The groupoid-syntax of type theory is a set

Szumi Xie
jww Thorsten Altenkirch and Ambrus Kaposi
Eötvös Loránd University (ELTE)

HoTTEST, 2 April 2026

Intrinsic syntax of type theory

Categories with families (CwFs)

CwFs are a notion of model of type theory.

It can be presented as a generalized algebraic theory.

The initial algebra can be defined as a higher inductive-inductive type (HIIT) in type theory, which can be seen as an intrisically typed syntax of type theory.

Components of a CwF

A category Con of contexts and substitutions

$$
\begin{aligned}
\mathsf{Con} &: \mathsf{Type} \\
\mathsf{Sub} &: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Type} \\
\mathsf{id} &: \mathsf{Sub}\,\Gamma\,\Gamma \\
- \circ - &: \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Theta\,\Gamma \\
\mathsf{idl} &: \mathsf{id} \circ \sigma = \sigma \\
\mathsf{idr} &: \sigma \circ \mathsf{id} = \sigma \\
\mathsf{assoc} &: \sigma \circ (\delta \circ \theta) = (\sigma \circ \delta) \circ \theta
\end{aligned}
$$

with a terminal object

$$
\begin{aligned}
&\diamond : \mathsf{Con} \\
&\mathsf{Sub}\,\Gamma\,\diamond \cong \mathbb{1}
\end{aligned}
$$

Components of a CwF (cont.)

A presheaf Ty over Con

$$
\begin{aligned}
\mathsf{Ty} \quad &: \mathsf{Con} \to \mathsf{Type} \\
-[-] &: \mathsf{Ty}\,\Gamma \to \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{Ty}\,\Delta \\
[\mathsf{id}] \quad &: A[\mathsf{id}] = A \\
[\circ] \quad &: A[\sigma \circ \delta] = A[\sigma][\delta]
\end{aligned}
$$

A presheaf Tm over $\int \mathsf{Ty}$

$$
\begin{aligned}
\mathsf{Tm} \quad &: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type} \\
-[-] &: \mathsf{Tm}\,\Gamma\,A \to (\sigma : \mathsf{Sub}\,\Delta\,\Gamma) \to \mathsf{Tm}\,\Delta\,(A[\sigma]) \\
&\ldots \text{ (+ functoriality)}
\end{aligned}
$$

Context comprehension

$$
\begin{aligned}
&- \triangleright - : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con} \\
&\mathsf{Sub}\,\Delta\,(\Gamma \triangleright A) \cong (\sigma : \mathsf{Sub}\,\Delta\,\Gamma) \times \mathsf{Tm}\,\Delta\,(A[\sigma])
\end{aligned}
$$

Type formers in a CwF

$$
\begin{aligned}
\mathsf{U} &: \mathsf{Ty}\,\Gamma \\
\mathsf{U}[] &: \mathsf{U}[\sigma] = \mathsf{U} \\
\\
\mathsf{El} &: \mathsf{Tm}\,\Gamma\,\mathsf{U} \to \mathsf{Ty}\,\Gamma \\
\mathsf{El}[] &: (\mathsf{El}\,t)[\sigma] = \mathsf{El}\,(\mathsf{U}[]_*\,(t[\sigma])) \\
\\
\Pi &: (A : \mathsf{Ty}\,\Gamma) \to \mathsf{Ty}\,(\Gamma \triangleright A) \to \mathsf{Ty}\,\Gamma \\
\Pi[] &: (\Pi\,A\,B)[\sigma] = \Pi\,(A[\sigma])\,(B[\sigma^+])
\end{aligned}
$$

… (+ term formers)

The wild syntax

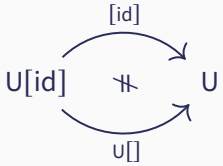
The type interpretation of the wild syntax

