Strictification of categories with families

Ambrus Kaposi

Eötvös Loránd University, Budapest

j.w.w. Loïc Pujet

HoTT/UF 2025
Genova
15 April 2025

What is type theory?

syntactic

- actual implementations: Agda, Coq, Idris, Lean
- extrinsic syntax: Abel–Öhman–Vezzosi 2018, Martin-Löf à la Coq, MetaRocq, Lean4Lean
- intrinsic unquotiented ASTs: Danielsson 2006, Chapman 2009
- CwF: Dybjer 1996, Altenkirch–K. 2016
- natural models: Awodey 2018
- comprehension categories: Brunerie–de Boer 2020
- LCCC: Seely 1984

semantic

What is type theory?

syntactic

- actual implementations: Agda, Cc
- extrinsic syntax: Abel–Öhman–Vezz
  MetaRocq, Lean4Lean
- intrinsic unquotiented ASTs: Dani
- CwF: Dybjer 1996, Altenkirch–K. 201
- natural models: Awodey 2018
- comprehension categories: Bruneri
- LCCC: Seely 1984

semantic

↑ more practical,
observable computations,
ad-hoc choices

↓ elegance, abstraction,
easier metatheory,
forced choices

Normalisation

- means that every term can be reduced to a normal form

  $$\mathsf{suc}\,(\mathsf{suc}\,((\lambda x.x + x)\;(\mathsf{suc}\,\mathsf{zero}))) \rightsquigarrow \mathsf{suc}\,(\mathsf{suc}\,(\mathsf{suc}\,(\mathsf{suc}\,\mathsf{zero})))$$

  - used in type checking

- In the CwF setting, we define normal forms inductively with a map back into terms:

  $$n ::= x \mid n\,v \qquad\qquad \ulcorner - \urcorner : \mathsf{Nf}\,\Gamma\,A \to \mathsf{Tm}\,\Gamma\,A$$
  $$v ::= n \mid \lambda x.v$$

  Then normalisation is a section of $\ulcorner - \urcorner$.

- Reduction-free normalisation:
  - Simple types (Altenkirch–Hofmann–Streicher 1995)
  - Dependent types (Altenkirch–K. 2016)
  - Main idea: proof-relevant logical relation (categorical gluing)

Computer formalisation

There are several formalisations of type theory in extrinsic style:

- Abel–Öhman–Vezzosi 1998, MetaRocq (2014–2025), Martin-Löf à la Coq (Adjedj–Lennon-Bertrand–Maillard–Pédrot–Pujet 2024), Lean4Lean (Carneiro 2024)

Only very small CwF-style formalisations, and they are difficult to use. No formalisation of gluing-style normalisation. Reasons:

1. the syntax needs quotients
2. transport hell

$$\mathsf{rev} : \mathsf{Vec}\,A\,n \to \mathsf{Vec}\,A\,n$$

$$\mathsf{rev}\,[] :\equiv []$$

$$\mathsf{rev}\,(x :: xs) \; \underbrace{(x :: xs)}_{:\mathsf{Vec}\,A\,(1+n)} \; :\equiv \mathsf{rev}\,xs \mathbin{{+}{+}} (x :: []) \; \underbrace{\mathsf{rev}\,xs \mathbin{{+}{+}} (x :: [])}_{:\mathsf{Vec}\,A\,(n+1)} (\mathsf{comm}_{+}\,n$$

$$\mathsf{revrev} : (xs : \mathsf{Vec}\,A\,n) \to \mathsf{rev}\,(\mathsf{rev}\,xs) = xs$$

$$\mathsf{revrev}\,[] :\equiv \mathsf{refl}$$

$$\mathsf{revrev}\,(x :: xs) :\equiv\, ? \cdot \mathsf{ap}\,(x :: -)\,(\mathsf{revrev}\,xs)$$

Computer formalisation

There are several formalisations of type theory in extrinsic style:

- Abel–Öhman–Vezzosi 1998, MetaRocq (2014–2025), Martin-Löf à la Coq (Adjedj–Lennon-Bertrand–Maillard–Pédrot–Pujet 2024), Lean4Lean (Carneiro 2024)

Only very small CwF-style formalisations, and they are difficult to use. No formalisation of gluing-style normalisation. Reasons:

