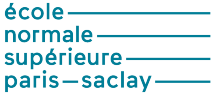
Dependent Type Theory in Polarised Sequent Calculus

Étienne Miquey Xavier Montillet Guillaume Munch-Maccagnoni

Deducteam Seminar
27/02/2020





Dependent type theories

Lot of features:

- for *programmers*
- for *logicians*
- for *proof-assisted* people
- for *proof-assisting* people

Ingredients:

- dependent product:

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : \Pi(x : A).B} \qquad \frac{\Gamma \vdash t : \Pi(x : A).B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B[u/x]}$$

- dependent sum:

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B[t/x]}{\Gamma \vdash (t, u) : \Sigma(x : A).B} \qquad \frac{\Gamma \vdash t : \Sigma(x : A).B}{\Gamma \vdash \mathtt{wit}\ t : A} \qquad \frac{\Gamma \vdash t : \Sigma(x : A).B}{\Gamma \vdash \mathtt{prf}\ t : B(\mathtt{wit}\ t)}$$

CPS & classical logic

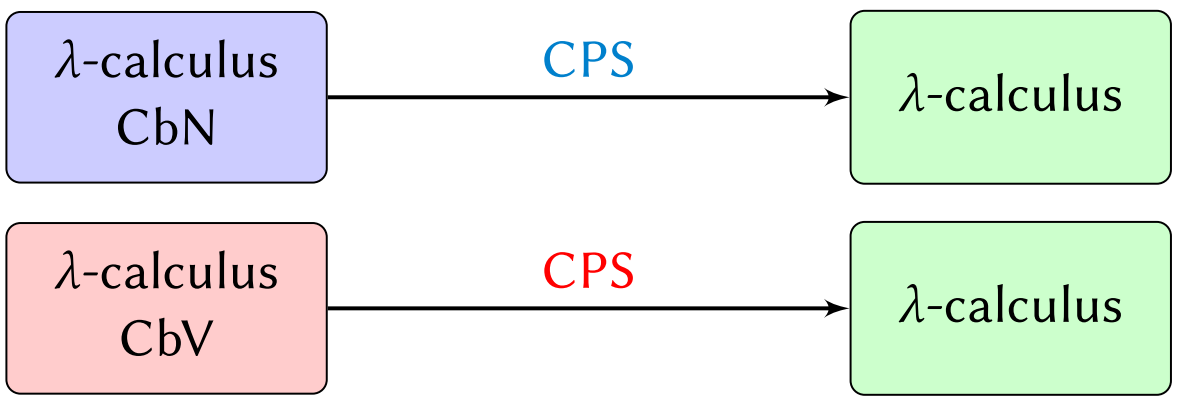
CPS:

1. provides semantics for **control operators:**

$$[\![\mathsf{catch}_\alpha\ t]\!] \triangleq \lambda\alpha.[\![t]\!] \qquad [\![\mathsf{throw}_\alpha\ t]\!] \triangleq \lambda\_.[\![t]\!]\alpha$$

2. explicit the flow of control, hence the **evaluation strategy**

	Call-by-name	Call-by-value
$[\![tu]\!]$	$\lambda k.[\![t]\!]\,[\![u]\!]\,k$	$\lambda k.[\![u]\!]\,(\lambda v.[\![t]\!]\,v\,k)$
$[\![A \to B]\!]$	$(\neg\neg A) \to \neg\neg B$	$A \to \neg\neg B$



Sequent calculus

Type system: sequents

$$\frac{\Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta}{\langle t \parallel e \rangle : (\Gamma \vdash \Delta)} \qquad \frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta} \qquad \frac{\Gamma \vdash u : A \mid \Delta \qquad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid u \cdot e : A \to B \vdash \Delta}$$

Calculi of abstract machines

$$\langle t\ u \parallel e \rangle \quad \rhd_{\text{abs}} \quad \langle t \parallel u \cdot e \rangle$$
$$\langle \lambda x.\, t \parallel u \cdot e \rangle \quad \rhd_{\text{abs}} \quad \langle t\,[u/x] \parallel e \rangle$$

CPS factorize through sequent calculi:

Sequent calculus

Type system: sequents

$$\frac{\Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta}{\langle t \parallel e \rangle : (\Gamma \vdash \Delta)} \qquad \frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta} \qquad \frac{\Gamma \vdash u : A \mid \Delta \qquad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid u \cdot e : A \to B \vdash \Delta}$$

Calculi of abstract machines

$$\begin{aligned} \langle t\ u \parallel e \rangle &\rhd_{\text{abs}} \langle t \parallel u \cdot e \rangle \\ \langle \lambda x.\, t \parallel u \cdot e \rangle &\rhd_{\text{abs}} \langle t\,[u/x] \parallel e \rangle \end{aligned}$$

CPS factorize through sequent calculi:

