The Category Interpretation of Polarized and Directed Type Theory

Thorsten Altenkirch and Jacob Neumann

HoTT-UF Workshop

23 April 2023

Where to find more detail

- These slides: jacobneu.github.io/research/slides/HoTT-UF-2023.pdf
- A preprint will appear here: jacobneu.github.io/research/preprints/polarTT.pdf
- Agda formalization coming soon (link will be added to preprint and slides)

Univalent Mathematics:

Groupoid Theory

versus

Category Theory

$\infty-$groupoids are easy in HoTT

Recall A type in HoTT can be viewed as a ∞-**groupoid**: the elements are the objects, the identity proofs are the morphisms, . . .

$\infty-$groupoids are easy in HoTT

Recall A type in HoTT can be viewed as a ∞-**groupoid**: the elements are the objects, the identity proofs are the morphisms, ...

A function $f : A \to B$ is automatically a functor w.r.t. this groupoid structure: using the J-rule, we can construct $\mathrm{ap}_f\, p \colon f(a) =_B f(a')$ for each $p : a =_A a'$ and prove this preserves identities (refl) and composition (path concatenation)

Recall A type in HoTT can be viewed as a **∞-groupoid**: the elements are the objects, the identity proofs are the morphisms, ...

A function $f : A \to B$ is automatically a functor w.r.t. this groupoid structure: using the J-rule, we can construct $\mathrm{ap}_f\ p \colon f(a) =_B f(a')$ for each $p : a =_A a'$ and prove this preserves identities (refl) and composition (path concatenation)

Key observation We don't need to inspect the definition of f to define ap_f or to prove it respects identities and composition – once we have f, we have its functoriality

Not so for univalent category theory

Not so for univalent category theory

To define a category, we must define its morphisms explicitly and prove they satisfy the given laws.

Not so for univalent category theory

To define a category, we must define its morphisms explicitly and prove they satisfy the given laws.

To define a functor, we must define its morphism part explicitly and prove functoriality by hand.

To define a category, we must define its morphisms explicitly and prove they satisfy the given laws.

To define a functor, we must define its morphism part explicitly and prove functoriality by hand.

If we want to do ∞-category theory...

Summary: In univalent mathematics, groupoids are *synthetic* but categories are *analytic*

Why?

Moral: We *ought* to do directed type theory

Some Existing Directed TT/Synthetic CT Projects

- Harper and Licata – *2-Dimensional Directed Type Theory* (2011) $\dagger\star$
- Nuyts – *Towards a Directed Homotopy Type Theory based on 4 Kinds of Variance* (2015) $\dagger\star$
- Riehl and Shulman – *A type theory for synthetic ∞-categories* (2017)
- Ahrens, North, and van der Wiede – *Semantics for two-dimensional type theory* (2022) $\star$
- Cisinski, Nguyen, and Walde – *Univalent Directed Type Theory* (2023)

$\dagger$ No model theory

$\star$ Includes a directed version of *judgmental equality*

Contribution

Contribution

Directed TT
using CwFs

Contribution

Directed TT
using CwFs

Deep
Polarity

Back in the 90s. . .

Back in the 90s...

Our goal is to develop directed type theory following in the tradition of several landmark papers from the 1990s that paved the way for homotopy type theory:

Back in the 90s. . .

Our goal is to develop directed type theory following in the tradition of several landmark papers from the 1990s that paved the way for homotopy type theory:

- Dybjer's *Internal Type Theory* (1995)
  - Introduced **categories with families** as a model theory for type theory

Back in the 90s...

Our goal is to develop directed type theory following in the tradition of several landmark papers from the 1990s that paved the way for homotopy type theory:

- Dybjer's *Internal Type Theory* (1995)
  - Introduced **categories with families** as a model theory for type theory
  - Generalized algebraic theory – more convenient to formalize in a computer proof assistant

Back in the 90s. . .

Our goal is to develop directed type theory following in the tradition of several landmark papers from the 1990s that paved the way for homotopy type theory:

- Dybjer's *Internal Type Theory* (1995)
  - Introduced **categories with families** as a model theory for type theory
  - Generalized algebraic theory – more convenient to formalize in a computer proof assistant
- Hofmann and Streicher's *The Groupoid Interpretation of Type Theory* (1995)

Back in the 90s. . .