1. the syntax needs quotients
2. transport hell

   $$\mathsf{rev} : \mathsf{Vec}\,A\,n \to \mathsf{Vec}\,A\,n$$

   $$\mathsf{rev}\,[] :\equiv []$$

   $$\mathsf{rev}\,(x :: xs)\ \underbrace{(x :: xs)}_{:\mathsf{Vec}\,A\,(1+n)}\ :\equiv \mathsf{rev}\,xs +\!\!+ (x :: [])\ \underbrace{\mathsf{rev}\,xs +\!\!+ (x :: [])}_{:\mathsf{Vec}\,A\,(n+1)}(\mathsf{comm}_+\,n$$

3. substitution laws are propositional, rather than definitional

   $$(\mathsf{app}\,t\,u)[\gamma] = \mathsf{app}\,(t[\gamma])\,(u[\gamma])$$

Problems with intrinsic syntax

1. the syntax needs quotients
   - Cubical Type Theory / OTT / OTT via rewrite rules
2. transport hell
3. substitution laws are propositional

Solution to 2,3: make the equations in the syntax definitional:

- extensional type theory, conservativity
- rewrite rules
- replace the weak model with an equivalent, stricter one
  - examples:
    - difference lists in the Haskell Prelude: replace List by $\mathsf{List} \to \mathsf{List}$
    - strictification of a category by Yoneda: replace $C(J, I)$ by $\mathsf{y}_J \stackrel{\cdot}{\to} \mathsf{y}_I$
  - non-example:
    - right adjoint splitting (Hofmann 1994)
    - left adjoint splitting, local universes (Lumsdaine–Warren 2015)

Weak CwF

$$
\begin{aligned}
&\mathsf{Con} &&: \mathsf{Set}\\
&\mathsf{Sub} &&: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Set}\\
&\mathsf{Ty} &&: \mathsf{Con} \to \mathsf{Set}\\
&\mathsf{Tm} &&: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Set}\\
&- \circ - &&: \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Theta\,\Gamma\\
&\mathsf{ass} &&: (\gamma \circ \delta) \circ \theta = \gamma \circ (\delta \circ \theta)\\
&\mathsf{id} &&: \mathsf{Sub}\,\Gamma\,\Gamma\\
&\mathsf{idl} &&: \mathsf{id} \circ \gamma = \gamma\\
&\mathsf{idr} &&: \gamma \circ \mathsf{id} = \gamma\\
&\diamond &&: \mathsf{Con}\\
&\epsilon &&: \mathsf{Sub}\,\Gamma\,\diamond\\
&\diamond\eta &&: (\sigma : \mathsf{Sub}\,\Gamma\,\diamond) \to \sigma = \epsilon\\
&-[-] &&: \mathsf{Ty}\,\Gamma \to \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{Ty}\,\Delta\\
&[\circ] &&: A[\gamma \circ \delta] = A[\gamma][\delta]\\
&[\mathsf{id}] &&: A[\mathsf{id}] = A
\end{aligned}
$$

$$
\begin{aligned}
&-[-] &&: \mathsf{Tm}\,\Gamma\,A \to (\gamma : \mathsf{Sub}\,\Delta\,\Gamma) \to\\
& && \quad \mathsf{Tm}\,\Delta\,(A[\gamma])\\
&[\circ] &&: [\circ]_*\,(a[\gamma \circ \delta]) = a[\gamma][\delta]\\
&[\mathsf{id}] &&: [\mathsf{id}]_*\,(a[\mathsf{id}]) = a\\
&- \triangleright - &&: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}\\
&-,- &&: (\gamma : \mathsf{Sub}\,\Delta\,\Gamma) \to \mathsf{Tm}\,\Delta\,(A[\gamma]) \to\\
& && \quad \mathsf{Sub}\,\Delta\,(\Gamma \triangleright A)\\
&,\circ &&: (\gamma, a) \circ \delta = (\gamma \circ \delta, [\circ]_*\,(a[\delta]))\\
&\mathsf{p} &&: \mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma\\
&\mathsf{q} &&: \mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}])\\
&\triangleright\beta_1 &&: \mathsf{p} \circ (\gamma, a) = \gamma\\
&\triangleright\beta_2 &&: ([\circ] \cdot \triangleright\beta_1)_*\,(\mathsf{q}[\gamma, a]) = a\\
&\triangleright\eta &&: \mathsf{id} = (\mathsf{p}, \mathsf{q})
\end{aligned}
$$

