# Type 2-theories

Michael Shulman

joint work with Dan Licata and Mitchell Riley

University of San Diego

April 12, 2018

HoTTEST

# Outline

1. Motivation: internal languages
2. Unary type 2-theories
3. Simple type 2-theories
4. Dependent type 2-theories

# The internal language hypothesis

**The Internal Language Hypothesis**

Homotopy type theory is an internal language for $(\infty, 1)$-toposes.

# The internal language hypothesis

**The Internal Language Hypothesis**

Homotopy type theory is an internal language for $(\infty, 1)$-toposes.

I propose this as analogous to the Homotopy Hypothesis, Stabilization Hypothesis, Cobordism Hypothesis, etc. from higher category theory.

> *"One can regard the above hypothesis, and those to follow, either as a conjecture pending a general definition. . . or as a feature one might desire of such a definition."*
>
> – John Baez and Jim Dolan, *Higher-Dimensional Algebra and Topological Quantum Field Theory*

# Applications of the internal language hypothesis

**The Internal Language Hypothesis**

Homotopy type theory is an internal language for $(\infty, 1)$-toposes.

Some theorems that are proven:

- $(\infty, 1)$-toposes are presented by CwAs w/ HITs[1]
- *Some* $\infty$-object classifiers are presented by CwA universes[2]
- Lex $(\infty, 1)$-categories are equivalent to CwAs w/ $\Sigma, \mathrm{Id}$[3]

Theorems we still need to prove:

- The syntax of type theory is the initial CwA w/ ...
- *All* $\infty$-object classifiers are presented by CwA universes

Definitions we still need to make:

- What is a general notion of "higher inductive type"?
- What is an "elementary $(\infty, 1)$-topos"?

[1] Cisinski, Gepner–Kock, Lumsdaine–S.

[2] Voevodsky, S.

[3] Kapulkin–Lumsdaine, Kapulkin–Szumiło

# Applications of the internal language hypothesis

**The Internal Language Hypothesis**

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

Definitions we should not make:

- Homotopy type theory consists of what's true in simplicial sets.
- Homotopy type theory consists of what's true in cubical sets.

**If you will forgive me saying it again**

- One model is not enough!
- Please don't talk about "the intended model"!

**Why?**

One reason: applications of homotopy type theory to new results in classical homotopy theory are much closer to our reach if we go through other models (e.g. Blakers–Massey in Goodwillie calculus).

# What is an $(\infty, 1)$-topos?

To a homotopy type theorist, the Internal Language Hypothesis can be a "working definition" of an $(\infty, 1)$-topos: a collection of objects and morphisms that can interpret the types and terms of HoTT.

**Examples**

- $\infty\mathcal{G}pd$: types are $\infty$-groupoids ("spaces")
  (The $\infty$-version of the 1-topos $\mathrm{Set}$)
- $\infty\mathcal{G}pd^{C^{\mathrm{op}}}$: types are presheaves of $\infty$-groupoids on $C$
- $\mathcal{S}h(X)$: types are sheaves of $\infty$-groupoids on $X$

But the situation for functors between $(\infty, 1)$-toposes is subtler.

# $(\infty, 1)$-geometric morphisms

**Definition**

A logical functor $L : \mathcal{E} \to \mathcal{S}$ preserves all relevant structure.

**Definition**

A geometric morphism $p : \mathcal{E} \to \mathcal{S}$ is an adjoint pair $p^* : \mathcal{F} \rightleftarrows \mathcal{E} : p_*$ such that $p^*$ preserves finite limits.

**Examples**

- $f : C \to D$ a functor, $f^* : \infty\mathcal{G}pd^{D^{\mathrm{op}}} \rightleftarrows \infty\mathcal{G}pd^{C^{\mathrm{op}}} : \mathrm{Ran}_f$ is a geometric morphism $\infty\mathcal{G}pd^{C^{\mathrm{op}}} \to \infty\mathcal{G}pd^{D^{\mathrm{op}}}$.
- If $f : X \to Y$ is a continuous map, there is a geometric morphism $\mathcal{S}h(f) : \mathcal{S}h(X) \to \mathcal{S}h(Y)$.
- Any $\mathcal{E}$ has a unique geometric morphism $p : \mathcal{E} \to \infty\mathcal{G}pd$:
    - $p_*(A) = \mathcal{E}(1, A)$ is the global sections
    - $p^*(X) = \coprod_X 1$ is a discrete or constant object on $X$.

# Internal languages for geometric morphisms?

## Fact

A lot of interesting theorems in $(\infty, 1)$-topos theory are not about just one topos, but about diagrams of toposes and geometric morphisms between them.

## Example

A $(\infty, 1)$-topos $\mathcal{E}$ is...

- $\infty$-connected if $p^* : \infty\mathcal{G}pd \to \mathcal{E}$ is fully faithful
- locally $\infty$-connected if $p^*$ has a left adjoint
- $\infty$-compact if $p_*$ preserves filtered colimits
- ...

## Problem

Is there a version of homotopy type theory that can be an internal language for diagrams of $(\infty, 1)$-toposes and geometric morphisms?

# Applications of a theory in progress

We claim that yes, there is such a type theory, where the functors $p^*, p_*$ appear as higher modalities. The fully general and dependently typed version is still a work in progress, but already it has been specialized to various applications:

- Internal universes in topos models (L.-Orton-Pitts-Spitters '18)
  - One modality $\flat$
- Spatial and real-cohesive type theory (S. '17)
  - Three modalities $\int \dashv \flat \dashv \sharp$
- Differential cohesion (L.-S.-Gross-New-Paykin-R.-Wellen – work in progress)
  - *Six* modalities $\int \dashv \flat \dashv \sharp$ and $\Re \dashv \Im \dashv \&$.
- Type theory for parametrized pointed spaces and spectra (Finster-L.-Morehouse-R. – work in progress)
  - One self-adjoint modality $\natural \dashv \natural$
  - Non-cartesian "smash product" monoidal structure
- Directed type theory with cores and opposites (work in progress)

# Outline

1. Motivation: internal languages
2. Unary type 2-theories
3. Simple type 2-theories
4. Dependent type 2-theories

# First steps in modal type theory

Suppose we have one geometric morphism $p : \mathcal{E} \to \mathcal{S}$.
We might imagine a type theory with:

- An "$\mathcal{E}$-type theory" $\Gamma \vdash_{\mathcal{E}} s : A$
- A separate "$\mathcal{S}$-type theory" $\Delta \vdash_{\mathcal{S}} t : B$
- An operation $p^*$ making any $\mathcal{S}$-type into an $\mathcal{E}$-type
- An operation $p_*$ making any $\mathcal{E}$-type into an $\mathcal{S}$-type

# First steps in modal type theory

Suppose we have one geometric morphism $p : \mathcal{E} \to \mathcal{S}$.
We might imagine a type theory with:

- An "$\mathcal{E}$-type theory" $\Gamma \vdash_{\mathcal{E}} s : A$
- A separate "$\mathcal{S}$-type theory" $\Delta \vdash_{\mathcal{S}} t : B$
- An operation $p^*$ making any $\mathcal{S}$-type into an $\mathcal{E}$-type
- An operation $p_*$ making any $\mathcal{E}$-type into an $\mathcal{S}$-type
- Functoriality rules for $p^*$ and $p_*$
- Adjunction rules for $p^* \dashv p_*$

Suppose we have one geometric morphism $p : \mathcal{E} \to \mathcal{S}$.
We might imagine a type theory with:

- An "$\mathcal{E}$-type theory" $\Gamma \vdash_{\mathcal{E}} s : A$
- A separate "$\mathcal{S}$-type theory" $\Delta \vdash_{\mathcal{S}} t : B$
- An operation $p^*$ making any $\mathcal{S}$-type into an $\mathcal{E}$-type
- An operation $p_*$ making any $\mathcal{E}$-type into an $\mathcal{S}$-type
- Functoriality rules for $p^*$ and $p_*$
- Adjunction rules for $p^* \dashv p_*$
- Higher functoriality rules for $p^*$ and $p_*$ on homotopies
- Higher adjunction rules for homotopies
- Coherence laws
- More coherence laws...

