On the Semantic Expressiveness of Recursive Types

Marco Patrignani[1,2] Eric M. Martin[1] Dominique Devriese[3]

19th November 2020

1 Stanford University 2 CISPA Helmholtz Center for Information Security 3 VUB Vrije Universiteit Brussel

What and Why?

What is the relative semantic expressiveness of **iso-** and *equi*-recursive types?

What and Why?

What is the relative semantic expressiveness of iso- and *equi*-recursive types?

- open question

What and Why?

What is the relative semantic expressiveness of **iso-** and *equi*-recursive types?

- open question
- clarifies the design of emerging languages

What and Why?

What is the relative semantic expressiveness of iso- and *equi*-recursive types?

- open question
- clarifies the design of emerging languages
- better understanding of how to answer language expressiveness questions

Contributions

- prove that **iso-recursive** and ($\textit{coinductive}$) $\textit{equi-recursive}$ types have the same semantic expressiveness

Contributions

- prove that **iso-recursive** and (*coinductive*) *equi-recursive* types have the same semantic expressiveness and the same as term-level recursion (fix)

Contributions

- prove that **iso-recursive** and *(coinductive) equi-recursive* types have the same semantic expressiveness and the same as term-level recursion (fix) (via fully-abstract compilation (FAC) proofs)

Contributions

- prove that **iso-recursive** and *(coinductive) equi-recursive* types have the same semantic expressiveness and the same as term-level recursion (fix) (via fully-abstract compilation (FAC) proofs)
- devise a new proof technique for FAC

Contributions

- prove that **iso-recursive** and (*coinductive*) *equi-recursive* types have the same semantic expressiveness and the same as term-level recursion (fix) (via fully-abstract compilation (FAC) proofs)
- devise a new proof technique for FAC based on approximate cross-**lan**_guage_ LR

Talk Outline

Recursive Types

Comparing Language Expressiveness

Iso is Co-Equi (is Fix)

Recursive Types

Recursive Types (in STLC)

$$\tau ::= \cdots \mid \mu\alpha.\, \tau \mid \alpha$$

Recursive Types (in STLC)

$$\tau ::= \cdots \mid \mu\alpha.\, \tau \mid \alpha$$

$$LN \stackrel{\text{def}}{=} \mu\alpha.\, Unit \uplus (Nat \times \alpha)$$

Recursive Types (in STLC)

$$\tau ::= \cdots \mid \mu\alpha.\, \tau \mid \alpha$$

$$LN \stackrel{\text{def}}{=} \mu\alpha.\, Unit \uplus (Nat \times \alpha)$$

Q: How are these types related:

$$LN \qquad Unit \uplus (Nat \times LN)$$

$$Unit \uplus (Nat \times Unit \uplus (Nat \times LN))$$

Recursive Types (in STLC)

$$\tau ::= \cdots \mid \mu\alpha.\,\tau \mid \alpha$$

$$LN \overset{\text{def}}{=} \mu\alpha.\,Unit \uplus (Nat \times \alpha)$$

Q: How are these types related:

$$LN \qquad Unit \uplus (Nat \times LN)$$

$$Unit \uplus (Nat \times Unit \uplus (Nat \times LN))$$

A1: they are isomorphic [Gordon *et al.*'79]

Recursive Types (in STLC)

$$\tau ::= \cdots \mid \mu\alpha.\,\tau \mid \alpha$$

$$LN \stackrel{\text{def}}{=} \mu\alpha.\,Unit \uplus (Nat \times \alpha)$$

Q: How are these types related:

$$LN \qquad Unit \uplus (Nat \times LN)$$

$$Unit \uplus (Nat \times Unit \uplus (Nat \times LN))$$

A1: they are **isomorphic** [Gordon *et al.*'79]

A2: they are *equivalent* [Morris '68]

Iso-recursive Types: $\lambda^{\mu}_{\mathrm{I}}$

$$\mathrm{t} ::= \cdots \mid \mathrm{fold}_{\mu\alpha.\tau}\ \mathrm{t} \mid \mathrm{unfold}_{\mu\alpha.\tau}\ \mathrm{t}$$
$$\mathrm{v} ::= \cdots \mid \mathrm{fold}_{\mu\alpha.\tau}\ \mathrm{v}$$

Iso-recursive Types: $\lambda_{\mathrm{I}}^{\mu}$

$$\mathrm{t} ::= \cdots \mid \mathbf{fold}_{\mu\alpha.\tau}\ \mathrm{t} \mid \mathbf{unfold}_{\mu\alpha.\tau}\ \mathrm{t}$$

$$\mathrm{v} ::= \cdots \mid \mathbf{fold}_{\mu\alpha.\tau}\ \mathrm{v}$$

Nil $\qquad [2, 1, Nil]$ $\qquad LN \stackrel{\text{def}}{=} \mu\alpha.\ Unit \uplus (Nat \times \alpha)$

$$\mathbf{fold}_{\mathrm{LN}}\ \mathbf{inl}\ \mathbf{unit}$$

$$\mathbf{fold}_{\mathrm{LN}}\ \mathbf{inr}\ \langle 2, \mathbf{fold}_{\mathrm{LN}}\ \mathbf{inr}\ \langle 1, \mathbf{fold}_{\mathrm{LN}}\ \mathbf{inl}\ \mathbf{unit}\rangle\rangle$$

