# Realizability Interpretation and Normalization of Typed Call-by-Need $\lambda$-calculus with Control

Étienne Miquey[1] & Hugo Herbelin[2]

[1]Équipe Gallinette, INRIA, LS2N, Université de Nantes
[2]Équipe $\pi r^2$, INRIA, IRIF, Université Paris-Diderot

FoSSaCS 2018

# Call-by-need

(*a.k.a.* short reminder from Delia's talk)

**Commercial speach**:

*It combines the best of call-by-name and call-by-value !*

**What they don't tell you**:

*Reasoning about it is much harder...*

**Thumb rules**:

- computations triggered *on demand*
- *memoization* of the results

# Classical logic and control operators

**Classical logic:**

Intuitionistic logic + $A \vee \neg A$

*(or $\neg\neg A \to A$, $((A \to B) \to A) \to A$, etc.*

**Classical Curry-Howard:**

$\lambda$-calculus + `call/cc`

*(Griffin'90:* `call/cc` $: \forall AB.((A \to B) \to A) \to A$*)*

**Continuation-passing style translation:**

- operational semantics for `call/cc`
- Gödel's negative translation

# Classical call-by-need

```
let a = call/cc (λk.(I, λx.throw k x))
    f = fst a
    q = snd a
in f q (I, I)
```

How should a call-by-need strategy compute?

# Classical call-by-need

```
let a = call/cc (λk.(I,λx.throw k x))
    f = fst a
    q = snd a
in f q (I,I)
```

How should a call-by-need strategy compute?

- Okasaki, Lee, Tarditi'94:

  *Only the chain of bindings forcing an effect are not shared.*

```
let a = (I,λx.throw k x)
    f = I
    q = λx.throw k x
in q (I,I)
```

$\longrightarrow$

```
let a = (I,I)
    f = fst a
    q = λx.throw k x
in f q (I,I)
```

$\rightarrow$ *loops forever...*

# Classical call-by-need

```
let a = call/cc (λk.(I,λx.throw k x))
    f = fst a
    q = snd a
in f q (I,I)
```

How should a call-by-need strategy compute?

- Ariola *et al.*'12:

*None of the bindings inside a side-effect are shared.*

```
let a = (I,λx.throw k x)
    f = I
    q = λx.throw k x
in throw k (I,I)
```

$\longrightarrow$

```
let a = (I,I)
    f = fst a
    q = snd a
in f q (I,I)
```

$\rightarrow (\mathrm{I},\mathrm{I})$

# This talk

**Ariola *et al.*'12**:

- defined a sequent calculus call-by-need
- used Danvy's semantics artifacts to derive a CPS

**Main question:**

Do simply-typed terms normalize?

↬ Proof by means of a realizability interpretation

**Underlying motivation:**

↬ *Normalization of Herbelin dPA$^\omega$?*

*(classical arithmetic with DC, using control and lazily evaluated coinductive objects)*

# This talk

**Ariola *et al.*'12**:

- defined a sequent calculus call-by-need
- used Danvy's semantics artifacts to derive a CPS

**Main question:**

Do simply-typed terms normalize?

↬ Proof by means of a realizability interpretation

**Underlying motivation:**

↬ *Normalization of Herbelin dPA$^\omega$ ?*

*(classical arithmetic with DC, using control and lazily evaluated coinductive objects)*

# This talk

**Ariola *et al.*'12**:

- defined a sequent calculus call-by-need
- used Danvy's semantics artifacts to derive a CPS

**Main question:**

Do simply-typed terms normalize?

↬ Proof by means of a realizability interpretation

**Underlying motivation:**

↬ *Normalization of Herbelin dPA$^\omega$?*

*(classical arithmetic with DC, using control and lazily evaluated coinductive objects)*

# This talk

**Ariola *et al.*'12**:

- defined a sequent calculus call-by-need
- used Danvy's semantics artifacts to derive a CPS

**Main question:**

Do simply-typed terms normalize?

↬ Proof by means of a realizability interpretation

**Underlying motivation:**

↬ *Normalization of Herbelin dPA$^\omega$?*

*(classical arithmetic with DC, using control and lazily evaluated coinductive objects)*

# Danvy's semantic artifacts

# CPS translation

**Continuation-passing style translation:** $[\![\cdot]\!] : \mathit{source} \to \lambda^{\text{something}}$

- preserving reduction

$$t \xrightarrow{1} t' \qquad \Rightarrow \qquad [\![t]\!] \xrightarrow{+} [\![t']\!]$$

- preserving typing

$$\Gamma \vdash t : A \qquad \Rightarrow \qquad [\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!]$$

- the type $[\![\bot]\!]$ is not inhabited

## Benefits

If $\lambda^{\text{something}}$ is sound and normalizing:

1. If $[\![t]\!]$ normalizes, then $t$ normalizes
2. If $t$ is typed, then $t$ normalizes
3. The source language is sound, *i.e.* there is no term $\vdash t : \bot$

# CPS translation

**Continuation-passing style translation:** $[\![\cdot]\!] : \mathit{source} \to \lambda^{\text{something}}$

- preserving reduction
- preserving typing
- the type $[\![\bot]\!]$ is not inhabited

## Benefits

If $\lambda^{\text{something}}$ is sound and normalizing:

1. If $[\![t]\!]$ normalizes, then $t$ normalizes
2. If $t$ is typed, then $t$ normalizes
3. The source language is sound, *i.e.* there is no term $\vdash t : \bot$

**Danvy's methodology**

1. an operational semantics
2. a small-step calculus or abstract machine
3. a continuation-passing style translation
4. a realizability model

*Defunctionalized Interpreters for Call-by-Need Evaluation*
Danvy et al. (2010)

# The $\lambda\mu\tilde\mu$-calculus

*The duality of computation*
Curien/Herbelin (2000)

**Syntax:**

$$\begin{array}{ll}
\text{(Terms )} & t ::= x \mid \lambda x.t \mid \mu\alpha.c \\
\text{(Contexts)} & e ::= \alpha \mid t \cdot e \mid \tilde\mu x.c \\
\text{(Commands)} & c ::= \langle t \| e \rangle
\end{array}$$

**Typing rules:**

$$\frac{\Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta}{\langle t \| e \rangle : (\Gamma \vdash \Delta)}$$

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A \mid \Delta} \qquad \frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta} \qquad \frac{c : (\Gamma \vdash \Delta, \alpha : A)}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}$$

$$\frac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta} \qquad \frac{\Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid t \cdot e : A \to B \vdash \Delta} \qquad \frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \tilde\mu x.c : A \vdash \Delta}$$

# The $\lambda\mu\tilde{\mu}$-calculus

*The duality of computation*
Curien/Herbelin (2000)

**Syntax:**

$$
\begin{array}{ll}
\text{(Terms )} & t ::= x \mid \lambda x.t \mid \mu\alpha.c \\
\text{(Contexts)} & e ::= \alpha \mid t \cdot e \mid \tilde{\mu}x.c \\
\text{(Commands)} & c ::= \langle t \parallel e \rangle
\end{array}
$$

**Typing rules:**

$$\frac{\Gamma \vdash A \mid \Delta \qquad \Gamma \mid A \vdash \Delta}{(\Gamma \vdash \Delta)}$$

$$\frac{A \in \Gamma}{\Gamma \vdash A \mid \Delta} \qquad \frac{\Gamma, A \vdash B \mid \Delta}{\Gamma \vdash A \to B \mid \Delta} \qquad \frac{\Gamma \vdash \Delta, A}{\Gamma \vdash A \mid \Delta}$$

$$\frac{A \in \Delta}{\Gamma \mid A \vdash \Delta} \qquad \frac{\Gamma \vdash A \mid \Delta \qquad \Gamma \mid B \vdash \Delta}{\Gamma \mid A \to B \vdash \Delta} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \mid A \vdash \Delta}$$

# Call-by-value $\lambda\mu\tilde{\mu}$-calculus

**Syntax:**

$$
\begin{array}{llll}
\text{(Terms)} & t ::= V \mid \mu\alpha.c & \text{(Contexts)} & e ::= E \mid \tilde{\mu}x.c \\
\text{(Values)} & V ::= x \mid \lambda x.t & \text{(Co-values)} & E ::= \alpha \mid t \cdot e
\end{array}
$$

$$\text{(Commands)} \quad c ::= \langle t \,\|\, e \rangle$$

**Reduction rules:**

$$
\begin{array}{rcl}
\langle \mu\alpha.c \,\|\, e \rangle & \rightarrow & c[e/\alpha] \\
\langle V \,\|\, \tilde{\mu}x.c \rangle & \rightarrow & c[V/x] \\
\langle \lambda x.t \,\|\, u \cdot e \rangle & \rightarrow & \langle u \,\|\, \tilde{\mu}x.\langle t \,\|\, e \rangle \rangle
\end{array}
$$

# Semantic artifacts

$$
\begin{array}{llll}
\text{(Terms)} & t ::= V \mid \mu\alpha.c & \text{(Contexts)} & e ::= E \mid \tilde{\mu}x.c \\
\text{(Values)} & V ::= x \mid \lambda x.t & \text{(Co-values)} & E ::= \alpha \mid t \cdot e
\end{array}
$$

$$\text{(Commands)} \quad c ::= \langle t \| e \rangle$$

**Small steps**

$$
\begin{array}{llcl}
t & \langle \mu\alpha.c \| e \rangle_t & \rightsquigarrow & c_t[e/\alpha] \\
 & \langle V \| e \rangle_t & \rightsquigarrow & \langle V \| e \rangle_e \\
e & \langle V \| \tilde{\mu}x.c \rangle_e & \rightsquigarrow & c_e[V/x] \\
 & \langle V \| u \cdot e \rangle_e & \rightsquigarrow & \langle V \| u \cdot e \rangle_V \\
V & \langle \lambda x.t \| u \cdot e \rangle_V & \rightsquigarrow & \langle u \| \tilde{\mu}x.\langle t \| e \rangle \rangle_t
\end{array}
$$