Our goal is to develop directed type theory following in the tradition of several landmark papers from the 1990s that paved the way for homotopy type theory:

- Dybjer's *Internal Type Theory* (1995)
  - Introduced **categories with families** as a model theory for type theory
  - Generalized algebraic theory – more convenient to formalize in a computer proof assistant
- Hofmann and Streicher's *The Groupoid Interpretation of Type Theory* (1995)
  - Introduced the **groupoid model** of type theory, a CwF structure on the category of groupoids

Back in the 90s...

Our goal is to develop directed type theory following in the tradition of several landmark papers from the 1990s that paved the way for homotopy type theory:

- Dybjer's *Internal Type Theory* (1995)
  - Introduced **categories with families** as a model theory for type theory
  - Generalized algebraic theory – more convenient to formalize in a computer proof assistant
- Hofmann and Streicher's *The Groupoid Interpretation of Type Theory* (1995)
  - Introduced the **groupoid model** of type theory, a CwF structure on the category of groupoids
  - Proved the independence of the *Uniqueness of Identity Proofs*

Next up in the *Directed CwFs Transatlantic Tour*

At the HoTT Conference (May 2023, Pittsburgh, USA), we'll present presheaf semantics for directed type theory, a directed analogue of these works:

- Hofmann and Streicher's *Lifting Grothendieck Universes* (1999, unpublished)
  - Established a technique for modelling universes in *presheaf models* of type theory
- Hofmann's *Semantical analysis of higher-order abstract syntax* (1999)
  - Gave presheaf semantics for a **higher-order abstract syntax**, which abstracts away cumbersome details about substitution and binding

Categories with Families

Defn. A **category with families (CwF)** is a (generalized) algebraic structure, consisting of:

- A category Con of *contexts* and *substitutions*, with a terminal object $\bullet$, the *empty context*
- A presheaf $\mathrm{Ty} : \mathrm{Con}^{\mathrm{op}} \to \mathrm{Set}$ of *types*
- A presheaf $\mathrm{Tm} : (\int \mathrm{Ty})^{\mathrm{op}} \to \mathrm{Set}$ of *terms*
- An operation of *context extension*:

$$\frac{\Gamma : \mathrm{Con} \quad A : \mathrm{Ty}\,\Gamma}{\Gamma \triangleright A : \mathrm{Con}}$$

so that $\Gamma \triangleright A$ is a 'locally representing object' (in the sense spelled out on the next slide)

The Local Representability Condition

For any Δ, Γ and any $A : \mathsf{Ty}\,\Gamma$,

$$\mathsf{Con}(\Delta, \Gamma \triangleright A) \quad \cong \quad \sum_{\gamma : \mathsf{Con}(\Delta,\Gamma)} \mathsf{Tm}(\Delta, A[\gamma])$$

natural in Δ.

The **groupoid model of type theory** is a CwF

The **groupoid model of type theory** is a CwF where

- Con is the category of groupoids

The **groupoid model of type theory** is a CwF where

- Con is the category of groupoids
- Ty Γ is the set of Γ-indexed families of groupoids (i.e. functors $\Gamma \to \mathrm{Grpd}$)
- ...

The **groupoid model of type theory** is a CwF where

- Con is the category of groupoids
- Ty Γ is the set of Γ-indexed families of groupoids (i.e. functors $\Gamma \to \mathrm{Grpd}$)
- ...

Further structure Can interpret dependent types and identity types in the groupoid model, and find types whose identity types violate UIP

Main Idea: Replace groupoids with categories!

The category model of type theory is a CwF where

- Con is the category of **categories**
- Ty Γ is the set of Γ-indexed families of **categories** (i.e. functors $\Gamma \to \mathsf{Cat}$)
- ...

The category model of type theory is a CwF where

- Con is the category of **categories**
- Ty Γ is the set of Γ-indexed families of **categories** (i.e. functors $\Gamma \to \mathsf{Cat}$)
- ...

Further structure

The category model of type theory is a CwF where

- Con is the category of **categories**
- Ty Γ is the set of Γ-indexed families of **categories** (i.e. functors $\Gamma \to \mathsf{Cat}$)
- ...

Further structure The category of categories comes equipped with the **opposite category** operation, which we can view as a functor $\mathsf{Cat} \to \mathsf{Cat}$.

The category model of type theory is a CwF where

- Con is the category of **categories**
- Ty Γ is the set of Γ-indexed families of **categories** (i.e. functors $\Gamma \to \mathsf{Cat}$)
- ...

