A Categorical Normalization Proof for the Modal Lambda-Calculus

Jason Hu and Brigitte Pientka

McGill University

on the Occasion of the 38th MFPS

- There is a correspondence between logic and type theory, e.g. propositional logic and STLC, etc.

- There is a correspondence between logic and type theory, e.g. propositional logic and STLC, etc.
- What modal logic corresponds to?

- There is a correspondence between logic and type theory, e.g. propositional logic and STLC, etc.
- What modal logic corresponds to?
- How to formulate modalities in type theory?

Long Journey to Modal Type Theories

- Early papers explore formulations of modal logics in natural deduction (Borghuis, 1994; Bierman and de Paiva, 2000; Bierman and de Paiva, 1996; Bellin et al., 2001; Pfenning and Wong, 1995, etc.)

Long Journey to Modal Type Theories

- Early papers explore formulations of modal logics in natural deduction (Borghuis, 1994; Bierman and de Paiva, 2000; Bierman and de Paiva, 1996; Bellin et al., 2001; Pfenning and Wong, 1995, etc.)
- Modalities are still popular recently (Pientka et al., 2019; Zyuzin and Nanevski, 2021; Gratzer et al., 2019, 2020; Kavvos, 2017, etc.)
  - HoTT (Licata et al., 2018; Shulman, 2018)
  - metaprogramming (Jang et al., 2022)

Long Journey to Modal Type Theories

- Early papers explore formulations of modal logics in natural deduction (Borghuis, 1994; Bierman and de Paiva, 2000; Bierman and de Paiva, 1996; Bellin et al., 2001; Pfenning and Wong, 1995, etc.)
- Modalities are still popular recently (Pientka et al., 2019; Zyuzin and Nanevski, 2021; Gratzer et al., 2019, 2020; Kavvos, 2017, etc.)
  - HoTT (Licata et al., 2018; Shulman, 2018)
  - metaprogramming (Jang et al., 2022)
- System $\lambda^{\to\Box}$ (Davies and Pfenning, 2001), $S4$ in Kripke style, corresponds to meta-programming in quasi-quote style

Contributions

- Unified substitutions; enabling a substitution calculus for Kripke-style modal systems

Contributions

- Unified substitutions; enabling a substitution calculus for Kripke-style modal systems
- A unified normalization proof for all modal systems K, T, $K4$ and $S4$

Contributions

- Unified substitutions; enabling a substitution calculus for Kripke-style modal systems
- A unified normalization proof for all modal systems K, T, $K4$ and $S4$
- A formulation of contextual types in Kripke style; a foundation of meta-programming with open code

Kripke Style and Kripke Semantics

- A stack of contexts: each context represents a Kripke world

$$\epsilon; \Gamma_1; \cdots ; \Gamma_n \vdash t : T \qquad \text{or} \qquad \vec{\Gamma} \vdash t : T$$

Kripke Style and Kripke Semantics

- A stack of contexts: each context represents a Kripke world

$$\epsilon; \Gamma_1; \cdots; \Gamma_n \vdash t : T \qquad \text{or} \qquad \overrightarrow{\Gamma} \vdash t : T$$

- $\square$ denotes the next world; move among worlds by its introduction and elimination

Kripke Style and Kripke Semantics

- A stack of contexts: each context represents a Kripke world

$$\epsilon; \Gamma_1; \cdots; \Gamma_n \vdash t : T \qquad \text{or} \qquad \vec{\Gamma} \vdash t : T$$

- $\square$ denotes the next world; move among worlds by its introduction and elimination
-

$$\frac{x : T \in \Gamma}{\vec{\Gamma}; \Gamma \vdash x : T}$$

- A stack of contexts: each context represents a Kripke world

$$\epsilon; \Gamma_1; \cdots; \Gamma_n \vdash t : T \qquad \text{or} \qquad \vec{\Gamma} \vdash t : T$$

- $\square$ denotes the next world; move among worlds by its introduction and elimination
-

$$\frac{x : T \in \Gamma}{\vec{\Gamma}; \Gamma \vdash x : T} \qquad \frac{\vec{\Gamma}; \cdot \vdash t : T}{\vec{\Gamma} \vdash \texttt{box}\ t : \square T}$$

