Modelling crisp type theory

Florrie Verity
Australian National University

JHU Category theory seminar
17 May 2023

Plan

1. Modalities and modal type theory
   - the view of crisp type theory from logic and from HoTT

2. Modelling dependent type theory
   - natural models approach

3. Modelling crisp type theory
   - the abstract to the (slightly more) concrete

① Modalities and modal type theory

The "traditional" view from logic

- modalities are operations on propositions

modal logic		linear logic	
$\Box A$	$\Diamond A$	$!A$	$?A$
comonad	monad	comonad	monad

e.g. modal logic S4

Axioms

(K) $\Box(A \to B) \to (\Box A \to \Box B)$

(T) $\Box A \to A$

(4) $\Box A \to \Box\Box A$

the data of a comonad

(i) $\Box : \mathcal{C} \to \mathcal{C}$ a functor

(ii) $\varepsilon : \Box \Rightarrow id_{\mathcal{C}}$

(iii) $\delta : \Box \Rightarrow \Box\Box$

$\Big\}$ natural transformations

- modal type theories originate in computer science to model "real" programming languages

↳ we'll see an alternative "logical" account due to Pfenning & Davies, 2007

The view from HoTT

points in a space
"hanging together"

"Axiomatic cohesion"
- Lawvere 2007

"Cohesive homotopy type theory"
- Schreiber and Shulman 2012

$$\begin{tikzcd}
\text{Spaces} \arrow[d, bend right=70, "\Pi_0"'] \arrow[d, "\Gamma"] \\
\text{Sets} \arrow[u, bend left=35, "\Delta"] \arrow[u, bend right=70, "\nabla"']
\end{tikzcd}
\qquad \Pi_0 \dashv \Delta \dashv \Gamma \dashv \nabla$$

generalised to $\rightsquigarrow$

$$\begin{tikzcd}
\text{topological } \infty\text{-groupoids} \arrow[d, bend right=70, "\Pi_0"'] \arrow[d, "\Gamma"] \\
\infty\text{-groupoids} \arrow[u, bend left=35, "\Delta"] \arrow[u, bend right=70, "\nabla"']
\end{tikzcd}
\qquad \Pi_0 \dashv \Delta \dashv \Gamma \dashv \nabla$$

types in HoTT

Modalities are endofunctors
on types/propositions

$$\begin{tikzcd}[row sep=huge, column sep=small]
\text{topological } \infty\text{-groupoids} \arrow[d, bend right=60, "\Pi_0"'] \arrow[d, shift left=2, "\Gamma"] \\
\infty\text{-groupoids} \arrow[u, bend left=40, "\Delta"] \arrow[u, bend right=60, "\nabla"']
\end{tikzcd}
\qquad \Pi_0 \dashv \Delta \dashv \Gamma \dashv \nabla$$

endofunctor
$\rightsquigarrow$
perspective

$$\begin{tikzcd}[row sep=huge, column sep=small]
\text{topological } \infty\text{-groupoids} \arrow[d, "\flat"] \\
\text{topological } \infty\text{-groupoids} \arrow[u, bend left=40, "\int"] \arrow[u, bend right=40, "\sharp"']
\end{tikzcd}
\qquad \int \dashv \flat \dashv \sharp$$

$\int = \Delta \Pi_0$

idempotent $\begin{cases} \flat = \Delta\Gamma & \text{comonad} \\ \sharp = \nabla\Gamma & \text{monad} \end{cases}$

Modalities

1) The "traditional" view from logic
2) The view from HoTT

How are these views connected?

