Internal ∞-Categories with Families

Nicolai Kraus

MSP 101 Seminar (Strathclyde), 18 Feb 2021

Talk based on arXiv:2009.01883.

introduction: Joshua Chen
(started PhD Oct '20)

$\Longrightarrow$ **Part 1:** Why do we want ∞-CwF's?

Part 2: How to define them?

Part 3: What works or is still missing?

Goal: Define what a model of type theory is – in type theory!
(in particular: intended initial model ∼ "syntax")

Peter Dybjer, 2005: *Internal Type Theory*

Danielsson 2006
Chapman 2009
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., 2020
...

```
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 GAT

Tm: $\int^{op}$ Ty $\to$ Set

Con	:	Type
Sub	:	Con $\to$ Con $\to$ Type
$\_ \diamond \_$	:	Sub $\Theta\,\Delta \to$ Sub $\Gamma\,\Theta \to$ Sub $\Gamma\,\Delta$
assoc	:	$(\sigma \diamond \delta) \diamond \nu = \sigma \diamond (\delta \diamond \nu)$
id	:	Sub $\Gamma\,\Gamma$
idl$_\sigma$	:	id $\diamond\, \sigma = \sigma$
idr$_\sigma$	:	$\sigma \diamond$ id $= \sigma$

$\mathcal{C}$ category of contexts + subs.

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

term. obj.

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

Ty : $\mathcal{C}^{op} \to$ Set

Tm	:	$(\Gamma :$ Con$) \to$ Ty $\Gamma \to$ Type	
$\_[\_]^{\mathsf{t}}$	:	Tm $\Delta\,A \to (\sigma :$ Sub $\Gamma\,\Delta) \to$ 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 $[\diamond]^{\mathsf{T}}$
$\_ \triangleright \_$	:	$(\Gamma :$ Con$) \to$ Ty $\Gamma \to$ Con	
p	:	Sub $(\Gamma \triangleright A)\,\Gamma$	
q	:	Tm $(\Gamma \triangleright A)\,(A[\mathsf{p}]^{\mathsf{T}})$	
$\_ , \_$	:	$(\sigma :$ Sub $\Gamma\,\Delta) \to$ Tm $\Gamma\,(A[\sigma]^{\mathsf{T}}) \to$ 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 $[\diamond]^{\mathsf{T}}$

context extension

..

representability of the functor $(\mathcal{C}/\Delta)^{op} \to$ Set, $(\Gamma, \sigma) \mapsto$ Tm Γ $A[\sigma]$

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

First example of a CwF: **"Syntax QIIT"**, a.k.a.
the initial model as a QIIT (Altenkirch-Kaposi 2016)

data Con : Type where

$\bullet$: Con

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

data Sub : Con $\to$ Con $\to$ Type where

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

$\ldots$

data Ty : Con $\to$ Type where

$\ldots$

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

$\ldots$

Initiality theorem (Brunerie, de Boer, Lumsdaine, Mörtberg 2019–today) implies:
Syntax QIIT $\simeq$ non-well-typed syntax with wellformedness predicates.

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 $\Pi(x:\Gamma).(A\,x)$ $(x : \Gamma) \to A\,x$
- all operations are canonical
- all equations hold judgmentally (in Agda)

The trouble with(out) UIP

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

$$(x\ y : A) \to (p\ q : x = y) \to (p = q)$$

The above definition of a CwF works assuming UIP.

What if UIP is not assumed?

Happens e.g. in HoTT and in Agda `{-# OPTIONS --without-K #-}`

Two canonical approaches:

(1) Ignore it: Do everything as before.

or

(2) Make up for it: Assume that Con, Sub, Ty, Tm are families of h-sets.

No UIP: problems of the canonical approaches

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

But then:

$$\begin{aligned} \mathsf{idl}_\sigma &: \ \mathsf{id} \diamond \sigma = \sigma \\ \mathsf{idr}_\sigma &: \ \sigma \diamond \mathsf{id} = \sigma \end{aligned}$$

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.