Sequent calculus

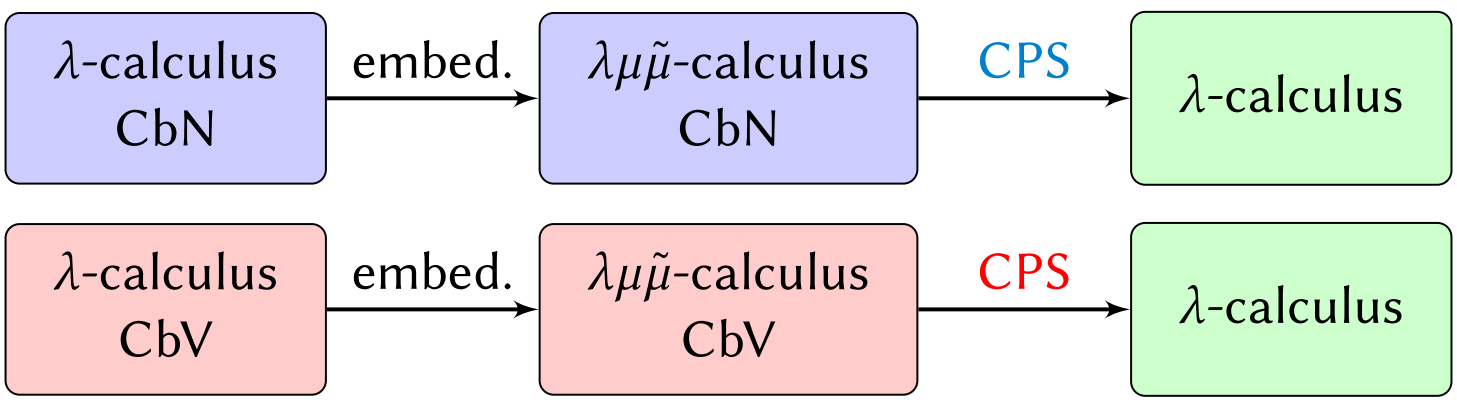
Type system: sequents

$$\frac{\Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta}{\langle t \parallel e \rangle : (\Gamma \vdash \Delta)} \qquad \frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta} \qquad \frac{\Gamma \vdash u : A \mid \Delta \qquad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid u \cdot e : A \to B \vdash \Delta}$$

Calculi of abstract machines

$$\begin{aligned} \langle t\ u \parallel e \rangle &\quad \rhd_{\text{abs}} \quad \langle t \parallel u \cdot e \rangle \\ \langle \lambda x.\, t \parallel u \cdot e \rangle &\quad \rhd_{\text{abs}} \quad \langle t\,[u/x] \parallel e \rangle \end{aligned}$$

CPS factorize through sequent calculi:



Polarised sequent calculus

Call-by-value / call-by-name: **global** restrictions on the evaluation

$\rightsquigarrow$ *ensure confluence in a classical settings*

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative polarity	Positive polarity
Every terms is a value	Every context is a covalue
Call-by-name	Call-by-value

Factorize both CPS:

Polarised sequent calculus

Call-by-value / call-by-name: **global** restrictions on the evaluation

$\rightsquigarrow$ *ensure confluence in a classical settings*

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative polarity	Positive polarity
Every terms is a value	Every context is a covalue
Call-by-name	Call-by-value

Factorize both CPS:

Polarised sequent calculus

Call-by-value / call-by-name: **global** restrictions on the evaluation

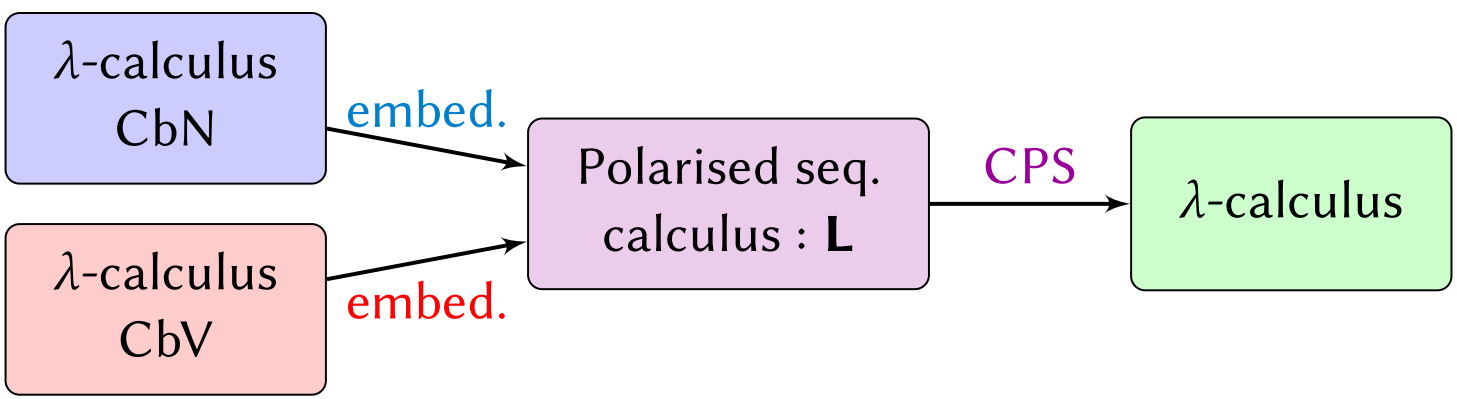
$\rightsquigarrow$ *ensure confluence in a classical settings*

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative polarity	Positive polarity
Every terms is a value	Every context is a covalue
Call-by-name	Call-by-value

Factorize both CPS:



Dependent Types & Classical Logic

On the Degeneracy of Σ-Types in Presence of Computational Classical Logic

H. Herbelin, TLCA 2005

Abstract. We show that a minimal dependent type theory based on Σ-types and equality is degenerated in presence of computational classical logic. By computational classical logic is meant a classical logic derived from a control operator equipped with reduction rules similar to the ones of Felleisen's $\mathcal{C}$ or Parigot's μ operators. As a consequence, formalisms such as Martin-Löf's type theory or the (`Set`-predicative variant of the) Calculus of Inductive Constructions are inconsistent in presence of computational classical logic. Besides, an analysis of the role of the η-rule for control operators through a set-theoretic model of computational classical logic is given.

Dependent Types & Classical Logic

Computational classical logic:

- $\mathtt{catch}_\alpha$ captures the *current continuation*
- $\mathtt{throw}_\alpha$ replaces the *current continuation* by α

Paradox:
One can define:

$$H_0 := \mathtt{catch}_\alpha(1, \mathtt{throw}_\alpha(0, \mathtt{refl})) : \Sigma(x : \mathbb{N}).x = 0$$

and reach a contradiction:

$$(\mathtt{wit}\, H_0, \mathtt{prf}\, H_0) \longrightarrow \underbrace{(1, \overbrace{\mathtt{refl}}^{0=0})}_{\cancel{\Sigma(x:\mathbb{N}).x=0}}$$

Morality:
↬ need to **restrict** dependencies to "*not too effectful*" computations

Dependent types & CPS

CPS Translating Inductive and Coinductive Types

G. Barthe & T. Uustalu, PEPM 2002

ABSTRACT

We investigate CPS translatability of typed λ-calculi with inductive and coinductive types. We show that tenable Plotkin-style call-by-name CPS translations exist for simply typed λ-calculi with a natural number type and stream types and, more generally, with arbitrary positive inductive and coinductive types. These translations also work in the presence of control operators and generalize for dependently typed calculi where case-like eliminations are only allowed in non-dependent forms. No translation is possible along the same lines for small Σ-types and sum types with dependent case.

