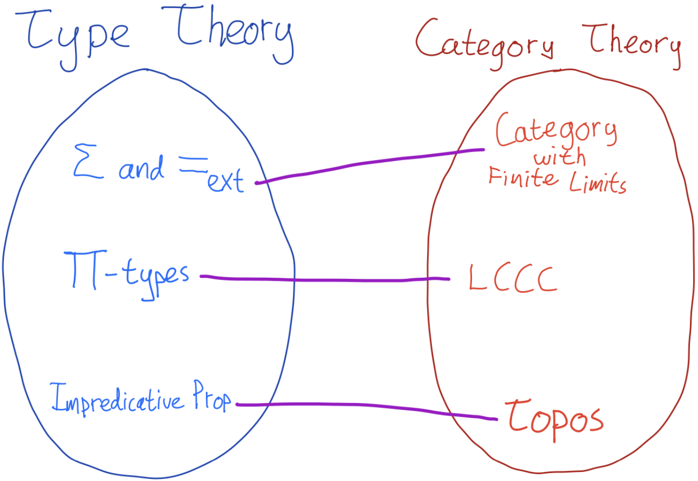
The Internal Languages of Univalent Categories

Niels van der Weide

November 21, 2024

Type Theory and Category Theory



Internal Language Theorems

Theorem (Theorem 6.1 in Clairambault&Dybjer 2014[1])

We have a biequivalence between the bicategories

- $\mathsf{CwF}_{\mathsf{dem}}^{\sum,=_{\mathsf{ext}}}$*: democratic categories with families with extensional identity types and sigma types*
- FinLim*: finitely complete categories*

This biequivalence can be extended to $\prod$*-types and LCCCs*

[1] "The biequivalence of locally cartesian closed categories and Martin-Löf type theories", by Clairambault and Dybjer

Internal Language Theorems

Theorem (Theorem 6.1 in Clairambault&Dybjer 2014[1])

We have a **biequivalence** *between the bicategories*

- $\mathsf{CwF}_{\mathsf{dem}}^{\sum,=_{\mathsf{ext}}}$*: democratic categories with families with extensional identity types and sigma types*
- FinLim*: finitely complete categories*

This biequivalence can be extended to $\prod$*-types and LCCCs*

[1] "The biequivalence of locally cartesian closed categories and Martin-Löf type theories", by Clairambault and Dybjer

Internal Language Up To Isomorphism

Final sentence of the paper by Clairambault and Dybjer:

> *So we can ask whether Martin-Löf type theory with extensional identity types, $\sum$- and $\prod$-types is an internal language for lcccs? And we can answer, yes, it is an internal language 'up to isomorphism'.*

Internal Language **Up To Isomorphism**

Final sentence of the paper by Clairambault and Dybjer:

> *So we can ask whether Martin-Löf type theory with extensional identity types, $\sum$- and $\prod$-types is an internal language for lcccs? And we can answer, yes, it is an internal language '***up to isomorphism***'.*

Univalent Categories

Definition

A category C is **univalent** if the canonical map $x = y \to x \cong y$ is an equivalence for all objects $x, y : \mathsf{C}$.

Univalent Categories

Definition

A category C is **univalent** if the canonical map $x = y \to x \cong y$ is an equivalence for all objects $x, y : \mathsf{C}$.

Key theorem about univalent categories: if C_1 and C_2 are univalent, then the canonical map from identities $\mathsf{C}_1 = \mathsf{C}_2$ to adjoint equivalence $\mathsf{C}_1 \simeq \mathsf{C}_2$ is an equivalence of types.

Univalent Categories

Definition

A category C is **univalent** if the canonical map $x = y \to x \cong y$ is an equivalence for all objects x, y : C.

Key theorem about univalent categories: if C_1 and C_2 are univalent, then the canonical map from identities $C_1 = C_2$ to adjoint equivalence $C_1 \simeq C_2$ is an equivalence of types.
So:

- Two objects x and y in a univalent category have the same properties if they are isomorphic
- Two univalent categories have the same properties if they are equivalent.

Internal Language Theorems for Univalent Categories

Theorem (Theorem of Today)

We have a biequivalence between the bicategories

- DFLCompCat*: democratic univalent full comprehension categories with sigma types, equalizer types, binary product types, and unit types*
- FinLim*: univalent finitely complete categories*

This biequivalence can be extended to LCCCs, pretoposes, arithmetic pretoposes, $\prod$-pretoposes, elementary toposes, and elementary toposes with $\mathbb{N}$.

This is formalized in UniMath[2], and there is a preprint: `https://arxiv.org/abs/2411.06636`

[2] `https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/ComprehensionCat`

Internal Language Theorems for Univalent Categories

Theorem (Theorem of Today)

We have a biequivalence between the bicategories

- DFLCompCat*: democratic* **univalent** *full comprehension categories with sigma types, equalizer types, binary product types, and unit types*
- FinLim*:* **univalent** *finitely complete categories*

This biequivalence can be extended to LCCCs, pretoposes, arithmetic pretoposes, $\prod$-pretoposes, elementary toposes, and elementary toposes with $\mathbb{N}$.

This is formalized in UniMath[2], and there is a preprint: `https://arxiv.org/abs/2411.06636`

[2] `https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/ComprehensionCat`

Internal Language Theorems for Univalent Categories

Theorem (Theorem of Today)

We have a biequivalence between the bicategories

- DFLCompCat*: democratic* **univalent full comprehension categories** *with sigma types, equalizer types, binary product types, and unit types*
- FinLim*:* **univalent** *finitely complete categories*

This biequivalence can be extended to LCCCs, pretoposes, arithmetic pretoposes, $\prod$-pretoposes, elementary toposes, and elementary toposes with $\mathbb{N}$.

