Guarded recursive types: categorical semantics and Pr(ω)

Daniel Gratzer & Adrien Guatto

Friday 23rd May, 2025

Aarhus University & University of Paris Cité

Introduction

Daniel

Adrien

Guarded recursion in types and terms

- An alternative to primitive (co)recursion and general recursion [Nak00]
- Applications to ~~programming with infinite data and to~~ logic and verification

What have we done?

Yesterday, Adrien told us about...

- Functional programming with infinite streams
- A model of stream programming with domain theory
- The topos of trees $\mathbf{Pr}(\omega)$
- The model of an STLC-variant in $\mathbf{Pr}(\omega)$.

Today we focus on $\mathbf{Pr}(\omega)$ more deeply.

What are we doing?

Our goal: to touch on the following topics

- Guarded higher-order logic and its model in $\mathbf{Pr}(\omega)$
- Guarded dependent type theory and its model in $\mathbf{Pr}(\omega)$
- Applications of the above to denotational semantics

What are we doing?

Our goal: to touch on the following topics

- Guarded higher-order logic and its model in $\mathbf{Pr}(\omega)$
- Guarded dependent type theory and its model in $\mathbf{Pr}(\omega)$
- Applications of the above to denotational semantics

Warning

A long march through (categorical) logic and type theory. Ask questions as we go!

First steps in the topos of trees

Recall that $\mathbf{Pr}(\omega) = [\omega^{\mathrm{op}}, \mathbf{Set}]$:

- Objects: ω-indexed collection of sets:

$$X(0) \longleftarrow X(1) \longleftarrow X(2) \longleftarrow \dots$$

- Morphisms: natural transformations:

$$\begin{tikzcd} X(0) \arrow[d] & X(1) \arrow[l] \arrow[d] & X(2) \arrow[l] \arrow[d] & \dots \arrow[l] \arrow[d] \\ Y(0) & Y(1) \arrow[l] & Y(2) \arrow[l] & \dots \arrow[l] \end{tikzcd}$$

First steps in the topos of trees

Recall that $\mathbf{Pr}(\omega) = [\omega^{\mathrm{op}}, \mathbf{Set}]$:

- Objects: ω-indexed collection of sets:

$$X(0) \longleftarrow X(1) \longleftarrow X(2) \longleftarrow \ldots$$

- Morphisms: natural transformations:

$$\begin{tikzcd}
X(0) & X(1) \arrow[l] & X(2) \arrow[l] & \ldots \arrow[l] \\
Y(0) \arrow[from=u] & Y(1) \arrow[l] \arrow[from=u] & Y(2) \arrow[l] \arrow[from=u] & \ldots \arrow[l] \arrow[from=u]
\end{tikzcd}$$

Theorem

$\mathbf{Pr}(\omega)$ *is bicartesian closed.*

Second steps in the topos of trees

Theorem

Pr(ω) *is a Grothendieck topos.*

- All small (co)limits,
- Right adjoints to pullback functors (Π-types)
- A subobject classifier (Prop)
- A hierarchy of “categorical” universes $(\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \dots)$
- . . .

Slogan: **Pr**(ω) behaves like **Set** (except for all the ways it doesn't).

Second steps in the topos of trees

Theorem

$\mathbf{Pr}(\omega)$ *is a Grothendieck topos.*

- All small (co)limits,
- Right adjoints to pullback functors (Π-types)
- A subobject classifier (Prop)
- A hierarchy of "categorical" universes ($\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \ldots$)
- $\ldots$

Slogan: $\mathbf{Pr}(\omega)$ behaves like **Set** (except for all the ways it doesn't).

In particular, $\mathbf{Pr}(\omega)$ doesn't support LEM or choice.

Quick recapitulation of higher-order logic

HOL starts with a simply-typed λ-calculus extended with a new type:

$$
\begin{array}{llll}
\textit{(Types)} & A, B & ::= & \cdots \mid \mathrm{Prop} \\
\textit{(Terms)} & t, u, \phi, \psi & ::= & \cdots \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi \mid \phi = \psi \mid \\
 & & & \forall x : A.\, \phi(x) \mid \exists x : A.\, \phi(x)
\end{array}
$$

We then add an entailment judgment:

$$\Gamma \mid \Theta \vdash \phi$$

where $\Gamma \vdash \phi : \mathrm{Prop}$ and $\Theta = \theta_0, \ldots, \theta_n$ such that $\Gamma \vdash \theta : \mathrm{Prop}$

Quick recapitulation of higher-order logic

HOL starts with a simply-typed λ-calculus extended with a new type:

$$
\begin{array}{llll}
\textit{(Types)} & A, B & ::= & \cdots \mid \mathsf{Prop} \\
\textit{(Terms)} & t, u, \phi, \psi & ::= & \cdots \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi \mid \phi = \psi \mid \\
& & & \forall x : A.\, \phi(x) \mid \exists x : A.\, \phi(x)
\end{array}
$$

We then add an entailment judgment:

$$\Gamma \mid \Theta \vdash \phi$$

where $\Gamma \vdash \phi : \mathsf{Prop}$ and $\Theta = \theta_0, \ldots, \theta_n$ such that $\Gamma \vdash \theta : \mathsf{Prop}$

Quick recapitulation of higher-order logic

HOL starts with a simply-typed λ-calculus extended with a new type:

$$
\begin{array}{llcl}
\textit{(Types)} & A, B & ::= & \cdots \mid \mathrm{Prop} \\
\textit{(Terms)} & t, u, \phi, \psi & ::= & \cdots \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi \mid \phi = \psi \mid \\
 & & & \forall x : A.\, \phi(x) \mid \exists x : A.\, \phi(x)
\end{array}
$$

We then add an entailment judgment:

$$\Gamma \mid \Theta \vdash \phi$$

where $\Gamma \vdash \phi : \mathrm{Prop}$ and $\Theta = \theta_0, \ldots, \theta_n$ such that $\Gamma \vdash \theta : \mathrm{Prop}$

Quick recapitulation of higher-order logic

HOL starts with a simply-typed λ-calculus extended with a new type:

$$
\begin{array}{llll}
\textit{(Types)} & A, B & ::= & \cdots \mid \mathrm{Prop} \\
\textit{(Terms)} & t, u, \phi, \psi & ::= & \cdots \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi \mid \phi = \psi \mid \\
 & & & \forall x : A.\, \phi(x) \mid \exists x : A.\, \phi(x)
\end{array}
$$

We then add an entailment judgment:

$$\Gamma \mid \Theta \vdash \phi$$

where $\Gamma \vdash \phi : \mathrm{Prop}$ and $\Theta = \theta_0, \ldots, \theta_n$ such that $\Gamma \vdash \theta : \mathrm{Prop}$

Rules for entailment

$$\frac{\Gamma \mid \Theta \vdash \phi_0 \qquad \Gamma \mid \Theta \vdash \phi_1}{\Gamma \mid \Theta \vdash \phi_0 \wedge \phi_1} \qquad \frac{\Gamma \mid \Theta \vdash \phi_0 \wedge \phi_1}{\Gamma \mid \Theta \vdash \phi_0 \qquad \Gamma \mid \Theta \vdash \phi_1}$$

$$\frac{\Gamma, x : A \mid \Theta \vdash \phi(x)}{\Gamma \mid \Theta \vdash \forall x : A.\, \phi(x)} \qquad \frac{\Gamma, x : A \mid \Theta, \phi(x) \vdash \psi \qquad \Gamma \mid \Theta \vdash \exists x : A.\, \phi(x)}{\Gamma \mid \Theta \vdash \psi}$$

Interpreting higher-order logic in Set

We interpret HOL into **Set** by extending the normal interpretation of STLC:

$$
\begin{aligned}
&[\![\mathbf{1}]\!] \triangleq \{\star\} \\
&[\![A \times B]\!] \triangleq [\![A]\!] \times [\![B]\!] \\
&[\![A \to B]\!] \triangleq [\![A]\!] \to [\![B]\!] \\
&\ldots \\
&[\![\mathrm{Prop}]\!] \triangleq \mathbf{2}
\end{aligned}
$$

Interpreting higher-order logic in Set

We interpret HOL into **Set** by extending the normal interpretation of STLC:

$$
\begin{aligned}
[\![\mathbf{1}]\!] &\triangleq \{\star\} \\
[\![A \times B]\!] &\triangleq [\![A]\!] \times [\![B]\!] \\
[\![A \to B]\!] &\triangleq [\![A]\!] \to [\![B]\!] \\
&\dots \\
[\![\mathrm{Prop}]\!] &\triangleq \mathbf{2}
\end{aligned}
$$

Apply the Broccoli Methodology for the terms of type Prop:

$$[\![\phi \wedge \psi]\!] \triangleq [\![\phi]\!] \wedge [\![\psi]\!]$$

Fancier phrasing

We interpret *free complete Heyting algebra (cHa)* into the cHa **2**.

