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
$\dfrac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\Rightarrow_E)$	$\dfrac{\Gamma \vdash t : A \rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash t\,u : B}\,(\rightarrow_E)$

Benefits:

Program your proofs!	*Prove your programs!*

Proofs-as-programs

Limitations

Mathematics	Computer Science
$A \lor \neg A$	`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

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

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

Classical Curry-Howard:

λ-calculus + `call/cc`

Other examples:

- quote instruction $\sim$ dependent choice
- monotonic memory $\sim$ 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 $\sim$ dependent choice
- monotonic memory $\sim$ 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:

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

The axiom of choice

Axiom of Choice:

$$\begin{array}{rcl} AC & : & \forall x^A.\exists y^B.P(x,y) \to \exists f^{A\to B}.\forall x^A.P(x,f(x)) \\ & := & \lambda H.(\lambda x.\mathtt{wit}\ (H\,x), \lambda x.\mathtt{prf}\ (H\,x)) \end{array}$$

Computational content through **dependent types:**

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

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

Incompatibility with classical logic

Bad news

dependent sum + classical logic = ☠

Choice:

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

Excluded-middle:

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

Take U undecidable:

$$\vdash t\,s : \exists f \in \{0,1\}^X.\forall x \in X.(U(x) \land f(x) = 1) \lor (\neg U(x) \land f(x) = 0)$$

$\looparrowright$ *i.e.* $\mathtt{wit}(t\,s)$ *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:

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

↬ *i.e.* $\texttt{wit}(t\,s)$ *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:

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

and reach a contradiction:

$$(\texttt{wit}\, H_0, \texttt{prf}\, H_0) \to \underbrace{(1, \overbrace{\;p\;}^{0=0})}_{\cancel{\exists x. x=0}}$$

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:

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

and reach a contradiction:

$$(\texttt{wit}\, H_0, \texttt{prf}\, H_0) \rightarrow \underbrace{({\color{red}1}, \overbrace{\ p\ }^{{\color{red}0}=0})}_{\cancel{\exists x.x=0}}$$

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:

$$\begin{aligned} AC := \lambda H.(\lambda n.&\texttt{if } n = 0 \texttt{ then } \texttt{wit}(H\ 0) \texttt{ else} \\ &\texttt{if } n = 1 \texttt{ then } \texttt{wit}(H\ 1) \texttt{ else } \dots, \\ \lambda n.&\texttt{if } n = 0 \texttt{ then } \texttt{prf}(H\ 0) \texttt{ else} \\ &\texttt{if } n = 1 \texttt{ then } \texttt{prf}(H\ 1) \texttt{ else } \dots) \end{aligned}$$

Toward a solution ?

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

- Restriction to countable choice:

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

- Proof:

$$\begin{aligned} AC_{\mathbb{N}} := \lambda H.&\mathtt{let}\ H_0 = H\ 0\ \mathtt{in} \\ &\mathtt{let}\ H_1 = H\ 1\ \mathtt{in} \\ &\ldots \\ &(\lambda n.\mathtt{if}\ n = 0\ \mathtt{then}\ \mathtt{wit}\ H_0\ \mathtt{else} \\ &\qquad \mathtt{if}\ n = 1\ \mathtt{then}\ \mathtt{wit}\ H_1\ \mathtt{else}\ \ldots, \\ &\ \lambda n.\mathtt{if}\ n = 0\ \mathtt{then}\ \mathtt{prf}\ H_0\ \mathtt{else} \\ &\qquad \mathtt{if}\ n = 1\ \mathtt{then}\ \mathtt{prf}\ H_1\ \mathtt{else}\ \ldots) \end{aligned}$$

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:

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

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:

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

dPA$^\omega$ (Herbelin's recipe)

A proof system:

- **classical**:

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

- with stratified **dependent types** :
  - terms: $t,u ::= \dots \mid \mathtt{wit}\, p$
  - 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**:

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

- with inductive and **coinductive** constructions:

$$p,q ::= \dots \mid \mathtt{fix}^t_{bn}[p_0 \mid p_S] \mid \mathtt{cofix}^t_{bn}\, p$$

- **lazy evaluation** for the cofix

dPA$^\omega$ (Herbelin's recipe)

A proof system:

- **classical**:

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

- with stratified **dependent types** :
  - terms: $t,u ::= \dots \mid \mathtt{wit}\, p$
  - 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**:

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

- with inductive and **coinductive** constructions:

$$p,q ::= \dots \mid \mathtt{fix}^t_{bn}[p_0 \mid p_S] \mid \mathtt{cofix}^t_{bn}\ p$$

- **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

$$\nvdash_{dPA^\omega} \bot$$

Roadmap

dPA$^\omega$ [Herbelin'12]:
+ *control operators*
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

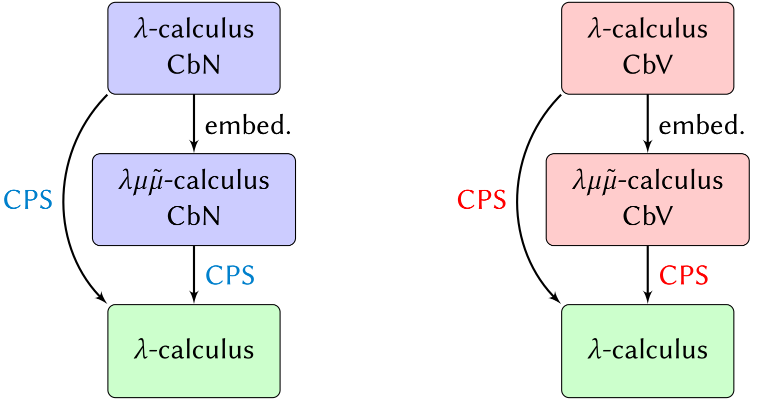
CPS-translation?