Case study: crisp type theory

Crisp type theory - overview

- Shulman's "Spatial type theory" (2018) incorporates $\flat$, $\sharp$ and $\int$
  ↳ "Crisp type theory" is the $\flat$- fragment
  - dependent version of Pfenning and Davies' 2001 system
- uses a "split context" $\Delta \mid \Gamma$

Applications -

- "Brouwer's fixed-point theorem in real-cohesive HoTT"
  - Shulman 2018
- "Internal universes in models of HoTT"
  - Licata, Orton, Pitts and Spitters, 2018

  ↳ "Kripke-Joyal forcing for type theory and uniform fibrations"
  - Awodey, Gambino and Hazratpour, 2021

"A judgemental reconstruction of modal logic"

– Pfenning and Davies, 2001

Judgements in logic

- A is a proposition

  we know what counts as a verification of A

- A is true

  we know how to verify A

NB: "A is true" presupposes "A is a proposition"

Example Conjunction

Explained by the following "inference rules"

- Formation rule
$$\frac{A \text{ prop} \quad B \text{ prop}}{A \wedge B \text{ prop}}$$

- Introduction rule
$$\frac{A \text{ true} \quad B \text{ true}}{A \wedge B \text{ true}}$$

- Elimination rule
$$\frac{A \wedge B \text{ true}}{A \text{ true}} \qquad \frac{A \wedge B \text{ true}}{B \text{ true}}$$

How do we explain implication, $A \Rightarrow B$?

Hypothetical judgements

$\underbrace{J_1, \ldots, J_n}_{\text{hypotheses}} \vdash J$

J assuming J_1 through J_n

e.g. $A_1 \text{ true}, \ldots, A_n \text{ true} \vdash A \text{ true}$

To explain implication:

$$\frac{A \text{ prop} \qquad B \text{ prop}}{A \Rightarrow B \text{ prop}}$$

$$\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \Rightarrow B \text{ true}}$$

we know how to verify $A \Rightarrow B$ if we know how to verify B under hypothesis "A true"

What about modality?

1) Introduce a third judgement

Definition (Validity)

1) If $\bullet \vdash A \text{ true}$ then $A \text{ valid}$.
2) if $A \text{ valid}$ then $\Gamma \vdash A \text{ true}$.

necessarily true

This may be used in hypothetical judgements

$$B_1 \text{ valid}, \dots, B_m \text{ valid} \mid A_1 \text{ true}, \dots, A_n \text{ true} \vdash A \text{ true},$$

abbreviated

$$\Delta \mid \Gamma \vdash A \text{ true}.$$

2) Internalise this judgement as a proposition

- Formation rule
$$\frac{A \text{ prop}}{\Box A \text{ prop}}$$

- Introduction rule
$$\frac{\Delta \mid \cdot \vdash A \text{ true}}{\Delta \mid \Gamma \vdash \Box A \text{ true}}$$

(follows from the definition of validity, updated with split contexts.

1) If $\Delta \mid \cdot \vdash A$ true then A valid.
2) if A valid then $\Delta \mid \Gamma \vdash A$ true.)

- Elimination rule

$$\frac{\Delta \mid \Gamma \vdash \Box A \text{ true}}{\Delta \mid \cdot \vdash A \text{ true}}$$ ✗ too strong

$$\frac{\Delta \mid \Gamma \vdash \Box A \text{ true}}{\Delta \mid \Gamma \vdash A \text{ true}}$$ ✗ too weak

$$\frac{\Delta \mid \Gamma \vdash \Box A \text{ true} \qquad \Delta, A \text{ valid} \mid \Gamma \vdash C \text{ true}}{\Delta \mid \Gamma \vdash C \text{ true}}$$ ✓

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \text{ true} \qquad \Gamma, B \vdash C \text{ true}}{\Gamma \vdash C \text{ true}} \vee E$$

Our formal system -

Propositions $\quad A ::= P \;\|\; \Box A$

True hypotheses $\quad \Gamma ::= \cdot \;\|\; \Gamma, A\ true$

Valid hypotheses $\quad \Delta ::= \cdot \;\|\; \Delta, A\ valid$

Inference rules

$$\frac{\Delta \mid \cdot \vdash A\ true}{\Delta \mid \Gamma \vdash \Box A\ true}\ \Box I$$

$$\frac{\Delta \mid \Gamma \vdash \Box A\ true \qquad \Delta, A\ valid \mid \Gamma \vdash C\ true}{\Delta \mid \Gamma \vdash C\ true}\ \Box E$$