Iso-recursive Typing & Semantics: $\lambda^{\mu}_{\mathrm{I}}$

($\lambda^{\mu}_{\mathrm{I}}$-Type-fold)

$$\frac{\Gamma \vdash \mathrm{t} : \tau[\mu\alpha.\,\tau/\alpha]}{\Gamma \vdash \mathrm{fold}_{\mu\alpha.\tau}\ \mathrm{t} : \mu\alpha.\,\tau}$$

Iso-recursive Typing & Semantics: $\lambda_{\mathrm{I}}^{\mu}$

($\lambda_{\mathrm{I}}^{\mu}$-Type-fold)

$$\frac{\Gamma \vdash \mathrm{t} : \tau[\mu\alpha.\,\tau/\alpha]}{\Gamma \vdash \mathrm{fold}_{\mu\alpha.\tau}\ \mathrm{t} : \mu\alpha.\,\tau}$$

($\lambda_{\mathrm{I}}^{\mu}$-Type-unfold)

$$\frac{\Gamma \vdash \mathrm{t} : \mu\alpha.\,\tau}{\Gamma \vdash \mathrm{unfold}_{\mu\alpha.\tau}\ \mathrm{t} : \tau[\mu\alpha.\,\tau/\alpha]}$$

Iso-recursive Typing & Semantics: $\lambda^{\mu}_{\mathrm{I}}$

($\lambda^{\mu}_{\mathrm{I}}$-Type-fold)

$$\frac{\Gamma \vdash \mathrm{t} : \tau[\mu\alpha.\,\tau/\alpha]}{\Gamma \vdash \mathrm{fold}_{\mu\alpha.\tau}\ \mathrm{t} : \mu\alpha.\,\tau}$$

($\lambda^{\mu}_{\mathrm{I}}$-Type-unfold)

$$\frac{\Gamma \vdash \mathrm{t} : \mu\alpha.\,\tau}{\Gamma \vdash \mathrm{unfold}_{\mu\alpha.\tau}\ \mathrm{t} : \tau[\mu\alpha.\,\tau/\alpha]}$$

($\lambda^{\mu}_{\mathrm{I}}$-Eval-fold)

$$\frac{}{\mathrm{unfold}_{\mu\alpha.\tau}\ (\mathrm{fold}_{\mu\alpha.\tau}\ \mathrm{v}) \hookrightarrow_{\mathrm{p}} \mathrm{v}}$$

$Equi$-recursive Types: λ^{μ}_{E}

No term-level annotation

$Equi$-recursive Types: λ_E^μ

No term-level annotation

(λ_E^μ-Type-eq ($\doteq$))

$$\frac{\Gamma \vdash t : \mu\alpha.\,\tau \qquad \mu\alpha.\,\tau \doteq \sigma}{\Gamma \vdash t : \sigma}$$

$Equi$-recursive Types: λ^{μ}_{E}

No term-level annotation

$$\frac{\Gamma \vdash t : \mu\alpha.\,\tau \qquad \mu\alpha.\,\tau \doteq \sigma}{\Gamma \vdash t : \sigma}\ (\lambda^{\mu}_{E}\text{-Type-eq } (\doteq))$$

Definitions of $\doteq$:

- inductive — e.g., [Abadi & Fiore '96]
- coinductive — e.g., [Cai *et al.* '16]

  (proven to be strictly stronger than the first one wrt expressing type equality)

$Equi$-recursive Types: λ^{μ}_{E}

No term-level annotation

(λ^{μ}_{E}-Type-eq ($\doteq$))

$$\frac{\Gamma \vdash t : \mu\alpha.\,\tau \qquad \mu\alpha.\,\tau \doteq \sigma}{\Gamma \vdash t : \sigma}$$

Definitions of $\doteq$:

- inductive — e.g., [Abadi & Fiore '96]
- coindductive — e.g., [Cai *et al.* '16]

  (proven to be strictly stronger than the first one wrt expressing type equality)

Coinductive Type Equality: λ^{μ}_{E}

$$(\doteq\text{-prim})\quad \frac{\iota = \mathit{Unit} \lor \mathit{Bool} \lor \alpha}{\iota \doteq \iota}$$

$$(\doteq\text{-bin})\quad \frac{\star \in \{\rightarrow, \times, \uplus\} \quad \tau_1 \doteq \sigma_1 \quad \tau_2 \doteq \sigma_2}{\tau_1 \star \tau_2 \doteq \sigma_1 \star \sigma_2}$$

$$(\doteq\text{-}\mu_l)\quad \frac{\tau[\mu\alpha.\,\tau/\alpha] \doteq \sigma \quad \tau \text{ contractive in } \alpha}{\mu\alpha.\,\tau \doteq \sigma}$$

$$(\doteq\text{-}\mu_r)\quad \frac{\tau \neq \mu\alpha.\,\tau' \quad \tau \doteq \sigma[\mu\alpha.\,\sigma/\alpha] \quad \sigma \text{ contractive in } \alpha}{\tau \doteq \mu\alpha.\,\sigma}$$

Contractiveness: α are used only after a $\star$

Non-contractiveness: (e.g., $\mu\alpha.\,\alpha$) are value-uninhabited

Knowns and Unknowns between $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$ & λ^{μ}_{E}

- typable $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$ $\iff$ typable λ^{μ}_{E} [Abadi & Fiore '96]

Knowns and Unknowns between $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E}

- typable $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ $\iff$ typable λ^{μ}_{E} [Abadi & Fiore '96]
- Q: termination $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ $\iff$ termination λ^{μ}_{E}?
- Q: (generally) how to compare relative semantic expressiveness of languages?