?-calculus

Normalization

Roadmap

Remark: *CPS usually factorize through sequent calculi!*



Roadmap

dPA$^\omega$ [Herbelin'12]:
+ *control operators*
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

typing/reduction preservation

dLPA$^\omega$?
+ *sequent calculus*
+ dependent types
+ co-fixpoints
+ sharing & laziness

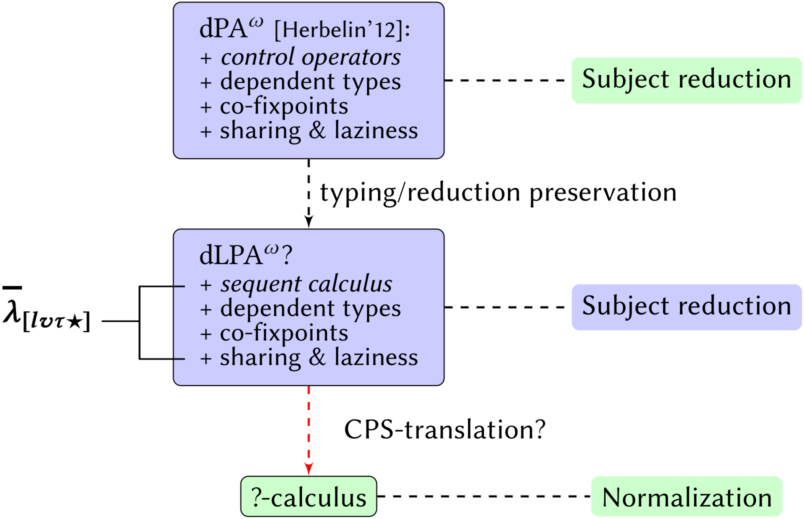
Subject reduction

CPS-translation?

?-calculus

Normalization

Roadmap


dPA$^\omega$ [Herbelin'12]:
+ *control operators*
+ dependent types
+ co-fixpoints
+ sharing & laziness
Subject reduction
typing/reduction preservation
dLPA$^\omega$?
+ *sequent calculus*
+ dependent types
+ co-fixpoints
+ sharing & laziness
$\bar{\lambda}_{[lv\tau\star]}$
Subject reduction
CPS-translation?
?-calculus
Normalization

Roadmap

dPA$^\omega$ [Herbelin'12]:
+ *control operators*
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

typing/reduction preservation

dLPA$^\omega$?
+ *sequent calculus*
+ dependent types
+ co-fixpoints
+ sharing & laziness

dL

Subject reduction

CPS-translation?

?-calculus

Normalization

Danvy's semantic artifacts

CPS translation

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

- 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 λ^{machin} 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: $[\![\cdot]\!] : \mathit{source} \to \lambda^{\text{machin}}$

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

Benefits

If λ^{machin} 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:

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

Typing rules:

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

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

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

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

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

Typing rules:

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

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

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

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

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

Typing rules:

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

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

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

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

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

Reduction:

$$\begin{array}{rcl}
\langle \lambda a.p \parallel q \cdot e \rangle & \to & \langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle \rangle \\
\langle p \parallel \tilde{\mu}a.c \rangle & \to & c[p/a] \\
\langle \mu\alpha.c \parallel e \rangle & \to & c[e/\alpha]
\end{array}$$

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

Critical pair:

$$\begin{array}{ccc}
& \langle \mu\alpha.c \parallel \tilde{\mu}a.c' \rangle & \\
\swarrow & & \searrow \\
c[\tilde{\mu}a.c'/\alpha] & & c'[\mu\alpha.c/a]
\end{array}$$

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

The duality of computation
Curien/Herbelin (2000)

Syntax:

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

Reduction:

$$\begin{array}{rcll} \langle \lambda a.p \,\|\, q \cdot e \rangle & \to & \langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, e \rangle \rangle & \\ \langle p \,\|\, \tilde{\mu}a.c \rangle & \to & c[p/a] & p \in \mathcal{P} \\ \langle \mu\alpha.c \,\|\, e \rangle & \to & c[e/\alpha] & e \in \mathcal{E} \end{array}$$

Critical pair:

$$\begin{array}{ccc} & \langle \mu\alpha.c \,\|\, \tilde{\mu}a.c' \rangle & \\ {}^{\text{CbV}}\swarrow & & \searrow^{\text{CbN}} \\ c[\tilde{\mu}a.c'/\alpha] & & c'[\mu\alpha.c/a] \end{array}$$

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

Syntax:

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

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

Reduction rules:

$$\begin{array}{rcl} \langle p \| \tilde{\mu}a.c \rangle & \to & c[p/a] \\ \langle \mu\alpha.c \| E \rangle & \to & c[E/\alpha] \\ \langle \lambda a.p \| q \cdot e \rangle & \to & \langle q \| \tilde{\mu}a.\langle p \| e \rangle \rangle \end{array}$$