# Semantic artifacts

$$
\begin{array}{llll}
\text{(Terms)} & t ::= V \mid \mu\alpha.c & \text{(Contexts)} & e ::= E \mid \tilde{\mu}x.c \\
\text{(Values)} & V ::= x \mid \lambda x.t & \text{(Co-values)} & E ::= \alpha \mid t \cdot e
\end{array}
$$

$$\text{(Commands)} \quad c ::= \langle t \parallel e \rangle$$

**Small steps** | **CPS**

$$
\begin{array}{llll}
t: & \langle \mu\alpha.c \parallel e\rangle_t \rightsquigarrow c_t[e/\alpha] & & [\![\mu\alpha.c]\!]_t \triangleq \lambda e.(\lambda\alpha.[\![c]\!]_c)\, e \\
 & \langle V \parallel e\rangle_t \rightsquigarrow \langle V \parallel e\rangle_e & & [\![V]\!]_t \triangleq \lambda e.e\,[\![V]\!]_V \\
e: & \langle V \parallel \tilde{\mu}x.c\rangle_e \rightsquigarrow c_e[V/x] & & [\![\tilde{\mu}x.c]\!]_e \triangleq \lambda V.(\lambda x.[\![c]\!]_c)\, V \\
 & \langle V \parallel u\cdot e\rangle_e \rightsquigarrow \langle V \parallel u\cdot e\rangle_V & & [\![u\cdot e]\!]_e \triangleq \lambda V.V\,[\![u]\!]_t\,[\![e]\!]_e \\
V: & \langle \lambda x.t \parallel u\cdot e\rangle_V \rightsquigarrow \langle u \parallel \tilde{\mu}x.\langle t \parallel e\rangle\rangle_t & & [\![\lambda x.t]\!]_V \triangleq \lambda ue.u\,(\lambda x.[\![t]\!]_t\, e)
\end{array}
$$

$$c \overset{1}{\rightsquigarrow} c' \quad \Rightarrow \quad [\![c]\!]_c \overset{+}{\rightarrow}_\beta [\![c']\!]_c$$

# Semantic artifacts

$$\begin{array}{llll}
\text{(Terms)} & t ::= V \mid \mu\alpha.c & \text{(Contexts)} & e ::= E \mid \tilde{\mu}x.c \\
\text{(Values)} & V ::= x \mid \lambda x.t & \text{(Co-values)} & E ::= \alpha \mid t \cdot e
\end{array}$$

$$\text{(Commands)} \quad c ::= \langle t \| e \rangle$$

**CPS**

**Types translation**

$t$

$$[\![\mu\alpha.c]\!]_t \triangleq \lambda e.(\lambda\alpha.[\![c]\!]_c)\, e$$

$$[\![V]\!]_t \triangleq \lambda e.e\, [\![V]\!]_V$$

$$[\![A]\!]_t \triangleq [\![A]\!]_e \to \bot$$

$e$

$$[\![\tilde{\mu}x.c]\!]_e \triangleq \lambda V.(\lambda x.[\![c]\!]_c)\, V$$

$$[\![u \cdot e]\!]_e \triangleq \lambda V.V\, [\![u]\!]_t\, [\![e]\!]_e$$

$$[\![A]\!]_e \triangleq [\![A]\!]_V \to \bot$$

$V$

$$[\![\lambda x.t]\!]_V \triangleq \lambda ue.u\,(\lambda x.[\![t]\!]_t\, e)$$

$$[\![A \to B]\!]_V \triangleq [\![A]\!]_t \to [\![A]\!]_e \to \bot$$

$$\Gamma \vdash t : A \mid \Delta \quad \Rightarrow \quad [\![\Gamma]\!]_V, [\![\Delta]\!]_e \vdash [\![t]\!]_t : [\![A]\!]_t$$

# Krivine realizability

# Realizability *à la* Krivine

## Intuition

- falsity value $\|A\|$: contexts, opponent to $A$
- truth value $|A|$ : terms, player of $A$
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \,\|\, e\rangle > c_0 > \cdots > c_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subseteq \Lambda \star \Pi$ closed by anti-reduction

Truth value defined by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall e \in \|A\|, \langle t \,\|\, e\rangle \in \perp\!\!\!\perp\}$$