Questions

- Is it *really* impossible?
- Is the problem limited to *call-by-name*?
- Is the problem limited to Σ*-types*?
- What about *value restriction*?
- Can't we get a *dependent and classical calculus?*

Questions

- Is it *really* impossible?
- Is the problem limited to *call-by-name*?
- Is the problem limited to Σ-*types*?
- What about *value restriction*?
- Can't we get a *dependent and classical calculus*?

No, it's not:

- Bowman *et al.* [POPL 2018]:

> *parametric* answer-types + *extensional type theory*

- M. [ESOP 2017]:

> *parametric and dependent* answer-types + *delimited continuations*

- Cong, Asai [ICFP 2018]

Questions

- Is it *really* impossible?
- Is the problem limited to *call-by-name*?
- Is the problem limited to *Σ-types*?
- What about *value restriction*?
- Can't we get a *dependent and classical calculus?*

Yes, we can:

- Lepigre [ESOP 2016]:

control operator + *full dependent types* (semantical restriction)

- M. [ESOP 2017]:

sequent calculus + (some) *dependent types* (syntactic restriction)

- This talk:

L_{dep}, a *dependent type theory* in *polarised sequent calculus*

Sequent Calculus

A matter of principles

Curien-Herbelin's duality of computation

Griffin (1990): classical logic ≅ control operator

Starting observation:

Computational duality:

Sequent calculus ≅ *abstract machine-like calculus*

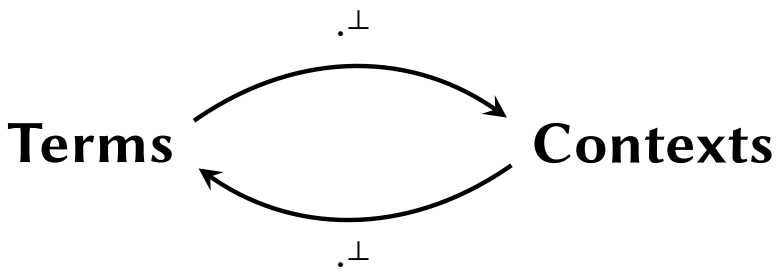
Curien-Herbelin's duality of computation

Griffin (1990): classical logic ≅ control operator

Starting observation:

calculus and $\lambda\mu$-calculus. Our starting point was the observation that the call-by-value discipline manipulates input much in the same way as (the classical extension of) λ-calculus manipulates output. Computing MN in call-by-

Computational duality:



Sequent calculus ≅ abstract machine-like calculus

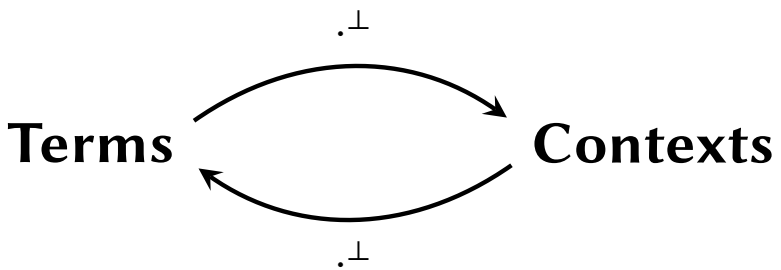
Curien-Herbelin's duality of computation

Griffin (1990): classical logic ≅ control operator

Starting observation:

calculus and $\lambda\mu$-calculus. Our starting point was the observation that the call-by-value discipline manipulates input much in the same way as (the classical extension of) λ-calculus manipulates output. Computing MN in call-by-

Computational duality:



Sequent calculus ≅ *abstract machine-like calculus*

Abstract machine

Reduction

$$\langle t\ u \parallel e\rangle \quad \triangleright_{\text{abs}} \quad \langle t \parallel u \cdot e\rangle$$
$$\langle \lambda x.\, t \parallel u \cdot e\rangle \quad \triangleright_{\text{abs}} \quad \langle t\,[u/x] \parallel e\rangle$$

Syntax

$$c ::= \langle t \parallel e\rangle \qquad \text{commands}$$

terms	$t, u ::=$	$e, f ::=$	contexts
variable	$\mid x, y, z$	$\mid \star$	empty
application	$\mid t\ u$	$\mid t \cdot e$	application stack
λ-abstraction	$\mid \lambda x.\, t$		

Introducing μ

$$\langle t\ u \parallel e \rangle \rhd_{\text{abs}} \langle t \parallel u \cdot e \rangle$$

This reduction **defines** $(t\ u)$:

It is the term that, when put against $|e\rangle$, reduces to $\langle t \parallel u \cdot e \rangle$.

Idea: introduce a more primitive syntax

$$\langle \mu\alpha.\, c \parallel e \rangle \rhd_{\mu} c\,[e/\alpha]$$

$$t\ u \triangleq \mu\alpha.\, \langle t \parallel u \cdot \alpha \rangle$$

(actually the intuitionistic version $\mu\star.c$ is enough)

Introducing μ

$$\langle t\ u \parallel e \rangle \rhd_{\text{abs}} \langle t \parallel u \cdot e \rangle$$

This reduction **defines** $(t\ u)$:

It is the term that, when put against $|e\rangle$, reduces to $\langle t \parallel u \cdot e \rangle$.

Idea: introduce a more primitive syntax

$$\langle \mu\alpha.\, c \parallel e \rangle \rhd_{\mu} c\,[e/\alpha]$$

$$t\ u \quad \triangleq \quad \mu\alpha.\, \langle t \parallel u \cdot \alpha \rangle$$

(actually the intuitionistic version $\mu\star.c$ is enough)

Introducing $\tilde{\mu}$

A regular syntax?

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{rl} t, u ::= & \\ & \mid x, y \\ & \mid \lambda x.t \\ & \mid \mu\alpha.\, c \end{array} \qquad \begin{array}{rl} e, f ::= & \\ & \mid \alpha, \beta \\ & \mid t \cdot e \\ & \mid \; ? \end{array}$$

Reminder:

$\langle (\lambda x.t)u \parallel e \rangle \rhd_{abs}$

calculus and $\lambda\mu$-calculus. Our starting point was the observation that the call-by-value discipline manipulates input much in the same way as (the classical extension of) λ-calculus manipulates output. Computing MN in call-by-

Introducing $\tilde{\mu}$

A regular syntax?