$$\frac{}{\Delta \mid \Gamma, A\ true, \Gamma' \vdash A\ true}\ hyp_1$$

$$\frac{}{\Delta, B\ valid, \Delta' \mid \Gamma \vdash B\ true}\ hyp_2$$

new

Moving to a type theory

Recall the judgement form

A is true

we know how to verify A

Let's give names to verifications and replace the above judgement with

$M : A$

M is a proof term for A

proof / term

proposition / type

Hypothetical version : $x : A$ $\quad$ $u :: A$

variables

Example Conjunction

- Formation rule

$$\frac{A\ \mathsf{prop} \quad B\ \mathsf{prop}}{A \wedge B\ \mathsf{prop}}$$

- Introduction rule

$$\frac{A\ \mathsf{true} \quad B\ \mathsf{true}}{A \wedge B\ \mathsf{true}} \rightsquigarrow \frac{M : A \quad N : B}{\langle M, N \rangle : A \wedge B}$$

- Elimination rule

$$\frac{A \wedge B\ \mathsf{true}}{A\ \mathsf{true}} \rightsquigarrow \frac{M : A \wedge B}{\mathsf{fst}\ M : A}$$

$$\frac{A \wedge B\ \mathsf{true}}{B\ \mathsf{true}} \rightsquigarrow \frac{M : A \wedge B}{\mathsf{snd}\ M : B}$$

- Computation rules

$$\mathsf{fst}\langle M, N \rangle \Rightarrow_R M$$

$$\mathsf{snd}\langle M, N \rangle \Rightarrow_R N$$

$$M : A \wedge B \Rightarrow_E \langle \mathsf{fst}\ M, \mathsf{snd}\ M \rangle$$

relate intro and elim rules

Our typed formal system -

Types	$A ::= P \parallel \Box A$
New ⇝ Terms	$M ::= x \parallel u \parallel \text{box } M \parallel \text{let box } u = M_1 \text{ in } M_2$
True contexts	$\Gamma ::= \cdot \parallel \Gamma, x:A$
Valid contexts	$\Delta ::= \cdot \parallel \Delta, u::A$

Inference rules -

$$\frac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash \text{box } M : \Box A}\ \Box I$$

$$\frac{\Delta \mid \Gamma \vdash M : \Box A \qquad \Delta, u::A \mid \Gamma \vdash N : C}{\Delta \mid \Gamma \vdash \text{let box } u = M \text{ in } N : C}\ \Box E$$

$$\frac{}{\Delta \mid \Gamma, x:A, \Gamma' \vdash x : A}\ hyp_1 \qquad \frac{}{\Delta, u::A, \Delta' \mid \Gamma \vdash u : A}\ hyp_2$$

Computation rules -

$$\text{let box } u = \text{box } M \text{ in } N \Rightarrow_R N[\![M/u]\!]$$

replace all instances of u in N with M

$$M : \Box A \Rightarrow_E \text{let box } u = M \text{ in } (\text{box } u)$$

Moving to Crisp type theory

$x : A \vdash B(x)$ type

- Crisp type theory is dependently-typed

  i.e. $x_1 : A_1, \ldots, x_n : A_n \vdash$ really means

  $x_1 : A_1, x_2 : A_2(x_1), x_3 : A_3(x_1, x_2), \ldots, x_n : A_n(x_1, \ldots, x_{n-1}) \vdash$

  – Substitution is a meta-operation on expressions (types & terms)

  $\phi[N/x]$ replace all instances of x in ϕ with N

  N.B. substitution is strictly functorial

- Terminology changes

  box modality $\Box A$ $\rightsquigarrow$ flat modality $\flat A$

  validity hypotheses $u :: A$ $\rightsquigarrow$ "crisp" hypotheses

  "crisp context / context of crisp variables" $\rightsquigarrow$ $\Delta \mid \Gamma$ $\leftsquigarrow$ "non-crisp context, context of non-crisp variables"