# Design principles for type theory and category theory

| Type theory | Category theory |
| --- | --- |
| Types should be defined by introduction, elimination, $\beta$ and $\eta$ rules. | Objects should be defined by universal properties. |
| Good type theories satisfy canonicity and normalization. | Structures defined by universal properties are automatically fully coherent. |

# Example: Cartesian products

## Type theory

$$\frac{p : A \times B}{\pi_1(p) : A \qquad \pi_2(p) : B}$$

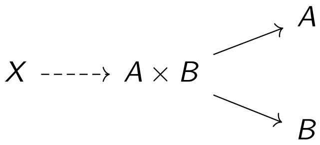
$$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$

$$\pi_1(a, b) = a$$
$$\pi_2(a, b) = b$$

$$p = (\pi_1(p), \pi_2(p))$$

## Category theory

$$A \leftarrow A \times B \rightarrow B$$




correct composites

uniqueness

# Example: Cartesian products

**Type theory**

$$\frac{p : A \times B}{\pi_1(p) : A \qquad \pi_2(p) : B}$$

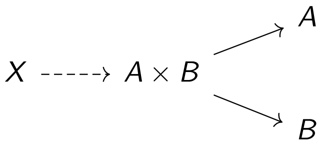
$$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$

$$\pi_1(a, b) = a$$
$$\pi_2(a, b) = b$$

$$p = (\pi_1(p), \pi_2(p))$$

**Category theory**

$$A \leftarrow A \times B \rightarrow B$$




correct composites

uniqueness

## Type theory

$$\frac{p : A \times B}{\pi_1(p) : A \qquad \pi_2(p) : B}$$

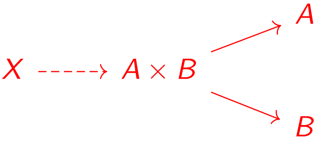
$$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$

$$\pi_1(a, b) = a$$
$$\pi_2(a, b) = b$$

$$p = (\pi_1(p), \pi_2(p))$$

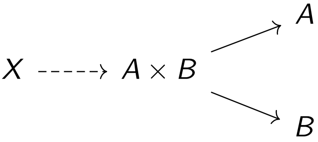
## Category theory

$$A \leftarrow A \times B \rightarrow B$$




correct composites

uniqueness

| Type theory | Category theory |
|---|---|
| $$\frac{p : A \times B}{\pi_1(p) : A \qquad \pi_2(p) : B}$$ | $$A \leftarrow A \times B \rightarrow B$$ |
| $$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$ |  |
| $$\pi_1(a, b) = a$$ $$\pi_2(a, b) = b$$ | correct composites |
| $$p = (\pi_1(p), \pi_2(p))$$ | uniqueness |

# Example: Cartesian products

## Type theory

$$\frac{p : A \times B}{\pi_1(p) : A \qquad \pi_2(p) : B}$$

$$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$

$$\pi_1(a, b) = a$$
$$\pi_2(a, b) = b$$

$$p = (\pi_1(p), \pi_2(p))$$

## Category theory

$$A \leftarrow A \times B \rightarrow B$$

$$X \dashrightarrow A \times B \quad \begin{matrix} \nearrow A \\ \searrow B \end{matrix}$$

correct composites

uniqueness

# Universal properties for adjunctions

**Definition**

A profunctor $\mathcal{E} \nrightarrow \mathcal{S}$ is a category $\mathcal{H}$ equipped with a functor $\mathcal{H} \to 2 = (0 \to 1)$, with fibers $\mathcal{H}_0 = \mathcal{S}$ and $\mathcal{H}_1 = \mathcal{E}$.

- Hom-sets $\mathcal{H}(X, A)$ of "heteromorphisms" for $X \in \mathcal{S}$, $A \in \mathcal{E}$
- With actions by arrows in $\mathcal{E}$ and $\mathcal{S}$

**Definition**

- A left representation of $\mathcal{H}$ at $X \in \mathcal{S}$ is $p^*X \in \mathcal{E}$ with an isomorphism $\mathcal{E}(p^*X, A) \cong \mathcal{H}(X, A)$.
- A right representation of $\mathcal{H}$ at $A \in \mathcal{E}$ is $p_*A \in \mathcal{S}$ with an isomorphism $\mathcal{S}(X, p_*A) \cong \mathcal{H}(X, A)$.

Insofar as they exist, we automatically have $p^* \dashv p_*$ since

$$\mathcal{E}(p^*X, A) \cong \mathcal{H}(X, A) \cong \mathcal{S}(X, p_*A).$$

# A hierarchy of type theories

Dependent type theory is very complicated, so we build up in stages.

1. Unary type theory: no dependency, only one type in context. Semantics in categories.

$$x : A \vdash s : B$$

2. Simple type theory: no dependency, multiple types in context. Semantics in categories with products, or multicategories.

$$x : A, y : B, z : C \vdash s : D$$

3. Dependent type theory: types can depend on previous ones. Semantics in lex categories (comprehension categories etc.)

$$x : A, y : B(x), z : C(x, y) \vdash s : D(x, y, z)$$

# Unary type theory for a profunctor

$$X \ \mathtt{type}_{\mathcal{S}} \qquad\qquad A \ \mathtt{type}_{\mathcal{E}}$$

$$x : X \vdash_{\mathcal{S}} t : Y \qquad\qquad x : X \vdash_{\mathcal{H}} s : A \qquad\qquad a : A \vdash_{\mathcal{E}} s : B$$

# Unary type theory for a profunctor

$$X\ \mathtt{type}_{\mathcal{S}} \qquad A\ \mathtt{type}_{\mathcal{E}}$$

$$x : X \vdash_{\mathcal{S}} t : Y \qquad x : X \vdash_{\mathcal{H}} s : A \qquad a : A \vdash_{\mathcal{E}} s : B$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{S}} s : Z}{x : X \vdash_{\mathcal{S}} s[t/y] : Z}$$

$$\frac{a : A \vdash_{\mathcal{E}} t : B \qquad b : B \vdash_{\mathcal{E}} s : C}{a : A \vdash_{\mathcal{E}} s[t/b] : C}$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{H}} s : A}{x : X \vdash_{\mathcal{H}} s[t/y] : A}$$

$$\frac{x : X \vdash_{\mathcal{H}} t : A \qquad a : A \vdash_{\mathcal{E}} t : B}{x : X \vdash_{\mathcal{H}} s[t/a] : B}$$

# Unary type theory for a profunctor

$$X \ \mathtt{type}_{\mathcal{S}} \qquad\qquad A \ \mathtt{type}_{\mathcal{E}}$$

$$x : X \vdash_{\mathcal{S}} t : Y \qquad x : X \vdash_{\mathcal{H}} s : A \qquad a : A \vdash_{\mathcal{E}} s : B$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{S}} s : Z}{x : X \vdash_{\mathcal{S}} s[t/y] : Z}$$

$$\frac{a : A \vdash_{\mathcal{E}} t : B \qquad b : B \vdash_{\mathcal{E}} s : C}{a : A \vdash_{\mathcal{E}} s[t/b] : C}$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{H}} s : A}{x : X \vdash_{\mathcal{H}} s[t/y] : A}$$

$$\frac{x : X \vdash_{\mathcal{H}} t : A \qquad a : A \vdash_{\mathcal{E}} t : B}{x : X \vdash_{\mathcal{H}} s[t/a] : B}$$

# Unary type theory for a profunctor

$$X \ \mathtt{type}_{\mathcal{S}} \qquad\qquad A \ \mathtt{type}_{\mathcal{E}}$$

$$x : X \vdash_{\mathcal{S}} t : Y \qquad\qquad x : X \vdash_{\mathcal{H}} s : A \qquad\qquad a : A \vdash_{\mathcal{E}} s : B$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{S}} s : Z}{x : X \vdash_{\mathcal{S}} s[t/y] : Z}$$

$$\frac{a : A \vdash_{\mathcal{E}} t : B \qquad b : B \vdash_{\mathcal{E}} s : C}{a : A \vdash_{\mathcal{E}} s[t/b] : C}$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{H}} s : A}{x : X \vdash_{\mathcal{H}} s[t/y] : A}$$

$$\frac{x : X \vdash_{\mathcal{H}} t : A \qquad a : A \vdash_{\mathcal{E}} t : B}{x : X \vdash_{\mathcal{H}} s[t/a] : B}$$

# Unary type theory for a profunctor

$$X \ \texttt{type}_{\mathcal{S}} \qquad\qquad A \ \texttt{type}_{\mathcal{E}}$$

$$x : X \vdash_{\mathcal{S}} t : Y \qquad x : X \vdash_{\mathcal{H}} s : A \qquad a : A \vdash_{\mathcal{E}} s : B$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{S}} s : Z}{x : X \vdash_{\mathcal{S}} s[t/y] : Z}$$

$$\frac{a : A \vdash_{\mathcal{E}} t : B \qquad b : B \vdash_{\mathcal{E}} s : C}{a : A \vdash_{\mathcal{E}} s[t/b] : C}$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{H}} s : A}{x : X \vdash_{\mathcal{H}} s[t/y] : A}$$

$$\frac{x : X \vdash_{\mathcal{H}} t : A \qquad a : A \vdash_{\mathcal{E}} t : B}{x : X \vdash_{\mathcal{H}} s[t/a] : B}$$