Interpretation into the standard/metacircular/type model

$$
\begin{aligned}
[\![\Gamma]\!] &: \mathsf{Type} \\
[\![\sigma]\!] &: [\![\Delta]\!] \to [\![\Gamma]\!] \\
[\![A]\!] &: [\![\Gamma]\!] \to \mathsf{Type} \\
[\![t]\!] &: (x : [\![\Gamma]\!]) \to [\![A]\!]\, x
\end{aligned}
$$

$$
\begin{aligned}
[\![\mathsf{id}]\!]\, x &:\equiv x \\
[\![\sigma \circ \delta]\!]\, x &:\equiv [\![\sigma]\!]\, ([\![\delta]\!]\, x) \\
[\![A[\sigma]]\!]\, x &:\equiv [\![A]\!]\, ([\![\sigma]\!]\, x) \\
&\dots
\end{aligned}
$$

Types do not form sets in the wild syntax



$$[\![A]\!] \quad : \mathsf{S}^1$$
$$[\![\mathsf{U}]\!] \quad :\equiv \mathsf{base} \qquad [\![[\mathsf{id}]]\!] :\equiv \mathsf{refl}$$
$$[\![A[\sigma]]\!] :\equiv [\![A]\!] \qquad [\![\mathsf{U}[]]\!] :\equiv \mathsf{loop}$$

Hedberg's theorem: types do not have decidable equality

The set-syntax

The set-syntax

We add:

$$\begin{aligned}\text{isSetSub} &: (p, q : \sigma = \delta) \to p = q \\ \text{isSetTy} &: (p, q : A = B) \to p = q \\ \text{isSetTm} &: (p, q : t = u) \to p = q\end{aligned}$$

We cannot directly interpret the set-syntax into the set model, since set is not an set.

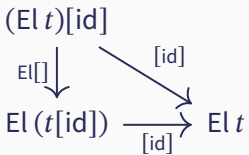
The groupoid-syntax

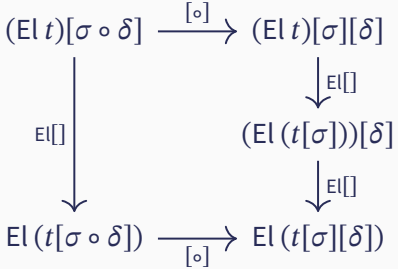
The groupoid-syntax

We replace isSetTy with

$$\mathsf{isGroupoidTy} : (p, q : A = B) \to (\alpha, \beta : p = q) \to \alpha = \beta$$

and add coherences





The set interpretation of the groupoid-syntax

$$
\begin{aligned}
&[\![\Gamma]\!] && : \mathsf{hSet}\\
&[\![\sigma]\!] && : [\![\Delta]\!] \to [\![\Gamma]\!]\\
&[\![A]\!] && : [\![\Gamma]\!] \to \mathsf{hSet}\\
&[\![t]\!] && : (x : [\![\Gamma]\!]) \to [\![A]\!]\, x
\end{aligned}
$$

$$
\begin{aligned}
&[\![\mathsf{id}]\!]\, x && \coloneqq x\\
&[\![\sigma \circ \delta]\!]\, x && \coloneqq [\![\sigma]\!]\, ([\![\delta]\!]\, x)\\
&[\![A[\sigma]]\!]\, x && \coloneqq [\![A]\!]\, ([\![\sigma]\!]\, x)\\
&[\![\mathsf{isGroupoidTy}]\!] && \coloneqq \mathsf{isGroupoidSet}\\
&\ldots
\end{aligned}
$$

α-normalization (better name: ζ-normalization?)

α-normal types

$$
\begin{aligned}
&\mathsf{NTy} && : \mathsf{Con} \to \mathsf{Type} \\
&\mathsf{U} && : \mathsf{NTy}\,\Gamma \\
&\mathsf{El} && : \mathsf{Tm}\,\Gamma\,\mathsf{U} \to \mathsf{NTy}\,\Gamma \\
&\Pi && : (A : \mathsf{NTy}\,\Gamma) \to \mathsf{NTy}\,(\Gamma \triangleright \ulcorner A \urcorner) \to \mathsf{NTy}\,\Gamma \\
&\ulcorner - \urcorner && : \mathsf{NTy}\,\Gamma \to \mathsf{Ty}\,\Gamma \\
&\ulcorner \mathsf{U} \urcorner && :\equiv \mathsf{U} \\
&\ulcorner \mathsf{El}\,t \urcorner && :\equiv \mathsf{El}\,t \\
&\ulcorner \Pi\,A\,B \urcorner && :\equiv \Pi\,\ulcorner A \urcorner\,\ulcorner B \urcorner
\end{aligned}
$$