Interpreting the entailment relation

How do we interpret $\Gamma \mid \Theta \vdash \phi$?

$$\llbracket \Gamma \mid \Theta \vdash \phi \rrbracket \in \mathbf{2}$$
$$\llbracket \Gamma \mid \Theta \vdash \phi \rrbracket \triangleq \forall \gamma \in \llbracket \Gamma \rrbracket.\ \llbracket \Gamma \vdash \Theta \rrbracket(\gamma) \Rightarrow \llbracket \Gamma \vdash \phi \rrbracket(\gamma)$$

Helpful to write this with the pointwise ordering on $X \to \mathbf{2}$:

$$\llbracket \Gamma \mid \Theta \vdash \phi \rrbracket \triangleq \llbracket \Gamma \vdash \Theta \rrbracket \sqsubseteq \llbracket \Gamma \vdash \phi \rrbracket$$

We then eat our vegetables and check that this definition validates all the rules.

Generalizing the Set interpretation

How can we generalize this recipe?

- We know how to interpret STLC into $\mathbf{Pr}(\omega)$
- What about Prop?

Theorem

If $\mathcal{C}$ is a cartesian closed category and $X : \mathcal{C}$ is a complete Heyting algebra in $\mathcal{C}$, then $\mathcal{C}$ supports a model of higher-order logic.

Generalizing the Set interpretation

How can we generalize this recipe?

- We know how to interpret STLC into $\mathbf{Pr}(\omega)$
- What about Prop?

Theorem

If $\mathcal{C}$ is a cartesian closed category and $X : \mathcal{C}$ is an internal complete Heyting algebra in $\mathcal{C}$, then $\mathcal{C}$ supports a model of higher-order logic.

Need some object $X : \mathbf{Pr}(\omega)$ which supports interpretations of all of the operations:

$$[\![\wedge]\!], [\![\vee]\!], [\![\Rightarrow]\!] : X \times X \to X \qquad [\![\forall_\tau]\!], [\![\exists_\tau]\!] : ([\![\tau]\!] \to X) \to X \qquad \ldots$$

satisfying the expected equations.

Generalizing the Set interpretation

How can we generalize this recipe?

- We know how to interpret STLC into $\mathbf{Pr}(\omega)$
- What about Prop?

Theorem

If $\mathcal{C}$ is a cartesian closed category and $X : \mathcal{C}$ is an internal complete Heyting algebra in $\mathcal{C}$, then $\mathcal{C}$ supports a model of higher-order logic.

Need some object $X : \mathbf{Pr}(\omega)$ which supports interpretations of all of the operations:

$$[\![\wedge]\!], [\![\vee]\!], [\![\Rightarrow]\!] : X \times X \to X \qquad [\![\forall_\tau]\!], [\![\exists_\tau]\!] : ([\![\tau]\!] \to X) \to X \qquad \ldots$$

satisfying the expected equations.

Don't worry: I'll show one equation and then we'll meet a concrete example!

Internal complete Heyting algebras

Suppose X is to be a complete Heyting algebra. We must then have...

1. A morphism $m : X \times X \longrightarrow X$ representing **m**eet
2. A morphism $t : \mathbf{1} \longrightarrow X$ representing **t**op
3. ...
4. The following diagram must commute ($\top \wedge a = a$):

$$\begin{tikzcd} X \times \mathbf{1} \arrow[r, "\pi_1"] \arrow[d, "X \times t"'] & X \arrow[d, "\mathrm{id}"] \\ X \times X \arrow[r, "m"'] & X \end{tikzcd}$$

5. ...

The subobject classifier $\Omega : \mathbf{Pr}(\omega)$

Definition

The subobject classifier $\Omega : \mathbf{Pr}(\omega)$ is defined as follows:

$$\Omega(n) \triangleq \{-1, \ldots, n\}$$
$$\Omega(n \leq m) : \Omega(m) \to \Omega(n)$$
$$\Omega(n \leq m) \triangleq \min(n, -)$$

(Exercise: check this is indeed a functor)

Temporal intuition for Ω

Temporal intuition for guarded recursion

$x \in X(n) \sim x$ is the results of computing a value of X for n steps

Elements of Ω are observations on truth values:

$$i \in \Omega(n) \sim \text{the truth value which holds for the first } i \text{ steps}$$

- $n \in \Omega(n)$ is true (so far).
- $-1 \in \Omega(n)$ is false right away.
- Other values interpolate between these extremes.

A check for intuition

Lemma

$\mathrm{Hom}(\mathbf{1}, \Omega) \cong \{-1, \infty\} \cup \omega$

Proof.

To the board. $\square$

Categorical intuition for Ω

We can also give Ω a nice universal property:

Lemma (Mac Lane and Moerdijk [MM92])

Ω is the subobject classifier: there is a natural isomorphism $\mathrm{Hom}(X, \Omega) \cong \mathbf{Sub}(X)$

- Maps $\mathrm{Hom}(X, \Omega)$ determine subobjects $A \hookrightarrow X$ (up to iso) of the domain
- Yields an ordering $\sqsubseteq$ on $\mathrm{Hom}(X, \Omega)$

Categorical intuition for Ω

We can also give Ω a nice universal property:

Lemma (Mac Lane and Moerdijk [MM92])

Ω is the subobject classifier: there is a natural isomorphism $\mathrm{Hom}(X, \Omega) \cong \mathbf{Sub}(X)$

- Maps $\mathrm{Hom}(X, \Omega)$ determine subobjects $A \hookrightarrow X$ (up to iso) of the domain
- Yields an ordering $\sqsubseteq$ on $\mathrm{Hom}(X, \Omega)$
- Compare with **Set** and **2**; a map $X \to \mathbf{2}$ characterizes a subset.
- Everything which follows could be done just using this universal property.

Theorem

Ω is an internal complete Heyting algebra.

Proof.

$$\top : \Omega \qquad \bot : \Omega$$
$$\top_n = n \qquad \bot_n = -1$$

$$\wedge : \mathrm{Hom}(\Omega \times \Omega, \Omega) \qquad \vee : \mathrm{Hom}(\Omega \times \Omega, \Omega)$$
$$i \wedge_n j = \min(i, j) \qquad i \vee_n j = \max(i, j)$$

$$\forall^X : \mathrm{Hom}(\Omega^X, \Omega) \qquad \exists^X : \mathrm{Hom}(\Omega^X, \Omega)$$
$$\forall^X_n \alpha = \forall m \leq n, x \in X(m).\ \alpha(x) \qquad \exists^X_n \alpha = \exists x \in X(n).\ \alpha(x)$$

Must verify that these satisfy the expected equations. □

An aside: the Kripke–Joyal semantics

Lemma

$\phi \in \Omega(n)$ "is" a monotone predicate on $\{0, \dots, n\}$. Elements of $\mathrm{Hom}(1, \Omega)$ "are" a monotone predicates on ω.

If $\Gamma \vdash t : \mathrm{Prop}$, $[\![t]\!] : [\![\Gamma]\!] \to \Omega$ which induces a predicate on $\sum_{n:\omega} [\![\Gamma]\!](n)$:

$$n \vDash [\![\Gamma \vdash t : \mathrm{Prop}]\!]_\gamma \triangleq [\![\Gamma \vdash t : \mathrm{Prop}]\!]_n(\gamma) = n$$

Chasing this through, we obtain the (familiar?) Kripke semantics over ω:

$$n \vDash [\![\Gamma \vdash \phi \Rightarrow \psi]\!]_\gamma = \forall m \leq n.\ m \vDash [\![\Gamma \vdash \phi : \mathrm{Prop}]\!]_\gamma \Rightarrow m \vDash [\![\Gamma \vdash \psi : \mathrm{Prop}]\!]_\gamma$$

An aside: the Kripke–Joyal semantics

Lemma

$\phi \in \Omega(n)$ "is" a monotone predicate on $\{0, \ldots, n\}$. Elements of $\mathrm{Hom}(1, \Omega)$ "are" a monotone predicates on ω.

If $\Gamma \vdash t : \mathsf{Prop}$, $[\![t]\!] : [\![\Gamma]\!] \to \Omega$ which induces a predicate on $\sum_{n:\omega} [\![\Gamma]\!](n)$:

$$n \vDash [\![\Gamma \vdash t : \mathsf{Prop}]\!]_\gamma \triangleq [\![\Gamma \vdash t : \mathsf{Prop}]\!]_n(\gamma) = n$$

Chasing this through, we obtain the (familiar?) Kripke semantics over ω:

$$n \vDash [\![\Gamma \vdash \phi \Rightarrow \psi]\!]_\gamma = \forall m \leq n.\ m \vDash [\![\Gamma \vdash \phi : \mathsf{Prop}]\!]_\gamma \Rightarrow m \vDash [\![\Gamma \vdash \psi : \mathsf{Prop}]\!]_\gamma$$

(Kripke semantics arise as interpretations in $\mathbf{Pr}(\mathcal{C})$; Beth semantics from $\mathbf{Sh}(\mathcal{C})$.)

A further aside: $\mathbf{Pr}(\omega)$ refutes LEM

We can now show that this model in $\mathbf{Pr}(\omega)$ refutes LEM. That is:

Lemma

$[\![\phi : \mathrm{Prop} \mid \top \vdash \phi \vee (\phi \Rightarrow \bot)]\!]$ *does not hold.*

Proof.