Semantic artifacts

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

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

Small steps

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

Semantic artifacts

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

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

	Small steps			CPS
e	$\langle p \| \tilde\mu a.c\rangle_e$	$\rightsquigarrow$	$c_e[p/a]$	$[\![\tilde\mu a.c]\!]_e\, p \triangleq (\lambda a.[\![c]\!]_c)\, p$
	$\langle p \| E\rangle_e$	$\rightsquigarrow$	$\langle p \| E\rangle_p$	$[\![E]\!]_e\, p \triangleq p\, [\![E]\!]_E$
p	$\langle \mu\alpha.c \| E\rangle_p$	$\rightsquigarrow$	$c_e[E/\alpha]$	$[\![\mu\alpha.c]\!]_p\, E \triangleq (\lambda\alpha.[\![c]\!]_c)\, E$
	$\langle V \| E\rangle_p$	$\rightsquigarrow$	$\langle V \| E\rangle_E$	$[\![V]\!]_p\, E \triangleq E\, [\![V]\!]_V$
E	$\langle V \| q \cdot e\rangle_E$	$\rightsquigarrow$	$\langle V \| q \cdot e\rangle_V$	$[\![q \cdot e]\!]_E\, V \triangleq V\, [\![q]\!]_p\, [\![e]\!]_e$
V	$\langle \lambda a.p \| q \cdot e\rangle_V$	$\rightsquigarrow$	$\langle q \| \tilde\mu a.\langle p \| e\rangle\rangle_e$	$[\![\lambda a.p]\!]_V\, q\, e \triangleq (\lambda a.e\, [\![p]\!]_p)\, q$

Semantic artifacts

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

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

Small steps

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

CPS

$$
\begin{array}{l}
[\![\tilde\mu a.c]\!]_e\, p \triangleq (\lambda a.[\![c]\!]_c)\, p \\
[\![E]\!]_e\, p \triangleq p\, [\![E]\!]_E \\
[\![\mu\alpha.c]\!]_p\, E \triangleq (\lambda\alpha.[\![c]\!]_c)\, E \\
[\![V]\!]_p\, E \triangleq E\, [\![V]\!]_V \\
[\![q \cdot e]\!]_E\, V \triangleq V\, [\![q]\!]_p\, [\![e]\!]_e \\
[\![\lambda a.p]\!]_V\, q\, e \triangleq (\lambda a.e\, [\![p]\!]_p)\, q
\end{array}
$$

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

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

e: $[\![\tilde\mu a.c]\!]_e\, p \triangleq (\lambda a.[\![c]\!]_c)\, p$

$[\![E]\!]_e\, p \triangleq p\, [\![E]\!]_E$

p: $[\![\mu\alpha.c]\!]_p\, E \triangleq (\lambda\alpha.[\![c]\!]_c)\, E$

$[\![V]\!]_p\, E \triangleq E\, [\![V]\!]_V$

E: $[\![q \cdot e]\!]_E\, V \triangleq V\, [\![q]\!]_p\, [\![e]\!]_e$

V: $[\![\lambda a.p]\!]_V\, q\, e \triangleq (\lambda a.e\, [\![p]\!]_p)\, q$

Types translation

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

$[\![A]\!]_p \triangleq [\![A]\!]_E \to \bot$

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

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

$$\Gamma \vdash p : A \mid \Delta \quad \Rightarrow \quad [\![\Gamma]\!]_p, [\![\Delta]\!]_E \vdash [\![p]\!]_p : [\![A]\!]_p$$

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 $\vdash t : [\![\bot]\!]_p$.

Proof. $[\![\bot]\!]_p = (\bot \to \bot) \to \bot$ *and* $\lambda x.x$ *is of type* $\bot \to \bot$. □

Soundness

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

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

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

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

Truth value defined by **orthogonality** :

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

Realizability *à la* Krivine (1/2)

Intuition

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

$$\langle p \| e \rangle > c_0 > \cdots > c_n \in ⫫?$$

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

Truth value defined by **orthogonality** :

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

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

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

$\rightsquigarrow \perp\!\!\!\perp \subset \Lambda \star \Pi$ 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 $\perp\!\!\!\perp$: commands, referee

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

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

Truth value defined by **orthogonality** :

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

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

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

$\rightsquigarrow \perp\!\!\!\perp \subset \Lambda \star \Pi$ 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 $\perp\!\!\!\perp$: commands, referee

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

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

Truth value defined by **orthogonality** :

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

Semantic artifacts++

(Terms)	$p ::= \mu\alpha.c \mid a \mid V$	(Contexts)	$e ::= \tilde\mu a.c \mid E$
(Values)	$V ::= \lambda a.p$	(Co-values)	$E ::= \alpha \mid p \cdot e$

Small steps

e	$\langle p \parallel \tilde\mu a.c\rangle_e$	$\rightsquigarrow$	$c_e[p/a]$
	$\langle p \parallel E\rangle_e$	$\rightsquigarrow$	$\langle p \parallel E\rangle_p$
p	$\langle \mu\alpha.c \parallel E\rangle_p$	$\rightsquigarrow$	$c_e[E/\alpha]$
	$\langle V \parallel E\rangle_p$	$\rightsquigarrow$	$\langle V \parallel E\rangle_E$
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++

(Terms)	$p ::= \mu\alpha.c \mid a \mid V$	(Contexts)	$e ::= \tilde\mu a.c \mid E$
(Values)	$V ::= \lambda a.p$	(Co-values)	$E ::= \alpha \mid p \cdot e$

Small steps

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

Realizability

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

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

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

Extension to second-order

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

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

(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.:

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

Truth value (player):

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

Falsity value (opponent):

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

Realizability *à la* Krivine (2/2)

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

Definition (Pole)

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

$$\forall c, c', (c' \in ⫫ \wedge c \to c') \quad \Rightarrow \quad c \in ⫫$$

Truth value (player):

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

Falsity value (opponent):

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

Realizability *à la* Krivine (2/2)

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

Definition (Pole)

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

$$\forall c, c', (c' \in ⫫ \land c \to c') \quad \Rightarrow \quad c \in ⫫$$

Truth value (player):

