Containers form a Groupoid CwF

Stefania Damato

Thorsten Altenkirch

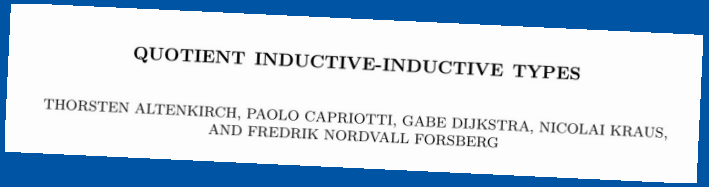
TYPES 2026

Motivation

- Provide semantics for Quotient Inductive-Inductive Types (QIITs) via containers.

Motivation

- Provide semantics for Quotient Inductive-Inductive Types (QIITs) via containers.
- Approach: Restrict existing QIIT semantics.

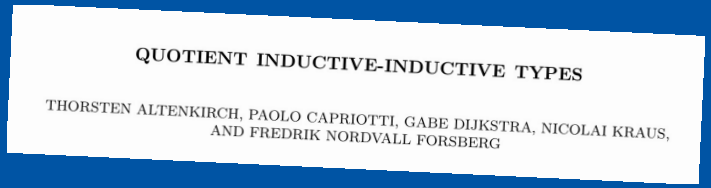


QUOTIENT INDUCTIVE-INDUCTIVE TYPES

THORSTEN ALTENKIRCH, PAOLO CAPRIOTTI, GABE DIJKSTRA, NICOLAI KRAUS, AND FREDRIK NORDVALL FORSBERG

Motivation

- Provide <u>semantics</u> for Quotient Inductive-Inductive Types (QIITs) via containers.
- Approach: Restrict existing QIIT semantics.



QUOTIENT INDUCTIVE-INDUCTIVE TYPES

THORSTEN ALTENKIRCH, PAOLO CAPRIOTTI, GABE DIJKSTRA, NICOLAI KRAUS, AND FREDRIK NORDVALL FORSBERG

⇝ Can we represent any statement in type theory as some container?

Background

- Containers a.k.a. polynomial functors.

set of shapes ← $S \triangleleft P$ → family of positions

Background

- Containers a.k.a. polynomial functors.

  set of shapes ← $S \triangleleft P$ → family of positions

- Category with families (CwF): A notion of model of type theory.

Background

- Containers a.k.a. polynomial functors.

  set of shapes ← $S \triangleleft P$ → family of positions

- Category with families (CwF): A notion of model of type theory.

- Groupoid CwF: A generalisation of CwF where types can be h-groupoids.

Related work

A container model of type theory*

Thorsten Altenkirch[1] and Ambrus Kaposi[12]

[1] University of Nottingham, Nottingham, United Kingdom
[2] Eötvös Loránd University, Budapest, Hungary

TYPES 2021 abstract

Related work

A container model of type theory*

Thorsten Altenkirch[1] and Ambrus Kaposi[12]

[1] University of Nottingham, Nottingham, United Kingdom
[2] Eötvös Loránd University, Budapest, Hungary

TYPES 2021 abstract

The Groupoid-Syntax of Type Theory Is a Set

Thorsten Altenkirch
University of Nottingham, UK

Ambrus Kaposi
Eötvös Loránd University (ELTE), Budapest, Hungary

Szumi Xie
Eötvös Loránd University (ELTE), Budapest, Hungary

CSL 2026

Related work

A container model of type theory*

Thorsten Altenkirch[1] and Ambrus Kaposi[12]

[1] University of Nottingham, Nottingham, United Kingdom
[2] Eötvös Loránd University, Budapest, Hungary

TYPES 2021 abstract

The Groupoid-Syntax of Type Theory Is a Set

Thorsten Altenkirch
University of Nottingham, UK
Ambrus Kaposi
Eötvös Loránd University (ELTE), Budapest, Hungary
Szumi Xie
Eötvös Loránd University (ELTE), Budapest, Hungary

CSL 2026

Polynomials and Models of Type Theory

Tamara von Glehn
Magdalene College
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2014

PhD thesis

The container model

- contexts
- types in Γ
- type substitution
- terms of A

$$(S_\Gamma : \mathrm{Set}) \triangleleft (P_\Gamma : S_\Gamma \to \mathrm{Set})$$

$$(S_A : \mathrm{Set}) \triangleleft (P_A : S_A \to \textstyle\int [\![\Gamma]\!])$$

"pullback $\triangleleft$ pushout"

$$\int_X (x_\Gamma : [\![\Gamma]\!]\, X) \to [\![A]\!](X, x_\Gamma)$$

components of A

- context extension $\Gamma \triangleright A$

Coherence issues

In a CwF, types should form an h-set.

Coherence issues

In a CwF, types should form an h-set.