Context rules

Crisp type theory	Pfenning and Davies' Modal type theory
$$\frac{}{\cdot \mid \cdot \vdash}\ \text{Emp}$$	$\Gamma ::= \cdot \parallel \Gamma, x : A$ $\Delta ::= \cdot \parallel \Delta, u :: A$
$$\frac{\Delta \mid \cdot \vdash A\ \text{type}}{\Delta, u :: A \mid \cdot \vdash}\ \flat\text{-ext}$$	—
$$\frac{\Delta \mid \Gamma \vdash A\ \text{type}}{\Delta \mid \Gamma, x : A \vdash}\ \text{ext}$$	—
$$\frac{\Delta, u :: A, \Delta' \mid \Gamma \vdash}{\Delta, u :: A, \Delta' \mid \Gamma \vdash u : A}\ \flat\text{-var}$$	$$\frac{}{\Delta, u :: A, \Delta' \mid \Gamma \vdash u : A}\ \text{hyp}_2$$
$$\frac{\Delta \mid \Gamma, x : A, \Gamma' \vdash}{\Delta \mid \Gamma, x : A, \Gamma' \vdash x : A}\ \text{Var}$$	$$\frac{}{\Delta \mid \Gamma, x : A, \Gamma' \vdash x : A}\ \text{hyp}_1$$

$\flat$ rules

Crisp type theory	Pfenning and Davies' Modal type theory
$\dfrac{\Delta \mid \cdot \vdash A \text{ type}}{\Delta \mid \Gamma \vdash \flat A \text{ type}}$ $\flat$-Form	(implicit in $\Box I$ rule)
$\dfrac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash M^{\flat} : \flat A}$ $\flat$-Intro	$\dfrac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash \mathsf{box}\, M : \Box A}\ \Box I$
$\dfrac{\Delta \mid \Gamma \vdash M : \flat A \qquad \Delta, u :: A \mid \Gamma \vdash N : C[u^{\flat}/x] \qquad \Delta \mid \Gamma, x : \flat A \vdash C \text{ type}}{\Delta \mid \Gamma \vdash (\mathsf{let}\ u^{\flat} := M \mathsf{\ in\ } N) : C[M/x]}$ $\flat$-Elim	$\dfrac{\Delta \mid \Gamma \vdash M : \Box A \qquad \Delta, u :: A \mid \Gamma \vdash N : C}{\Delta \mid \Gamma \vdash \mathsf{let\ box}\ u = M \mathsf{\ in\ } N : C}\ \Box E$

(plus computation rules)

② Modelling dependent type theory

Modelling dependent type theory

Let $\mathcal{C}$ be a category with a class of maps $\mathcal{D} \subseteq \mathcal{C}^{\to}$.

"display maps" – all pullbacks of members of $\mathcal{D}$ exist and belong to $\mathcal{D}$

Ingredients of a type theory

contexts

Γ, Δ, Θ

types-in-context

$\Gamma \vdash A \text{ type}$

terms-in-context

$\Gamma \vdash M : A$

substitution

$$\frac{x : A \vdash B(x) \text{ type} \qquad y : C \vdash N : A}{y : C \vdash B(N) \text{ type}}$$

objects Γ, Δ, Θ in $\mathcal{C}$

display maps
$$\begin{tikzcd} A \arrow[d] \\ \Gamma \end{tikzcd}$$

sections of display maps
$$\begin{tikzcd} A \arrow[d] \\ \Gamma \arrow[u, bend left, "M"] \end{tikzcd}$$

pullback
$$\begin{tikzcd} B(N) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d] \\ C \arrow[r, "N"'] & A \end{tikzcd}$$

Problem

- Substitution in type theory is strictly functorial, while pullback in general is not.

Solutions

comprehension categories, display map categories, categories with attributes, contextual categories, categories with families, natural models

Advantages of Natural models (Awodey, 2016)

- smaller distance between the syntax and the categorical model
- distinguishes between a type in context and extension by a single type

Definition A natural model is a category $\mathcal{C}$ with

objects $\Gamma, \Delta, \ldots$

morphisms $\sigma : \Delta \to \Gamma$

contexts "$\Gamma \vdash$", substitutions

and

(i) a specified terminal object $1_{\mathcal{C}}$

empty context "$\bullet \vdash$"

(ii) presheaves $\mathcal{U}, \tilde{\mathcal{U}}$ over $\mathcal{C}$

$\mathcal{U}(\Gamma)$ set of types in context Γ

$\tilde{\mathcal{U}}(\Gamma)$ set of terms in context Γ

(iii) a natural transformation $ty : \tilde{\mathcal{U}} \longrightarrow \mathcal{U}$