$\Rightarrow$ "Syntax QIIT" (example 1) is not initial.

(2) 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" (example 2) fails.

Why we really want both examples (syntax QIIT 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 a solution:

$$\begin{tikzcd} \boxed{\begin{array}{l}\text{Syntax}\\ \text{(untyped)}\end{array}} \arrow[r, leftrightarrow, "\text{initiality thm}"] & \boxed{\begin{array}{l}\text{Syntax}\\ \text{(as QIIT)}\end{array}} \arrow[r, "\text{by initiality}"] & \boxed{\begin{array}{l}\text{universe } \mathcal{U}\\ \text{(standard model)}\end{array}} \end{tikzcd}$$

only works w/ UIP

does not work w/ UIP

Back to the definition from slide 4:

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

$idl_{id} = idr_{id}$

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

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.

Part 1: Why do we want ∞-CwF's?

$\Longrightarrow$ **Part 2:** How to define them?

Part 3: What works or is still missing?

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. This talk: ∞-categories (the first point).

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

Strategy:
(1) Start with a *semisimplicial type* ("basic structure")
(2) Add Segal condition ($\Rightarrow$ ∞-semicategory)
(3) Add identities ($\Rightarrow$ ∞-category)

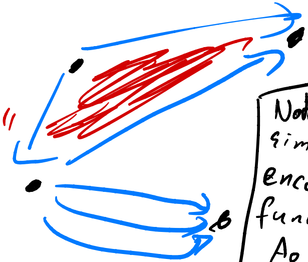
(1) Recall: semisimplicial type up to level 2 is tuple (A_0, A_1, A_2) where

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

A_0 is type of "points"

A_1 is type of "lines"

A_2 is type of "triangle fillers"



Note: A semi-simplicial type encodes a functor $\Delta_+^{op} \to Type$

$$A_0$$
$$\uparrow\uparrow$$
$$\Sigma(x\, y : A_0).\ A_1\, x\, y$$
$$\uparrow\uparrow\uparrow$$
$$\Sigma(x\, y\, z : A_0).\ \ldots A_2 \ldots$$

Caveat: Known open problem to construct this in HoTT for general n.

"Solution": Use 2LTT.

(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)

$A_0 : \mathsf{Type}$

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

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

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

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

$x \longmapsto (\lambda y \to x = y\ ,\ \text{singletons-are-contr})$

$\mathrm{fst}(\mathrm{fst}\ c) \longleftarrow (P, c)$

(Empty page)

$$\begin{tikzcd} & z \\ x \arrow[ur, dashed, "h"] \arrow[r, "f"'] & y \arrow[u, "g"'] \end{tikzcd}$$

h_2 says:
given f, g there $\exists$
exactly one h
s.t. $A_2\, g\, f\, h$

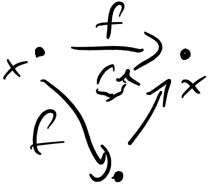
(3) Add identities/degeneracies

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 morphism $f : A_1\, x\, y$ 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.

$x \xrightarrow{f} x$

gives: $A_1\, x\, y \to A_1\, x$

two maps: pre- and post composition

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.)

Part 1: Why do we want ∞-CwF's?

Part 2: How to define them?

$\Longrightarrow$ **Part 3:** What works or is still missing?

Done (see paper):

- Definition of ∞-CwF's
- Variations, such as univalent or finite-dimensional ∞-CwF's
- Syntax QIIT as an ∞-CwF
- Standard model as an ∞-CwF $= ?$
- Initial ∞-CwF (given appropriate techniques)
- Slice ∞-CwF's

To do:

- Initiality of the Syntax QIIT
- Interderivability (in some suitable sense) of the two open problems "Can HoTT eat itself?" and "Can we define semisimplicial types?"
- add Π, Σ, $\mathbb{N}$, $\mathcal{U}$, ...

(Thanks for your attention! The end.)