34th EACSL Annual Conference on Computer Science Logic
the 27th of February 2026

The biequivalence of path categories and axiomatic Martin-Löf type theories

Daniël Otten
University of Amsterdam

Matteo Spadetto *(speaker)*
University of Nantes

Content, *roughly*

1. Categorical semantics of dependent type theory
2. Computation axioms
3. Strictifying pre-models

Content, *roughly*

1. Categorical semantics of dependent type theory
2. Computation axioms
3. Strictifying pre-models

A semantics of a dependent type theory: a class of categorical structures that reproduce the given theory.

The type theoretic language of a category

If $\mathcal{C}$ is an appropriate category (e.g. a finitely complete category, display map category):

- Objects Γ of $\mathcal{C}$ are contexts.

The type theoretic language of a category

If $\mathcal{C}$ is an appropriate category (e.g. a finitely complete category, display map category):

- Objects Γ of $\mathcal{C}$ are contexts.
- Display maps $\Gamma.A \to \Gamma$ of codomain Γ are types A in context Γ.

The type theoretic language of a category

If $\mathcal{C}$ is an appropriate category (e.g. a finitely complete category, display map category):

- Objects Γ of $\mathcal{C}$ are contexts.
- Display maps $\Gamma.A \to \Gamma$ of codomain Γ are types A in context Γ.
- Sections of $\Gamma.A \to \Gamma$ are terms of A in context Γ.

Seely, *Locally cartesian closed categories and type theory*, 1983.

Substitution

If we are given $\Delta \xrightarrow{f} \Gamma$ and $\Gamma.A \to \Gamma$ then the judgement:

$$\Delta \vdash A[f] : \textsc{Type}$$

is represented by the pullback:

$$\Delta.A[f] \to \Delta \qquad \begin{array}{ccc} \Delta.A[f] & \longrightarrow & \Gamma.A \\ \downarrow & \lrcorner & \downarrow \\ \Delta & \xrightarrow{\quad f \quad} & \Gamma \end{array}$$

of $\Gamma.A \to \Gamma$ along f.

Substitution

If we are given $\Delta \xrightarrow{f} \Gamma$ and $\Gamma.A \to \Gamma$ then the judgement:

$$\Delta \vdash A[f] : \textsc{Type}$$

is represented by the pullback:

$$\Delta.A[f] \to \Delta \qquad \begin{array}{ccc} \Delta.A[f] & \longrightarrow & \Gamma.A \\ \downarrow & \lrcorner & \downarrow \\ \Delta & \xrightarrow{\quad f \quad} & \Gamma \end{array}$$

of $\Gamma.A \to \Gamma$ along f. If we are given a section $\Gamma \xrightarrow{t} \Gamma.A$ of $\Gamma.A \to \Gamma$ then the judgement:

$$\Delta \vdash t[f] : A[f]$$

is represented by the unique section of $\Delta.A[f] \to \Delta$ such that:

Substitution

If we are given $\Delta \xrightarrow{f} \Gamma$ and $\Gamma.A \to \Gamma$ then the judgement:

$$\Delta \vdash A[f] : \textsc{Type}$$

is represented by the pullback:

$$\Delta.A[f] \to \Delta \qquad\qquad \begin{array}{ccc} \Delta.A[f] & \longrightarrow & \Gamma.A \\ \downarrow & \lrcorner & \downarrow \\ \Delta & \xrightarrow{\quad f \quad} & \Gamma \end{array}$$

of $\Gamma.A \to \Gamma$ along f. If we are given a section $\Gamma \xrightarrow{t} \Gamma.A$ of $\Gamma.A \to \Gamma$ then the judgement:

$$\Delta \vdash t[f] : A[f]$$

is represented by the unique section of $\Delta.A[f] \to \Delta$ such that:

$$\begin{array}{ccc} \Delta & \xrightarrow{\quad f \quad} & \Gamma \\ \downarrow & & \downarrow t \\ \Delta.A[f] & \longrightarrow & \Gamma.A \\ \downarrow & \lrcorner & \downarrow \\ \Delta & \xrightarrow{\quad f \quad} & \Gamma \end{array}$$

commutes.