Strict CwF

$$
\begin{aligned}
&\mathsf{Con} && : \mathsf{Set} \\
&\mathsf{Sub} && : \mathsf{Con} \to \mathsf{Con} \to \mathsf{Set} \\
&\mathsf{Ty} && : \mathsf{Con} \to \mathsf{Set} \\
&\mathsf{Tm} && : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Set} \\
&- \circ - && : \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{Sub}\,\Theta\,\Delta \to \mathsf{Sub}\,\Theta\,\Gamma \\
&\mathsf{ass} && : (\gamma \circ \delta) \circ \theta \equiv \gamma \circ (\delta \circ \theta) \\
&\mathsf{id} && : \mathsf{Sub}\,\Gamma\,\Gamma \\
&\mathsf{idl} && : \mathsf{id} \circ \gamma \equiv \gamma \\
&\mathsf{idr} && : \gamma \circ \mathsf{id} \equiv \gamma \\
&\diamond && : \mathsf{Con} \\
&\epsilon && : \mathsf{Sub}\,\Gamma\,\diamond \\
&\diamond\eta && : (\sigma : \mathsf{Sub}\,\Gamma\,\diamond) \to \sigma \equiv \epsilon \\
&-[-] && : \mathsf{Ty}\,\Gamma \to \mathsf{Sub}\,\Delta\,\Gamma \to \mathsf{Ty}\,\Delta \\
&[\circ] && : A[\gamma \circ \delta] \equiv A[\gamma][\delta] \\
&[\mathsf{id}] && : A[\mathsf{id}] \equiv A
\end{aligned}
$$

$$
\begin{aligned}
&-[-] && : \mathsf{Tm}\,\Gamma\,A \to (\gamma : \mathsf{Sub}\,\Delta\,\Gamma) \to \\
& && \quad \mathsf{Tm}\,\Delta\,(A[\gamma]) \\
&[\circ] && : a[\gamma \circ \delta] \equiv a[\gamma][\delta] \\
&[\mathsf{id}] && : a[\mathsf{id}] \equiv a \\
&- \triangleright - && : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con} \\
&-,- && : (\gamma : \mathsf{Sub}\,\Delta\,\Gamma) \to \mathsf{Tm}\,\Delta\,(A[\gamma]) \to \\
& && \quad \mathsf{Sub}\,\Delta\,(\Gamma \triangleright A) \\
&,\circ && : (\gamma, a) \circ \delta \equiv (\gamma \circ \delta, a[\delta]) \\
&\mathsf{p} && : \mathsf{Sub}\,(\Gamma \triangleright A)\,\Gamma \\
&\mathsf{q} && : \mathsf{Tm}\,(\Gamma \triangleright A)\,(A[\mathsf{p}]) \\
&\triangleright\beta_1 && : \mathsf{p} \circ (\gamma, a) \equiv \gamma \\
&\triangleright\beta_2 && : \mathsf{q}[\gamma, a] \equiv a \\
&\triangleright\eta && : \mathsf{id} = (\mathsf{p}, \mathsf{q})
\end{aligned}
$$

Booleans in a weak CwF (i)

$$
\begin{aligned}
&\mathsf{Bool} &&: \mathsf{Ty}\,\Gamma \\
&\mathsf{Bool}[] &&: \mathsf{Bool}[\gamma] = \mathsf{Bool} \\
&\mathsf{true} &&: \mathsf{Tm}\,\Gamma\,\mathsf{Bool} \\
&\mathsf{true}[] &&: \mathsf{Bool}[]_*\,(\mathsf{true}[\gamma]) = \mathsf{true} \\
&\mathsf{false} &&: \mathsf{Tm}\,\Gamma\,\mathsf{Bool} \\
&\mathsf{false}[] &&: \mathsf{Bool}[]_*\,(\mathsf{false}[\gamma]) = \mathsf{false} \\
&\mathsf{ind} &&: (P : \mathsf{Ty}\,(\Gamma \triangleright \mathsf{Bool})) \to \mathsf{Tm}\,\Gamma\,(P[\langle \mathsf{true} \rangle]) \to \mathsf{Tm}\,\Gamma\,(P[\langle \mathsf{false} \rangle]) \to \\
& && \quad (b : \mathsf{Tm}\,\Gamma\,\mathsf{Bool}) \to \mathsf{Tm}\,\Gamma\,(P[\langle b \rangle]) \\
&\mathsf{ind}[] &&: (\alpha\,b)_*\,((\mathsf{ind}\,P\,p\,p'\,b)[\gamma]) = \\
& && \quad \mathsf{ind}\,(\mathsf{Bool}[]_*\,(P[\gamma^{\uparrow}]))\,(\mathsf{true}[]_*\,((\alpha\,\mathsf{true})_*\,(p[\gamma])))\,(\mathsf{false}[]_*\,((\alpha\,\mathsf{false})_*\,(p'[\gamma]))) \\
& && \qquad (\mathsf{Bool}[]_*\,(b[\gamma])) \\
&\mathsf{Bool}\beta_1 &&: \mathsf{ind}\,P\,p\,p'\,\mathsf{true} = p \\
&\mathsf{Bool}\beta_2 &&: \mathsf{ind}\,P\,p\,p'\,\mathsf{false} = p'
\end{aligned}
$$