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

Falsity value (opponent):

$$\begin{array}{rcl}
\|\dot{F}(e_1,\ldots,e_k)\|_E & = & F([\![e_1]\!],\ldots,[\![e_k]\!]) \\
\|A \to B\|_E & = & \{q \cdot e : \ q \in |A|_p \land e \in \|B\|_e\} \\
\|\forall x.A\|_E & = & \bigcup_{n\in\mathbb{N}} \|A[n/x]\|_E \\
\|\forall X.A\|_E & = & \bigcup_{F:\mathbb{N}^k \to \mathcal{P}(\Pi)} \|A[\dot{F}/X]\|_E \\
|A|_p & = & \|A\|_E{}^{⫫} \ = \ \{p : \ \forall e \in \|A\|_E, \langle p \,\|\, e\rangle \in ⫫\} \\
\|A\|_e & = & |A|_p{}^{⫫} \ = \ \{e : \ \forall p \in |A|_p, \langle p \,\|\, e\rangle \in ⫫\}
\end{array}$$

Adequacy

Valuation ρ:

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

Substitution σ:

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

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

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. $c : (\Gamma \vdash \Delta) \;\Rightarrow\; c[\sigma] \in \perp\!\!\perp$

Proof. By mutual induction over the typing derivation. □

Results

Normalizing commands

$\perp\!\!\!\perp_{\Downarrow} \triangleq \{c : c \text{ normalizes}\}$ 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 $\perp\!\!\!\perp_{\Downarrow}$. □

Soundness

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

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

Classical call-by-need

Reminder

dPA$^\omega$ [Herbelin'12]:
+ *control operators*
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

typing/reduction preservation

dLPA$^\omega$?
+ *sequent calculus*
+ dependent types
+ co-fixpoints
+ sharing & laziness

$\overline{\lambda}_{[lv\tau\star]}$

Subject reduction

CPS-translation?

?-calculus

Normalization

Classical call-by-need

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

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

Syntax:

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

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

Reduction rules:

$$
\begin{array}{lrcl}
\textit{(Lazy storage)} & \langle p \| \tilde\mu a.c\rangle\tau & \to & c\tau[a := p] \\
 & \langle \mu\alpha.c \| E\rangle\tau & \to & (c[E/\alpha])\tau \\
\textit{(Lookup)} & \langle a \| F\rangle\tau[a := \boldsymbol{p}]\tau' & \to & \langle \boldsymbol{p} \| \tilde\mu[a].\langle a \| F\rangle\tau'\rangle\tau \\
\textit{(Forced eval.)} & \langle \boldsymbol{V} \| \tilde\mu[a].\langle a \| F\rangle\tau'\rangle\tau & \to & \langle \boldsymbol{V} \| F\rangle\tau[a := \boldsymbol{V}]\tau' \\
 & \langle \lambda a.p \| q \cdot E\rangle\tau & \to & \langle q \| \tilde\mu a.\langle p \| E\rangle\rangle\tau
\end{array}
$$

Semantic artifacts

Small steps:

$$
\begin{array}{llcl}
e & \langle p \,\|\, \tilde\mu a.c\rangle_e \tau & \to & c_e \tau[a := p] \\
 & \langle p \,\|\, E\rangle_e \tau & \to & \langle p \,\|\, E\rangle_p \tau \\
p & \langle \mu\alpha.c \,\|\, E\rangle_p \tau & \to & (c[E/\alpha])\tau \\
 & \langle V \,\|\, E\rangle_p \tau & \to & \langle V \,\|\, E\rangle_E \tau \\
E & \langle V \,\|\, \tilde\mu[a].\langle a \,\|\, F\rangle\tau'\rangle_E \tau & \to & \langle V \,\|\, F\rangle_V \tau[a := V]\tau' \\
 & \langle V \,\|\, F\rangle_E \tau & \to & \langle V \,\|\, F\rangle_V \tau \\
V & \langle a \,\|\, F\rangle_V \tau[a := p]\tau' & \to & \langle p \,\|\, \tilde\mu[a].\langle a \,\|\, F\rangle\tau'\rangle_p \tau \\
 & \langle \lambda a.p \,\|\, F\rangle_V \tau & \to & \langle \lambda a.p \,\|\, F\rangle_F \tau \\
F & \langle \lambda a.p \,\|\, q \cdot E\rangle_F \tau & \to & \langle q \,\|\, \tilde\mu a.\langle p \,\|\, E\rangle\rangle_e \tau
\end{array}
$$

Semantic artifacts

CPS :

$$
\begin{array}{rcl}
[\![\langle p \,\|\, e\rangle\tau]\!] & := & [\![e]\!]_e\ [\![\tau]\!]_\tau\ [\![p]\!]_p \\
\\
e \qquad [\![\tilde\mu a.c]\!]_e & := & \lambda\tau p.[\![c]\!]\ \tau[a := p] \\
[\![E]\!]_e & := & \lambda\tau p.p\ \tau\ [\![E]\!]_E \\
\\
p \qquad [\![\mu\alpha.c]\!]_p & := & \lambda\tau E.([\![c]\!]_c\ \tau)[E/\alpha] \\
[\![V]\!]_p & := & \lambda\tau E.E\ \tau\ [\![V]\!]_v \\
\\
E \qquad [\![\tilde\mu[a].\langle a \,\|\, F\rangle\tau']\!]_E & := & \lambda\tau V.V\ \tau[a := V]\tau'\ [\![F]\!]_F \\
[\![F]\!]_E & := & \lambda\tau V.V\ \tau\ [\![F]\!]_F \\
\\
V \qquad [\![a]\!]_v & := & \lambda\tau F.\tau(a)\ \tau\ (\lambda\tau V.V\ \tau[a := V]\tau'\ [\![F]\!]_F) \\
[\![\lambda a.p]\!]_v & := & \lambda\tau F.F\ \tau\ (\lambda q\tau E.[\![p]\!]_p\ \tau[a := q]\ E) \\
\\
F \qquad [\![q \cdot E]\!]_F & := & \lambda\tau v.v\ [\![q]\!]_p\ \tau\ [\![E]\!]_E
\end{array}
$$

