

A sequent calculus with dependent types for classical arithmetic

Étienne MIQUEY

Équipe Gallinette, INRIA
LS2N, Université de Nantes

Workshop Réalisabilité
13 Juin 2018



A constructive proof of dependent choice
compatible with classical logic

Proofs-as-programs

The Curry-Howard correspondence

Mathematics	Computer Science
Proofs	Programs
Propositions	Types
Deduction rules	Typing rules
$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} (\Rightarrow_E)$	$\frac{\Gamma \vdash t : A \rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B} (\rightarrow_E)$

Benefits:

Program your proofs! | *Prove your programs!*

Proofs-as-programs

Limitations	
Mathematics	Computer Science
$A \vee \neg A$	try. . . catch . . .
$\neg\neg A \Rightarrow A$	x := 42
All sets can be well-ordered	random()
Sets that have the same elements are equal	stop
	goto

↪ *We want more !*

Extending Curry-Howard

$$\text{Classical logic} = \text{Intuitionistic logic} + A \vee \neg A$$

1990: Griffin discovered that call/cc can be typed by Peirce's law
(well-known fact: Peirce's law $\Rightarrow A \vee \neg A$)

Classical Curry-Howard:

$$\lambda\text{-calculus} + \text{call/cc}$$

Other examples:

- quote instruction ~ dependent choice
- monotonic memory ~ Cohen's forcing
- ...

The motto

With side-effects come new reasoning principles.

Extending Curry-Howard

Classical logic = Intuitionistic logic + $A \vee \neg A$

1990: Griffin discovered that call/cc can be typed by Peirce’s law
 (well-known fact: Peirce’s law $\Rightarrow A \vee \neg A$)

Classical Curry-Howard:

λ -calculus + call/cc

Other examples:

- quote instruction ~ dependent choice
- monotonic memory ~ Cohen’s forcing
- ...

The motto
With side-effects come new reasoning principles.

Extending Curry-Howard

Classical logic = Intuitionistic logic + $A \vee \neg A$

1990: Griffin discovered that call/cc can be typed by Peirce's law
(well-known fact: Peirce's law $\Rightarrow A \vee \neg A$)

Classical Curry-Howard:

λ -calculus + call/cc

Other examples:

- quote instruction ~ dependent choice
- monotonic memory ~ Cohen's forcing
- ...

The motto

With side-effects come new reasoning principles.

Teaser

The motto

With side-effects come new reasoning principles.

We will use several **computational features**:

- dependent types
- streams
- lazy evaluation
- shared memory

to get a **proof** for the axioms of **dependent and countable choice** that is compatible with **classical logic**.

The axiom of choice

Axiom of Choice:

$$AC : \forall x^A. \exists y^B. P(x, y) \rightarrow \exists f^{A \rightarrow B}. \forall x^A. P(x, f(x))$$

The axiom of choice

Axiom of Choice:

$$AC : \forall x^A. \exists y^B. P(x, y) \rightarrow \exists f^{A \rightarrow B}. \forall x^A. P(x, f(x))$$

$$:= \lambda H. (\lambda x. \mathbf{wit} (H x), \lambda x. \mathbf{prf} (H x))$$

Computational content through **dependent types**:

$$\frac{\Gamma, x : T \vdash t : A}{\Gamma \vdash \lambda x. t : \forall x^T. A} \quad (\forall_I) \qquad \frac{\Gamma \vdash p : A[t/x] \quad \Gamma \vdash t : T}{\Gamma \vdash (t, p) : \exists x^T. A} \quad (\exists_I)$$

$$\frac{\Gamma \vdash p : \exists x^T. A(x)}{\Gamma \vdash \mathbf{wit} p : T} \quad (\mathbf{wit}) \qquad \frac{\Gamma \vdash p : \exists x^T. A(x)}{\Gamma \vdash \mathbf{prf} p : A(\mathbf{wit} p)} \quad (\mathbf{prf})$$

Incompatibility with classical logic

Bad news
dependent sum + classical logic = ☠

Choice:

$$\vdash t : \forall x \in A. \exists y \in B. P(x, y) \rightarrow \exists f \in B^A. \forall x \in A. P(x, f(x))$$

Excluded-middle:

$$\vdash s : \forall x \in X. \exists y \in \{0, 1\}. (U(x) \wedge y = 1) \vee (\neg U(x) \wedge y = 0)$$

Take U undecidable:

$$\vdash ts : \exists f \in \{0, 1\}^X. \forall x \in X. (U(x) \wedge f(x) = 1) \vee (\neg U(x) \wedge f(x) = 0)$$

↪ *i.e. $wit(ts)$ computes the uncomputable...*

Incompatibility with classical logic

Bad news
dependent sum + classical logic = ☠

Choice:

$$\vdash t : \forall x \in A. \exists y \in B. P(x, y) \rightarrow \exists f \in B^A. \forall x \in A. P(x, f(x))$$

Excluded-middle:

$$\vdash s : \forall x \in X. \exists y \in \{0, 1\}. (U(x) \wedge y = 1) \vee (\neg U(x) \wedge y = 0)$$

Take U undecidable:

$$\vdash ts : \exists f \in \{0, 1\}^X. \forall x \in X. (U(x) \wedge f(x) = 1) \vee (\neg U(x) \wedge f(x) = 0)$$

↯ *i.e. $wit(ts)$ computes the uncomputable...*

Incompatibility with classical logic

Bad news
dependent sum + classical logic = ☠

On the degeneracy of Σ-Types in presence of ...
Herbelin (2005)

One can define:

$$H_0 := \text{call/cc}_\alpha(1, \text{throw}_\alpha(0, p)) : \exists x. x = 0$$

and reach a contradiction:

$$(\text{wit } H_0, \text{prf } H_0) \rightarrow \underbrace{(1, p)}_{\exists x. x=0} \quad \text{with } 0=0$$

We need to:

↪ share

↪ restrict dependent types

Incompatibility with classical logic

Bad news
dependent sum + classical logic = ☠

*On the degeneracy of Σ-Types
in presence of ...
Herbelin (2005)*

One can define:

$$H_0 := \text{call/cc}_\alpha(1, \text{throw}_\alpha(0, p)) : \exists x. x = 0$$

and reach a contradiction:

$$(\text{wit } H_0, \text{prf } H_0) \rightarrow \underbrace{(1, p)}_{\exists x. x = 0}$$

We need to:

↪ **share**

↪ **restrict dependent types**

Toward a solution ?

A constructive proof of dependent choice, compatible with ...
Herbelin (2012)

- Restriction to countable choice:

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}}. \exists y^B. P(x, y) \rightarrow \exists f^{\mathbb{N} \rightarrow B}. \forall x^{\mathbb{N}}. P(x, f(x))$$

- Proof:

$$AC := \lambda H. (\lambda n. \text{if } n = 0 \text{ then wit}(H\ 0) \text{ else} \\ \text{if } n = 1 \text{ then wit}(H\ 1) \text{ else } \dots, \\ \lambda n. \text{if } n = 0 \text{ then prf}(H\ 0) \text{ else} \\ \text{if } n = 1 \text{ then prf}(H\ 1) \text{ else } \dots)$$

Toward a solution ?

A constructive proof of dependent choice, compatible with ...
Herbelin (2012)

- Restriction to countable choice:

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}}. \exists y^B. P(x, y) \rightarrow \exists f^{\mathbb{N} \rightarrow B}. \forall x^{\mathbb{N}}. P(x, f(x))$$

- Proof:

$$AC_{\mathbb{N}} := \lambda H. \text{let } H_0 = H \ 0 \text{ in}$$

$$\quad \text{let } H_1 = H \ 1 \text{ in}$$

$$\quad \dots$$

$$\quad (\lambda n. \text{if } n = 0 \text{ then wit } H_0 \text{ else}$$

$$\quad \quad \text{if } n = 1 \text{ then wit } H_1 \text{ else } \dots ,$$

$$\quad \lambda n. \text{if } n = 0 \text{ then prf } H_0 \text{ else}$$

$$\quad \quad \text{if } n = 1 \text{ then prf } H_1 \text{ else } \dots)$$

Toward a solution ?

A constructive proof of dependent choice, compatible with ...
Herbelin (2012)

- Restriction to countable choice:

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}}. \exists y^B. P(x, y) \rightarrow \exists f^{\mathbb{N} \rightarrow B}. \forall x^{\mathbb{N}}. P(x, f(x))$$

- Proof:

$$AC_{\mathbb{N}} := \lambda H. \text{let } H_{\infty} = (H\ 0, H\ 1, \dots, H\ n, \dots) \text{ in} \\ (\lambda n. \text{wit } (\text{nth } n\ H_{\infty}), \lambda n. \text{prf } (\text{nth } n\ H_{\infty}))$$

Toward a solution ?

A constructive proof of dependent choice, compatible with ...
Herbelin (2012)

- Restriction to countable choice:

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}}. \exists y^B. P(x, y) \rightarrow \exists f^{\mathbb{N} \rightarrow B}. \forall x^{\mathbb{N}}. P(x, f(x))$$

- Proof:

$$AC_{\mathbb{N}} := \lambda H. \text{let } H_{\infty} = \text{cofix}_{bn}^0(H \ n, b(S(n))) \text{ in} \\ (\lambda n. \text{wit } (nth \ n \ H_{\infty}), \lambda n. \text{prf } (nth \ n \ H_{\infty}))$$

dPA^ω (Herbelin's recipe)

A proof system:

- **classical:**

$$p, q ::= \dots \mid \text{catch}_\alpha p \mid \text{throw}_\alpha p$$

- with stratified **dependent types** :

- terms: $t, u ::= \dots \mid \text{wit } p$
- formulas: $A, B ::= \dots \mid \forall x^T. A \mid \exists x^T. A \mid \Pi_{(a:A)}. B \mid t = u$
- proofs: $p, q ::= \dots \mid \lambda x. p \mid (t, p) \mid \lambda a. p$

- a **syntactical restriction** of dependencies to NEF proofs

- call-by-value and **sharing**:

$$p, q ::= \dots \mid \text{let } a = q \text{ in } p$$

- with inductive and **coinductive** constructions:

$$p, q ::= \dots \mid \text{fix}_{bn}^t [p_0 \mid p_S] \mid \text{cofix}_{bn}^t p$$

- **lazy evaluation** for the cofix

dPA^ω (Herbelin's recipe)

A proof system:

- **classical:**

$$p, q ::= \dots \mid \text{catch}_\alpha p \mid \text{throw}_\alpha p$$

- with stratified **dependent types** :

- terms: $t, u ::= \dots \mid \text{wit } p$
- formulas: $A, B ::= \dots \mid \forall x^T. A \mid \exists x^T. A \mid \Pi_{(a:A)}. B \mid t = u$
- proofs: $p, q ::= \dots \mid \lambda x. p \mid (t, p) \mid \lambda a. p$

- a **syntactical restriction** of dependencies to NEF proofs

- call-by-value and **sharing**:

$$p, q ::= \dots \mid \text{let } a = q \text{ in } p$$

- with inductive and **coinductive** constructions:

$$p, q ::= \dots \mid \text{fix}_{bn}^t [p_0 \mid p_S] \mid \text{cofix}_{bn}^t p$$

- **lazy evaluation** for the cofix

State of the art

Subject reduction

If $\Gamma \vdash p : A$ and $p \rightarrow q$, then $\Gamma \vdash q : A$.

Normalization

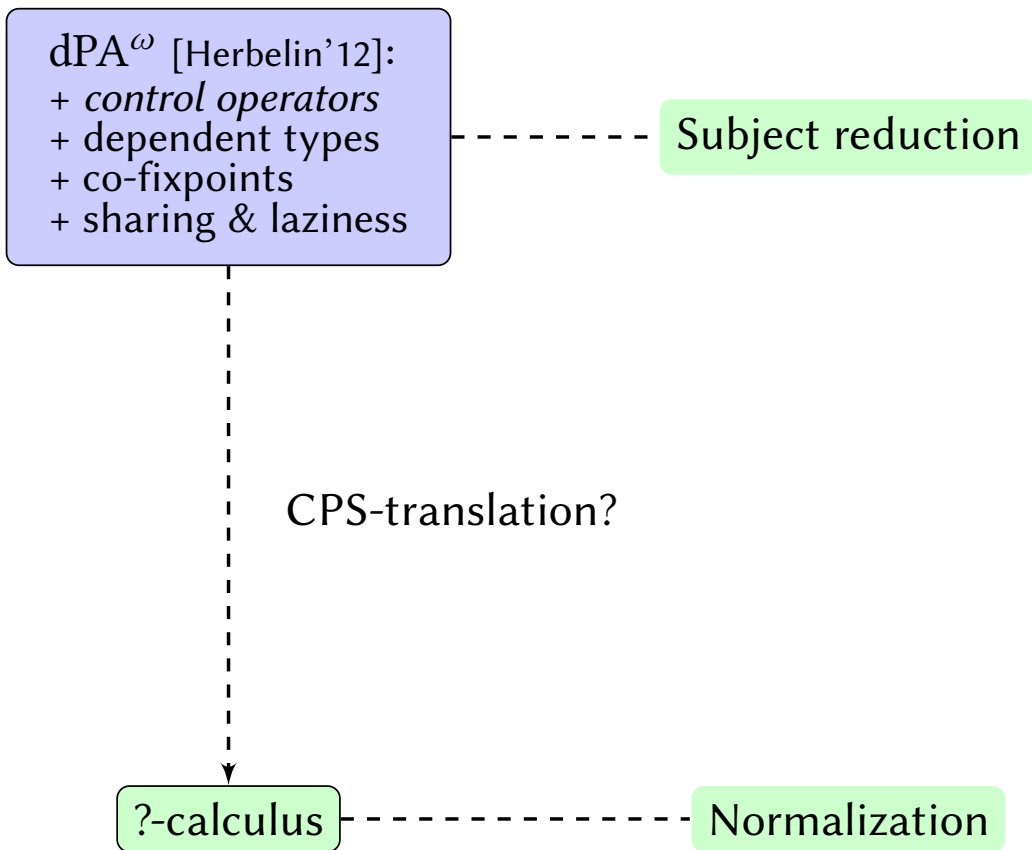
If $\Gamma \vdash p : A$ then p is normalizable.

