}]^{\mathsf{T}}$

$[\diamond]^{\mathsf{t}} : t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}}t$ over $[\diamond]^{\mathsf{T}}$

another functor

$\_\triangleright\_ : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}$

$\mathsf{p} : \mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma$

$\mathsf{q} : \mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}]^{\mathsf{T}})$

$\_,\_ : (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) \to \mathsf{Sub}\,\Gamma\,(\Delta \triangleright A)$

$\triangleright\beta_1 : \mathsf{p} \diamond (\sigma, t) = \sigma$

$\triangleright\beta_2 : \mathsf{q}[\sigma, t]^{\mathsf{t}} = tt$ over $[\diamond]^{\mathsf{T}}$ and $\triangleright\beta_1$

$\triangleright\eta : (\mathsf{p}, \mathsf{q}) = \mathsf{id}$

$,\diamond : (\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}})t$ over $[\diamond]^{\mathsf{T}}$

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

CwF definition as a Generalised Algebraic Theory

category

Con	:	Type
Sub	:	Con $\to$ Con $\to$ Type
$\_\diamond\_$	:	$\mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$
id	:	$\mathsf{Sub}\,\Gamma\,\Gamma$
idl_σ	:	$\mathsf{id} \diamond \sigma = \sigma$
idr_σ	:	$\sigma \diamond \mathsf{id} = \sigma$

terminal object

$\bullet$	:	Con
ϵ	:	$\mathsf{Sub}\,\Gamma\,\bullet$
$\bullet\eta$	:	$\forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$

presheaf

Ty	:	Con $\to$ Type
$\_[\_]^\mathsf{T}$	:	$\mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$
$[\mathsf{id}]^\mathsf{T}$	:	$A[\mathsf{id}]^\mathsf{T} = A$
$[\diamond]^\mathsf{T}$	:	$A[\sigma \diamond \delta]^\mathsf{T} = A[\sigma]^\mathsf{T}[\delta]^\mathsf{T}$

another functor

Tm	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$	
$\_[\_]^\mathsf{t}$	:	$\mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^\mathsf{T})$	
$[\mathsf{id}]^\mathsf{t}$	:	$t[\mathsf{id}]^\mathsf{t} = t$	over $[\mathsf{id}]^\mathsf{T}$
$[\diamond]^\mathsf{t}$	:	$t[\sigma \diamond \delta]^\mathsf{t} = t[\sigma]^\mathsf{t}[\delta]^\mathsf{t}t$	over $[\diamond]^\mathsf{T}$

context extension

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

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

CwF definition as a Generalised Algebraic Theory

Con	:	Type
Sub	:	Con → Con → Type
$\_ \diamond \_$	:	$\mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$
id	:	$\mathsf{Sub}\,\Gamma\,\Gamma$
idl_σ	:	$\mathsf{id} \diamond \sigma$
idr_σ	:	$\sigma \diamond \mathsf{id}$
$\bullet$	:	Con
ϵ	:	$\mathsf{Sub}\,\Gamma$
$\bullet\eta$	:	$\forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$
Ty	:	Con → Type
$\_[\_]^{\mathsf{T}}$	:	$\mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$
$[\mathsf{id}]^{\mathsf{T}}$	:	$A[\mathsf{id}]^{\mathsf{T}} = A$
$[\diamond]^{\mathsf{T}}$	:	$A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}$

Tm	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$	
$\_[\_]^{\mathsf{t}}$	:	$\mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}})$	
$[\mathsf{id}]^{\mathsf{t}}$	:	$t[\mathsf{id}]^{\mathsf{t}} = t$	over $[\mathsf{id}]^{\mathsf{T}}$
$[\diamond]^{\mathsf{t}}$	:	$t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}} t$	over $[\diamond]^{\mathsf{T}}$
		$\Gamma\,(\Delta \triangleright A)$	
$\triangleright\beta_2$	:	$\mathsf{q}[\sigma, t]^{\mathsf{t}} = tt$	over $[\diamond]^{\mathsf{T}}$ and $\triangleright\beta_1$
$\triangleright\eta$	:	$(\mathsf{p}, \mathsf{q}) = \mathsf{id}$	
$,\diamond$	:	$(\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}})t$	over $[\diamond]^{\mathsf{T}}$

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

$\Rightarrow$ Initial by the **Initiality Theorem** (Brunerie, de Boer, Lumsdaine, Mörtberg 2019–20).