Kripke Style and Kripke Semantics

- A stack of contexts: each context represents a Kripke world

$$\epsilon; \Gamma_1; \cdots ; \Gamma_n \vdash t : T \qquad \text{or} \qquad \overrightarrow{\Gamma} \vdash t : T$$

- $\Box$ denotes the next world; move among worlds by its introduction and elimination
-

$$\frac{x : T \in \Gamma}{\overrightarrow{\Gamma}; \Gamma \vdash x : T} \qquad \frac{\overrightarrow{\Gamma}; \cdot \vdash t : T}{\overrightarrow{\Gamma} \vdash \text{box } t : \Box T} \qquad \frac{\overrightarrow{\Gamma} \vdash t : \Box T \quad |\overrightarrow{\Delta}| = n}{\overrightarrow{\Gamma}; \overrightarrow{\Delta} \vdash \texttt{unbox}_n\ t : T}$$

Kripke Structure

$$\frac{\overrightarrow{\Gamma} \vdash t : \Box T \qquad |\overrightarrow{\Delta}| = n}{\overrightarrow{\Gamma} ; \overrightarrow{\Delta} \vdash \texttt{unbox}_n \; t : T}$$

Kripke Structure

$$\frac{\vec{\Gamma} \vdash t : \Box T \qquad |\vec{\Delta}| = n}{\vec{\Gamma}; \vec{\Delta} \vdash \mathtt{unbox}_n\ t : T}$$

$\mathtt{unbox}$ level n	Axiom \ System	K	T	$K4$	$S4$
$n = 1$	K: $\Box(S \longrightarrow T) \to \Box S \to \Box T$	✓	✓	✓	✓
$n = 0$	T: $\Box T \to T$		✓		✓
$n \geq 2$	4: $\Box T \to \Box\Box T$			✓	✓
	combined $\mathtt{unbox}$ level (UL)	$\{1\}$	$\{0, 1\}$	$\mathbb{N}^+$	$\mathbb{N}$

Kripke Structure

$$\frac{\vec{\Gamma} \vdash t : \Box T \qquad |\vec{\Delta}| = n}{\vec{\Gamma} ; \vec{\Delta} \vdash \mathtt{unbox}_n \; t : T}$$

unbox level n	Axiom \ System	K	T	$K4$	$S4$
$n = 1$	K: $\Box(S \longrightarrow T) \to \Box S \to \Box T$	✓	✓	✓	✓
$n = 0$	T: $\Box T \to T$		✓		✓
$n \geq 2$	4: $\Box T \to \Box\Box T$			✓	✓
	combined unbox level (UL)	$\{1\}$	$\{0, 1\}$	$\mathbb{N}^+$	$\mathbb{N}$

Kripke Structure

$$\frac{\vec{\Gamma} \vdash t : \Box T \qquad |\vec{\Delta}| = n}{\vec{\Gamma} ; \vec{\Delta} \vdash \texttt{unbox}_n\ t : T}$$

$\texttt{unbox}$ level n	Axiom \ System	K	T	$K4$	$S4$
$n = 1$	K: $\Box(S \longrightarrow T) \to \Box S \to \Box T$	✓	✓	✓	✓
$n = 0$	T: $\Box T \to T$		✓		✓
$n \geq 2$	4: $\Box T \to \Box\Box T$			✓	✓
	combined $\texttt{unbox}$ level (UL)	$\{1\}$	$\{0, 1\}$	$\mathbb{N}^+$	$\mathbb{N}$

A Challenge

$\lambda^{\to\square}$ in Davies and Pfenning (2001) involves two operations that don't play well together:

- ordinary substitutions (required for β for functions)
- modal transformations (MoTs, required for β for $\square$)

Substitutions and Dynamics

Substitutions needed due to β equivalence for functions

$$\frac{\overrightarrow{\Gamma}; (\Gamma, x : S) \vdash t : T \qquad \overrightarrow{\Gamma}; \Gamma \vdash s : S}{\overrightarrow{\Gamma}; \Gamma \vdash (\lambda x.t)s \approx t[s/x] : T}$$

Substitutions and Dynamics

Substitutions needed due to β equivalence for functions

$$\frac{\overrightarrow{\Gamma}; (\Gamma, x : S) \vdash t : T \qquad \overrightarrow{\Gamma}; \Gamma \vdash s : S}{\overrightarrow{\Gamma}; \Gamma \vdash (\lambda x.t)s \approx t[s/x] : T}$$

Right hand side requires substitution and substitution property.