requires

Consistency

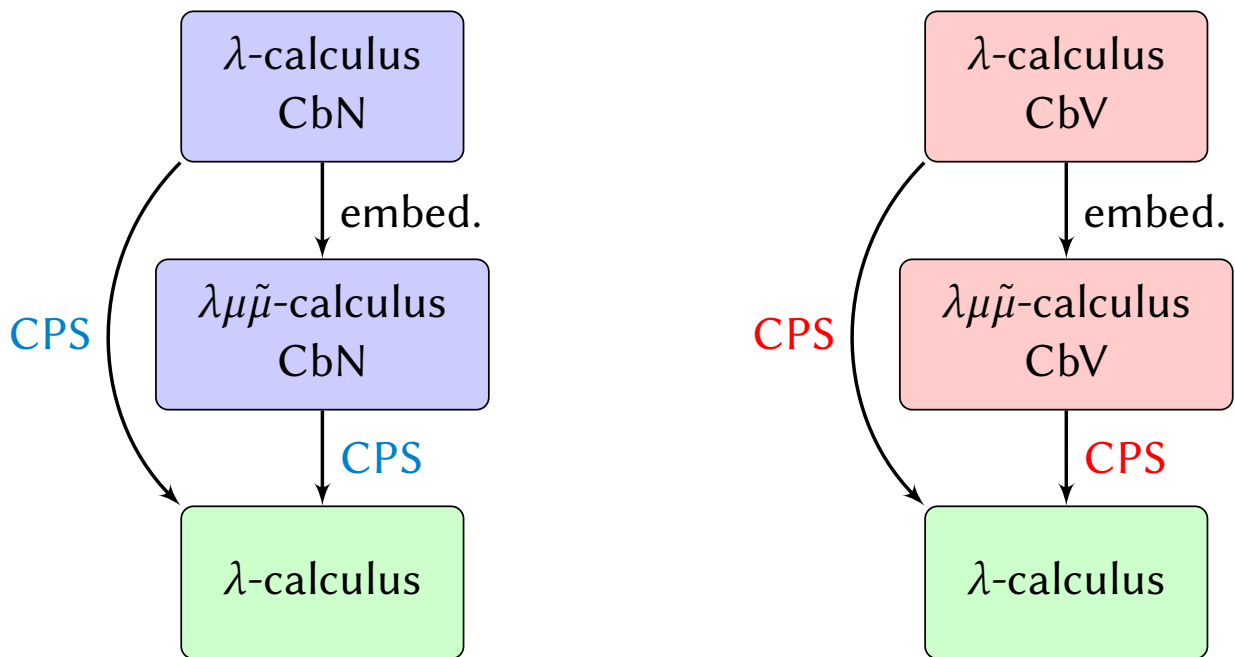
$\not\vdash_{dPA^\omega} \perp$

Roadmap

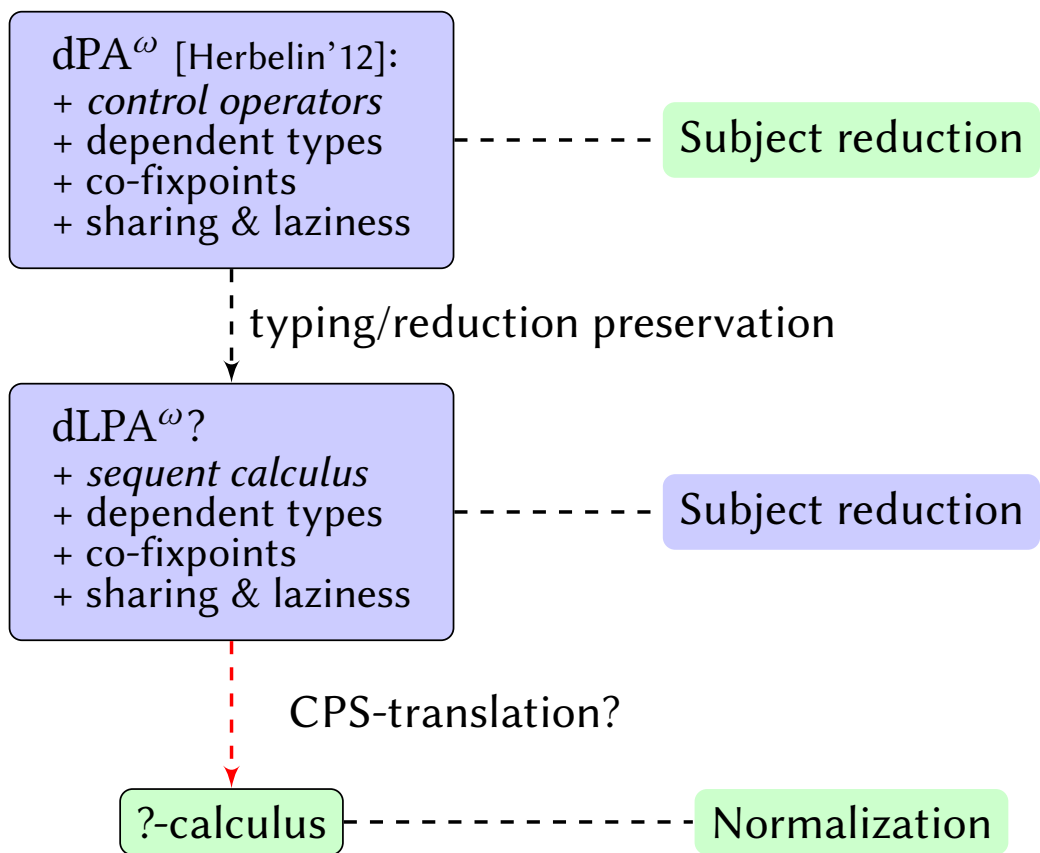


Roadmap

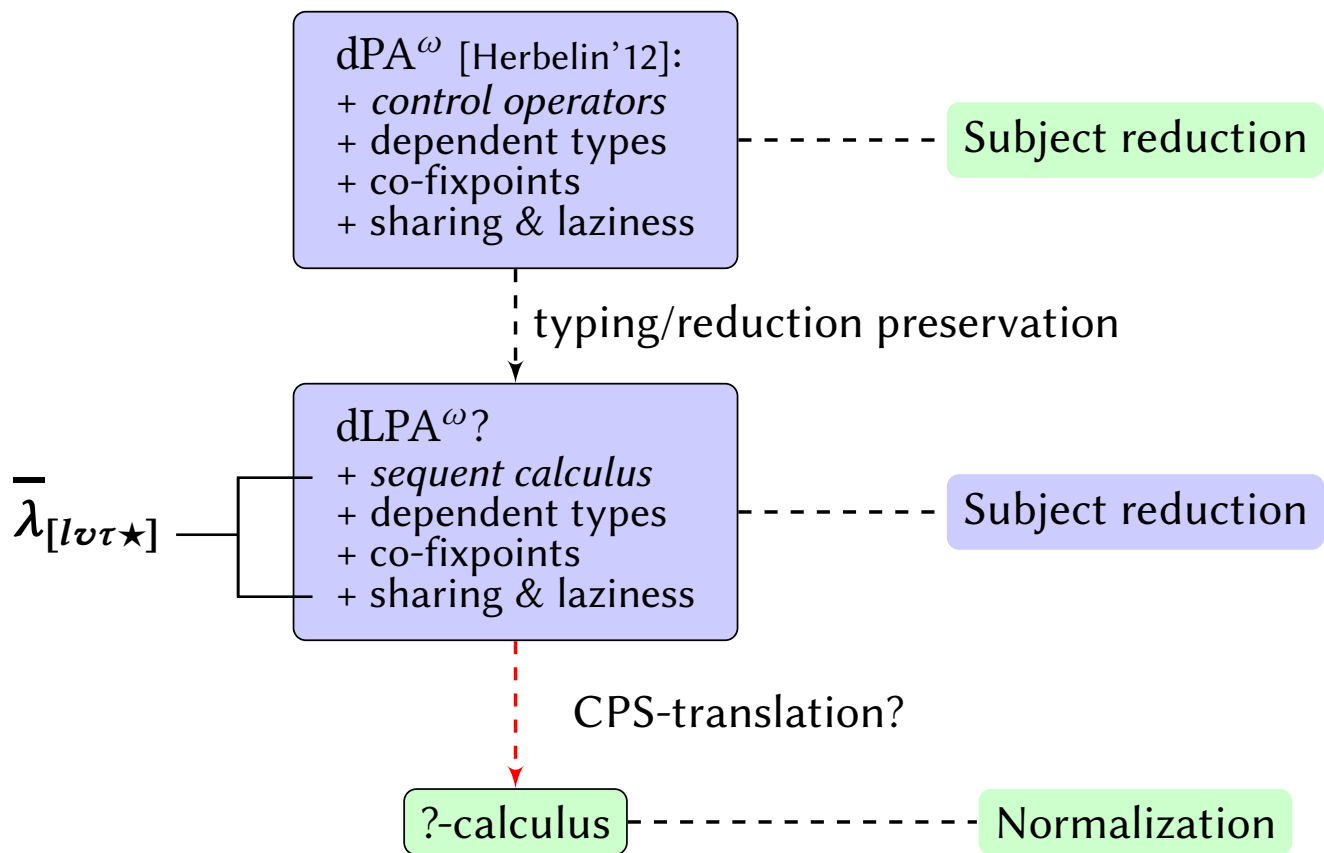
Remark: *CPS usually factorize through sequent calculi!*



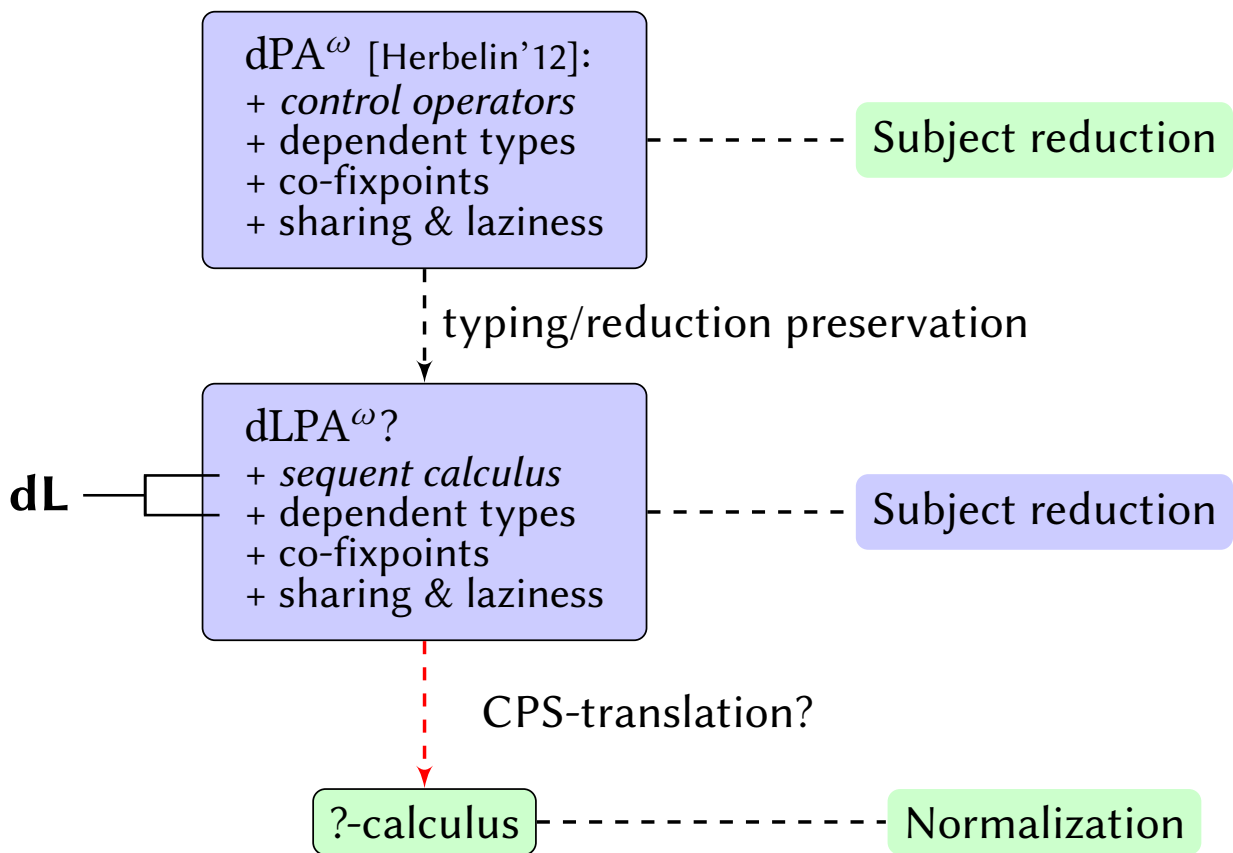
Roadmap



Roadmap



Roadmap



Danvy's semantic artifacts

CPS translation

Continuation-passing style translation: $\llbracket \cdot \rrbracket : source \rightarrow \lambda^{\text{machin}}$

- preserving reduction

$$t \xrightarrow{1} t' \quad \Rightarrow \quad \llbracket t \rrbracket \xrightarrow{+} \llbracket t' \rrbracket$$

- preserving typing

$$\Gamma \vdash t : A \quad \Rightarrow \quad \llbracket \Gamma \rrbracket \vdash \llbracket t \rrbracket : \llbracket A \rrbracket$$