where

$$\alpha : (u : \mathsf{Tm}\,\Gamma\,\mathsf{Bool}) \to P[\langle u \rangle][\gamma] = P[\mathsf{Bool}[]_*\,(\gamma^{\uparrow})][\langle \mathsf{Bool}[]_*\,(u[\gamma]) \rangle]$$

Booleans in a weak CwF (ii)

$$
\begin{aligned}
\alpha\, u : & P[\langle u\rangle][\gamma] && =([\circ]) \\
& P[\langle u\rangle \circ \gamma] && \equiv \\
& P[(\mathsf{id}, [\mathsf{id}]_* \, u) \circ \gamma] && =(,\circ) \\
& P[\mathsf{id} \circ \gamma, [\circ]_* \, (([\mathsf{id}]_* \, u)[\gamma])] && =(-[-] \text{ and transport}) \\
& P[\mathsf{id} \circ \gamma, [\circ]_* \, ([\mathsf{id}]_* \, (u[\gamma]))] && =(\mathsf{idl}) \\
& P[\gamma, \mathsf{idl}_* \, ([\circ]_* \, ([\mathsf{id}]_* \, (u[\gamma])))] && =(\cdot_*) \\
& P[\gamma, ([\mathsf{id}] \cdot [\circ] \cdot \mathsf{idl})_* \, (u[\gamma])] && \equiv \\
& P[\gamma, u[\gamma]] && \equiv \\
& P[\gamma, ([\mathsf{id}] \cdot [\mathsf{id}])_* \, (u[\gamma])] && =(\cdot_*) \\
& P[\gamma, [\mathsf{id}]_* \, ([\mathsf{id}]_* \, (u[\gamma]))] && =(\triangleright\beta_2) \\
& P[\gamma, [\mathsf{id}]_* \, (([\circ] \cdot \triangleright\beta_1)_* \, (\mathsf{q}[\langle u[\gamma]\rangle]))] && \equiv \\
& P[\gamma, [\mathsf{id}]_* \, (([\circ] \cdot [\circ] \cdot \mathsf{ass} \cdot \triangleright\beta_1 \cdot \mathsf{idr} \cdot [\mathsf{id}])_* \, (\mathsf{q}[\langle u[\gamma]\rangle]))] && =(\cdot_*) \\
& P[\gamma, ([\circ] \cdot [\circ] \cdot \mathsf{ass} \cdot \triangleright\beta_1 \cdot \mathsf{idr} \cdot [\mathsf{id}] \cdot [\mathsf{id}])_* \, (\mathsf{q}[\langle u[\gamma]\rangle])] && \equiv \\
& P[\gamma, ([\circ] \cdot [\circ] \cdot \mathsf{ass} \cdot \triangleright\beta_1 \cdot \mathsf{idr})_* \, (\mathsf{q}[\langle u[\gamma]\rangle])] && =(\cdot_*) \\
& P[\gamma, \mathsf{idr}_* \, (\triangleright\beta_{1*} \, (\mathsf{ass}_* \, ([\circ]_* \, ([\circ]_* \, (\mathsf{q}[\langle u[\gamma]\rangle])))))] && =(\mathsf{idr}) \\
& P[\gamma \circ \mathsf{id}, \triangleright\beta_{1*} \, (\mathsf{ass}_* \, ([\circ]_* \, ([\circ]_* \, (\mathsf{q}[\langle u[\gamma]\rangle]))))] && =(\triangleright\beta_1) \\
& P[\gamma \circ (\mathsf{p} \circ \langle u[\gamma]\rangle), \mathsf{ass}_* \, ([\circ]_* \, ([\circ]_* \, (\mathsf{q}[\langle u[\gamma]\rangle])))] && =(\mathsf{ass}) \\
& P[(\gamma \circ \mathsf{p}) \circ \langle u[\gamma]\rangle, [\circ]_* \, ([\circ]_* \, (\mathsf{q}[\langle u[\gamma]\rangle]))] && =(-[-] \text{ and transport}) \\
& P[(\gamma \circ \mathsf{p}) \circ \langle u[\gamma]\rangle, [\circ]_* \, (([\circ]_* \, \mathsf{q})[\langle u[\gamma]\rangle])] && =(,\circ) \\
& P[(\gamma \circ \mathsf{p}, [\circ]_* \, \mathsf{q}) \circ \langle u[\gamma]\rangle] && \equiv \\
& P[(\gamma \circ \mathsf{p}, [\circ]_* \, \mathsf{q}) \circ \langle (\mathsf{Bool}[] \cdot \mathsf{Bool}[])_* \, (u[\gamma])\rangle] && =(\cdot_*) \\
& P[(\gamma \circ \mathsf{p}, [\circ]_* \, \mathsf{q}) \circ \langle \mathsf{Bool}[]_* \, (\mathsf{Bool}[]_* \, (u[\gamma]))\rangle] && =(\langle - \rangle \text{ and transport}) \\
& P[(\gamma \circ \mathsf{p}, [\circ]_* \, \mathsf{q}) \circ \mathsf{Bool}[]_* \, \langle \mathsf{Bool}[]_* \, (u[\gamma])\rangle] && =(- \circ - \text{ and transport}) \\
& P[(\mathsf{Bool}[]_* \, (\gamma \circ \mathsf{p}, [\circ]_* \, \mathsf{q})) \circ \langle \mathsf{Bool}[]_* \, (u[\gamma])\rangle] && \equiv \\
& P[(\mathsf{Bool}[]_* \, (\gamma^{\uparrow})) \circ \langle \mathsf{Bool}[]_* \, (u[\gamma])\rangle] && =([\circ]) \\
& P[\mathsf{Bool}[]_* \, (\gamma^{\uparrow})][\langle \mathsf{Bool}[]_* \, (u[\gamma])\rangle] &&
\end{aligned}
$$