What about $\Box$?

- What about $\Box$?

$$\frac{\overrightarrow{\Gamma} ; \cdot \vdash t : T \qquad |\overrightarrow{\Delta}| = n}{\overrightarrow{\Gamma} ; \overrightarrow{\Delta} \vdash \texttt{unbox}_n\ (\texttt{box}\ t) \approx ?? : T}$$

What about $\square$?

- What about $\square$?

$$\frac{\overrightarrow{\Gamma}; \cdot \vdash t : T \qquad |\overrightarrow{\Delta}| = n}{\overrightarrow{\Gamma}; \overrightarrow{\Delta} \vdash \mathtt{unbox}_n\ (\mathtt{box}\ t) \approx ?? : T}$$

- Need an operation transforming t from $\overrightarrow{\Gamma}; \cdot$ to $\overrightarrow{\Gamma}; \overrightarrow{\Delta}$

What about $\square$?

- What about $\square$?

$$\frac{\overrightarrow{\Gamma}; \cdot \vdash t : T \qquad |\overrightarrow{\Delta}| = n}{\overrightarrow{\Gamma}; \overrightarrow{\Delta} \vdash \texttt{unbox}_n\ (\texttt{box}\ t) \approx t\{n/0\} : T}$$

- Need an operation transforming t from $\overrightarrow{\Gamma}; \cdot$ to $\overrightarrow{\Gamma}; \overrightarrow{\Delta}$
- Modal transformation: $\overrightarrow{\Gamma}; \overrightarrow{\Delta} \vdash t\{n/0\} : T$

Modal Transformations (MoTs)

- MoTs transform a term from one world to another (if possible) by changing the `unbox` levels

Modal Transformations (MoTs)

- MoTs transform a term from one world to another (if possible) by changing the `unbox` levels
- $t\{k/l\}$ defined by recursion on t

Modal Transformations (MoTs)

- MoTs transform a term from one world to another (if possible) by changing the $\mathtt{unbox}$ levels
- $t\{k/l\}$ defined by recursion on t
- $\mathtt{unbox}$ case has case distinction:

$$\mathtt{unbox}_n\ t\{k/l\} := \begin{cases} \mathtt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \mathtt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

Modal Transformations (MoTs)

- MoTs transform a term from one world to another (if possible) by changing the `unbox` levels
- $t\{k/l\}$ defined by recursion on t
- `unbox` case has case distinction:

$$\mathtt{unbox}_n\ t\{k/l\} := \begin{cases} \mathtt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \mathtt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

- x2 cases for every MoT $\Rightarrow O(2^n)$ cases for composition of n MoTs

Modal Transformations (MoTs)

- MoTs transform a term from one world to another (if possible) by changing the $\mathtt{unbox}$ levels
- $t\{k/l\}$ defined by recursion on t
- $\mathtt{unbox}$ case has case distinction:

$$\mathtt{unbox}_n\ t\{k/l\} := \begin{cases} \mathtt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \mathtt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

- x2 cases for every MoT $\Rightarrow O(2^n)$ cases for composition of n MoTs
  $\Rightarrow$ too difficult to reason

Unified Substitutions

Need a substitution calculus to reason both ordinary substitutions and MoTs

Unified Substitutions

Need a substitution calculus to reason both ordinary substitutions and MoTs
Insight: substitutions and MoTs are one concept.

Unified Substitutions

Need a substitution calculus to reason both ordinary substitutions and MoTs

Insight: substitutions and MoTs are one concept.

Key idea: keep track of `unbox` offsets in substitutions

σ Local substitutions; lists of terms as usual

Unified Substitutions

Need a substitution calculus to reason both ordinary substitutions and MoTs
Insight: substitutions and MoTs are one concept.
Key idea: keep track of `unbox` offsets in substitutions

σ Local substitutions; lists of terms as usual

$\overrightarrow{\sigma}, \overrightarrow{\delta} := \varepsilon; \sigma \mid \overrightarrow{\sigma}; \Uparrow^n \sigma$ Unified substitutions, Substs

Unified Substitutions

Need a substitution calculus to reason both ordinary substitutions and MoTs
Insight: substitutions and MoTs are one concept.
Key idea: keep track of `unbox` offsets in substitutions

