Categorical models of homotopy type theory

Michael Shulman

12 April 2012

Outline

1. Homotopy type theory in model categories
2. The universal Kan fibration
3. Models in $(\infty, 1)$-toposes

Homotopy type theory in higher categories

Recall:

homotopy type theory		$(\infty, 1)$-categories
$\times$, $+$ types	$\longleftrightarrow$	products, coproducts
equality types $(x = y)$	$\longleftrightarrow$	diagonals
$\prod$ types	$\longleftrightarrow$	local cartesian closure
univalent universe Type	$\longleftrightarrow$	object classifier

Homotopy type theory in higher categories

Recall:

homotopy type theory		$(\infty, 1)$-categories
$\times$, $+$ types	$\longleftrightarrow$	products, coproducts
equality types $(x = y)$	$\longleftrightarrow$	diagonals
$\prod$ types	$\longleftrightarrow$	local cartesian closure
univalent universe Type	$\longleftrightarrow$	object classifier

Two kinds of equality

Problem

Type theory is stricter than $(\infty, 1)$-categories.

Two kinds of equality

Problem

Type theory is stricter than $(\infty, 1)$-categories.

In type theory, we have two kinds of "equality":

1. Equality witnessed by inhabitants of equality types (= paths).
2. Computational equality: $(\lambda x.b)(a)$ evaluates to $b[a/x]$.

Two kinds of equality

Problem

Type theory is stricter than $(\infty, 1)$-categories.

In type theory, we have two kinds of "equality":

1. Equality witnessed by inhabitants of equality types (= paths).
2. Computational equality: $(\lambda x.b)(a)$ evaluates to $b[a/x]$.

These play different roles: type checking depends on computational equality.

- if a evaluates to b, and $c : C(a)$, then also $c : C(b)$.
  - In particular, if a evaluates to b, then $\mathsf{refl}_b : (a = b)$.
- if $p : (a = b)$ and $c : C(a)$, then only $\mathsf{transport}(p, c) : C(b)$.

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$g \circ f := \lambda x.g(f(x))$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$\begin{aligned} g \circ f &:\equiv \lambda x. g(f(x)) \\ h \circ (g \circ f) &= \lambda x. h\Big((g \circ f)(x)\Big) \end{aligned}$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$
\begin{aligned}
g \circ f &:= \lambda x.g(f(x)) \\
h \circ (g \circ f) &= \lambda x.h\Big(\big(\lambda x.g(f(x))\big)(x)\Big)
\end{aligned}
$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$g \circ f :\equiv \lambda x.g(f(x))$$

$$h \circ (g \circ f) = \lambda x.h\Big(\big(\lambda x.g(f(x))\big)(x)\Big) \rightsquigarrow \lambda x.h(g(f(x)))$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$
\begin{aligned}
g \circ f &:= \lambda x. g(f(x)) \\
h \circ (g \circ f) &= \lambda x. h\Big(\big(\lambda x. g(f(x))\big)(x) \Big) \rightsquigarrow \lambda x. h(g(f(x))) \\
(h \circ g) \circ f &= \lambda x. \Big(h \circ g \Big)(f(x))
\end{aligned}
$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$
\begin{aligned}
g \circ f &:= \lambda x.g(f(x)) \\
h \circ (g \circ f) &= \lambda x.h\Big(\big(\lambda x.g(f(x))\big)(x) \Big) \rightsquigarrow \lambda x.h(g(f(x))) \\
(h \circ g) \circ f &= \lambda x.\Big(\lambda y.h(g(y))\Big)(f(x))
\end{aligned}
$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$
\begin{aligned}
g \circ f &:= \lambda x. g(f(x)) \\
h \circ (g \circ f) &= \lambda x. h\Big(\big(\lambda x. g(f(x))\big)(x) \Big) \rightsquigarrow \lambda x. h(g(f(x))) \\
(h \circ g) \circ f &= \lambda x. \Big(\lambda y. h(g(y)) \Big)(f(x)) \rightsquigarrow \lambda x. h(g(f(x)))
\end{aligned}
$$

Two kinds of equality

But computational equality is also stricter.

Example

Composition is computationally strictly associative.

$$g \circ f := \lambda x.g(f(x))$$
$$h \circ (g \circ f) = \lambda x.h\Big((\lambda x.g(f(x)))(x)\Big) \rightsquigarrow \lambda x.h(g(f(x)))$$
$$(h \circ g) \circ f = \lambda x.\Big(\lambda y.h(g(y))\Big)(f(x)) \rightsquigarrow \lambda x.h(g(f(x)))$$

- This is the sort of issue that homotopy theorists are intimately familiar with!
- We need a model for $(\infty, 1)$-categories with (at least) a strictly associative composition law.

Display map categories

Forget everything you know about homotopy theory; let's see how the type theorists come at it.

Definition

A display map category is a category with

- A terminal object.
- A subclass of its morphisms called the display maps, denoted $P \twoheadrightarrow A$ or $P \rightarrowtriangle A$.
- Any pullback of a display map exists and is a display map.

Display map categories

Forget everything you know about homotopy theory; let's see how the type theorists come at it.

Definition

A display map category is a category with

- A terminal object.
- A subclass of its morphisms called the display maps, denoted $P \twoheadrightarrow A$ or $P \rightarrowtriangle A$.
- Any pullback of a display map exists and is a display map.

- A display map $P \twoheadrightarrow A$ is a type dependent on A.
- A display map $A \twoheadrightarrow 1$ is a plain type (dependent on nothing).
- Pullback is substitution.