# Realizability *à la* Krivine

## Intuition

- falsity value $\|A\|$: contexts, opponent to $A$
- truth value $|A|$ : terms, player of $A$
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \| e \rangle > c_0 > \cdots > c_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subseteq \Lambda \star \Pi$ closed by anti-reduction

Truth value defined by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall e \in \|A\|, \langle t \| e \rangle \in \perp\!\!\!\perp\}$$

# Realizability *à la* Krivine

## Intuition

- falsity value $\|A\|$: contexts, opponent to $A$
- truth value $|A|$ : terms, player of $A$
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \| e \rangle \succ c_0 \succ \cdots \succ c_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subseteq \Lambda \star \Pi$ closed by anti-reduction

Truth value defined by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall e \in \|A\|, \langle t \| e \rangle \in \perp\!\!\!\perp\}$$

# Realizability *à la* Krivine

## Intuition

- falsity value $\|A\|$: contexts, opponent to $A$
- truth value $|A|$ : terms, player of $A$
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \| e \rangle \succ c_0 \succ \cdots \succ c_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subseteq \Lambda \star \Pi$ closed by anti-reduction

Truth value defined by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall e \in \|A\|, \langle t \| e \rangle \in \perp\!\!\!\perp\}$$

# Realizability *à la* Krivine

## Intuition

- falsity value $\|A\|$: contexts, opponent to $A$
- truth value $|A|$ : terms, player of $A$
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \| e \rangle \succ c_0 \succ \cdots \succ c_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subseteq \Lambda \star \Pi$ closed by anti-reduction

Truth value defined by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall e \in \|A\|, \langle t \| e \rangle \in \perp\!\!\!\perp\}$$

# Realizability *à la* Krivine

## Intuition

- falsity value $\|A\|$: contexts, opponent to $A$
- truth value $|A|$ : terms, player of $A$
- pole $\perp\!\!\!\perp$: commands, referee

$$\langle t \| e \rangle \succ c_0 \succ \cdots \succ c_n \in \perp\!\!\!\perp ?$$

$\rightsquigarrow \perp\!\!\!\perp \subseteq \Lambda \star \Pi$ closed by anti-reduction

Truth value defined by **orthogonality** :

$$|A| = \|A\|^{\perp\!\!\!\perp} = \{t \in \Lambda : \forall e \in \|A\|, \langle t \| e \rangle \in \perp\!\!\!\perp\}$$

# Semantic artifacts++

| | | | |
|---|---|---|---|
| (Terms) | $t ::= \mu\alpha.c \mid x \mid V$ | (Contexts) | $e ::= \tilde{\mu}x.c \mid E$ |
| (Values) | $V ::= \lambda x.t$ | (Co-values) | $E ::= \alpha \mid u \cdot e$ |

**Small steps**

$$\langle \mu\alpha.c \,\|\, e\rangle_t \rightsquigarrow c_t[e/\alpha]$$
$$\langle V \,\|\, e\rangle_t \rightsquigarrow \langle V \,\|\, e\rangle_e$$
$$\langle V \,\|\, \tilde{\mu}x.c\rangle_e \rightsquigarrow c_e[V/x]$$
$$\langle V \,\|\, u \cdot e\rangle_e \rightsquigarrow \langle V \,\|\, u \cdot e\rangle_V$$
$$\langle \lambda x.t \,\|\, u \cdot e\rangle_V \rightsquigarrow \langle u \,\|\, \tilde{\mu}x.\langle t \,\|\, e\rangle\rangle_t$$

**Realizability**

$$|A|_t \triangleq \|A\|_E^{\perp\!\!\perp}$$
$$\|A\|_e \triangleq |A|_V^{\perp\!\!\perp}$$
$$|A \to B|_V \triangleq \{\lambda x.t : \forall V \in |A|_V, t[V/x] \in |B|_t\}$$

# Adequacy

**Substitution** $\sigma$:

$$\sigma ::= \varepsilon \mid \sigma, x := V \mid \sigma, \alpha := e$$

$$\sigma \Vdash \Gamma \cup \Delta \quad \triangleq \quad \begin{cases} \sigma(x) \in |A|_V & \forall (x : A) \in \Gamma \\ \sigma(\alpha) \in \|A\|_e & \forall (\alpha : A) \in \Delta \end{cases}$$

## Adequacy

*For any pole* $\perp\!\!\!\perp$, if $\sigma \Vdash \Gamma \cup \Delta$, then:

1. $\Gamma \vdash t : A \mid \Delta \quad \Rightarrow \quad t[\sigma] \in |A|_t$
2. $\Gamma \mid e : A \vdash \Delta \quad \Rightarrow \quad e[\sigma] \in \|A\|_e$
3. $c : (\Gamma \vdash \Delta) \quad \Rightarrow \quad c[\sigma] \in \perp\!\!\!\perp$

*Proof.* *By mutual induction over the typing derivation.* □

# Results

## Normalizing commands

$\bot\!\!\!\bot_{\Downarrow} \triangleq \{c : c \text{ normalizes}\}$ defines a valid pole.