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}}$$

$$\frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \quad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}}$$

$$\frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \qquad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

The category $\mathcal{C}$ is a model of these rules when there exist appropriate choice functions reproducing these rules in $\mathcal{C}$ (syntactic description).

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}} \qquad \frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \quad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

The category $\mathcal{C}$ is a model of these rules when there exist appropriate choice functions reproducing these rules in $\mathcal{C}$ (syntactic description).

Sometimes, a type constructor can be characterised in terms of a categorical property on the family of display maps (categorical description).

Example. Extensional identity types:

$$\text{If } t(x, x', p) : C(x, x', p) \text{ then } t(x, x', p) \equiv \mathsf{J}(t(\cdot, \cdot, r(\cdot)), x, x', p).$$

Proposition. *A model of extensional identity types is a display map category such that, for every display map* $\Gamma.A \to \Gamma$*, the diagonal arrow* $\Gamma.A \to \Gamma.A.A^{\blacktriangledown}$ *is isomorphic to a display map.*

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}} \qquad \frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \quad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

The category $\mathcal{C}$ is a model of these rules when there exist appropriate choice functions reproducing these rules in $\mathcal{C}$ (syntactic description).

Sometimes, a type constructor can be characterised in terms of a categorical property on the family of display maps (categorical description).

Example. Extensional identity types:

$$\text{If } t(x, x', p) : C(x, x', p) \text{ then } t(x, x', p) \equiv \mathsf{J}(t(\cdot, \cdot, r(\cdot)), x, x', p).$$

Proposition. *A model of extensional identity types is a display map category such that, for every display map* $\Gamma.A \to \Gamma$*, the diagonal arrow* $\Gamma.A \to \Gamma.A.A^{\blacktriangledown}$ *is isomorphic to a display map.*

Theorem (Seely)**.** *A finitely complete category is a model of extensional identity types, extensional unit types, and extensional dependent sum types.*

Approaches to describe a model

The categorical description is clear and conceptually simple for **extensional** type formers.

This is not the case for **intensional** type formers: there aren't obvious categorical properties to characterise intensional inference rules.

Approaches to describe a model

The categorical description is clear and conceptually simple for **extensional** type formers.

This is not the case for **intensional** type formers: there aren't obvious categorical properties to characterise intensional inference rules.

In these cases, the homotopy theoretic description may be useful: we *rely on a primitive notion of* ***weak equivalence*** *in the categorical structure to phrase the type formers.*

Approaches to describe a model

The categorical description is clear and conceptually simple for **extensional** type formers.

This is not the case for **intensional** type formers: there aren't obvious categorical properties to characterise intensional inference rules.

In these cases, the homotopy theoretic description may be useful: we *rely on a primitive notion of* ***weak equivalence*** *in the categorical structure to phrase the type formers.* Cf. higher-categorical description.

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}} \qquad \frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \quad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

The category $\mathcal{C}$ is a model of these rules when there exist appropriate choice functions reproducing these rules in $\mathcal{C}$ (syntactic description).

Sometimes, a type constructor can be characterised in terms of a categorical property on the family of display maps (categorical description).

Example. Extensional identity types:

$$\text{If } t(x, x', p) : C(x, x', p) \text{ then } t(x, x', p) \equiv \mathsf{J}(t(\cdot, \cdot, r(\cdot)), x, x', p).$$

Proposition. *A model of extensional identity types is a display map category such that, for every display map $\Gamma.A \to \Gamma$, the diagonal arrow $\Gamma.A \to \Gamma.A.A^{\blacktriangledown}$ is isomorphic to a display map.*

Theorem (Seely). *A finitely complete category is a model of extensional identity types, extensional unit types, and extensional dependent sum types.*

Approaches to describe a model

Using this *translation*, we can say when a categorical structure $\mathcal{C}$ is a model of any type constructor, like the following:

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}} \qquad \frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \quad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

The category $\mathcal{C}$ is a model of these rules when there exist appropriate choice functions reproducing these rules in $\mathcal{C}$ (syntactic description).

Sometimes, a type constructor can be characterised in terms of a categorical property on the family of display maps (categorical description).