Dependent sums of display maps

$$(x \colon A) \vdash (B(x) : \mathrm{Type})$$

If the types $B(x)$ are the fibers of $B \twoheadrightarrow A$, their dependent sum $\sum_{x \colon A} B(x)$ should be the object B.

$$(x \colon A) \vdash (B(x) : \mathrm{Type}) \qquad \begin{array}{c} B \\ \downarrow\!\!\!\!\downarrow \\ A \\ \downarrow\!\!\!\!\downarrow \\ 1 \end{array}$$

$$\vdash \left(\sum_{x \colon A} B(x) : \mathrm{Type} \right) \qquad \begin{array}{c} B \\ \downarrow\!\!\!\!\downarrow \\ 1 \end{array}$$

Dependent sums in context

More generally:

$$(x\colon A), (y\colon B(x)) \vdash (C(x,y) : \mathsf{Type}) \qquad \begin{array}{c} C \\ \downarrow\!\!\downarrow \\ B \\ \downarrow\!\!\downarrow \\ A \end{array}$$

$$(x\colon A) \vdash \left(\sum_{y\colon B(x)} C(x,y) : \mathsf{Type} \right) \qquad \begin{array}{c} C \\ \downarrow\!\!\downarrow \\ A \end{array}$$

Dependent sums in context

More generally:

$$(x\colon A), (y\colon B(x)) \vdash (C(x,y) : \mathrm{Type}) \qquad \begin{array}{c} C \\ \downarrow\!\!\!\downarrow \\ B \\ \downarrow\!\!\!\downarrow \\ A \end{array}$$

$$(x\colon A) \vdash \left(\textstyle\sum_{y\colon B(x)} C(x,y) : \mathrm{Type}\right) \qquad \begin{array}{c} C \\ \downarrow\!\!\!\downarrow \\ A \end{array}$$

Dependent sums $\longleftrightarrow$ display maps compose

Aside: adjoints to pullback

- In a category $\mathscr{C}$, if pullbacks along $f\colon A \to B$ exist, then the functor
$$f^*\colon \mathscr{C}/B \longrightarrow \mathscr{C}/A$$
has a left adjoint Σ_f given by composition with f.

Aside: adjoints to pullback

- In a category $\mathscr{C}$, if pullbacks along $f\colon A \to B$ exist, then the functor

$$f^*\colon \mathscr{C}/B \longrightarrow \mathscr{C}/A$$

  has a left adjoint Σ_f given by composition with f.

- If f is a display map and display maps compose, then Σ_f restricts to a functor

$$(\mathscr{C}/A)_{\mathsf{disp}} \longrightarrow (\mathscr{C}/B)_{\mathsf{disp}}$$

  implementing dependent sums.

Aside: adjoints to pullback

- In a category $\mathscr{C}$, if pullbacks along $f\colon A \to B$ exist, then the functor

$$f^*\colon \mathscr{C}/B \longrightarrow \mathscr{C}/A$$

has a left adjoint Σ_f given by composition with f.

- If f is a display map and display maps compose, then Σ_f restricts to a functor

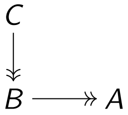
$$(\mathscr{C}/A)_{\mathsf{disp}} \longrightarrow (\mathscr{C}/B)_{\mathsf{disp}}$$

implementing dependent sums.

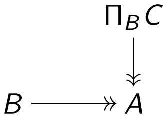
- A right adjoint to f^*, if one exists, is an "object of sections". $\mathscr{C}$ is locally cartesian closed iff all such right adjoints Π_f exist.

Dependent products of display maps

$$(x\colon A), (y\colon B(x)) \vdash (C(x,y) : \mathsf{Type})$$

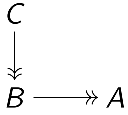


$$(x\colon A) \vdash \left(\prod_{y\colon B(x)} C(x,y) : \mathsf{Type}\right)$$

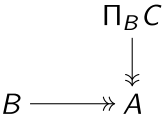


Dependent products of display maps

$$(x\colon A), (y\colon B(x)) \vdash (C(x,y) : \mathsf{Type})$$



$$(x\colon A) \vdash \left(\prod_{y\colon B(x)} C(x,y) : \mathsf{Type}\right)$$



Dependent products $\longleftrightarrow$ “display maps exponentiate”

Identity types for display maps

The dependent identity type

$$(x\colon A), (y\colon A) \vdash ((x = y) : \mathsf{Type})$$

must be a display map

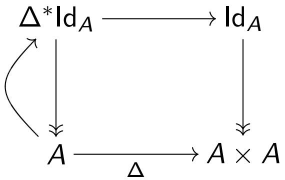
$$\begin{array}{c} \mathsf{Id}_A \\ \big\downarrow\!\!\!\!\downarrow \\ A \times A \end{array}$$

Identity types for display maps

The reflexivity constructor

$$(x\colon A) \ \vdash \ (\mathsf{refl}(x) : (x = x))$$

must be a section

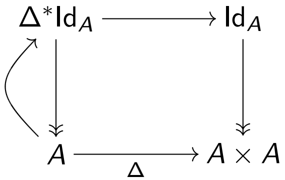


Identity types for display maps

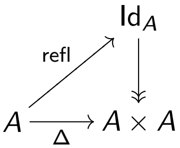
The reflexivity constructor

$$(x\colon A)\ \vdash\ (\mathsf{refl}(x) : (x = x))$$

must be a section

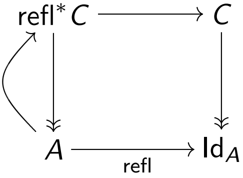


or equivalently a lifting

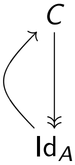


Identity types for display maps

The eliminator says given a dependent type with a section

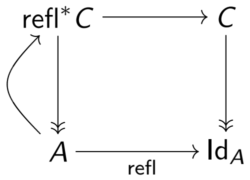


there exists
a compatible
section

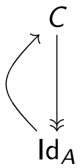


Identity types for display maps

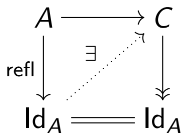
The eliminator says given a dependent type with a section



there exists a compatible section



In other words, we have the lifting property



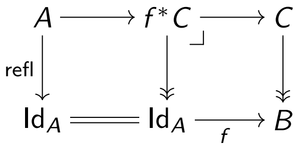
Identity types for display maps

In fact, refl has the left lifting property w.r.t. all display maps.

$$\begin{array}{ccc} A & \longrightarrow & C \\ {\scriptstyle \mathsf{refl}}\big\downarrow & & \big\downarrow\!\!\!\!\downarrow \\ \mathsf{Id}_A & \xrightarrow[f]{} & B \end{array}$$

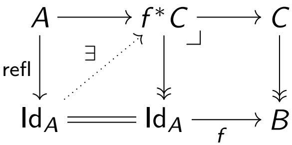
Identity types for display maps

In fact, refl has the left lifting property w.r.t. all display maps.