This is formalized in UniMath[2], and there is a preprint: https://arxiv.org/abs/2411.06636

[2] https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/ComprehensionCat

Internal Language Theorems for Univalent Categories

Theorem (Theorem of Today)

We have a biequivalence between the bicategories

- DFLCompCat*: democratic* **univalent full comprehension categories** *with sigma types,* **equalizer types, binary product types, and unit types**
- FinLim*:* **univalent** *finitely complete categories*

This biequivalence can be extended to LCCCs, pretoposes, arithmetic pretoposes, $\prod$-pretoposes, elementary toposes, and elementary toposes with $\mathbb{N}$.

This is formalized in UniMath[2], and there is a preprint: https://arxiv.org/abs/2411.06636

[2] https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/ComprehensionCat

Internal Language Theorems for Univalent Categories

Theorem (Theorem of Today)

We have a biequivalence between the bicategories

- DFLCompCat*: democratic* **univalent full comprehension categories** *with sigma types,* **equalizer types, binary product types, and unit types**
- FinLim*:* **univalent** *finitely complete categories*

This biequivalence can be extended to LCCCs, **pretoposes, arithmetic pretoposes, $\prod$-pretoposes, elementary toposes, and elementary toposes with** $\mathbb{N}$*.*

This is formalized in UniMath[2], and there is a preprint: https://arxiv.org/abs/2411.06636

[2] https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/ComprehensionCat

Today's talk

We shall study the internal language theorem by Clairambault and Dybjer for univalent categories, and we discuss extensions.

Today's talk

We shall study the internal language theorem by Clairambault and Dybjer for univalent categories, and we discuss extensions. More specifically,

- We argue why comprehension categories are more suitable for interpreting dependent types than CwFs in UF
- We recall the interpretation of type formers in comprehension categories and the notion of democracy
- We show that DFLCompCat and FinLim are biequivalent
- We discuss how to modularly extend this theorem using displayed biequivalences

Throughout the talk, we highlight how univalence affects the development

Introduction

Why Comprehension Categories?

The Bicategory of Comprehension Categories

The Biequivalence for Categories with Finite Limits

Extensions

Conclusion

Introduction

Why Comprehension Categories?

The Bicategory of Comprehension Categories

The Biequivalence for Categories with Finite Limits

Extensions

Conclusion

Categories with Families

Definition

A **category with families**[3] is given by

- a category C
- a functor $T : \mathsf{C} \to \mathsf{Fam}$, which sends every $\Gamma : \mathsf{C}$ to a set $\mathsf{Ty}(\Gamma)$ and a family $\mathsf{Tm}_\Gamma : \mathsf{Ty}(\Gamma) \to \mathsf{Set}$

together with a representing object for the functor sending $f : \Delta \to \Gamma$ to $\mathsf{Tm}_\Gamma(f^*(A))$ for all $\Gamma : \mathsf{C}$ and $A : \mathsf{Ty}(\Gamma)$.

Recall: the objects of Fam are given by sets A together with a family B of sets indexed by A.

[3] "Internal type theory", Dybjer

Categories with Families

Definition

A **category with families**[3] is given by

- a category C
- a functor $T : \mathsf{C} \to \mathsf{Fam}$, which sends every $\Gamma : \mathsf{C}$ to a set $\mathsf{Ty}(\Gamma)$ and a family $\mathsf{Tm}_\Gamma : \mathsf{Ty}(\Gamma) \to \mathsf{Set}$

together with a representing object for the functor sending $f : \Delta \to \Gamma$ to $\mathsf{Tm}_\Gamma(f^*(A))$ for all $\Gamma : \mathsf{C}$ and $A : \mathsf{Ty}(\Gamma)$.

Recall: the objects of Fam are given by sets A together with a family B of sets indexed by A.

[3] "Internal type theory", Dybjer

Categories with Families

Definition

A **category with families**[3] is given by

- a category C
- a functor $T : \mathrm{C} \to \mathrm{Fam}$, which sends every $\Gamma : \mathrm{C}$ to a **set** $\mathrm{Ty}(\Gamma)$ and a family $\mathrm{Tm}_\Gamma : \mathrm{Ty}(\Gamma) \to \mathrm{Set}$

together with a representing object for the functor sending $f : \Delta \to \Gamma$ to $\mathrm{Tm}_\Gamma(f^*(A))$ for all $\Gamma : \mathrm{C}$ and $A : \mathrm{Ty}(\Gamma)$.

Recall: the objects of Fam are given by sets A together with a family B of sets indexed by A.

[3] "Internal type theory", Dybjer

The Set Model

However, we do not have the following CwF

- Contexts are hSets
- Types in context Γ are equivalent to families $\Gamma \to \mathsf{Set}$
- Terms are sections of such families

This is because the type of all sets is not a set itself.

Models from Univalent Categories

Given a univalent category C with finite limits, **we do not have the following CwF**

- Contexts are objects in C
- Types in context Γ are equivalent to pairs of an object $\Gamma.A$ and a morphism $\Gamma.A \to \Gamma$
- Terms are sections

This is because the type of objects in a univalent category generally only forms a **1-type**.

Discrete Models for Dependent Types

Various categorical structures to interpret type theory are built around **discrete fibrations** (aka presheaves).