α-normal types form sets

The encode-decode argument:

$$
\begin{aligned}
&\mathsf{Code} && : \mathsf{NTy}\,\Gamma \to \mathsf{NTy}\,\Gamma \to \mathsf{hProp} \\
&\mathsf{Code}\ \mathsf{U}\quad \mathsf{U} && :\equiv \mathbb{1} \\
&\mathsf{Code}\ (\mathsf{El}\ t)\quad (\mathsf{El}\ t') && :\equiv (t = t') \\
&\mathsf{Code}\ (\Pi\ A\ B)\ (\Pi\ A'\ B') && :\equiv (c : \mathsf{Code}\ A\ A') \times \mathsf{Code}\ ((\mathsf{decode}\ c)_*\ B)\ B' \\
&\text{otherwise} && :\equiv \mathbb{0}
\end{aligned}
$$

$$\mathsf{decode} : \mathsf{Code}\ A\ A' \to A = A'$$

By induction, we have $\mathsf{Code}\ A\ A' \cong (A = A')$, which implies $(A = A')$ is a proposition, that is, $\mathsf{NTy}\ \Gamma$ is a set.

α-normalization

To show that $\mathsf{Ty}\,\Gamma$ is a set, we want that $\mathsf{NTy}\,\Gamma \cong \mathsf{Ty}\,\Gamma$, so we define the following:

$$
\begin{aligned}
[\![A]\!] &: (A : \mathsf{NTy}\,\Gamma) \times (\ulcorner A \urcorner = A) \\
[\![\mathsf{U}]\!] &:\equiv (\mathsf{U}, \mathsf{refl}) \\
[\![\mathsf{El}\,t]\!] &:\equiv (\mathsf{El}\,t, \mathsf{refl}) \\
[\![A[\sigma]]\!] &:\equiv \textbf{?}
\end{aligned}
$$

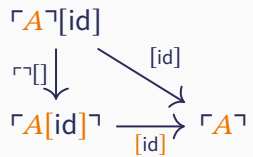
Substitution of α-normal types

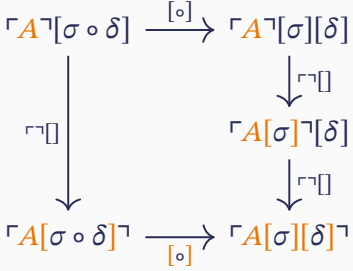
$$
\begin{aligned}
-[-] &: \mathsf{NTy}\,\Gamma \to \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{NTy}\,\Delta \\
\mathsf{U}[\sigma] &:\equiv \mathsf{U} \\
(\mathsf{El}\,t)[\sigma] &:\equiv \mathsf{El}\,(t[\sigma]) \\
(\Pi\,A\,B)[\sigma] &:\equiv \Pi\,(A[\sigma])\,(\ulcorner\urcorner[]_*\,(B[\sigma^+]))
\end{aligned}
$$

$$
\ulcorner\urcorner[] : \ulcorner A \urcorner[\sigma] = \ulcorner A[\sigma] \urcorner
$$

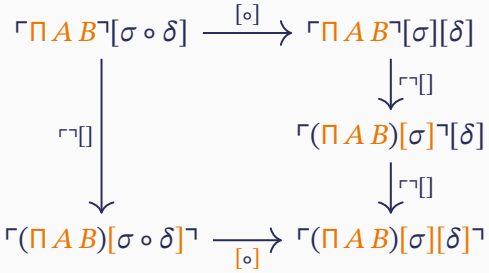
$$
\begin{aligned}
[\mathsf{id}] &: A[\mathsf{id}] = A \\
[\circ] &: A[\sigma \circ \delta] = A[\sigma][\delta]
\end{aligned}
$$

Substitution of α-normal types (coherences)

