The correspondence between LCCCs and dependent type theories from a univalent perspective

Niels van der Weide

27 October, 2021

Categorical Models

- Categorical models have been used to prove meta-theoretical properties of type theories (consistency, canonicity, conservativity, normalization)
- Such properties are proven by finding a nice model
- Often, we can acquire something stronger, namely an **equivalence** between a class of models and logic

Lambek's Theorem

Theorem (Lambek[1])

The categories of lambda-calculi and of cartesian categories are equivalent.

Slogan: **the simply-typed lambda calculus is the internal language of cartesian closed categories**.

[1] Lambek, J. "Cartesian closed categories and typed λ-calculi."

Dependent Types

- Dependent types: types can dependent on variables.
- This way, one can represent propositions as types.
- Example: the identity type, $x = y$ (**the identity type**).

In this talk, we look at models of dependent type theory with

- Extensional identity types
- Sigma types
- Dependent products

Call it $\mathbf{MLTT}_{\prod,\sum,\mathsf{ExtId}}$

Seely's Theorem

Recall:

- objects of the **slice category** $\mathcal{C}/Y$ are arrows $X \to Y$.
- $\mathcal{C}$ is **locally cartesian closed** if $\mathcal{C}/Y$ is cartesian closed for every object Y.

Theorem (Seely[2])

The categories of $\mathsf{MLTT}_{\prod,\sum,\mathsf{ExtId}}$ *and of LCCCs are equivalent.*

[2] Seely, R.A.G. "Locally cartesian closed categories and type theory."

Core Idea

We interpret type theory as follows

- Contexts Γ as objects
- Types A in context Γ: morphisms $A \to \Gamma$
- Terms of type A: sections of $A \to \Gamma$
- Substitutions from Γ_1 to Γ_2: morphisms $\Gamma_1 \to \Gamma_2$
- Substitution of types: pullbacks

A Problem!

$$\begin{tikzcd} A[\mathbf{id}] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner" very near start] & A \arrow[d] \\ \Gamma \arrow[r, "\mathbf{id}"'] & \Gamma \end{tikzcd}$$

- In type theory, $A[\mathbf{id}]$ must be equal to A.
- However, we can only guarantee they are isomorphic!

General challenge: **how to interpret dependent type theory in categories?**

Categorical Models of Dependent Type Theory

- Comprehension categories[3] (based on the notion of Grothendieck fibration)
- Categories with families[4]
- many others (categories with attributes, natural models, ...)

[3] Jacobs, B. "Comprehension categories and the semantics of type dependency."

[4] Dybjer, P. "Internal type theory."

Categorical Models of Dependent Type Theory

- **Comprehension categories**[3] (based on the notion of Grothendieck fibration)
- **Categories with families**[4]
- many others (categories with attributes, natural models, ...)

[3] Jacobs, B. "Comprehension categories and the semantics of type dependency."

[4] Dybjer, P. "Internal type theory."

Grothendieck Fibrations

Definition

A functor $P : \mathcal{E} \to \mathcal{B}$ is called a **Grothendieck fibration** if for every arrow $x \to P(\overline{y})$ (in $\mathcal{B}$) there is an $\overline{x}$ in $\mathcal{E}$ and a cartesian morphism $\overline{x} \to \overline{y}$ such that $P(\overline{x}) = x$.

$$
\begin{tikzcd}
\mathcal{E} \arrow[d, "P"'] \\
\mathcal{B}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\overline{x} \arrow[r] & \overline{y} \\
x \arrow[r] & P(\overline{y})
\end{tikzcd}
$$

Used to interpret substitution.

Split Fibrations

Lift of identity arrow:

$$\begin{tikzcd} \mathcal{E} \arrow[d, "P"'] \\ \mathcal{B} \end{tikzcd} \qquad\qquad \begin{tikzcd} \overline{z} \arrow[rr] & & \overline{x} \\ \\ P(\overline{x}) \arrow[rr, "\mathbf{id}"'] & & P(\overline{x}) \end{tikzcd}$$

(We can draw a similar diagram for the composition)

Definition

A Grothendieck fibration is called **split** if

- the lift of $\mathbf{id} : x \to x$ is the identity arrow
- the lift of a composition is the composition of the lifts.