Comparing Language Expressiveness

History (applied to $\boldsymbol{\lambda}_{\mathrm{I}}^{\mu}$ & λ_{E}^{μ})

Felleisen ’91, Mitchell ’93

History (applied to $\lambda^{\mu}_{\mathrm{I}}$ & λ^{μ}_{E})

Felleisen '91, Mitchell '93

- observe the interaction between terms (t) and contexts (C) over an interface ($C\,[t]$)

History (applied to $\lambda^{\mu}_{\mathrm{I}}$ & λ^{μ}_{E})

Felleisen '91, Mitchell '93

- observe the interaction between terms (t) and contexts (C) over an interface ($C[t]$)
- C's language affects interface choice

History (applied to $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E})

Felleisen '91, Mitchell '93

- observe the interaction between terms (t) and contexts (C) over an interface ($C\,[t]$)
- C's language affects interface choice
  e.g., $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E} have rec.-typed interface

History (applied to $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E})

Felleisen '91, Mitchell '93

- observe the interaction between terms (t) and contexts (C) over an interface ($C[t]$)
- C's language affects interface choice e.g., $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E} have rec.-typed interface
- Hp: take the same t in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E}

History (applied to $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E})

Felleisen '91, Mitchell '93

- observe the interaction between terms (t) and contexts (C) over an interface ($C[t]$)
- C's language affects interface choice
  e.g., $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E} have rec.-typed interface
- Hp: take the same t in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E}
- Q: does $\mathbf{C}[t]$ behave differently from $C[t]$?

The same t in $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$ & λ^{μ}_{E}

- Often: compiler (language translator)

$$[\![\cdot]\!] : \mathbf{t} \to t$$

The same t in $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$ & λ^{μ}_{E}

- Often: compiler (language translator)

$$[\![\cdot]\!] : \mathbf{t} \to t$$

- not just <u>any</u> $[\![\cdot]\!]$

The same t in $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$ & λ^{μ}_{E}

- Often: compiler (language translator)

$$[\![\cdot]\!] : \mathbf{t} \to t$$

- not just <u>any</u> $[\![\cdot]\!]$
- use canonical / well-behaved $[\![\cdot]\!]$

or the result is trivial [Parrow '08, Gorla & Nestmann '16]

The same t in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ & λ^{μ}_{E}

- Often: compiler (language translator)

$$[\![\cdot]\!] : \mathbf{t} \to t$$

- not just <u>any</u> $[\![\cdot]\!]$
- use canonical / well-behaved $[\![\cdot]\!]$

  or the result is trivial [Parrow ’08, Gorla & Nestmann ’16]

- our $[\![\cdot]\!]$: identity and erase **fold** / **unfold**

Telling if $\mathbf{C}[\ t\ ]$ behaves differently from $C[\ t\ ]$

- reason about equivalences

Telling if $\mathbf{C}[\,t\,]$ behaves differently from $C[\,t\,]$

- reason about equivalences
- take two programs $\mathbf{t_1}$ and $\mathbf{t_2}$

Telling if $\mathbf{C}[\ t\ ]$ behaves differently from $C[\ t\ ]$

- reason about equivalences
- take two programs $\mathbf{t_1}$ and $\mathbf{t_2}$
    - if they are eq. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$
      (i.e., $\mathbf{C}$ cannot differentiate them semantically)

Telling if $\mathbf{C}[\ t\ ]$ behaves differently from $C[\ t\ ]$

- reason about equivalences
- take two programs $\mathbf{t_1}$ and $\mathbf{t_2}$
  - if they are eq. in $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$
    (i.e., $\mathbf{C}$ cannot differentiate them semantically)
  - they must be eq. in λ^{μ}_{E}
    (C cannot tell $[\![\mathbf{t_1}]\!]$ from $[\![\mathbf{t_2}]\!]$)

Telling if $\mathbf{C}[\ t\ ]$ behaves differently from $C[\ t\ ]$

- reason about equivalences
- take two programs $\mathbf{t_1}$ and $\mathbf{t_2}$
    - if they are eq. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$
      (i.e., $\mathbf{C}$ cannot differentiate them semantically)
    - they must be eq. in λ^{μ}_{E}
      (C cannot tell $[\![\mathbf{t_1}]\!]$ from $[\![\mathbf{t_2}]\!]$)
    - and vice-versa (sanity check on $[\![\cdot]\!]$)

Telling if $\mathbf{C}[\ t\ ]$ behaves differently from $C[\ t\ ]$

- reason about equivalences
- take two programs $\mathbf{t_1}$ and $\mathbf{t_2}$
  - if they are eq. in $\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}$
    ., $\mathbf{C}$ cannot differentiate them semantically)

eq. preservation

  - they must be eq. in λ^{μ}_{E}
    (C cannot tell $[\![\mathbf{t_1}]\!]$ from $[\![\mathbf{t_2}]\!]$)

eq. reflection

  nd vice-versa (sanity check on $[\![\cdot]\!]$)

Telling if $\mathbf{C}[\,t\,]$ behaves differently from $C[\,t\,]$

- reason about equivalences
- take two programs $\mathbf{t_1}$ and $\mathbf{t_2}$
    - if they are eq. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$
      e., $\mathbf{C}$ cannot differentiate them semantically)
    - they must be eq. in λ^{μ}_{E}
      (C cannot tell $[\![\mathbf{t_1}]\!]$ from $[\![\mathbf{t_2}]\!]$)
    - nd vice-versa (sanity check on $[\![\cdot]\!]$)

eq. preservation

eq. reflection

- semantic differentiation: $\Uparrow$ vs $\Downarrow$

Fully Abstract Compilation (FAC) [Abadi (*et al.*) '99]