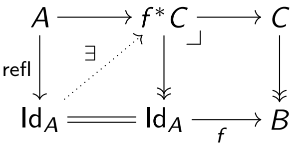
Identity types for display maps

In fact, refl has the left lifting property w.r.t. all display maps.



Identity types for display maps

In fact, refl has the left lifting property w.r.t. all display maps.



Conclusion

Identity types factor $\Delta\colon A \to A \times A$ as

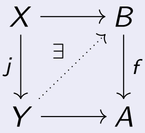
$$A \xrightarrow{\text{refl}} \mathsf{Id}_A \xrightarrow{q} A \times A$$

where q is a display map and refl lifts against all display maps.

Weak factorization systems

Definition

We say $j \boxslash f$ if any commutative square



admits a (non-unique) diagonal filler.

Weak factorization systems

Definition

We say $j \boxslash f$ if any commutative square

$$\begin{array}{ccc} X & \longrightarrow & B \\ {\scriptstyle j}\downarrow & \exists \nearrow & \downarrow{\scriptstyle f} \\ Y & \longrightarrow & A \end{array}$$

admits a (non-unique) diagonal filler.

- $\mathcal{J}^{\boxslash} = \{\, f \mid j \boxslash f \quad \forall j \in \mathcal{J} \,\}$
- ${}^{\boxslash}\mathcal{F} = \{\, j \mid j \boxslash f \quad \forall f \in \mathcal{F} \,\}$

Weak factorization systems

Definition

We say $j \boxslash f$ if any commutative square

$$\begin{array}{ccc} X & \longrightarrow & B \\ {\scriptstyle j}\downarrow & \overset{\exists}{\nearrow} & \downarrow{\scriptstyle f} \\ Y & \longrightarrow & A \end{array}$$

admits a (non-unique) diagonal filler.

- $\mathcal{J}^{\boxslash} = \{\, f \mid j \boxslash f \quad \forall j \in \mathcal{J} \,\}$
- ${}^{\boxslash}\mathcal{F} = \{\, j \mid j \boxslash f \quad \forall f \in \mathcal{F} \,\}$

Definition

A weak factorization system in a category is $(\mathcal{J}, \mathcal{F})$ such that

1. $\mathcal{J} = {}^{\boxslash}\mathcal{F}$ and $\mathcal{F} = \mathcal{J}^{\boxslash}$.
2. Every morphism factors as $f \circ j$ for some $f \in \mathcal{F}$ and $j \in \mathcal{J}$.

General factorizations

Theorem (Gambino–Garner)

In a display map category that models identity types, any morphism $g\colon A \to B$ factors as

$$A \xrightarrow{j} Ng \xrightarrow{f} B$$

where f is a display map, and j lifts against all display maps.

General factorizations

Theorem (Gambino–Garner)

In a display map category that models identity types, any morphism $g\colon A \to B$ factors as

$$A \xrightarrow{j} Ng \xrightarrow{f} B$$

where f is a display map, and j lifts against all display maps.

$$(y\colon B) \vdash Ng(y) :\equiv \mathsf{hfiber}(g, y) :\equiv \sum_{x\colon A} (g(x) = y)$$

is the type-theoretic mapping path space.

The identity type wfs

Corollary (Gambino-Garner)

In a type theory with identity types,

$$\left({}^{\boxslash}(\textit{display maps}), ({}^{\boxslash}(\textit{display maps}))^{\boxslash}\right)$$

is a weak factorization system.

The identity type wfs

Corollary (Gambino-Garner)

In a type theory with identity types,

$$\left({}^{\boxslash}(\textit{display maps}), ({}^{\boxslash}(\textit{display maps}))^{\boxslash} \right)$$

is a weak factorization system.

This behaves very much like (acyclic cofibrations, fibrations):

- Dependent types are like fibrations (recall "transport").
- Every map in ${}^{\boxslash}$(display maps) is an equivalence; in fact, the inclusion of a deformation retract.

Modeling identity types

Conversely:

Theorem (Awodey–Warren,Garner–van den Berg)

In a display map category, if

$$\left({}^{\boxslash}(\textit{display maps}), ({}^{\boxslash}(\textit{display maps}))^{\boxslash}\right)$$

is a "pullback-stable" weak factorization system, then the category (almost) models identity types.*

identity types $\longleftrightarrow$ weak factorization systems

Model categories

Definition (Quillen)

A model category is a category **C** with limits and colimits and three classes of maps:

- $\mathcal{C}$ = cofibrations
- $\mathcal{F}$ = fibrations
- $\mathcal{W}$ = weak equivalences

such that

1. $\mathcal{W}$ has the 2-out-of-3 property.
2. $(\mathcal{C} \cap \mathcal{W}, \mathcal{F})$ and $(\mathcal{C}, \mathcal{F} \cap \mathcal{W})$ are weak factorization systems.

Type-theoretic model categories

Corollary

Let $\mathcal{M}$ be a model category such that

1. $\mathcal{M}$ (as a category) is locally cartesian closed.
2. $\mathcal{M}$ is right proper.
3. The cofibrations are the monomorphisms.

Then $\mathcal{M}$ (almost*) models type theory with dependent sums, dependent products, and identity types.

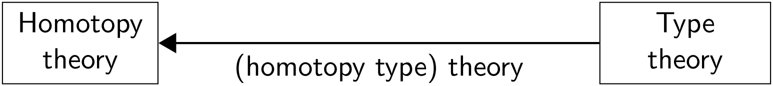
Type-theoretic model categories

Corollary

Let $\mathcal{M}$ be a model category such that

1. $\mathcal{M}$ (as a category) is locally cartesian closed.
2. $\mathcal{M}$ is right proper.
3. The cofibrations are the monomorphisms.

Then $\mathcal{M}$ (almost*) models type theory with dependent sums, dependent products, and identity types.



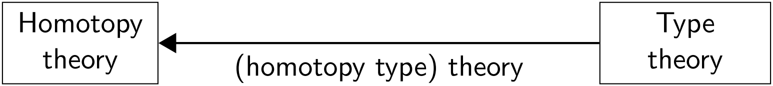
Type-theoretic model categories

Corollary

Let $\mathcal{M}$ be a model category such that

1. $\mathcal{M}$ (as a category) is locally cartesian closed.
2. $\mathcal{M}$ is right proper.
3. The cofibrations are the monomorphisms.