Substitution-strict booleans in a strict CwF

$$
\begin{aligned}
&\mathsf{Bool} &&: \mathsf{Ty}\,\Gamma \\
&\mathsf{Bool}[] &&: \mathsf{Bool}[\gamma] \equiv \mathsf{Bool} \\
&\mathsf{true} &&: \mathsf{Tm}\,\Gamma\,\mathsf{Bool} \\
&\mathsf{true}[] &&: \mathsf{true}[\gamma] \equiv \mathsf{true} \\
&\mathsf{false} &&: \mathsf{Tm}\,\Gamma\,\mathsf{Bool} \\
&\mathsf{false}[] &&: \mathsf{false}[\gamma] \equiv \mathsf{false} \\
&\mathsf{ind} &&: (P : \mathsf{Ty}\,(\Gamma \triangleright \mathsf{Bool})) \to \mathsf{Tm}\,\Gamma\,(P[\langle \mathsf{true} \rangle]) \to \mathsf{Tm}\,\Gamma\,(P[\langle \mathsf{false} \rangle]) \to \\
& &&\quad (b : \mathsf{Tm}\,\Gamma\,\mathsf{Bool}) \to \mathsf{Tm}\,\Gamma\,(P[\langle b \rangle]) \\
&\mathsf{ind}[] &&: (\mathsf{ind}\,P\,p\,p'\,b)[\gamma] \equiv \mathsf{ind}\,(P[\gamma^{\uparrow}])\,(p[\gamma])\,(p'[\gamma])\,(b[\gamma]) \\
&\mathsf{Bool}\beta_1 &&: \mathsf{ind}\,P\,p\,p'\,\mathsf{true} = p \\
&\mathsf{Bool}\beta_2 &&: \mathsf{ind}\,P\,p\,p'\,\mathsf{false} = p'
\end{aligned}
$$

Higher-order abstract syntax

- $\mathsf{Psh}(C)$ is a CwF with Π types for any C
- if C is a model of type theory, then *internal to* $\mathsf{Psh}(C)$ we have

$$\mathsf{Ty} : \mathsf{Set}, \qquad \mathsf{Tm} : \mathsf{Ty} \to \mathsf{Set}.$$