Set model: contexts $\Gamma : \mathrm{Set}$

types in Γ $A : \Gamma \to \mathrm{Set}$

Coherence issues

In a CwF, types should form an h-set.

Set model: contexts $\Gamma : \mathrm{Set}$
types in Γ $A : \Gamma \to \mathrm{Set}$

Presheaf model: contexts $\Gamma : \mathcal{C}^{op} \to \mathrm{Set}$
types in Γ $A : \prod_{x:\mathcal{C}} \Gamma(x) \to \mathrm{Set}$

Coherence issues

In a CwF, types should form an h-set.

Set model: contexts $\Gamma : \mathrm{Set}$
types in Γ $A : \Gamma \to \mathrm{Set}$

Presheaf model: contexts $\Gamma : \mathcal{C}^{op} \to \mathrm{Set}$
types in Γ $A : \prod_{X:\mathcal{C}} \Gamma(X) \to \mathrm{Set}$

Without UIP, types are not h-sets.

Coherence issues

In a CwF, types should form an h-set.

Container model:

contexts $(S_\Gamma : \mathrm{Set}) \triangleleft (P_\Gamma : S_\Gamma \to \mathrm{Set})$

types in Γ $(S_A : \mathrm{Set}) \triangleleft (P_A : S_A \to \int \llbracket \Gamma \rrbracket)$

Without UIP, types are not h-sets.

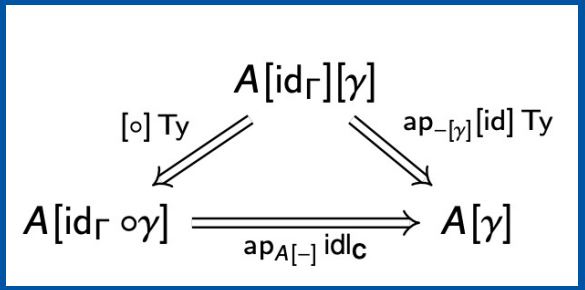
Groupoid CwFs

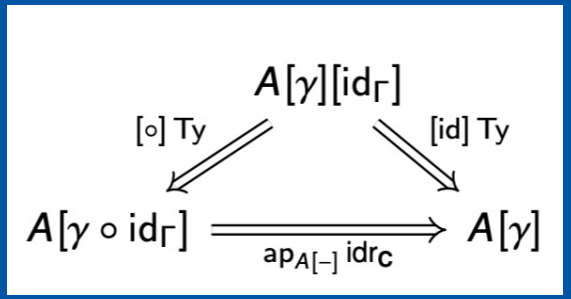
✓ Types are now h-groupoids.

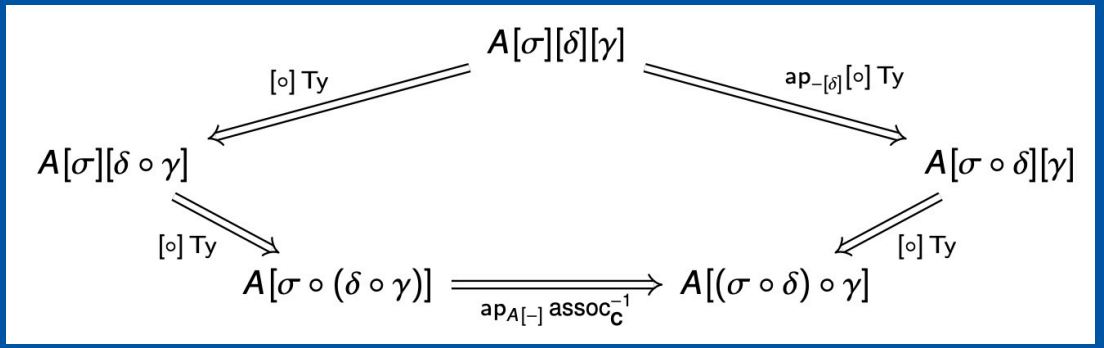
Groupoid CwFs

✓ Types are now h-groupoids.

But we have to prove extra coherences!





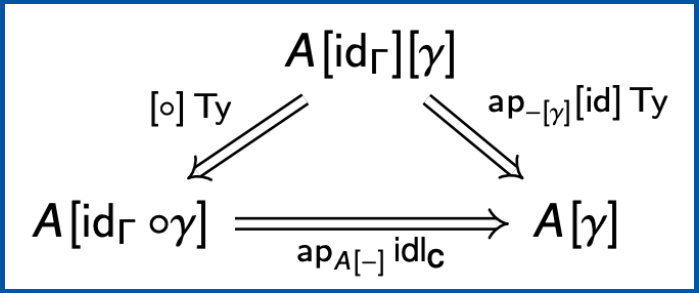


Our contribution

- Proved the coherence laws in a cubical setting.
- Fully formalised the container groupoid CwF in Cubical Agda.