σ Local substitutions; lists of terms as usual

$\vec{\sigma}, \vec{\delta} \ := \varepsilon; \sigma \mid \vec{\sigma}; \Uparrow^n \sigma$ Unified substitutions, Substs

$$\frac{\sigma : \vec{\Gamma} \Rightarrow \Gamma}{\varepsilon; \sigma : \vec{\Gamma} \Rightarrow \epsilon; \Gamma} \qquad \frac{\vec{\sigma} : \vec{\Gamma} \Rightarrow \vec{\Delta} \qquad |\vec{\Gamma}'| = n \qquad \sigma : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow \Delta}{\vec{\sigma}; \Uparrow^n \sigma : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow \vec{\Delta}; \Delta}$$

Operating on Unified Substitutions

- Truncation $(\overrightarrow{\sigma} \mid n)$: drops topmost n step cases from $\overrightarrow{\sigma}$

Operating on Unified Substitutions

- Truncation $(\overrightarrow{\sigma} \mid n)$: drops topmost n step cases from $\overrightarrow{\sigma}$
  - Recall

$$\frac{\overrightarrow{\sigma} : \overrightarrow{\Gamma} \Rightarrow \overrightarrow{\Delta} \qquad |\overrightarrow{\Gamma}'| = m \qquad \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \Delta}{\overrightarrow{\sigma}; \Uparrow^m \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \overrightarrow{\Delta}; \Delta}$$

Operating on Unified Substitutions

- Truncation $(\overrightarrow{\sigma} \mid n)$: drops topmost n step cases from $\overrightarrow{\sigma}$
  - Recall

$$\frac{\overrightarrow{\sigma} : \overrightarrow{\Gamma} \Rightarrow \overrightarrow{\Delta} \qquad |\overrightarrow{\Gamma}'| = m \qquad \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \Delta}{\overrightarrow{\sigma}; \Uparrow^m \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \overrightarrow{\Delta}; \Delta}$$

  - $(\overrightarrow{\sigma}; \Uparrow^m \sigma) \mid 1 + n = \overrightarrow{\sigma} \mid n$

Operating on Unified Substitutions

- Truncation $(\vec{\sigma} \mid n)$: drops topmost n step cases from $\vec{\sigma}$
  - Recall

$$\frac{\vec{\sigma} : \vec{\Gamma} \Rightarrow \vec{\Delta} \qquad |\vec{\Gamma}'| = m \qquad \sigma : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow \Delta}{\vec{\sigma}; \Uparrow^m \sigma : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow \vec{\Delta}; \Delta}$$

  - $(\vec{\sigma}; \Uparrow^m \sigma) \mid 1 + n = \vec{\sigma} \mid n$
  - If $\vec{\delta} : \vec{\Gamma} \Rightarrow \vec{\Delta}$ and $n < |\vec{\Delta}|$, then

$$\vec{\delta} \mid n : \vec{\Gamma} \mid ?? \Rightarrow \vec{\Delta} \mid n$$

Operating on Unified Substitutions

- Truncation ($\overrightarrow{\sigma} \mid n$): drops topmost n step cases from $\overrightarrow{\sigma}$
  - Recall

$$\frac{\overrightarrow{\sigma} : \overrightarrow{\Gamma} \Rightarrow \overrightarrow{\Delta} \qquad |\overrightarrow{\Gamma}'| = m \qquad \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \Delta}{\overrightarrow{\sigma}; \Uparrow^m \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \overrightarrow{\Delta}; \Delta}$$

  - $(\overrightarrow{\sigma}; \Uparrow^m \sigma) \mid 1 + n = \overrightarrow{\sigma} \mid n$
  - If $\overrightarrow{\delta} : \overrightarrow{\Gamma} \Rightarrow \overrightarrow{\Delta}$ and $n < |\overrightarrow{\Delta}|$, then

$$\overrightarrow{\delta} \mid n : \overrightarrow{\Gamma} \mid ?? \Rightarrow \overrightarrow{\Delta} \mid n$$

- Truncation offset ($\mathcal{O}(\overrightarrow{\sigma}, n)$): computes the sum of topmost n offsets