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	$\langle \lambda a.p \parallel q \cdot E \rangle_v \tau$	$\rightarrow$	$\dots$

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:

$$(\perp\!\!\!\perp \subseteq \quad ? \quad)$$

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

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

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

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

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

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

Semantic artifacts

Small-step:

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

Realizability:

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

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

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

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

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

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

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

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:

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

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

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

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

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

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

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

Realizability interpretation

A few novelties:

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

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

- **Pole** : set of closures $⫫$ which is:
  - *saturated*:

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

  - *closed by store extension*:

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

- **Orthogonality** :

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

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

Realizability interpretation

A few novelties:

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

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

- **Pole** : set of closures $⫫$ which is:
  - *saturated*:

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

  - *closed by store extension*:

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

- **Orthogonality** :

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

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

Realizability interpretation

Adequacy

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

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.*

Realizability interpretation

Adequacy

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

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.*

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

dPA$^\omega$ [Herbelin'12]:
+ *control operators*
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

typing/reduction preservation

dL ?

dLPA$^\omega$?
+ *sequent calculus*
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

CPS-translation?

?-calculus

Normalization

A classical sequent calculus with dependent types

Can this work?

$$\dfrac{\dfrac{\begin{array}{c}\Pi_p \\ \vdots \\ \Gamma, a : A \vdash p : B[a] \mid \Delta\end{array}}{\Gamma \vdash \lambda a.p : \Pi_{(a:A)}.B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{array}{c}\Pi_q \\ \vdots \\ \Gamma \vdash q : A \mid \Delta\end{array} \quad \begin{array}{c}\Pi_e \\ \vdots \\ \Gamma \mid e : B[q] \vdash \Delta\end{array} \quad q \in V}{\Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta}\,(\to_l)}{\langle \lambda a.p \,\|\, q \cdot e \rangle : (\Gamma \vdash \Delta)}\,(\textsc{Cut})$$

$$\longrightarrow$$

A classical sequent calculus with dependent types

Can this work?

$$\dfrac{\dfrac{\begin{array}{c}\Pi_p\\ \vdots\\ \Gamma, a:A \vdash p : B[a] \mid \Delta\end{array}}{\Gamma \vdash \lambda a.p : \Pi_{(a:A)}.B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{array}{c}\Pi_q\\ \vdots\\ \Gamma\vdash q:A\mid\Delta\end{array} \quad \begin{array}{c}\Pi_e\\ \vdots\\ \Gamma \mid e : B[q]\vdash\Delta\end{array}\quad q\in V}{\Gamma \mid q\cdot e : \Pi_{(a:A)}.B \vdash \Delta}\,(\to_l)}{\langle \lambda a.p \,\|\, q\cdot e\rangle : (\Gamma\vdash\Delta)}\,(\textsc{Cut})$$

$$\longrightarrow$$

$$\dfrac{\begin{array}{c}\Pi_q\\ \vdots\\ \Gamma\vdash q:A\mid\Delta\end{array}\qquad \dfrac{\dfrac{\Gamma,a:A\vdash p:\cancel{B[a]}\mid\Delta \quad \Gamma,a:A\mid e:\cancel{B[q]}\vdash\Delta}{\langle p\,\|\,e\rangle:(\Gamma,a:A\vdash\Delta)}\ \textit{Mismatch}}{\Gamma\mid\tilde\mu a.\langle p\,\|\,e\rangle : A\vdash\Delta}\,(\tilde\mu)}{\langle q\,\|\,\tilde\mu a.\langle p\,\|\,e\rangle\rangle:(\Gamma\vdash\Delta)}\,(\textsc{Cut})$$

A classical sequent calculus with dependent types

Can this work? ✓

$$\dfrac{\dfrac{\begin{array}{c}\Pi_p\\ \vdots\\ \Gamma, a : A \vdash p : B[a] \mid \Delta\end{array}}{\Gamma \vdash \lambda a.p : \Pi_{(a:A)}.B \mid \Delta}\,(\to_r) \quad \dfrac{\begin{array}{c}\Pi_q\\ \vdots\\ \Gamma \vdash q : A \mid \Delta\end{array} \quad \begin{array}{c}\Pi_e\\ \vdots\\ \Gamma \mid e : B[q] \vdash \Delta\end{array} \quad q \in V}{\Gamma \mid q \cdot e : \Pi_{(a:A)}.B \vdash \Delta}\,(\to_l)}{\langle \lambda a.p \,\|\, q \cdot e \rangle : (\Gamma \vdash \Delta)}\,(\textsc{Cut})$$

$$\longrightarrow$$

$$\dfrac{\begin{array}{c}\Pi_q\\ \vdots\\ \Gamma \vdash q : A \mid \Delta\end{array} \quad \dfrac{\dfrac{\Gamma, a : A \vdash p : B[a] \mid \Delta \quad \Gamma, a : A \mid e : B[q] \vdash \Delta; \{\cdot|p\}\{a|q\}}{\langle p \,\|\, e \rangle : \Gamma, a : A \vdash \Delta; \{a|q\}}\,(\textsc{Cut})}{\Gamma \mid \tilde\mu a.\langle p \,\|\, e \rangle : A \vdash \Delta; \{.|q\}}\,(\tilde\mu)}{\langle q \,\|\, \tilde\mu a.\langle p \,\|\, e \rangle \rangle : (\Gamma \vdash \Delta); \{\cdot|\cdot\}}\,(\textsc{Cut})$$

dL

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

- a **list of dependencies**:

$$\frac{\Gamma \vdash p : A \mid \Delta; \sigma \qquad \Gamma \mid e : A' \vdash \Delta; \sigma\{\cdot|p\} \qquad A' \in A_\sigma}{\langle p \| e \rangle : (\Gamma \vdash \Delta; \sigma)} \ (\textsc{Cut})$$