Example. Extensional identity types:

$$\text{If } t(x, x', p) : C(x, x', p) \text{ then } t(x, x', p) \equiv \mathsf{J}(t(\cdot, \cdot, r(\cdot)), x, x', p).$$

Proposition. *A model of extensional identity types is a display map category such that, for every display map $\Gamma.A \to \Gamma$, the diagonal arrow $\Gamma.A \to \Gamma.A.A^{\blacktriangledown}$ is isomorphic to a display map.*

Theorem (Seely)**.** *A finitely complete category is a pre-model of extensional identity types, extensional unit types, and extensional dependent sum types.*

Finitely complete categories

Seely proved that, as display map categories, finitely complete categories are *almost* models of extensional identity types, extensional dependent sum types, and extensional unit types.

Finitely complete categories

Seely proved that, as display map categories, finitely complete categories are *almost* models of extensional identity types, extensional dependent sum types, and extensional unit types.

If we are given $\Omega \xrightarrow{g} \Delta \xrightarrow{f} \Gamma$ and A in context Γ and t of type A, then:

- $\Omega.A[f][g] \cong \Omega.A[fg]$ and not necessarily $\Omega.A[f][g] \equiv \Omega.A[fg]$

Finitely complete categories

Seely proved that, as display map categories, finitely complete categories are *almost* models of extensional identity types, extensional dependent sum types, and extensional unit types.

If we are given $\Omega \xrightarrow{g} \Delta \xrightarrow{f} \Gamma$ and A in context Γ and t of type A, then:

- $\Omega.A[f][g] \cong \Omega.A[fg]$ and not necessarily $\Omega.A[f][g] \equiv \Omega.A[fg]$
- $$\begin{array}{ccc} & \overset{t[f][g]}{\nearrow} & \Omega.A[f][g] \\ \Omega & & \wr\Vert \\ & \underset{t[fg]}{\searrow} & \Omega.A[fg] \end{array}$$ and not necessarily $t[f][g] \equiv t[fg]$

Finitely complete categories

Seely proved that, as display map categories, finitely complete categories are *almost* models of extensional identity types, extensional dependent sum types, and extensional unit types.

If we are given $\Omega \xrightarrow{g} \Delta \xrightarrow{f} \Gamma$ and A in context Γ and t of type A, then:

- $\Omega.A[f][g] \cong \Omega.A[fg]$ and not necessarily $\Omega.A[f][g] \equiv \Omega.A[fg]$
- $$\begin{array}{ccc} & \xrightarrow{t[f][g]} & \Omega.A[f][g] \\ \Omega & & \wr\| \\ & \xrightarrow{t[fg]} & \Omega.A[fg] \end{array}$$ and not necessarily $t[f][g] \equiv t[fg]$

However $\Omega \vdash A[f][g] \equiv A[f[g]]$ and $\Omega \vdash t[f][g] \equiv t[f[g]]$ are derivable: in this sense we do not necessarily have a genuine model.

Hofmann's coherence result

However, in:

Hofmann, *On the interpretation of type theory in locally cartesian closed categories*, 1994.

every finitely complete category is shown to be equivalent to a *split* display map category (still endowed with extensional type formers). Here 'split' means that there is a choice of pullback squares and $A[f][g] \equiv A[f[g]]$.

Hofmann's coherence result

However, in:

Hofmann, *On the interpretation of type theory in locally cartesian closed categories*, 1994.

every finitely complete category is shown to be equivalent to a *split* display map category (still endowed with extensional type formers). Here 'split' means that there is a choice of pullback squares and $A[f][g] \equiv A[f[g]]$.

In general:

Theorem (the right adjoint-splitting theorem). *The inclusion:*

$$\{\textit{split display map categories over } \mathcal{C}\} \hookrightarrow \{\textit{display map categories over } \mathcal{C}\}$$

has a right adjoint that:

Hofmann's coherence result

However, in:

Hofmann, *On the interpretation of type theory in locally cartesian closed categories*, 1994.

every finitely complete category is shown to be equivalent to a *split* display map category (still endowed with extensional type formers). Here 'split' means that there is a choice of pullback squares and $A[f][g] \equiv A[f[g]]$.