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

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

$\Rightarrow$ Initial by construction.

First example: the **syntax** / (intended) **initial CwF**

Possible implementation:

(I) via **raw syntax**

- possibly ill-typed expressions plus wellformedness predicates

$\Rightarrow$ Initial by the **Initiality Theorem**
(Brunerie, de Boer, Lumsdaine, Mörtberg 2019–20).

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

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

$\Rightarrow$ Initial by construction.

First example: the **syntax** / (intended) **initial CwF**

Possible implementation:

(I) via **raw syntax**

- possibly ill-typed expressions plus wellformedness predicates

$\Rightarrow$ Initial by the **Initiality Theorem** (Brunerie, de Boer, Lumsdaine, Mörtberg 2019–20).

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

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

$\Rightarrow$ Initial by construction.

Second example of a CwF: **"Standard Model"**, a.k.a.
the universe with the obvious structure

- Con is the universe $\mathcal{U}$
- $\mathsf{Sub}\,\Gamma\,\Delta$ is the function type $(\Gamma \to \Delta)$
- $\mathsf{Ty}\,\Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\mathsf{Tm}\,\Gamma\,A$ is given as $\Pi(x:\Gamma).(A\,x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough η-laws)

Second example of a CwF: **"Standard Model"**, a.k.a.
the universe with the obvious structure

- Con is the universe $\mathcal{U}$
- $\mathsf{Sub}\,\Gamma\,\Delta$ is the function type $(\Gamma \to \Delta)$
- $\mathsf{Ty}\,\Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\mathsf{Tm}\,\Gamma\,A$ is given as $\Pi(x : \Gamma).(A\,x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough η-laws)

Second example of a CwF: **"Standard Model"**, a.k.a.
the universe with the obvious structure

- Con is the universe $\mathcal{U}$
- $\mathsf{Sub}\,\Gamma\,\Delta$ is the function type $(\Gamma \to \Delta)$
- $\mathsf{Ty}\,\Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\mathsf{Tm}\,\Gamma\,A$ is given as $\Pi(x : \Gamma).(A\,x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough η-laws)

Second example of a CwF: **"Standard Model"**, a.k.a.
the universe with the obvious structure

- Con is the universe $\mathcal{U}$
- $\mathsf{Sub}\,\Gamma\,\Delta$ is the function type $(\Gamma \to \Delta)$
- $\mathsf{Ty}\,\Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\mathsf{Tm}\,\Gamma\,A$ is given as $\Pi(x : \Gamma).(A\,x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough η-laws)

Second example of a CwF: **"Standard Model"**, a.k.a.
the universe with the obvious structure

- Con is the universe $\mathcal{U}$
- $\mathsf{Sub}\,\Gamma\,\Delta$ is the function type $(\Gamma \to \Delta)$
- $\mathsf{Ty}\,\Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\mathsf{Tm}\,\Gamma\,A$ is given as $\Pi(x : \Gamma).(A\,x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough η-laws)

Second example of a CwF: **"Standard Model"**, a.k.a.
the universe with the obvious structure

- Con is the universe $\mathcal{U}$
- $\mathsf{Sub}\,\Gamma\,\Delta$ is the function type $(\Gamma \to \Delta)$
- $\mathsf{Ty}\,\Gamma$ is given as $(\Gamma \to \mathcal{U})$
- $\mathsf{Tm}\,\Gamma\,A$ is given as $\Pi(x : \Gamma).(A\,x)$
- all operations are canonical
- all equations hold judgmentally (assuming enough η-laws)

e.g. in Agda

The trouble with(out) UIP

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

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

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:

$$\mathbf{\Pi(x\;y : A).\Pi(p\;q : x = y).(p = q)}$$

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:

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

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:

$$\mathsf{idl}_\sigma : \mathsf{id} \diamond \sigma = \sigma$$
$$\mathsf{idr}_\sigma : \sigma \diamond \mathsf{id} = \sigma$$

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

$\Rightarrow$ Initial model is **not** based on h-sets & does **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: $\mathsf{idl}_\sigma : \mathsf{id} \diamond \sigma = \sigma$
$\mathsf{idr}_\sigma : \sigma \diamond \mathsf{id} = \sigma$

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

$\Rightarrow$ Initial model is **not** based on h-sets & does **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: $\mathsf{idl}_\sigma : \mathsf{id} \diamond \sigma = \sigma$

$\mathsf{idr}_\sigma : \sigma \diamond \mathsf{id} = \sigma$

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

$\Rightarrow$ Initial model is **not** based on h-sets & does **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.