$p_\Gamma : \tilde{\mathcal{U}}(\Gamma) \longrightarrow \mathcal{U}(\Gamma)$

sends a term to its unique type

Observation

Given $\mathcal{U}, \tilde{\mathcal{U}} \in [\mathcal{C}^{op}, \mathrm{Set}]$ and $ty : \tilde{\mathcal{U}} \longrightarrow \mathcal{U}$,

by Yoneda we have

$$\frac{\alpha \in \mathcal{U}(\Gamma)}{\Gamma = \mathbf{y}_\Gamma \xrightarrow{\ \alpha\ } \mathcal{U}} \, , \qquad \frac{a \in \tilde{\mathcal{U}}(\Gamma)}{\Gamma = \mathbf{y}_\Gamma \xrightarrow{\ a\ } \tilde{\mathcal{U}}}$$

so "typing" corresponds to a commutative triangle

$$\begin{tikzcd} & \tilde{\mathcal{U}} \arrow[d, "ty"] \\ \Gamma \arrow[ur, "a"] \arrow[r, "\alpha"'] & \mathcal{U} \end{tikzcd} \qquad \Gamma \vdash a : \alpha$$

(iv) specified pullbacks, i.e.

for each Γ in $\mathcal{C}_0$ and each $\alpha : \Gamma \longrightarrow \mathcal{U}$ in $\widehat{\mathcal{C}}$,
there is a specified pullback

$$\begin{tikzcd}
\Gamma.\alpha \arrow[r, "q_\alpha"] \arrow[d, "p_\alpha"'] \arrow[dr, phantom, "\lrcorner", very near start] & \tilde{\mathcal{U}} \arrow[d, "ty"] \\
\Gamma \arrow[r, "\alpha"'] & \mathcal{U}
\end{tikzcd}$$

$\Gamma, \alpha \vdash q_\alpha : \alpha[p_\alpha]$

Remarks

- (ii)–(iv) abbreviated by "$ty : \tilde{\mathcal{U}} \to \mathcal{U}$ is locally representable"
- substitution into a type is now given by composition, which is strictly associative!

$$\begin{tikzcd} & & \tilde{\mathcal{U}} \arrow[d, "ty"] \\ \Delta \arrow[r, "\sigma"'] & \Gamma \arrow[ur, "a"] \arrow[r, "\alpha"'] & \mathcal{U} \end{tikzcd}$$

⚠ explicit substitution

$$\frac{\Gamma \vdash a : \alpha \qquad \sigma : \Delta \to \Gamma}{\Delta \vdash a[\sigma] : \alpha[\sigma]}$$

- We can define structure-preserving maps between categories with natural model structure, so we have a category

<u>NMCat</u> objects - natural model categories
arrows - natural model functors

- We won't look at type constructors.

Example – presheaf topos

Proposition Suppose $\mathcal{C}$ has a class of display maps $D \in \mathcal{C}_1$. Then there is a representable natural transformation

$$ty : \tilde{\mathcal{U}} \longrightarrow \mathcal{U}$$

over $\mathcal{C}$ defined as follows:

1) $$\mathcal{U}(\Gamma) := \left\{ \begin{tikzcd} & \Theta \arrow[d, "\alpha"] \\ \Gamma \arrow[r, "s"'] & \Delta \end{tikzcd} \;\middle|\; \alpha \in D \right\}$$

2) $$\tilde{\mathcal{U}}(\Gamma) := \left\{ \begin{tikzcd} & \Theta \arrow[d, "\alpha"] \\ \Gamma \arrow[r, "s"'] \arrow[ur, "a"] & \Delta \end{tikzcd} \;\middle|\; \alpha \in D \right\}$$

3 Modelling crisp type theory

Existing work

- de Paiva and Ritter
  - "Fibrational modal type theory" 2016
- Shulman
  - "Semantics of multimodal adjoint type theory" 2023
- Zwanziger
  - "The natural display topos of coalgebras" PhD thesis, 2023