# Unary type theory for a profunctor

$$X \ \mathtt{type}_{\mathcal{S}} \qquad\qquad A \ \mathtt{type}_{\mathcal{E}}$$

$$x : X \vdash_{\mathcal{S}} t : Y \qquad\qquad x : X \vdash_{\mathcal{H}} s : A \qquad\qquad a : A \vdash_{\mathcal{E}} s : B$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{S}} s : Z}{x : X \vdash_{\mathcal{S}} s[t/y] : Z}$$

$$\frac{a : A \vdash_{\mathcal{E}} t : B \qquad b : B \vdash_{\mathcal{E}} s : C}{a : A \vdash_{\mathcal{E}} s[t/b] : C}$$

$$\frac{x : X \vdash_{\mathcal{S}} t : Y \qquad y : Y \vdash_{\mathcal{H}} s : A}{x : X \vdash_{\mathcal{H}} s[t/y] : A}$$

$$\frac{x : X \vdash_{\mathcal{H}} t : A \qquad a : A \vdash_{\mathcal{E}} t : B}{x : X \vdash_{\mathcal{H}} s[t/a] : B}$$

# Unary type theory for an adjunction

$$\frac{A\ \mathtt{type}_{\mathcal{E}}}{p_* A\ \mathtt{type}_{\mathcal{S}}}\ p_*\text{-FORM} \qquad \frac{X\ \mathtt{type}_{\mathcal{S}}}{p^* X\ \mathtt{type}_{\mathcal{E}}}\ p^*\text{-FORM}$$

$$\frac{(x : X) \vdash_{\mathcal{H}} (s : A)}{(x : X) \vdash_{\mathcal{S}} (s^\sharp : p_* A)}\ p_*\text{-INTRO} \qquad \frac{(x : X) \vdash_{\mathcal{S}} (s : p_* A)}{(x : X) \vdash_{\mathcal{H}} (s_\sharp : A)}\ p_*\text{-ELIM}$$

$$\frac{(y : Y) \vdash_{\mathcal{S}} (t : X)}{(y : Y) \vdash_{\mathcal{H}} (t^\flat : p^* X)}\ p^*\text{-INTRO}$$

$$\frac{(b : B) \vdash_{\mathcal{E}} (s : p^* X) \qquad (x : X) \vdash_{\mathcal{H}} (c : C)}{(b : B) \vdash_{\mathcal{E}} ((\text{let } x^\flat := s \text{ in } c) : C)}\ p^*\text{-ELIM}$$

$${s^\sharp}_\sharp = s \qquad {s_\sharp}^\sharp = s \qquad (\text{let } x^\flat := t^\flat \text{ in } c) = c$$

$$(\text{let } x^\flat := t \text{ in } c[x^\flat/y]) = c[t/y]$$

# Unary type theory for an adjunction

$$\frac{A\ \mathtt{type}_{\mathcal{E}}}{p_*A\ \mathtt{type}_{\mathcal{S}}}\ p_*\text{-FORM} \qquad \frac{X\ \mathtt{type}_{\mathcal{S}}}{p^*X\ \mathtt{type}_{\mathcal{E}}}\ p^*\text{-FORM}$$

$$\frac{(x:X)\vdash_{\mathcal{H}}(s:A)}{(x:X)\vdash_{\mathcal{S}}(s^{\sharp}:p_*A)}\ p_*\text{-INTRO} \qquad \frac{(x:X)\vdash_{\mathcal{S}}(s:p_*A)}{(x:X)\vdash_{\mathcal{H}}(s_{\sharp}:A)}\ p_*\text{-ELIM}$$

$$\frac{(y:Y)\vdash_{\mathcal{S}}(t:X)}{(y:Y)\vdash_{\mathcal{H}}(t^{\flat}:p^*X)}\ p^*\text{-INTRO}$$

$$\frac{(b:B)\vdash_{\mathcal{E}}(s:p^*X) \qquad (x:X)\vdash_{\mathcal{H}}(c:C)}{(b:B)\vdash_{\mathcal{E}}((\mathsf{let}\ x^{\flat}:=s\ \mathsf{in}\ c):C)}\ p^*\text{-ELIM}$$

$$s^{\sharp}{}_{\sharp}=s \qquad s_{\sharp}{}^{\sharp}=s \qquad (\mathsf{let}\ x^{\flat}:=t^{\flat}\ \mathsf{in}\ c)=c$$

$$(\mathsf{let}\ x^{\flat}:=t\ \mathsf{in}\ c[x^{\flat}/y])=c[t/y]$$

# Some examples

**Example**

S4 modal logic has a modality $\Box$, with $\Box P$ sometimes interpreted as "$P$ is necessarily true", satisfying laws:

$$\Box(P \wedge Q) = \Box P \wedge \Box Q \qquad \Box\top = \top \qquad \Box P \to P \qquad \Box P \to \Box\Box P$$

In other words, $\Box$ is a product-preserving comonad.
Pfenning–Davies gave a type theory for $\Box$, and Reed decomposed it as $p^* p_*$ for an adjunction $p^* \dashv p_*$, inspiring our framework.

**Example**

In a cohesive topos $p : \mathcal{E} \to \mathcal{S}$:

- the objects of $\mathcal{E}$ are "spaces" or "manifolds"
- $p^* X$ gives $X$ the "discrete topology"
- $p_* X$ is the underlying set of points of $X$

# Unary type theory for diagrams

$\mathcal{M}$ a category (the opposite "shape" of a diagram of toposes).

## Unary modal type theory

- A unary type theory $x : X \vdash_{1_\mathfrak{m}} t : Y$ for each $\mathfrak{m} \in \mathcal{M}$.
- Hetero-judgments $(x : X)_\mathfrak{m} \vdash_\mathfrak{p} (s : A)_\mathfrak{n}$ for each $\mathfrak{p} : \mathfrak{m} \to \mathfrak{n}$.
- Appropriate cut rules, and type operations as desired:

$$\frac{X \ \mathtt{type}_\mathfrak{m}}{\mathfrak{p}^* X \ \mathtt{type}_\mathfrak{n}} \qquad \frac{A \ \mathtt{type}_\mathfrak{n}}{\mathfrak{p}_* X \ \mathtt{type}_\mathfrak{m}}$$

## Semantics

It has semantics in categories $\mathcal{H} \to \mathcal{M}$ over $\mathcal{M}$, where

- If all $\mathfrak{p}^*$ exist, the functor $\mathcal{H} \to \mathcal{M}$ is an opfibration.
- If all $\mathfrak{p}_*$ exist, the functor $\mathcal{H} \to \mathcal{M}$ is a fibration.

If $\mathfrak{p}^*/\mathfrak{p}_*$ all exist, $\mathcal{H} \to \mathcal{M}$ is a bifibration, hence equivalent to a functor $\mathcal{M}^{\mathrm{op}} \to \mathcal{C}at_{\mathrm{radj}}$.

# 1-categories aren't enough

Actually want functors $\mathcal{M}^{\mathrm{op}} \to \mathcal{C}at_{\mathrm{radj}}$ where $\mathcal{M}$ is a 2-category.

**Examples**

- If $\mathcal{M}$ contains an adjunction, get an adjoint triple.
- If $\mathcal{M}$ contains a monad, get an adjoint monad/comonad pair.

These arise naturally on local/cohesive/tangent toposes.

**Definition (Hermida, Buckley)**

A 2-functor $\pi : \mathcal{H} \to \mathcal{M}$ is:

- A local fibration if each functor on hom-categories $\mathcal{H}(X, A) \to \mathcal{M}(\pi X, \pi A)$ is a fibration (+ axioms).
- A 2-fibration if it is a local fibration and has $\mathfrak{p}_*$'s.
- A 2-opfibration if it is a local fibration and has $\mathfrak{p}^*$'s.

**Theorem (Baković, Buckley)**

*(Locally discrete 2-bifib. $\mathcal{H} \to \mathcal{M}$) $\simeq$ (functors $\mathcal{M}^{\mathrm{op}} \to \mathcal{C}at_{\mathrm{radj}}$).*

# Unary type theory for diagrams

$\mathcal{M}$ a 2-category (the opposite "shape" of a diagram of toposes).

## Unary modal type theory (Licata-S. '16)

- A unary type theory $x : X \vdash_{1_\mathfrak{m}} t : Y$ for each $\mathfrak{m} \in \mathcal{M}$.
- Hetero-judgments $(x : X)_\mathfrak{m} \vdash_\mathfrak{p} (s : A)_\mathfrak{n}$ for each $\mathfrak{p} : \mathfrak{m} \to \mathfrak{n}$.
- Appropriate cut rules and type operations $\mathfrak{p}^*, \mathfrak{p}_*$
- Structural rules for 2-cells $\mathfrak{u} : \mathfrak{p} \Rightarrow \mathfrak{q} : \mathfrak{m} \to \mathfrak{n}$

$$\frac{(x : X)_\mathfrak{m} \vdash_\mathfrak{q} (s : A)_\mathfrak{n}}{(x : X)_\mathfrak{m} \vdash_\mathfrak{p} (\mathfrak{u}^* s : A)_\mathfrak{n}}$$

## Semantics

Locally discrete 2-bifibrations $\mathcal{H} \to \mathcal{M}$, hence functors $\mathcal{M}^{\mathrm{op}} \to \mathcal{C}at_{\mathrm{radj}}$.

The objects of $\mathcal{M}$ are sometimes called modes (cf. modal logic).