Then $\mathcal{M}$ (almost*) models type theory with dependent sums, dependent products, and identity types.



Examples

- Simplicial sets with the Quillen model structure.

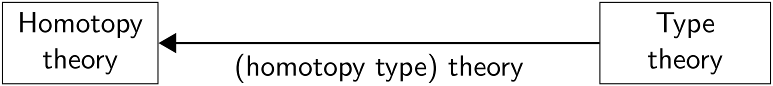
Type-theoretic model categories

Corollary

Let $\mathcal{M}$ be a model category such that

1. $\mathcal{M}$ (as a category) is locally cartesian closed.
2. $\mathcal{M}$ is right proper.
3. The cofibrations are the monomorphisms.

Then $\mathcal{M}$ (almost*) models type theory with dependent sums, dependent products, and identity types.



Examples

- Simplicial sets with the Quillen model structure.
- Any injective model structure on simplicial presheaves.

Homotopy type theory in categories

$$(x\colon A) \vdash p\colon \mathrm{isProp}(B(x))$$

$$\iff (x\colon A), (u\colon B(x)), (v\colon B(x)) \vdash (p_{u,v}\colon (u = v))$$

$\iff$ The path object $P_A B$ has a section in $\mathcal{M}/A$

$\iff$ Any two maps into B are homotopic over A

Homotopy type theory in categories

$$(x\colon A) \vdash p\colon \mathsf{isProp}(B(x))$$
$$\iff (x\colon A), (u\colon B(x)), (v\colon B(x)) \vdash (p_{u,v}\colon (u = v))$$
$\iff$ The path object $P_A B$ has a section in $\mathcal{M}/A$

$\iff$ Any two maps into B are homotopic over A

$$(x\colon A) \vdash p\colon \mathsf{isContr}(B(x))$$
$$\iff (x\colon A) \vdash p\colon \mathsf{isProp}(B(x)) \times B(x)$$
$\iff$ Any two maps into B are homotopic over A and $B \twoheadrightarrow A$ has a section

$\iff$ $B \twoheadrightarrow A$ is an acyclic fibration

Homotopy type theory in categories

For $f\colon A \to B$,

$$\begin{aligned} \vdash\ p\colon \mathsf{isEquiv}(f) &\iff \vdash \prod_{y\colon B} \mathsf{isContr}(\mathsf{hfiber}(f,y)) \\ &\iff (y\colon B) \vdash \mathsf{isContr}(\mathsf{hfiber}(f,y)) \\ &\iff \mathsf{hfiber}(f) \twoheadrightarrow A \text{ is an acyclic fibration} \\ &\iff f \text{ is a (weak) equivalence} \end{aligned}$$

(Recall hfiber is the factorization $A \to Nf \twoheadrightarrow B$ of f.)

Homotopy type theory in categories

For $f \colon A \to B$,

$$
\begin{aligned}
\vdash\ p\colon \mathsf{isEquiv}(f) &\iff \vdash \prod_{y\colon B} \mathsf{isContr}(\mathsf{hfiber}(f,y)) \\
&\iff (y\colon B) \vdash \mathsf{isContr}(\mathsf{hfiber}(f,y)) \\
&\iff \mathsf{hfiber}(f) \twoheadrightarrow A \text{ is an acyclic fibration} \\
&\iff f \text{ is a (weak) equivalence}
\end{aligned}
$$

(Recall hfiber is the factorization $A \to Nf \twoheadrightarrow B$ of f.)

Conclusion

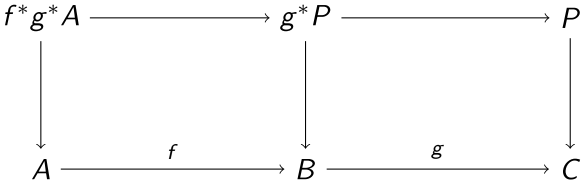
Any theorem about "equivalences" that we can prove in type theory yields a conclusion about weak equivalences in appropriate model categories.

Coherence

Another Problem

Type theory is even stricter than 1-categories!

Recall that substitution is pullback.



$a\colon A \vdash P(g(f(a)))$ $\quad$ $b\colon B \vdash P(g(b))$ $\quad$ $c\colon C \vdash P(c)$

Coherence

Another Problem

Type theory is even stricter than 1-categories!

Recall that substitution is pullback.

$$
\begin{array}{ccc}
(g \circ f)^* A & \longrightarrow & P \\
\downarrow & & \downarrow \\
A & \xrightarrow{\;g \circ f\;} & C
\end{array}
$$

$a \colon A \vdash P((g \circ f)(a))$ $\qquad$ $c \colon C \vdash P(c)$

Coherence

Another Problem

Type theory is even stricter than 1-categories!

Recall that substitution is pullback.

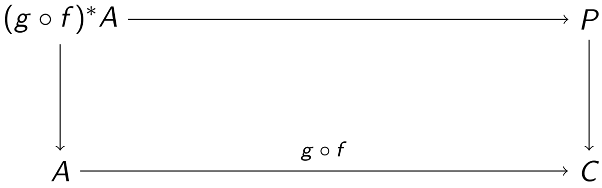
$$\begin{array}{ccc} (g \circ f)^* A & \longrightarrow & P \\ \downarrow & & \downarrow \\ A & \xrightarrow{g \circ f} & C \end{array}$$

$a \colon A \vdash P((\lambda x.g(f(x)))(a))$ $\qquad$ $c \colon C \vdash P(c)$

Another Problem

Type theory is even stricter than 1-categories!

Recall that substitution is pullback.



$a \colon A \vdash P(g(f(a)))$

$c \colon C \vdash P(c)$

Coherence

Another Problem

Type theory is even stricter than 1-categories!

Recall that substitution is pullback.

$$\begin{array}{ccc} (g \circ f)^* A & \longrightarrow & P \\ \downarrow & & \downarrow \\ A & \xrightarrow{g \circ f} & C \end{array}$$

$a \colon A \vdash P(g(f(a)))$ $\qquad\qquad$ $c \colon C \vdash P(c)$

But, of course, $f^* g^* P$ is only isomorphic to $(g \circ f)^* P$.