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 $\mathcal{U}_n$?

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 $\mathcal{U}$ (standard model)

Why we really want both examples (syntax and standard model)

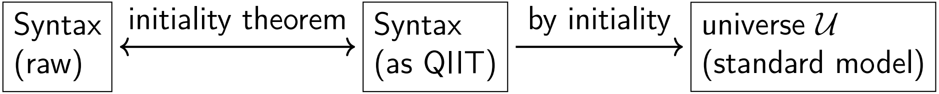
Shulman 2014:
Is the n^{th} universe a model of HoTT with (n-1) universes?

I.e.: Can we define the syntax and *interpret* it in $\mathcal{U}_n$?

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:



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 $\mathcal{U}_n$?

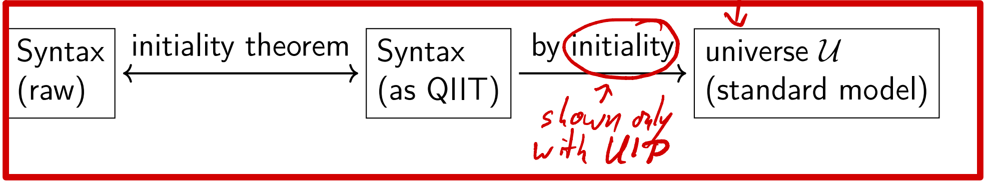
Work by: Escardó-Xu, K., Bucholtz, Lumsdaine, Kaposi-Kovaćs, Altenkirch, . . .

However: Even the simplest[1] version of this is still open!

[1] (where the core problem occurs)

doesn't work with UIP

The two examples would give us:



Back to the definition from slide 4:

Con	:	Type
Sub	:	$\mathsf{Con} \to \mathsf{Con} \to \mathsf{Type}$
$\_ \diamond \_$	:	$\mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Gamma\,\Theta \to \mathsf{Sub}\,\Gamma\,\Delta$
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$
id	:	$\mathsf{Sub}\,\Gamma\,\Gamma$
idl_σ	:	$\mathsf{id} \diamond \sigma = \sigma$
idr_σ	:	$\sigma \diamond \mathsf{id} = \sigma$
$\bullet$	:	Con
ϵ	:	$\mathsf{Sub}\,\Gamma\,\bullet$
$\bullet\eta$	:	$\forall(\sigma : \mathsf{Sub}\,\Gamma\,\bullet).\ \sigma = \epsilon$
Ty	:	$\mathsf{Con} \to \mathsf{Type}$
$\_[\_]^{\mathsf{T}}$	:	$\mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma$
$[\mathsf{id}]^{\mathsf{T}}$	:	$A[\mathsf{id}]^{\mathsf{T}} = A$
$[\diamond]^{\mathsf{T}}$	:	$A[\sigma \diamond \delta]^{\mathsf{T}} = A[\sigma]^{\mathsf{T}}[\delta]^{\mathsf{T}}$