# Outline

# Simple type theory for an adjunction

In unary type theory, we can think of $(x : X) \vdash_{\mathcal{H}} (s : A)$ as representing a morphism $p^* X \to A$.

**Idea for simple type theory**

Allow a term in an $\mathcal{E}$-type to depend on multiple variables, some in $\mathcal{S}$-types and others in $\mathcal{E}$-types.

- $(x : X)_{\mathcal{S}}, (y : Y)_{\mathcal{S}}, (a : A)_{\mathcal{E}}, (b : B)_{\mathcal{E}} \vdash (t : C)_{\mathcal{E}}$ represents a morphism $p^* X \times p^* Y \times A \times B \to C$.
- This turns out to require/imply that $p^*$ preserves products.
- $(x : X)_{\mathcal{S}} \vdash (t : C)_{\mathcal{E}}$ is the old $(x : X) \vdash_{\mathcal{H}} (t : C)$.
- $(a : A)_{\mathcal{E}} \vdash (t : C)_{\mathcal{E}}$ is the old $(a : A) \vdash_{\mathcal{E}} (t : C)$.
- Still have $(x : X)_{\mathcal{S}} \vdash (s : Y)_{\mathcal{S}}$, the old $(x : X) \vdash_{\mathcal{S}} (s : Y)$. Terms in $\mathcal{S}$-types are not allowed to depend on variables in $\mathcal{E}$-types. "Only left adjoints can appear in contexts."

$$\frac{\Gamma_{\mathcal{S}} \vdash (s : A)_{\mathcal{E}}}{\Gamma_{\mathcal{S}} \vdash (s^{\sharp} : p_* A)_{\mathcal{S}}}\ p_*\text{-INTRO} \qquad \frac{\Gamma_{\mathcal{S}} \vdash (t : p_* A)_{\mathcal{S}}}{\Gamma_{\mathcal{S}}, \Delta_{\mathcal{E}} \vdash (t_{\sharp} : A)_{\mathcal{E}}}\ p_*\text{-ELIM}$$

$$\frac{\Gamma_{\mathcal{S}} \vdash (t : X)_{\mathcal{S}}}{\Gamma_{\mathcal{S}}, \Delta_{\mathcal{E}} \vdash (t^{\flat} : p^* X)_{\mathcal{E}}}\ p^*\text{-INTRO}$$

$$\frac{\Gamma_{\mathcal{S}}, \Delta_{\mathcal{E}} \vdash (t : p^* X)_{\mathcal{E}} \qquad \Gamma_{\mathcal{S}}, \Delta_{\mathcal{E}}, (x : X)_{\mathcal{S}} \vdash (c : C)_{\mathcal{E}}}{\Gamma_{\mathcal{S}}, \Delta_{\mathcal{E}} \vdash ((\mathsf{let}\ x^{\flat} := t\ \mathsf{in}\ c) : C)_{\mathcal{E}}}\ p^*\text{-ELIM}$$

# Towards profunctors for simple type theory

On the categorical side, we should replace:

- categories $\rightsquigarrow$ cartesian monoidal categories
- 2-categories $\rightsquigarrow$ cartesian monoidal 2-categories
- objects in a 2-category $\rightsquigarrow$ cartesian monoidal objects

**Definition**

A cartesian monoidal object $\mathfrak{m} \in \mathcal{M}$ is one with right adjoints to $\Delta : \mathfrak{m} \to \mathfrak{m} \times \mathfrak{m}$ and $! : \mathfrak{m} \to 1$.

# Cartesian monoidal profunctors

Let $\mathcal{M}$ be the cartesian monoidal 2-category freely generated by two cartesian monoidal objects $\mathfrak{e}, \mathfrak{s}$ and a cartesian morphism $\mathfrak{p} : \mathfrak{s} \to \mathfrak{e}$.

- Objects like $\mathfrak{s} \times \mathfrak{s} \times \mathfrak{e} \times \mathfrak{e} \times \mathfrak{e}$
- Morphisms like $\mathfrak{s} \times \mathfrak{s} \to \mathfrak{s}$ and $\mathfrak{e} \times \mathfrak{e} \times \mathfrak{e} \to \mathfrak{e}$ and $\mathfrak{s} \times \mathfrak{s} \times \mathfrak{e} \times \mathfrak{e} \times \mathfrak{e} \to \mathfrak{e} \times \mathfrak{e} \times \mathfrak{e} \times \mathfrak{e} \times \mathfrak{e} \to \mathfrak{e}$

**Definition**

A cartesian monoidal profunctor $\mathcal{E} \nrightarrow \mathcal{S}$ is a cartesian monoidal local fibration $\mathcal{H} \to \mathcal{M}$ with fibers $\mathcal{H}_\mathfrak{e} = \mathcal{E}$ and $\mathcal{H}_\mathfrak{s} = \mathcal{S}$.

- Think of the fiber $\mathcal{H}_{\mathfrak{s}\times\mathfrak{s}\times\mathfrak{e}\times\mathfrak{e}\times\mathfrak{e}}$ as $\mathcal{S} \times \mathcal{S} \times \mathcal{E} \times \mathcal{E} \times \mathcal{E}$ ("Contexts of a specified length and shape")[4]
- Heteromorphisms like $(X_\mathfrak{s}, Y_\mathfrak{s}, A_\mathfrak{e}, B_\mathfrak{e}, C_\mathfrak{e}) \to D_\mathfrak{e}$.
- Local fibration condition gives $[(A, A) \to B] \rightsquigarrow [A \to B]$

[4] This isn't quite true, but the problem goes away if we use cartesian 2-multicategories.

# Simple modal type theory

Let $\mathcal{M}$ be a cartesian monoidal 2-category.

## Simple modal type theory (Licata-S.-Riley '17)

- A class of types for each $\mathfrak{m} \in \mathcal{M}$ (the modes).
- Terms like $(x : X)_{\mathfrak{m}_1}, (y : Y)_{\mathfrak{m}_2} \vdash_{\mathfrak{p}} (s : A)_{\mathfrak{n}}$ for each $\mathfrak{p} : \mathfrak{m}_1 \times \mathfrak{m}_2 \to \mathfrak{n}$.
- Appropriate cut rules and type operations $\mathfrak{p}^*, \mathfrak{p}_*$
- Structural rules for 2-cells $\mathfrak{u} : \mathfrak{p} \Rightarrow \mathfrak{q} : (\mathfrak{m}_1, \ldots, \mathfrak{m}_n) \to \mathfrak{n}$

$$\frac{\Gamma \vdash_{\mathfrak{q}} a : A}{\Gamma \vdash_{\mathfrak{p}} \mathfrak{u}^* s : A}$$

## Semantics

Locally discrete 2-bifibrations $\mathcal{H} \to \mathcal{M}$.

# An unexpected bonus

In a cartesian monoidal 2-category, we can also talk about:

- Objects with non-cartesian monoidal structure $\otimes : \mathfrak{m} \times \mathfrak{m} \to \mathfrak{m}$
- Objects with multiple monoidal structures (e.g. $\otimes, \times$)
- Adjunctions between cartesian and non-cartesian objects
- etc.

We therefore immediately get as special cases of our type theory:

- Intuitionistic linear logic
- Bunched implication
- A decomposition like $\Box = p^* p_*$ for the linear-logic modality !
- etc.

# An unexpected bonus

In a cartesian monoidal 2-category, we can also talk about:

- Objects with non-cartesian monoidal structure $\otimes : \mathfrak{m} \times \mathfrak{m} \to \mathfrak{m}$
- Objects with multiple monoidal structures (e.g. $\otimes, \times$)
- Adjunctions between cartesian and non-cartesian objects
- etc.

We therefore immediately get as special cases of our type theory:

- Intuitionistic linear logic
- Bunched implication
- A decomposition like $\square = p^* p_*$ for the linear-logic modality !
- etc.

Furthermore:

- Product types and function types

$$A \times B \qquad A \to B \qquad A \otimes B \qquad A \multimap B$$

are unified with $p^*, p_*$ as (op)fibrational actions in $\mathcal{H} \to \mathcal{M}$.

# One last enhancement

The cartesian monoidal 2-category $\mathcal{M}$ can also be presented by a type-theoretic syntax!

**Example**

$$
\begin{aligned}
x : \mathfrak{e}, y : \mathfrak{e} &\vdash x \times y : \mathfrak{e} \\
x : \mathfrak{s}, y : \mathfrak{s} &\vdash x \times y : \mathfrak{s} \\
x : \mathfrak{s} &\vdash \mathfrak{p}(x) : \mathfrak{e} \\
x : \mathfrak{s}, y : \mathfrak{s} &\vdash \mathfrak{p}(x \times y) = \mathfrak{p}(x) \times \mathfrak{p}(y) \\
x : \mathfrak{s} &\vdash x \Rightarrow x \times x \\
&\vdots
\end{aligned}
$$

This is the type 2-theory of a cartesian adjunction, written in simple type 3-theory. What do I mean by that?