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{rl} t, u ::= & \\ & \mid x, y \\ & \mid \lambda x.t \\ & \mid \mu\alpha.\, c \end{array} \qquad \begin{array}{rl} e, f ::= & \\ & \mid \alpha, \beta \\ & \mid t \cdot e \\ & \mid \; ? \end{array}$$

Same idea, in the **dual situation**:

$$\langle (\lambda x.t)u \parallel e \rangle \rhd_{\text{abs}} \langle \text{let } x = t \text{ in } u \parallel e \rangle \quad \rhd_{\text{abs}} \quad \left\langle t \,\middle\|\, \text{“let } x = \square \text{ in } \langle u \parallel e \rangle'' \right\rangle$$

$$\langle t \parallel \tilde{\mu}x.c \rangle \quad \rhd_{\tilde{\mu}} \quad c\,[t/x]$$

Introducing $\tilde{\mu}$

A regular syntax

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{rl} t, u ::= & \\ & \mid x, y \\ & \mid \lambda x.t \\ & \mid \mu\alpha.\, c \end{array} \qquad \begin{array}{rl} e, f ::= & \\ & \mid \alpha, \beta \\ & \mid t \cdot e \\ & \mid \tilde{\mu}x.\, c \end{array}$$

Same idea, in the **dual situation**:

$$\langle (\lambda x.t)u \parallel e \rangle \rhd_{\text{abs}} \langle \text{let } x = t \text{ in } u \parallel e \rangle \quad \rhd_{\text{abs}} \quad \Big\langle t \parallel \underbrace{\text{“let } x = \square \text{ in } \langle u \parallel e \rangle''}_{\tilde{\mu}x.\langle u \parallel e \rangle} \Big\rangle$$

$$\langle t \parallel \tilde{\mu}x.c \rangle \quad \rhd_{\tilde{\mu}} \quad c\,[t/x]$$

Curien-Herbelin's $\lambda\mu\tilde{\mu}$-calculus

Syntax:

$$
\begin{array}{lll}
t, u ::= & c ::= \langle t \parallel e \rangle & e, f ::= \\
\quad \mid x, y & & \quad \mid \alpha, \beta \\
\quad \mid \lambda x.t & & \quad \mid t \cdot e \\
\quad \mid \mu\alpha.\, c & & \quad \mid \tilde{\mu} x.\, c
\end{array}
$$

Reduction:

$$
\begin{array}{rcl}
\langle \lambda x.t \parallel u \cdot e \rangle & \longrightarrow & \left\langle u \,\middle\|\, \tilde{\mu} x.\langle t \parallel e \rangle \right\rangle \\
\langle t \parallel \tilde{\mu} x.c \rangle & \longrightarrow & c[t/x] \\
\langle \mu\alpha.c \parallel e \rangle & \longrightarrow & c[e/\alpha]
\end{array}
$$

Curien-Herbelin's $\lambda\mu\tilde{\mu}$-calculus

Syntax:

$$
\begin{array}{lll}
t, u ::= & c ::= \langle t \parallel e \rangle & e, f ::= \\
\quad \mid x, y & & \quad \mid \alpha, \beta \\
\quad \mid \lambda x.t & & \quad \mid t \cdot e \\
\quad \mid \mu\alpha.\, c & & \quad \mid \tilde{\mu}x.\, c
\end{array}
$$

Reduction:

$$
\begin{array}{rcl}
\langle \lambda x.t \parallel u \cdot e \rangle & \longrightarrow & \left\langle u \,\middle\|\, \tilde{\mu}x.\langle t \parallel e \rangle \right\rangle \\
\langle t \parallel \tilde{\mu}x.c \rangle & \longrightarrow & c[t/x] \\
\langle \mu\alpha.c \parallel e \rangle & \longrightarrow & c[e/\alpha]
\end{array}
$$

Critical pair:

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

Curien-Herbelin's $\lambda\mu\tilde{\mu}$-calculus

Syntax:

$$t, u ::= \quad | \; x, y \quad | \; \lambda x.t \quad | \; \mu\alpha.\, c \qquad \text{(Values)}$$

$$c ::= \langle t \parallel e \rangle$$

$$e, f ::= \quad | \; \alpha, \beta \quad | \; t \cdot e \quad | \; \tilde{\mu}x.\, c \qquad \text{(Co-values)}$$

Reduction:

$$\begin{aligned} \langle \lambda x.t \parallel u \cdot e \rangle &\to \langle u \parallel \tilde{\mu}x.\langle t \parallel e \rangle \rangle \\ \langle t \parallel \tilde{\mu}x.c \rangle &\to c[t/x] \qquad t \in \mathcal{V} \\ \langle \mu\alpha.c \parallel e \rangle &\to c[e/\alpha] \qquad e \in \mathcal{E} \end{aligned}$$

Critical pair:

$$\langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle$$

CbV ↙ $c[\tilde{\mu}x.c'/\alpha]$

CbN ↘ $c'[\mu\alpha.c/x]$

Detroducing λ

$$\langle \lambda x.\, t \parallel u \cdot e \rangle \rhd_{\text{abs}} \langle t\,[u/x] \parallel e \rangle$$

Idea: λ pattern-matches on the context, deconstructing $u \cdot e$.

A more principled term:

$$\mu(x \cdot \alpha).\, c$$

together with:

$$\langle \mu(x \cdot \alpha).\, c \parallel u \cdot e \rangle \rhd_{\text{fun}} c\,[u/x, e/\alpha]$$

$$\lambda x.\, t \triangleq \mu(x \cdot \alpha).\, \langle t \parallel \alpha \rangle$$

Detroducing λ

$$\langle \lambda x.\, t \parallel u \cdot e \rangle \rhd_{\text{abs}} \langle t\,[u/x] \parallel e \rangle$$

Idea: λ pattern-matches on the context, deconstructing $u \cdot e$.