To the board. $\square$

Exercise for the curious: show that $\forall \phi.\ \neg\phi \vee \neg\neg\phi$ does hold in this model.[1]

[1] Question: does anyone have a name for this?

Taking stock

For those who slept through the first n minutes, what has happened?

- Thus far we have shown how to interpret HOL into $\mathbf{Pr}(\omega)$.
- However, nothing thus far is truly *guarded* about this.
- Next: we extend HOL with some guarded-recursion specific connectives.

Part of a more general pattern: we're after the internal logic of $\mathbf{Pr}(\omega)$.

Guarded higher-order logic

We return to the previous definition of higher-order logic:

$$
\begin{array}{llll}
\textit{(Types)} & A, B & ::= & \cdots \mid \mathrm{Prop} \\
\textit{(Terms)} & t, u, \phi, \psi & ::= & \cdots \mid \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi \mid \phi = \psi \mid \\
 & & & \forall x : A.\, \phi(x) \mid \exists x : A.\, \phi(x) \mid \blacktriangleright \phi
\end{array}
$$

$$
\frac{\Gamma \mid \Theta \vdash \phi}{\Gamma \mid \Theta \vdash \blacktriangleright\phi}
\qquad
\frac{\Gamma \mid \Theta, \blacktriangleright\phi \vdash \phi}{\Gamma \mid \Theta \vdash \phi}
\qquad
\frac{\Gamma \mid \Theta \vdash \blacktriangleright\phi \wedge \blacktriangleright\psi}{\Gamma \mid \Theta \vdash \blacktriangleright(\phi \wedge \psi)}
\qquad
\frac{\Gamma \mid \Theta \vdash \blacktriangleright(\phi \wedge \psi)}{\Gamma \mid \Theta \vdash \blacktriangleright\phi \wedge \blacktriangleright\psi}
$$

Extending the interpretation III

We must now

1. Choose $\blacktriangleright : \mathrm{Hom}(\Omega, \Omega)$ such that $[\![\blacktriangleright\phi]\!] = \blacktriangleright \circ [\![\phi]\!]$
2. Show that $\blacktriangleright$ satisfies the expected inference rules.

Let's begin with the definition:

$$\blacktriangleright : \mathrm{Hom}(\Omega, \Omega)$$
$$\blacktriangleright_n\, i = \min(i + 1, n)$$

(Remember: i is "true for the first i ticks" so $\blacktriangleright_n i$ is then true for $i + 1$ ticks)

Extending the interpretation II

Lemma

If $[\![\Gamma \mid \Theta \vdash \phi]\!]$ *then* $[\![\Gamma \mid \Theta \vdash \blacktriangleright\phi]\!]$.

- Suffices to show that $\mathrm{id} \sqsubseteq \blacktriangleright \in \mathrm{Hom}(\Omega, \Omega)$
- Fix $n : \omega$ and $i \in \Omega(n)$; suffices to show $i \leq \min(n, i+1)$.

Lemma

$[\![\Gamma \mid \Theta \vdash \blacktriangleright(\phi \wedge \psi)]\!] = [\![\Gamma \mid \Theta \vdash \blacktriangleright\phi \wedge \blacktriangleright\psi]\!]$.

- Suffices to show that $\blacktriangleright \circ \wedge = \wedge \circ (\blacktriangleright, \blacktriangleright) \in \mathrm{Hom}(\Omega \times \Omega, \Omega)$
- Fix $n : \omega$ and $i, j \in \Omega(n)$; suffices to show $\min(n, 1 + \min(i, j)) = \min(\min(n, i+1), \min(n, j+1))$.

Extending the interpretation I

Lemma

If $[\![\Gamma \mid \Theta, \blacktriangleright\phi \vdash \phi]\!]$ *then* $[\![\Gamma \mid \Theta \vdash \phi]\!]$.

- Fix n and $\gamma \in [\![\Gamma]\!](n)$. Want to show $[\![\Theta]\!](n, \gamma) \leq [\![\phi]\!](n, \gamma)$.
- By assumption, we know $[\![\Theta, \blacktriangleright\phi]\!](n, \gamma) \leq [\![\Theta, \phi]\!](n, \gamma)$
- Iterating n times with other rules for $\blacktriangleright$ we get $[\![\Theta, \blacktriangleright^n\phi]\!](n, \gamma) \leq [\![\Theta, \phi]\!](n, \gamma)$

$$\begin{aligned}
[\![\Theta, \blacktriangleright^n\phi]\!](n, \gamma) &= \min([\![\Theta]\!](n, \gamma), [\![\blacktriangleright^n\phi]\!](n, \gamma)) \\
&= \min([\![\Theta]\!](n, \gamma), \min(n, n + [\![\phi]\!](n, \gamma))) \\
&= \min([\![\Theta]\!](n, \gamma), n) \\
&= [\![\Theta]\!](n, \gamma)
\end{aligned}$$

Extending the interpretation I

Lemma

If $[\![\Gamma \mid \Theta, \blacktriangleright\phi \vdash \phi]\!]$ *then* $[\![\Gamma \mid \Theta \vdash \phi]\!]$.

- Fix n and $\gamma \in [\![\Gamma]\!](n)$. Want to show $[\![\Theta]\!](n,\gamma) \leq [\![\phi]\!](n,\gamma)$.
- By assumption, we know $[\![\Theta, \blacktriangleright\phi]\!](n,\gamma) \leq [\![\Theta, \phi]\!](n,\gamma)$
- Iterating n times with other rules for $\blacktriangleright$ we get $[\![\Theta, \blacktriangleright^n\phi]\!](n,\gamma) \leq [\![\Theta, \phi]\!](n,\gamma)$

$$\begin{aligned}
[\![\Theta, \blacktriangleright^n\phi]\!](n,\gamma) &= \min([\![\Theta]\!](n,\gamma), [\![\blacktriangleright^n\phi]\!](n,\gamma)) \\
&= \min([\![\Theta]\!](n,\gamma), \min(n, n + [\![\phi]\!](n,\gamma))) \\
&= \min([\![\Theta]\!](n,\gamma), n) \\
&= [\![\Theta]\!](n,\gamma)
\end{aligned}$$

(Secretly same idea as Adrien's proof of rec; just cooler)

Other structure in $\mathbf{Pr}(\omega)$

Q How do get guarded recursive propositions?

A1 Could add bespoke fixed-point operator $\mu x.\ \phi(x)$

Other structure in $\mathbf{Pr}(\omega)$

Q How do get guarded recursive propositions?

A1 Could add bespoke fixed-point operator $\mu x.\ \phi(x)$

A2 Reuse ordinary term-level guarded recursion $\mathsf{rec}_A : (\triangleright A \to A) \to A$.

$$\textit{(Types)} \quad A, B \quad ::= \quad \cdots \mid \mathsf{Prop} \mid \ \triangleright A$$

Now define $\mu \triangleq \mathsf{rec}_{\mathsf{Prop}} : (\triangleright \mathsf{Prop} \to \mathsf{Prop}) \to \mathsf{Prop}$... but what use is $\triangleright \mathsf{Prop}$?

Other structure in $\mathbf{Pr}(\omega)$

Q How do get guarded recursive propositions?

A1 Could add bespoke fixed-point operator $\mu x.\ \phi(x)$

A2 Reuse ordinary term-level guarded recursion $\mathsf{rec}_A : (\triangleright A \to A) \to A$.

$$\textit{(Types)} \quad A, B \quad ::= \quad \cdots \mid \mathsf{Prop} \mid \ \triangleright A$$

Now define $\mu \triangleq \mathsf{rec}_{\mathsf{Prop}} : (\triangleright \mathsf{Prop} \to \mathsf{Prop}) \to \mathsf{Prop}$... but what use is $\triangleright\mathsf{Prop}$?

- Need to add an operator $\widehat{\blacktriangleright}$ to use the IH

$$\frac{\Gamma \vdash \phi : \triangleright \mathsf{Prop}}{\Gamma \vdash \widehat{\blacktriangleright}\phi : \mathsf{Prop}}$$

- Add an equation $\widehat{\blacktriangleright}(\mathsf{delay}(\phi)) = \blacktriangleright\phi$ to connect this with existing $\blacktriangleright$.

Exercise: prove $\mu x. \bot \vee \widehat{\blacktriangleright} x$.

What is $\widehat{\blacktriangleright}$?

Let's show how to define $\widehat{\blacktriangleright} : \triangleright\Omega \to \Omega$ in $\mathbf{Pr}(\omega)$.

$$\widehat{\blacktriangleright}_n\, i = i + 1$$

Calculation now shows $\widehat{\blacktriangleright} \circ [\![\mathrm{delay}]\!] = \blacktriangleright$

A quick aside: applications

Why is guarded higher-order logic useful?