- 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**:

$$\frac{\Gamma \vdash p : A \mid \Delta; \sigma \qquad \Gamma \mid e : A' \vdash \Delta; \sigma\{\cdot|p\} \qquad A' \in A_\sigma}{\langle p \,\|\, e\rangle : (\Gamma \vdash \Delta; \sigma)}\ (\textsc{Cut})$$

- 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**:

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

- a **value restriction**

Is it enough?

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

$$[\![q]\!]\, [\![\tilde{\mu}a.\langle p \| e \rangle]\!] \;=\; \underbrace{[\![q]\!]}_{\neg\neg A}\,(\lambda a.\, \underbrace{[\![p]\!]}_{\neg\neg B(a)}\, \underbrace{[\![e]\!]}_{\neg B(q)})$$

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 ?

$$[\![\langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, e\rangle\rangle]\!] = [\![q]\!]\,(\lambda a.[\![p]\!]\,[\![e]\!])$$

Motto: $[\![p]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![q]\!]$ *has reduced*

$$([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!]$$

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 ?

$$[\![\langle q \,\|\, \tilde\mu a.\langle p \,\|\, e\rangle\rangle]\!] \;=\; [\![q]\!]\,(\lambda a.[\![p]\!]\,[\![e]\!])$$

Motto: $[\![p]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![q]\!]$ *has reduced*

$$([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

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 ?

$$[\![\langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, e\rangle\rangle]\!] = [\![q]\!]\,(\lambda a.[\![p]\!]\,[\![e]\!])$$

Motto: $[\![p]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![q]\!]$ *has reduced*

$$([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

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 ?

$$[\![\langle q \parallel \tilde{\mu}a.\langle p \parallel e\rangle\rangle]\!] = [\![q]\!]\,(\lambda a.[\![p]\!]\,[\![e]\!])$$

Motto: $[\![p]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![q]\!]$ *has reduced*

$$([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

So, we're looking for:

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

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 ?

$$[\![\langle q \,\|\, \tilde\mu a.\langle p \,\|\, e\rangle\rangle]\!] = [\![q]\!]\,(\lambda a.[\![p]\!]\,[\![e]\!])$$

Motto: $[\![p]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![q]\!]$ *has reduced*

$$([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

So, we're looking for:

$$\langle \lambda a.p \,\|\, q \cdot e\rangle \to \langle \mu\hat{\mathrm{tp}}.\langle q \,\|\, \tilde\mu a.\langle p \,\|\, \hat{\mathrm{tp}}\rangle\rangle \,\|\, e\rangle$$

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e \rangle]\!] \xrightarrow{?} ([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

Questions:

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

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ?

- If q eventually gives a value V:

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda a.[\![p]\!])[\![V]\!])[\![e]\!] \to [\![p]\!][[\![V]\!]/a][\![e]\!] = [\![p[V/a]]\!][\![e]\!] \quad ✓$$

- If $[\![q]\!] \to \lambda\_.t$ and drops its continuation (meaning $t : \bot$):

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda\_.t)\lambda a.[\![p]\!])[\![e]\!] \to t[\![e]\!] \quad ✗$$

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ?

- If q eventually gives a value V:

$$([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda a.[\![p]\!])[\![V]\!])[\![e]\!] \to [\![p]\!][[\![V]\!]/a][\![e]\!] = [\![p[V/a]]\!][\![e]\!] \quad ✓$$

- If $[\![q]\!] \to \lambda\_.t$ and drops its continuation (meaning $t : \bot$):

$$([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda\_.t)\lambda a.[\![p]\!])[\![e]\!] \to t[\![e]\!] \quad ✗$$

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

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

- If q eventually gives a value V:

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda a.[\![p]\!])[\![V]\!])[\![e]\!] \to [\![p]\!][[\![V]\!]/a][\![e]\!] = [\![p[V/a]]\!][\![e]\!] \quad ✓$$

- If $[\![q]\!] \to \lambda\_.t$ and drops its continuation (meaning $t : \bot$):

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda\_.t)\lambda a.[\![p]\!])[\![e]\!] \to t[\![e]\!] \quad ✗$$

Negative-elimination free (Herbelin'12)

Values + one continuation variable + no application

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

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

Naive attempt:

$$(\underbrace{[\![q]\!]}_{(A\to\bot)\to\bot}\ (\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))\ \underbrace{[\![e]\!]}_{\neg B[q]}$$

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e \rangle]\!] \stackrel{?}{\longrightarrow} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

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

Naive attempt:

$$\underbrace{(\underbrace{[\![q]\!]}_{(A\to ?)\to ?}\ (\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))}_{\neg\neg B(q)}\ \underbrace{[\![e]\!]}_{\neg B[q]}$$

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

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

Friedman's trick:

$$\underbrace{(\underbrace{[\![q]\!]}_{\forall R.(A\to R?)\to R?}\ (\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))}_{\neg\neg B}\ \underbrace{[\![e]\!]}_{\neg B[q]}$$

Toward a CPS translation (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

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

Better:

$$\underbrace{(\underbrace{[\![q]\!]}_{\forall R.(\Pi_{(a:A)}R(a))\to R(q)}\ (\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))}_{\neg\neg B(q)}\ \underbrace{[\![e]\!]}_{\neg B[q]}$$