*Proof.* If $c \to c'$ and $c'$ normalizes, so does $c$. □

## Normalization

For any command $c$, if $c : \Gamma \vdash \Delta$, then $c$ normalizes.

*Proof.* By adequacy, any typed command $c$ belongs to the pole $\bot\!\!\!\bot_{\Downarrow}$. □

## Soundness

There is no term $t$ such that $\vdash t : \bot \mid$.

*Proof.* Otherwise, $t \in |\bot|_t = \Pi^{\bot\!\!\!\bot}$ for any pole, absurd ($\bot\!\!\!\bot \triangleq \emptyset$). □

## Normalization of classical call-by-need

# Classical call-by-need

*Classical Call-by-Need Sequent Calculi: ...*
Ariola *et al.* (2012)

**The $\overline{\lambda}_{[lv\tau\star]}$-calculus:**

- a sequent calculus with explicit "stores"
- semantics artifacts:
  1. small-step reduction rules
  2. (untyped) CPS

**Questions:**

↬ Does it normalize?

↬ Can we define a realizability interpretation?

↬ Can the CPS be typed?

# The $\overline{\lambda}_{[lv\tau\star]}$-calculus

## Syntax:

$$
\begin{array}{rlcl}
\text{(Terms)} & t ::= V \mid \mu\alpha.c & e ::= E \mid \tilde{\mu}x.c & \text{(Contexts)} \\
\text{(Weak values)} & V ::= v \mid x & E ::= \alpha \mid F \mid \tilde{\mu}[x].\langle x \| F\rangle\tau & \text{(Catchable contexts)} \\
\text{(Strong values)} & v ::= \lambda x.t \mid \boldsymbol{k} & F ::= t \cdot E \mid \boldsymbol{\kappa} & \text{(Forcing contexts)}
\end{array}
$$

$$
\begin{array}{rl}
\text{(Commands)} & c ::= \langle t \| e\rangle \\
\text{(Closures)} & l ::= c\tau \\
\text{(Store)} & \tau ::= \epsilon \mid \tau[x := t]
\end{array}
$$

## Reduction rules:

$$
\begin{array}{lrcl}
\textit{(Lazy storage)} & \langle t \| \tilde{\mu}x.c\rangle\tau & \to & c\tau[x := t] \\
& \langle \mu\alpha.c \| E\rangle\tau & \to & (c)\tau[\alpha := E] \\
\textit{(Lookup)} & \langle x \| F\rangle\tau[x := \boldsymbol{t}]\tau' & \to & \langle \boldsymbol{t} \| \tilde{\mu}[x].\langle x \| F\rangle\tau'\rangle\tau \\
\textit{(Forced eval.)} & \langle \boldsymbol{V} \| \tilde{\mu}[x].\langle x \| F\rangle\tau'\rangle\tau & \to & \langle \boldsymbol{V} \| F\rangle\tau[x := \boldsymbol{V}]\tau' \\
& \langle \lambda x.t \| u \cdot E\rangle\tau & \to & \langle u \| \tilde{\mu}x.\langle t \| E\rangle\rangle\tau
\end{array}
$$

# Typing rules

**One-sided sequents** for the $\lambda\mu\tilde{\mu}$-calculus plus:

- Sequents indexed by the syntactic categories:

$$\frac{\Gamma \vdash_t V : A}{\Gamma \vdash_V V : A}\ (\uparrow^t) \qquad \frac{\Gamma, x : A \vdash_t t : B}{\Gamma \vdash_V \lambda x.t : A \to B}\ (\to_r)$$

- Stores are typed with typing hypotheses $\Gamma$:

$$\frac{\Gamma, \Gamma' \vdash_c c \qquad \Gamma \vdash_\tau \tau : \Gamma'}{\Gamma \vdash_l c\tau}\ (l) \qquad \frac{}{\Gamma \vdash_\tau \varepsilon : \varepsilon}\ (\varepsilon)$$

$$\frac{\Gamma \vdash_\tau \tau : \Gamma' \quad \Gamma, \Gamma' \vdash_t t : A}{\Gamma \vdash_\tau \tau[x := t] : \Gamma', x : A}\ (\tau_t) \qquad \frac{\Gamma \vdash_\tau \tau : \Gamma' \quad \Gamma, \Gamma' \vdash_E E : A^{\perp\!\!\!\perp}}{\Gamma \vdash_\tau \tau[\alpha := E] : \Gamma', \alpha : A^{\perp\!\!\!\perp}}\ (\tau_E)$$