- One compelling application of gHOL: step-indexed logical relation.
- We can use $\blacktriangleright$/step-indexing to handle μ, **ref**, etc. [AM01; Ahm04; App+07]
- Using $\blacktriangleright$, we can hide the step-indices from the user [App+07; DAB11]
- There's a whole cottage industry of these applications!

We're soon on to more theory, but perhaps one slide of examples.

Step-indexed logical relations

Logical relations in a hurry:

- a LR inductively assigns a predicate/relation to every type
- Here, the relation $R_\tau(v_0, v_1)$ tells us when v_0 **r**efines v_1.
- Describe the relation on values $\mathcal{R}^{\mathsf{val}}$ then uniformly extend to expressions $\mathcal{R}^{\mathsf{expr}}$.

$$
\begin{aligned}
&\mathcal{R}^{\mathsf{val}}_\tau : \mathsf{Val}(\tau) \times \mathsf{Val}(\tau) \to \mathsf{Prop}\\
&\mathcal{R}^{\mathsf{val}}_{\mathbf{1}}(v_0, v_1) = v_0 = \star \wedge v_1 = \star\\
&\mathcal{R}^{\mathsf{val}}_{\tau \to \sigma}(f, g) = \forall v_0, v_1 : \mathsf{Val}(\tau).\ R_\tau(v_0, v_1) \Rightarrow R^{\mathsf{expr}}_\sigma(f(v_0), g(v_1))\\
&\ldots
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{R}^{\mathrm{expr}}_\tau : \mathsf{Expr}(\tau) \times \mathsf{Expr}(\tau) \to \mathsf{Prop}\\
&\mathcal{R}^{\mathrm{expr}}_\tau(e_0, e_1) = \forall v_0.\ e_0 \mapsto^* v_0 \Rightarrow \exists v_1.\ e_1 \mapsto^* v_1 \wedge \mathcal{R}^{\mathsf{val}}_\tau(v_0, v_1)
\end{aligned}
$$

μ in logical relations

- $\mu\alpha.\tau(\alpha)$ is a headache... want $\mathcal{R}_{\tau[\mu\alpha.\tau(\alpha)/\alpha]}$ but have only $\mathcal{R}_\tau$.
- Solution: replace "inductive" with "guarded recursive" [AM01]

$$\mathcal{R}^{\mathsf{val}}_{\tau} : \mathsf{Val}(\tau) \times \mathsf{Val}(\tau) \to \mathsf{Prop}$$
$$\mathcal{R}^{\mathsf{val}}_{\mu\alpha.\tau(\alpha)}(v_0, v_1) = \widehat{\blacktriangleright}\mathcal{R}^{\mathrm{expr}}_{\tau[\mu\alpha.\tau(\alpha)/\alpha]}(\mathsf{unfold}(v_0), \mathsf{unfold}(v_1))$$

$$\mathcal{R}^{\mathrm{expr}}_{\tau} : \mathsf{Expr}(\tau) \times \mathsf{Expr}(\tau) \to \mathsf{Prop}$$
$$\mathcal{R}^{\mathrm{expr}}_{\tau}(e_0, e_1) = \forall v_0, n.\ e_0 \mapsto^{n} v_0 \Rightarrow \exists v_1.\ e_1 \mapsto^{*} v_1 \wedge \blacktriangleright^{n}\mathcal{R}^{\mathsf{val}}_{\tau}(v_0, v_1)$$

μ in logical relations

- $\mu\alpha.\tau(\alpha)$ is a headache... want $\mathcal{R}_{\tau[\mu\alpha.\tau(\alpha)/\alpha]}$ but have only $\mathcal{R}_\tau$.
- Solution: replace "inductive" with "guarded recursive" [AM01]

$$\mathcal{R}^{\mathsf{val}}_\tau : \mathsf{Val}(\tau) \times \mathsf{Val}(\tau) \to \mathsf{Prop}$$
$$\mathcal{R}^{\mathsf{val}}_{\mu\alpha.\tau(\alpha)}(v_0, v_1) = \hat{\blacktriangleright}\mathcal{R}^{\mathrm{expr}}_{\tau[\mu\alpha.\tau(\alpha)/\alpha]}(\mathsf{unfold}(v_0), \mathsf{unfold}(v_1))$$

$$\mathcal{R}^{\mathrm{expr}}_\tau : \mathsf{Expr}(\tau) \times \mathsf{Expr}(\tau) \to \mathsf{Prop}$$
$$\mathcal{R}^{\mathrm{expr}}_\tau(e_0, e_1) = \forall v_0, n.\ e_0 \mapsto^n v_0 \Rightarrow \exists v_1.\ e_1 \mapsto^* v_1 \wedge \blacktriangleright^n \mathcal{R}^{\mathsf{val}}_\tau(v_0, v_1)$$

Basis of ongoing work e.g., Iris [Jun+18] (`https://iris-project.org`).

From logic to type theory

From higher-order logic to dependent types

Our next goal: switch from HOL to Martin-Löf type theory.

- We've met type theory before this week, but a brief reminder!
- Logic no longer confined to Prop; propositions and ordinary types intermingle.
- Formally, we shall treat type theory as consisting of 4 judgments:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

(each with a corresponding notion of equality)

From higher-order logic to dependent types

Our next goal: switch from HOL to Martin-Löf type theory.

- We've met type theory before this week, but a brief reminder!
- Logic no longer confined to Prop; propositions and ordinary types intermingle.
- Formally, we shall treat type theory as consisting of 4 judgments:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

(each with a corresponding notion of equality)

From higher-order logic to dependent types

Our next goal: switch from HOL to Martin-Löf type theory.

- We've met type theory before this week, but a brief reminder!
- Logic no longer confined to Prop; propositions and ordinary types intermingle.
- Formally, we shall treat type theory as consisting of 4 judgments:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

  (each with a corresponding notion of equality)

From higher-order logic to dependent types

Our next goal: switch from HOL to Martin-Löf type theory.

- We've met type theory before this week, but a brief reminder!
- Logic no longer confined to Prop; propositions and ordinary types intermingle.
- Formally, we shall treat type theory as consisting of 4 judgments:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

(each with a corresponding notion of equality)

From higher-order logic to dependent types

Our next goal: switch from HOL to Martin-Löf type theory.

- We've met type theory before this week, but a brief reminder!
- Logic no longer confined to Prop; propositions and ordinary types intermingle.
- Formally, we shall treat type theory as consisting of 4 judgments:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

(each with a corresponding notion of equality)

From higher-order logic to dependent types

Our next goal: switch from HOL to Martin-Löf type theory.

- We've met type theory before this week, but a brief reminder!
- Logic no longer confined to Prop; propositions and ordinary types intermingle.
- Formally, we shall treat type theory as consisting of 4 judgments:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

  (each with a corresponding notion of equality)

My bias: type theory is a (generalized) algebraic theory; makes models easier

A few basic rules

$$\frac{}{\vdash \mathbf{1}} \qquad \frac{\vdash \Gamma \qquad \Gamma \vdash A}{\vdash \Gamma.A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A}{\Delta \vdash A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A}{\Delta \vdash a[\gamma] : A[\gamma]}$$

$$\frac{\vdash \Gamma \qquad \Gamma \vdash A}{\Gamma.A \vdash \uparrow : \Gamma} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Delta \vdash a : A[\gamma]}{\Delta \vdash \gamma.a : \Gamma.A} \qquad \frac{\vdash \Gamma \qquad \Gamma \vdash A}{\Gamma.A \vdash \mathbf{v} : A[\uparrow]}$$

- Contexts are formal syntactic objects: not just lists!
- Substitution application is a term/type former; variables are nameless (de Bruijn)

A few basic rules

$$\frac{}{\vdash \mathbf{1}} \qquad \frac{\vdash \Gamma \qquad \Gamma \vdash A}{\vdash \Gamma.A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A}{\Delta \vdash A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A}{\Delta \vdash a[\gamma] : A[\gamma]}$$

$$\frac{\vdash \Gamma \qquad \Gamma \vdash A}{\Gamma.A \vdash \uparrow : \Gamma} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Delta \vdash a : A[\gamma]}{\Delta \vdash \gamma.a : \Gamma.A} \qquad \frac{\vdash \Gamma \qquad \Gamma \vdash A}{\Gamma.A \vdash \mathbf{v} : A[\uparrow]}$$

- Contexts are formal syntactic objects: not just lists!
- Substitution application is a term/type former; variables are nameless (de Bruijn)
- Self-promotion: "Principles of Dependent Type Theory" by Angiuli and G.

All the type formers

Unfortunately, it's not practical to try and introduce all of MLTT

- Let me instead rely on intuition from Rocq
- We have dependent functions, dependent pairs, disjoint unions, universes...

All the type formers

Unfortunately, it's not practical to try and introduce all of MLTT

- Let me instead rely on intuition from Rocq
- We have dependent functions, dependent pairs, disjoint unions, universes...