- Categories with attributes[4]
- Categories with families[5]
- Natural models[6]

In such models, the types in each context must form a set. This requirement is not satisfied by the objects of univalent categories.

[4] "Generalised Algebraic Theories and Contextual Categories", Cartmell

[5] "Internal type theory", Dybjer

[6] "Natural models of homotopy type theory", Awodey

Rejecting Discreteness

There also are categorical structures to interpret type theory built around general **fibrations** (aka pseudofunctors into Cat).

- (Full) comprehension categories[7]
- Judgmental theories/Generalized CwFs[8] [9]
- Coherent CwFs[10]

[7] "Categorical Logic and Type Theory", Jacobs

[8] "Context, Judgement, Deduction", Coraglia and Di Liberti

[9] "A 2-categorical analysis of context comprehension", Coraglia and Emmenegger

[10] https://types2024.itu.dk/slides/Thorsten%20Altenkirch%20-%20Coherent%20Categories%20with%20Families.pdf

Rejecting Discreteness

There also are categorical structures to interpret type theory built around general **fibrations** (aka pseudofunctors into Cat).

- (Full) comprehension categories[7]
- Judgmental theories/Generalized CwFs[8] [9]
- Coherent CwFs[10]

In such structures, types are not required to be a set
Note: we do not assume splitness. Since we focus on univalent categories, the necessary identities for substitution hold in general

[7] "Categorical Logic and Type Theory", Jacobs

[8] "Context, Judgement, Deduction", Coraglia and Di Liberti

[9] "A 2-categorical analysis of context comprehension", Coraglia and Emmenegger

[10] `https://types2024.itu.dk/slides/Thorsten%20Altenkirch%20-%20Coherent%20Categories%20with%20Families.pdf`

Intermezzo: Accepting Discreteness

Note: one could also use iterative sets intead of hSets. Since iterative sets form a set, we can use them to construct a CwF representing the set model of type theory[11]

[11] "The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations", Gratzer, Gylterud, Mörtberg, and Stenholm

Intermezzo: Accepting Discreteness

Note: one could also use iterative sets intead of hSets. Since iterative sets form a set, we can use them to construct a CwF representing the set model of type theory[11]
However, our focus is on models that arise from univalent categories

[11] "The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations", Gratzer, Gylterud, Mörtberg, and Stenholm

Fibrations in UF: Displayed Categories

To define fibrations, we use **displayed categories**[12]

Definition

A **displayed category** D over C consists of

- For each x : C a type D_x of objects over x
- For each $f : x \to y$, $\overline{x} : \mathsf{D}_x$, and $\overline{y} : \mathsf{D}_y$ a set $\overline{x} \to_f \overline{y}$ of morphisms over f from $\overline{x}$ to $\overline{y}$

with suitable identity and composition operations.

[12] "Displayed Categories", Ahrens and Lumsdaine

Fibrations in UF: Displayed Categories

To define fibrations, we use **displayed categories**[12]

Definition

A **displayed category** D over C consists of

- For each $x : \mathsf{C}$ a type D_x of objects over x
- For each $f : x \to y$, $\overline{x} : \mathsf{D}_x$, and $\overline{y} : \mathsf{D}_y$ a set $\overline{x} \to_f \overline{y}$ of morphisms over f from $\overline{x}$ to $\overline{y}$

with suitable identity and composition operations.

Note:

- We can use displayed categories to define fibrations
- Analogously, we can define the notions of displayed functor and displayed natural transformation

[12] "Displayed Categories", Ahrens and Lumsdaine

Comprehension Categories

Definition

A **univalent full comprehension category** is given by

- a univalent category C with a terminal object $\langle\rangle$
- a univalent displayed category D over C
- a cleaving for D
- a displayed functor $\chi : \mathsf{D} \to \mathsf{C}^{\to}$ over the identity (here $\mathsf{C}^{\to}$ is the displayed category representing the arrow category)

such that χ is fully faithful and χ preserves Cartesian morphisms.

Pictorially, we represent this data as follows

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

Example

We have the following univalent full comprehension category

$$\begin{tikzcd}
\mathsf{Set}^{\to} \arrow[rr, "\mathsf{id}"] & & \mathsf{Set}^{\to} \\
& \mathsf{Set} &
\end{tikzcd}$$

Introduction

Why Comprehension Categories?

The Bicategory of Comprehension Categories

The Biequivalence for Categories with Finite Limits

Extensions

Conclusion

Type Formers in the Internal Language

Clairambault and Dybjer use the following type formers

- Extensional identity types
- Dependent sums (with the η-rule)

They also assume democracy (discussed after 2 slides)
From these, one can derive unit types, binary product types, and equalizer types

Type Formers in Comprehension Categories

We use the following type formers:

- Binary product types: fiberwise binary products

[13] Theorem 10.5.10 in “Categorical Logic and Type Theory” by Jacobs

Type Formers in Comprehension Categories

We use the following type formers:

- Binary product types: fiberwise binary products
- Extensional identity types: left adjoints to contraction, satisfying Beck-Chevalley and reflection. **We use fiberwise equalizers, which are equivalent if there are $\sum$-types**[13]

[13] Theorem 10.5.10 in "Categorical Logic and Type Theory" by Jacobs

Type Formers in Comprehension Categories

We use the following type formers:

- Binary product types: fiberwise binary products
- Extensional identity types: left adjoints to contraction, satisfying Beck-Chevalley and reflection. **We use fiberwise equalizers, which are equivalent if there are $\sum$-types**[13]
- Unit types: fiberwise terminal objects and χ preserves terminal objects