- Preservation and reflection of contextual equivalence
- $\vdash [\![\cdot]\!] : \mathsf{FAC} \overset{\text{def}}{=} \forall \mathbf{t_1}, \mathbf{t_2}$

  $\mathbf{t_1} \simeq_{\mathbf{ctx}} \mathbf{t_2} \iff [\![\mathbf{t_1}]\!] \simeq_{\mathrm{ctx}} [\![\mathbf{t_2}]\!]$

Fully Abstract Compilation (FAC) [Abadi (*et al.*) '99]

- Preservation and reflection of contextual equivalence
- $\vdash [\![\cdot]\!] : \mathsf{FAC} \overset{\text{def}}{=} \forall \mathbf{t_1}, \mathbf{t_2}$

$$\mathbf{t_1} \simeq_{\mathbf{ctx}} \mathbf{t_2} \iff [\![\mathbf{t_1}]\!] \simeq_{\mathrm{ctx}} [\![\mathbf{t_2}]\!]$$

or:

$$(\forall \mathbf{C}. \mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Updownarrow$$

$$(\forall C. C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Updownarrow$$

$$(\forall C.C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Updownarrow$$

$$(\forall C.C[\llbracket \mathbf{t_1} \rrbracket]\Downarrow \iff C[\llbracket \mathbf{t_2} \rrbracket]\Downarrow)$$

- TH: $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E} are eq. expressive

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Updownarrow$$

$$(\forall C.C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

- TH: $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E} are eq. expressive
- T.S.: $[\![\cdot]\!]$ (well-behaved) $\vdash$ FAC

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Updownarrow$$

$$(\forall C.C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

- TH: $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E} are eq. expressive
- T.S.: $[\![\cdot]\!]$ (well-behaved) $\vdash$ FAC
- take 2 progs. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and the same in λ^{μ}_{E}

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Updownarrow$$

$$(\forall C.C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

- TH: $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E} are eq. expressive
- T.S.: $[\![\cdot]\!]$ (well-behaved) $\vdash$ FAC
- take 2 progs. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and the same in λ^{μ}_{E}
- T.S.: interactions over $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ interfaces reveal as much as interactions over λ^{μ}_{E} ones

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Uparrow \text{simple}$$

$$(\forall C.C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

- TH: $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E} are eq. expressive
- T.S.: $[\![\cdot]\!]$ (well-behaved) $\vdash$ FAC
- take 2 progs. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and the same in λ^{μ}_{E}
- T.S.: interactions over $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ interfaces reveal as much as interactions over λ^{μ}_{E} ones

FAC for Language Expressiveness

$$(\forall \mathbf{C}.\mathbf{C}[\mathbf{t_1}]\Downarrow \iff \mathbf{C}[\mathbf{t_2}]\Downarrow)$$

$$\Downarrow \text{hard}$$

$$(\forall C.C[[\![\mathbf{t_1}]\!]]\Downarrow \iff C[[\![\mathbf{t_2}]\!]]\Downarrow)$$

- TH: $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and λ^{μ}_{E} are eq. expressive
- T.S.: $[\![\cdot]\!]$ (well-behaved) $\vdash$ FAC
- take 2 progs. in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ and the same in λ^{μ}_{E}
- T.S.: interactions over $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$ interfaces reveal as much as interactions over λ^{μ}_{E} ones

Iso is Co-Equi (is Fix)

Preservation via Step-Idx LR

[Devriese *et al.*'16]

$$\mathbf{t}_1 \simeq_{\mathrm{ctx}} \mathbf{t}_2$$

ctx. eq. preservation

$$[\![\mathbf{t}_1]\!] \overset{?}{\simeq}_{\mathrm{ctx}} [\![\mathbf{t}_2]\!]$$

Preservation via Step-Idx LR

[Devriese *et al.*'16]

$$\mathbf{t_1} \simeq_{\mathbf{ctx}} \mathbf{t_2}$$

ctx. eq. preservation

$$C\left[[\![\mathbf{t_1}]\!]\right]\Downarrow_n \quad \overset{?}{\Rightarrow} \quad C\left[[\![\mathbf{t_2}]\!]\right]\Downarrow_{\_}$$

$$[\![\mathbf{t_1}]\!] \overset{?}{\simeq}_{\mathrm{ctx}} [\![\mathbf{t_2}]\!]$$

Preservation via Step-Idx LR

[Devriese *et al.*'16]

$$\mathbf{t_1} \simeq_{\mathbf{ctx}} \mathbf{t_2}$$

$\mathbf{t_1} \gtrsim_{\_} [\![\mathbf{t_1}]\!]$

$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \gtrsim_n C$

(1)

$$C\left[[\![\mathbf{t_1}]\!]\right]\Downarrow_n \quad \overset{?}{\Rightarrow} \quad C\left[[\![\mathbf{t_2}]\!]\right]\Downarrow_{\_}$$

$$[\![\mathbf{t_1}]\!] \overset{?}{\simeq}_{\mathrm{ctx}} [\![\mathbf{t_2}]\!]$$

ctx. eq. preservation

Preservation via Step-Idx LR

[Devriese *et al.*'16]