# Semantic artifacts

**Small steps:**

| | | | |
|---|---|---|---|
| $e$ | $\langle t \,\|\, \tilde\mu x.c\rangle_e \tau$ | $\rightarrow$ | $c_e \tau[x := t]$ |
| | $\langle t \,\|\, E\rangle_e \tau$ | $\rightarrow$ | $\langle t \,\|\, E\rangle_t \tau$ |
| $t$ | $\langle \mu\alpha.c \,\|\, E\rangle_t \tau$ | $\rightarrow$ | $(c_e[E/\alpha])\tau$ |
| | $\langle V \,\|\, E\rangle_t \tau$ | $\rightarrow$ | $\langle V \,\|\, E\rangle_E \tau$ |
| $E$ | $\langle V \,\|\, \tilde\mu[x].\langle x \,\|\, F\rangle\tau'\rangle_E \tau$ | $\rightarrow$ | $\langle V \,\|\, F\rangle_V \tau[x := V]\tau'$ |
| | $\langle V \,\|\, F\rangle_E \tau$ | $\rightarrow$ | $\langle V \,\|\, F\rangle_V \tau$ |
| $V$ | $\langle x \,\|\, F\rangle_V \tau[x := t]\tau'$ | $\rightarrow$ | $\langle t \,\|\, \tilde\mu[x].\langle x \,\|\, F\rangle\tau'\rangle_t \tau$ |
| | $\langle \lambda x.t \,\|\, F\rangle_V \tau$ | $\rightarrow$ | $\langle \lambda x.t \,\|\, F\rangle_F \tau$ |
| $F$ | $\langle \lambda x.t \,\|\, u \cdot E\rangle_F \tau$ | $\rightarrow$ | $\langle u \,\|\, \tilde\mu x.\langle t \,\|\, E\rangle\rangle_e \tau$ |

# Semantic artifacts

**CPS :**

$$
\begin{array}{lrcl}
 & \llbracket \langle t \| e \rangle \tau \rrbracket & := & \llbracket e \rrbracket_e \; \llbracket \tau \rrbracket_\tau \; \llbracket t \rrbracket_t \\
e & \llbracket \tilde{\mu} x.c \rrbracket_e & := & \lambda \tau t. \llbracket c \rrbracket_c \; \tau[x := t] \\
 & \llbracket E \rrbracket_e & := & \lambda \tau t. t \; \tau \; \llbracket E \rrbracket_E \\
t & \llbracket \mu \alpha.c \rrbracket_t & := & \lambda \tau E. (\llbracket c \rrbracket_c \; \tau)[E/\alpha] \\
 & \llbracket V \rrbracket_t & := & \lambda \tau E. E \; \tau \; \llbracket V \rrbracket_\upsilon \\
E & \llbracket \tilde{\mu}[x].\langle x \| F \rangle \tau' \rrbracket_E & := & \lambda \tau V. V \; \tau[x := V]\tau' \; \llbracket F \rrbracket_F \\
 & \llbracket F \rrbracket_E & := & \lambda \tau V. V \; \tau \; \llbracket F \rrbracket_F \\
V & \llbracket x \rrbracket_\upsilon & := & \lambda \tau F. \tau(x) \; \tau \; (\lambda \tau V. V \; \tau[x := V]\tau' \; \llbracket F \rrbracket_F) \\
 & \llbracket \lambda x.t \rrbracket_\upsilon & := & \lambda \tau F. F \; \tau \; (\lambda u \tau E. \llbracket t \rrbracket_t \; \tau[x := u] \; E) \\
F & \llbracket u \cdot E \rrbracket_F & := & \lambda \tau \upsilon. \upsilon \; \llbracket t \rrbracket_t \; \tau \; \llbracket E \rrbracket_E
\end{array}
$$

# Semantic artifacts

**Small-step:**

| | | | |
|---|---|---|---|
| $e$ | $\langle t \,\|\, \tilde{\mu}x.c\rangle_e \tau$ | $\rightarrow$ | $\ldots$ |
| | $\langle t \,\|\, E\rangle_e \tau$ | $\rightarrow$ | $\ldots$ |
| $t$ | $\langle \mu\alpha.c \,\|\, E\rangle_t \tau$ | $\rightarrow$ | $\ldots$ |
| | $\langle V \,\|\, E\rangle_t \tau$ | $\rightarrow$ | $\ldots$ |
| $E$ | $\langle V \,\|\, \tilde{\mu}[x].\langle x \,\|\, F\rangle\tau'\rangle_E \tau$ | $\rightarrow$ | $\ldots$ |
| | $\langle V \,\|\, F\rangle_E \tau$ | $\rightarrow$ | $\ldots$ |
| $V$ | $\langle x \,\|\, F\rangle_V \tau[x := t]\tau'$ | $\rightarrow$ | $\ldots$ |
| | $\langle v \,\|\, F\rangle_V \tau$ | $\rightarrow$ | $\ldots$ |
| $F$ | $\langle v \,\|\, u \cdot E\rangle_F \tau$ | $\rightarrow$ | $\ldots$ |
| $v$ | $\langle \lambda x.t \,\|\, u \cdot E\rangle_v \tau$ | $\rightarrow$ | $\ldots$ |