[13] Theorem 10.5.10 in "Categorical Logic and Type Theory" by Jacobs

Type Formers in Comprehension Categories

We use the following type formers:

- Binary product types: fiberwise binary products
- Extensional identity types: left adjoints to contraction, satisfying Beck-Chevalley and reflection. **We use fiberwise equalizers, which are equivalent if there are $\sum$-types**[13]
- Unit types: fiberwise terminal objects and χ preserves terminal objects
- Dependent sums: left adjoints to weakening, satisfying Beck-Chevalley
- **Note: we require a strongness assumption to interpret the η-rule for $\sum$-types**. This assumption says that $\Gamma.A.B \cong \Gamma.\sum_{a:A} B\ a$

[13] Theorem 10.5.10 in "Categorical Logic and Type Theory" by Jacobs

Democratic Comprehension Categories

Definition

Suppose, we have a comprehension category as follows.

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

This comprehension is **democratic** if for every $\Gamma : \mathsf{C}$ there is $A : \mathsf{D}[\langle\rangle]$ and an isomorphism $\Gamma \cong \langle\rangle.A$ (domain of $\chi(A)$).

Democratic Comprehension Categories

Definition

Suppose, we have a comprehension category as follows.

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

This comprehension is **democratic** if for every $\Gamma : \mathsf{C}$ there is $A : \mathsf{D}[\langle\rangle]$ and an isomorphism $\Gamma \cong \langle\rangle.A$ (domain of $\chi(A)$).

A **democratic finite limit comprehension category** is a democratic univalent full comprehension category with

- unit types
- binary product types
- equalizer types
- dependent sums

Morphisms of Comprehension Categories

Morphisms from $\chi_1 : \mathsf{D}_1 \to \mathsf{C}_1^{\to}$ to $\chi_2 : \mathsf{D}_2 \to \mathsf{C}_2^{\to}$ are given by functors $\mathsf{F} : \mathsf{C}_1 \to \mathsf{C}_2$ and displayed functor $\overline{\mathsf{F}} : \mathsf{D}_1 \to \mathsf{D}_2$ over F together with a displayed **natural isomorphism** over the identity as follows

$$\begin{tikzcd}
\mathsf{D}_1 \arrow[r, "\chi_1"] \arrow[d, "\overline{\mathsf{F}}"'] \arrow[dr, phantom, "\Downarrow \cong"] & \mathsf{C}_1^{\to} \arrow[d, "\mathsf{F}^{\to}"] \\
\mathsf{D}_2 \arrow[r, "\chi_2"'] & \mathsf{C}_2^{\to}
\end{tikzcd}$$

such that F preserves fiberwise terminal objects, fiberwise binary products, and fiberwise equalizers.

Morphisms of Comprehension Categories

Morphisms from $\chi_1 : \mathsf{D}_1 \to \mathsf{C}_1^{\to}$ to $\chi_2 : \mathsf{D}_2 \to \mathsf{C}_2^{\to}$ are given by functors $\mathsf{F} : \mathsf{C}_1 \to \mathsf{C}_2$ and displayed functor $\overline{\mathsf{F}} : \mathsf{D}_1 \to \mathsf{D}_2$ over F together with a displayed **natural isomorphism** over the identity as follows

$$\begin{tikzcd}
\mathsf{D}_1 \arrow[r, "\chi_1"] \arrow[d, "\overline{\mathsf{F}}"'] \arrow[dr, phantom, "\Downarrow \cong"] & \mathsf{C}_1^{\to} \arrow[d, "\mathsf{F}^{\to}"] \\
\mathsf{D}_2 \arrow[r, "\chi_2"'] & \mathsf{C}_2^{\to}
\end{tikzcd}$$

such that F preserves fiberwise terminal objects, fiberwise binary products, and fiberwise equalizers.

Note: we can derive that

- dependent sums are preserved (see Clairambault and Dybjer)
- democracy is preserved (if you want to know, you can ask after the talk)

The Bicategory of Comprehension Categories

Definition

We have a univalent bicategory DFLCompCat such that

- Objects are democratic finite limit comprehension categories
- Morphisms are pseudomorphisms as discussed on the previous slide
- 2-cells: if you want to know, ask a question after the talk

Theorem

DFLCompCat *is univalent*

Introduction

Why Comprehension Categories?

The Bicategory of Comprehension Categories

The Biequivalence for Categories with Finite Limits

Extensions

Conclusion

Univalent Categories with Finite Limits

Theorem

We have a biequivalence between FinLim *and* DFLCompCat*.*

Here is FinLim the bicategory whose

- objects are univalent categories with finite limits,
- 1-cells are functors that preserve finite limits
- 2-cells are natural transformations

Overview of the Biequivalence

We must construct:

- a pseudofunctor $\mathsf{H} : \mathsf{FinLim} \to \mathsf{DFLCompCat}$
- a pseudofunctor $\mathsf{U} : \mathsf{DFLCompCat} \to \mathsf{FinLim}$
- a pseudotransformation $\xi : \mathsf{H} \cdot \mathsf{U} \Rightarrow \mathsf{id}_{\mathsf{FinLim}}$
- a pseudotransformation $\zeta : \mathsf{id}_{\mathsf{DFLCompCat}} \Rightarrow \mathsf{U} \cdot \mathsf{H}$

and we must prove that ξ and ζ are pointwise adjoint equivalences.