$$\mathbf{t}_1 \simeq_{\text{ctx}} \mathbf{t}_2$$

$$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}}\left[\mathbf{t}_1\right] \Downarrow_{\_}$$

$$\mathbf{t}_1 \gtrsim_{\_} [\![\mathbf{t}_1]\!]$$

$$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \gtrsim_n C$$

(1)

$$C\left[[\![\mathbf{t}_1]\!]\right] \Downarrow_n \quad \stackrel{?}{\Rightarrow} \quad C\left[[\![\mathbf{t}_2]\!]\right] \Downarrow_{\_}$$

$$[\![\mathbf{t}_1]\!] \stackrel{?}{\simeq}_{\text{ctx}} [\![\mathbf{t}_2]\!]$$

ctx. eq. preservation

Preservation via Step-Idx LR

[Devriese *et al.*'16]

$$\mathbf{t_1} \simeq_{\mathrm{ctx}} \mathbf{t_2}$$

$$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}}\left[\mathbf{t_1}\right]\Downarrow_{\_} \underset{(2)}{\Rightarrow} \langle\!\langle C \rangle\!\rangle_{\mathbf{n}}\left[\mathbf{t_2}\right]\Downarrow_{\_}$$

$$\mathbf{t_1} \gtrsim_{\_} [\![\mathbf{t_1}]\!]$$

$$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \gtrsim_n C$$

(1)

$$C\left[[\![\mathbf{t_1}]\!]\right]\Downarrow_n \overset{?}{\Rightarrow} C\left[[\![\mathbf{t_2}]\!]\right]\Downarrow_{\_}$$

$$[\![\mathbf{t_1}]\!] \overset{?}{\simeq}_{\mathrm{ctx}} [\![\mathbf{t_2}]\!]$$

ctx. eq. preservation

Preservation via Step-Idx LR

[Devriese *et al.*'16]

$$\mathbf{t_1} \simeq_{\mathrm{ctx}} \mathbf{t_2}$$

$$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}}\left[\mathbf{t_1}\right] \Downarrow_{\_} \quad \underset{(2)}{\Rightarrow} \quad \langle\!\langle C \rangle\!\rangle_{\mathbf{n}}\left[\mathbf{t_2}\right] \Downarrow_{\_}$$

$\mathbf{t_1} \gtrsim_{\_} \llbracket \mathbf{t_1} \rrbracket$

$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \gtrsim_n C$

(1) (3)

$\mathbf{t_2} \lesssim_{\_} \llbracket \mathbf{t_2} \rrbracket$

$\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \lesssim_{\_} C$

$$C\left[\llbracket \mathbf{t_1} \rrbracket\right] \Downarrow_n \quad \overset{?}{\Rightarrow} \quad C\left[\llbracket \mathbf{t_2} \rrbracket\right] \Downarrow_{\_}$$

$$\llbracket \mathbf{t_1} \rrbracket \overset{?}{\simeq}_{\mathrm{ctx}} \llbracket \mathbf{t_2} \rrbracket$$

ctx. eq. preservation

Questions

- Q1: what are logical approximations $\lesssim_n$ and $\gtrsim_n$?

Questions

- Q1: what are logical approximations $\lesssim_n$ and $\gtrsim_n$?

- Q2: what is an approximate backtranslation $\langle\!\langle \cdot \rangle\!\rangle_n : C \to \mathbf{C}$?

Questions

- Q1: what are logical approximations $\lesssim_n$ and $\gtrsim_n$? if one terminates, so does the other [Devriese *et al.* '16]

- Q2: what is an approximate backtranslation $\langle\!\langle \cdot \rangle\!\rangle_n : C \to \mathbf{C}$?

The Need to Approximate

- $[\![\cdot]\!]$: $\mathbf{t} \rightarrow t$ is defined on $\mathbf{t}$'s syntax

The Need to Approximate

- $[\![\cdot]\!]$: $\mathbf{t} \to t$ is defined on $\mathbf{t}$'s syntax
- $\langle\!\langle\cdot\rangle\!\rangle : C \to \mathbf{C}$

The Need to Approximate

- $[\![\cdot]\!]$: $\mathbf{t} \to t$ is defined on $\mathbf{t}$'s syntax
- $\langle\!\langle \cdot \rangle\!\rangle : C \to \mathbf{C}$
    - cannot define on C's syntax (how to type $\langle\!\langle C \rangle\!\rangle$?)

The Need to Approximate

- $[\![\cdot]\!]$: $\mathbf{t} \to t$ is defined on $\mathbf{t}$'s syntax
- $\langle\!\langle\cdot\rangle\!\rangle : C \to \mathbf{C}$
  - cannot define on C's syntax (how to type $\langle\!\langle C\rangle\!\rangle$?)
  - define on C's derivation (but $\doteq$ is coinductive)

The Need to Approximate

- $[\![\cdot]\!]$: $\mathbf{t} \to t$ is defined on $\mathbf{t}$'s syntax
- $\langle\!\langle\cdot\rangle\!\rangle : C \to \mathbf{C}$
  - cannot define on C's syntax (how to type $\langle\!\langle C \rangle\!\rangle$?)
  - define on C's derivation (but $\triangleq$ is coinductive)
  - approximate! $\langle\!\langle\cdot\rangle\!\rangle \to \langle\!\langle\cdot\rangle\!\rangle_n$

The Need to Approximate

- $[\![\cdot]\!]$: $\mathbf{t} \to t$ is defined on $\mathbf{t}$'s syntax
- $\langle\!\langle\cdot\rangle\!\rangle : C \to \mathbf{C}$
  - cannot define on C's syntax (how to type $\langle\!\langle C \rangle\!\rangle$?)
  - define on C's derivation (but $\triangleq$ is coinductive)
  - approximate! $\langle\!\langle\cdot\rangle\!\rangle \to \langle\!\langle\cdot\rangle\!\rangle_n$
- Since $C\,[\,[\![\mathbf{t}]\!]\,]\Downarrow_n$, n-unfolding of recursive types in C suffices to replicate $\Downarrow\_$ in $\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}$

Approximate Backtranslation Type

- n not-known statically
- Q: what if we encounter a $n + 1$ unfolding?