On modelling crisp type theory

– Licata, Orton, Pitts, Spitters 2018, referencing Shulman 2018

"very little is required of a category $\mathcal{C}$ for the presheaf topos $\widehat{\mathcal{C}}$ to soundly interpret [crisp type theory] using the comonad $\flat$ … Although the details remain to be worked out, it appears that … the only additional condition needed is that this comonad is idempotent"

$$\begin{tikzcd} \widehat{\mathbb{C}} \arrow[loop above, "\flat"] \arrow[d, bend left, "\Gamma"] \\ \mathrm{Set} \arrow[u, bend left, "\Delta"] \arrow[u, phantom, "\dashv"] \end{tikzcd}$$

constant presheaf on X $\;\mapsfrom\;$ X

$A \;\mapsto\; \{1 \xrightarrow{a} A\}$ global sections

Is it obvious how this is a model?

Zooming out

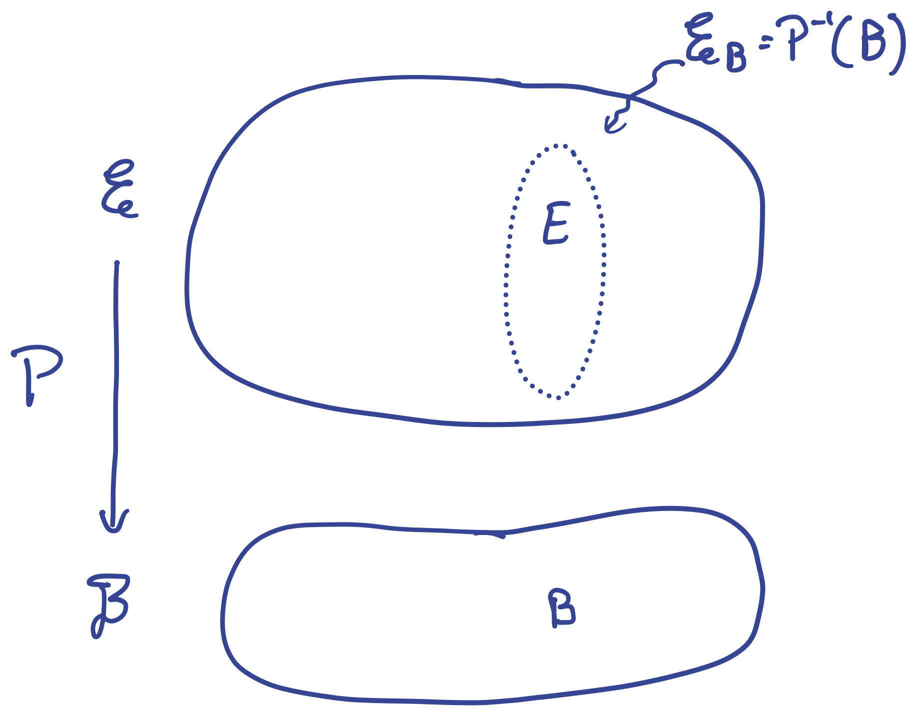
Question what are the features of the language and how might we model them more abstractly?

Feature 1: Split context

Grothendieck fibration

For a context $\Delta \mid \Gamma$, want to capture the dependency of Γ on Δ.