Coherence with a universe

There are several resolutions; perhaps the cleanest is:

Solution (Voevodsky)

Represent dependent types by their classifying maps into a universe object.

Now substitution is composition, which is strictly associative (in our model category):

$$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{P} U$$

$$A \xrightarrow{\quad g \circ f \quad} C \xrightarrow{P} U$$

We needed a universe object anyway, to model the type Type and prove univalence.

Coherence with a universe

There are several resolutions; perhaps the cleanest is:

Solution (Voevodsky)

Represent dependent types by their classifying maps into a universe object.

Now substitution is composition, which is strictly associative (in our model category):

$$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{P} U$$

$$A \xrightarrow{\quad g \circ f \quad} C \xrightarrow{P} U$$

We needed a universe object anyway, to model the type Type and prove univalence.

New problem

Need very strict models for universe objects.

Outline

Representing fibrations

(Following Kapulkin–Lumsdaine–Voevodsky)

Goal

A universe object in simplicial sets giving coherence and univalence.

Representing fibrations

(Following Kapulkin–Lumsdaine–Voevodsky)

Goal

A universe object in simplicial sets giving coherence and univalence.

Simplicial sets are a presheaf category, so there is a standard trick to build representing objects.

$$U_n \cong \mathrm{Hom}(\Delta^n, U) \simeq \{\text{fibrations over } \Delta^n\}$$

Representing fibrations

(Following Kapulkin–Lumsdaine–Voevodsky)

Goal

A universe object in simplicial sets giving coherence and univalence.

Simplicial sets are a presheaf category, so there is a standard trick to build representing objects.

$$U_n \cong \mathrm{Hom}(\Delta^n, U) \simeq \{\text{fibrations over } \Delta^n\}$$

But $n \mapsto \{\text{fibrations over } \Delta^n\}$ is only a pseudofunctor; we need to rigidify it.

Well-ordered fibrations

A technical device (Voevodsky)

A well-ordered Kan fibration is a Kan fibration $p\colon E \to B$ together with, for every $x \in B_n$, a well-ordering on $p^{-1}(x) \subseteq E_n$.

Well-ordered fibrations

A technical device (Voevodsky)

A well-ordered Kan fibration is a Kan fibration $p\colon E \to B$ together with, for every $x \in B_n$, a well-ordering on $p^{-1}(x) \subseteq E_n$.

Two well-ordered Kan fibrations are isomorphic in at most one way which preserves the orders.

Well-ordered fibrations

A technical device (Voevodsky)

A well-ordered Kan fibration is a Kan fibration $p \colon E \to B$ together with, for every $x \in B_n$, a well-ordering on $p^{-1}(x) \subseteq E_n$.

Two well-ordered Kan fibrations are isomorphic in at most one way which preserves the orders.

Definition

$$U_n := \left\{ X \twoheadrightarrow \Delta^n \text{ a well-ordered fibration} \right\} \Big/ _{\text{ordered } \cong}$$

Well-ordered fibrations

A technical device (Voevodsky)

A well-ordered Kan fibration is a Kan fibration $p\colon E \to B$ together with, for every $x \in B_n$, a well-ordering on $p^{-1}(x) \subseteq E_n$.

Two well-ordered Kan fibrations are isomorphic in at most one way which preserves the orders.

Definition

$$U_n := \left\{ X \twoheadrightarrow \Delta^n \text{ a well-ordered fibration} \right\} \Big/_{\text{ordered } \cong}$$

$$\widetilde{U}_n := \left\{ (X, x) \,\middle|\, X \twoheadrightarrow \Delta^n \text{ well-ordered fibration, } x \in X_n \right\} \Big/_{\text{ordered } \cong}$$

Well-ordered fibrations

A technical device (Voevodsky)

A well-ordered Kan fibration is a Kan fibration $p \colon E \to B$ together with, for every $x \in B_n$, a well-ordering on $p^{-1}(x) \subseteq E_n$.

Two well-ordered Kan fibrations are isomorphic in at most one way which preserves the orders.

Definition

$$U_n := \left\{ X \twoheadrightarrow \Delta^n \text{ a well-ordered fibration} \right\} \Big/_{\text{ordered} \cong}$$

$$\widetilde{U}_n := \left\{ (X, x) \;\middle|\; X \twoheadrightarrow \Delta^n \text{ well-ordered fibration, } x \in X_n \right\} \Big/_{\text{ordered} \cong}$$

(with some size restriction, to make them sets).

The universal Kan fibration

Theorem

The forgetful map $\widetilde{U} \to U$ is a Kan fibration.

Proof.

A map $E \to B$ is a Kan fibration if and only if every pullback

$$\begin{array}{ccc} b^*E & \longrightarrow & E \\ \downarrow & \lrcorner & \downarrow \\ \Delta^n & \xrightarrow[b]{} & B \end{array}$$

is such, since the horns $\Lambda^n_k \hookrightarrow \Delta^n$ have codomain Δ^n. □

The universal Kan fibration

Theorem

The forgetful map $\widetilde{U} \to U$ is a Kan fibration.

Proof.

A map $E \to B$ is a Kan fibration if and only if every pullback

$$\begin{array}{ccc} b^*E & \longrightarrow & E \\ \downarrow & \lrcorner & \downarrow \\ \Delta^n & \xrightarrow[b]{} & B \end{array}$$

is such, since the horns $\Lambda^n_k \hookrightarrow \Delta^n$ have codomain Δ^n. □

Thus, of course, every pullback of $\widetilde{U} \to U$ is a Kan fibration.

The universal Kan fibration

Theorem

Every (small) Kan fibration $E \to B$ is some pullback of $\widetilde{U} \to U$:

$$\begin{array}{ccc} E & \longrightarrow & \widetilde{U} \\ \downarrow & \lrcorner & \downarrow \\ B & \longrightarrow & U \end{array}$$

Proof.

Choose a well-ordering on each fiber, and map $x \in B_n$ to the isomorphism class of the well-ordered fibration $b^*(E) \twoheadrightarrow \Delta^n$. $\square$

The universal Kan fibration

Theorem

Every (small) Kan fibration $E \to B$ is some pullback of $\widetilde{U} \to U$:

$$\begin{array}{ccc} E & \longrightarrow & \widetilde{U} \\ \downarrow & \lrcorner & \downarrow \\ B & \longrightarrow & U \end{array}$$

Proof.