Approximate Backtranslation Type

- n not-known statically
- Q: what if we encounter a $n+1$ unfolding?
- A: backtranslate at <u>Backtranslation Type</u>

  NO: $\langle\!\langle \cdot \rangle\!\rangle_n : \tau \rightarrow \boldsymbol{\tau}$ YES: $\langle\!\langle \cdot \rangle\!\rangle_n : \tau \rightarrow \mathbf{BtT}_{\mathbf{n};\boldsymbol{\tau}}$

Approximate Backtranslation Type

- n not-known statically
- Q: what if we encounter a $n+1$ unfolding?
- A: backtranslate at Backtranslation Type

NO: $\langle\!\langle\cdot\rangle\!\rangle_n : \tau \to \boldsymbol{\tau}$ YES: $\langle\!\langle\cdot\rangle\!\rangle_n : \tau \to \mathbf{BtT}_{\mathbf{n};\boldsymbol{\tau}}$

$$\mathbf{BtT}_{\mathbf{0};\tau} \overset{\text{def}}{=} \mathbf{Unit}$$

$$\mathbf{BtT}_{\mathbf{n+1};\tau} \overset{\text{def}}{=} \begin{cases} \mathbf{Unit} \uplus \mathbf{Unit} & \text{if } \tau = \mathit{Unit} \\ (\mathbf{BtT}_{\mathbf{n};\boldsymbol{\tau}} \to \mathbf{BtT}_{\mathbf{n};\boldsymbol{\tau}'}) \uplus \mathbf{Unit} & \text{if } \tau = \tau \to \tau' \\ (\mathbf{BtT}_{\mathbf{n};\boldsymbol{\tau}} \uplus \mathbf{BtT}_{\mathbf{n};\boldsymbol{\tau}'}) \uplus \mathbf{Unit} & \text{if } \tau = \tau \uplus \tau' \\ \mathbf{BtT}_{\mathbf{n+1};\boldsymbol{\tau}'[\boldsymbol{\mu\alpha.\tau}'/\boldsymbol{\alpha}]} \uplus \mathbf{Unit} & \text{if } \tau = \mu\alpha.\,\tau' \end{cases}$$

Backtranslation Example and Relation

$\langle\!\langle unit \rangle\!\rangle_{n>0}$ = ?

$\mathbf{BtT}_{\mathbf{n+1};Unit} = \mathbf{Unit} \uplus \mathbf{Unit}$

Backtranslation Example and Relation

$\langle\!\langle unit \rangle\!\rangle_{n>0}$ = $\mathbf{inl\ unit}$

$\mathbf{BtT}_{\mathbf{n+1};\,Unit} = \mathbf{Unit} \uplus \mathbf{Unit}$

Backtranslation Example and Relation

$\langle\!\langle unit \rangle\!\rangle_{n>0}$ = $\mathbf{inl\ unit}$

$\mathbf{BtT}_{\mathbf{n+1};Unit}$ = $\mathbf{Unit} \uplus \mathbf{Unit}$

Cannot relate using normal LR:

$$\mathcal{V}\left[\!\left[\mathbf{Unit}\right]\!\right] \stackrel{\text{def}}{=} \{(\mathbf{unit}, unit)\}$$

Backtranslation Example and Relation

$\langle\langle unit \rangle\rangle_{n>0} = \textbf{inl unit}$ $\qquad$ $\mathbf{BtT}_{\mathbf{n+1};Unit} = \mathbf{Unit} \uplus \mathbf{Unit}$

Cannot relate using normal LR:

$$\mathcal{V}\,[\![\mathbf{Unit}]\!] \stackrel{\text{def}}{=} \{(\mathbf{unit}, unit)\}$$

Need a special value relation:

$$\mathcal{V}\,[\![\mathbf{BtT}_{\mathbf{n+1};\tau}]\!] \stackrel{\text{def}}{=} \Big\{(\mathbf{v}, v) \mid \text{either } \mathbf{v} = \textbf{inr unit} \\ \text{or } \tau = Unit \text{ and } \exists \mathbf{v}'.\ \mathbf{v} = \textbf{inl } \mathbf{v}' \text{ and} \\ (\mathbf{v}', v) \in \mathcal{V}\,[\![\mathbf{Unit}]\!]\Big\}$$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}}\,[: ?]$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle\, C \,\rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$ e.g.,

  $\overbrace{[\![\boldsymbol{\mu\alpha}.\,\mathbf{Unit} \uplus \boldsymbol{\alpha}]\!]}^{\tau_u} = \mu\alpha.\, \mathit{Unit} \uplus \alpha \neq \mathbf{BtT}_{\mathbf{1};\tau_u} =$

  $(\mathbf{Unit} \uplus \mathbf{Unit}) \uplus \mathbf{Unit}$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$ e.g.,