All of these can be (laboriously) specified by rules like before:

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathbf{1}} \qquad \frac{\Delta \vdash \gamma : \Gamma}{\Gamma \vdash \mathbf{1}[\gamma] = \mathbf{1}} \qquad \frac{\vdash \Gamma}{\Gamma \vdash \star : \mathbf{1}} \qquad \frac{\vdash \Gamma \qquad \Gamma \vdash u : \mathbf{1}}{\Gamma \vdash u = \star : \mathbf{1}}$$

Want to see all the rules? Angiuli and G.'s Appendix A (9 pages).

Scaling up Adrien's calculus from yesterday

Our goal today: not MLTT, but guarded MLTT.

Q How do we add a modality like $\triangleright$ to MLTT?

A Carefully!

Scaling up Adrien's calculus from yesterday

Our goal today: not MLTT, but guarded MLTT.

Q How do we add a modality like $\triangleright$ to MLTT?

A Carefully!

- Must account for substitutions, new equations, etc., etc.
- In particular, the well-formedness of $\triangleright A$ is much more complex! (More akin to $\widehat{\blacktriangleright}$).

Modal dependent type theory

Lots of approaches to modal type theory; today, we follow Birkedal et al. [Bir+20].

- Same idea as Adrien's calculus: $\triangleleft \dashv \triangleright$ realized by 🔒.
- More syntactically naïve (no 🔓); we'll discuss problems at the end.
- Main advantage: the rules of this calculus are very short.

The calculus of dependent right adjoints

$$\frac{\vdash \Gamma}{\vdash \Gamma.\text{🔒}} \qquad \frac{\Delta \vdash \gamma : \Gamma}{\Delta.\text{🔒} \vdash \gamma.\text{🔒} : \Gamma.\text{🔒}}$$

$$\frac{\Gamma.\text{🔒} \vdash A}{\Gamma \vdash \triangleright A} \qquad \frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma \vdash \mathsf{guard}(a) : \triangleright A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma.\text{🔒} \vdash \mathsf{open}(a) : A}$$

$$\frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma.\text{🔒} \vdash \mathsf{open}(\mathsf{guard}(a)) = a : A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma \vdash \mathsf{guard}(\mathsf{open}(a)) = a : \triangleright A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma.\text{🔒} \vdash a : A}{\Delta.\text{🔒} \vdash \mathsf{open}(a)[\gamma.\text{🔒}] = \mathsf{open}(a[\gamma]) : A} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash a : \triangleright A}{\Delta \vdash \mathsf{guard}(a)[\gamma] = \mathsf{guard}(a[\gamma.\text{🔒}]) : \triangleright(A[\gamma.\text{🔒}])}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma.\lock} \qquad \frac{\Delta \vdash \gamma : \Gamma}{\Delta.\lock \vdash \gamma.\lock : \Gamma.\lock}$$

$$\frac{\Gamma.\lock \vdash A}{\Gamma \vdash \triangleright A} \qquad \frac{\Gamma.\lock \vdash a : A}{\Gamma \vdash \mathrm{guard}(a) : \triangleright A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma.\lock \vdash \mathrm{open}(a) : A}$$

$$\frac{\Gamma.\lock \vdash a : A}{\Gamma.\lock \vdash \mathrm{open}(\mathrm{guard}(a)) = a : A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma \vdash \mathrm{guard}(\mathrm{open}(a)) = a : \triangleright A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma.\lock \vdash a : A}{\Delta.\lock \vdash \mathrm{open}(a)[\gamma.\lock] = \mathrm{open}(a[\gamma]) : A} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash a : \triangleright A}{\Delta \vdash \mathrm{guard}(a)[\gamma] = \mathrm{guard}(a[\gamma.\lock]) : \triangleright(A[\gamma.\lock])}$$

The calculus of dependent right adjoints

$$\frac{\vdash \Gamma}{\vdash \Gamma.\text{🔒}} \qquad \frac{\Delta \vdash \gamma : \Gamma}{\Delta.\text{🔒} \vdash \gamma.\text{🔒} : \Gamma.\text{🔒}}$$

$$\frac{\Gamma.\text{🔒} \vdash A}{\Gamma \vdash \triangleright A} \qquad \frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma \vdash \mathsf{guard}(a) : \triangleright A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma.\text{🔒} \vdash \mathsf{open}(a) : A}$$

$$\frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma.\text{🔒} \vdash \mathsf{open}(\mathsf{guard}(a)) = a : A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma \vdash \mathsf{guard}(\mathsf{open}(a)) = a : \triangleright A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma.\text{🔒} \vdash a : A}{\Delta.\text{🔒} \vdash \mathsf{open}(a)[\gamma.\text{🔒}] = \mathsf{open}(a[\gamma]) : A} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash a : \triangleright A}{\Delta \vdash \mathsf{guard}(a)[\gamma] = \mathsf{guard}(a[\gamma.\text{🔒}]) : \triangleright(A[\gamma.\text{🔒}])}$$

The calculus of dependent right adjoints

$$\frac{\vdash \Gamma}{\vdash \Gamma.\text{🔒}} \qquad \frac{\Delta \vdash \gamma : \Gamma}{\Delta.\text{🔒} \vdash \gamma.\text{🔒} : \Gamma.\text{🔒}}$$

$$\frac{\Gamma.\text{🔒} \vdash A}{\Gamma \vdash \triangleright A} \qquad \frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma \vdash \mathsf{guard}(a) : \triangleright A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma.\text{🔒} \vdash \mathsf{open}(a) : A}$$

$$\frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma.\text{🔒} \vdash \mathsf{open}(\mathsf{guard}(a)) = a : A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma \vdash \mathsf{guard}(\mathsf{open}(a)) = a : \triangleright A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma.\text{🔒} \vdash a : A}{\Delta.\text{🔒} \vdash \mathsf{open}(a)[\gamma.\text{🔒}] = \mathsf{open}(a[\gamma]) : A} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash a : \triangleright A}{\Delta \vdash \mathsf{guard}(a)[\gamma] = \mathsf{guard}(a[\gamma.\text{🔒}]) : \triangleright(A[\gamma.\text{🔒}])}$$

The calculus of dependent right adjoints

$$\frac{\vdash \Gamma}{\vdash \Gamma.\text{🔒}} \qquad \frac{\Delta \vdash \gamma : \Gamma}{\Delta.\text{🔒} \vdash \gamma.\text{🔒} : \Gamma.\text{🔒}}$$

$$\frac{\Gamma.\text{🔒} \vdash A}{\Gamma \vdash \triangleright A} \qquad \frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma \vdash \mathrm{guard}(a) : \triangleright A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma.\text{🔒} \vdash \mathrm{open}(a) : A}$$

$$\frac{\Gamma.\text{🔒} \vdash a : A}{\Gamma.\text{🔒} \vdash \mathrm{open}(\mathrm{guard}(a)) = a : A} \qquad \frac{\Gamma \vdash a : \triangleright A}{\Gamma \vdash \mathrm{guard}(\mathrm{open}(a)) = a : \triangleright A}$$

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma.\text{🔒} \vdash a : A}{\Delta.\text{🔒} \vdash \mathrm{open}(a)[\gamma.\text{🔒}] = \mathrm{open}(a[\gamma]) : A} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash a : \triangleright A}{\Delta \vdash \mathrm{guard}(a)[\gamma] = \mathrm{guard}(a[\gamma.\text{🔒}]) : \triangleright(A[\gamma.\text{🔒}])}$$

Guarded dependent type theory

We must add a few operations which specialize things to guarded recursion.

$$\frac{\vdash \Gamma}{\Gamma.\text{🔒} \vdash \mathsf{adv} : \Gamma}$$

Allows us to define $\mathsf{delay}(a) \triangleq \mathsf{guard}(a[\mathsf{adv}])$.

Guarded dependent type theory

We must add a few operations which specialize things to guarded recursion.

$$\frac{\vdash \Gamma}{\Gamma.\text{🔒} \vdash \mathsf{adv} : \Gamma}$$

Allows us to define $\mathsf{delay}(a) \triangleq \mathsf{guard}(a[\mathsf{adv}])$.

Now, rec (n.b. complicated by the need to balance substitutions).

$$\frac{\Gamma \vdash A \qquad \Gamma. \rhd A[\mathsf{adv}] \vdash a : A[\uparrow]}{\Gamma \vdash \mathsf{rec}(a) : A} \qquad \frac{\Gamma. \rhd A[\mathsf{adv}] \vdash a : A[\uparrow]}{\Gamma \vdash \mathsf{rec}(a) = a[\mathsf{id}.\mathsf{delay}(\mathsf{rec}(a))] : A}$$

Guarded dependent type theory

We must add a few operations which specialize things to guarded recursion.

$$\frac{\vdash \Gamma}{\Gamma.\text{🔒} \vdash \mathsf{adv} : \Gamma}$$

Allows us to define $\mathsf{delay}(a) \triangleq \mathsf{guard}(a[\mathsf{adv}])$.