# Functorial semantics

**Type theory**

A theory is a collection of generating types, terms, and axioms

A model of a theory sends its types/terms to objects/morphisms

**Category theory**

A theory is the structured category $L_T$ freely generated by a model

A model of a theory $T$ in $\mathbf{C}$ is a morphism $L_T \to \mathbf{C}$

# Functorial semantics

| Type theory | Category theory |
| --- | --- |
| A doctrine specifies a "kind of type theory": the type forming operations and their rules | A doctrine is a 2-category of structured categories, such as "cartesian monoidal categories" |
| A theory in a doctrine is a collection of generating types, terms, and axioms | A theory in a doctrine $\mathcal{K}$ is the $L_T \in \mathcal{K}$ freely generated by a model |
| A model of a theory sends its types/terms to objects/morphisms | A model of a theory $T$ in $\mathbf{C}$ is a morphism $L_T \to \mathbf{C}$ |

# What is "a type theory"?

**Remark**

Unfortunately, the phrase "type theory" gets applied to both theories and doctrines.

- When we state the ILH as "the category of dependent type theories is equivalent to the category of lcccs", each such "dependent type theory" is a theory.
- But "Martin-Löf dependent type theory" is a doctrine (namely, the doctrine in which the above theories are written).

This is a source of some confusion.

# Reifying the doctrines

### Standard approach to type theory

1. Given a categorical structure, find a syntactic doctrine.
2. OR: given a syntactic doctrine, find a categorical structure.
3. Prove metatheorems like initiality, canonicity, . . .

### Problems with this

- Without a general framework, can be hard to "find" correspondents.
- Have to prove all the metatheorems over and over again for each doctrine.

### Our approach

Treat doctrines as "categorified theories" in a "categorified doctrine", about which we can prove the theorems once and for all.

# Functorial semantics, bis

| Type theory | Category theory |
| --- | --- |
| A doctrine specifies a "kind of type theory": the type forming operations and their rules | A doctrine is a 2-category of structured categories, such as "cartesian monoidal categories" |
| A theory in a doctrine is a collection of generating types, terms, and axioms | A theory in a doctrine $\mathcal{K}$ is the $L_T \in \mathcal{K}$ freely generated by a model |
| A model of a theory sends its types/terms to objects/morphisms | A model of a theory $T$ in $\mathbf{C}$ is a morphism $L_T \to \mathbf{C}$ |

# Functorial semantics, bis

| Type theory | Category theory |
|---|---|
| A 2-theory specifies a "kind of type theory": the type forming operations and their rules | A 2-theory is a structured 2-category freely generated by something |
| A 1-theory in a 2-theory is a collection of generating types, terms, and axioms | A 1-theory in a 2-theory is a morphism $L_K \to \mathbf{Cat}$ |
| A 0-theory in a 1-theory sends its types/terms to objects/morphisms | A 0-theory in a 1-theory $T$ in $\mathbf{C}$ is a morphism $L_T \to \mathbf{C}$ |

# Functorial semantics, bis

| Type theory | Category theory |
| --- | --- |
| A 3-theory is like "unary type theory", "simple type theory", or "dependent type theory" | A 3-theory is a 3-category like "2-categories", "cartesian monoidal 2-categories", or . . . |
| A 2-theory specifies a "kind of type theory": the type forming operations and their rules | A 2-theory in a 3-theory is an object of it freely generated by something |
| ⋮ | ⋮ |

# Functorial semantics, bis

| Type theory | Category theory |
| --- | --- |
| A 3-theory is like "unary type theory", "simple type theory", or "dependent type theory" | A 3-theory is a 3-category like "2-categories", "cartesian monoidal 2-categories", or ... |
| A 2-theory specifies a "kind of type theory": the type forming operations and their rules | A 2-theory in a 3-theory is an object of it freely generated by something |

We can generate a semantic 2-theory using a syntactic type 2-theory (or mode theory) expressed in a particular 3-theory.

# The hierarchy of 3-theories

Now we can describe the process of "building up to the full complexity of dependent type theory" as a progression through richer 3-theories:

1. LS'16: unary type theory, semantics in 2-categories.

$$x : A \vdash s : B$$

2. LSR'17: simple type theory, semantics in cartesian 2-categories.

$$x : A, y : B, z : C \vdash s : D$$

3. Now: dependent type theory, semantics in comprehension 2-categories.

$$x : A, y : B(x), z : C(x, y) \vdash s : D(x, y, z)$$

# Outline

# What about dependent type theory?

Dependent type theory is a lot more complicated because. . .

1. In $\Gamma \vdash_{\mathfrak{p}} t : Z$, the type $Z$ must also depend on $\Gamma$ in some way that is recorded, $\Gamma \vdash_{\mathfrak{q}} Z \;\mathtt{type}_{\mathfrak{e}}$.
2. Each type *in* $\Gamma$ also depends on the previous ones in some way that must be also be recorded.
3. These dependencies have to be related in some coherent way.

**Example**

If $(x : A)_{\mathfrak{s}} \vdash_{\mathfrak{p}} B \;\mathtt{type}_{\mathfrak{e}}$, then $(x : A)_{\mathfrak{s}}, (y : B)_{\mathfrak{e}} \vdash C \;\mathtt{type}_{\mathfrak{t}}$ must depend on $x$ through some $\mathfrak{r} : \mathfrak{s} \to \mathfrak{t}$ and on $y$ through some $\mathfrak{q} : \mathfrak{e} \to \mathfrak{t}$. Do we need $\mathfrak{q}\mathfrak{p} = \mathfrak{r}$? (In fact, $\mathfrak{r} \Rightarrow \mathfrak{q}\mathfrak{p}$ is enough.)

# More about dependent type theory

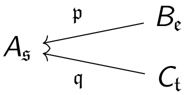
Dependent type theory is a lot more complicated because. . .

4. We need "dependency graphs" that are more complicated than linear. In ordinary DTT, if $B$ and $C$ both depend on $A$ we can write $x : A, y : B(x), z : C(x)$ in order, where the dependence of $C$ on $y$ is trivial. But in modal type theory such a "trivial dependency" may not even be syntactically well-formed.

## Example

If $(x : A)_{\mathfrak{s}} \vdash_{\mathfrak{p}} B\ \mathtt{type}_{\mathfrak{e}}$ for $\mathfrak{p} : \mathfrak{s} \to \mathfrak{e}$, and $(x : A)_{\mathfrak{s}} \vdash_{\mathfrak{q}} C\ \mathtt{type}_{\mathfrak{t}}$ for $\mathfrak{q} : \mathfrak{s} \to \mathfrak{t}$, we want to allow a context like $(x : A)_{\mathfrak{s}}, (y : B)_{\mathfrak{e}}, (z : C)_{\mathfrak{t}}$. But there may be no morphism $\mathfrak{e} \to \mathfrak{t}$ at all, hence no way for $C$ to depend on $y$ even "trivially".

The context has to be structured like a directed acyclic graph or inverse category:



# A disclaimer

This is a snapshot of work in progress.
Tomorrow it might look very different.
(And then the day after that different yet again.)

# A semantic approach

One of the usual semantic correspondents of ordinary DTT is:

**Definition**

A comprehension category is a commuting triangle of functors

$$\begin{array}{ccc} T & \xrightarrow{\chi} & C^{\to} \\ & {\scriptstyle \pi} \searrow \quad \swarrow {\scriptstyle \mathrm{cod}} & \\ & C & \end{array}$$

where. . .

1. $C$ has a terminal object.
2. $C^{\to}$ is the arrow category, with $\mathrm{cod}$ the codomain projection.
3. $\pi : T \to C$ is a fibration.
4. $\chi$ preserves cartesian arrows.

Objects of $C$ are "contexts", objects of $T$ are "types in context".

By analogy with our use of 2-categories in the unary case, and cartesian monoidal 2-categories in the simple case, we define:

## Definition

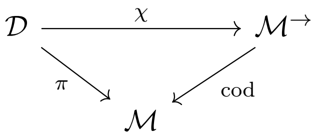
A comprehension 2-category is a commuting triangle of 2-functors

$$\begin{array}{ccc} \mathcal{D} & \xrightarrow{\chi} & \mathcal{M}^{\to} \\ & {}_{\pi}\searrow \quad \swarrow_{\mathrm{cod}} & \\ & \mathcal{M} & \end{array}$$

where. . .

1. $\mathcal{M}$ has a terminal object.
2. $\mathcal{M}^{\to}$ is the arrow 2-category, $\mathrm{cod}$ the codomain projection.
3. $\pi : \mathcal{D} \to \mathcal{M}$ is a 2-fibration.
4. $\chi$ preserves cartesian 1-cells and 2-cells.