$\mathsf{idl}_{\mathsf{id}} \overset{?}{=} \mathsf{idr}_{\mathsf{id}}$

Tm	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$	
$\_[\_]^{\mathsf{t}}$	:	$\mathsf{Tm}\,\Delta\,A \to (\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}})$	
$[\mathsf{id}]^{\mathsf{t}}$	:	$t[\mathsf{id}]^{\mathsf{t}} = t$	over $[\mathsf{id}]^{\mathsf{T}}$
$[\diamond]^{\mathsf{t}}$	:	$t[\sigma \diamond \delta]^{\mathsf{t}} = t[\sigma]^{\mathsf{t}}[\delta]^{\mathsf{t}} t$	over $[\diamond]^{\mathsf{T}}$
$\_ \rhd \_$	:	$(\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}$	
p	:	$\mathsf{Sub}\,(\Gamma \rhd A)\,\Gamma$	
q	:	$\mathsf{Tm}\,(\Gamma \rhd A)\,(A[\mathsf{p}]^{\mathsf{T}})$	
$\_ , \_$	:	$(\sigma : \mathsf{Sub}\,\Gamma\,\Delta) \to \mathsf{Tm}\,\Gamma\,(A[\sigma]^{\mathsf{T}}) \to \mathsf{Sub}\,\Gamma\,(\Delta \rhd A)$	
$\rhd\beta_1$	:	$\mathsf{p} \diamond (\sigma, t) = \sigma$	
$\rhd\beta_2$	:	$\mathsf{q}[\sigma, t]^{\mathsf{t}} = tt$	over $[\diamond]^{\mathsf{T}}$ and $\rhd\beta_1$
$\rhd\eta$	:	$(\mathsf{p}, \mathsf{q}) = \mathsf{id}$	
$,\diamond$	:	$(\sigma, t) \diamond \nu = (\sigma \diamond \nu, t[\nu]^{\mathsf{t}}) t$	over $[\diamond]^{\mathsf{T}}$

Goal: Make this coherent! E.g. we really need $\mathsf{idl}_{\mathsf{id}} = \mathsf{idr}_{\mathsf{id}}$.

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

Proposed method: Use higher categories $\Longrightarrow$ $(\infty, 1)$-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 ∞-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 ∞-categorify everything.

Problem: Unknown how to define (“half-synthetic”) ∞-categories in type theory.

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 ∞-categorify everything.

Problem: Unknown how to define ("half-synthetic") ∞-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 ∞-categorify everything.

Problem: Unknown how to define ("half-synthetic") ∞-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 ∞-categorify everything.

Problem: Unknown how to define ("half-synthetic") ∞-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 ∞-category? Model used: Rezk's Segal spaces.

Strategy:

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

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

(3) Add identities ($\Rightarrow$ ∞-category)

What is an ∞-category? Model used: Rezk's Segal spaces.

Strategy:

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

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

(3) Add identities ($\Rightarrow$ ∞-category)

What is an ∞-category? Model used: Rezk's Segal spaces.

Strategy:

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

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

(3) Add identities ($\Rightarrow$ ∞-category)

What is an ∞-category? Model used: Rezk's Segal spaces.

Strategy:

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

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

(3) Add identities ($\Rightarrow$ ∞-category)

What is an ∞-category? Model used: Rezk's Segal spaces.

Strategy:

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

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

(3) Add identities ($\Rightarrow$ ∞-category)

propositional properties!

(1) Recall: semisimplicial type up to dimension 2 is tuple (A_0, A_1, A_2) where

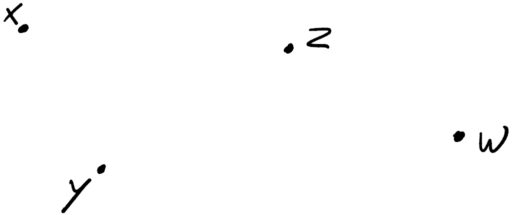
$$A_0 : \mathsf{Type}$$

$$A_1 : A_0 \to A_0 \to \mathsf{Type}$$

$$A_2 : \Pi\{x\,y\,z : A_0\}.\,(A_1\,x\,y) \to (A_1\,y\,z) \to (A_1\,x\,z) \to \mathsf{Type}$$

(1) Recall: semisimplicial type up to dimension 2 is tuple (A_0, A_1, A_2) where

$$A_0 : \mathsf{Type}$$
$$A_1 : A_0 \to A_0 \to \mathsf{Type}$$
$$A_2 : \Pi\{x\, y\, z : A_0\}.\, (A_1\, x\, y) \to (A_1\, y\, z) \to (A_1\, x\, z) \to \mathsf{Type}$$

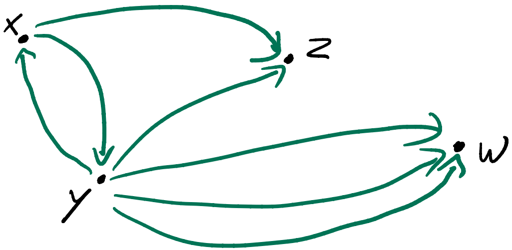


(1) Recall: semisimplicial type up to dimension 2 is tuple (A_0, A_1, A_2) where

$$A_0 : \text{Type}$$

$$A_1 : A_0 \to A_0 \to \text{Type}$$

$$A_2 : \Pi\{x\, y\, z : A_0\}.\,(A_1\, x\, y) \to (A_1\, y\, z) \to (A_1\, x\, z) \to \text{Type}$$