Choose a well-ordering on each fiber, and map $x \in B_n$ to the isomorphism class of the well-ordered fibration $b^*(E) \twoheadrightarrow \Delta^n$. □

It is essential that we have actual pullbacks here, not just homotopy pullbacks.

Type theory in the universe

Let the size-bound for U be inaccessible (a Grothendieck universe). Then small fibrations are closed under all categorical constructions.

Type theory in the universe

Let the size-bound for U be inaccessible (a Grothendieck universe). Then small fibrations are closed under all categorical constructions.

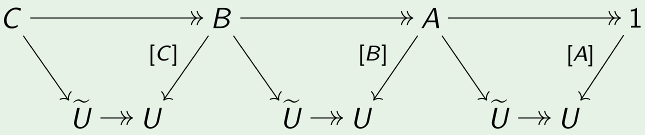
Now we can interpret type theory with coherence, using morphisms into U for dependent types.

Example

A context

$$(x \colon A), (y \colon B(x)), (z \colon C(x,y))$$

becomes a sequence of fibrations together with classifying maps:



in which each trapezoid is a pullback.

Strict cartesian products

Every type-theoretic operation can be done once over U, then implemented by composition.

Example (Cartesian product)

- Pull $\widetilde{U}$ back to $U \times U$ along the two projections π_1, π_2.

Strict cartesian products

Every type-theoretic operation can be done once over U, then implemented by composition.

Example (Cartesian product)

- Pull $\widetilde{U}$ back to $U \times U$ along the two projections π_1, π_2.
- Their fiber product over $U \times U$ admits a classifying map:

$$\begin{array}{ccc} (\pi_1^* \widetilde{U}) \times_{U \times U} (\pi_2^* \widetilde{U}) & \longrightarrow & \widetilde{U} \\ \downarrow & \lrcorner & \downarrow \\ U \times U & \xrightarrow[{[\times]}]{} & U \end{array}$$

Strict cartesian products

Every type-theoretic operation can be done once over U, then implemented by composition.

Example (Cartesian product)

- Pull $\widetilde{U}$ back to $U \times U$ along the two projections π_1, π_2.
- Their fiber product over $U \times U$ admits a classifying map:

$$
\begin{array}{ccc}
(\pi_1^* \widetilde{U}) \times_{U \times U} (\pi_2^* \widetilde{U}) & \longrightarrow & \widetilde{U} \\
\downarrow & \lrcorner & \downarrow \\
U \times U & \xrightarrow[{[\times]}]{} & U
\end{array}
$$

- Define the product of $[A] \colon X \to U$ and $[B] \colon X \to U$ to be

$$X \xrightarrow{([A],[B])} U \times U \xrightarrow{[\times]} U$$

This has strict substitution.

Nested universes

Problem

So far the object U lives outside the type theory.
We want it inside, giving a universe type "Type" and univalence.

Nested universes

Problem

So far the object U lives outside the type theory.
We want it inside, giving a universe type "Type" and univalence.

Solution

Let U' be a bigger universe. If U is U'-small and fibrant, then it has a classifying map:

$$\begin{array}{ccc} U & \longrightarrow & \widetilde{U}' \\ \big\downarrow & \lrcorner & \big\downarrow \\ 1 & \xrightarrow[u]{} & U' \end{array}$$

and the type theory defined using U' has a universe type u.

U is fibrant

Theorem

U is fibrant.

U is fibrant

Theorem

U is fibrant.

Outline of proof.

$$\begin{array}{ccc} \Lambda^n_k & \xrightarrow{f} & U \\ {\scriptstyle j}\big\downarrow & \overset{?}{\nearrow} & \\ \Delta^n & & \end{array}$$

U is fibrant

Theorem

U is fibrant.

Outline of proof.

$$\begin{array}{ccc} \Lambda^n_k & \xrightarrow{f} & U \\ {\scriptstyle j}\downarrow & \overset{?}{\nearrow} & \\ \Delta^n & & \end{array}$$

With hard work, we can extend $f^*\widetilde{U}$ to a fibration over Δ^n:

$$\begin{array}{ccc} f^*\widetilde{U} & \longrightarrow & P \\ \downarrow & \lrcorner & \downarrow \\ \Lambda^n_k & \xrightarrow[j]{} & \Delta^n \end{array}$$

U is fibrant

Theorem

U is fibrant.

Outline of proof.

$$\begin{array}{ccc} \Lambda^n_k & \xrightarrow{f} & U \\ {\scriptstyle j}\downarrow & \overset{?}{\nearrow} & \\ \Delta^n & & \end{array}$$

With hard work, we can extend $f^*\widetilde{U}$ to a fibration over Δ^n:

$$\begin{array}{ccc} f^*\widetilde{U} & \longrightarrow & P \\ \downarrow & \lrcorner & \downarrow \\ \Lambda^n_k & \xrightarrow[j]{} & \Delta^n \end{array}$$

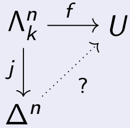
and extend the well-ordering of $f^*\widetilde{U}$ to P, yielding $g\colon \Delta^n \to U$ with $gj = f$ (and $g^*\widetilde{U} \cong P$). $\square$

U is fibrant

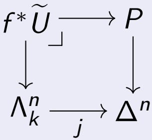
Theorem

U is fibrant.

Outline of proof.



With hard work, we can extend $f^*\widetilde{U}$ to a fibration over Δ^n:



and extend the well-ordering of $f^*\widetilde{U}$ to P, yielding $g\colon \Delta^n \to U$ with $gj = f$ (and $g^*\widetilde{U} \cong P$). □

Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ *is the pullback of some fibration over* Δ^n.

Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ is the pullback of some fibration over Δ^n.

Proof.

- Let $P' \subseteq P$ be a minimal subfibration.

Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ is the pullback of some fibration over Δ^n.

Proof.

- Let $P' \subseteq P$ be a minimal subfibration.
- There is a retraction $P \to P'$ that is an acyclic fibration.

Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ is the pullback of some fibration over Δ^n.

Proof.

- Let $P' \subseteq P$ be a minimal subfibration.
- There is a retraction $P \to P'$ that is an acyclic fibration.
- Since Λ^n_k is contractible, the minimal fibration $P' \to \Lambda^n_k$ is isomorphic to a trivial bundle $\Lambda^n_k \times F \to \Lambda^n_k$.

Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ is the pullback of some fibration over Δ^n.

Proof.

- Let $P' \subseteq P$ be a minimal subfibration.
- There is a retraction $P \to P'$ that is an acyclic fibration.
- Since Λ^n_k is contractible, the minimal fibration $P' \to \Lambda^n_k$ is isomorphic to a trivial bundle $\Lambda^n_k \times F \to \Lambda^n_k$.

$$\begin{array}{ccc} P & & \\ \downarrow & & \\ P' \cong \Lambda^n_k \times F & & \\ \downarrow & & \\ \Lambda^n_k & \xrightarrow[j]{} & \Delta^n \end{array}$$

□

Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ is the pullback of some fibration over Δ^n.

Proof.

- Let $P' \subseteq P$ be a minimal subfibration.
- There is a retraction $P \to P'$ that is an acyclic fibration.
- Since Λ^n_k is contractible, the minimal fibration $P' \to \Lambda^n_k$ is isomorphic to a trivial bundle $\Lambda^n_k \times F \to \Lambda^n_k$.

$$\begin{array}{ccc} P & & \\ \downarrow & & \\ P' \cong \Lambda^n_k \times F & \xrightarrow[j \times F]{} & \Delta^n \times F \\ \downarrow & \lrcorner & \downarrow \\ \Lambda^n_k & \xrightarrow[j]{} & \Delta^n \end{array}$$

□

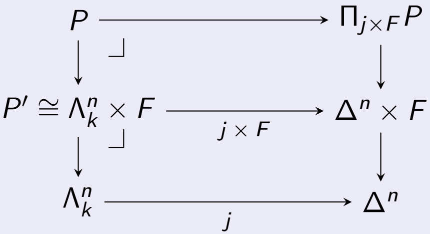
Extending fibrations

Lemma

Any fibration $P \to \Lambda^n_k$ is the pullback of some fibration over Δ^n.

Proof.

- Let $P' \subseteq P$ be a minimal subfibration.
- There is a retraction $P \to P'$ that is an acyclic fibration.
- Since Λ^n_k is contractible, the minimal fibration $P' \to \Lambda^n_k$ is isomorphic to a trivial bundle $\Lambda^n_k \times F \to \Lambda^n_k$.



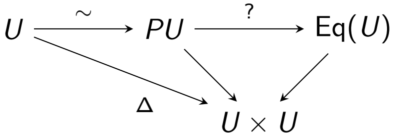
□

Univalence

We want to show that $PU \to \mathrm{Eq}(U)$ is an equivalence:

$$\begin{array}{ccc} PU & \xrightarrow{?} & \mathrm{Eq}(U) \\ & \searrow \quad \swarrow & \\ & U \times U & \end{array}$$

We want to show that $PU \to \mathrm{Eq}(U)$ is an equivalence:

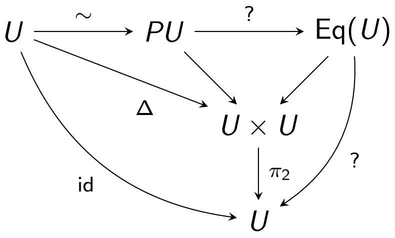


It suffices to show:

1. The composite $U \to \mathrm{Eq}(U)$ is an equivalence.

Univalence

We want to show that $PU \to \mathrm{Eq}(U)$ is an equivalence:

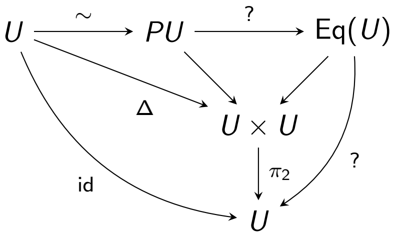


It suffices to show:

1. The composite $U \to \mathrm{Eq}(U)$ is an equivalence.
2. The projection $\mathrm{Eq}(U) \to U$ is an equivalence.

Univalence

We want to show that $PU \to \mathrm{Eq}(U)$ is an equivalence:



It suffices to show:

1. The composite $U \to \mathrm{Eq}(U)$ is an equivalence.
2. The projection $\mathrm{Eq}(U) \to U$ is an equivalence.
3. The projection $\mathrm{Eq}(U) \to U$ is an acyclic fibration.

Univalence

By representability, a commutative square

$$
\begin{array}{ccc}
\partial\Delta^n & \longrightarrow & \mathrm{Eq}(U) \\
{\scriptstyle i}\downarrow & & \downarrow \\
\Delta^n & \longrightarrow & U
\end{array}
$$

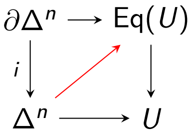
corresponds to a diagram

$$
\begin{array}{ccccc}
E_1 & & & & \\
 & \searrow & & & \\
\downarrow & & E_2 & \longrightarrow & \overline{E}_2 \\
 & \swarrow & \lrcorner & & \downarrow \\
\partial\Delta^n & & \xrightarrow[i]{} & & \Delta^n
\end{array}
$$

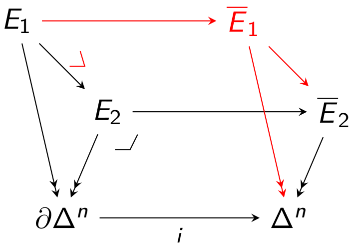
with $E_1 \to E_2$ an equivalence.

Univalence

By representability, a commutative square with a lift

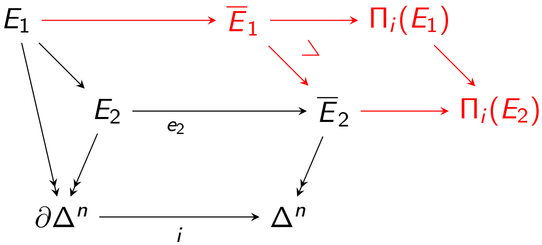


corresponds to a diagram



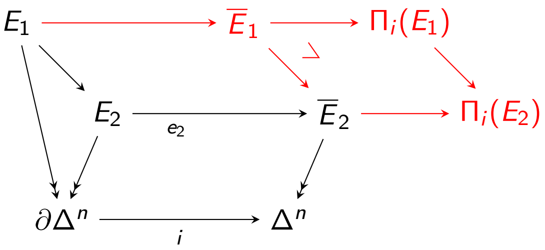
with $E_1 \to E_2$ and $\overline{E}_1 \to \overline{E}_2$ equivalences.