Further structure The category of categories comes equipped with the **opposite category** operation, which we can view as a functor $\mathsf{Cat} \to \mathsf{Cat}$.

- For each context Γ, there is a context Γ^-

The category model of type theory is a CwF where

- Con is the category of **categories**
- Ty Γ is the set of Γ-indexed families of **categories** (i.e. functors $\Gamma \to \mathsf{Cat}$)
- ...

Further structure The category of categories comes equipped with the **opposite category** operation, which we can view as a functor $\mathsf{Cat} \to \mathsf{Cat}$.

- For each context Γ, there is a context Γ^-
- For each $A : \mathrm{Ty}\ \Gamma$, there is a type $A^- : \mathrm{Ty}\ \Gamma$

A **polarized category with families (PCwF)** is a (generalized) algebraic structure, consisting of:

Polarized Categories with Families

A **polarized category with families (PCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF

A **polarized category with families (PCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF
- A functor $(\_)^{-} : \mathrm{Con} \to \mathrm{Con}$ such that $(\Gamma^{-})^{-} = \Gamma$ and $\bullet^{-} = \bullet$

A **polarized category with families (PCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF
- A functor $(\_)^{-}$: Con $\rightarrow$ Con such that $(\Gamma^{-})^{-} = \Gamma$ and $\bullet^{-} = \bullet$
- For each Γ : Con, a function $(\_)^{-}$: Ty $\Gamma \rightarrow$ Ty Γ such that $(A^{-})^{-} = A$

Polarized Categories with Families

A **polarized category with families (PCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF
- A functor $(\_)^-$: Con $\to$ Con such that $(\Gamma^-)^- = \Gamma$ and $\bullet^- = \bullet$
- For each Γ: Con, a function $(\_)^-$: Ty $\Gamma \to$ Ty Γ such that $(A^-)^- = A$
- Two operations of *context extension*: for s either $+$ or $-$,

$$\frac{\Gamma : \mathrm{Con} \quad A : \mathrm{Ty}\ \Gamma^s}{\Gamma \rhd^s A : \mathrm{Con}}$$

The Local Representability Conditions

For any Δ, Γ and any $A : \mathsf{Ty}\,\Gamma$,

$$\mathsf{Con}(\Delta, \Gamma \triangleright^s A) \quad \cong \quad \sum_{\gamma : \mathsf{Con}(\Delta, \Gamma)} \mathsf{Tm}(\Delta^s, A[\gamma^s]^s)$$

natural in Δ.

Further structure

Hom Types

Further structure In the groupoid model, we were able to interpret *identity types*. In the category model, we have **hom types**

Hom Types

Further structure In the groupoid model, we were able to interpret *identity types*. In the category model, we have **hom types**.

$$\frac{A : \mathrm{Ty}\,\Gamma \qquad a_0 : \mathrm{Tm}(\Gamma, A^-) \qquad a_1 : \mathrm{Tm}(\Gamma, A)}{a_0 \Rightarrow_A a_1 : \mathrm{Ty}\,\Gamma}$$

Hom Types

Further structure In the groupoid model, we were able to interpret *identity types*. In the category model, we have **hom types**.

$$\frac{A : \mathrm{Ty}\,\Gamma \quad a_0 : \mathrm{Tm}(\Gamma, A^-) \quad a_1 : \mathrm{Tm}(\Gamma, A)}{a_0 \Rightarrow_A a_1 : \mathrm{Ty}\,\Gamma}$$

Note the use of polarities to mark variances!

Further structure In the groupoid model, we were able to interpret *identity types*. In the category model, we have **hom types**.

$$\frac{A : \mathrm{Ty}\,\Gamma \quad a_0 : \mathrm{Tm}(\Gamma, A^-) \quad a_1 : \mathrm{Tm}(\Gamma, A)}{a_0 \Rightarrow_A a_1 : \mathrm{Ty}\,\Gamma}$$

Note the use of polarities to mark variances!

Notice This is the essential ingredient in making our types into **synthetic categories**.

Further structure

Core Types

Further structure The groupoid model also 'lives inside' the category model: we can take the **core** of a category $\mathbb{C}$, which is the largest groupoid that is a subcategory of $\mathbb{C}$ (and of $\mathbb{C}^{op}$).