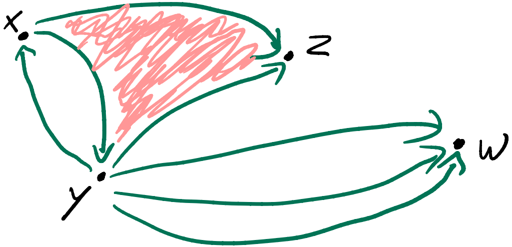


(1) Recall: semisimplicial type up to dimension 2 is tuple (A_0, A_1, A_2) where

A_0 : Type

A_1 : $A_0 \to A_0 \to$ Type

A_2 : $\Pi\{x\, y\, z : A_0\}.\, (A_1\, x\, y) \to (A_1\, y\, z) \to (A_1\, x\, z) \to$ Type



(2) Adding the Segal condition

Semicategory (beginning)

Ob : Type

Hom : Ob $\to$ Ob $\to$ Type

_ $\circ$ _ : $\{x\, y\, z : \mathsf{Ob}\} \to (\mathsf{Hom}\, y\, z) \to (\mathsf{Hom}\, x\, y) \to (\mathsf{Hom}\, x\, z)$

Semisimplicial type (beginning)

A_0 : Type

$A_1 : A_0 \to A_0 \to \mathsf{Type}$

$A_2 : \{x\, y\, z : A_0\} \to (A_1\, y\, z) \to (A_1\, x\, y) \to (A_1\, x\, z) \to \mathsf{Type}$

Lemma: For X : Type, we have $X \simeq \Sigma(P : X \to \mathsf{Type}).\ \mathsf{isContr}(\Sigma(x : X).P\, x)$.

(2) Adding the Segal condition

Semicategory (beginning)

$\mathsf{Ob} : \mathsf{Type}$

$\mathsf{Hom} : \mathsf{Ob} \to \mathsf{Ob} \to \mathsf{Type}$

$$\_ \circ \_ : \{x\, y\, z : \mathsf{Ob}\} \to (\mathsf{Hom}\, y\, z) \to (\mathsf{Hom}\, x\, y) \to (\mathsf{Hom}\, x\, z)$$

Semisimplicial type (beginning)

$A_0 : \mathsf{Type}$

$A_1 : A_0 \to A_0 \to \mathsf{Type}$

$$A_2 : \{x\, y\, z : A_0\} \to (A_1\, y\, z) \to (A_1\, x\, y) \to (A_1\, x\, z) \to \mathsf{Type}$$

Lemma: For $X : \mathsf{Type}$, we have $X \simeq \Sigma(P : X \to \mathsf{Type}).\, \mathsf{isContr}(\Sigma(x : X).P\, x)$.

(2) Adding the Segal condition

Semicategory (beginning)

$\mathsf{Ob} : \mathsf{Type}$

$\mathsf{Hom} : \mathsf{Ob} \to \mathsf{Ob} \to \mathsf{Type}$

$\_ \circ \_ : \{x\, y\, z : \mathsf{Ob}\} \to (\mathsf{Hom}\, y\, z) \to (\mathsf{Hom}\, x\, y) \to (\mathsf{Hom}\, x\, z)$

Semisimplicial type (beginning)

$A_0 : \mathsf{Type}$

$A_1 : A_0 \to A_0 \to \mathsf{Type}$

$A_2 : \{x\, y\, z : A_0\} \to (A_1\, y\, z) \to (A_1\, x\, y) \to (A_1\, x\, z) \to \mathsf{Type}$

Lemma: For $X : \mathsf{Type}$, we have $X \simeq \Sigma(P : X \to \mathsf{Type}).\ \mathsf{isContr}(\Sigma(x : X).P\, x)$.