Construction of H : FinLim → DFLCompCat

Every univalent category with finite limits gives rise to the following DFL comprehension category

$$\begin{tikzcd}
\mathsf{C}^{\to} \arrow[rr, "\mathsf{id}"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

Note:

- one can show that this comprehension category has all necessary type formers
- one can show that this is suitably pseudofunctorial (matter of having some willpower)

Short Intermezzo: Splitting the Fibration

In set theory, one also need to replace the fibration by a split one:

$$\begin{tikzcd}
\mathsf{Split}(\mathsf{C}^{\rightarrow}) \arrow[r, "\simeq"] & \mathsf{C}^{\rightarrow} \arrow[r, "\mathsf{id}"] & \mathsf{C}^{\rightarrow} \\
 & \mathsf{C} &
\end{tikzcd}$$

Construction of U : DFLCompCat → FinLim

Every DFL comprehension category

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

gives rise to a category

Construction of U : DFLCompCat $\to$ FinLim

Every DFL comprehension category

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

gives rise to a category C

Construction of U : DFLCompCat → FinLim

Every DFL comprehension category

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\rightarrow} \\
& \mathsf{C} &
\end{tikzcd}$$

gives rise to a category C
The challenge lies in verifying that C is finitely complete.

Construction of U : DFLCompCat → FinLim

Key lemma:

Lemma

If we have a

$$\begin{tikzcd} \mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\rightarrow} \\ & \mathsf{C} & \end{tikzcd}$$

then the functor χ is fiberwise (split) essentially surjective.

Note: we assumed that χ is fully faithful

Construction of U : DFLCompCat → FinLim

Key lemma:

Lemma

If we have a

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

then the functor χ is fiberwise (split) essentially surjective.

Note: we assumed that χ is fully faithful

This lemma gives us an equivalence from D to $\mathsf{C}^{\to}$, and univalence gives us the relevant transport principles

In particular, we can transport structure/properties from fiber categories $\mathsf{D}[x]$ to slice categories C/x.

Construction of $\mathsf{U} : \mathsf{DFLCompCat} \to \mathsf{FinLim}$

Key lemma:

Lemma

If we have a

$$\begin{tikzcd}
\mathsf{D} \arrow[rr, "\chi"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

then the functor χ is fiberwise (split) essentially surjective.

Note: we assumed that χ is fully faithful

This lemma gives us an equivalence from D to $\mathsf{C}^{\to}$, and univalence gives us the relevant transport principles

In particular, we can transport structure/properties from fiber categories $\mathsf{D}[x]$ to slice categories C/x.

Since $\mathsf{D}[\langle\rangle]$ is finitely complete, so is C.

Construction of ξ

If C is a univalent category with finite limits, then U(H(C)) is C.
So: for ξ we take the pointwise identity.

Construction of ζ

Starting with $\chi : \mathsf{D} \to \mathsf{C}^{\to}$, we obtain the following DFL comprehension category

$$\begin{tikzcd}
\mathsf{C}^{\to} \arrow[rr, "\mathsf{id}"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

Construction of ζ

Starting with $\chi : \mathsf{D} \to \mathsf{C}^{\to}$, we obtain the following DFL comprehension category

$$\begin{tikzcd}
\mathsf{C}^{\to} \arrow[rr, "\mathsf{id}"] & & \mathsf{C}^{\to} \\
& \mathsf{C} &
\end{tikzcd}$$

For ζ we use the following adjoint equivalence of DFL comprehension categories:

$$\begin{tikzcd}
\mathsf{D} \arrow[r, "\chi"] \arrow[d, "\chi"'] & \mathsf{C}^{\to} \arrow[d, "\mathsf{id}^{\to}"] \\
\mathsf{C}^{\to} \arrow[r, "\mathsf{id}"'] & \mathsf{C}^{\to}
\end{tikzcd}$$

By the key lemma: this is an adjoint equivalence

Univalence

There are several points in the proof where we use univalence:

- interpreting substitution
- transporting along equivalences
- characterizing adjoint equivalences (of comprehension categories and of pseudofunctors)

Pseudonatural Adjoint Equivalences

We use the following theorem

Theorem

A pseudotransformation is an adjoint equivalence in the bicategory of pseudofunctors if it is a pointwise adjoint equivalence.

This can be proven using mate calculus[14]

[14] Proposition 6.2.16 in "2-Dimensional Categories" by Johnson and Yau

Pseudonatural Adjoint Equivalences

We use the following theorem

Theorem

A pseudotransformation is an adjoint equivalence in the bicategory of pseudofunctors if it is a pointwise adjoint equivalence.

This can be proven using mate calculus[14]

However, with univalence we can get a nicer proof

[14] Proposition 6.2.16 in “2-Dimensional Categories” by Johnson and Yau

Pseudonatural Adjoint Equivalences

We use the following theorem

Theorem

A pseudotransformation is an adjoint equivalence in the bicategory of pseudofunctors if it is a pointwise adjoint equivalence.