(*Remark: not possible without* $q \in$ NEF)

$\mathrm{dL}_{\hat{tp}}$

An extension of dL with:

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

$\mathrm{dL}_{\hat{\mathrm{tp}}}$

An extension of dL with:

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

Reduction rules:

$$\begin{array}{rcll}
\langle \mu\hat{\mathrm{tp}}.\langle p \,\|\, \hat{\mathrm{tp}}\rangle \,\|\, e\rangle & \to & \langle p \,\|\, e\rangle & \\
c \to c' \Rightarrow \langle \mu\hat{\mathrm{tp}}.c \,\|\, e\rangle & \to & \langle \mu\hat{\mathrm{tp}}.c' \,\|\, e\rangle & \\
& \vdots & & \\
\langle \lambda a.p \,\|\, q \cdot e\rangle & \to & \langle \mu\hat{\mathrm{tp}}.\langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, \hat{\mathrm{tp}}\rangle\rangle \,\|\, e\rangle & (q \in \mathrm{NEF}) \\
\langle \lambda a.p \,\|\, q \cdot e\rangle & \to & \langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, e\rangle\rangle & (q \notin \mathrm{NEF}) \\
\langle \mathrm{prf}\; p \,\|\, e\rangle & \to & \langle \mu\hat{\mathrm{tp}}.\langle p \,\|\, \tilde{\mu}a.\langle \mathrm{prf}\; a \,\|\, \hat{\mathrm{tp}}\rangle\rangle \,\|\, e\rangle &
\end{array}$$

$\text{dL}_{\hat{\text{tp}}}$

An extension of dL with:

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

Typing rules:

Regular mode

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

Dependent mode

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

$\mathrm{dL}_{\hat{\mathrm{tp}}}$

An extension of dL with:

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

Typing rules:

Regular mode

$$\frac{\Gamma \vdash p : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta}{\langle p \,\|\, e\rangle : \Gamma \vdash \Delta}$$

Dependent mode

$$\frac{\Gamma \vdash p : A \mid \Delta \qquad \Gamma \mid e : A \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma\{\cdot|p\}}{\langle p \,\|\, e\rangle : \Gamma \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma}$$

Use of σ limited to $\hat{\mathrm{tp}}$:

$$\frac{c : (\Gamma \vdash_d \Delta, \hat{\mathrm{tp}} : A; \{\cdot|\cdot\})}{\Gamma \vdash \mu\hat{\mathrm{tp}}.c : A \mid \Delta}\ \hat{\mathrm{tp}}_I \qquad \frac{B \in A_\sigma}{\Gamma \mid \hat{\mathrm{tp}} : A \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma\{\cdot|p\}}\ \hat{\mathrm{tp}}_E$$

dL$_{\hat{\mathrm{tp}}}$

An extension of dL with:

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

Typing rules:

Regular mode

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

Dependent mode

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

Use of σ limited to $\hat{\mathrm{tp}}$:

$$\frac{c : (\Gamma \vdash_d \Delta, \hat{\mathrm{tp}} : A; \{\cdot|\cdot\})}{\Gamma \vdash \mu\hat{\mathrm{tp}}.c : A \mid \Delta}\ \hat{\mathrm{tp}}_I \qquad \frac{B \in A_\sigma}{\Gamma \mid \hat{\mathrm{tp}} : A \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma\{\cdot|p\}}\ \hat{\mathrm{tp}}_E$$

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

Typed CPS translation

Target language:

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

Normalization:

If $[\![c]\!]$ 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:

$\nvdash p : \bot.$

Proof. $[\![\bot]\!] = (\bot \to \bot) \to \bot.$ □

Bilan

An extension of dL with:

- **delimited continuations**
- dependent types restricted to the NEF **fragment**
- *Regular mode*
  $$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle p \| e \rangle : \Gamma \vdash \Delta}$$
  Dependent mode
  $$\frac{\Gamma \vdash p : A \mid \Delta \quad \Gamma \mid e : A \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma\{\cdot|p\}}{\langle p \| e \rangle : \Gamma \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma}$$
- delimited scope of dependencies:

$$\frac{c : (\Gamma \vdash_d \Delta, \hat{\mathrm{tp}} : A; \{\cdot|\cdot\})}{\Gamma \vdash \mu\hat{\mathrm{tp}}.c : A \mid \Delta}\ \hat{\mathrm{tp}}_I \qquad \frac{B \in A_\sigma}{\Gamma \mid \hat{\mathrm{tp}} : A \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma\{\cdot|p\}}\ \hat{\mathrm{tp}}_E$$

- 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**

Regular mode | *Dependent mode*

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

- delimited scope of dependencies:

$$\frac{c : (\Gamma \vdash_d \Delta, \hat{\mathrm{tp}} : A; \{\cdot|\cdot\})}{\Gamma \vdash \mu\hat{\mathrm{tp}}.c : A \mid \Delta}\ \hat{\mathrm{tp}}_I \qquad \frac{B \in A_\sigma}{\Gamma \mid \hat{\mathrm{tp}} : A \vdash_d \Delta, \hat{\mathrm{tp}} : B; \sigma\{\cdot|p\}}\ \hat{\mathrm{tp}}_E$$

- 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:

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

- 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:

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

- 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:

$$\textsc{nef} \subseteq \text{semantical values}$$

We define an embedding of proofs and types that:

- is **correct** with respect to typing

$$\Gamma \vdash p : A \mid \Delta \qquad \Rightarrow \qquad (\Gamma \cup \Delta)^* \vdash [\![p]\!]_p : A^*$$