Operating on Unified Substitutions

- Truncation ($\vec{\sigma} \mid n$): drops topmost n step cases from $\vec{\sigma}$
  - Recall

$$\frac{\vec{\sigma} : \vec{\Gamma} \Rightarrow \vec{\Delta} \qquad |\vec{\Gamma}'| = m \qquad \sigma : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow \Delta}{\vec{\sigma}; \Uparrow^m \sigma : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow \vec{\Delta}; \Delta}$$

  - $(\vec{\sigma}; \Uparrow^m \sigma) \mid 1 + n = \vec{\sigma} \mid n$
  - If $\vec{\delta} : \vec{\Gamma} \Rightarrow \vec{\Delta}$ and $n < |\vec{\Delta}|$, then

$$\vec{\delta} \mid n : \vec{\Gamma} \mid ?? \Rightarrow \vec{\Delta} \mid n$$

- Truncation offset ($\mathcal{O}(\vec{\sigma}, n)$): computes the sum of topmost n offsets
  - $\mathcal{O}((\vec{\sigma}; \Uparrow^m \sigma), 1 + n) = m + \mathcal{O}(\vec{\sigma}, n)$

Operating on Unified Substitutions

- Truncation ($\overrightarrow{\sigma} \mid n$): drops topmost n step cases from $\overrightarrow{\sigma}$
  - Recall

$$\frac{\overrightarrow{\sigma} : \overrightarrow{\Gamma} \Rightarrow \overrightarrow{\Delta} \qquad |\overrightarrow{\Gamma}'| = m \qquad \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \Delta}{\overrightarrow{\sigma}; \Uparrow^m \sigma : \overrightarrow{\Gamma}; \overrightarrow{\Gamma}' \Rightarrow \overrightarrow{\Delta}; \Delta}$$

  - $(\overrightarrow{\sigma}; \Uparrow^m \sigma) \mid 1 + n = \overrightarrow{\sigma} \mid n$
  - If $\overrightarrow{\delta} : \overrightarrow{\Gamma} \Rightarrow \overrightarrow{\Delta}$ and $n < |\overrightarrow{\Delta}|$, then

$$\overrightarrow{\delta} \mid n : \overrightarrow{\Gamma} \mid ?? \Rightarrow \overrightarrow{\Delta} \mid n$$

- Truncation offset ($\mathcal{O}(\overrightarrow{\sigma}, n)$): computes the sum of topmost n offsets
  - $\mathcal{O}((\overrightarrow{\sigma}; \Uparrow^m \sigma), 1 + n) = m + \mathcal{O}(\overrightarrow{\sigma}, n)$
  -

$$\overrightarrow{\delta} \mid n : \overrightarrow{\Gamma} \mid \mathcal{O}(\overrightarrow{\delta}, n) \Rightarrow \overrightarrow{\Delta} \mid n$$

Structure of Unified Substitutions

- Recall MoTs have case analysis

$$\mathtt{unbox}_n\ t\{k/l\} := \begin{cases} \mathtt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \mathtt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

Structure of Unified Substitutions

- Recall MoTs have case analysis

$$\texttt{unbox}_n\ t\{k/l\} := \begin{cases} \texttt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \texttt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

- No case analysis anymore

$$\texttt{unbox}_n\ t[\overrightarrow{\sigma}] := \texttt{unbox}_{\mathcal{O}(\overrightarrow{\sigma},n)}\ (t[\overrightarrow{\sigma} \mid n])$$

Structure of Unified Substitutions

- Recall MoTs have case analysis

$$\texttt{unbox}_n\ t\{k/l\} := \begin{cases} \texttt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \texttt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

- No case analysis anymore

$$\texttt{unbox}_n\ t[\overrightarrow{\sigma}] := \texttt{unbox}_{\mathcal{O}(\overrightarrow{\sigma},n)}\ (t[\overrightarrow{\sigma} \mid n])$$

- Form a category: identity and composition

Structure of Unified Substitutions

- Recall MoTs have case analysis

$$\mathtt{unbox}_n\ t\{k/l\} := \begin{cases} \mathtt{unbox}_n\ (t\{k/l-n\}) & \text{if } n \leq l \\ \mathtt{unbox}_{k+n-1}\ t & \text{if } n > l \end{cases}$$

- No case analysis anymore

$$\mathtt{unbox}_n\ t[\vec{\sigma}] := \mathtt{unbox}_{\mathcal{O}(\vec{\sigma},n)}\ (t[\vec{\sigma} \mid n])$$

- Form a category: identity and composition
- See definitions in paper

Then, What is so Great?