A more principled term:

$$\mu(x \cdot \alpha).\, c$$

together with:

$$\langle \mu(x \cdot \alpha).\, c \parallel u \cdot e \rangle \rhd_{\text{fun}} c\,[u/x, e/\alpha]$$

$$\lambda x.\, t \triangleq \mu(x \cdot \alpha).\, \langle t \parallel \alpha \rangle$$

Other datatypes

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{rl} t, u ::= & \\ & \mid x, y \\ & \mid \mu(x \cdot \alpha).\, c \\ & \mid \mu\alpha.\, c \\ & \mid (t, u) \end{array} \qquad \begin{array}{rl} e, f ::= & \\ & \mid \alpha, \beta \\ & \mid t \cdot e \\ & \mid \tilde{\mu}x.\, c \\ & \mid \; ? \end{array}$$

Same idea:

$$\langle \text{let } (x_1, x_2) = t \text{ in } u \parallel e \rangle \rhd_{\text{abs}} \left\langle t \,\middle\|\, \text{let } (x_1, x_2) = \Box \text{ in } \langle u \parallel e \rangle \right\rangle$$

$$\text{let } (x_1, x_2) = t \text{ in } u \quad \triangleq \quad \mu\alpha.\, \langle t \parallel \tilde{\mu}(x_1, x_2).\, \langle u \parallel \alpha \rangle \rangle$$

$$\langle \mu\alpha.\, c \parallel e \rangle \quad \rhd_{\mu} \quad c\,[e/\alpha]$$

$$\langle t \parallel \tilde{\mu}x.\, c \rangle \quad \rhd_{\tilde{\mu}} \quad c\,[t/x]$$

$$\langle (t_1, t_2) \parallel \tilde{\mu}(x_1, x_2).\, c \rangle \quad \rhd \quad c\,[t_1/x_1, t_2/x_2]$$

Other datatypes

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{ll} t, u ::= & e, f ::= \\ \quad \mid x, y & \quad \mid \alpha, \beta \\ \quad \mid \mu(x \cdot \alpha).\, c & \quad \mid t \cdot e \\ \quad \mid \mu\alpha.\, c & \quad \mid \tilde{\mu}x.\, c \\ \quad \mid (t, u) & \quad \mid \tilde{\mu}(x_1, x_2).\, c \end{array}$$

Same idea:

$$\langle \text{let } (x_1, x_2) = t \text{ in } u \parallel e \rangle \rhd_{\text{abs}} \left\langle t \,\middle\|\, \text{let } (x_1, x_2) = \Box \text{ in } \langle u \parallel e \rangle \right\rangle$$

$$\text{let } (x_1, x_2) = t \text{ in } u \quad \triangleq \quad \mu\alpha.\, \left\langle t \,\middle\|\, \tilde{\mu}(x_1, x_2).\, \langle u \parallel \alpha \rangle \right\rangle$$

Other datatypes

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{ll} t, u ::= & e, f ::= \\ \quad \mid x, y & \quad \mid \alpha, \beta \\ \quad \mid \mu(x \cdot \alpha).\, c & \quad \mid t \cdot e \\ \quad \mid \mu\alpha.\, c & \quad \mid \tilde{\mu}x.\, c \\ \quad \mid (t, u) & \quad \mid \tilde{\mu}(x_1, x_2).\, c \end{array}$$

$$\begin{array}{rcl} \langle \mu\alpha.\, c \parallel e \rangle & \rhd_{\mu} & c\,[e/\alpha] \\ \langle t \parallel \tilde{\mu}x.\, c \rangle & \rhd_{\tilde{\mu}} & c\,[t/x] \\ \langle (t_1, t_2) \parallel \tilde{\mu}(x_1, x_2).\, c \rangle & \rhd_{\otimes} & c\,[t_1/x_1, t_2/x_2] \\ \langle \mu(x \cdot \alpha).\, c \parallel t \cdot e \rangle & \rhd_{\to} & c\,[t/x, e/\alpha] \end{array}$$

Other datatypes

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{ll} t, u ::= & e, f ::= \\ \quad \mid x, y & \quad \mid \alpha, \beta \\ \quad \mid \mu(x \cdot \alpha).\, c & \quad \mid t \cdot e \\ \quad \mid \mu\alpha.\, c & \quad \mid \tilde{\mu}x.\, c \\ \quad \mid (t, u) & \quad \mid \tilde{\mu}(x_1, x_2).\, c \\ \quad \mid \; ? & \quad \mid \pi_i \; e \end{array}$$

$$\begin{array}{rcl} \langle \mu\alpha.\, c \parallel e \rangle & \rhd_{\mu} & c\,[e/\alpha] \\ \langle t \parallel \tilde{\mu}x.\, c \rangle & \rhd_{\tilde{\mu}} & c\,[t/x] \\ \langle (t_1, t_2) \parallel \tilde{\mu}(x_1, x_2).\, c \rangle & \rhd_{\otimes} & c\,[t_1/x_1, t_2/x_2] \\ \langle \mu(x \cdot \alpha).\, c \parallel t \cdot e \rangle & \rhd_{\to} & c\,[t/x, e/\alpha] \end{array}$$

Other datatypes

$$c ::= \langle t \parallel e \rangle$$

$$
\begin{array}{ll}
t, u ::= & \\
\quad \mid & x, y \\
\quad \mid & \mu(x \cdot \alpha).\, c \\
\quad \mid & \mu\alpha.\, c \\
\quad \mid & (t, u) \\
\quad \mid & \mu[(\pi_1\ \alpha_1).\, c_1 \mid (\pi_2\ \alpha_2).\, c_2]
\end{array}
\qquad
\begin{array}{ll}
e, f ::= & \\
\quad \mid & \alpha, \beta \\
\quad \mid & t \cdot e \\
\quad \mid & \tilde{\mu}x.\, c \\
\quad \mid & \tilde{\mu}(x_1, x_2).\, c \\
\quad \mid & \pi_i\ e
\end{array}
$$

$$
\begin{array}{rcl}
\langle \mu\alpha.\, c \parallel e \rangle & \rhd_{\mu} & c\,[e/\alpha] \\
\langle t \parallel \tilde{\mu}x.\, c \rangle & \rhd_{\tilde{\mu}} & c\,[t/x] \\
\langle (t_1, t_2) \parallel \tilde{\mu}(x_1, x_2).\, c \rangle & \rhd_{\otimes} & c\,[t_1/x_1, t_2/x_2] \\
\langle \mu(x \cdot \alpha).\, c \parallel t \cdot e \rangle & \rhd_{\to} & c\,[t/x, e/\alpha] \\
\langle \mu[(\pi_1\ \alpha_1).\, c_1 \mid (\pi_2\ \alpha_2).\, c_2] \parallel \pi_i\ e \rangle & \rhd_{\times} & c_i\,[e/\alpha_i]
\end{array}
$$