  $\overbrace{[\![\boldsymbol{\mu\alpha}.\,\mathbf{Unit} \uplus \boldsymbol{\alpha}]\!]}^{\tau_u} = \mu\alpha.\ \mathit{Unit} \uplus \alpha \neq \mathbf{BtT}_{\mathbf{1};\tau_u} =$

  $(\mathbf{Unit} \uplus \mathbf{Unit}) \uplus \mathbf{Unit}$
- $\langle\!\langle [\cdot] \rangle\!\rangle_{\mathbf{n}} = \left[\mathbf{inject}_{\mathbf{n};\tau}\ \cdot\right]$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$ e.g.,

  $\overbrace{[\![\boldsymbol{\mu\alpha}.\,\mathbf{Unit} \uplus \boldsymbol{\alpha}]\!]}^{\tau_u} = \mu\alpha.\ Unit \uplus \alpha \neq \mathbf{BtT}_{\mathbf{1};\tau_u} =$

  $(\mathbf{Unit} \uplus \mathbf{Unit}) \uplus \mathbf{Unit}$
- $\langle\!\langle [\cdot] \rangle\!\rangle_{\mathbf{n}} = \left[\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}}\ \cdot\right]$

  $\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}} : \boldsymbol{\tau} \to \mathbf{BtT}_{\mathbf{n};[\![\tau]\!]}$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$ e.g.,

  $\overbrace{[\![\boldsymbol{\mu\alpha}.\,\mathbf{Unit} \uplus \boldsymbol{\alpha}]\!]}^{\tau_u} = \mu\alpha.\,Unit \uplus \alpha \neq \mathbf{BtT}_{\mathbf{1};\tau_u} =$

  $(\mathbf{Unit} \uplus \mathbf{Unit}) \uplus \mathbf{Unit}$

- $\langle\!\langle [\cdot] \rangle\!\rangle_{\mathbf{n}} = \left[\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}}\ \cdot\right]$

  $\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}} : \boldsymbol{\tau} \to \mathbf{BtT}_{\mathbf{n};[\![\tau]\!]}$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$ e.g.,

$$\overbrace{[\![\boldsymbol{\mu\alpha}.\,\mathbf{Unit} \uplus \boldsymbol{\alpha}]\!]}^{\tau_u} = \mu\alpha.\,Unit \uplus \alpha \neq \mathbf{BtT}_{\mathbf{1};\tau_u} =$$

$$(\mathbf{Unit} \uplus \mathbf{Unit}) \uplus \mathbf{Unit}$$

- $\langle\!\langle [\cdot] \rangle\!\rangle_{\mathbf{n}} = \left[\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}}\ \cdot\right]$

$\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}} : \boldsymbol{\tau} \to \mathbf{BtT}_{\mathbf{n};[\![\tau]\!]}$

Backtranslating Contexts

- Since $\mathbf{t} : \boldsymbol{\tau}$ implies $[\![\mathbf{t}]\!] : \overbrace{[\![\boldsymbol{\tau}]\!]}^{\tau}$
- And $C\,[: \tau]$
- $\langle\!\langle C \rangle\!\rangle_{\mathbf{n}} \left[: \mathbf{BtT}_{\mathbf{n};\tau}\right]$
- Mismatch! $[\![\boldsymbol{\tau}]\!] \neq \mathbf{BtT}_{\mathbf{n};\tau}$ e.g.,

  $\overbrace{[\![\boldsymbol{\mu\alpha}.\,\mathbf{Unit} \uplus \boldsymbol{\alpha}]\!]}^{\tau_u} = \mu\alpha.\,Unit \uplus \alpha \neq \mathbf{BtT}_{\mathbf{1};\tau_u} =$

  $(\mathbf{Unit} \uplus \mathbf{Unit}) \uplus \mathbf{Unit}$
- $\langle\!\langle [\cdot] \rangle\!\rangle_{\mathbf{n}} = \left[\mathbf{inject}_{\mathbf{n};\tau}\ \cdot\right]$

  $\mathbf{inject}_{\mathbf{n};\boldsymbol{\tau}} : \boldsymbol{\tau} \to \mathbf{BtT}_{\mathbf{n};[\![\tau]\!]}$

Our Contributions, Visually

$$\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}} \underset{\langle\!\langle \cdot \rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} n}{\overset{[\![\cdot]\!]^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}}{\rightleftarrows}} \lambda^{\mu}_{E}$$

$[\![\cdot]\!]^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}$ erases $\mathbf{fold}$/$\mathbf{unfold}$, $\langle\!\langle \cdot \rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} n$ is approximate

Our Contributions, Visually

$$
\begin{tikzcd}
 & \lambda^{\mathsf{fx}} \arrow[dl, "\llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}"'] & \\
\boldsymbol{\lambda}^{\mu}_{\mathbf{I}} \arrow[ur, dotted, "\langle\!\langle\cdot\rangle\!\rangle^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}}\, n"'] \arrow[rr, "\llbracket\cdot\rrbracket^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}"] & & \lambda^{\mu}_{E} \arrow[ll, dotted, bend left, "\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}\, n"]
\end{tikzcd}
$$

$\llbracket\cdot\rrbracket^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}$ erases **fold** / **unfold**, $\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}\, n}$ is approximate

$\llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}$ compiles fix into Z-comb, $\langle\!\langle\cdot\rangle\!\rangle^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}\, n}$ is approximate