Core Types

Further structure The groupoid model also ‘lives inside’ the category model: we can take the **core** of a category $\mathbb{C}$, which is the largest groupoid that is a subcategory of $\mathbb{C}$ (and of $\mathbb{C}^{\mathrm{op}}$). We could perhaps treat this as an operation on contexts, but we’re mainly interested in it at the type level:

$$\frac{A : \mathrm{Ty}\,\Gamma}{A^0 : \mathrm{Ty}\,\Gamma}$$

Further structure The groupoid model also 'lives inside' the category model: we can take the **core** of a category $\mathbb{C}$, which is the largest groupoid that is a subcategory of $\mathbb{C}$ (and of $\mathbb{C}^{\mathrm{op}}$). We could perhaps treat this as an operation on contexts, but we're mainly interested in it at the type level:

$$\frac{A : \mathrm{Ty}\,\Gamma}{A^0 : \mathrm{Ty}\,\Gamma} \qquad \frac{a : \mathrm{Tm}(\Gamma, A^0)}{+a : \mathrm{Tm}(\Gamma, A) \qquad -a : \mathrm{Tm}(\Gamma, A^-)}$$

Refl and J

Core types allow us to state the **introduction rule** for hom types:

$$\frac{a : \mathrm{Tm}(\Gamma, A^0)}{\mathrm{refl}_a : \mathrm{Tm}(\Gamma, -a \Rightarrow_A +a)}$$

Refl and J

Core types allow us to state the **introduction rule** for hom types:

$$\frac{a : \mathrm{Tm}(\Gamma, A^0)}{\mathrm{refl}_a : \mathrm{Tm}(\Gamma, -a \Rightarrow_A +a)}$$

as well as the appropriate **J-rules**: for any $a' : \mathrm{Tm}(\Gamma, A^0)$

$$\frac{m : \mathrm{Tm}(\Gamma, M(+a', \mathrm{refl}_{a'})) \quad a'' : \mathrm{Tm}(\Gamma, A) \quad q : \mathrm{Tm}(\Gamma, -a' \Rightarrow a'')}{J_M^+\ m\ q : \mathrm{Tm}(\Gamma, M(a'', q))}$$

$$\frac{n : \mathrm{Tm}(\Gamma, N(-a', \mathrm{refl}_{a'})) \quad a : \mathrm{Tm}(\Gamma, A^-) \quad p : \mathrm{Tm}(\Gamma, a \Rightarrow +a')}{J_N^-\ n\ p : \mathrm{Tm}(\Gamma, N(a, p))}$$

Proof of concept: Composition

Given

- $x : \mathrm{Tm}(\Gamma, A^{-})$
- $y : \mathrm{Tm}(\Gamma, A^{0})$
- $z : \mathrm{Tm}(\Gamma, A)$

- $f : \mathrm{Tm}(\Gamma, x \Rightarrow +y)$
- $g : \mathrm{Tm}(\Gamma, -y \Rightarrow z)$

Proof of concept: Composition

Given

- $x : \mathrm{Tm}(\Gamma, A^-)$
- $y : \mathrm{Tm}(\Gamma, A^0)$
- $z : \mathrm{Tm}(\Gamma, A)$

- $f : \mathrm{Tm}(\Gamma, x \Rightarrow +y)$
- $g : \mathrm{Tm}(\Gamma, -y \Rightarrow z)$

Define $f \cdot g : \mathrm{Tm}(\Gamma, x \Rightarrow z)$ as either

$$J^+_M\ f\ g \qquad \text{or} \qquad J^-_N\ g\ f$$

where

$$M(a'', q) :\equiv x \Rightarrow a'' \qquad \text{and} \qquad N(a, p) :\equiv a \Rightarrow z$$

Directed Categories with Families

A **directed category with families (DCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF
- The negation operations $(\_)^{-}$ and context extensions $\triangleright^{s}$ as in the definition of PCwF

A **directed category with families (DCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF
- The negation operations $(\_)^{-}$ and context extensions $\triangleright^{s}$ as in the definition of PCwF
- Core types and the $+$ and $-$ operations on terms

Directed Categories with Families

A **directed category with families (DCwF)** is a (generalized) algebraic structure, consisting of:

- Con, $\bullet$, Ty, Tm as in the definition of CwF
- The negation operations $(\_)^{-}$ and context extensions $\triangleright^{s}$ as in the definition of PCwF
- Core types and the $+$ and $-$ operations on terms
- The $\_ \Rightarrow \_$ type former with refl constructor and J eliminators

Thank you!