- Immediate and simple normalization proof!

Then, What is so Great?

- Immediate and simple normalization proof!
- A direct and minimal extension of standard presheaf model (Altenkirch et al., 1995)

Then, What is so Great?

- Immediate and simple normalization proof!
- A direct and minimal extension of standard presheaf model (Altenkirch et al., 1995)
- Simultaneously done for all four systems (K, T, $K4$, $S4$)

Unified Weakenings

- Serve as the base category of a presheaf model for normalization proof

Unified Weakenings

- Serve as the base category of a presheaf model for normalization proof
- Extend standard category of weakenings (Altenkirch et al., 1995)

Unified Weakenings

- Serve as the base category of a presheaf model for normalization proof
- Extend standard category of weakenings (Altenkirch et al., 1995)
-

$$\frac{}{\varepsilon : \epsilon; \cdot \Rightarrow_w \epsilon; \cdot} \qquad \frac{\vec{\gamma} : \vec{\Gamma}; \Gamma \Rightarrow_w \vec{\Delta}; \Delta}{q(\vec{\gamma}) : \vec{\Gamma}; (\Gamma, x : T) \Rightarrow_w \vec{\Delta}; (\Delta, x : T)}$$

$$\frac{\vec{\gamma} : \vec{\Gamma}; \Gamma \Rightarrow_w \vec{\Delta}; \Delta}{p(\vec{\gamma}) : \vec{\Gamma}; (\Gamma, x : T) \Rightarrow_w \vec{\Delta}; \Delta} \qquad \frac{\vec{\gamma} : \vec{\Gamma} \Rightarrow_w \vec{\Delta} \quad |\vec{\Gamma}'| = n}{\vec{\gamma}; \Uparrow^n : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow_w \vec{\Delta}; \cdot}$$

Unified Weakenings

- Serve as the base category of a presheaf model for normalization proof
- Extend standard category of weakenings (Altenkirch et al., 1995)
-

$$\frac{}{\varepsilon : \epsilon; \cdot \Rightarrow_w \epsilon; \cdot} \qquad \frac{\vec{\gamma} : \vec{\Gamma}; \Gamma \Rightarrow_w \vec{\Delta}; \Delta}{q(\vec{\gamma}) : \vec{\Gamma}; (\Gamma, x : T) \Rightarrow_w \vec{\Delta}; (\Delta, x : T)}$$

$$\frac{\vec{\gamma} : \vec{\Gamma}; \Gamma \Rightarrow_w \vec{\Delta}; \Delta}{p(\vec{\gamma}) : \vec{\Gamma}; (\Gamma, x : T) \Rightarrow_w \vec{\Delta}; \Delta} \qquad \frac{\vec{\gamma} : \vec{\Gamma} \Rightarrow_w \vec{\Delta} \qquad |\vec{\Gamma}'| = n}{\vec{\gamma}; \Uparrow^n : \vec{\Gamma}; \vec{\Gamma}' \Rightarrow_w \vec{\Delta}; \cdot}$$

- Slogan: MoTs are just weakenings

Normalization

- Moderate and simple extension of Altenkirch et al. (1995)

Normalization

- Moderate and simple extension of Altenkirch et al. (1995)
- $$[\![\square T]\!]_{\vec{\Delta}} := [\![T]\!]_{\vec{\Delta};\cdot}$$

Normalization

- Moderate and simple extension of Altenkirch et al. (1995)
-
$$[\![\square T]\!]_{\vec{\Delta}} := [\![T]\!]_{\vec{\Delta};\cdot}$$
-
$$\begin{aligned}[\![\epsilon;\Gamma]\!]_{\vec{\Delta}} &:= [\![\Gamma]\!]_{\vec{\Delta}} \\ [\![\vec{\Gamma};\Gamma]\!]_{\vec{\Delta}} &:= (\Sigma n < |\vec{\Delta}|.[\![\vec{\Gamma}]\!]_{\vec{\Delta}|n}) \times [\![\Gamma]\!]_{\vec{\Delta}}\end{aligned}$$