- the type $\llbracket \perp \rrbracket$ is not inhabited

Benefits

If λ^{machin} is sound and normalizing:

- 1 If $\llbracket t \rrbracket$ normalizes, then t normalizes
- 2 If t is typed, then t normalizes
- 3 The source language is sound, *i.e.* there is no term $\vdash t : \perp$

CPS translation

Continuation-passing style translation: $\llbracket \cdot \rrbracket : source \rightarrow \lambda^{\text{machin}}$

- preserving reduction
- preserving typing
- the type $\llbracket \perp \rrbracket$ is not inhabited

Benefits

If λ^{machin} is sound and normalizing:

- 1 If $\llbracket t \rrbracket$ normalizes, then t normalizes
- 2 If t is typed, then t normalizes
- 3 The source language is sound, *i.e.* there is no term $\vdash t : \perp$

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:

(Proofs) $p ::= a \mid \lambda a.p \mid \mu\alpha.c$
 (Contexts) $e ::= \alpha \mid p \cdot e \mid \tilde{\mu}a.c$
 (Commands) $c ::= \langle p \parallel e \rangle$

Typing rules:

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

$$\frac{(a : A) \in \Gamma}{\Gamma \vdash a : A \mid \Delta}$$

$$\frac{\Gamma, a : A \vdash p : B \mid \Delta}{\Gamma \vdash \lambda a.p : A \rightarrow B \mid \Delta}$$

$$\frac{c : (\Gamma \vdash \Delta, \alpha : A)}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}$$

$$\frac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta}$$

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid p \cdot e : A \rightarrow B \vdash \Delta}$$

$$\frac{c : (\Gamma, a : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}a.c : A \vdash \Delta}$$

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

(Proofs) $p ::= a \mid \lambda a.p \mid \mu\alpha.c$
 (Contexts) $e ::= \alpha \mid p \cdot e \mid \tilde{\mu}a.c$
 (Commands) $c ::= \langle p \parallel e \rangle$

Typing rules:

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

$$\frac{A \in \Gamma}{\Gamma \vdash A \mid \Delta}$$

$$\frac{\Gamma, A \vdash B \mid \Delta}{\Gamma \vdash A \rightarrow B \mid \Delta}$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash A \mid \Delta}$$

$$\frac{A \in \Delta}{\Gamma \mid A \vdash \Delta}$$

$$\frac{\Gamma \vdash A \mid \Delta \quad \Gamma \mid B \vdash \Delta}{\Gamma \mid A \rightarrow B \vdash \Delta}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \mid A \vdash \Delta}$$

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

(Proofs) $p ::= a \mid \lambda a.p \mid \mu\alpha.c$
 (Contexts) $e ::= \alpha \mid p \cdot e \mid \tilde{\mu}a.c$
 (Commands) $c ::= \langle p \parallel e \rangle$

Typing rules:

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

$$\frac{(a : A) \in \Gamma}{\Gamma \vdash a : A \mid \Delta}$$

$$\frac{\Gamma, a : A \vdash p : B \mid \Delta}{\Gamma \vdash \lambda a.p : A \rightarrow B \mid \Delta}$$

$$\frac{c : (\Gamma \vdash \Delta, \alpha : A)}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}$$

$$\frac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta}$$

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid p \cdot e : A \rightarrow B \vdash \Delta}$$

$$\frac{c : (\Gamma, a : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}a.c : A \vdash \Delta}$$

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

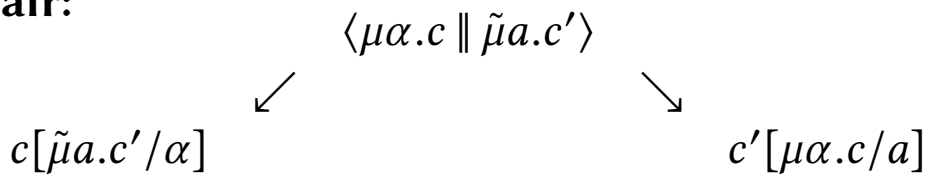
(Proofs) $p ::= a \mid \lambda a.p \mid \mu\alpha.c$
 (Contexts) $e ::= \alpha \mid p \cdot e \mid \tilde{\mu}a.c$
 (Commands) $c ::= \langle p \parallel e \rangle$

Reduction:

$\langle \lambda a.p \parallel q \cdot e \rangle \rightarrow \langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle \rangle$
 $\langle p \parallel \tilde{\mu}a.c \rangle \rightarrow c[p/a]$
 $\langle \mu\alpha.c \parallel e \rangle \rightarrow c[e/\alpha]$

$p \in \mathcal{P}$
 $e \in \mathcal{E}$

Critical pair:



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

The duality of computation
Curien/Herbelin (2000)

Syntax:

(Proofs) $p ::= V \mid \mu\alpha.c$

(Contexts) $e ::= E \mid \tilde{\mu}a.c$

(Commands) $c ::= \langle p \parallel e \rangle$

(Values) $V ::= a \mid \lambda a.p$

(Co-values) $E ::= \alpha \mid p \cdot e$

Reduction:

$\langle \lambda a.p \parallel q \cdot e \rangle \rightarrow \langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle \rangle$

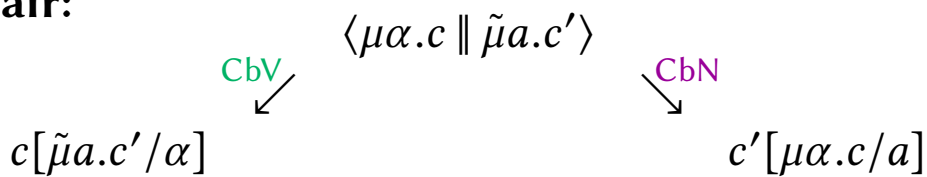
$\langle p \parallel \tilde{\mu}a.c \rangle \rightarrow c[p/a]$

$\langle \mu\alpha.c \parallel e \rangle \rightarrow c[e/\alpha]$

$p \in \mathcal{P}$

$e \in \mathcal{E}$

Critical pair:



Call-by-name $\lambda\mu\tilde{\mu}$ -calculus

Syntax:

(Proofs)	$p ::= V \mid \mu\alpha.c$	(Contexts)	$e ::= E \mid \tilde{\mu}a.c$
(Values)	$V ::= a \mid \lambda a.p$	(Co-values)	$E ::= \alpha \mid p \cdot e$
	(Commands)		$c ::= \langle p \parallel e \rangle$

Reduction rules:

$\langle p \parallel \tilde{\mu}a.c \rangle$	\rightarrow	$c[p/a]$
$\langle \mu\alpha.c \parallel E \rangle$	\rightarrow	$c[E/\alpha]$
$\langle \lambda a.p \parallel q \cdot e \rangle$	\rightarrow	$\langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle \rangle$

Semantic artifacts

(Proofs) $p ::= V \mid \mu\alpha.c$ (Contexts) $e ::= E \mid \tilde{\mu}a.c$
 (Values) $V ::= a \mid \lambda a.p$ (Co-values) $E ::= \alpha \mid p \cdot e$
 (Commands) $c ::= \langle p \parallel e \rangle$

Small steps

e	$\langle p \parallel \tilde{\mu}a.c \rangle_e$	\rightsquigarrow	$c_e[p/a]$
p	$\langle p \parallel E \rangle_e$	\rightsquigarrow	$\langle p \parallel E \rangle_p$
p	$\langle \mu\alpha.c \parallel E \rangle_p$	\rightsquigarrow	$c_e[E/\alpha]$
p	$\langle V \parallel E \rangle_p$	\rightsquigarrow	$\langle V \parallel E \rangle_E$
E	$\langle V \parallel q \cdot e \rangle_E$	\rightsquigarrow	$\langle V \parallel q \cdot e \rangle_V$
V	$\langle \lambda a.p \parallel q \cdot e \rangle_V$	\rightsquigarrow	$\langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle \rangle_e$

Semantic artifacts

(Proofs) $p ::= V \mid \mu\alpha.c$ (Contexts) $e ::= E \mid \tilde{\mu}a.c$
 (Values) $V ::= a \mid \lambda a.p$ (Co-values) $E ::= \alpha \mid p \cdot e$
 (Commands) $c ::= \langle p \parallel e \rangle$

Small steps

CPS

e	$\langle p \parallel \tilde{\mu}a.c \rangle_e$	\rightsquigarrow	$c_e[p/a]$	\triangleq	$[[\tilde{\mu}a.c]]_e p$	\triangleq	$(\lambda a. [[c]]_c) p$
	$\langle p \parallel E \rangle_e$	\rightsquigarrow	$\langle p \parallel E \rangle_p$		$[[E]]_e p$	\triangleq	$p [[E]]_E$
p	$\langle \mu\alpha.c \parallel E \rangle_p$	\rightsquigarrow	$c_e[E/\alpha]$	\triangleq	$[[\mu\alpha.c]]_p E$	\triangleq	$(\lambda\alpha. [[c]]_c) E$
	$\langle V \parallel E \rangle_p$	\rightsquigarrow	$\langle V \parallel E \rangle_E$		$[[V]]_p E$	\triangleq	$E [[V]]_V$
E	$\langle V \parallel q \cdot e \rangle_E$	\rightsquigarrow	$\langle V \parallel q \cdot e \rangle_V$	\triangleq	$[[q \cdot e]]_E V$	\triangleq	$V [[q]]_p [[e]]_e$
V	$\langle \lambda a.p \parallel q \cdot e \rangle_V$	\rightsquigarrow	$\langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle_e$	\triangleq	$[[\lambda a.p]]_V q e$	\triangleq	$(\lambda a. e [[p]]_p) q$

Semantic artifacts

(Proofs) $p ::= V \mid \mu\alpha.c$ (Contexts) $e ::= E \mid \tilde{\mu}a.c$
 (Values) $V ::= a \mid \lambda a.p$ (Co-values) $E ::= \alpha \mid p \cdot e$
 (Commands) $c ::= \langle p \parallel e \rangle$

Small steps

CPS

	e	$\langle p \parallel \tilde{\mu}a.c \rangle_e$	\rightsquigarrow	$c_e[p/a]$	$[[\tilde{\mu}a.c]]_e p \triangleq (\lambda a. [[c]]_c) p$
		$\langle p \parallel E \rangle_e$	\rightsquigarrow	$\langle p \parallel E \rangle_p$	$[[E]]_e p \triangleq p [[E]]_E$
	p	$\langle \mu\alpha.c \parallel E \rangle_p$	\rightsquigarrow	$c_e[E/\alpha]$	$[[\mu\alpha.c]]_p E \triangleq (\lambda\alpha. [[c]]_c) E$
		$\langle V \parallel E \rangle_p$	\rightsquigarrow	$\langle V \parallel E \rangle_E$	$[[V]]_p E \triangleq E [[V]]_V$
	E	$\langle V \parallel q \cdot e \rangle_E$	\rightsquigarrow	$\langle V \parallel q \cdot e \rangle_V$	$[[q \cdot e]]_E V \triangleq V [[q]]_p [[e]]_e$
	V	$\langle \lambda a.p \parallel q \cdot e \rangle_V$	\rightsquigarrow	$\langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle_e$	$[[\lambda a.p]]_V q e \triangleq (\lambda a. e [[p]]_p) q$

$$c \rightsquigarrow^1 c' \quad \Rightarrow \quad [[c]]_c \xrightarrow{\beta^+} [[c']]_c$$

Semantic artifacts

(Proofs) $p ::= V \mid \mu\alpha.c$ (Contexts) $e ::= E \mid \tilde{\mu}a.c$
 (Values) $V ::= a \mid \lambda a.p$ (Co-values) $E ::= \alpha \mid p \cdot e$
 (Commands) $c ::= \langle p \parallel e \rangle$

CPS

Types translation

e	$\llbracket \tilde{\mu}a.c \rrbracket_e p \triangleq (\lambda\alpha. \llbracket c \rrbracket_c) p$ $\llbracket E \rrbracket_e p \triangleq p \llbracket E \rrbracket_E$	$\llbracket A \rrbracket_e \triangleq \llbracket A \rrbracket_p \rightarrow \perp$
p	$\llbracket \mu\alpha.c \rrbracket_p E \triangleq (\lambda\alpha. \llbracket c \rrbracket_c) E$ $\llbracket V \rrbracket_p E \triangleq E \llbracket V \rrbracket_V$	$\llbracket A \rrbracket_p \triangleq \llbracket A \rrbracket_E \rightarrow \perp$
E	$\llbracket q \cdot e \rrbracket_E V \triangleq V \llbracket q \rrbracket_p \llbracket e \rrbracket_e$	$\llbracket A \rrbracket_E \triangleq \llbracket A \rrbracket_V \rightarrow \perp$
V	$\llbracket \lambda a.p \rrbracket_V q e \triangleq (\lambda a.e \llbracket p \rrbracket_p) q$	$\llbracket A \rightarrow B \rrbracket_V \triangleq \llbracket A \rrbracket_p \rightarrow \llbracket A \rrbracket_e \rightarrow \perp$

$$\Gamma \vdash p : A \mid \Delta \quad \Rightarrow \quad \llbracket \Gamma \rrbracket_p, \llbracket \Delta \rrbracket_E \vdash \llbracket p \rrbracket_p : \llbracket A \rrbracket_p$$

Consequences

Normalization

Typed commands of the call-by-name $\lambda\mu\tilde{\mu}$ -calculus normalize.

Inhabitation

There is no simply-typed λ -term t such that $\vdash t : \llbracket \perp \rrbracket_p$.

Proof. $\llbracket \perp \rrbracket_p = (\perp \rightarrow \perp) \rightarrow \perp$ and $\lambda x.x$ is of type $\perp \rightarrow \perp$. □

Soundness

There is no proof p such that $\vdash p : \perp \mid$.