Type system...

$$c ::= \langle t \parallel e \rangle$$

$$\begin{array}{rcllll} t, u & ::= & x, y, z & \mid \mu\alpha.\, c & \mid \mu(x \cdot \alpha).\, c & \mid (t, u) \\ e, f & ::= & \star, \alpha, \beta & \mid \tilde{\mu}x.\, c & \mid t \cdot e & \mid \tilde{\mu}(x_1, x_2).\, c \end{array}$$

Sequents: $c : (\Gamma \vdash \Delta)$ $\quad \Gamma \vdash t : A \mid \Delta$ $\quad \Gamma \mid e : A \vdash \Delta$

Type system...

$$c ::= \langle t \parallel e \rangle \qquad \begin{array}{rcllll} t, u & ::= & x, y, z & \mid \mu\alpha.\, c & \mid \mu(x \cdot \alpha).\, c & \mid (t, u) \\ e, f & ::= & \star, \alpha, \beta & \mid \tilde{\mu}x.\, c & \mid t \cdot e & \mid \tilde{\mu}(x_1, x_2).\, c \end{array}$$

Sequents: $\boxed{c : (\Gamma \vdash \Delta)}$ $\boxed{\Gamma \vdash t : A \mid \Delta}$ $\boxed{\Gamma \mid e : A \vdash \Delta}$

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

$$\frac{c : (\Gamma, x_1 : A_1, x_2 : A_2 \vdash \Delta)}{\Gamma \mid \tilde{\mu}(x_1, x_2).\, c : A_1 \otimes A_2 \vdash \Delta} \qquad \frac{\Gamma \vdash t_1 : A_1 \mid \Delta \qquad \Gamma \vdash t_2 : A_2 \mid \Delta}{\Gamma \vdash (t_1, t_2) : A_1 \otimes A_2 \mid \Delta}$$

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

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

Type system...

$$c ::= \langle t \parallel e \rangle \qquad \begin{array}{rcllll} t, u & ::= & x, y, z & \mid \mu\alpha.\, c & \mid \mu(x \cdot \alpha).\, c & \mid (t, u) \\ e, f & ::= & \star, \alpha, \beta & \mid \tilde{\mu}x.\, c & \mid t \cdot e & \mid \tilde{\mu}(x_1, x_2).\, c \end{array}$$

Sequents: $c : (\Gamma \vdash \Delta)$ $\quad \Gamma \vdash t : A \mid \Delta$ $\quad \Gamma \mid e : A \vdash \Delta$

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

$$\frac{c : (\Gamma, x_1 : A_1, x_2 : A_2 \vdash \Delta)}{\Gamma \mid \tilde{\mu}(x_1, x_2).\, c : A_1 \otimes A_2 \vdash \Delta} \qquad \frac{\Gamma \vdash t_1 : A_1 \mid \Delta \qquad \Gamma \vdash t_2 : A_2 \mid \Delta}{\Gamma \vdash (t_1, t_2) : A_1 \otimes A_2 \mid \Delta}$$

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

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

... and logic

$$\boxed{\Gamma \vdash t : A \mid \Delta} \qquad \boxed{\Gamma \mid e : A \vdash \Delta} \qquad \boxed{c : (\Gamma \vdash \Delta)}$$

Multiple co-variables: **classical** logic. *(nicer than $\Lambda\mu$ or primitive call/cc)*

$$\text{call/cc}(f) \triangleq \mu\alpha.\left\langle f \parallel (\mu(x \cdot \beta).\langle x \parallel \alpha\rangle) \cdot \alpha \right\rangle$$

Intuitionistic restrictions, either:

- linear co-context

$$\frac{}{\Gamma, x : A \vdash x : A \mid \emptyset} \qquad \frac{\Gamma \vdash t : A \mid \Delta_1 \qquad \Gamma \mid e : B \vdash \Delta_2}{\Gamma \mid t \cdot e : A \to B \vdash \Delta_1, \Delta_2}$$

- single variable + shadowing

$$t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle \qquad \lambda x.\, t \triangleq \mu(x \cdot \alpha).\langle t \parallel \alpha\rangle$$

$$t\ u \triangleq \mu\star.\langle t \parallel u \cdot \star\rangle \qquad \lambda x.\, t \triangleq \mu(x \cdot \star).\langle t \parallel \star\rangle$$

... and logic

$$\boxed{\Gamma \vdash t : A \mid \Delta} \qquad \boxed{\Gamma \mid e : A \vdash \Delta} \qquad \boxed{c : (\Gamma \vdash \Delta)}$$

Multiple co-variables: **classical** logic. *(nicer than $\Lambda\mu$ or primitive call/cc)*

$$\text{call/cc}(f) \triangleq \mu\alpha.\left\langle f \parallel (\mu(x \cdot \beta).\langle x \parallel \alpha\rangle) \cdot \alpha \right\rangle$$

Intuitionistic restrictions, either:

- linear co-context

$$\frac{}{\Gamma, x : A \vdash x : A \mid \emptyset} \qquad \frac{\Gamma \vdash t : A \mid \Delta_1 \qquad \Gamma \mid e : B \vdash \Delta_2}{\Gamma \mid t \cdot e : A \to B \vdash \Delta_1, \Delta_2}$$

- single variable + shadowing

$$t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle \qquad \lambda x.\, t \triangleq \mu(x \cdot \alpha).\langle t \parallel \alpha\rangle$$

$$t\ u \triangleq \mu\star.\langle t \parallel u \cdot \star\rangle \qquad \lambda x.\, t \triangleq \mu(x \cdot \star).\langle t \parallel \star\rangle$$