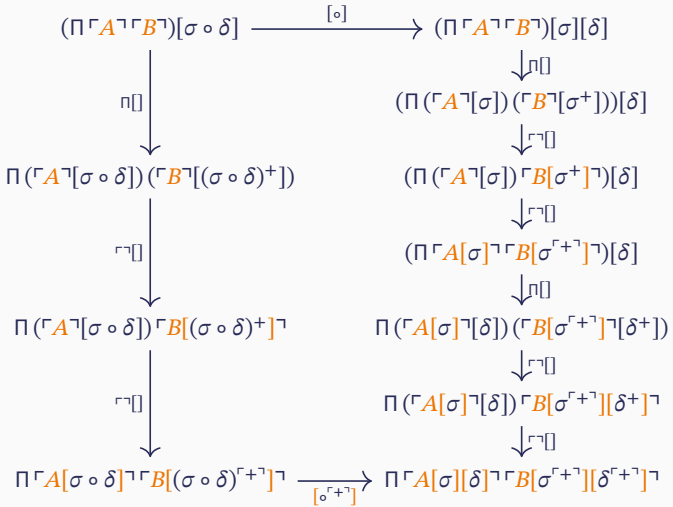




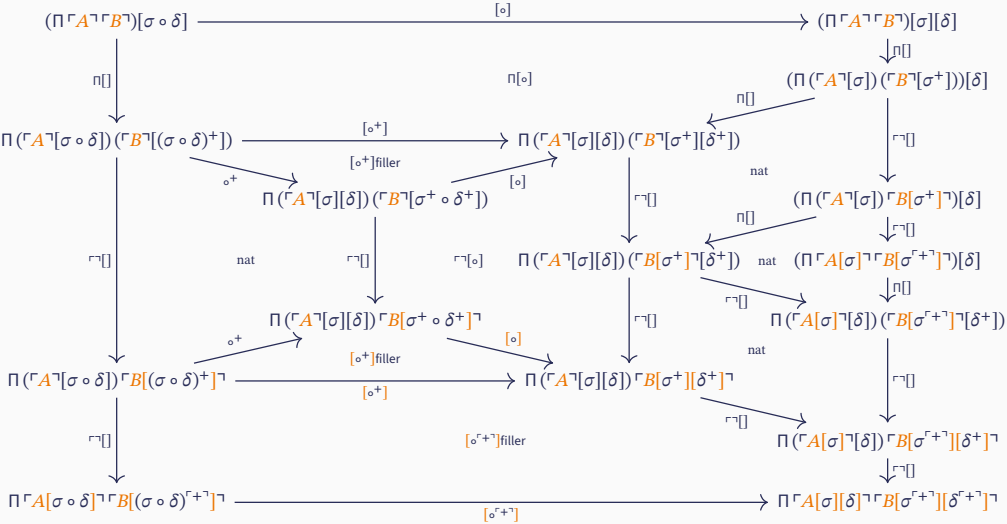
Example proof goal



Example proof goal (computed)



Example proof



α-normalization

$$
\begin{aligned}
[\![A]\!] &: (A : \mathsf{NTy}\,\Gamma) \times (\ulcorner A \urcorner = A) \\
[\![\mathsf{U}]\!] &:= (\mathsf{U}, \mathsf{refl}) \\
[\![\mathsf{El}\,t]\!] &:= (\mathsf{El}\,t, \mathsf{refl}) \\
[\![A[\sigma]]\!] &:= ([\![A]\!][\sigma], \ulcorner\urcorner[]) \\
[\![[\mathsf{id}]]\!] &:= ([\mathsf{id}], \dots) \\
&\dots
\end{aligned}
$$

The groupoid-syntax is a set

The result

We have NTy $\Gamma \cong$ Ty Γ, which implies Ty Γ is a set.

By induction, the groupoid-syntax isomorphic to the set-syntax, thus we can interpret the set-syntax into the set model.

$$\text{set-syntax} \longrightarrow \text{groupoid-syntax} \longrightarrow \text{set model}$$

We did not need full normalization, α-normalization works even with an extensional identity type.