Substitution laws hold up to equality.

Example of a fibration

If $\mathcal{C}$ has pullbacks, then codomain functor $\mathrm{cod} : \mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$ is a fibration. Not a split fibration in general.

Hofmann's Solution[6]

Main idea:

- We have a category with pullbacks.
- Then the codomain functor is a fibration.
- Replace it by a split fibration.

Theorem (Bénabou[5])

Every fibration is equivalent to a split fibration. In fact, the 2-category of fibrations is biequivalent to the 2-category of split fibrations.

[5] Bénabou, J. "Des Catégories Fibrées lecture notes by J.R. Roisin "

[6] Hofmann, M. "On the interpretation of type theory in locally cartesian closed categories."

Curien's Solution[7]

- Use a type theory in which the substitution laws only hold up to isomorphism
- So: type equalities are annotated in terms
- Prove coherence (all proofs of $A = B$ have the same interpretation)

[7] Curien, P.L. "Substitution up to Isomorphism."

Acquiring an equivalence

Democracy: for every context Γ there is a type $\overline{\Gamma}$ such that Γ is isomorphic to the empty context extended with $\overline{\Gamma}$.

Theorem (Clairambault and Dybjer[8])

- *The bicategories of finitely complete categories and democratic CwFs with extensional identity types and sigma types are biequivalent.*

[8] Clairambault, P., and Dybjer, P. "The biequivalence of locally cartesian closed categories and Martin-Löf type theories."

Acquiring an equivalence

Democracy: for every context Γ there is a type $\overline{\Gamma}$ such that Γ is isomorphic to the empty context extended with $\overline{\Gamma}$.

Theorem (Clairambault and Dybjer[8])

- *The bicategories of finitely complete categories and democratic CwFs with extensional identity types and sigma types are biequivalent.*
- *The bicategories of LCCCs and democratic CwFs with extensional identity types, sigma types, and dependent products are biequivalent.*

[8] Clairambault, P., and Dybjer, P. "The biequivalence of locally cartesian closed categories and Martin-Löf type theories."

Acquiring an equivalence

Democracy: for every context Γ there is a type $\overline{\Gamma}$ such that Γ is isomorphic to the empty context extended with $\overline{\Gamma}$.

Theorem (Clairambault and Dybjer[8])

- *The bicategories of finitely complete categories and democratic CwFs with extensional identity types and sigma types are biequivalent.*
- *The bicategories of LCCCs and democratic CwFs with extensional identity types, sigma types, and dependent products are biequivalent.*

The proof makes use of Hofmann's solution.

[8] Clairambault, P., and Dybjer, P. "The biequivalence of locally cartesian closed categories and Martin-Löf type theories."

Univalent Foundations (UF)

- Types as spaces, terms as points, equalities as paths
- Equality is proof relevant: not every proof that $x = y$ has to be equal
- Univalence axiom: equality of types is equivalence of types
- Model in simplicial sets[9]

[9] Kapulkin, K. and Lumsdaine, P. L. "The simplicial model of univalent foundations (after Voevodsky)"

The Univalence Axiom

Note: there is a canonical map τ sending equalities $X = Y$ to equivalences $X \simeq Y$.

Axiom (The Univalence Axiom)

The map τ is an equivalence.

Equality is Proof Relevant!

- By univalence, the types $X = Y$ and $X \simeq Y$ are equivalent for all types X and Y.
- Let $\mathbf{2}$ be the type with two inhabitants.
- There are two equivalences from $\mathbf{2}$ to $\mathbf{2}$.
- Hence, not all proofs that $\mathbf{2} = \mathbf{2}$ are equal.

Sets in UF

Definition

A type X is called a **set** if for all $x, y : X$ and $p, q : x = y$, we have $p = q$.

- Examples: the unit type, the type of natural numbers
- Non-example: the universe

Strict Categories

Category: usual definitions, but for all objects X and Y the morphisms $X \to Y$ must be a set.