Univalence



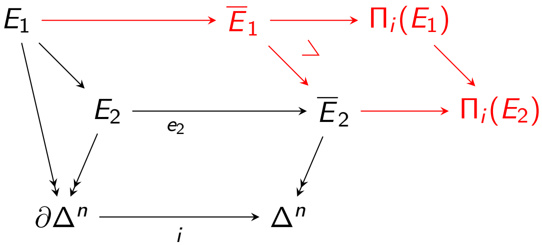
E_1
$\overline{E}_1$
$\Pi_i(E_1)$
E_2
e_2
$\overline{E}_2$
$\Pi_i(E_2)$
$\partial\Delta^n$
i
Δ^n

Univalence



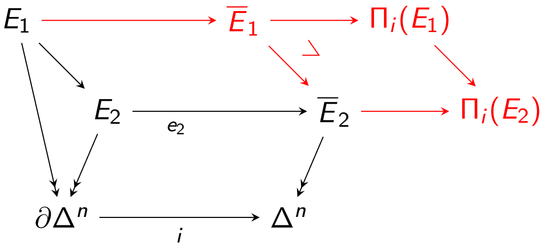
- By factorization, consider separately the cases when $E_1 \to E_2$ is (1) an acyclic fibration or (2) an acyclic cofibration.

Univalence



- By factorization, consider separately the cases when $E_1 \to E_2$ is (1) an acyclic fibration or (2) an acyclic cofibration.
- (1) $\overline{E}_1 \to \overline{E}_2$ is an acyclic fibration (Π_i preserves such).

Univalence



- By factorization, consider separately the cases when $E_1 \to E_2$ is (1) an acyclic fibration or (2) an acyclic cofibration.
- (1) $\overline{E}_1 \to \overline{E}_2$ is an acyclic fibration (Π_i preserves such).
- (2) $\overline{E}_1$ is a deformation retract of $\overline{E}_2$.

Outline

1. Homotopy type theory in model categories
2. The universal Kan fibration
3. Models in $(\infty, 1)$-toposes

$(\infty, 1)$-toposes

Definition

An $(\infty, 1)$-topos is an $(\infty, 1)$-category that is a left-exact localization of an $(\infty, 1)$-presheaf category.

Examples

- ∞-groupoids (plays the role of the 1-topos Set)
- Parametrized homotopy theory over any space X
- G-equivariant homotopy theory for any group G
- ∞-sheaves/stacks on any space
- "Smooth ∞-groupoids" (or "algebraic" etc.)

Univalence in categories

Definition (Rezk)

An object classifier in an $(\infty, 1)$-category $\mathcal{C}$ is a morphism $\widetilde{U} \to U$ such that pullback

$$\begin{array}{ccc} B & \longrightarrow & \widetilde{U} \\ \downarrow & \lrcorner & \downarrow \\ A & \longrightarrow & U \end{array}$$

induces an equivalence of ∞-groupoids

$$\mathrm{Hom}(A, U) \xrightarrow{\simeq} \mathrm{Core}(\mathcal{C}/A)_{\mathsf{small}}$$

("Core" is the maximal sub-∞-groupoid.)

$(\infty, 1)$-toposes

Theorem (Rezk)

An $(\infty, 1)$-category $\mathcal{C}$ is an $(\infty, 1)$-topos if and only if

1. *$\mathcal{C}$ is locally presentable.*
2. *$\mathcal{C}$ is locally cartesian closed.*
3. *κ-compact objects have object classifiers for $\kappa \gg 0$.*

$(\infty, 1)$-toposes

Theorem (Rezk)

An $(\infty, 1)$-category $\mathcal{C}$ is an $(\infty, 1)$-topos if and only if

1. *$\mathcal{C}$ is locally presentable.*
2. *$\mathcal{C}$ is locally cartesian closed.*
3. *κ-compact objects have object classifiers for $\kappa \gg 0$.*

Corollary

If a combinatorial model category $\mathcal{M}$ interprets dependent type theory as before (i.e. it is locally cartesian closed, right proper, and the cofibrations are the monomorphisms), and contains universes for κ-compact objects that satisfy the univalence axiom, then the $(\infty, 1)$-category that it presents is an $(\infty, 1)$-topos.

$(\infty, 1)$-toposes

Conjecture

Every $(\infty, 1)$-topos can be presented by a model category which interprets dependent type theory with the univalence axiom.

Homotopy type theory is the internal logic of $(\infty, 1)$-toposes.

$(\infty, 1)$-toposes

Conjecture

Every $(\infty, 1)$-topos can be presented by a model category which interprets dependent type theory with the univalence axiom.

Homotopy type theory is the internal logic of $(\infty, 1)$-toposes.

If this is true, then anything we prove in homotopy type theory (which we can also verify with a computer) will automatically be true internally to any $(\infty, 1)$-topos. The "constructive core" of homotopy theory should be provable in this way, in a uniform way for "all homotopy theories".

Status of the conjecture

$\infty Gpd \dashrightarrow (\infty, 1)$-presheaves $\qquad (\infty, 1)$-toposes

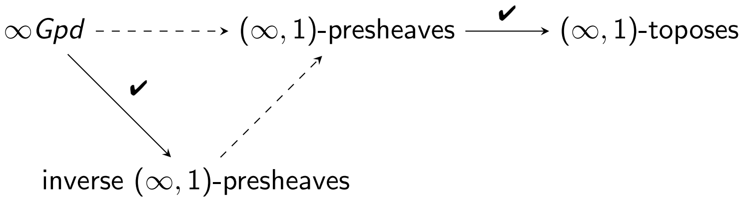
Status of the conjecture

$$\infty Gpd \dashrightarrow (\infty, 1)\text{-presheaves} \dashrightarrow (\infty, 1)\text{-toposes}$$

Status of the conjecture

$$\infty Gpd \dashrightarrow (\infty, 1)\text{-presheaves} \xrightarrow{\checkmark} (\infty, 1)\text{-toposes}$$

Status of the conjecture



Status of the conjecture