Normalization

- Moderate and simple extension of Altenkirch et al. (1995)
- $$[\![\Box T]\!]_{\vec{\Delta}} := [\![T]\!]_{\vec{\Delta};\cdot}$$
- $$[\![\epsilon;\Gamma]\!]_{\vec{\Delta}} := [\![\Gamma]\!]_{\vec{\Delta}}$$
  $$[\![\vec{\Gamma};\Gamma]\!]_{\vec{\Delta}} := (\Sigma n < |\vec{\Delta}|.[\![\vec{\Gamma}]\!]_{\vec{\Delta}|_n}) \times [\![\Gamma]\!]_{\vec{\Delta}}$$
- That's it! Simple yet works for all four systems

Normalization

- Moderate and simple extension of Altenkirch et al. (1995)
- $$[\![\Box T]\!]_{\vec{\Delta}} := [\![T]\!]_{\vec{\Delta};\cdot}$$
- $$[\![\epsilon;\Gamma]\!]_{\vec{\Delta}} := [\![\Gamma]\!]_{\vec{\Delta}}$$
  $$[\![\vec{\Gamma};\Gamma]\!]_{\vec{\Delta}} := (\Sigma n < |\vec{\Delta}|.[\![\vec{\Gamma}]\!]_{\vec{\Delta}|n}) \times [\![\Gamma]\!]_{\vec{\Delta}}$$
- That's it! Simple yet works for all four systems
- Secrete: the additional case in unified weakenings

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)
  - but $\square$ only handles closed code!

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)
  - but $\Box$ only handles closed code!
- Contextual types (Nanevski et al., 2008) also handle open code, but only available in dual-context style

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)
  - but $\Box$ only handles closed code!
- Contextual types (Nanevski et al., 2008) also handle open code, but only available in dual-context style
- Contextual types in Kripke style, allowing open code w.r.t. context stacks:

$$\frac{\vec{\Gamma}; \vec{\Delta} \vdash t : T}{\vec{\Gamma} \vdash \ulcorner \vec{\Delta} \vdash t \urcorner : \ulcorner \vec{\Delta} \vdash T \urcorner} \qquad \frac{\vec{\Gamma} \mid \mathcal{O}(\vec{\sigma}) \vdash t : \ulcorner \vec{\Delta} \vdash T \urcorner \qquad \vec{\sigma} : \vec{\Gamma} \Rightarrow_s \vec{\Delta}}{\vec{\Gamma} \vdash \lfloor t \rfloor_{\vec{\sigma}} : T}$$

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)
  - but $\square$ only handles closed code!
- Contextual types (Nanevski et al., 2008) also handle open code, but only available in dual-context style
- Contextual types in Kripke style, allowing open code w.r.t. context stacks:

$$\frac{\vec{\Gamma}; \vec{\Delta} \vdash t : T}{\vec{\Gamma} \vdash \lceil \vec{\Delta} \vdash t \rceil : \lceil \vec{\Delta} \vdash T \rceil} \qquad \frac{\vec{\Gamma} \mid \mathcal{O}(\vec{\sigma}) \vdash t : \lceil \vec{\Delta} \vdash T \rceil \qquad \vec{\sigma} : \vec{\Gamma} \Rightarrow_s \vec{\Delta}}{\vec{\Gamma} \vdash \lfloor t \rfloor_{\vec{\sigma}} : T}$$

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)
  - but $\square$ only handles closed code!
- Contextual types (Nanevski et al., 2008) also handle open code, but only available in dual-context style
- Contextual types in Kripke style, allowing open code w.r.t. context stacks:

$$\frac{\vec{\Gamma}; \vec{\Delta} \vdash t : T}{\vec{\Gamma} \vdash \lceil \vec{\Delta} \vdash t \rceil : \lceil \vec{\Delta} \vdash T \rceil} \qquad \frac{\vec{\Gamma} \mid \mathcal{O}(\vec{\sigma}) \vdash t : \lceil \vec{\Delta} \vdash T \rceil \qquad \vec{\sigma} : \vec{\Gamma} \Rightarrow_s \vec{\Delta}}{\vec{\Gamma} \vdash \lfloor t \rfloor_{\vec{\sigma}} : T}$$