Ask for a functor, viewed as a display family of categories



$\Delta \rightsquigarrow B \in \mathcal{B}$

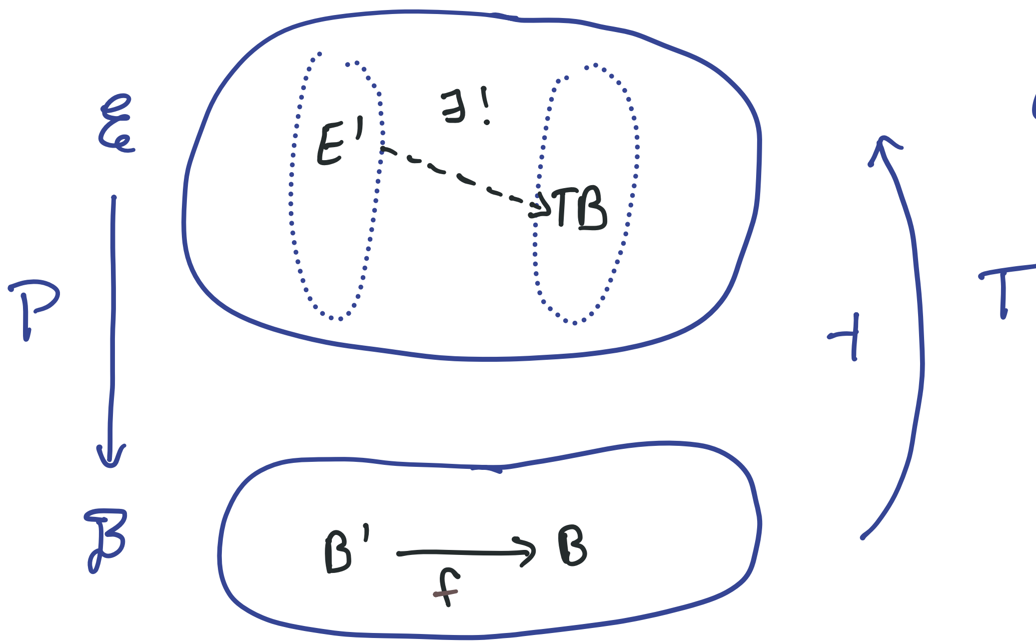
$\Delta \mid \Gamma \rightsquigarrow E \in \mathcal{E}_B$

Feature 2: Empty contexts

The context may have only crisp variables

$$\Delta \mid \cdot$$

Ask for a right adjoint right inverse to P.



TB is the terminal object in the fibre $\mathcal{E}_B$

$$\Delta \rightsquigarrow B \in \mathcal{B}$$
$$\Delta \mid \Gamma \rightsquigarrow E \in \mathcal{E}_B$$
$$\Delta \mid \cdot \rightsquigarrow TB \in \mathcal{E}_B$$
$$\Delta' \xrightarrow[\sigma]{} \Delta \rightsquigarrow B' \xrightarrow[f]{} B \in \mathcal{B}$$

Feature 2: Empty contexts (cont.)

The context may be empty

$\cdot \mid \cdot$

Ask for a terminal object in $\mathcal{B}$

Consequence: $\mathcal{E}$ has a terminal object

Feature 3: Extension of the non-crisp context

$$\frac{\Delta \mid \Gamma \vdash \alpha \text{ type}}{\Delta \mid \Gamma, \alpha \text{ context}}$$

To implement type extension in the total space, ask for a locally representable map

$$ty_{\mathcal{E}} : \tilde{\mathcal{U}}_{\mathcal{E}} \longrightarrow \mathcal{U}_{\mathcal{E}} \quad \text{in } \hat{\mathcal{E}}$$

such that the specified pullback along $E \xrightarrow{\alpha} \mathcal{U}_{\mathcal{E}}$,

$$\begin{tikzcd} E.\alpha \arrow[r] \arrow[d, "p"'] \arrow[dr, phantom, "\lrcorner", very near start] & \tilde{\mathcal{U}}_{\mathcal{E}} \arrow[d, "ty_{\mathcal{E}}"] \\ E \arrow[r, "\alpha"'] & \mathcal{U}_{\mathcal{E}} \end{tikzcd},$$

lies in the fibre $\mathcal{E}_{PE}$

$\Delta \mid \Gamma \rightsquigarrow E \in \mathcal{E}_B$

$\Delta \mid \Gamma \vdash \alpha \text{ type} \rightsquigarrow E \xrightarrow{\alpha} \mathcal{U}_{\mathcal{E}} \in \hat{\mathcal{E}}$

$\Delta \mid \Gamma, \alpha \text{ context} \rightsquigarrow E.\alpha \xrightarrow{p} E \in \mathcal{E}_B$

Feature 3: Extension of the non-crisp context (cont.)

$\Rightarrow$ the fibres are natural model categories

In the fibre $\mathcal{E}_B$ over B there is

- a specified terminal object TB
- a locally representable map

$$\tilde{\mathcal{U}}|_{\mathcal{E}_B} \xrightarrow{ty} \mathcal{U}|_{\mathcal{E}_B}$$

The natural model structure is preserved between the fibres.