# The pieces of a comprehension 2-category




| | |
|---|---|
| objects of $\mathcal{M}$ (mode contexts) | possible "shapes" of contexts (record modes and dependency structure) |
| morphisms of $\mathcal{M}$ (mode substitutions) | shapes of context morphisms |
| objects of $\mathcal{D}$ (mode types) | modes/shapes of dependent types (record mode and dependency structure) |
| sections of $\chi(\mathfrak{m})$ (mode terms) | shapes of terms |

# The basic comprehension 2-category

- If a 1-category $C$ has pullbacks, then $\mathrm{cod} : C^{\to} \to C$ is a fibration, corresponding to the pseudofunctor $c \mapsto C/c$.
- Even if a 2-category $\mathcal{M}$ has pullbacks, $\mathrm{cod} : \mathcal{M}^{\to} \to \mathcal{M}$ is not a 2-fibration! ($\mathfrak{m} \mapsto \mathcal{M}/\mathfrak{m}$ is not functorial on 2-cells.)

- If a 1-category $C$ has pullbacks, then $\mathrm{cod} : C^{\to} \to C$ is a fibration, corresponding to the pseudofunctor $c \mapsto C/c$.
- Even if a 2-category $\mathcal{M}$ has pullbacks, $\mathrm{cod} : \mathcal{M}^{\to} \to \mathcal{M}$ is not a 2-fibration! ($\mathfrak{m} \mapsto \mathcal{M}/\mathfrak{m}$ is not functorial on 2-cells.)
- What is functorial is $\mathfrak{m} \mapsto \mathcal{F}ib(\mathcal{M})/\mathfrak{m}$, the internal fibrations over $\mathfrak{m}$.

**Example (The basic (semantic) comprehension 2-category)**

$$\begin{array}{ccc} \mathcal{F}ib(\mathcal{M}) & \hookrightarrow & \mathcal{M}^{\to} \\ & {\scriptstyle \pi} \searrow \quad \swarrow {\scriptstyle \mathrm{cod}} & \\ & \mathcal{M} & \end{array}$$

More generally, in any comprehension 2-category, $\chi : \mathcal{D} \to \mathcal{M}^{\to}$ lands inside $\mathcal{F}ib(\mathcal{M})$.

# Internal comprehension categories

- For simple type theory, $\mathcal{M}$ is generated by (e.g.) a cartesian monoidal object.
- For dependent type theory, we should instead use "objects with finite limits".
- But the type-theoretic way to talk about categories with finite limits is using *dependent type theory* with $\Sigma, \mathtt{Id}$, which semantically means a comprehension category.

**Definition**

A comprehension object in a comprehension 2-category $\mathcal{D} \to \mathcal{M}$ is:

- An object $\mathfrak{C} \in \mathcal{M}$ with an internal terminal object $\diamond$.
- An object $\mathfrak{T}$ in the fiber $\mathcal{D}_{\mathfrak{c}}$ (a "formal fibration" over $\mathfrak{C}$).
- A morphism $\mathfrak{C}.\mathfrak{T} \to \mathfrak{C}^{\to}$ of internal fibrations over $\mathfrak{C}$ in $\mathcal{M}$ (here $\mathfrak{C}^{\to}$ is the copower by the arrow category in $\mathcal{M}$).

# Building context shapes

- From a comprehension category we can define categories of Reedy fibrant diagrams on inverse categories.
- We can internalize this for a comprehension object in a comprehension 2-category.
- Thus: the "context shapes" are inverse categories.

# Building context shapes

- From a comprehension category we can define categories of Reedy fibrant diagrams on inverse categories.
- We can internalize this for a comprehension object in a comprehension 2-category.
- Thus: the "context shapes" are inverse categories.

One further improvement:

- Reedy fibrant diagrams are tedious to describe in comprehension-category language.
- But we already have a better notation for comprehension categories: dependent type theory itself!
- Use a mini-DTT to describe the modes and mode contexts.[5]

[5]cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*

# Dependent type 2-theory

If we have one comprehension object $\mathfrak{T} \to \mathfrak{C}^{\to}$, a generic mode context (object of $\mathcal{M}$) looks something like this:

$$(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$$

This is a bit hard to parse, so here's some help:

- $()$, $(x : X)$, and $(x : X, y : Y(x))$ are elements of $\mathfrak{C}$, represented by "mini-contexts".
    - $() : \mathfrak{C}$ is in the empty mode context.
    - $(x : X) : \mathfrak{C}$ is in the mode context $(X : \mathfrak{T}())$.
    - $(x : X, y : Y(x))$ in mode context $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$.
- $\mathfrak{T}()$, $\mathfrak{T}(x : X)$, and $\mathfrak{T}(x : X, y : Y(x))$ are mode types (objects of $\mathcal{D}$) obtained as pullbacks of $\mathfrak{T}$ to these elements.
- The whole thing is obtained by repeatedly extending by a variable $(X, Y, Z)$ belonging to a mode type that's well-defined in the context of the previous variables.

# Dependent type 2-theory

If we have one comprehension object $\mathfrak{T} \to \mathfrak{C}^{\to}$, a generic mode context (object of $\mathcal{M}$) looks something like this:

$$(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$$

This is a bit hard to parse, so here's some help:

- $()$, $(x : X)$, and $(x : X, y : Y(x))$ are elements of $\mathfrak{C}$, represented by "mini-contexts".
  - $() : \mathfrak{C}$ is in the empty mode context.
  - $(x : X) : \mathfrak{C}$ is in the mode context $(X : \mathfrak{T}())$.
  - $(x : X, y : Y(x))$ in mode context $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$.
- $\mathfrak{T}()$, $\mathfrak{T}(x : X)$, and $\mathfrak{T}(x : X, y : Y(x))$ are mode types (objects of $\mathcal{D}$) obtained as pullbacks of $\mathfrak{T}$ to these elements.
- The whole thing is obtained by repeatedly extending by a variable $(X, Y, Z)$ belonging to a mode type that's well-defined in the context of the previous variables.

# Dependent type 2-theory

If we have one comprehension object $\mathfrak{T} \to \mathfrak{C}^{\to}$, a generic mode context (object of $\mathcal{M}$) looks something like this:

$$(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$$

This is a bit hard to parse, so here's some help:

- $()$, $(x : X)$, and $(x : X, y : Y(x))$ are elements of $\mathfrak{C}$, represented by "mini-contexts".
  - $() : \mathfrak{C}$ is in the empty mode context.
  - $(x : X) : \mathfrak{C}$ is in the mode context $(X : \mathfrak{T}())$.
  - $(x : X, y : Y(x))$ in mode context $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$.
- $\mathfrak{T}()$, $\mathfrak{T}(x : X)$, and $\mathfrak{T}(x : X, y : Y(x))$ are mode types (objects of $\mathcal{D}$) obtained as pullbacks of $\mathfrak{T}$ to these elements.
- The whole thing is obtained by repeatedly extending by a variable $(X, Y, Z)$ belonging to a mode type that's well-defined in the context of the previous variables.

# Dependent type 2-theory

If we have one comprehension object $\mathfrak{T} \to \mathfrak{C}^{\to}$, a generic mode context (object of $\mathcal{M}$) looks something like this:

$$\color{red}{(X : \mathfrak{T}())}, (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$$

This is a bit hard to parse, so here's some help:

- $()$, $(x : X)$, and $(x : X, y : Y(x))$ are elements of $\mathfrak{C}$, represented by "mini-contexts".
    - $() : \mathfrak{C}$ is in the empty mode context.
    - $(x : X) : \mathfrak{C}$ is in the mode context $(X : \mathfrak{T}())$.
    - $(x : X, y : Y(x))$ in mode context $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$.
- $\mathfrak{T}()$, $\mathfrak{T}(x : X)$, and $\mathfrak{T}(x : X, y : Y(x))$ are mode types (objects of $\mathcal{D}$) obtained as pullbacks of $\mathfrak{T}$ to these elements.
- The whole thing is obtained by repeatedly extending by a variable $(X, Y, Z)$ belonging to a mode type that's well-defined in the context of the previous variables.

# Dependent type 2-theory

If we have one comprehension object $\mathfrak{T} \to \mathfrak{C}^{\to}$, a generic mode context (object of $\mathcal{M}$) looks something like this:

$$(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$$

This is a bit hard to parse, so here's some help:

- $()$, $(x : X)$, and $(x : X, y : Y(x))$ are elements of $\mathfrak{C}$, represented by "mini-contexts".
    - $() : \mathfrak{C}$ is in the empty mode context.
    - $(x : X) : \mathfrak{C}$ is in the mode context $(X : \mathfrak{T}())$.
    - $(x : X, y : Y(x))$ in mode context $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$.
- $\mathfrak{T}()$, $\mathfrak{T}(x : X)$, and $\mathfrak{T}(x : X, y : Y(x))$ are mode types (objects of $\mathcal{D}$) obtained as pullbacks of $\mathfrak{T}$ to these elements.
- The whole thing is obtained by repeatedly extending by a variable $(X, Y, Z)$ belonging to a mode type that's well-defined in the context of the previous variables.