This can be proven using mate calculus[14]

However, with univalence we can get a nicer proof

We combine two ideas:

- Displayed bicategories
- Equivalence induction

[14] Proposition 6.2.16 in "2-Dimensional Categories" by Johnson and Yau

Displayed Bicategories

Definition

A **displayed bicategory**[15] D over a bicategory B consists of

- For each $x : \mathsf{B}$ a type D_x of objects over x
- For each $f : x \to y$, $\overline{x} : \mathsf{D}_x$, and $\overline{y} : \mathsf{D}_y$ a type $\overline{x} \to_f \overline{y}$ of 1-cells over f from $\overline{x}$ to $\overline{y}$
- For each 2-cell $\tau : f \Rightarrow g$ and 1-cells $\overline{f} : \overline{x} \to_f \overline{y}$ and $\overline{g} : \overline{x} \to_g \overline{y}$ a set $\overline{f} \Rightarrow_\tau \overline{g}$ of 2-cells over τ
- and much more

[15] "Bicategories in univalent foundations", Ahrens, Frumin, Maggesi, Veltri, Van der Weide

Displayed Bicategories

Definition

A **displayed bicategory**[15] D over a bicategory B consists of

- For each x : B a type D_x of objects over x
- For each $f : x \to y$, $\overline{x} : \mathrm{D}_x$, and $\overline{y} : \mathrm{D}_y$ a type $\overline{x} \to_f \overline{y}$ of 1-cells over f from $\overline{x}$ to $\overline{y}$
- For each 2-cell $\tau : f \Rightarrow g$ and 1-cells $\overline{f} : \overline{x} \to_f \overline{y}$ and $\overline{g} : \overline{x} \to_g \overline{y}$ a set $\overline{f} \Rightarrow_\tau \overline{g}$ of 2-cells over τ
- and much more

Example of a displayed bicategory over UnivCat

- Objects over C: fibrations on C
- 1-cells over F: Cartesian functors over F
- 2-cells over τ: natural transformations over τ

[15] "Bicategories in univalent foundations", Ahrens, Frumin, Maggesi, Veltri, Van der Weide

Characterizing Adjoint Equivalences

Often we want to show that some pseudofunctor reflects adjoint equivalences

$$\begin{tikzcd}
e : x \to y & \mathcal{B}_2 \arrow[d, "p"] \\
pe : px \simeq py & \mathcal{B}_1
\end{tikzcd}$$

Characterizing Adjoint Equivalences

If we use **displayed bicategories**, we can use equivalence induction

$\bar{e} : \bar{x} \to_{e} \bar{y}$ $\quad\quad$ $\mathcal{D}$

$e : x \simeq y$ $\quad\quad$ $\mathcal{B}$

By induction on *e*: we only have to consider morphisms over identities

Introduction

Why Comprehension Categories?

The Bicategory of Comprehension Categories

The Biequivalence for Categories with Finite Limits

Extensions

Conclusion

Extending the Biequivalence ($\prod$-types)

Theorem

We have a biequivalence between the bicategories

- $\mathsf{DFLCompCat}_{\prod}$*: democratic univalent full comprehension categories with equalizer types, binary product types, unit types, sigma types, and* $\prod$*-types*
- LCCC*: univalent locally Cartesian closed categories*

Extending the Biequivalence ($\prod$-types)

Theorem

We have a biequivalence between the bicategories

- DFLCompCat$_{\prod}$*: democratic univalent full comprehension categories with equalizer types, binary product types, unit types, sigma types, and* $\prod$*-types*
- LCCC*: univalent locally Cartesian closed categories*

Key observation:

- This biequivalence is an extension of the biequivalence between DFLCompCat and FinLim.
- We can use **displayed biequivalences**

Total Bicategories

Definition

Every displayed bicategory D over B gives rise to a **total bicategory** $\int$ D:

- Objects: pairs x : B and $\overline{x}$ over x
- 1-cells from $(x, \overline{x})$ to $(y, \overline{y})$: pairs $f : x \rightarrow y$ and $\overline{f} : \overline{x} \rightarrow_f \overline{y}$
- 2-cells from $(f, \overline{f})$ to $(g, \overline{g})$: pairs $\tau : f \rightarrow g$ and $\overline{\tau} : \overline{f} \Rightarrow_\tau \overline{g}$

The total bicategory $\int$ D extends B with the structure in D

Total Bicategories

Definition

Every displayed bicategory D over B gives rise to a **total bicategory** $\int$ D:

- Objects: pairs x : B and $\overline{x}$ over x
- 1-cells from $(x, \overline{x})$ to $(y, \overline{y})$: pairs $f : x \to y$ and $\overline{f} : \overline{x} \to_f \overline{y}$
- 2-cells from $(f, \overline{f})$ to $(g, \overline{g})$: pairs $\tau : f \to g$ and $\overline{\tau} : \overline{f} \Rightarrow_\tau \overline{g}$

The total bicategory $\int$ D extends B with the structure in D

Examples:

- LCCC: total bicategory of the displayed bicategory over FinLim whose objects over C are proofs that C is locally Cartesian closed.
- DFLCompCat$_{\prod}$: total bicategory of the displayed bicategory over DFLCompCat whose objects over a comprehension category are interpretations of $\prod$-types.

Displayed Gadgets and Modularity

We have many gadgets in our toolkit for modular constructions

- **displayed bicategories** for constructing bicategories
- **displayed pseudofunctors** for constructing pseudofunctors
- **displayed pseudotransformations** for constructing pseudotransformations
- **displayed invertible modifications** for constructing invertible modifications
- **displayed biequivalences** for constructing biequivalences

The definitions of pseudofunctors, displayed pseudotransformations, displayed invertible modifications, and displayed biequivalences are left to your imagination (and the literature[16])

[16] "Bicategories in univalent foundations", Ahrens, Frumin, Maggesi, Veltri, Van der Weide

Extending the Biequivalence ($\prod$-types)

Theorem

We have a biequivalence between the bicategories

- $\mathsf{DFLCompCat}_{\prod}$*: democratic univalent full comprehension categories with equalizer types, binary product types, unit types, sigma types, and* $\prod$*-types*
- LCCC*: univalent locally Cartesian closed categories*

We prove this by constructing a displayed biequivalence over the biequivalence between DFLCompCat and FinLim.