Realizability à la Krivine (1/2)

Intuition

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

$$\langle p \parallel e \rangle > c_0 > \dots > c_n \in \perp\!\!\!\perp?$$

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

Truth value defined by **orthogonality** :

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

Realizability à la Krivine (1/2)

Intuition

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

$$\langle p \parallel e \rangle \succ c_0 \succ \dots \succ c_n \in \perp\!\!\!\perp?$$

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

Truth value defined by **orthogonality** :

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

Realizability à la Krivine (1/2)

Intuition

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

$$\langle p \parallel e \rangle \succ c_0 \succ \dots \succ c_n \in \perp?$$

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

Truth value defined by **orthogonality** :

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

Realizability à la Krivine (1/2)

Intuition

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

$$\langle p \parallel e \rangle \succ c_0 \succ \dots \succ c_n \in \perp?$$

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

Truth value defined by **orthogonality** :

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

Realizability *à la* Krivine (1/2)

Intuition

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

$$\langle p \parallel e \rangle \succ c_0 \succ \dots \succ c_n \in \perp?$$

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

Truth value defined by **orthogonality** :

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

Realizability à la Krivine (1/2)

Intuition

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

$$\langle p \parallel e \rangle \succ c_0 \succ \dots \succ c_n \in \perp ?$$

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

Truth value defined by **orthogonality** :

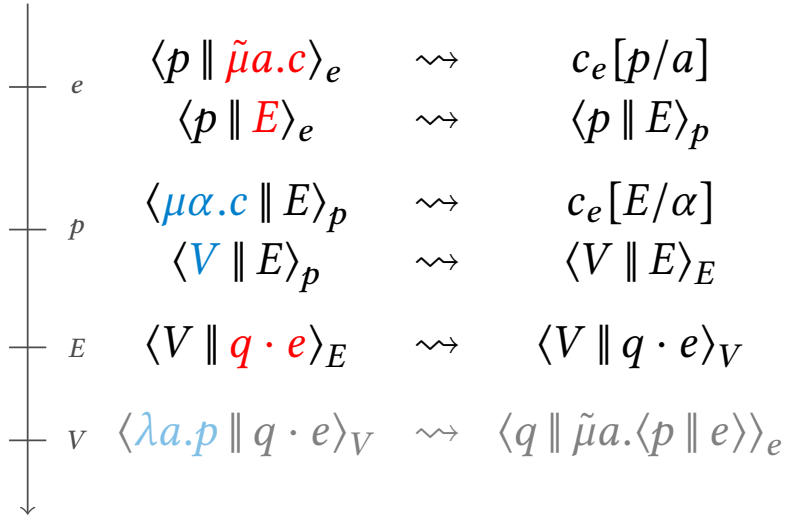
$$|A| = \|A\|^\perp = \{p \in \Lambda : \forall e \in \|A\|, \langle p \parallel e \rangle \in \perp\}$$

Semantic artifacts++

(Terms) $p ::= \mu\alpha.c \mid a \mid V$
(Values) $V ::= \lambda a.p$

(Contexts) $e ::= \tilde{\mu}a.c \mid E$
(Co-values) $E ::= \alpha \mid p \cdot e$

Small steps



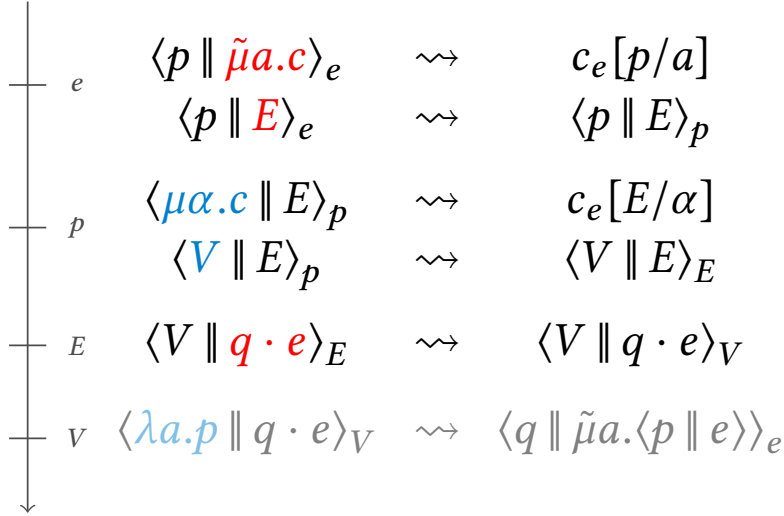
Semantic artifacts++

(Terms) $p ::= \mu\alpha.c \mid a \mid V$
(Values) $V ::= \lambda a.p$

(Contexts) $e ::= \tilde{\mu}a.c \mid E$
(Co-values) $E ::= \alpha \mid p \cdot e$

Small steps

Realizability



$$\|A\|_e \triangleq |A|_p^{\perp\perp}$$

$$|A|_p \triangleq \|A\|_E^{\perp\perp}$$

$$\|A \rightarrow B\|_E \triangleq \{q \cdot e : q \in |A|_p \wedge e \in \|B\|_e\}$$

Extension to second-order

$$\frac{\Gamma \mid e : A[n/x] \vdash \Delta}{\Gamma \mid e : \forall x.A \vdash \Delta} \quad (\forall_l^1)$$

$$\frac{\Gamma \vdash p : A \mid \Delta \quad x \notin FV(\Gamma, \Delta)}{\Gamma \vdash p : \forall x.A \mid \Delta} \quad (\forall_r^1)$$

$$\frac{\Gamma \mid e : A[B/X] \vdash \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta} \quad (\forall_l^2)$$

$$\frac{\Gamma \vdash p : A \mid \Delta \quad X \notin FV(\Gamma, \Delta)}{\Gamma \vdash p : \forall X.A \mid \Delta} \quad (\forall_r^2)$$

(Curry-style)

Realizability à la Krivine (2/2)

Standard model \mathbb{N} for 1st-order expressions

Definition (Pole)

$\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ of commands s.t.:

$$\forall c, c', (c' \in \perp\!\!\!\perp \wedge c \rightarrow c') \Rightarrow c \in \perp\!\!\!\perp$$

Truth value (player):

$$|A|_p = \|A\|_E^{\perp\!\!\!\perp} = \{p \in \Lambda : \forall e \in \|A\|, \langle p \parallel e \rangle \in \perp\!\!\!\perp\}$$

Falsity value (opponent):

$$\begin{aligned} \|\dot{F}(e_1, \dots, e_k)\|_E &= F([e_1], \dots, [e_k]) \\ \|A \rightarrow B\|_E &= \{q \cdot e : q \in |A|_p \wedge e \in \|B\|_e\} \\ \|\forall x. A\|_E &= \bigcup_{n \in \mathbb{N}} \|A[n/x]\|_E \\ \|\forall X. A\|_E &= \bigcup_{F: \mathbb{N}^k \rightarrow \mathcal{P}(\Pi)} \|A[\dot{F}/X]\|_E \\ |A|_p &= \|A\|_E^{\perp\!\!\!\perp} = \{p : \forall e \in \|A\|_E, \langle p \parallel e \rangle \in \perp\!\!\!\perp\} \\ \|A\|_e &= |A|_p^{\perp\!\!\!\perp} = \{e : \forall p \in |A|_p, \langle p \parallel e \rangle \in \perp\!\!\!\perp\} \end{aligned}$$

Realizability *à la* Krivine (2/2)

Standard model \mathbb{N} for 1st-order expressions

Definition (Pole)

$\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ of commands s.t.:

$$\forall c, c', (c' \in \perp\!\!\!\perp \wedge c \rightarrow c') \Rightarrow c \in \perp\!\!\!\perp$$

Truth value (**player**):

$$|A|_p = \|A\|_E^{\perp\!\!\!\perp} = \{p \in \Lambda : \forall e \in \|A\|, \langle p \parallel e \rangle \in \perp\!\!\!\perp\}$$

Falsity value (**opponent**):

$$\begin{aligned} \|\dot{F}(e_1, \dots, e_k)\|_E &= F([e_1], \dots, [e_k]) \\ \|A \rightarrow B\|_E &= \{q \cdot e : q \in |A|_p \wedge e \in \|B\|_e\} \\ \|\forall x. A\|_E &= \bigcup_{n \in \mathbb{N}} \|A[n/x]\|_E \\ \|\forall X. A\|_E &= \bigcup_{F: \mathbb{N}^k \rightarrow \mathcal{P}(\Pi)} \|A[\dot{F}/X]\|_E \\ |A|_p &= \|A\|_E^{\perp\!\!\!\perp} = \{p : \forall e \in \|A\|_E, \langle p \parallel e \rangle \in \perp\!\!\!\perp\} \\ \|A\|_e &= |A|_p^{\perp\!\!\!\perp} = \{e : \forall p \in |A|_p, \langle p \parallel e \rangle \in \perp\!\!\!\perp\} \end{aligned}$$

Realizability *à la* Krivine (2/2)

Standard model \mathbb{N} for 1st-order expressions

Definition (Pole)

$\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ of commands s.t.:

$$\forall c, c', (c' \in \perp\!\!\!\perp \wedge c \rightarrow c') \Rightarrow c \in \perp\!\!\!\perp$$

Truth value (**player**):

$$|A|_p = \|A\|_E^{\perp\!\!\!\perp} = \{p \in \Lambda : \forall e \in \|A\|, \langle p \parallel e \rangle \in \perp\!\!\!\perp\}$$

Falsity value (**opponent**):

$$\begin{aligned} \|\dot{F}(e_1, \dots, e_k)\|_E &= F(\llbracket e_1 \rrbracket, \dots, \llbracket e_k \rrbracket) \\ \|A \rightarrow B\|_E &= \{q \cdot e : q \in |A|_p \wedge e \in \|B\|_e\} \\ \|\forall x. A\|_E &= \bigcup_{n \in \mathbb{N}} \|A[n/x]\|_E \\ \|\forall X. A\|_E &= \bigcup_{F: \mathbb{N}^k \rightarrow \mathcal{P}(\Pi)} \|A[\dot{F}/X]\|_E \\ |A|_p &= \|A\|_E^{\perp\!\!\!\perp} = \{p : \forall e \in \|A\|_E, \langle p \parallel e \rangle \in \perp\!\!\!\perp\} \\ \|A\|_e &= |A|_p^{\perp\!\!\!\perp} = \{e : \forall p \in |A|_p, \langle p \parallel e \rangle \in \perp\!\!\!\perp\} \end{aligned}$$

Adequacy

Valuation ρ :

$$\rho(x) \in \mathbb{N} \qquad \rho(X) : \mathbb{N}^k \rightarrow \mathcal{P}(\Pi)$$

Substitution σ :

$$\sigma ::= \varepsilon \mid \sigma, a := p \mid \sigma, \alpha := E$$

$$\sigma \Vdash \Gamma \triangleq \begin{cases} \sigma(a) \in |A|_p & \forall (a : A) \in \Gamma \\ \sigma(\alpha) \in \|A\|_E & \forall (\alpha : A^\perp) \in \Gamma \end{cases}$$

Adequacy

If $\sigma \Vdash (\Gamma \cup \Delta)[\rho]$, then:

- 1 $\Gamma \vdash p : A \mid \Delta \Rightarrow p[\sigma] \in |A[\rho]|_p$
- 2 $\Gamma \mid e : A \vdash \Delta \Rightarrow e[\sigma] \in \|A[\rho]\|_e$
- 3 $c : (\Gamma \vdash \Delta) \Rightarrow c[\sigma] \in \perp$

Proof. By mutual induction over the typing derivation.

□

Results

Normalizing commands

$\perp\!\!\!\downarrow \triangleq \{c : c \text{ normalizes}\}$ defines a valid pole.

Proof. If $c \rightarrow c'$ 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 $\perp\!\!\!\downarrow$. □

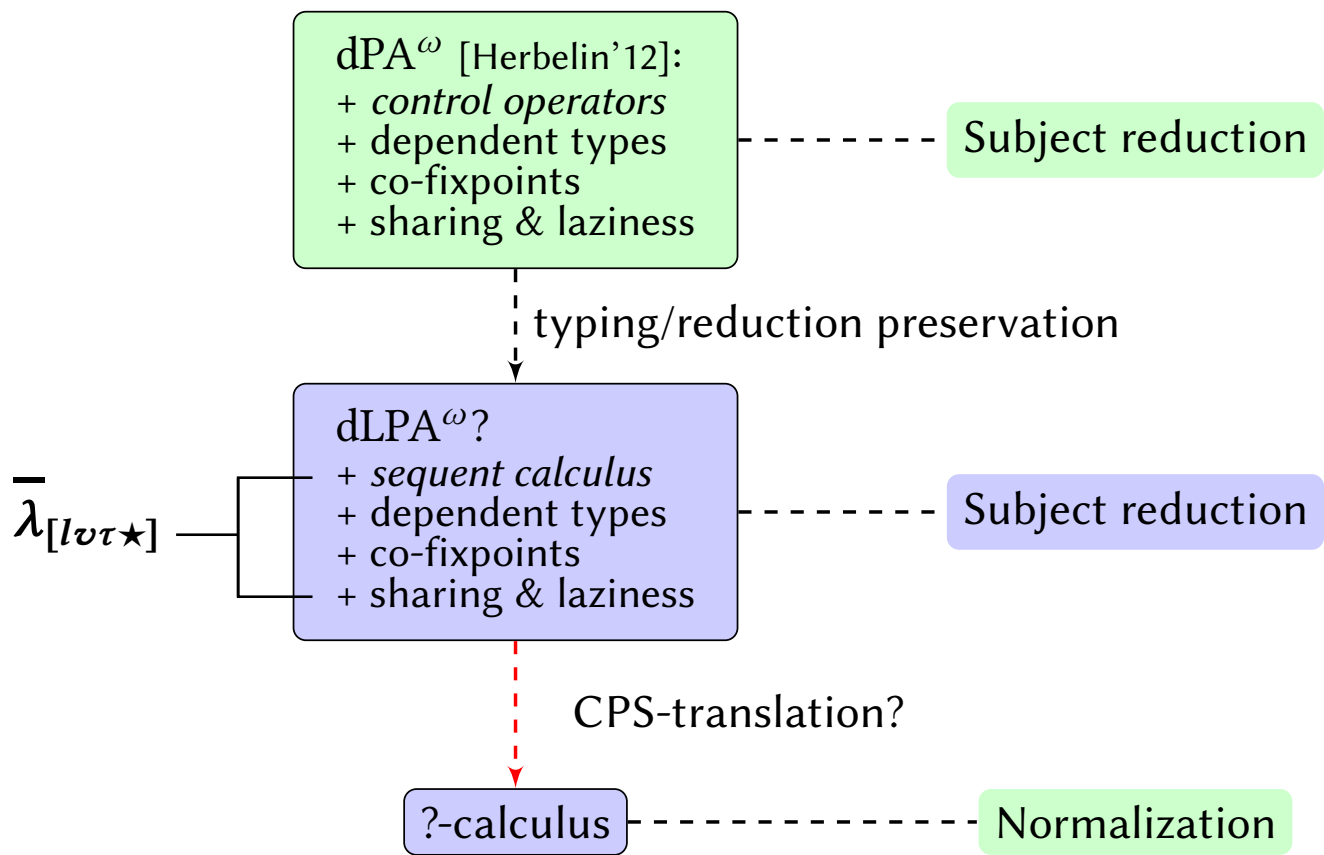
Soundness

There is no proof p such that $\vdash p : \perp \mid$.