# Dependent type 2-theory

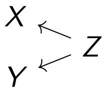
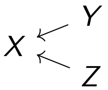
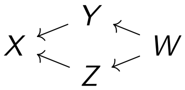
If we have one comprehension object $\mathfrak{T} \to \mathfrak{C}^{\to}$, a generic mode context (object of $\mathcal{M}$) looks something like this:

$$(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$$

This is a bit hard to parse, so here's some help:

- $()$, $(x : X)$, and $(x : X, y : Y(x))$ are elements of $\mathfrak{C}$, represented by "mini-contexts".
  - $() : \mathfrak{C}$ is in the empty mode context.
  - $(x : X) : \mathfrak{C}$ is in the mode context $(X : \mathfrak{T}())$.
  - $(x : X, y : Y(x))$ in mode context $\big(X : \mathfrak{T}()\big), \big(Y : \mathfrak{T}(x : X)\big)$.
- $\mathfrak{T}()$, $\mathfrak{T}(x : X)$, and $\mathfrak{T}(x : X, y : Y(x))$ are mode types (objects of $\mathcal{D}$) obtained as pullbacks of $\mathfrak{T}$ to these elements.
- The whole thing is obtained by repeatedly extending by a variable $(X, Y, Z)$ belonging to a mode type that's well-defined in the context of the previous variables.

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

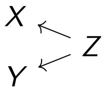
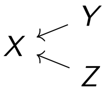
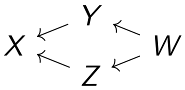
| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ |  |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ |  |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$ $(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ |  |

# Inverse categories via mini-contexts

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ | $X \leftarrow Z \rightarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ | $Y \rightarrow X \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$ $(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ | $X \leftarrow Y \leftarrow W$, $X \leftarrow Z \leftarrow W$ |

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ |  |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ |  |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$ $(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ |  |

# Inverse categories via mini-contexts

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ | $X \leftarrow Z \rightarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ | $Y \rightarrow X \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$ $(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ | $X \leftarrow Y \leftarrow W$, $X \leftarrow Z \leftarrow W$ |

# Inverse categories via mini-contexts

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ | $X \leftarrow Z$, $Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ | $X \leftarrow Y$, $X \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$ $(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ | $X \leftarrow Y \leftarrow W$, $X \leftarrow Z \leftarrow W$ |

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ | $\begin{array}{ccc} X & & \\ & \nwarrow & \\ & & Z \\ & \swarrow & \\ Y & & \end{array}$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ | $\begin{array}{ccc} & & Y \\ & \swarrow & \\ X & & \\ & \nwarrow & \\ & & Z \end{array}$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$ $(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ | $\begin{array}{ccccc} & & Y & & \\ & \swarrow & & \nwarrow & \\ X & & & & W \\ & \nwarrow & & \swarrow & \\ & & Z & & \end{array}$ |

# Inverse categories via mini-contexts

[cf. Tsementzis–Weaver, *Finite Inverse Categories as Signatures*]

| Mode context | Inverse category |
|---|---|
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X))$ | $X \leftarrow Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X, y : Y(x)))$ | $X \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X, x' : X))$ | $X \leftleftarrows Y$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}()), (Z : \mathfrak{T}(x : X, y : Y))$ | $\begin{array}{ccc} X & & \\ & \nwarrow & \\ & & Z \\ & \swarrow & \\ Y & & \end{array}$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X))$ | $\begin{array}{ccc} & & Y \\ & \swarrow & \\ X & & \\ & \nwarrow & \\ & & Z \end{array}$ |
| $(X : \mathfrak{T}()), (Y : \mathfrak{T}(x : X)), (Z : \mathfrak{T}(x : X)),$<br>$(W : \mathfrak{T}(x : X, y : Y(x), z : Z(x)))$ | $\begin{array}{ccccc} & & Y & & \\ & \swarrow & & \nwarrow & \\ X & & & & W \\ & \nwarrow & & \swarrow & \\ & & Z & & \end{array}$ |

# Dependent type theory over dependent type 2-theory

Semantically, we have a fibration of comprehension 2-categories over $\mathcal{D} \to \mathcal{M}^{\to}$. Syntactically, we have judgments-over-judgments:

$$\begin{array}{cccc} (a : A), & (b : B(a)) & \vdash & C(a,b)\ \mathtt{type} \\ (X{:}\mathfrak{T}()), & (Y{:}\mathfrak{T}(x{:}X)) & \vdash & \mathfrak{T}(x{:}X, y{:}Y(x))\ \mathtt{mode} \end{array}$$

---

$$\begin{array}{cccc} (a : A), & (b : B) & \vdash & C(a,b)\ \mathtt{type} \\ (X{:}\mathfrak{T}()), & (Y{:}\mathfrak{T}()) & \vdash & \mathfrak{T}(x{:}X, y{:}Y)\ \mathtt{mode} \end{array}$$

---

$$\begin{array}{cccc} (a : A), & (b : B(a)) & \vdash & C(a) \\ (X{:}\mathfrak{T}()), & (Y{:}\mathfrak{T}(x{:}X)) & \vdash & \mathfrak{T}(x{:}X)\ \mathtt{mode} \end{array}$$

Now suppose we have two comprehension objects $\mathfrak{T}_{\mathfrak{s}} \to \mathfrak{C}_{\mathfrak{s}}^{\to}$ and $\mathfrak{T}_{\mathfrak{e}} \to \mathfrak{C}_{\mathfrak{e}}^{\to}$, with a "comprehension morphism" consisting of terms

$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}) \vdash \mathfrak{p}\Gamma : \mathfrak{C}_{\mathfrak{e}}$$
$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}), (X : \mathfrak{T}_{\mathfrak{s}}(\Gamma)) \vdash \mathfrak{p}X : \mathfrak{T}_{\mathfrak{e}}(\mathfrak{p}\Gamma)$$

which "commute with comprehension". We have a "purely $\mathfrak{s}$" 2-DTT and a "purely $\mathfrak{e}$" 2-DTT, plus e.g.

| Mode context | "Inverse category" |
|---|---|
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X))$ | $\mathfrak{p}^*(X) \leftarrow Y$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)),$<br>$(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{s}}(x : X)),$<br>$(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : \mathfrak{p}Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y) \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)), (Z : \mathfrak{T}_{\mathfrak{s}}(x : X))$ | $\mathfrak{p}^*(X) \begin{smallmatrix} \leftarrow \mathfrak{p}^*(Y) \\ \leftarrow Z \end{smallmatrix}$ |

# Dependent type 2-theory for an adjunction

Now suppose we have two comprehension objects $\mathfrak{T}_{\mathfrak{s}} \to \mathfrak{C}_{\mathfrak{s}}^{\to}$ and $\mathfrak{T}_{\mathfrak{e}} \to \mathfrak{C}_{\mathfrak{e}}^{\to}$, with a "comprehension morphism" consisting of terms

$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}) \vdash \mathfrak{p}\Gamma : \mathfrak{C}_{\mathfrak{e}}$$
$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}), (X : \mathfrak{T}_{\mathfrak{s}}(\Gamma)) \vdash \mathfrak{p}X : \mathfrak{T}_{\mathfrak{e}}(\mathfrak{p}\Gamma)$$

which "commute with comprehension". We have a "purely $\mathfrak{s}$" 2-DTT and a "purely $\mathfrak{e}$" 2-DTT, plus e.g.

| Mode context | "Inverse category" |
|---|---|
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X))$ | $\mathfrak{p}^*(X) \leftarrow Y$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)),$ $(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{s}}(x : X)),$ $(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : \mathfrak{p}Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y) \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)), (Z : \mathfrak{T}_{\mathfrak{s}}(x : X))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y)$, $\mathfrak{p}^*(X) \leftarrow Z$ |

Now suppose we have two comprehension objects $\mathfrak{T}_{\mathfrak{s}} \to \mathfrak{C}_{\mathfrak{s}}^{\to}$ and $\mathfrak{T}_{\mathfrak{e}} \to \mathfrak{C}_{\mathfrak{e}}^{\to}$, with a "comprehension morphism" consisting of terms

$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}) \vdash \mathfrak{p}\Gamma : \mathfrak{C}_{\mathfrak{e}}$$

$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}), (X : \mathfrak{T}_{\mathfrak{s}}(\Gamma)) \vdash \mathfrak{p}X : \mathfrak{T}_{\mathfrak{e}}(\mathfrak{p}\Gamma)$$

which "commute with comprehension". We have a "purely $\mathfrak{s}$" 2-DTT and a "purely $\mathfrak{e}$" 2-DTT, plus e.g.

| Mode context | "Inverse category" |
|---|---|
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X))$ | $\mathfrak{p}^*(X) \leftarrow Y$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)),$<br>$(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{s}}(x : X)),$<br>$(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : \mathfrak{p}Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y) \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)), (Z : \mathfrak{T}_{\mathfrak{s}}(x : X))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y)$<br>$\mathfrak{p}^*(X) \leftarrow Z$ |