**Realizability:**

$$(\perp\!\!\!\perp \subseteq \Lambda \times \Pi \times \tau)$$

$$\|A\|_e := \{\ e? \ \in |A|_t{}^{\perp\!\!\!\perp}\}$$

$$|A|_t := \{\ t? \ \in \|A\|_E{}^{\perp\!\!\!\perp}\}$$

$$\|A\|_E := \{\ E? \ \in |A|_V{}^{\perp\!\!\!\perp}\}$$

$$|A|_V := \{\ V? \ \in \|A\|_F{}^{\perp\!\!\!\perp}\}$$

$$\|A\|_F := \{\ F? \ \in |A|_v{}^{\perp\!\!\!\perp}\}$$

$$|A \rightarrow B|_v := \{\ \lambda x.t? \ : \ u? \ \in |A|_t \Rightarrow t[u/x]? \ \in |B|_t\}$$

# Semantic artifacts

**Small-step:**

$e$:
$$\langle t \| \tilde\mu x.c\rangle_e \tau \rightarrow \dots$$
$$\langle t \| E\rangle_e \tau \rightarrow \dots$$

$t$:
$$\langle \mu\alpha.c \| E\rangle_t \tau \rightarrow \dots$$
$$\langle V \| E\rangle_t \tau \rightarrow \dots$$

$E$:
$$\langle V \| \tilde\mu[x].\langle x \| F\rangle\tau'\rangle_E \tau \rightarrow \dots$$
$$\langle V \| F\rangle_E \tau \rightarrow \dots$$

$V$:
$$\langle x \| F\rangle_V \tau[x:=t]\tau' \rightarrow \dots$$
$$\langle v \| F\rangle_V \tau \rightarrow \dots$$

$F$:
$$\langle v \| u\cdot E\rangle_F \tau \rightarrow \dots$$

$v$:
$$\langle \lambda x.t \| u\cdot E\rangle_v \tau \rightarrow \dots$$

**Realizability:**

$$(\perp\!\!\!\perp \subseteq \Lambda \times \Pi \times \tau)$$

$$\|A\|_e := \{(e|\tau) \in |A|_t{}^{\perp\!\!\!\perp}\}$$

$$|A|_t := \{(t|\tau) \in \|A\|_E{}^{\perp\!\!\!\perp}\}$$

$$\|A\|_E := \{(E|\tau) \in |A|_V{}^{\perp\!\!\!\perp}\}$$

$$|A|_V := \{(V|\tau) \in \|A\|_F{}^{\perp\!\!\!\perp}\}$$

$$\|A\|_F := \{(F|\tau) \in |A|_v{}^{\perp\!\!\!\perp}\}$$

$$|A \rightarrow B|_v := \{(\lambda x.t|\tau) : (u|\tau') \in |A|_t \Rightarrow (t|\overline{\tau\tau'}[x:=u]) \in |B|_t\}$$

# Realizability interpretation

A few novelties:

- **Term-in-store** $(t|\tau)$:

$$FV(t) \subseteq \mathrm{dom}(\tau),\ \tau \text{ closed}$$

- **Pole** : set of closures $\perp\!\!\!\perp$ which is:
  - *closed by anti-reduction*:

$$c'\tau' \in \perp\!\!\!\perp \quad \text{and} \quad c\tau \to c'\tau' \quad \text{implies} \quad c\tau \in \perp\!\!\!\perp$$

  - *closed by store extension*:

$$c\tau \in \perp\!\!\!\perp \quad \text{and} \quad \tau \lhd \tau' \quad \text{implies} \quad c\tau' \in \perp\!\!\!\perp$$

- **Orthogonality** :

$$(t|\tau)\perp\!\!\!\perp(e|\tau') \triangleq \quad \tau,\tau' \text{ compatible} \quad \wedge \quad \langle t \,\|\, e\rangle\overline{\tau\tau'} \in \perp\!\!\!\perp.$$

- **Realizers**: definitions derived from the small-step rules!

# Realizability interpretation

A few novelties:

- **Term-in-store** $(t|\tau)$:

$$FV(t) \subseteq \text{dom}(\tau),\ \tau \text{ closed}$$

- **Pole** : set of closures $\perp\!\!\!\perp$ which is:
  - *closed by anti-reduction*:

$$c'\tau' \in \perp\!\!\!\perp \quad \text{and} \quad c\tau \rightarrow c'\tau' \quad \text{implies} \quad c\tau \in \perp\!\!\!\perp$$

  - *closed by store extension*:

$$c\tau \in \perp\!\!\!\perp \quad \text{and} \quad \tau \triangleleft \tau' \quad \text{implies} \quad c\tau' \in \perp\!\!\!\perp$$

- **Orthogonality** :

$$(t|\tau)\perp\!\!\!\perp(e|\tau') \triangleq \quad \tau,\tau' \text{ compatible} \quad \wedge \quad \langle t \,\|\, e\rangle\overline{\tau\tau'} \in \perp\!\!\!\perp.$$

- **Realizers**: definitions derived from the small-step rules!