Now, rec (n.b. complicated by the need to balance substitutions).

$$\frac{\Gamma \vdash A \qquad \Gamma.\triangleright A[\mathsf{adv}] \vdash a : A[\uparrow]}{\Gamma \vdash \mathsf{rec}(a) : A} \qquad \frac{\Gamma.\triangleright A[\mathsf{adv}] \vdash a : A[\uparrow]}{\Gamma \vdash \mathsf{rec}(a) = a[\mathsf{id}.\mathsf{delay}(\mathsf{rec}(a))] : A}$$

Other approaches to rec

If this seems ad hoc, that's because it is! Many different attempts to integrate $\triangleright$

- Extensional guarded type theory [Bir+12]
- Guarded type theory with clocks [Møg14]
- Clocked type theory [BGM17]
- Guarded cubical type theory [Bir+19]
- Clocked cubical type theory [KMV22]
- Stratified guarded type theory [GB22]
- Gatsby [Gra25]

Why all of this churn?

Fundamentally, we want to satisfy 4 goals simultaneously:

1. Include $\triangleright$ (and companion modalities)
2. Include rec with a propositional equation for unfolding
3. Closed elements of type $\mathbb{N}$ are convertible with actual numerals (*canonicity*)
4. Decidable type-checking (via decidable definitional equality)

Two possible type theories satisfy all 4 goals (CCloTT, Gatsby), but only conjecturally.

Why guarded dependent types?

Dependence has remarkable interactions with guarded recursion.

1. We can interpret recursive programs using the guarded delay monad [PMB15]:

$$L(A) = A + \triangleright(L(A))$$

   Recursive types [Pav16], non-determinism [BBM14], π-calculus [VV20] higher-order store [SGB23], etc.

2. Even better, we can use $\triangleright + \mathcal{U}$ to solve domain equations (as seen in Iris):

$$\mathsf{Stream}\, A = \mathsf{rec}(S.A \times \triangleright(\mathsf{open}(S)))$$
$$D^{\mathrm{g}}_{\infty} = \mathsf{rec}(S. \triangleright (\mathsf{open}(S) \to \mathsf{open}(S)))$$

`https://www.jonmsterling.com/jms-005S/index.xml`

Modeling (guarded) type theory

How do we interpret dependent type theory?

Our goal for the rest of the lecture: interpret this calculus into $\mathbf{Pr}(\omega)$.

We therefore need to...

- Work out how to model plain type theory in a category (categories with families)
- Show how to extend this to our calculus (CwFs + structure)
- Actual carry out this procedure for $\mathbf{Pr}(\omega)$ (a small instance of coherence)

How do we interpret dependent type theory?

Our goal for the rest of the lecture: interpret this calculus into $\mathbf{Pr}(\omega)$.

We therefore need to...

- Work out how to model plain type theory in a category (categories with families)
- Show how to extend this to our calculus (CwFs + structure)
- Actual carry out this procedure for $\mathbf{Pr}(\omega)$ (a small instance of coherence)

A complete example of working in categorical type theory [Bir+12; BM13].

Models of type theory

The motivation for algebraic approaches to type theory: it makes models easy.

A model a generalized algebraic theory [Car78] consists of...

- To every judgment, an indexed family of sets.
- To every operation, a function between to the these sets.
- To every equation, a proof that the corresponding functions agree.

Theorem

There is a category of models of every GAT; syntax is initial in this category.

Models of type theory

The motivation for algebraic approaches to type theory: it makes models easy.

A model a generalized algebraic theory [Car78] consists of...

- To every judgment, an indexed family of sets.
- To every operation, a function between to the these sets.
- To every equation, a proof that the corresponding functions agree.

Theorem

There is a category of models of every GAT; syntax is initial in this category.

Models of type theory

The motivation for algebraic approaches to type theory: it makes models easy.

A model a generalized algebraic theory [Car78] consists of...

- To every judgment, an indexed family of sets.
- To every operation, a function between to the these sets.
- To every equation, a proof that the corresponding functions agree.

Theorem

There is a category of models of every GAT; syntax is initial in this category.

Really, this is the domain of *logical frameworks*. I picked the "simplest" one.

Models of type theory II

Type theory:

$$\vdash \Gamma \qquad \Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Gamma \vdash a : A$$

If we unfold this, a model $\mathcal{M}$ of type theory requires...

1. A set Cx of contexts
2. For every pair of contexts $\Delta, \Gamma \in \mathsf{Cx}$, a set of substitutions $\mathsf{Sb}(\Delta, \Gamma)$.
3. For every $\Gamma \in \mathsf{Cx}$, a set of types $\mathsf{Ty}(\Gamma)$.
4. For every $\Gamma \in \mathsf{Cx}$ and $A \in \mathsf{Ty}(A)$, a set of terms $\mathsf{Tm}(\Gamma, A)$.
5. All of the operations and equations...

But where have the categories gone?

Categories with families

Theorem (Categories with families (CwFs) [Dyb96])

A model of type theory consists of

1. *A category $\mathcal{C}$ of contexts and substitutions.*
2. *A presheaf* $\mathrm{Ty} : \mathbf{Pr}(\mathcal{C})$ *of types.*
3. *A presheaf* $\mathrm{Tm} : \mathbf{Pr}(\int \mathrm{Ty})$ *of terms.*
4. *All of the rest of the operations and equations.*

Repackaging of generated model; we still have Cartmell's theorem!

Categories with families

Theorem (Categories with families (CwFs) [Dyb96])

A model of type theory consists of

1. *A category $\mathcal{C}$ of contexts and substitutions.*
2. *A presheaf* $\mathsf{Ty} : \mathbf{Pr}(\mathcal{C})$ *of types.*
3. *A presheaf* $\mathsf{Tm} : \mathbf{Pr}(\int \mathsf{Ty})$ *of terms.*
4. *All of the rest of the operations and equations.*

The categorical version of Σ

Repackaging of generated model; we still have Cartmell's theorem!

Categories with families

Theorem (Categories with families (CwFs) [Dyb96])

A model of type theory consists of

1. *A category $\mathcal{C}$ of contexts and substitut*
2. *A presheaf* $\mathrm{Ty} : \mathbf{Pr}(\mathcal{C})$ *of types.*
3. *A presheaf* $\mathrm{Tm} : \mathbf{Pr}(\int \mathrm{Ty})$ *of terms.*
4. *All of the rest of the operations and equations.*

A family of presheaves over Ty

Repackaging of generated model; we still have Cartmell's theorem!

Categories with families

Theorem (Categories with families (CwFs) [Dyb96])

A model of type theory consists of

1. *A category $\mathcal{C}$ of contexts and substitutions.*
2. *A presheaf* $\mathrm{Ty} : \mathbf{Pr}(\mathcal{C})$ *of types.*
3. *A presheaf* $\mathrm{Tm} : \mathbf{Pr}(\int \mathrm{Ty})$ *of terms.*
4. *All of the rest of the operations and equations.*

Repackaging of generated model; we still have Cartmell's theorem!

Definition

A model of type theory in $\mathcal{C}$ is a CwF where (1) is given by $\mathcal{C}$.

From universes to CwFs

Continuous problem for type theorists: CwFs simply do not arise in nature.

- Require *coherence constructions* to rectify $\mathcal{C}$ to support Ty/Tm.
- Today, we use the simplest coherence construction I know.
- Downside for simplicity: wasteful use of universes.

This construction is folklore, but see Voevodsky [Voe14] or Angiuli and G.

Universes in a category

Q. Given $\mathcal{C}$, how should we define Ty?

A. Assume $\mathcal{C}$ has an object of types *(a universe)* U; $\mathrm{Ty} = \mathrm{Hom}(-, U)$.

Need a description of universes in a category. See Streicher [Str05].

Universes in a category

Q. Given $\mathcal{C}$, how should we define Ty?

A. Assume $\mathcal{C}$ has an object of types *(a universe)* U; $\mathrm{Ty} = \mathrm{Hom}(-, U)$.

Need a description of universes in a category. See Streicher [Str05].

Theorem (Hofmann and Streicher [HS97])

- *In* $\mathbf{Pr}(\omega)$, *have* $\pi : U^\bullet \longrightarrow U$ *and a bijection* $\iota : \mathrm{Hom}(X, U) \simeq \mathbf{Pr}_{\mathrm{small}}(\int X)$.
- *Moreover, if* $A : X \to U$ *then lifts along* π *corresponds to* $\mathrm{Hom}(\mathbf{1}, \iota(A))$.

$$\begin{tikzcd} & U^\bullet \arrow[d, "\pi"] \\ X \arrow[ur, "a"] \arrow[r, "A"'] & U \end{tikzcd} \qquad \sim \qquad \begin{array}{l} \iota(A) : \mathbf{Pr}_{\mathrm{small}}(\int X) \\[2em] \iota(a) : \mathrm{Hom}_{\mathbf{Pr}_{\mathrm{small}}(\int X)}(\mathbf{1}, \iota(A)) \end{array}$$

Universes in a category

With our Hofmann–Streicher universe to hand, we can describe the skeleton of a CwF:

$$\begin{aligned}
&\mathrm{Ty} : \mathbf{Pr}(\omega)^{\mathrm{op}} \to \mathbf{Set} \\
&\mathrm{Ty}(X) = \mathrm{Hom}(X, U) \\
&\mathrm{Tm} : \textstyle\int \mathrm{Ty}^{\mathrm{op}} \to \mathbf{Set} \\
&\mathrm{Tm}(X, A) = \{a \in \mathrm{Hom}(X, U^{\bullet}) \mid A = \pi \circ a\}
\end{aligned}$$

Elbow grease to close under all the operations (see Hofmann [Hof97] or Angiuli and G.)