Open question: equations in Ty, e.g. Booleans with large elimination:

$$\begin{aligned}
&\text{ElimBool} : \text{Ty}\,\Gamma \to \text{Ty}\,\Gamma \to \text{Tm}\,\Gamma\,\text{Bool} \to \text{Ty}\,\Gamma\\
&\text{Bool}\beta_1 \quad : \text{ElimBool}\,A\,B\,\text{false} = A\\
&\text{Bool}\beta_2 \quad : \text{ElimBool}\,A\,B\,\text{true} = B
\end{aligned}$$

Groupoid categories with families

Groupoid CwFs (synthetic)

A category Con

A pseudofunctor Ty from $\mathsf{Con}^{\mathsf{op}}$ to the bicategory of h-groupoids

A presheaf Tm over $\int \mathsf{Ty}$

Context comprehension

Definable components of Ty:

$$
\begin{aligned}
&\mathsf{Tym} \;:\; \mathsf{Ty}\,\Gamma \to \mathsf{Ty}\,\Gamma \to \mathsf{hSet}\\
&\mathsf{Tym}\,A\,B \coloneqq (A = B)\\
&- \circ - \;:\; \mathsf{Tym}\,A\,B \to \mathsf{Tym}\,B\,C \to \mathsf{Tym}\,A\,C\\
&\mathsf{inv} \;:\; \mathsf{Tym}\,A\,B \to \mathsf{Tym}\,B\,A\\
&-[-] \;:\; \mathsf{Tym}\,A\,B \to (\sigma : \mathsf{Sub}\,\Delta\,\Gamma) \to \mathsf{Tym}\,(A[\sigma])\,(B[\sigma])\\
&\circ[] \;:\; (f \circ g)[\sigma] = f[\sigma] \circ g[\sigma]
\end{aligned}
$$

Groupoid CwFs (non-synthetic)

A category Con

A pseudofunctor Ty from $\mathsf{Con}^{\mathrm{op}}$ to the bicategory of groupoids

A presheaf Tm over $\int \mathsf{Ty}$

Context comprehension

Some components of Ty:

$$\mathsf{Tym} \;:\; \mathsf{Ty}\,\Gamma \to \mathsf{Ty}\,\Gamma \to \mathsf{hSet}$$

$$\begin{aligned}
-\circ- &: \mathsf{Tym}\,A\,B \to \mathsf{Tym}\,B\,C \to \mathsf{Tym}\,A\,C \\
\mathsf{inv} &: \mathsf{Tym}\,A\,B \to \mathsf{Tym}\,B\,A \\
-[-] &: \mathsf{Tym}\,A\,B \to (\sigma : \mathsf{Sub}\,\Delta\,\Gamma) \to \mathsf{Tym}\,(A[\sigma])\,(B[\sigma]) \\
\circ[] &: (f \circ g)[\sigma] = f[\sigma] \circ g[\sigma]
\end{aligned}$$

Category CwFs

A category Con

A pseudofunctor Ty from $\mathsf{Con}^{\mathsf{op}}$ to the bicategory of categories

A presheaf Tm over $\int \mathsf{Ty}$

Context comprehension

Some components of Ty:

$$\mathsf{Tym} \;:\; \mathsf{Ty}\,\Gamma \to \mathsf{Ty}\,\Gamma \to \mathsf{hSet}$$

$$- \circ - \;:\; \mathsf{Tym}\,A\,B \to \mathsf{Tym}\,B\,C \to \mathsf{Tym}\,A\,C$$

$$-[-] \;:\; \mathsf{Tym}\,A\,B \to (\sigma : \mathsf{Sub}\,\Delta\,\Gamma) \to \mathsf{Tym}\,(A[\sigma])\,(B[\sigma])$$

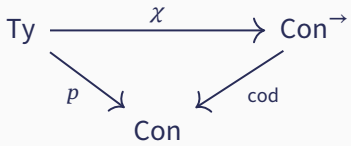
$$\circ[] \;:\; (f \circ g)[\sigma] = f[\sigma] \circ g[\sigma]$$

Comprehension categories

A category Con

A fibration $p : \mathrm{Ty} \to \mathrm{Con}$

A cartesian morphism preserving functor χ from Ty to $\mathrm{Con}^{\to}$, such that the following diagram commutes:



Conjecture: the 2-category of comprehension categories is equivalent to some 2-category of category CwFs.

Conclusion

Conclusion

We defined groupoid CwFs as a generalization of CwFs, where types are groupoids rather than sets.

The initial groupoid CwF with type formers (the groupoid-syntax) is “coherent”, types form sets, proven using α-normalization.

This allows us to interpret the set-syntax into the set model.

We formalized this in Cubical Agda.