Feature 4: extension of the crisp context

$$\frac{\Delta \mid \cdot \vdash \alpha \ \mathsf{type}}{\Delta, \alpha \mid \cdot \ \mathsf{context}}$$

To implement type extension in the base, ask that the following map defined using $ty_{\mathcal{E}}$ in $\widehat{\mathcal{E}}$ is locally representable in $\widehat{\mathcal{B}}$:

$$\begin{array}{c} \widetilde{\mathcal{U}}_{\mathcal{B}} := \widetilde{\mathcal{U}}_{\mathcal{E}} \circ T^{op} \\ \Big\downarrow {\scriptstyle ty} \\ \mathcal{U}_{\mathcal{B}} := \mathcal{U}_{\mathcal{E}} \circ T^{op} \end{array}$$

$\Delta \vdash \alpha$ type
$\Delta \mid \cdot \vdash \alpha$ type
have the same interpretation

So $\mathcal{B}$ has a relativised natural model structure

Summary

Let $P : \mathcal{E} \to \mathcal{B}$ be a functor.

Axioms

1) P has a right adjoint right inverse, T.
2) $\mathcal{B}$ has a specified terminal object.
3) There is a locally representable map

$$ty : \tilde{\mathcal{U}}_{\mathcal{E}} \longrightarrow \mathcal{U}_{\mathcal{E}} \text{ in } \widehat{\mathcal{E}}$$

whose local representatives are given fibrewise.

4) $\tilde{\mathcal{U}}_{\mathcal{E}} \circ T^{op} \longrightarrow \mathcal{U}_{\mathcal{E}} \circ T^{op}$ in $\widehat{\mathcal{B}}$ is locally representable.

Claim: This models the context in crisp type theory.

Zooming back in

Let $\mathcal{C}$ be a category with

1) a terminal object
2) a class of display maps $\mathcal{D}$
3) an idempotent comonad $(\flat, \varepsilon_C : \flat C \to C)$ where $\flat$ preserves
   - the terminal object
   - display maps and their pullbacks

example:
$$\begin{tikzcd} \widehat{\mathbb{C}} \arrow[loop right, "\flat"] \arrow[d, bend left, "\Gamma"] \arrow[d, phantom, "\dashv"] \\ \mathrm{Set} \arrow[u, bend left, "\Delta"] \end{tikzcd}$$

Theorem The above category possesses the structure of our abstract model.

Proof sketch

Let $\mathcal{B} = \mathcal{C}_b \hookrightarrow \mathcal{C}$

$\mathcal{E} = \mathcal{C} \downarrow \mathcal{C}_b$

$P = \mathrm{cod} : \mathcal{C} \downarrow \mathcal{C}_b \longrightarrow \mathcal{C}_b$

$T : \mathcal{C}^b \longrightarrow \mathcal{C} \downarrow \mathcal{C}_b$

full subcategory of objects c in $\mathcal{C}_0$ with $\varepsilon_c = id_c$

1) Show T is a right adjoint right inverse to P
   ↳ (a general result about comma categories)

2) Show that $\mathcal{C}_b$ has a terminal object

3) Define a locally representable map $\tilde{\mathcal{U}} \xrightarrow{ty} \mathcal{U}$ in $\mathcal{C} \downarrow \mathcal{C}_b$

4) Show that the restriction of ty to T is locally representable

$$\tilde{\mathcal{U}} \circ T^{op} \longrightarrow \mathcal{U} \circ T^{op}$$

Ongoing work

- Modelling $\Box$ and the "let" construct

$$\Box : \mathcal{U} \circ T^{op} \longrightarrow \mathcal{U} \circ T^{op}$$

$$\Box : \tilde{\mathcal{U}} \circ T^{op} \longrightarrow \tilde{\mathcal{U}} \circ T^{op}$$

- Formalising the relationship between the type theory and the categorical model

$$\begin{tikzcd}
\text{Theories} \arrow[rr, bend right, "\text{syntactic category}"'] \arrow[rr, phantom, "\top"] & & \text{Categories} \arrow[ll, bend right, "\text{internal logic}"']
\end{tikzcd}$$

Thanks