Universes in a category

With our Hofmann–Streicher universe to hand, we can describe the skeleton of a CwF:

$$\mathrm{Ty} : \mathbf{Pr}(\omega)^{\mathrm{op}} \to \mathbf{Set}$$
$$\mathrm{Ty}(X) = \mathrm{Hom}(X, U)$$
$$\mathrm{Tm} : \textstyle\int \mathrm{Ty}^{\mathrm{op}} \to \mathbf{Set}$$
$$\mathrm{Tm}(X, A) = \{a \in \mathrm{Hom}(X, U^{\bullet}) \mid A = \pi \circ a\}$$

Only have access to small types ☹

Elbow grease to close under all the operations (see Hofmann [Hof97] or Angiuli and G.)

Taking stock

At this point, we have seen the outline of the following result:

Theorem

There is a model of MLTT in $\mathbf{Pr}(\omega)$.

- Lots of closure conditions to prove, but we have the basic definitions in place.
- What remains: extending this to guarded type theory.

So, we need to describe 🔒, $\triangleright$, $\mathrm{guard}(-)$, $\mathrm{open}(-)$, adv, rec.

Taking stock

At this point, we have seen the outline of the following result:

Theorem

There is a model of MLTT in $\mathbf{Pr}(\omega)$.

- Lots of closure conditions to prove, but we have the basic definitions in place.
- What remains: extending this to guarded type theory.

So, we need to describe 🔒, $\triangleright$, guard$(-)$, open$(-)$, adv, ~~rec~~.

We have no new judgments, so we merely add operations and equations.

For directness, I've also rephrased these operations using the language of categories:

Syntax	Categorical rephrasing
$\Gamma.\text{🔒}$, $\gamma.\text{🔒}$, and equations	$\lhd : \mathbf{Pr}(\omega) \to \mathbf{Pr}(\omega)$
adv and equations	a natural transformation $\lhd \to \mathsf{id}$
$\rhd$ and substitution equation	a natural family of maps $\widehat{\rhd} : \mathsf{Ty}(\lhd\Gamma) \to \mathsf{Ty}(\Gamma)$
guard$(-)$, open$(-)$, and all equations	a natural bijection $\mathsf{Tm}(\lhd\Gamma, A) \cong \mathsf{Tm}(\Gamma, \widehat{\rhd}A)$

Translating these connectives into operations

We have no new judgments, so we merely add operations and equations.

For directness, I've also rephrased these operations using the language of categories:

Syntax	Categorical rephrasing
$\Gamma.\text{🔒}$, $\gamma.\text{🔒}$, and equations	$\triangleleft : \mathbf{Pr}(\omega) \to \mathbf{Pr}(\omega)$ $\checkmark$
adv and equations	a natural transformation $\triangleleft \to \mathrm{id}$ $\checkmark$
$\triangleright$ and substitution equation	a natural family of maps $\widehat{\triangleright} : \mathsf{Ty}(\triangleleft\Gamma) \to \mathsf{Ty}(\Gamma)$
guard$(-)$, open$(-)$, and all equations	a natural bijection $\mathsf{Tm}(\triangleleft\Gamma, A) \cong \mathsf{Tm}(\Gamma, \widehat{\triangleright} A)$

Translating these connectives into operations

We have no new judgments, so we merely add operations and equations.

For directness, I've also rephrased these operations using the language of categories:

Syntax	Categorical rephrasing
$\Gamma.🔒$, $\gamma.🔒$, and equations	$\lhd : \mathbf{Pr}(\omega) \to \mathbf{Pr}(\omega)$ ✓
adv and equations	a natural transformation $\lhd \to \mathsf{id}$ ✓
$\rhd$ and substitution equation	a natural family of maps $\widehat{\rhd} : \mathsf{Ty}(\lhd\Gamma) \to \mathsf{Ty}(\Gamma)$
guard$(-)$, open$(-)$, and all equations	a natural bijection $\mathsf{Tm}(\lhd\Gamma, A) \cong \mathsf{Tm}(\Gamma, \widehat{\rhd}A)$

Constructing $\widehat{\triangleright}$

A natural family of maps $\widehat{\triangleright} : \mathsf{Ty}(\triangleleft\Gamma) \to \mathsf{Ty}(\Gamma)$

Let us recall that $\mathsf{Ty}(\Gamma) = \mathsf{Hom}(\Gamma, U) \simeq \mathbf{Pr}_{\mathrm{small}}(\int \Gamma)$.

$$\widehat{\triangleright}_\Gamma : \mathbf{Pr}_{\mathrm{small}}(\triangleleft \int \Gamma) \to \mathbf{Pr}_{\mathrm{small}}(\int \Gamma)$$

$$\widehat{\triangleright}_\Gamma(X) = \lambda n, \gamma.\ ?$$

($X : \mathbf{Pr}_{\mathrm{small}}(\int \triangleleft\Gamma)$ so in particular $X : (n : \omega) \to \Gamma(n+1) \to \mathbf{Set}$)

Constructing $\widehat{\triangleright}$

A natural family of maps $\widehat{\triangleright} : \mathsf{Ty}(\triangleleft\Gamma) \to \mathsf{Ty}(\Gamma)$

Let us recall that $\mathsf{Ty}(\Gamma) = \mathsf{Hom}(\Gamma, U) \simeq \mathbf{Pr}_{\mathrm{small}}(\int \Gamma)$.

$$\widehat{\triangleright}_\Gamma : \mathbf{Pr}_{\mathrm{small}}(\triangleleft \textstyle\int \Gamma) \to \mathbf{Pr}_{\mathrm{small}}(\textstyle\int \Gamma)$$

$$\widehat{\triangleright}_\Gamma(X) = \lambda n, \gamma.\ \begin{cases} X(n-1, \gamma) & n > 0 \\ \{\star\} & \text{otherwise} \end{cases}$$

($X : \mathbf{Pr}_{\mathrm{small}}(\int \triangleleft\Gamma)$ so in particular $X : (n : \omega) \to \Gamma(n+1) \to \mathbf{Set}$)

The introduction and elimination forms for $\widehat{\triangleright}$

A natural bijection $\mathsf{Tm}(\triangleleft\Gamma, A) \cong \mathsf{Tm}(\Gamma, \widehat{\triangleright} A)$.

Boils down to two observations:

1. It suffices to show $\mathrm{Hom}_{\mathbf{Pr}_{\mathrm{small}}(\int \triangleleft\Gamma)}(\mathbf{1}, \iota(A)) \cong \mathrm{Hom}_{\mathbf{Pr}_{\mathrm{small}}(\int \Gamma)}(\mathbf{1}, \widehat{\triangleright}\iota(A))$
2. $\widehat{\triangleright}_\Gamma$ is right adjoint to precomposition functor $\mathbf{Pr}(\int \Gamma) \to \mathbf{Pr}(\int \triangleleft\Gamma)$.

Result is now an exercise in adjoint yoga.

Deep breath, we're done with math

Why the calculus we just saw is actually unusable

Disappointing reality: this calculus has no substitution lemma:

$$\mathrm{open}(a)[\gamma.🔒] = \mathrm{open}(a[\gamma])$$

What about $\mathrm{open}(a)[\gamma]$ where $\gamma \neq \_.🔒$?

Why the calculus we just saw is actually unusable

Disappointing reality: this calculus has no substitution lemma:

$$\mathsf{open}(a)[\gamma.\faLock] = \mathsf{open}(a[\gamma])$$

What about $\mathsf{open}(a)[\gamma]$ where $\gamma \neq \_.\faLock$?

My tired joke:

> We have a name for type theory with no substitution lemma, category theory.

Possible solutions

1. Ad-hoc tricks to make this work specifically for $\triangleright$ [BGM17; GSB19]:

$$\frac{\Gamma \vdash a : \triangleright A}{\Gamma.🔒.A_1.\dots.A_n \vdash \mathsf{open}(a) : A[\uparrow^n]}$$

2. Utilize additional structure of earlier [Gra+22] (🔓)
3. Use a more complex modal framework [Gra+20] (MTT)

None of these fundamentally impact the model; this is purely a matter of syntax.

Open questions in guarded type theory

1. Does there exist “a well-adapted guarded recursive dependent type theory”
    - $\triangleright$ (plus other modalities)
    - rec with propositional unfolding
    - Canonicity
    - Decidable type-checking.
2. What is the correct set of modalities for guarded recursion?
3. Is Löb induction the right primitive for guarded recursion?
4. Many questions in synthetic guarded domain theory (weak bisimulation? guarded equational theories?)