Now suppose we have two comprehension objects $\mathfrak{T}_{\mathfrak{s}} \to \mathfrak{C}_{\mathfrak{s}}^{\to}$ and $\mathfrak{T}_{\mathfrak{e}} \to \mathfrak{C}_{\mathfrak{e}}^{\to}$, with a “comprehension morphism” consisting of terms

$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}) \vdash \mathfrak{p}\Gamma : \mathfrak{C}_{\mathfrak{e}}$$
$$(\Gamma : \mathfrak{C}_{\mathfrak{s}}), (X : \mathfrak{T}_{\mathfrak{s}}(\Gamma)) \vdash \mathfrak{p}X : \mathfrak{T}_{\mathfrak{e}}(\mathfrak{p}\Gamma)$$

which “commute with comprehension”. We have a “purely $\mathfrak{s}$” 2-DTT and a “purely $\mathfrak{e}$” 2-DTT, plus e.g.

| Mode context | “Inverse category” |
|---|---|
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X))$ | $\mathfrak{p}^*(X) \leftarrow Y$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)),$ <br> $(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{s}}(x : X)),$ <br> $(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X, y : \mathfrak{p}Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y) \leftarrow Z$ |
| $(X : \mathfrak{T}_{\mathfrak{s}}()), (Y : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{p}X)), (Z : \mathfrak{T}_{\mathfrak{s}}(x : X))$ | $\mathfrak{p}^*(Y) \rightarrow \mathfrak{p}^*(X) \leftarrow Z$ |

# Dependent type 2-theory for an adjunction

Now suppose we have two comprehension objects $\mathfrak{T}_\mathfrak{s} \to \mathfrak{C}_\mathfrak{s}^\to$ and $\mathfrak{T}_\mathfrak{e} \to \mathfrak{C}_\mathfrak{e}^\to$, with a "comprehension morphism" consisting of terms

$$(\Gamma : \mathfrak{C}_\mathfrak{s}) \vdash \mathfrak{p}\Gamma : \mathfrak{C}_\mathfrak{e}$$
$$(\Gamma : \mathfrak{C}_\mathfrak{s}), (X : \mathfrak{T}_\mathfrak{s}(\Gamma)) \vdash \mathfrak{p}X : \mathfrak{T}_\mathfrak{e}(\mathfrak{p}\Gamma)$$

which "commute with comprehension". We have a "purely $\mathfrak{s}$" 2-DTT and a "purely $\mathfrak{e}$" 2-DTT, plus e.g.

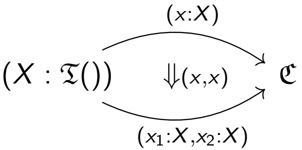
| Mode context | "Inverse category" |
|---|---|
| $(X : \mathfrak{T}_\mathfrak{s}()), (Y : \mathfrak{T}_\mathfrak{e}(x : \mathfrak{p}X))$ | $\mathfrak{p}^*(X) \leftarrow Y$ |
| $(X : \mathfrak{T}_\mathfrak{s}()), (Y : \mathfrak{T}_\mathfrak{e}(x : \mathfrak{p}X)),$ $(Z : \mathfrak{T}_\mathfrak{e}(x : \mathfrak{p}X, y : Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow Y \leftarrow Z$ |
| $(X : \mathfrak{T}_\mathfrak{s}()), (Y : \mathfrak{T}_\mathfrak{s}(x : X)),$ $(Z : \mathfrak{T}_\mathfrak{e}(x : \mathfrak{p}X, y : \mathfrak{p}Y(x)))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y) \leftarrow Z$ |
| $(X : \mathfrak{T}_\mathfrak{s}()), (Y : \mathfrak{T}_\mathfrak{e}(x : \mathfrak{p}X)), (Z : \mathfrak{T}_\mathfrak{s}(x : X))$ | $\mathfrak{p}^*(X) \leftarrow \mathfrak{p}^*(Y)$, $\mathfrak{p}^*(X) \leftarrow Z$ |

# Dependent type theory for an adjunction

$$\frac{\dfrac{\begin{array}{cccc}(x:X)_{\mathfrak{s}}, & (a:A(x))_{\mathfrak{e}} & \vdash & B(x,a)\ \mathtt{type}_{\mathfrak{e}} \\ (X{:}\mathfrak{T}_{\mathfrak{s}}()), & (Y{:}\mathfrak{T}_{\mathfrak{e}}(x{:}\mathfrak{p}(X))) & \vdash & \mathfrak{T}_{\mathfrak{e}}(x{:}\mathfrak{p}(X), y{:}Y(x))\ \mathtt{mode}\end{array}}{\begin{array}{cccc}(x:X)_{\mathfrak{s}}, & (y:Y(x))_{\mathfrak{s}} & \vdash & A(x,y)\ \mathtt{type}_{\mathfrak{e}} \\ (X{:}\mathfrak{T}_{\mathfrak{s}}()), & (Y{:}\mathfrak{T}_{\mathfrak{s}}(x{:}X)) & \vdash & \mathfrak{T}_{\mathfrak{e}}(x{:}\mathfrak{p}(X), y{:}\mathfrak{p}(Y)(x))\ \mathtt{mode}\end{array}}}{\begin{array}{cccc}(x:X)_{\mathfrak{s}}, & (a:A(x))_{\mathfrak{e}} & \vdash & B(x)\ \mathtt{type}_{\mathfrak{s}} \\ (X{:}\mathfrak{T}_{\mathfrak{s}}()), & (Y{:}\mathfrak{T}_{\mathfrak{e}}(x{:}\mathfrak{p}(X))) & \vdash & \mathfrak{T}_{\mathfrak{s}}(x{:}X)\ \mathtt{mode}\end{array}}$$

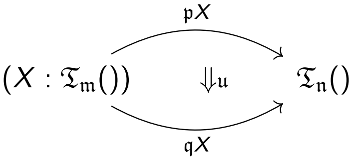
$\mathcal{M}$ and $\mathcal{D}$ are 2-categories, so we have 2-cell judgments. These include variable-for-variable substitutions on mini-contexts:

$$(X : \mathfrak{T}()) \mid (x : X) \vDash (x, x) : (x_1 : X, x_2 : X) : \mathfrak{C}$$




as well as generating 2-cells between generating mode morphisms:

$$(X : \mathfrak{T}_{\mathfrak{m}}()) \mid (x : \mathfrak{p}X) \vDash \mathfrak{u}(x) : \mathfrak{q}X : \mathfrak{T}_{\mathfrak{n}}()$$



# Modal dependency

Suppose comprehension objects labeled $\mathfrak{m}, \mathfrak{n}, \mathfrak{e}$ with morphisms

$$\mathfrak{p} : \mathfrak{m} \to \mathfrak{n} \qquad \mathfrak{q} : \mathfrak{n} \to \mathfrak{e} \qquad \mathfrak{r} : \mathfrak{m} \to \mathfrak{e}$$

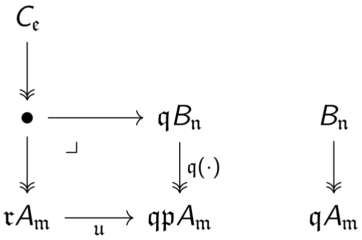
and a 2-cell $\mathfrak{u} : \mathfrak{r} \Rightarrow \mathfrak{q}\mathfrak{p}$. Then we have a mode context

$$\big(X : \mathfrak{T}_{\mathfrak{m}}()\big), \big(Y : \mathfrak{T}_{\mathfrak{n}}(x : \mathfrak{p}X)\big), \big(Z : \mathfrak{T}_{\mathfrak{e}}(x : \mathfrak{r}X, y : \mathfrak{q}Y(\mathfrak{u}(x)))\big)$$

Note how the type of $Z$ typechecks: $x : \mathfrak{r}X$, so $\mathfrak{u}(x) : \mathfrak{q}\mathfrak{p}(X)$ which is what $\mathfrak{q}Y$ depends on.

$$(a : A)_{\mathfrak{m}},\quad (b : B(a))_{\mathfrak{n}} \quad\vdash\quad C(a, b)\ \mathtt{type}_{\mathfrak{e}}$$

$$(X{:}\mathfrak{T}_{\mathfrak{m}}()),\quad (Y{:}\mathfrak{T}_{\mathfrak{n}}(x{:}\mathfrak{p}X)) \quad\vdash\quad (Z{:}\mathfrak{T}_{\mathfrak{e}}(x{:}\mathfrak{r}X, y{:}\mathfrak{q}Y(\mathfrak{u}(x))))\ \mathtt{mode}$$




In general, what we get semantically is the oplax limit of an oplax diagram of comprehension categories.

# Conclusion

1. All kinds of "type doctrines", including geometric morphisms, modalities, non-cartesian monoidal structures, and all kinds of dependency, can be expressed syntactically as "dependent type 2-theories".
2. Each such 2-theory generates a class of 1-theories that specialize to "dependent modal type theories" for describing structures on, and diagrams of, $(\infty, 1)$-toposes.
3. We can hope to prove metatheorems like canonicity and initiality once and for all, and then simply specialize them to every new 2-theory.