Definition (Strict Category[10])

A category is called **strict** if the type of objects is a set.

[10] Ahrens, B., Kapulkin, K., and Shulman, M. "Univalent categories and the Rezk completion."

Univalent Categories

Note: there is a map $\tau_{X,Y}$ that sends equalities $X = Y$ of objects to isomorphisms $X \cong Y$.

Definition (Univalent Category[11])

A category is called **univalent** if the map $\tau_{X,Y}$ is an equivalence for all objects X and Y.

Note: the category **Set** of sets is univalent and **not** strict.

[11] Ahrens, B., Kapulkin, K., and Shulman, M. "Univalent categories and the Rezk completion."

Motivation

- Problem in type theory: define the syntax of type theory within type theory? See: the initiality project (De Boer, Brunerie, Lumsdaine, Mörtberg)[12], Altenkirch and Kaposi[13].
- These are based on **strict** categories.
- Drawback: **Set** falls out of the scope.

[12] De Boer, M., Brunerie, G., Lumsdaine, P.L., Mörtberg, A. "A formalization of the initiality conjecture in Agda."

[13] Altenkirch, T., and Kaposi, A. "Type theory in type theory using quotient inductive types."

Our goal

Find an analogue of Clairambault's and Dybjer's theorem for univalent categories.

Categories with Families

Definition (Just a part)

A **category with families** consists of

- A category $\mathcal{C}$
- A functor T from $\mathcal{C}^{\mathsf{op}}$ to the category of families of sets
- ...

Note:

- Objects of $\mathcal{C}$ are called **contexts**, morphisms in $\mathcal{C}$ are called **substitutions**.
- From T, we get a functor $\mathrm{Ty} : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$

Problem!

- This is too restrictive!
- In **Set**, types in the empty context are just sets. This does not form a set.
- Hence, CwFs cannot be used for our purposes.

Intermezzo: Displayed Categories

One can define fibrations without referring to the equality of objects by using **displayed categories**.

Definition (Displayed Category[14])

Let $\mathcal{C}$ be a category. A **displayed category** $\mathcal{D}$ over $\mathcal{C}$ consists of

- For every object $x : \mathcal{C}$ a type $\mathcal{D}_x$ of **displayed objects over** x
- For all morphism $f : x \to y$ and displayed objects $\overline{x} : \mathcal{D}_x$ and $\overline{y} : \mathcal{D}_y$ a **set** $\overline{x} \to_f \overline{y}$ of **displayed arrows over** f
- (dependent versions of identity, composition, and the laws)

So: **avoid equality of objects by talking about objects over**

[14] Ahrens, B., and Lumsdaine, P.L. "Displayed categories."

Full Comprehension Categories

Definition

A **comprehension category** consists of a fibration P and a cartesian functor χ such that the following diagram commutes on the nose

$$\begin{tikzcd} \mathcal{E} \arrow[rr, "\chi"] \arrow[dr, "P"'] & & \mathcal{C}^{\rightarrow} \arrow[dl, "\mathsf{cod}"] \\ & \mathcal{C} & \end{tikzcd}$$

A comprehension category is called **full** if χ is full and faithful.

Main conjecture

Conjecture

The bicategories of univalent left exact categories and democratic full comprehension categories with extensional identity types and sigma types are biequivalent.

Formalization in UniMath is work in progress[15].

[15] https://github.com/nmvdw/UniMath/tree/comp-cat

The Role of Intensionality

- Univalent foundations is an **intensional type theory**
- As such, type equalities are annotated in terms
- Equality of objects in a category is isomorphism
- So: when writing down syntactical rules, we annotate the term with the relevant isomorphisms.
- Also needed: coherencies for type equalities

Compare to

- Curien's solution
- Coherent type theory[16]

[16] https://bitbucket.org/akaposi/qiitcont/src/master/TT/Coh/

Conclusion

- Clairambault's and Dybjer's theorem is interesting to study in univalent foundations
- Due to intensionality: type equalities are annotated (similar to Curien)
- Due to univalent categories: coherencies are needed in the syntax
- In addition, CwFs need to be replaced by full comprehension categories