- is **adequate** with his realizability model

$$\Gamma \vdash p : A \mid \Delta \quad \wedge \quad \sigma \Vdash (\Gamma \cup \Delta)^* \qquad \Rightarrow \qquad [\![p]\!]_p \sigma \in |A|$$

- allows to transfer Rodolphe's safety results

$$\nvdash p : \bot$$

dLPA$^\omega$: a sequent calculus with dependent types for classical arithmetic

dLPA$^\omega$

A **classical sequent calculus** with:

- stratified **dependent types** :
  - terms: $t,u ::= \dots \mid \mathtt{wit}\, p$
  - 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:

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

- **stores**:

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

- inductive and **coinductive** constructions:

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

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

End of the road

dPA$^\omega$
+ control operators
+ dependent types
+ co-fixpoints
+ sharing & laziness

Subject reduction

?

dLPA$^\omega$?
+ sequent calculus
+ dependent types
+ co-fixpoints
+ sharing & laziness

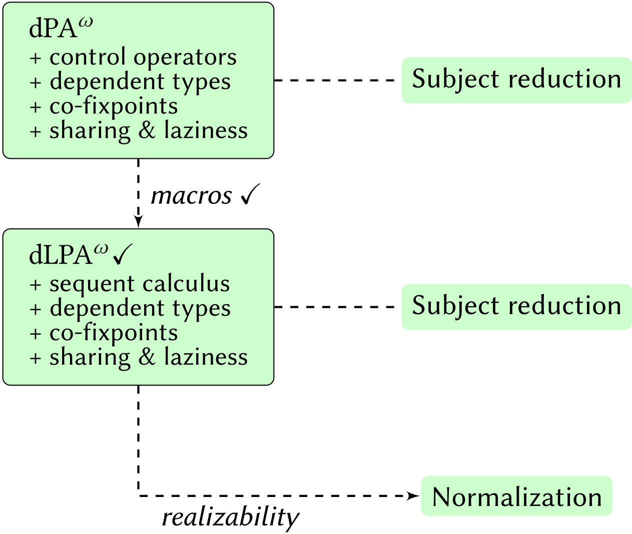
Subject reduction

?

?-calculus

Normalization

End of the road



Realizability interpretation

Same methodology:

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

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

- dependent types from Rodolphe's calculus:

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

- co-inductive formulas

Realizability interpretation

Same methodology:

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

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

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

$$\|\nu^t_{Xx}A\|_f \triangleq \bigcup_{n\in\mathbb{N}} \|F^n_{A,t}\|_f$$

Realizability interpretation

Same methodology:

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

Resembles $\overline{\lambda}_{[lv\tau\star]}$-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

$$\nvdash_{\text{dLPA}^\omega} p : \bot$$

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$^\omega$:
  - soundness and normalization,
  - realizability interpretation of co-fixpoints

Further work

1. **Classical call-by-need**:
   - typing the CPS with Kripke forcing
2. **$\mathrm{dL}_{\hat{tp}}$**:
   - 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. **$\mathrm{dL}_{\hat{\mathrm{tp}}}$**:
    - 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$_{\hat{\mathrm{tp}}}$**:
   - Connection with:
     - Pédrot-Tabareau's Baclofen Type Theory?
     - Vákár's categorical presentation?
     - Bowman *et. al.* CPS for CC?
   - **Dependent types & effects**:

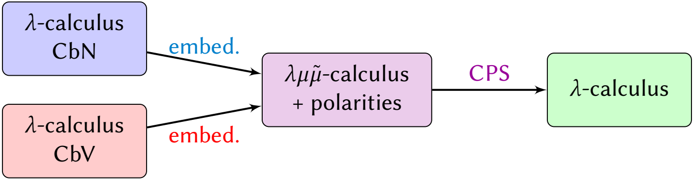
$$
\begin{tikzcd}
\begin{array}{c}\lambda\text{-calculus}\\ \text{CbN}\end{array} \arrow[r, "\text{embed.}"] & \begin{array}{c}\lambda\mu\tilde{\mu}\text{-calculus}\\ \text{CbN}\end{array} \arrow[r, "\text{CPS}"] & \lambda\text{-calculus} \\
\begin{array}{c}\lambda\text{-calculus}\\ \text{CbV}\end{array} \arrow[r, "\text{embed.}"] & \begin{array}{c}\lambda\mu\tilde{\mu}\text{-calculus}\\ \text{CbV}\end{array} \arrow[r, "\text{CPS}"] & \lambda\text{-calculus}
\end{tikzcd}
$$

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?

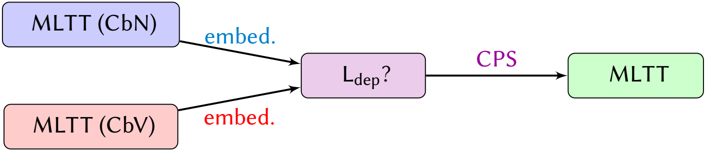
   - **Dependent types & effects**:



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

Further work

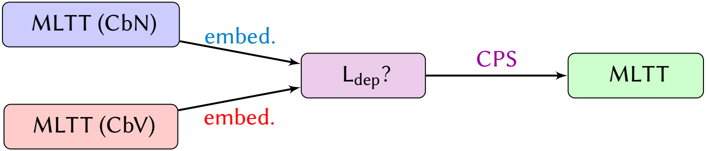
1. **Classical call-by-need**:
   - typing the CPS with Kripke forcing
2. **dL$_{\hat{tp}}$**:
   - Connection with:
     - Pédrot-Tabareau's Baclofen Type Theory?
     - Vákár's categorical presentation?
     - Bowman *et. al.* CPS for CC?
   - **Dependent types & effects**:



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?

Thank you for you attention.