Extending the Biequivalence (local properties)

Many type formers, such as quotients, disjoint sums, a type of propositions, are interpreted in essentially the same way: the slice categories C/x must have some categorical structure and the substitution functors preserve it

- Disjoint sum types: the slices C/x are extensive, substitution preserves coproducts
- Quotient types: the slices C/x are exact, substitution preserves regular epis
- Type of propositions: the slices C/x have a subobject classifier, substitution preserves subobject classifiers

General notion: **local property**[17]

[17] "Modular correspondence between dependent type theories and categories including pretopoi and topoi" by Maietti

Local Properties

Definition

A **local property** is given by

- a proposition $\mathsf{P}_{\mathsf{Cat}}(\mathsf{C})$ for each univalent category C with finite limits;
- a proposition $\mathsf{P}_{\mathsf{Fun}}(\mathsf{F})$ for each functor $\mathsf{F} : \mathsf{C}_1 \rightarrow \mathsf{C}_2$ such that F preserves finite limits and such that $\mathsf{P}_{\mathsf{Cat}}(\mathsf{C}_1)$ and $\mathsf{P}_{\mathsf{Cat}}(\mathsf{C}_2)$.

such that

- $\mathsf{P}_{\mathsf{Cat}}$ is closed under slicing
- the identity satisfies $\mathsf{P}_{\mathsf{Fun}}$ and $\mathsf{P}_{\mathsf{Fun}}$ is closed under composition
- If $\mathsf{F} : \mathsf{C}_1 \rightarrow \mathsf{C}_2$ is a functor preserving finite limits such that $\mathsf{P}_{\mathsf{Fun}}(\mathsf{F})$, then we also have $\mathsf{P}_{\mathsf{Fun}}(\mathsf{F}/x)$ for each $x : \mathsf{C}_1$. This is not in Maietti, but necessary for a biequivalence

Examples of Local Properties

Examples of local properties:

- being exact
- being extensive
- being a pretopos
- having a subobject classifier
- having a (parameterized) natural number objects

Extensions with Local Properties

Theorem

Let P *be a local property. We have a biequivalence between the bicategories*

- $\mathsf{DFLCompCat}_{\mathsf{P}}$*: democratic univalent full comprehension categories with equalizer types, binary product types, unit types, sigma types, such that each fiber satisfies* $\mathsf{P}_{\mathsf{Cat}}$ *and the substitution functors satisfy* $\mathsf{P}_{\mathsf{Fun}}$*.*
- $\mathsf{FinLim}_{\mathsf{P}}$*: univalent categories with finite limit that satisfy* $\mathsf{P}_{\mathsf{Cat}}$

Extensions to Toposes

Theorem

We have a biequivalence between the bicategories

$$\mathsf{PreTop} \simeq \mathsf{CompCat}_{\mathsf{PreTop}}$$

$$\mathsf{PreTop}_{\mathbb{N}} \simeq \mathsf{CompCat}_{\mathsf{PreTop}_{\mathbb{N}}}$$

$$\mathsf{PreTop}_{\prod} \simeq \mathsf{CompCat}_{\mathsf{PreTop}_{\prod}}$$

$$\mathsf{ElemTop} \simeq \mathsf{CompCat}_{\mathsf{ElemTop}}$$

$$\mathsf{ElemTop}_{\mathbb{N}} \simeq \mathsf{CompCat}_{\mathsf{ElemTop}_{\mathbb{N}}}$$

Introduction

Why Comprehension Categories?

The Bicategory of Comprehension Categories

The Biequivalence for Categories with Finite Limits

Extensions

Conclusion

Summary

- We proved an analogue of the theorem by Clairambault and Dybjer for univalent categories
- To do so, we used **non-discrete** structures (comprehension categories) instead of discrete ones (CwFs)
- Univalence simplified several constructions and proofs (transporting along equivalences, characterizing adjoint equivalences)
- We also extended this theorem to various classes of toposes following Maietti
- Displayed biequivalences were our main tool to construct extensions

Future Work

There are several interesting future directions:

- What would be a suitable syntax for univalent full comprehension categories?
- What about the semantics of intensional type theory using univalent categories? Relevant work: equivalence between path categories and certain comprehension categories[18]

[18] "Semantics of Axiomatic Type Theory" by Otten and Spadetto, https://types2024.itu.dk/abstracts.pdf#page=206

Internal Languages for Univalent Categories

So we can ask whether Martin-Löf type theory with extensional identity types, $\sum$- and $\prod$-types is an internal language for **univalent** *lcccs? And we can answer, yes. We can answer the analogous question in the same way for pretoposes, $\prod$-pretoposes, elementary toposes, and elementary toposes with an NNO.*

Preprint: https://arxiv.org/abs/2411.06636

Morphisms of Comprehension Categories

Morphisms from $\chi_1 : \mathsf{D}_1 \to \mathsf{C}_1^{\to}$ to $\chi_2 : \mathsf{D}_2 \to \mathsf{C}_2^{\to}$ are given by functors $\mathsf{F} : \mathsf{C}_1 \to \mathsf{C}_2$ and displayed functor $\overline{\mathsf{F}} : \mathsf{D}_1 \to \mathsf{D}_2$ over F together with a displayed **natural isomorphism** over the identity as follows

$$\begin{tikzcd}
\mathsf{D}_1 \arrow[r, "\chi_1"] \arrow[d, "\overline{\mathsf{F}}"'] \arrow[dr, phantom, "\Downarrow \cong"] & \mathsf{C}_1^{\to} \arrow[d, "\mathsf{F}^{\to}"] \\
\mathsf{D}_2 \arrow[r, "\chi_2"'] & \mathsf{C}_2^{\to}
\end{tikzcd}$$

Call the natural isomorphism F_χ.