In general:

Theorem (the right adjoint-splitting theorem). *The inclusion:*

$$\{\textit{split display map categories over } \mathcal{C}\} \hookrightarrow \{\textit{display map categories over } \mathcal{C}\}$$

has a right adjoint that:

- *maps every given display map category $\mathcal{C}$ into a* split *display map category which is equivalent to $\mathcal{C}$;*
- *(under the pseudo-stability condition) preserves extensional structure.*

Hofmann's coherence result

However, in:

Hofmann, *On the interpretation of type theory in locally cartesian closed categories*, 1994.

every finitely complete category is shown to be equivalent to a *split* display map category (still endowed with extensional type formers). Here 'split' means that there is a choice of pullback squares and $A[f][g] \equiv A[f[g]]$.

In general:

Theorem (the right adjoint-splitting theorem). *The inclusion:*

$$\{\textit{split display map categories over } \mathcal{C}\} \hookrightarrow \{\textit{display map categories over } \mathcal{C}\}$$

has a right adjoint that:

- *maps every given display map category $\mathcal{C}$ into a* split *display map category which is equivalent to $\mathcal{C}$;*
- *(under the pseudo-stability condition) preserves extensional structure.*

Warren, *Homotopy theoretic aspects of constructive type theory*, 2008.

Streicher, *Fibred categories à la Jean Bénabou*, 2018.

Hofmann's coherence result

Clairambault & Dybjer, and Maietti, proved that:

Theorem. *There exists a biequivalence between:*

- *the 2-category of finitely complete categories*
- *an appropriate 2-category for the* contextual models of extensional identity types, dependent sum types, and unit types.

Hofmann's coherence result

Clairambault & Dybjer, and Maietti, proved that:

Theorem. *There exists a biequivalence between:*

- *the 2-category of finitely complete categories*
- *an appropriate 2-category for the* contextual models of extensional identity types, dependent sum types, and unit types.

Additionally:

Theorem (the left-adjoint splitting theorem)**.** *The inclusion:*

$$\{\textit{split display map categories over } \mathcal{C}\} \hookrightarrow \{\textit{cloven display map categories over } \mathcal{C}\}$$

has left adjoint. It:

Hofmann's coherence result

Clairambault & Dybjer, and Maietti, proved that:

Theorem. *There exists a biequivalence between:*

- *the 2-category of finitely complete categories*
- *an appropriate 2-category for the* contextual models of extensional identity types, dependent sum types, and unit types.

Additionally:

Theorem (the left-adjoint splitting theorem)**.** *The inclusion:*

$$\{\textit{split display map categories over } \mathcal{C}\} \hookrightarrow \{\textit{cloven display map categories over } \mathcal{C}\}$$

has left adjoint. It:

- *maps every cloven display map category $\mathcal{C}$ into a split display map category which is equivalent to the given one;*
- *(under the weak-stability condition) preserves the semantic intensional type formers.*

Lumsdaine, Warren, *The local universes model*, 2015.

One may ask if there exists a coherence result for axiomatic identity types as well.

One may ask if there exists a coherence result for axiomatic identity types as well.

Intensional identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}}$$

$$\frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \quad \mathsf{J}(q, x, x, r(x)) \equiv q(x) \end{array}}$$

One may ask if there exists a coherence result for axiomatic identity types as well.

Axiomatic identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}}$$

$$\frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \mathsf{H}(q, x) : \mathsf{J}(q, x, x, r(x)) = q(x) \end{array}}$$

One may ask if there exists a coherence result for axiomatic identity types as well.

Axiomatic identity types

$$\frac{\vdash A : \textsc{Type}}{\begin{array}{c} x, x' : A \vdash x = x' : \textsc{Type} \\ x : A \vdash r(x) : x = x \end{array}}$$

$$\frac{\begin{array}{c} \vdash A : \textsc{Type} \\ x, x' : A;\ p : x = x' \vdash C(x, x', p) : \textsc{Type} \\ x : A \vdash q(x) : C(x, x, r(x)) \end{array}}{\begin{array}{c} x, x' : A;\ p : x = x' \vdash \mathsf{J}(q, x, x', p) : C(x, x', p) \\ x : A \vdash \mathsf{H}(q, x) : \mathsf{J}(q, x, x, r(x)) = q(x) \end{array}}$$

Some reasons to study axiomatic identity types: broader concept of semantics, conservativity.