Proof. Otherwise, $p \in \perp\!\!\!\downarrow_p = \Pi^\perp$ for any pole, absurd ($\perp\!\!\!\downarrow \triangleq \emptyset$). □

Classical call-by-need

Reminder



Classical call-by-need

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

The $\bar{\lambda}_{[l\ v\ \tau\ \star]}$ -calculus:

- a sequent calculus with explicit “stores”
- Danvy’s method of semantics artifact:
 - 1 derive a small-step reduction system
 - 2 derive context-free small-step reduction rules
 - 3 derive an (untyped) CPS

Questions:

- ↪ Does it normalize?
- ↪ Can the CPS be typed?
- ↪ Can we define a realizability interpretation?

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

Syntax:

(Proofs)	$p ::= V \mid \mu\alpha.c$	$e ::= E \mid \tilde{\mu}a.c$	(Contexts)
(Weak values)	$V ::= v \mid a$	$E ::= \alpha \mid F \mid \tilde{\mu}[a].\langle a \parallel F \rangle\tau$	(Catchable contexts)
(Strong values)	$v ::= \lambda a.p \mid \mathbf{k}$	$F ::= p \cdot E \mid \kappa$	(Forcing contexts)
	(Commands)	$c ::= \langle p \parallel e \rangle$	
	(Closures)	$l ::= c\tau$	
	(Store)	$\tau ::= \epsilon \mid \tau[a := p]$	

Reduction rules:

(Lazy storage)	$\langle p \parallel \tilde{\mu}a.c \rangle\tau$	\rightarrow	$c\tau[a := p]$
	$\langle \mu\alpha.c \parallel E \rangle\tau$	\rightarrow	$(c[E/\alpha])\tau$
(Lookup)	$\langle a \parallel F \rangle\tau[a := \mathbf{p}]\tau'$	\rightarrow	$\langle \mathbf{p} \parallel \tilde{\mu}[a].\langle a \parallel F \rangle\tau' \rangle\tau$
(Forced eval.)	$\langle \mathbf{V} \parallel \tilde{\mu}[a].\langle a \parallel F \rangle\tau' \rangle\tau$	\rightarrow	$\langle \mathbf{V} \parallel F \rangle\tau[a := \mathbf{V}]\tau'$
	$\langle \lambda a.p \parallel q \cdot E \rangle\tau$	\rightarrow	$\langle q \parallel \tilde{\mu}a.\langle p \parallel E \rangle\tau \rangle\tau$

Semantic artifacts

Small steps:

	e	$\langle p \parallel \tilde{\mu}a.c \rangle_{e\tau}$	→	$c_e\tau[a := p]$
		$\langle p \parallel E \rangle_{e\tau}$	→	$\langle p \parallel E \rangle_p\tau$
	p	$\langle \mu\alpha.c \parallel E \rangle_p\tau$	→	$(c[E/\alpha])\tau$
		$\langle V \parallel E \rangle_p\tau$	→	$\langle V \parallel E \rangle_E\tau$
	E	$\langle V \parallel \tilde{\mu}[a].\langle a \parallel F \rangle\tau' \rangle_E\tau$	→	$\langle V \parallel F \rangle_V\tau[a := V]\tau'$
		$\langle V \parallel F \rangle_E\tau$	→	$\langle V \parallel F \rangle_V\tau$
	V	$\langle a \parallel F \rangle_V\tau[a := p]\tau'$	→	$\langle p \parallel \tilde{\mu}[a].\langle a \parallel F \rangle\tau' \rangle_p\tau$
		$\langle \lambda a.p \parallel F \rangle_V\tau$	→	$\langle \lambda a.p \parallel F \rangle_F\tau$
	F	$\langle \lambda a.p \parallel q \cdot E \rangle_F\tau$	→	$\langle q \parallel \tilde{\mu}a.\langle p \parallel E \rangle \rangle_{e\tau}$

Semantic artifacts

CPS :

$$\llbracket \langle p \parallel e \rangle \tau \rrbracket := \llbracket e \rrbracket_e \llbracket \tau \rrbracket_\tau \llbracket p \rrbracket_p$$

$$\llbracket \tilde{\mu} a. c \rrbracket_e := \lambda \tau p. \llbracket c \rrbracket \tau [a := p]$$

$$\llbracket E \rrbracket_e := \lambda \tau p. p \tau \llbracket E \rrbracket_E$$

$$\llbracket \mu \alpha. c \rrbracket_p := \lambda \tau E. (\llbracket c \rrbracket_c \tau) [E/\alpha]$$

$$\llbracket V \rrbracket_p := \lambda \tau E. E \tau \llbracket V \rrbracket_v$$

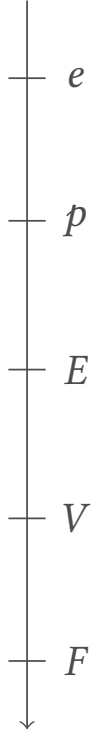
$$\llbracket \tilde{\mu} [a]. \langle a \parallel F \rangle \tau' \rrbracket_E := \lambda \tau V. V \tau [a := V] \tau' \llbracket F \rrbracket_F$$

$$\llbracket F \rrbracket_E := \lambda \tau V. V \tau \llbracket F \rrbracket_F$$

$$\llbracket a \rrbracket_v := \lambda \tau F. \tau(a) \tau (\lambda \tau V. V \tau [a := V] \tau' \llbracket F \rrbracket_F)$$

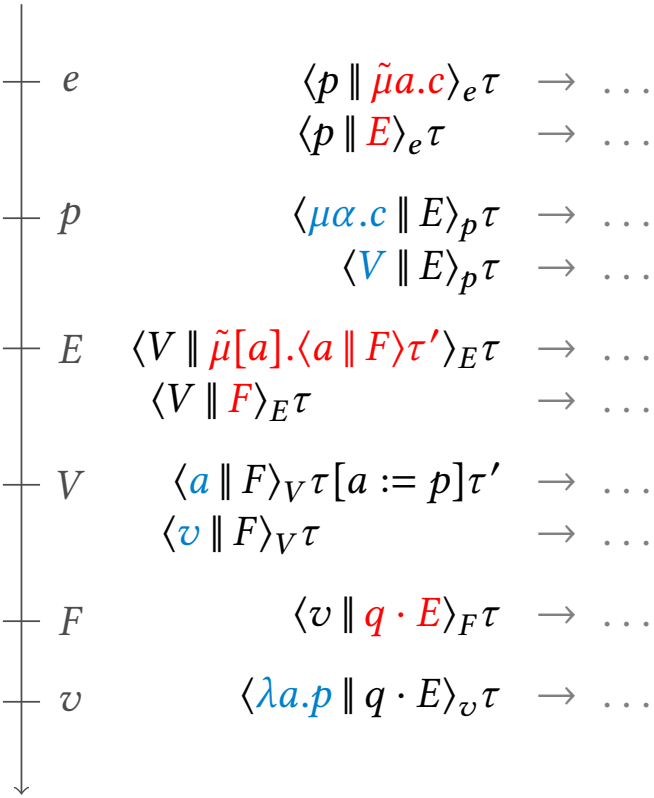
$$\llbracket \lambda a. p \rrbracket_v := \lambda \tau F. F \tau (\lambda q \tau E. \llbracket p \rrbracket_p \tau [a := q] E)$$

$$\llbracket q \cdot E \rrbracket_F := \lambda \tau v. v \llbracket q \rrbracket_p \tau \llbracket E \rrbracket_E$$



Semantic artifacts

Small-step:



Semantic artifacts

Small-step:

Realizability:

			$(\perp \subseteq ?)$
e	$\langle p \parallel \tilde{\mu}a.c \rangle_e \tau \rightarrow \dots$ $\langle p \parallel E \rangle_e \tau \rightarrow \dots$		$\ A\ _e := \{ e? \in A _p^\perp \}$
p	$\langle \mu\alpha.c \parallel E \rangle_p \tau \rightarrow \dots$ $\langle V \parallel E \rangle_p \tau \rightarrow \dots$		$ A _p := \{ p? \in \ A\ _E^\perp \}$
E	$\langle V \parallel \tilde{\mu}[a].\langle a \parallel F \rangle \tau' \rangle_E \tau \rightarrow \dots$ $\langle V \parallel F \rangle_E \tau \rightarrow \dots$		$\ A\ _E := \{ E? \in A _V^\perp \}$
V	$\langle a \parallel F \rangle_V \tau[a := p] \tau' \rightarrow \dots$ $\langle v \parallel F \rangle_V \tau \rightarrow \dots$		$ A _V := \{ V? \in \ A\ _F^\perp \}$
F	$\langle v \parallel q \cdot E \rangle_F \tau \rightarrow \dots$		$\ A\ _F := \{ F? \in A _v^\perp \}$
v	$\langle \lambda a.p \parallel q \cdot E \rangle_v \tau \rightarrow \dots$		$ A \rightarrow B _v := \{ \lambda a.p? : q? \in A _t$ $\Rightarrow p[q/a]? \in B _t \}$

Semantic artifacts

Small-step:

	e	$\langle p \parallel \tilde{\mu}a.c \rangle_e \tau \rightarrow \dots$ $\langle p \parallel E \rangle_e \tau \rightarrow \dots$
	p	$\langle \mu\alpha.c \parallel E \rangle_p \tau \rightarrow \dots$ $\langle V \parallel E \rangle_p \tau \rightarrow \dots$
	E	$\langle V \parallel \tilde{\mu}[a].\langle a \parallel F \rangle \tau' \rangle_E \tau \rightarrow \dots$ $\langle V \parallel F \rangle_E \tau \rightarrow \dots$
	V	$\langle a \parallel F \rangle_V \tau[a := p] \tau' \rightarrow \dots$ $\langle v \parallel F \rangle_V \tau \rightarrow \dots$
	F	$\langle v \parallel q \cdot E \rangle_F \tau \rightarrow \dots$
	v	$\langle \lambda a.p \parallel q \cdot E \rangle_v \tau \rightarrow \dots$
↓		

Realizability:

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

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

$$|A|_p := \{ p? \in \|A\|_E^\perp \}$$

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

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

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

$$|A \rightarrow B|_v := \{ \lambda a.p? : q? \in |A|_t \Rightarrow p[q/a]? \in |B|_t \}$$

Semantic artifacts

Small-step:

	e	$\langle p \parallel \tilde{\mu}a.c \rangle_e \tau \rightarrow \dots$	$\langle p \parallel E \rangle_e \tau \rightarrow \dots$
	p	$\langle \mu\alpha.c \parallel E \rangle_p \tau \rightarrow \dots$	$\langle V \parallel E \rangle_p \tau \rightarrow \dots$
	E	$\langle V \parallel \tilde{\mu}[a].\langle a \parallel F \rangle \tau' \rangle_E \tau \rightarrow \dots$	$\langle V \parallel F \rangle_E \tau \rightarrow \dots$
	V	$\langle a \parallel F \rangle_V \tau [a := p] \tau' \rightarrow \dots$	$\langle v \parallel F \rangle_V \tau \rightarrow \dots$
	F	$\langle v \parallel q \cdot E \rangle_F \tau \rightarrow \dots$	
	v	$\langle \lambda a.p \parallel q \cdot E \rangle_v \tau \rightarrow \dots$	
↓			

Realizability:

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

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

$$|A|_p := \{(p|\tau) \in \|A\|_E^\perp\}$$

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

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

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

$$|A \rightarrow B|_v := \{(\lambda a.p|\tau) : (q|\tau') \in |A|_t \Rightarrow (p|\overline{\tau\tau'}[a := q]) \in |B|_t\}$$

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:

- *saturated*:

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

- *closed by store extension*:

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

- **Orthogonality** :

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

- **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:

- *saturated*:

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

- *closed by store extension*:

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

- **Orthogonality** :