- `unbox` offsets are insufficient for elimination

Contextual Types

- $S4$ corresponds to meta-programming (Davies and Pfenning, 2001)
  - but $\square$ only handles closed code!
- Contextual types (Nanevski et al., 2008) also handle open code, but only available in dual-context style
- Contextual types in Kripke style, allowing open code w.r.t. context stacks:

$$\frac{\vec{\Gamma}; \vec{\Delta} \vdash t : T}{\vec{\Gamma} \vdash \lceil \vec{\Delta} \vdash t \rceil : \lceil \vec{\Delta} \vdash T \rceil} \qquad \frac{\vec{\Gamma} \mid \mathcal{O}(\vec{\sigma}) \vdash t : \lceil \vec{\Delta} \vdash T \rceil \qquad \vec{\sigma} : \vec{\Gamma} \Rightarrow_s \vec{\Delta}}{\vec{\Gamma} \vdash \lfloor t \rfloor_{\vec{\sigma}} : T}$$

- `unbox` offsets are insufficient for elimination
  - need *semi-substitutions* ($\vec{\sigma}$)

Contributions (Again)

- Unified substitutions; enabling a substitution calculus for Kripke-style modal systems
- A unified normalization proof for all modal systems K, T, $K4$ and $S4$
- A formulation of contextual types in Kripke style; a foundation of meta-programming with open code

Altenkirch, T., Hofmann, M., and Streicher, T. (1995). Categorical reconstruction of a reduction free normalization proof. In Pitt, D., Rydeheard, D. E., and Johnstone, P., editors, *Category Theory and Computer Science*, pages 182–199, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bellin, G., de Paiva, V. C. V., and Ritter, E. (2001). Extended Curry-Howard Correspondence for a Basic Constructive Modal Logic. In *In Proceedings of Methods for Modalities*.

Bierman, G. M. and de Paiva, V. (2000). On an Intuitionistic Modal Logic. *Studia Logica*, 65(3):383–416.

Bierman, G. M. and de Paiva, V. C. V. (1996). Intuitionistic Necessity Revisited. Technical report, University of Birmingham.

Borghuis, V. A. J. (1994). *Coming to terms with modal logic : on the interpretation of modalities in typed lambda-calculus*. PhD Thesis, Mathematics and Computer Science.

Clouston, R. (2018). Fitch-Style Modal Lambda Calculi. In Baier, C. and Dal Lago, U., editors, *Foundations of Software Science and Computation Structures*, Lecture Notes in Computer Science, pages 258–275, Cham. Springer International Publishing.

Davies, R. and Pfenning, F. (2001). A modal analysis of staged computation. *Journal of the ACM*, 48(3):555–604.

Gratzer, D., Kavvos, G. A., Nuyts, A., and Birkedal, L. (2020). Multimodal Dependent Type Theory. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 492–506, New York, NY, USA. Association for Computing Machinery.

Gratzer, D., Sterling, J., and Birkedal, L. (2019). Implementing a modal dependent type theory. *Proceedings of the ACM on Programming Languages*, 3(ICFP):107:1–107:29.

Jang, J., Gélineau, S., Monnier, S., and Pientka, B. (2022). Moebius: Metaprogramming using contextual types – the stage where system f can pattern match on itself. *Proc. ACM Program. Lang. (PACMPL)*, (POPL).

Kavvos, G. A. (2017). Dual-context calculi for modal logic. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12.

Licata, D. R., Orton, I., Pitts, A. M., and Spitters, B. (2018). Internal Universes in Models of Homotopy Type Theory. In Kirchner, H., editor, *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, volume 108 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:17, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISSN: 1868-8969.

Nanevski, A., Pfenning, F., and Pientka, B. (2008). Contextual modal type theory. *ACM Transactions on Computational Logic*, 9(3):23:1–23:49.

Pfenning, F. and Davies, R. (2001). A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11(04).

Pfenning, F. and Wong, H.-C. (1995). On a Modal λ-Calculus for S4. *Electronic Notes in Theoretical Computer Science*, 1:515–534.

Pientka, B., Abel, A., Ferreira, F., Thibodeau, D., and Zucchini, R. (2019). A type theory for defining logics and proofs. In *34th IEEE/ ACM Symposium on Logic in Computer Science (LICS'19)*, pages 1–13. IEEE Computer Society.

Shulman, M. (2018). Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science*, 28(6):856–941. Publisher: Cambridge University Press.

Zyuzin, N. and Nanevski, A. (2021). Contextual modal types for algebraic effects and handlers. *Proc. ACM Program. Lang.*, 5(ICFP):1–29.