Polarised Sequent Caluclus

A computational wonderland

A bit of polarity

Call-by-value / call-by-name: **global** restrictions on the evaluation

$$c[\tilde{\mu}x.c'/\alpha] \xleftarrow{\text{CbV}} \langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \xrightarrow{\text{CbN}} c'[\mu\alpha.c/x]$$

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative terms (by-name) are inert values.
Positive co-terms (by-value) inert co-values.

$$\begin{array}{rrcl} \textbf{Commands} & c & ::= & \langle V \parallel e \rangle^- \mid \langle t \parallel S \rangle^+ \\ \textbf{Terms} & t, u & ::= & \mu^+\alpha.\, c \mid V \\ \textbf{Values} & V, W & ::= & \mu^-\alpha.\, c \mid x^+, x^- \mid \mu(x^\varepsilon \cdot \alpha^\varepsilon).c \mid (V, W) \\ \textbf{Contexts} & e, f & ::= & \tilde{\mu}^- x.\, c \mid S \\ \textbf{Stacks} & S & ::= & \tilde{\mu}^+ x.\, c \mid \star, \alpha^+, \alpha^- \mid V \cdot S \mid \tilde{\mu}(x_1^\varepsilon, x_2^\varepsilon).\, c \end{array}$$

A bit of polarity

Call-by-value / call-by-name: **global** restrictions on the evaluation

$$c[\tilde{\mu}x.c'/\alpha] \xleftarrow{\text{CbV}} \langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \xrightarrow{\text{CbN}} c'[\mu\alpha.c/x]$$

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative terms (by-name) are inert values.
Positive co-terms (by-value) inert co-values.

Commands	c	$::=$	$\langle V \parallel e \rangle^-$	$\mid \langle t \parallel S \rangle^+$
Terms	t, u	$::=$	$\mu^+\alpha.\, c$	$\mid V$
Values	V, W	$::=$	$\mu^-\alpha.\, c$	$\mid x^+, x^-$ $\mid \mu(x^\varepsilon \cdot \alpha^\varepsilon).c \mid (V, W)$
Contexts	e, f	$::=$	$\tilde{\mu}^- x.\, c$	$\mid S$
Stacks	S	$::=$	$\tilde{\mu}^+ x.\, c$	$\mid \star, \alpha^+, \alpha^-$ $\mid V \cdot S$ $\mid \tilde{\mu}(x_1^\varepsilon, x_2^\varepsilon).\, c$

A bit of polarity

Call-by-value / call-by-name: **global** restrictions on the evaluation

$$c[\tilde{\mu}x.c'/\alpha] \xleftarrow{\text{CbV}} \langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \xrightarrow{\text{CbN}} c'[\mu\alpha.c/x]$$

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative terms (by-name) are inert values.
Positive co-terms (by-value) inert co-values.

$$\textbf{Commands} \qquad c ::= \langle V \parallel e \rangle^- \mid \langle t \parallel S \rangle^+$$

$$\begin{array}{lrcl} \textbf{Terms} & t, u & ::= & \mu^+\alpha.\, c \mid V \\ \textbf{Values} & V, W & ::= & \mu^-\alpha.\, c \mid x^+, x^- \mid \mu(x^\varepsilon \cdot \alpha^\varepsilon).c \mid (V, W) \\ \textbf{Contexts} & e, f & ::= & \tilde{\mu}^- x.\, c \mid S \\ \textbf{Stacks} & S & ::= & \tilde{\mu}^+ x.\, c \mid \star, \alpha^+, \alpha^- \mid V \cdot S \mid \tilde{\mu}(x_1^\varepsilon, x_2^\varepsilon).\, c \end{array}$$

A bit of polarity

Call-by-value / call-by-name: **global** restrictions on the evaluation

$$c[\tilde{\mu}x.c'/\alpha] \xleftarrow{\text{CbV}} \langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \xrightarrow{\text{CbN}} c'[\mu\alpha.c/x]$$

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative terms (by-name) are inert values.
Positive co-terms (by-value) inert co-values.

Commands	c	$::=$	$\langle V \parallel e \rangle^- \mid \langle t \parallel S \rangle^+$	
Terms	t, u	$::=$	$\mu^+\alpha.\, c \mid V$	
Values	V, W	$::=$	$\mu^-\alpha.\, c \mid x^+, x^- \mid \mu(x^\varepsilon \cdot \alpha^\varepsilon).c \mid (V, W)$	
Contexts	e, f	$::=$	$\tilde{\mu}^- x.\, c \mid S$	
Stacks	S	$::=$	$\tilde{\mu}^+ x.\, c \mid \star, \alpha^+, \alpha^- \mid V \cdot S \mid \tilde{\mu}(x_1^\varepsilon, x_2^\varepsilon).\, c$	

A bit of polarity

Call-by-value / call-by-name: **global** restrictions on the evaluation

$$c[\tilde{\mu}x.c'/\alpha] \xleftarrow{\text{CbV}} \langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \xrightarrow{\text{CbN}} c'[\mu\alpha.c/x]$$

Polarities: a **type-directed** solution.

$$\begin{array}{rcl} P, Q & ::= & X^+, Y^+ \mid P \otimes Q \mid P \oplus Q \mid \cdots \mid \Downarrow N \\ M, N & ::= & X^-, Y^- \mid P \to N \mid M \times N \mid \cdots \mid \Uparrow P \end{array}$$

Negative terms (by-name) are inert values.
Positive co-terms (by-value) inert co-values.