2-cells of comprehension categories

A 2-cell from $(F, \overline{F}, F_\chi)$ to $(G, \overline{G}, G_\chi)$ consists of

- a natural transformation $\tau : F \Rightarrow G$
- a displayed natural transformation $\overline{\tau} : \overline{F} \Rightarrow \overline{G}$ over τ

such that the compositions below are equal.

$$\begin{tikzcd}
D_1 \arrow[r, "\chi_1"] \arrow[d, bend right=60, "\overline{F}"'] \arrow[d, bend left=60, "\overline{G}"] \arrow[d, phantom, "\overset{\overline{\tau}}{\Longrightarrow}"] \arrow[dr, phantom, "\overset{G_\chi}{\Longrightarrow}"] & C_1^\to \arrow[d, "G^\to"] \\
D_2 \arrow[r, "\chi_2"'] & C_2^\to
\end{tikzcd}
\qquad
\begin{tikzcd}
D_1 \arrow[r, "\chi_1"] \arrow[d, "\overline{F}"'] \arrow[dr, phantom, "\overset{F_\chi}{\Longrightarrow}"] & C_1^\to \arrow[d, bend right=60, "F^\to"'] \arrow[d, bend left=60, "G^\to"] \arrow[d, phantom, "\overset{\tau^\to}{\Longrightarrow}"] \\
D_2 \arrow[r, "\chi_2"'] & C_2^\to
\end{tikzcd}$$

Pseudonatural Adjoint Equivalences

Theorem

A pseudotransformation is an adjoint equivalence in the bicategory of pseudofunctors if it is a pointwise adjoint equivalence.

Pseudonatural Adjoint Equivalences

$\mathcal{B}_1$ and $\mathcal{B}_2$ bicat

$$\begin{tikzcd}
 & \mathrm{Psfunctor}(\mathcal{B}_1, \mathcal{B}_2) \arrow[d] & \\
\text{no coherences} \arrow[r, dash] & \mathrm{RawPsfunctor}(\mathcal{B}_1, \mathcal{B}_2) \arrow[d] & \text{reflect adjequiv} \\
\text{map on 0- and 1-cell} \arrow[r, dash] & \mathrm{MorMap}(\mathcal{B}_1, \mathcal{B}_2) \arrow[d] & \\
\text{map on objects} \arrow[r, dash] & \mathrm{ObjMap}(\mathcal{B}_1, \mathcal{B}_2) &
\end{tikzcd}$$

Pseudonatural Adjoint Equivalences

Theorem

A morphism $(\mathsf{F}, \overline{\mathsf{F}}, \mathsf{F}_\chi)$ *of full comprehension categories is an adjoint equivalence if* F *and* $\overline{\mathsf{F}}$ *are adjoint equivalences.*

Same Idea for Comprehension Categories

$$\begin{tikzcd}
& \mathsf{FullCompCat}_{\mathsf{D}} \arrow[d] & \\
& \mathsf{CompCat}_{\mathsf{lax}} \arrow[d] & \\
& \mathsf{Cleaving}_{\mathsf{T}} \arrow[d] & \\
\mathsf{UnivDispCat} \arrow[dr] & \mathsf{UnivDispCat}_{\mathsf{T}} \arrow[l] \arrow[r] & \mathsf{UnivCat}_{\mathsf{T}} \arrow[dl] \\
& \mathsf{UnivCat} &
\end{tikzcd}$$

Preservation of Democracy

Suppose, that we have democratic full univalent comprehension categories $\chi_1 : D_1 \to C_1^{\to}$ and $\chi_2 : D_2 \to C_2^{\to}$, and a morphism $(F, \overline{F}, F_\chi)$ from χ_1 to χ_2. Then for each $\Gamma : C_1$ there is an isomorphism $d_\Gamma : F(\overline{\Gamma}) \cong \diamond^*_{F(\Gamma)}(\overline{F(\Gamma)})$ making the following diagram commute.

$$\begin{tikzcd}
F(\Gamma) \arrow[rr, "F(\gamma_\Gamma)"] \arrow[d, "\gamma_{F(\Gamma)}"'] & & F(\langle\rangle.\overline{\Gamma}) \arrow[d, "F_\chi(\overline{\Gamma})"] \\
\langle\rangle.\overline{F(\Gamma)} & F(\langle\rangle).\diamond^*_{F(\Gamma)}(\overline{F(\Gamma)}) \arrow[l, "\langle \diamond_{F(\langle\rangle)}, q\rangle"'] & F(\langle\rangle).\diamond^*_{F(\Gamma)}(F(\overline{\Gamma})) \arrow[l, "F(\langle\rangle).d_\Gamma"']
\end{tikzcd}$$

Here $\gamma_\Gamma : \Gamma \cong \langle\rangle.\overline{\Gamma}$ and $\gamma_{F(\Gamma)} : F(\Gamma) \cong \langle\rangle.\overline{F(\Gamma)}$