Path categories i.e. non-genuine models of axiomatic identity types

A path category $\mathcal{C}$ is a category with a terminal object, a class of fibrations and a class of weak equivalences such that the following properties are satisfied:

1. The composition of two fibrations is a fibration as well.
2. Every pullback of a fibration exists and is a fibration as well.
3. Every pullback of an acyclic fibration is a trivial fibration as well.
4. Weak equivalences satisfy 2-out-of-six.
5. Every isomorphism is a trivial fibration and every trivial fibration has a section.
6. For every object X of $\mathcal{C}$ there is an object PX, called path object on X, together with a weak equivalence $X \xrightarrow{r} PX$ and a fibration $PX \xrightarrow{\langle s,t \rangle} X \times X$ such that $(X \xrightarrow{r} PX \xrightarrow{\langle s,t \rangle} X \times X) = (X \to X \times X)$.
7. Every arrow of target a terminal object is a fibration.

Path categories i.e. non-genuine models of axiomatic identity types

A path category $\mathcal{C}$ is a category with a terminal object, a class of fibrations and a class of weak equivalences such that the following properties are satisfied:

1. The composition of two fibrations is a fibration as well.
2. Every pullback of a fibration exists and is a fibration as well.
3. Every pullback of an acyclic fibration is a trivial fibration as well.
4. Weak equivalences satisfy 2-out-of-six.
5. Every isomorphism is a trivial fibration and every trivial fibration has a section.
6. For every object X of $\mathcal{C}$ there is an object PX, called path object on X, together with a weak equivalence $X \xrightarrow{r} PX$ and a fibration $PX \xrightarrow{\langle s,t \rangle} X \times X$ such that $(X \xrightarrow{r} PX \xrightarrow{\langle s,t \rangle} X \times X) = (X \to X \times X)$.
7. Every arrow of target a terminal object is a fibration.

Theorem. *Path categories are contextual display map categories with extensional unit and dependent sum types, and axiomatic =-types, and vice versa.*

Path categories i.e. non-genuine models of axiomatic identity types

A path category $\mathcal{C}$ is a category with a terminal object, a class of fibrations and a class of weak equivalences such that the following properties are satisfied:

1. The composition of two fibrations is a fibration as well.
2. Every pullback of a fibration exists and is a fibration as well.
3. Every pullback of an acyclic fibration is a trivial fibration as well.
4. Weak equivalences satisfy 2-out-of-six.
5. Every isomorphism is a trivial fibration and every trivial fibration has a section.
6. For every object X of $\mathcal{C}$ there is an object PX, called path object on X, together with a weak equivalence $X \xrightarrow{r} PX$ and a fibration $PX \xrightarrow{\langle s,t \rangle} X \times X$ such that $(X \xrightarrow{r} PX \xrightarrow{\langle s,t \rangle} X \times X) = (X \to X \times X)$.
7. Every arrow of target a terminal object is a fibration.

Theorem. *Path categories are contextual display map categories with extensional unit and dependent sum types, and axiomatic =-types, and vice versa.*

This statement extends to a result à la Clairambault & Dybjer.

Rough statement

There exists a *biequivalence* between:

- the 2-category of the *contextual models of axiomatic* $=$*-types* (+ other constructors)

Rough statement

There exists a *biequivalence* between:

- the 2-category of the *contextual models of axiomatic =-types* (+ other constructors)
- the 2-category of *path categories*

Rough statement

There exists a *biequivalence* between:

- the 2-category of the *contextual models of axiomatic =-types* (+ other constructors)
- the 2-category of *path categories*

in the manner of the biequivalence between:

- the 2-category of the *contextual models of extensional =-types* (+ other constructors)
- the 2-category of finitely complete categories

studied by Seely, Hofmann, Clairambault & Dybjer, and Maietti.