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

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

Realizability interpretation

Adequacy

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

Normalization

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

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

Realizability interpretation

Adequacy

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

Normalization

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

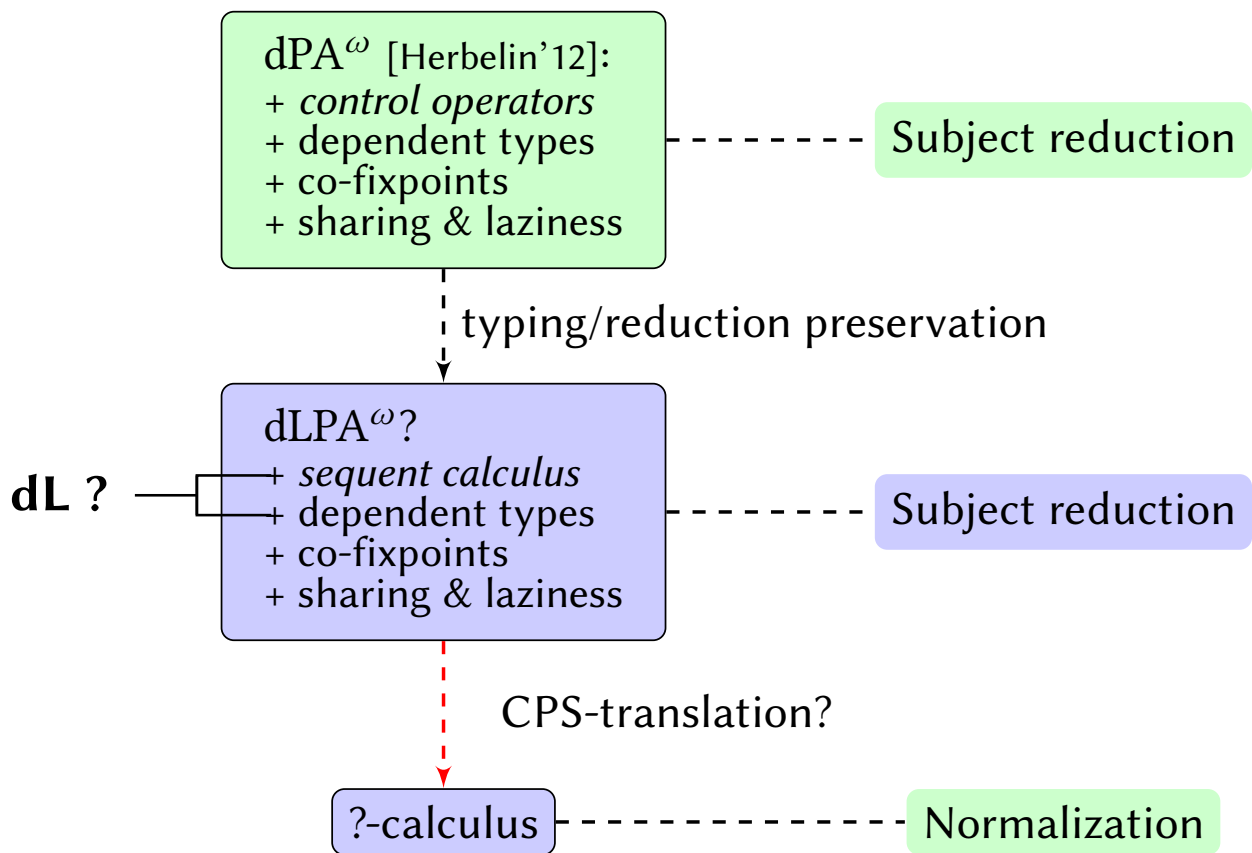
Proof: The set $\perp_{\downarrow} = \{c\tau \in C_0 : c\tau \text{ normalizes}\}$ is a pole.

Initial questions:

- ↪ Does it normalize? *Yes!*
- ↪ Can the CPS be typed? *Yes!* (but it is complicated...)
- ↪ Can we define a realizability interpretation? *Yes!*

A sequent calculus with dependent types

Reminder



A classical sequent calculus with dependent types

Can this work?

$$\frac{
 \frac{
 \frac{
 \Pi_p
 \vdots
 }{
 \Gamma, a : A \vdash p : B[a] \mid \Delta
 }
 }{
 \Gamma \vdash \lambda a.p : \Pi_{(a:A)}.B \mid \Delta
 }
 (\rightarrow_r)
 \quad
 \frac{
 \frac{
 \frac{
 \Pi_q
 \vdots
 }{
 \Gamma \vdash q : A \mid \Delta
 }
 \quad
 \frac{
 \frac{
 \Pi_e
 \vdots
 }{
 \Gamma \mid e : B[q] \vdash \Delta
 }
 \quad
 q \in V
 }{
 \Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta
 }
 (\rightarrow_l)
 }{
 \Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta
 }
 }{
 \langle \lambda a.p \parallel q \cdot e \rangle : (\Gamma \vdash \Delta)
 }
 }
 }
 }
 (\text{CUT})$$



A classical sequent calculus with dependent types

Can this work?

$$\frac{\frac{\frac{\Pi_p \vdots}{\Gamma, a : A \vdash p : B[a] \mid \Delta} (\rightarrow_r)}{\Gamma \vdash \lambda a.p : \Pi_{(a:A)}.B \mid \Delta} \quad \frac{\frac{\frac{\Pi_q \vdots}{\Gamma \vdash q : A \mid \Delta} \quad \frac{\frac{\Pi_e \vdots}{\Gamma \mid e : B[q] \vdash \Delta} \quad q \in V}{\Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta} (\rightarrow_l)}{\Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta} (\text{CUT})}}{\langle \lambda a.p \parallel q \cdot e \rangle : (\Gamma \vdash \Delta)} (\text{CUT})$$

→

$$\frac{\frac{\frac{\Pi_q \vdots}{\Gamma \vdash q : A \mid \Delta} \quad \frac{\frac{\Gamma, a : A \vdash p : \cancel{B[a]} \mid \Delta \quad \Gamma, a : A \mid e : \cancel{B[q]} \vdash \Delta}{\langle p \parallel e \rangle : (\Gamma, a : A \vdash \Delta)} (\tilde{\mu})}}{\Gamma \mid \tilde{\mu} a. \langle p \parallel e \rangle : A \vdash \Delta} (\text{CUT})}}{\langle q \parallel \tilde{\mu} a. \langle p \parallel e \rangle \rangle : (\Gamma \vdash \Delta)} (\text{CUT}) \quad \text{Mismatch}$$

A classical sequent calculus with dependent types

Can this work? ✓

$$\frac{\frac{\frac{\Pi_p \vdots}{\Gamma, a : A \vdash p : B[a] \mid \Delta}}{\Gamma \vdash \lambda a.p : \Pi_{(a:A)}.B \mid \Delta} (\rightarrow_r) \quad \frac{\frac{\frac{\Pi_q \vdots}{\Gamma \vdash q : A \mid \Delta} \quad \frac{\frac{\Pi_e \vdots}{\Gamma \mid e : B[q] \vdash \Delta}}{\Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta} (\rightarrow_l)}{\Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta} (\text{CUT})}}{\langle \lambda a.p \parallel q \cdot e \rangle : (\Gamma \vdash \Delta)} (\text{CUT})$$

→

$$\frac{\frac{\frac{\Pi_q \vdots}{\Gamma \vdash q : A \mid \Delta} \quad \frac{\frac{\Gamma, a : A \vdash p : B[a] \mid \Delta \quad \Gamma, a : A \mid e : B[q] \vdash \Delta; \{\cdot|p\}\{a|q\}}{\langle p \parallel e \rangle : \Gamma, a : A \vdash \Delta; \{a|q\}} (\text{CUT})}{\Gamma \mid \tilde{\mu}a.\langle p \parallel e \rangle : A \vdash \Delta; \{\cdot|q\}} (\tilde{\mu})}}{\langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle \rangle : (\Gamma \vdash \Delta); \{\cdot|\cdot\}} (\text{CUT})$$

dL

$\lambda\mu\tilde{\mu}$ -calculus + dependent types with:

- a **list of dependencies**:

$$\frac{\Gamma \vdash p : A \mid \Delta; \sigma \quad \Gamma \mid e : A' \vdash \Delta; \sigma\{\cdot|p\} \quad A' \in A_\sigma}{\langle p \parallel e \rangle : (\Gamma \vdash \Delta; \sigma)} \text{ (CUT)}$$

- a **value restriction**

Is it enough?

- subject reduction
- normalization
- consistency as a logic
- suitable for CPS translation

dL

$\lambda\mu\tilde{\mu}$ -calculus + dependent types with:

- a **list of dependencies**:

$$\frac{\Gamma \vdash p : A \mid \Delta; \sigma \quad \Gamma \mid e : A' \vdash \Delta; \sigma\{\cdot|p\} \quad A' \in A_\sigma}{\langle p \parallel e \rangle : (\Gamma \vdash \Delta; \sigma)} \text{ (CUT)}$$

- a **value restriction**

Is it enough?

- subject reduction ✓
- normalization ✓
- consistency as a logic ✓
- suitable for CPS translation ✗

dL

$\lambda\mu\tilde{\mu}$ -calculus + dependent types with:

- a **list of dependencies**:

$$\frac{\Gamma \vdash p : A \mid \Delta; \sigma \quad \Gamma \mid e : A' \vdash \Delta; \sigma\{\cdot|p\} \quad A' \in A_\sigma}{\langle p \parallel e \rangle : (\Gamma \vdash \Delta; \sigma)} \text{ (CUT)}$$

- a **value restriction**

Is it enough?

- subject reduction ✓
- normalization ✓
- consistency as a logic ✓
- suitable for CPS translation ✗

$$\llbracket q \rrbracket \llbracket \tilde{\mu}a. \langle p \parallel e \rangle \rrbracket = \underbrace{\llbracket q \rrbracket}_{\neg\neg A} (\lambda a. \underbrace{\llbracket p \rrbracket}_{\neg\neg B(a)} \underbrace{\llbracket e \rrbracket}_{\neg B(q)})$$

Toward a CPS translation (1/2)

This is quite normal:

- we observed a desynchronization
 - we compensated only within the type system
- ↪ *we need to do this already in the calculus!*

Who's guilty ?

$$\llbracket \langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle \rrbracket = \llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket \llbracket e \rrbracket)$$

Motto: $\llbracket p \rrbracket$ shouldn't be applied to $\llbracket e \rrbracket$ before $\llbracket q \rrbracket$ has reduced

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

So, we're looking for:

Toward a CPS translation (1/2)

This is quite normal:

- we observed a desynchronization
 - we compensated only within the type system
- ↪ *we need to do this already in the calculus!*

Who's guilty ?

$$\llbracket \langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle \rrbracket = \llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket \llbracket e \rrbracket)$$

Motto: $\llbracket p \rrbracket$ shouldn't be applied to $\llbracket e \rrbracket$ before $\llbracket q \rrbracket$ has reduced

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

So, we're looking for:

Toward a CPS translation (1/2)

This is quite normal:

- we observed a desynchronization
 - we compensated only within the type system
- ↪ *we need to do this already in the calculus!*

Who's guilty ?

$$\llbracket \langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle \rrbracket = \llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket \llbracket e \rrbracket)$$

Motto: $\llbracket p \rrbracket$ shouldn't be applied to $\llbracket e \rrbracket$ before $\llbracket q \rrbracket$ has reduced

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

So, we're looking for:

Toward a CPS translation (1/2)

This is quite normal:

- we observed a desynchronization
- we compensated only within the type system

↪ *we need to do this already in the calculus!*

Who's guilty ?

$$\llbracket \langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle \rrbracket = \llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket \llbracket e \rrbracket)$$

Motto: $\llbracket p \rrbracket$ shouldn't be applied to $\llbracket e \rrbracket$ before $\llbracket q \rrbracket$ has reduced

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

So, we're looking for:

$$\langle \lambda a. p \parallel q \cdot e \rangle \rightarrow \langle \mu ? . \langle q \parallel \tilde{\mu}a. \langle p \parallel ? \rangle \rangle \parallel e \rangle$$

Toward a CPS translation (1/2)

This is quite normal:

- we observed a desynchronization
- we compensated only within the type system

↪ *we need to do this already in the calculus!*

Who's guilty ?

$$\llbracket \langle q \parallel \tilde{\mu}a. \langle p \parallel e \rangle \rangle \rrbracket = \llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket \llbracket e \rrbracket)$$

Motto: $\llbracket p \rrbracket$ shouldn't be applied to $\llbracket e \rrbracket$ before $\llbracket q \rrbracket$ has reduced

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

So, we're looking for:

$$\langle \lambda a. p \parallel q \cdot e \rangle \rightarrow \langle \mu \hat{t}p. \langle q \parallel \tilde{\mu}a. \langle p \parallel \hat{t}p \rangle \rangle \parallel e \rangle$$

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

- 1 Is any q compatible with such a reduction ?
- 2 Is this typable ?