Our Contributions, Visually

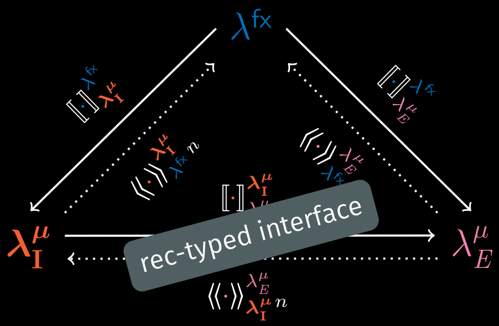
$$
\begin{tikzcd}[column sep=large, row sep=huge]
 & \lambda^{\mathsf{fx}} \arrow[dl, "{\llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}}"', shift right] \arrow[dr, "{\llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\lambda^{\mu}_{E}}}", shift left] & \\
\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}} \arrow[ur, dotted, "{\langle\!\langle\cdot\rangle\!\rangle^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}} n}"', shift right] \arrow[rr, "{\llbracket\cdot\rrbracket^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}}", shift left] & & \lambda^{\mu}_{E} \arrow[ul, dotted, "{\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\lambda^{\mathsf{fx}}} n}", shift left] \arrow[ll, dotted, "{\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} n}", shift left]
\end{tikzcd}
$$

$\llbracket\cdot\rrbracket^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}$ erases **fold**/**unfold**, $\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} n$ is approximate

$\llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}$ compiles fix into Z-comb, $\langle\!\langle\cdot\rangle\!\rangle^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}} n$ is approximate

$\llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\lambda^{\mu}_{E}} = \left\llbracket \llbracket\cdot\rrbracket^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} \right\rrbracket^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}$, $\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\lambda^{\mathsf{fx}}} n = \left\langle\!\!\left\langle \langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} n \right\rangle\!\!\right\rangle^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}} n$

Our Contributions, Visually

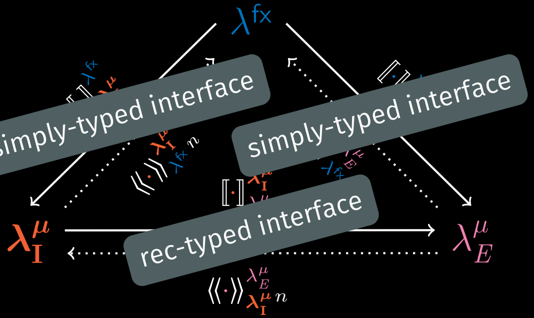


$[\![\cdot]\!]^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}$ erases **fold**/**unfold**, $\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}\, n$ is approximate

$[\![\cdot]\!]^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}$ compiles fix into Z-comb, $\langle\!\langle\cdot\rangle\!\rangle^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}}\, n$ is approximate

$$[\![\cdot]\!]^{\lambda^{\mathsf{fx}}}_{\lambda^{\mu}_{E}} = \left[\!\!\left[[\![\cdot]\!]^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}} \right]\!\!\right]^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mu}_{E}},\ \langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\lambda^{\mathsf{fx}}}\, n = \left\langle\!\!\left\langle \langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}\, n \right\rangle\!\!\right\rangle^{\boldsymbol{\lambda}^{\mu}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}}\, n$$

Our Contributions, Visually



$[\![\cdot]\!]^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}$ erases **fold** / **unfold**, $\langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}\,n}$ is approximate

$[\![\cdot]\!]^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}$ compiles fix into Z-comb, $\langle\!\langle\cdot\rangle\!\rangle^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}\,n}$ is approximate

$$[\![\cdot]\!]^{\lambda^{\mathsf{fx}}}_{\lambda^{\mu}_{E}} = \left[\!\!\left[[\![\cdot]\!]^{\lambda^{\mathsf{fx}}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}} \right]\!\!\right]^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mu}_{E}}, \quad \langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\lambda^{\mathsf{fx}}\,n} = \left\langle\!\!\left\langle \langle\!\langle\cdot\rangle\!\rangle^{\lambda^{\mu}_{E}}_{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}\,n} \right\rangle\!\!\right\rangle^{\boldsymbol{\lambda}^{\boldsymbol{\mu}}_{\mathbf{I}}}_{\lambda^{\mathsf{fx}}\,n}$$

Questions?

System F? F Omega?

- $\forall\alpha.\tau$ & $\exists\alpha.\tau$ types: orthogonal to ours

System F? F Omega?

- $\forall\alpha.\tau$ & $\exists\alpha.\tau$ types: orthogonal to ours
- Conjecture: complex proofs but sem. eq. holds

System F? F Omega?

- $\forall \alpha.\, \tau$ & $\exists \alpha.\, \tau$ types: orthogonal to ours
- Conjecture: complex proofs but sem. eq. holds

- Conjecture: holds for fully applied rec. types: [Kireev *et al.* '19]

$$(\mu\alpha :: K.\, \tau)\tau_1 \cdots \tau_n \quad \text{for } K = K_1 \Rightarrow \cdots K_n \Rightarrow *$$

System F? F Omega?

- $\forall\alpha.\,\tau$ & $\exists\alpha.\,\tau$ types: orthogonal to ours
- Conjecture: complex proofs but sem. eq. holds

- Conjecture: holds for <u>fully applied</u> rec. types: [Kireev *et al.* '19]
  $(\mu\alpha :: K.\,\tau)\tau_1 \cdots \tau_n$ for $K = K_1 \Rightarrow \cdots K_n \Rightarrow *$
- Arbitrary kinds: unknown