The precise statement

Semantics of $=_{\mathrm{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$

$$\simeq$$

$$\{\text{models of } =_{\mathrm{ext}} \text{ (and } \Sigma_{\mathrm{ext}}, 1_{\mathrm{ext}})\}$$

The precise statement

Semantics of $=_{\text{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ext}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}})\}$$

Semantics of $=_{\text{ax}}$

$$\{\text{path categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ax}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}})\}$$

The precise statement

Universal data are pseudo-unique

The precise statement

Universal data are pseudo-unique

Homotopy universal data are weakly-unique

The precise statement

Semantics of $=_{\text{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ext}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}})\}$$

Semantics of $=_{\text{ax}}$

$$\{\text{path categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ax}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}})\}$$

The precise statement

Semantics of $=_{\text{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ext}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}}) \text{ and data pseudo-preserving morphisms}\}$$

Semantics of $=_{\text{ax}}$

$$\{\text{path categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ax}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}})\}$$

The precise statement

Semantics of $=_{\text{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ext}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}}) \text{ and data pseudo-preserving morphisms}\}$$

Semantics of $=_{\text{ax}}$

$$\{\text{path categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ax}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}}) \text{ and data weakly-preserving morphisms}\}$$

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

Finitely complete categories induce *pseudo-stable data*.

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

Finitely complete categories induce *pseudo-stable data*.

$\Longrightarrow$

These data can be preserved by the ***right***-*adjoint splitting*.

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

Finitely complete categories induce *pseudo-stable data*.

$\Longrightarrow$

These data can be preserved by the ***right***-*adjoint splitting*.

Path categories induce *weakly-stable data*.

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

Finitely complete categories induce *pseudo-stable data*.

$\Longrightarrow$

These data can be preserved by the ***right***-*adjoint splitting*.

Path categories induce *weakly-stable data*.

$\Longrightarrow$

These data are only preserved by the ***left***-*adjoint splitting*.

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

Finitely complete categories induce *pseudo-stable data*.
$\Longrightarrow$
These data can be preserved by the ***right****-adjoint splitting*.

Path categories induce *weakly-stable data*.
$\Longrightarrow$
These data are only preserved by the ***left****-adjoint splitting*.
$\Longrightarrow$
Models in input need to be *cloven*.

The precise statement

In general, finitely complete categories and path categories are *pre-models that we need to strictify*.

Finitely complete categories induce *pseudo-stable data*.

$\Longrightarrow$

These data can be preserved by the ***right**-adjoint splitting*.

Path categories induce *weakly-stable data*.

$\Longrightarrow$

These data are only preserved by the ***left**-adjoint splitting*.

$\Longrightarrow$

Models in input need to be *cloven*. Additionally, they need to have *function variable contexts* (LF condition).

The precise statement

Semantics of $=_{\mathrm{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\mathrm{ext}} \text{ (and } \Sigma_{\mathrm{ext}}, 1_{\mathrm{ext}}) \text{ and data pseudo-preserving morphisms}\}$$

Semantics of $=_{\mathrm{ax}}$

$$\{\text{path categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\mathrm{ax}} \text{ (and } \Sigma_{\mathrm{ext}}, 1_{\mathrm{ext}}) \text{ and data weakly-preserving morphisms}\}$$

The precise statement

Semantics of $=_{\text{ext}}$

$$\{\text{finitely complete categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ext}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}}) \text{ and data pseudo-preserving morphisms}\}$$

Semantics of $=_{\text{ax}}$

$$\{\text{LF cloven path categories and structure preserving functors}\}$$
$$\simeq$$
$$\{\text{models of } =_{\text{ax}} \text{ (and } \Sigma_{\text{ext}}, 1_{\text{ext}}) \text{ and data weakly-preserving morphisms}\}$$

Comments

- In particular, we obtained a *coherence result* for path categories.
- These results can be specialised to include other type formers (axiomatic Π-types) and generalised to drop extensional ones (display map path categories).
- There is also a *categorical counterpart*, where models are formulated using *higher category theoretic* data rather than *homotopy theoretic data*.

Future work

- Display map path categories and additional axiomatic type formers (Daniël Otten).
- The higher category theory of path categories.
- Replacing the class of path categories with an appropriate class of model categories.
- A directed version of the notion of a path category, based on the work of Fernando Chu and Paige Randal North (Calum Hughes).