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

1 Is any q compatible with such a reduction ?

- If q eventually gives a value V :

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket \rightarrow ((\lambda a. \llbracket p \rrbracket) \llbracket V \rrbracket) \llbracket e \rrbracket \rightarrow \llbracket p \rrbracket \llbracket \llbracket V \rrbracket / a \rrbracket \llbracket e \rrbracket = \llbracket p[V/a] \rrbracket \llbracket e \rrbracket \quad \checkmark$$

- If $\llbracket q \rrbracket \rightarrow \lambda _ . t$ and drops its continuation (meaning $t : \perp$):

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket \rightarrow ((\lambda _ . t) \lambda a. \llbracket p \rrbracket) \llbracket e \rrbracket \rightarrow t \llbracket e \rrbracket \quad \times$$

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

1 Is any q compatible with such a reduction ?

- If q eventually gives a value V :

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket \rightarrow ((\lambda a. \llbracket p \rrbracket) \llbracket V \rrbracket) \llbracket e \rrbracket \rightarrow \llbracket p \rrbracket \llbracket \llbracket V \rrbracket / a \rrbracket \llbracket e \rrbracket = \llbracket p[V/a] \rrbracket \llbracket e \rrbracket \quad \checkmark$$

- If $\llbracket q \rrbracket \rightarrow \lambda _ . t$ and drops its continuation (meaning $t : \perp$):

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket \rightarrow ((\lambda _ . t) \lambda a. \llbracket p \rrbracket) \llbracket e \rrbracket \rightarrow t \llbracket e \rrbracket \quad \times$$

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

1 Is any q compatible with such a reduction? $\rightsquigarrow q \in \text{NEF}$

- If q eventually gives a value V :

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket \rightarrow ((\lambda a. \llbracket p \rrbracket) \llbracket V \rrbracket) \llbracket e \rrbracket \rightarrow \llbracket p \rrbracket \llbracket \llbracket V \rrbracket / a \rrbracket \llbracket e \rrbracket = \llbracket p[V/a] \rrbracket \llbracket e \rrbracket \quad \checkmark$$

- If $\llbracket q \rrbracket \rightarrow \lambda _ . t$ and drops its continuation (meaning $t : \perp$):

$$(\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket \rightarrow ((\lambda _ . t) \lambda a. \llbracket p \rrbracket) \llbracket e \rrbracket \rightarrow t \llbracket e \rrbracket \quad \times$$

Negative-elimination free (Herbelin'12)

Values + one continuation variable + no application

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

- 1 Is any q compatible with such a reduction? $\rightsquigarrow q \in \text{NEF}$
- 2 Is this typable?

Naive attempt:

$$\left(\underbrace{\llbracket q \rrbracket}_{(A \rightarrow \perp) \rightarrow \perp} \quad \left(\underbrace{\lambda a. \llbracket p \rrbracket}_{\Pi_{(a:A)} \neg \neg B(a)} \right) \right) \underbrace{\llbracket e \rrbracket}_{\neg B[q]}$$

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

- 1 Is any q compatible with such a reduction? $\rightsquigarrow q \in \text{NEF}$
- 2 Is this typable?

Naive attempt:

$$\underbrace{\left(\underbrace{\llbracket q \rrbracket}_{(A \rightarrow ?) \rightarrow ?} \quad \left(\underbrace{\lambda a. \llbracket p \rrbracket}_{\Pi_{(a:A)} \neg \neg B(a)} \right) \right)}_{\neg \neg B(q)} \quad \underbrace{\llbracket e \rrbracket}_{\neg B[q]}$$

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

- 1 Is any q compatible with such a reduction? $\rightsquigarrow q \in \text{NEF}$
- 2 Is this typable?

Friedman's trick:

$$\underbrace{\left(\underbrace{\llbracket q \rrbracket}_{\forall R. (A \rightarrow R?) \rightarrow R?} \quad \left(\underbrace{\lambda a. \llbracket p \rrbracket}_{\Pi_{(a:A)} \neg \neg B(a)} \right) \right)}_{\neg \neg B} \quad \underbrace{\llbracket e \rrbracket}_{\neg B[q]}$$

Toward a CPS translation (2/2)

$$\llbracket \langle \lambda a. p \parallel q \cdot e \rangle \rrbracket \xrightarrow{?} (\llbracket q \rrbracket (\lambda a. \llbracket p \rrbracket)) \llbracket e \rrbracket$$

Questions:

- 1 Is any q compatible with such a reduction? $\rightsquigarrow q \in \text{NEF}$
- 2 Is this typable? \rightsquigarrow *parametric return-type*

Better:

$$\underbrace{\left(\underbrace{\llbracket q \rrbracket}_{\forall R. (\prod_{(a:A)} R(a)) \rightarrow R(q)} \quad \left(\underbrace{\lambda a. \llbracket p \rrbracket}_{\prod_{(a:A)} \neg \neg B(a)} \right) \right)}_{\neg \neg B(q)} \quad \underbrace{\llbracket e \rrbracket}_{\neg B[q]}$$

(Remark: not possible without $q \in \text{NEF}$)

dL_{tp}[^]

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF fragment**

dL_{t̂p}

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF** fragment

Reduction rules:

$$\begin{aligned}
 \langle \mu t\hat{p}. \langle p \parallel t\hat{p} \rangle \parallel e \rangle &\rightarrow \langle p \parallel e \rangle \\
 c \rightarrow c' &\Rightarrow \langle \mu t\hat{p}. c \parallel e \rangle \rightarrow \langle \mu t\hat{p}. c' \parallel e \rangle \\
 &\vdots \\
 \langle \lambda a. p \parallel q \cdot e \rangle &\rightarrow \langle \mu t\hat{p}. \langle q \parallel \tilde{\mu} a. \langle p \parallel t\hat{p} \rangle \rangle \parallel e \rangle && (q \in \text{NEF}) \\
 \langle \lambda a. p \parallel q \cdot e \rangle &\rightarrow \langle q \parallel \tilde{\mu} a. \langle p \parallel e \rangle \rangle && (q \notin \text{NEF}) \\
 \langle \text{prf } p \parallel e \rangle &\rightarrow \langle \mu t\hat{p}. \langle p \parallel \tilde{\mu} a. \langle \text{prf } a \parallel t\hat{p} \rangle \rangle \parallel e \rangle
 \end{aligned}$$

dL_{t̂p}

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF fragment**

Typing rules:

Regular mode

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle p \parallel e \rangle : \Gamma \vdash \Delta}$$

Dependent mode

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash_d \Delta, \hat{t}p : B; \sigma \{ \cdot | p \}}{\langle p \parallel e \rangle : \Gamma \vdash_d \Delta, \hat{t}p : B; \sigma}$$

dL_{t̂p}

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF** fragment

Typing rules:

Regular mode

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle p \parallel e \rangle : \Gamma \vdash \Delta}$$

Dependent mode

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot|p\}}{\langle p \parallel e \rangle : \Gamma \vdash_d \Delta, \hat{t}p : B; \sigma}$$

Use of σ limited to $\hat{t}p$:

$$\frac{c : (\Gamma \vdash_d \Delta, \hat{t}p : A; \{\cdot|\cdot\})}{\Gamma \vdash \mu\hat{t}p.c : A \mid \Delta} \hat{t}p_I$$

$$\frac{B \in A_\sigma}{\Gamma \mid \hat{t}p : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot|p\}} \hat{t}p_E$$

dL_{t̂p}

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF** fragment

Typing rules:

Regular mode

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle p \parallel e \rangle : \Gamma \vdash \Delta}$$

Dependent mode

$$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot|p\}}{\langle p \parallel e \rangle : \Gamma \vdash_d \Delta, \hat{t}p : B; \sigma}$$

Use of σ limited to $\hat{t}p$:

$$\frac{c : (\Gamma \vdash_d \Delta, \hat{t}p : A; \{\cdot|\cdot\})}{\Gamma \vdash \mu\hat{t}p.c : A \mid \Delta} \hat{t}p_I \quad \frac{B \in A_\sigma}{\Gamma \mid \hat{t}p : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot|p\}} \hat{t}p_E$$

$$c : (\Gamma \vdash \Delta) \quad \wedge \quad c \rightarrow c' \quad \Rightarrow \quad c' : (\Gamma \vdash \Delta)$$

Typed CPS translation

Target language:

$$\top \mid \perp \mid t = u \mid \forall x^{\mathbb{N}}.A \mid \exists x^{\mathbb{N}}.A \mid \Pi_{(a:A)}B \mid \forall X.A$$

Normalization:

If $\llbracket c \rrbracket$ normalizes so does c .

Proof. Thorough analysis of the several reduction rules. □

Types-preserving:

The translation is well-typed.

Proof. Using parametric return types for terms and NEF proofs. □

Consistency:

$\not\vdash p : \perp$.

Proof. $\llbracket \perp \rrbracket = (\perp \rightarrow \perp) \rightarrow \perp$. □

Bilan

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF fragment**

<p><i>Regular mode</i></p> $\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle p \parallel e \rangle : \Gamma \vdash \Delta}$		<p><i>Dependent mode</i></p> $\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot \mid p\}}{\langle p \parallel e \rangle : \Gamma \vdash_d \Delta, \hat{t}p : B; \sigma}$
---	--	---

- delimited scope of dependencies:

$\frac{c : (\Gamma \vdash_d \Delta, \hat{t}p : A; \{\cdot \mid \cdot\})}{\Gamma \vdash \mu \hat{t}p.c : A \mid \Delta} \hat{t}p_I$		$\frac{B \in A_\sigma}{\Gamma \mid \hat{t}p : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot \mid p\}} \hat{t}p_E$
--	--	--

- Mission accomplished?
 - subject reduction
 - normalization
 - consistency as a logic
 - CPS translation
- (Bonus) embedding into Rodolphe's calculus ✓
 - ↪ realizability interpretation

Bilan

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the **NEF fragment**

<p><i>Regular mode</i></p> $\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle p \parallel e \rangle : \Gamma \vdash \Delta}$		<p><i>Dependent mode</i></p> $\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot \mid p\}}{\langle p \parallel e \rangle : \Gamma \vdash_d \Delta, \hat{t}p : B; \sigma}$
---	--	---

- delimited scope of dependencies:

$\frac{c : (\Gamma \vdash_d \Delta, \hat{t}p : A; \{\cdot \mid \cdot\})}{\Gamma \vdash \mu \hat{t}p.c : A \mid \Delta} \hat{t}p_I$		$\frac{B \in A_\sigma}{\Gamma \mid \hat{t}p : A \vdash_d \Delta, \hat{t}p : B; \sigma\{\cdot \mid p\}} \hat{t}p_E$
--	--	--

- Mission accomplished ✓
 - subject reduction ✓
 - normalization ✓
 - consistency as a logic ✓
 - CPS translation ✓
 - (Bonus) embedding into Rodolphe's calculus ✓
 - ↗ realizability interpretation

Rodolphe's calculus in a nutshell

Recipe:

- Call-by-value evaluation
- Classical language ($\mu\alpha.t$ control operator)
- Second-order logic, with encoding of dependent product:

$$\Pi_{(a:A)}B \triangleq \forall a(a \in A \rightarrow B)$$

- Semantical value restriction
- Soundness and type safety proved by a realizability model:

$$\Gamma \vdash t : A \quad \Rightarrow \quad \rho \Vdash \Gamma \quad \Rightarrow \quad t[\rho] \in \|A\|_{\rho}^{\perp\perp}$$

Semantical value restriction:

- observational equivalence: $t \equiv u$
- $u \in A$ restricted to values
- typing rules up to this equivalence (hence undecidable!)

Rodolphe's calculus in a nutshell

Recipe:

- Call-by-value evaluation
- Classical language ($\mu\alpha.t$ control operator)
- Second-order logic, with encoding of dependent product:

$$\Pi_{(a:A)}B \triangleq \forall a(a \in A \rightarrow B)$$

- Semantical value restriction
- Soundness and type safety proved by a realizability model:

$$\Gamma \vdash t : A \quad \Rightarrow \quad \rho \Vdash \Gamma \quad \Rightarrow \quad t[\rho] \in \|A\|_{\rho}^{\perp\perp}$$

Semantical value restriction:

- observational equivalence: $t \equiv u$
- $u \in A$ restricted to values
- typing rules up to this equivalence (hence undecidable!)

Embedding

Easy check:

NEF \subseteq semantical values

We define an embedding of proofs and types that:

- is **correct** with respect to typing

$$\Gamma \vdash p : A \mid \Delta \quad \Rightarrow \quad (\Gamma \cup \Delta)^* \vdash \llbracket p \rrbracket_p : A^*$$

- is **adequate** with his realizability model

$$\Gamma \vdash p : A \mid \Delta \quad \wedge \quad \sigma \Vdash (\Gamma \cup \Delta)^* \quad \Rightarrow \quad \llbracket p \rrbracket_p \sigma \in |A|$$

- allows to transfer Rodolphe's safety results

$$\not\vdash p : \perp$$

dLPA^ω: a sequent calculus with dependent types for classical arithmetic

dLPA^ω

A **classical sequent calculus** with:

- stratified **dependent types** :

- terms: $t, u ::= \dots \mid \text{wit } p$
- formulas: $A, B ::= \dots \mid \forall x^T. A \mid \exists x^T. A \mid \Pi_{(a:A)}. B \mid t = u$
- proofs: $p, q ::= \dots \mid \lambda x. p \mid (t, p) \mid \lambda a. p$

- a restriction to the **NEF fragment**
- **arithmetical** terms:

$$t, u ::= \dots \mid 0 \mid S(t) \mid \text{rec}_{xy}^t [t_0 \mid t_S] \mid \lambda x. t \mid t u$$

- **stores**:

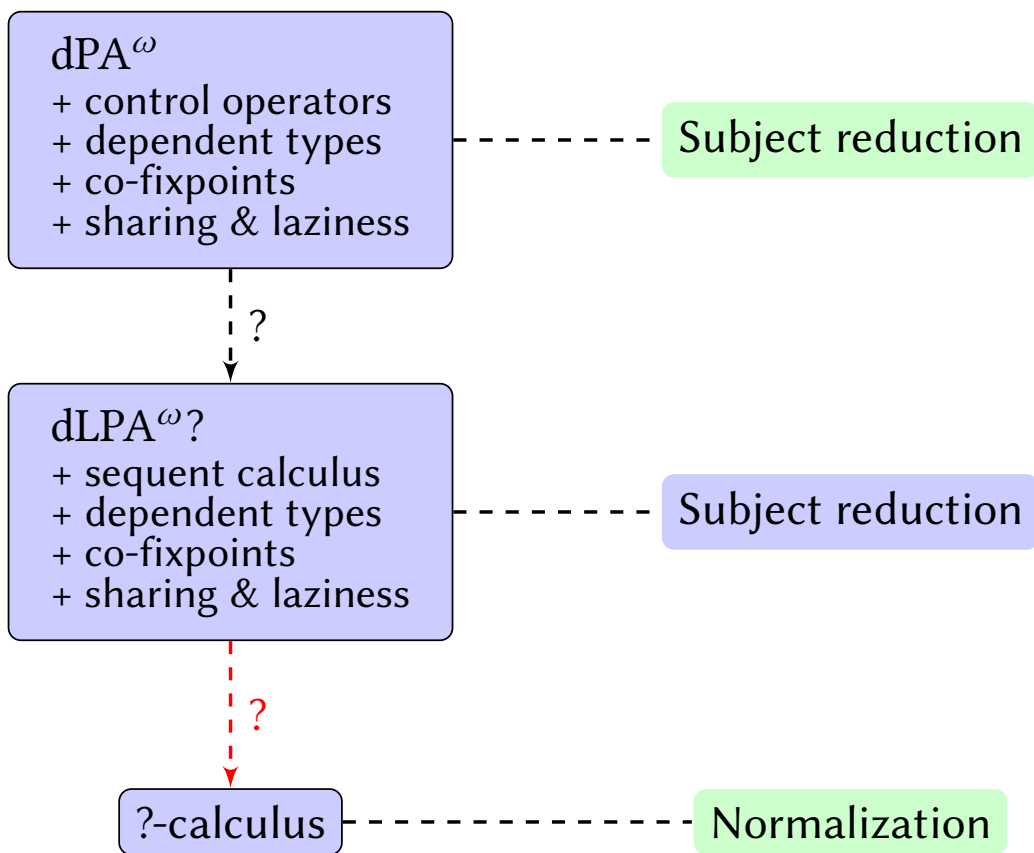
$$\tau ::= \varepsilon \mid \tau[a := p_\tau] \mid \tau[\alpha := e]$$

- inductive and **coinductive** constructions:

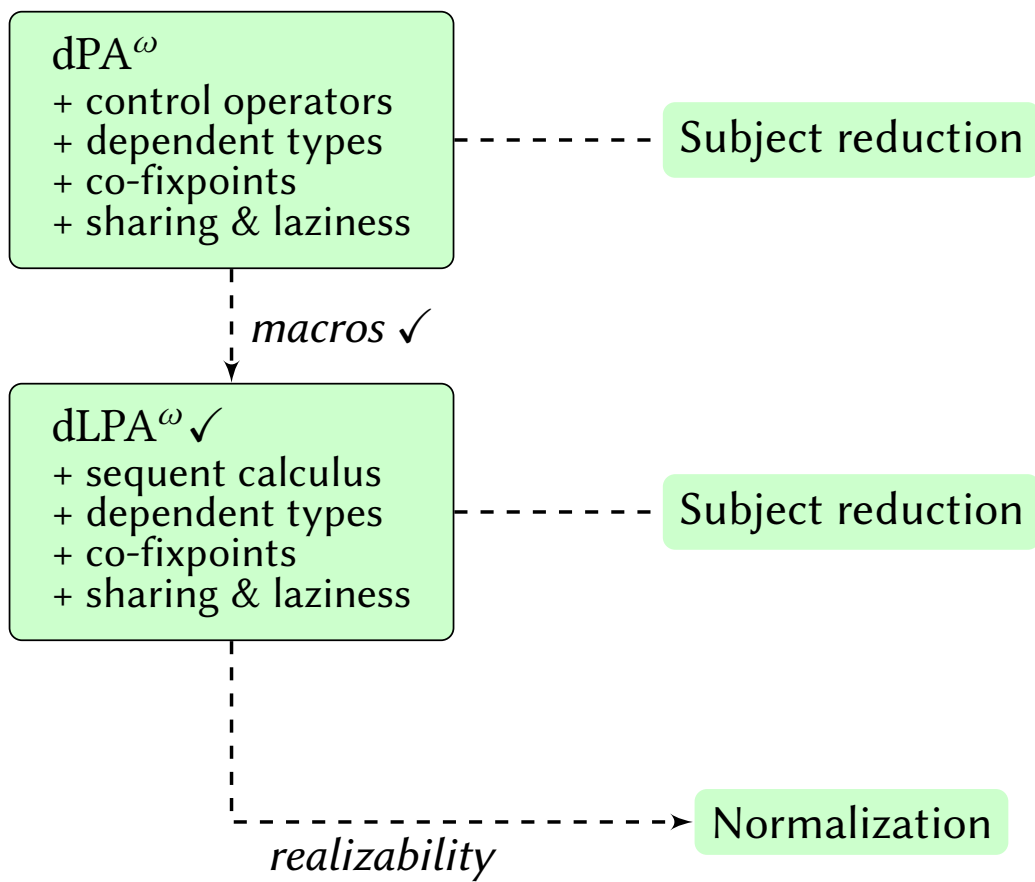
$$p, q ::= \dots \mid \text{fix}_{bn}^t [p \mid p] \mid \text{cofix}_{bn}^t p$$

- a **call-by-value** reduction and **lazy evaluation** of cofix

End of the road



End of the road



Realizability interpretation

Same methodology:

- 1 small-step reductions
- 2 derive the realizability interpretation

Resembles $\bar{\lambda}_{[lv\tau\star]}$ -interpretation, plus:

- dependent types from Rodolphe's calculus
- co-inductive formulas

Realizability interpretation

Same methodology:

- 1 small-step reductions
- 2 derive the realizability interpretation

Resembles $\bar{\lambda}_{[lv\tau\star]}$ -interpretation, plus:

- dependent types from Rodolphe's calculus:

$$\Pi_{(a:A)}.B \triangleq \forall a.(a \in A \rightarrow B)$$

- co-inductive formulas

Realizability interpretation

Same methodology:

- 1 small-step reductions
- 2 derive the realizability interpretation

Resembles $\bar{\lambda}_{[lv\tau\star]}$ -interpretation, plus:

- dependent types from Rodolphe's calculus
- co-inductive formulas: *by finite approximations*

$$\|v_{Xx}^t A\|_f \triangleq \bigcup_{n \in \mathbb{N}} \|F_{A,t}^n\|_f$$

Realizability interpretation

Same methodology:

- 1 small-step reductions
- 2 derive the realizability interpretation

Resembles $\bar{\lambda}_{[lv\tau\star]}$ -interpretation, plus:

- dependent types from Rodolphe's calculus
- co-inductive formulas: *by finite approximations*

Consequences of adequacy:

Normalization

If $\Gamma \vdash_{\sigma} c$, then c is normalizable.

Consistency

$\not\vdash_{\text{dLPA}^{\omega}} p : \perp$

Conclusion

What did we learn?

- classical call-by-need:
 - **realizability** interpretation
 - **typed** continuation-and-store-passing style translation
- dependent classical sequent calculus:
 - list of dependencies
 - use of **delimited continuations** for soundness
 - **dependently-typed** continuation-passing style translation
- dLPA^ω:
 - **soundness** and normalization,
 - realizability interpretation of co-fixpoints

Further work

1 Classical call-by-need:

- typing the CPS with Kripke forcing

2 $dL_{\hat{tp}}$:

- Connection with:
 - Pédrot-Tabareau's Baclofen Type Theory?
 - Vákár's categorical presentation?
 - Bowman *et. al.* CPS for CC?
- Dependent types & effects:

3 Realizability:

- Connection with realizer for DC using bar recursion?
- Algebraic counterpart of side-effects in realizability structures?

Further work

1 Classical call-by-need:

- typing the CPS with Kripke forcing

2 $dL_{\hat{tp}}$:

- Connection with:
 - Pédrot-Tabareau's Baclofen Type Theory?
 - Vákár's categorical presentation?
 - Bowman *et. al.* CPS for CC?

- Dependent types & effects:

3 Realizability:

- Connection with realizer for DC using bar recursion?
- Algebraic counterpart of side-effects in realizability structures?

Further work

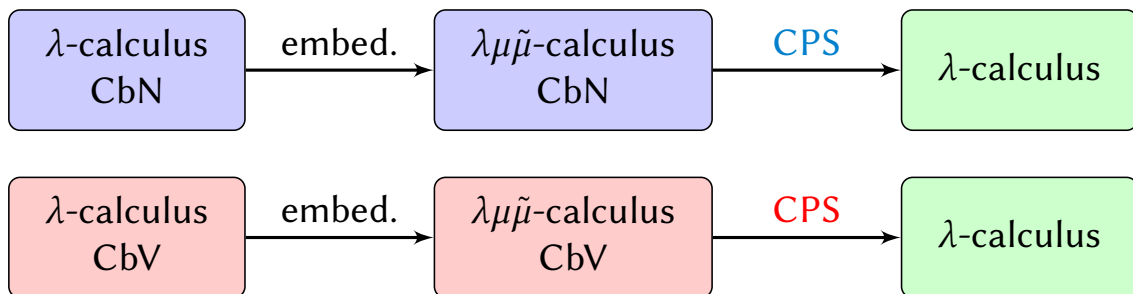
1 **Classical call-by-need:**

- typing the CPS with Kripke forcing

2 **dL_{tp}^ω:**

- Connection with:
 - Pédrot-Tabareau’s Baclofen Type Theory?
 - Vákár’s categorical presentation?
 - Bowman *et. al.* CPS for CC?

• **Dependent types & effects:**

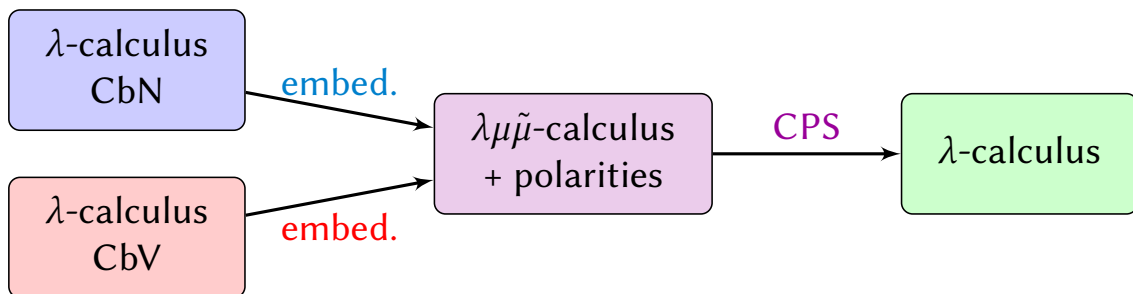


3 **Realizability:**

- Connection with realizer for DC using bar recursion?
- Algebraic counterpart of side-effects in realizability structures?

Further work

- 1 **Classical call-by-need:**
 - typing the CPS with Kripke forcing
- 2 **dL_{tp}^ω:**
 - Connection with:
 - Pédrot-Tabareau’s Baclofen Type Theory?
 - Vákár’s categorical presentation?
 - Bowman *et. al.* CPS for CC?
 - **Dependent types & effects:**



- 3 **Realizability:**
 - Connection with realizer for DC using bar recursion?
 - Algebraic counterpart of side-effects in realizability structures?

Further work

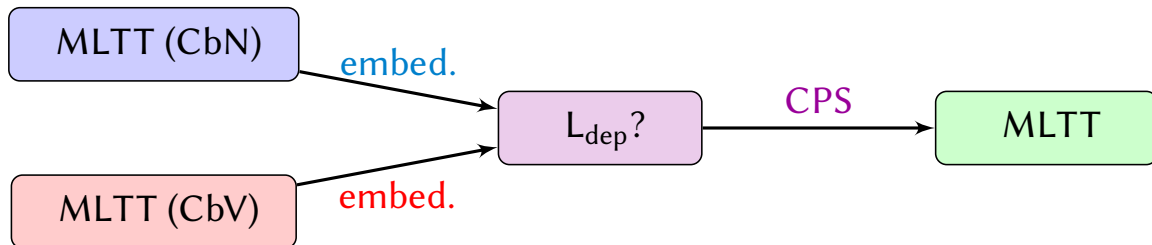
1 **Classical call-by-need:**

- typing the CPS with Kripke forcing

2 **dL_{tp}^ω:**

- Connection with:
 - Pédrot-Tabareau’s Baclofen Type Theory?
 - Vákár’s categorical presentation?
 - Bowman *et. al.* CPS for CC?

- **Dependent types & effects:**



3 **Realizability:**

- Connection with realizer for DC using bar recursion?
- Algebraic counterpart of side-effects in realizability structures?

Further work

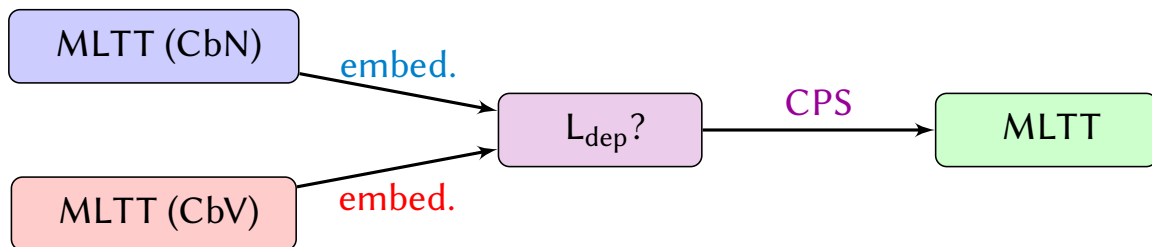
1 **Classical call-by-need:**

- typing the CPS with Kripke forcing

2 **dL_{tp}^ω:**

- Connection with:
 - Pédrot-Tabareau’s Baclofen Type Theory?
 - Vákár’s categorical presentation?
 - Bowman *et. al.* CPS for CC?

- **Dependent types & effects:**



3 **Realizability:**

- Connection with realizer for DC using bar recursion?
- Algebraic counterpart of side-effects in realizability structures?

A constructive proof of DC
○○○○○○○○

Semantic artifacts
○○○○○○○○○○

Classical call-by-need
○○○○

dL
○○○○○○○○○○

dLPA^ω
○○○○

Thank you for you attention.