- if C supports Π, then this universe is closed under Π:

$$\Pi : (A : \mathsf{Ty}) \to (\mathsf{Tm}\, A \to \mathsf{Ty}) \to \mathsf{Ty}$$

- *internal to* $\mathsf{Psh}(C)$, we define a model closed under the same type formers as C:

$$\begin{aligned} &\mathsf{Con} &&:\equiv \mathsf{Set} && \mathsf{Ty}\,\Gamma &&:\equiv \Gamma \to \mathsf{Ty} \\ &\mathsf{Sub}\,\Delta\,\Gamma &&:\equiv \Delta \to \Gamma && \mathsf{Tm}\,\Gamma\,A &&:\equiv (\gamma_\bullet : \Gamma) \to \mathsf{Tm}\,(A\,\gamma_\bullet) \\ &A[\gamma] &&:\equiv A \circ \gamma && \Pi\,A\,B\,\gamma_\bullet &&:\equiv \Pi\,(A\,\gamma_\bullet)\,(\lambda a_\bullet.B\,(\gamma_\bullet, a_\bullet)) \end{aligned}$$

- This is a substitution-strict model, we call it the contextualisation of P.
- We would like to externalise.

HOAS externally, abstractly

- We assume P a strict CwF_{Π}. We think about it as $P \equiv \mathsf{Psh}(\mathcal{C})$.
- A P-universe closed under Π and Bool is:

$$\begin{array}{llll} \mathsf{Ty} : \mathsf{Con}_P & \Pi : \mathsf{Sub}_P\,(\mathsf{Ty} \triangleright \mathsf{Tm} \Rightarrow \mathsf{K}\,\mathsf{Ty})\,\mathsf{Ty} & \dots \\ \mathsf{Tm} : \mathsf{Ty}_P\,\mathsf{Ty} & \mathsf{Bool} : \mathsf{Sub}_P\,\diamond\,\mathsf{Ty} & \end{array}$$

  - If $P \equiv \mathsf{Psh}(C)$, then $\mathcal{C}$ with a P-universe is almost the same as a model.

- The P-contextualisation of a P-universe:

$$\begin{array}{llll} \mathsf{Con} & :\equiv \mathsf{Con}_P & \mathsf{Ty}\,\Gamma & :\equiv \mathsf{Sub}_P\,\Gamma\,\mathsf{Ty} \\ \mathsf{Sub}\,\Delta\,\Gamma & :\equiv \mathsf{Sub}_P\,\Delta\,\Gamma & \mathsf{Tm}\,\Gamma\,A & :\equiv \mathsf{Tm}_P\,\Gamma\,(\mathsf{Tm}[A]) \\ A[\gamma] & :\equiv A \circ_P \gamma & \Pi\,A\,B & :\equiv \Pi \circ_P (A, \mathsf{lam}_P\,B) \end{array}$$

- Yoneda relates the syntax to the P-contextualisation model:

$$\begin{array}{ll} \mathsf{y} : \mathsf{Con}_{\mathsf{I}} \to \mathsf{Con}_P & \mathsf{y} : \mathsf{Ty}_{\mathsf{I}}\,\Gamma \cong \mathsf{Sub}_P\,(\mathsf{y}\,\Gamma)\,\mathsf{Ty} \\ \mathsf{y} : \mathsf{Sub}_{\mathsf{I}}\,\Delta\,\Gamma \cong \mathsf{Sub}_P\,(\mathsf{y}\,\Delta)\,(\mathsf{y}\,\Gamma) & \mathsf{y} : \mathsf{Tm}_{\mathsf{I}}\,\Gamma\,A \cong \mathsf{Tm}_P\,(\mathsf{y}\,\Gamma)\,(\mathsf{Tm}[\mathsf{y}\,A]) \end{array}$$

- Finally we replace Con_P with telescopes

$P :\equiv \mathsf{Psh}(\mathcal{C})$ (presheaves)

- The CwF of presheaves is strict
  - This relies crucially on SProp
  - Needs a trick for Ty (notion of dependent presheaf):

$$\begin{aligned} \mathsf{Ty}\,\Gamma :\equiv\ & (A : (I : \mathcal{C}) \to \Gamma\, I \to \mathsf{U}) \\ & \times (-[-]_A : A\, I\, \gamma_I \to (f : \mathcal{C}(J, I)) \to \gamma_I[f]_\Gamma = \gamma_J \to A\, J\, \gamma_J) \\ & \times \text{functoriality} \end{aligned}$$