(2) Adding the Segal condition

Semicategory (beginning)

$$\mathsf{Ob} : \mathsf{Type}$$

$$\mathsf{Hom} : \mathsf{Ob} \to \mathsf{Ob} \to \mathsf{Type}$$

$$\_ \circ \_ : \{x\, y\, z : \mathsf{Ob}\} \to (\mathsf{Hom}\, y\, z) \to (\mathsf{Hom}\, x\, y) \to (\mathsf{Hom}\, x\, z)$$

notation for $\Pi(x\; y\; z : \mathsf{Ob}). \ldots$

Semisimplicial type (beginning)

$$A_0 : \mathsf{Type}$$

$$A_1 : A_0 \to A_0 \to \mathsf{Type}$$

$$A_2 : \{x\, y\, z : A_0\} \to (A_1\, y\, z) \to (A_1\, x\, y) \to (A_1\, x\, z) \to \mathsf{Type}$$

Lemma: For X : Type, we have $X \simeq \Sigma(P : X \to \mathsf{Type}).\ \mathsf{isContr}(\Sigma(x : X).P\, x)$.

(2) Adding the Segal condition

Semicategory (beginning)

$\mathsf{Ob} : \mathsf{Type}$

$\mathsf{Hom} : \mathsf{Ob} \to \mathsf{Ob} \to \mathsf{Type}$

$\_ \circ \_ : \{x\, y\, z : \mathsf{Ob}\} \to (\mathsf{Hom}\, y\, z)$
$\quad \to (\mathsf{Hom}\, x\, y) \to (\mathsf{Hom}\, x\, z)$

Semisimplicial type (beginning)

$A_0 : \mathsf{Type}$

$A_1 : A_0 \to A_0 \to \mathsf{Type}$

$A_2 : \{x\, y\, z : A_0\} \to (A_1\, y\, z)$
$\quad \to (A_1\, x\, y) \to (A_1\, x\, z) \to \mathsf{Type}$

$h_2 : \{x\, y\, z : A_0\} \to (g : A_1\, y\, z) \to (f : A_1\, x\, y)$
$\quad \to \mathsf{isContr}(\Sigma(h : A_1\, x\, z).\, A_2\, g\, f\, h)$

Lemma: For $X : \mathsf{Type}$, we have $X \simeq \Sigma(P : X \to \mathsf{Type})$.
$\mathsf{isContr}(\Sigma(x : X).P\, x)$.

(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).

syntax is (maybe) not univalent...

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.

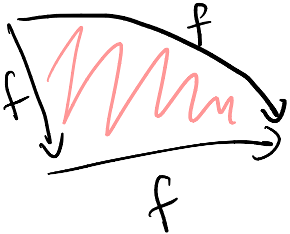
(3) Add identities/degeneracies

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

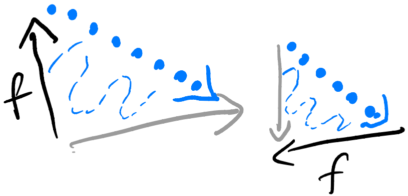
Here: We don't want built-in univalence. Instead:

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 ∞-*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 ∞-*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 ∞-CwF

  $\Rightarrow$ the syntax is an ∞-CwF
- Every "wild" 1-CwF, where equations hold strictly, is an ∞-CwF

  $\Rightarrow$ standard model (universe) is an ∞-CwF
- Other constructions, e.g. slice ∞-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 ∞-CwF
  $\Rightarrow$ the syntax is an ∞-CwF
- Every "wild" 1-CwF, where equations hold strictly, is an ∞-CwF
  $\Rightarrow$ standard model (universe) is an ∞-CwF
- Other constructions, e.g. slice ∞-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 ∞-CwF

  $\Rightarrow$ the syntax is an ∞-CwF
- Every "wild" 1-CwF, where equations hold strictly, is an ∞-CwF

  $\Rightarrow$ standard model (universe) is an ∞-CwF
- Other constructions, e.g. slice ∞-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 ∞-CwF

  $\Rightarrow$ the syntax is an ∞-CwF
- Every "wild" 1-CwF, where equations hold strictly, is an ∞-CwF

  $\Rightarrow$ standard model (universe) is an ∞-CwF
- Other constructions, e.g. slice ∞-CwF ("working with assumptions")

Main unsolved problem:

- Is the syntax initial?

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

THANKS!