Proof sketch: Left identity coherence

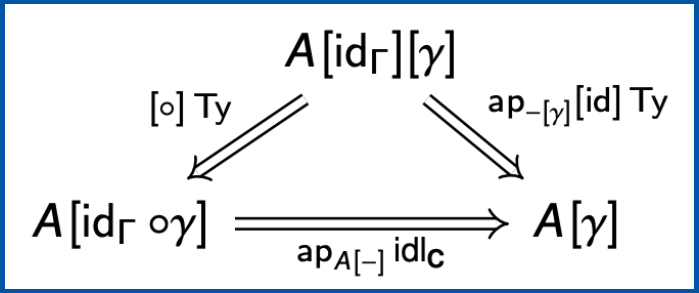
To prove:



Thanks Axel!

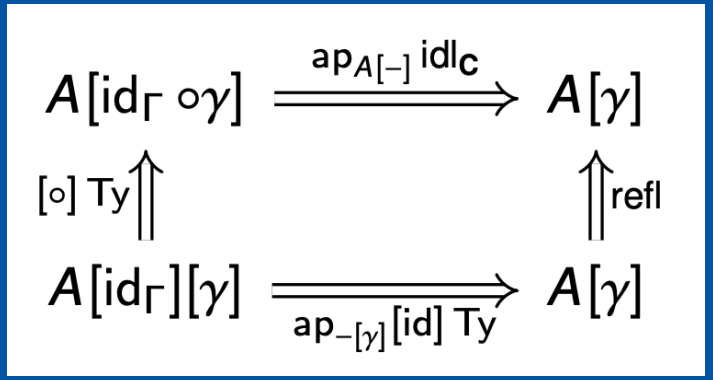
Proof sketch: Left identity coherence

To prove:



Thanks Axel!

Step 1: Write it as a square.



Proof sketch: Left identity coherence

Step 2: Rewrite the square as

where $\mathsf{uaGenCon} : A \simeq_{\mathsf{GenCon}} B \to A \equiv B$.

Proof sketch: Left identity coherence

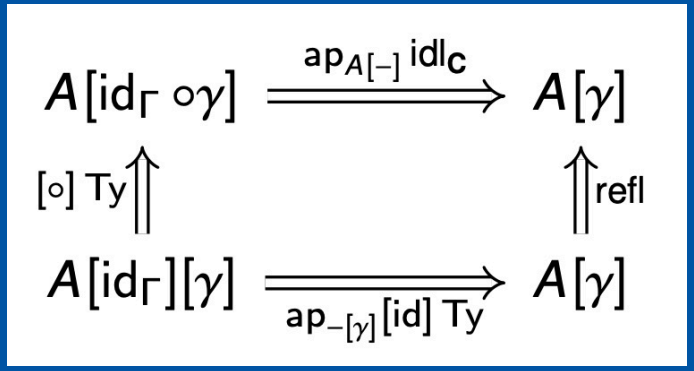
Step 2: Rewrite the square as

where $\mathsf{uaGenCon} : A \simeq_{\mathsf{GenCon}} B \to A \equiv B$.

Step 3: Prove commutativity of the $\simeq_{\mathsf{GenCon}}$ square.

Proof sketch: Left identity coherence

Step 4: Obtain the original square



by massaging equalities.

E.g. Lemma: $\mathsf{uaGenCon}\ \mathrm{id}_{\simeq \mathsf{GenCon}} \equiv \mathsf{refl}$.

Type formers

Σ-types: Similar to presheaf model.

- $[\![A]\!] : \int [\![\Gamma]\!] \to \mathrm{Set}$
- $[\![B]\!] : \int [\![\Gamma \triangleright A]\!] \to \mathrm{Set}$
- $[\![\Sigma A B]\!] : \int [\![\Gamma]\!] \to \mathrm{Set}$

Type formers

Σ-types: Similar to presheaf model.

- $[\![A]\!] : \int [\![\Gamma]\!] \to \mathrm{Set}$
- $[\![B]\!] : \int [\![\Gamma \triangleright A]\!] \to \mathrm{Set}$
- $[\![\Sigma A B]\!] : \int [\![\Gamma]\!] \to \mathrm{Set}$

$$[\![\Sigma A B]\!] (X, x_\Gamma) :=$$
$$\Sigma \left(x_A : [\![A]\!] (X, x_\Gamma) \right) \left([\![B]\!] (X, x_\Gamma, x_A) \right)$$

Type formers

Π-types : Still an open problem.

Type formers

Π-types: Still an open problem.

- Previous approach - unclear how to write result as a container.

Type formers

Π-types: Still an open problem.

- Previous approach - unclear how to write result as a container.
- We think they do not exist in general.

Conclusion & Future work

- Fully formalised container GCwF in Cubical Agda:

  github.com/stefaniatadama/container-gcwf

- Type formers: Π, $\mathcal{U}$, Id.
- Containerified QIIT semantics.

Thank you!