- Π is not strict:

$$\begin{aligned} & \Pi\, A\, B\, I\, \gamma_I :\equiv \mathsf{Tm}\,(\mathsf{y}\, I \triangleright A[\mathsf{yl}\,\gamma_I])\,(B[(\mathsf{yl}\,\gamma_I)^{\uparrow}]) \\ & (\Pi\, A\, B)[\gamma]\, I\, \delta_I & \equiv \\ & \Pi\, A\, B\, I\,(\gamma\, \delta_I) & \equiv \\ & \mathsf{Tm}\,(\mathsf{y}\, I \triangleright A[\mathsf{yl}\,(\gamma\, \delta_I)])\,(B[(\mathsf{yl}\,(\gamma\,\delta_I))^{\uparrow}]) & = \\ & \mathsf{Tm}\,(\mathsf{y}\, I \triangleright A[\gamma \circ \mathsf{yl}\,\delta_I])\,(B[(\gamma \circ \mathsf{yl}\,\delta_I)^{\uparrow}]) & \equiv \\ & \Pi\,(A[\gamma])\,(B[\gamma^{\uparrow}])\, I\, \delta_I \end{aligned}$$

- Π with locally representable domain has a similar problem

$P :\equiv \mathsf{Pfa}(\mathcal{C})$ (prefascist sets, Pédrot 2020)

- Prefascist sets come from a right Kan extension:

$$F : \mathsf{Disc}(\mathcal{C}) \to \mathcal{C} \qquad \mathsf{Psh}(\mathcal{C}) \underset{F_*}{\overset{F^*}{\rightleftarrows}} \mathsf{Psh}(\mathsf{Disc}(\mathcal{C})) \quad (\perp)$$

$$\begin{aligned} |\mathsf{Pfa}(\mathcal{C})| :\equiv\ & (\Gamma : \mathcal{C} \to \mathsf{Set}) \\ & \times \big((I : \mathcal{C}) \to ((J : \mathcal{C}) \to \mathcal{C}(J, I) \to \Gamma\, J) \to \mathsf{Prop}\big) \end{aligned}$$

- The CwF of prefascist sets is strict (except $\triangleright\eta : \mathsf{id} = (\mathsf{p}, \mathsf{q})$)
- Π is strict
- Yoneda can be defined

Application

- In Agda, we defined the QIIT of the CwF-syntax of a type theory with Π and Bool with large elimination.
  - All equations are propositional.
  - We call it I.
- We formalised the CwF_Π of prefascist sets,
- defined the $\mathsf{Pfa}(\mathsf{I})$-universe.
- showed that I is isomorphic to the telescopic $\mathsf{Pfa}(\mathsf{I})$-contextualisation of the universe.
- We derived the elimination principle of our new syntax.
- We proved gluing-style canonicity.
  - The canonicity displayed model is substitution-strict.
  - Agda hangs when we try to compute with it.

Related work

- Logical frameworks (Harper–Honsell–Plotkin 1993): work in the internal lanaguage of a presheaf model, figure out how to do proofs in there
  - Same semantics as HOAS (Hofmann 1999), 2-level type theory (Annenkov–Capriotti–Kraus–Sattler 2023)
  - Synthethic Tait Computability (Sterling 2021), internal sconing (Bocquet–K.–Sattler 2023)
  - Talk by Kovács at the WG6 meeting
  - We are staying external, no special metatheory
- Generic strictification via conservativity of equality reflection (Hofmann 1995, Oury 2005, Winterhalter 2019)
- Local universe consruction also provides more definitional equalities (Donkó–K. 2021)
- Other ad-hoc strictification methods:
  - Redefining $-[-]$ recursively (K. TYPES 2023)
  - Shallow embedding (K.–Kovács–Kraus 2019)

Summary

- We analysed Hofmann's semantics of HOAS in an intensional setting
- A method for strictifying substitutions in a model of type theory
- This eliminates the disadvantages of intrinsic syntax when compared to extrinsic syntax:
  - \+ more abstract
  - \+ more definitional equalities
  - – explicit weakenings
- Future work
  - Conjecture: works for any SOGAT (Uemura 2023, K.–Xie 2024)
  - Without SProp?
  - Make it compute
  - Formalise larger type theories