References

[Ahm04] Amal Jamil Ahmed. "Semantics of Types for Mutable State". AAI3136691. PhD thesis. USA: Princeton University, 2004 (cit. on p. 41).

[AM01] Andrew W. Appel and David McAllester. "An indexed model of recursive types for foundational proof-carrying code". In: *ACM Transactions on Programming Languages and Systems* 23.5 (2001), pp. 657–683. DOI: 10.1145/504709.504712 (cit. on pp. 41, 43, 44).

[App+07] Andrew W. Appel, Paul-André Melliès, Christina D. Richards, and Jérôme Vouillon. "A very modal model of a modern, major, general type system". In: *Proceedings of the 34th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '07. Nice, France: Association for Computing Machinery, 2007, pp. 109–122. ISBN: 1595935754. DOI: 10.1145/1190216.1190235 (cit. on p. 41).

[BBM14] Ales Bizjak, Lars Birkedal, and Marino Miculan. "A Model of Countable Nondeterminism in Guarded Type Theory". In: *Rewriting and Typed Lambda Calculi – Joint International Conference, RTA-TLCA 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*. Ed. by Gilles Dowek. Vol. 8560. Lecture Notes in Computer Science. Springer, 2014, pp. 108–123. ISBN: 978-3-319-08917-1. DOI: 10.1007/978-3-319-08918-8_8 (cit. on p. 69).

[BGM17] Patrick Bahr, Hans Bugge Grathwohl, and Rasmus Ejlers Møgelberg. “The clocks are ticking: No more delays!” In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 2017. DOI: 10.1109/LICS.2017.8005097. URL: http://www.itu.dk/people/mogel/papers/lics2017.pdf (cit. on pp. 67, 97).

[Bir+12] Lars Birkedal, Rasmus Møgelberg, Jan Schwinghammer, and Kristian Støvring. “First steps in synthetic guarded domain theory: step-indexing in the topos of trees”. In: *Logical Methods in Computer Science* 8.4 (2012). Ed. by Patrick Baillot. DOI: 10.2168/LMCS-8(4:1)2012 (cit. on pp. 67, 71, 72).

[Bir+19] Lars Birkedal, Aleš Bizjak, Ranald Clouston, Hans Bugge Grathwohl, Bas Spitters, and Andrea Vezzosi. "Guarded Cubical Type Theory". In: *Journal of Automated Reasoning* 63 (2019), pp. 211–253 (cit. on p. 67).

[Bir+20] Lars Birkedal, Ranald Clouston, Bassel Mannaa, Rasmus Ejlers Møgelberg, Andrew M. Pitts, and Bas Spitters. "Modal dependent type theory and dependent right adjoints". In: *Mathematical Structures in Computer Science* 30.2 (2020), pp. 118–138. DOI: 10.1017/S0960129519000197. eprint: 1804.05236 (cit. on p. 58).

[BM13] Lars Birkedal and Rasmus Ejlers Møgelberg. "Intensional Type Theory with Guarded Recursive Types qua Fixed Points on Universes". In: *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '13. USA: IEEE Computer Society, 2013, pp. 213–222. ISBN: 9780769550206. DOI: 10.1109/LICS.2013.27 (cit. on pp. 71, 72).

[Car78] John Cartmell. "Generalised Algebraic Theories and Contextual Categories". PhD thesis. University of Oxford, 1978 (cit. on pp. 73–75).

[DAB11] Derek Dreyer, Amal Ahmed, and Lars Birkedal. "Logical Step-Indexed Logical Relations". In: *Logical Methods in Computer Science* Volume 7, Issue 2 (June 2011). DOI: `10.2168/lmcs-7(2:16)2011` (cit. on p. 41).

[Dyb96] Peter Dybjer. "Internal type theory". In: *Types for Proofs and Programs*. Ed. by Stefano Berardi and Mario Coppo. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 120–134. DOI: `10.1007/3-540-61780-9_66` (cit. on pp. 77–80).

[GB22] Daniel Gratzer and Lars Birkedal. "A Stratified Approach to Löb Induction". In: *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*. Ed. by Amy Felty. Vol. 228. Leibniz International Proceedings in Informatics (LIPIcs). Saarbrücken, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2022. DOI: 10.4230/LIPIcs.FSCD.2022.3. URL: https://jozefg.github.io/papers/a-stratified-approach-to-lob-induction.pdf (cit. on p. 67).

[Gra+20] Daniel Gratzer, G.A. Kavvos, Andreas Nuyts, and Lars Birkedal. "Multimodal Dependent Type Theory". In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '20. ACM, 2020. DOI: 10.1145/3373718.3394736 (cit. on p. 97).

[Gra+22] Daniel Gratzer, Evan Cavallo, G. A. Kavvos, Adrien Guatto, and Lars Birkedal. “Modalities and Parametric Adjoints”. In: *ACM Trans. Comput. Logic* 23.3 (Apr. 2022). ISSN: 1529-3785. DOI: 10.1145/3514241. URL: https://doi.org/10.1145/3514241 (cit. on p. 97).

[Gra25] Daniel Gratzer. “A Modal Deconstruction of Löb Induction”. In: *Proc. ACM Program. Lang.* 9.POPL (Jan. 2025). DOI: 10.1145/3704866 (cit. on p. 67).

[GSB19] Daniel Gratzer, Jonathan Sterling, and Lars Birkedal. “Implementing a Modal Dependent Type Theory”. In: *Proc. ACM Program. Lang.* 3 (ICFP 2019). DOI: 10.1145/3341711 (cit. on p. 97).

[Hof97] Martin Hofmann. "Syntax and Semantics of Dependent Types". In: *Semantics and Logics of Computation*. Ed. by Andrew M. Pitts and P. Dybjer. Cambridge University Press, 1997, pp. 79–130. DOI: 10.1017/CBO9780511526619.004. URL: https://www.tcs.ifi.lmu.de/mitarbeiter/martin-hofmann/pdfs/syntaxandsemanticsof-dependenttypes.pdf (cit. on pp. 84, 85).

[HS97] Martin Hofmann and Thomas Streicher. "Lifting Grothendieck Universes". Unpublished note. 1997. URL: https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf (cit. on pp. 82, 83).

[Jun+18] Ralf Jung, Robbert Krebbers, Jacques-Henri Jourdan, Aleš Bizjak, Lars Birkedal, and Derek Dreyer. "Iris from the ground up: A modular foundation for higher-order concurrent separation logic". In: *Journal of Functional Programming* 28 (2018). DOI: 10.1017/S0956796818000151 (cit. on pp. 43, 44).

[KMV22] Magnus Baunsgaard Kristensen, Rasmus Ejlers Møgelberg, and Andrea Vezzosi. "Greatest HITs: Higher inductive types in coinductive definitions via induction under clocks". In: *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*. New York, NY, USA: Association for Computing Machinery, 2022. DOI: 10.1145/3531130.3533359 (cit. on p. 67).

[MM92] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic : a first introduction to topos theory*. Universitext. New York: Springer, 1992. ISBN: 0-387-97710-4 (cit. on pp. 25, 26).

[Møg14] Rasmus Ejlers Møgelberg. "A Type Theory for Productive Coprogramming via Guarded Recursion". In: *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. CSL-LICS '14. 2014. DOI: 10.1145/2603088.2603132 (cit. on p. 67).

[Nak00] H. Nakano. "A modality for recursion". In: *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.99CB36332)*. IEEE Computer Society, 2000, pp. 255–266 (cit. on p. 2).

[Pav16] Marco Paviotti. “Denotational semantics in Synthetic Guarded Domain Theory”. PhD thesis. IT University of Copenhagen, 2016 (cit. on p. 69).

[PMB15] Marco Paviotti, Rasmus Ejlers Møgelberg, and Lars Birkedal. “A Model of PCF in Guarded Type Theory”. In: *Electronic Notes in Theoretical Computer Science* 319.Supplement C (2015). The 31st Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXI), pp. 333–349. ISSN: 1571-0661. DOI: `10.1016/j.entcs.2015.12.020` (cit. on p. 69).

[SGB23] Jonathan Sterling, Daniel Gratzer, and Lars Birkedal. *Denotational semantics of general store and polymorphism*. 2023. arXiv: `2210.02169 [cs.PL]` (cit. on p. 69).

[Str05] Thomas Streicher. “Universes in toposes”. In: *From Sets and Types to Topology and Analysis: Towards practical foundations for constructive mathematics*. Ed. by Laura Crosilla and Peter Schuster. Vol. 48. Oxford Logical Guides. Oxford: Oxford University Press, 2005, pp. 78–90. ISBN: 978-0-19-856651-9. DOI: `10.1093/acprof:oso/9780198566519.001.0001` (cit. on pp. 82, 83).

[Voe14] Vladimir Voevodsky. “A C-system defined by a universe category”. In: *Theory and Applications of Categories* 30 (Sept. 2014) (cit. on p. 81).

[VV20] Niccolò Veltri and Andrea Vezzosi. “Formalizing π-calculus in guarded cubical Agda”. In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*. 2020, pp. 270–283 (cit. on p. 69).