# Happy end

## Adequacy

For all $\perp\!\!\!\perp$, if $\tau \Vdash \Gamma$ and $\Gamma \vdash_c c$, then $c\tau \in \perp\!\!\!\perp$.

*Proof: By induction on typing derivations.*

## Normalization

If $\vdash_l c\tau$ then $c\tau$ normalizes.

*Proof: The set* $\perp\!\!\!\perp_{\Downarrow} = \{c\tau \in C_0 : \ c\tau \text{ normalizes } \}$ *is a pole.*

# To sum up

**Initial questions:**

- ✓ Does typed terms normalize? *Yes!*
- ✓ Can we define a realizability interpretation? *Yes!*

**Bonus:**

- Scales to 2nd order types for free
- Seems to be a generic method for calculi with memory

***Next episodes:***

- ✓ *Can the CPS be typed? Yes, using Kripke forcing*
- ✓ *Leads to a normalization proof for dLPA$^\omega$* *(coming soon at LICS)*
- ? *Algebraic consequences on the induced realizability model?*

# To sum up

**Initial questions:**

- ✓ Does typed terms normalize? *Yes!*
- ✓ Can we define a realizability interpretation? *Yes!*

**Bonus:**

- Scales to 2nd order types for free
- Seems to be a generic method for calculi with memory

***Next episodes:***

- ✓ *Can the CPS be typed? Yes, using Kripke forcing*
- ✓ *Leads to a normalization proof for dLPA$^{\omega}$* *(coming soon at LICS)*
- ? *Algebraic consequences on the induced realizability model?*

# To sum up

**Initial questions:**

- ✓ Does typed terms normalize? *Yes!*
- ✓ Can we define a realizability interpretation? *Yes!*

**Bonus:**

- Scales to 2nd order types for free
- Seems to be a generic method for calculi with memory

***Next episodes:***

- ✓ *Can the CPS be typed? Yes, using Kripke forcing*
- ✓ *Leads to a normalization proof for dLPA$^{\omega}$* *(coming soon at LICS)*
- ? *Algebraic consequences on the induced realizability model?*

Thanks for your attention.

$$
\begin{array}{lcl}
[\![\Gamma \vdash_e e : A^{\perp\!\!\perp}]\!] & \triangleq & \vdash [\![e]\!]_e : [\![\Gamma]\!]_\Gamma \triangleright_e \iota(A) \\
[\![\Gamma \vdash_t t : A]\!] & \triangleq & \vdash [\![t]\!]_t : [\![\Gamma]\!]_\Gamma \triangleright_t \iota(A) \\
\ldots & & \ldots \quad \ldots \\
[\![\Gamma \vdash_c c]\!] & \triangleq & \vdash [\![c]\!]_c : [\![\Gamma]\!]_\Gamma \triangleright_c \bot \\
[\![\Gamma \vdash_l l]\!] & \triangleq & \vdash [\![l]\!]_l^{|\Gamma|} : [\![\Gamma]\!]_\Gamma \triangleright_c \bot \\
[\![\Gamma \vdash_\tau \tau : \Gamma']\!] & \triangleq & \vdash [\![\tau]\!]_\tau : [\![\Gamma]\!]_\Gamma \triangleright_\tau [\![\Gamma']\!]_\Gamma
\end{array}
$$

---

$$
[\![\varepsilon]\!]_\Gamma \triangleq \varepsilon \qquad [\![\Gamma, x_i : A]\!]_\Gamma \triangleq [\![\Gamma]\!]_\Gamma, \iota(A) \qquad [\![\Gamma, \alpha_i : A^{\perp\!\!\perp}]\!]_\Gamma \triangleq [\![\Gamma]\!]_\Gamma, \iota(A)^{\perp\!\!\perp}
$$

---

$$
\begin{array}{lcl}
\Upsilon \triangleright_c A & \triangleq & \forall Y <: \Upsilon.\, Y \to \bot \\
\Upsilon \triangleright_e A & \triangleq & \forall Y <: \Upsilon.\, Y \to (Y \triangleright_t A) \to \bot \\
\ldots & & \ldots\ldots \\
\Upsilon \triangleright_v A \to B & \triangleq & \forall Y <: \Upsilon.\, Y \to (Y \triangleright_t A) \to (Y \triangleright_E B) \to \bot \\
\Upsilon \triangleright_v X & \triangleq & X
\end{array}
$$