Commands	c	$::=$	$\langle V \parallel e \rangle^-$	$\mid \langle t \parallel S \rangle^+$	
Terms	t, u	$::=$	$\mu^+\alpha.\, c$	$\mid V$	
Values	V, W	$::=$	$\mu^-\alpha.\, c$	$\mid x^+, x^-$	$\mid \mu(x^\varepsilon \cdot \alpha^\varepsilon).c \mid (V, W)$
Contexts	e, f	$::=$	$\tilde{\mu}^-x.\, c$	$\mid S$	
Stacks	S	$::=$	$\tilde{\mu}^+x.\, c$	$\mid \star, \alpha^+, \alpha^-$	$\mid V \cdot S \mid \tilde{\mu}(x_1^\varepsilon, x_2^\varepsilon).\, c$

Polarized reduction

$\lambda\mu\tilde{\mu}$

$$
\begin{array}{rcl}
\langle \mu\alpha.\, c \parallel e \rangle & \rhd_{\mu} & c\,[e/\alpha] \\
\langle t \parallel \tilde{\mu}x.\, c \rangle & \rhd_{\tilde{\mu}} & c\,[t/x] \\
\langle (t_1, t_2) \parallel \tilde{\mu}(x_1, x_2).\, c \rangle & \rhd_{\otimes} & c\,[t_1/x_1, t_2/x_2] \\
\langle \mu(x \cdot \alpha).\, c \parallel t \cdot e \rangle & \rhd_{\to} & c\,[t/x, e/\alpha]
\end{array}
$$

$$
\begin{array}{rcllll}
c & ::= & \langle V \parallel e \rangle^{-} & \mid \langle t \parallel S \rangle^{+} & & \\
t, u & ::= & \mu^{+}\alpha.\, c & \mid V & & \\
V, W & ::= & \mu^{-}\alpha.\, c & \mid x^{+}, x^{-} & \mid \mu(x \cdot \alpha).\, c & \mid (V, W) \\
e, f & ::= & \tilde{\mu}^{-}x.\, c & \mid S & & \\
S & ::= & \tilde{\mu}^{+}x.\, c & \mid \star, \alpha^{+}, \alpha^{-} & \mid V \cdot S & \mid \tilde{\mu}(x_1, x_2).\, c
\end{array}
$$

L

$$
\begin{array}{rcl}
\langle \mu^{\varepsilon}\alpha.\, c \parallel S \rangle & \rhd_{\mu} & c\,[S/\alpha^{\varepsilon}] \\
\langle V \parallel \tilde{\mu}^{\varepsilon}x.\, c \rangle & \rhd_{\tilde{\mu}} & c\,[V/x^{\varepsilon}] \\
\langle (V_1, V_2) \parallel \tilde{\mu}(x_1, x_2).\, c \rangle & \rhd_{\otimes} & c\,[V_1/x_1, V_2/x_2] \\
\langle \mu(x \cdot \alpha).\, c \parallel V \cdot S \rangle & \rhd_{\to} & c\,[V/x, S/\alpha]
\end{array}
$$

Strong values

$$
\begin{array}{rcllll}
c & ::= & \langle V \parallel e \rangle^{-} & \mid \langle t \parallel S \rangle^{+} & & \\
t, u & ::= & \mu^{+}\alpha.\, c & \mid V & & \\
V, W & ::= & \mu^{-}\alpha.\, c & \mid x, y, z & \mid \mu(x \cdot \alpha).\, c & \mid (V, W) \\
e, f & ::= & \tilde{\mu}^{-}x.\, c & \mid S & & \\
S & ::= & \tilde{\mu}^{+}x.\, c & \mid \star, \alpha, \beta & \mid V \cdot S & \mid \tilde{\mu}(x_1, x_2).\, c
\end{array}
$$

What about (t, u) or $t \cdot e$ when t (for example) is not a value?

Two design options:

- Force the user to **decide an evaluation order** by writing in ANF form.

$$\langle t \parallel \tilde{\mu}x.\, \langle (x, u) \parallel \dots \rangle \rangle \qquad \langle t \parallel \tilde{\mu}\langle \dots \parallel x \cdot e \rangle.\, \rangle$$

- Add an **automatic reduction** with arbitrary order: Wadler's ζ-rules.

$$\langle (t, u) \parallel S \rangle^{+} \rhd^{t \notin V}_{\zeta\_\text{pair}\_1} \langle t \parallel \tilde{\mu}x.\, \langle (x, u) \parallel S \rangle^{+} \rangle$$

Strong values

$$
\begin{array}{rcllll}
c & ::= & \langle V \parallel e \rangle^- & \mid \langle t \parallel S \rangle^+ & & \\
t, u & ::= & \mu^+\alpha.\, c & \mid V & & \\
V, W & ::= & \mu^-\alpha.\, c & \mid x, y, z & \mid \mu(x \cdot \alpha).\, c & \mid (V, W) \\
e, f & ::= & \tilde{\mu}^- x.\, c & \mid S & & \\
S & ::= & \tilde{\mu}^+ x.\, c & \mid \star, \alpha, \beta & \mid V \cdot S & \mid \tilde{\mu}(x_1, x_2).\, c
\end{array}
$$

What about (t, u) or $t \cdot e$ when t (for example) is not a value?

Two design options:

- Force the user to **decide an evaluation order** by writing in ANF form.

$$\left\langle t \parallel \tilde{\mu} x.\, \langle (x, u) \parallel \dots \rangle \right\rangle \qquad \left\langle t \parallel \tilde{\mu} \langle \dots \parallel x \cdot e \rangle.\, \right\rangle$$

- Add an **automatic reduction** with arbitrary order: Wadler's ζ-rules.

$$\langle (t, u) \parallel S \rangle^+ \rhd^{t \notin V}_{\zeta\_\text{pair\_1}} \left\langle t \parallel \tilde{\mu} x.\, \langle (x, u) \parallel S \rangle^+ \right\rangle$$

Polarity and CPS

How to define a CPS translation from such a calculus?

Easy: follow the syntax! ☺

$$[\![\langle V \parallel e \rangle^-]\!] \triangleq [\![e]\!]_2\,[\![V]\!]_1 \quad \text{(CbV)}$$
$$[\![\langle t \parallel S \rangle^+]\!] \triangleq [\![t]\!]_2\,[\![S]\!]_1 \quad \text{(CbN)}$$

$$[\![\mu^+\alpha.\,c]\!]_2 \triangleq \lambda\alpha.[\![c]\!]$$
$$[\![V]\!]_2 \triangleq \lambda k.k\,[\![V]\!]_0$$

$$[\![\mu^-\alpha.\,c]\!]_1 \triangleq \lambda\alpha.[\![c]\!]$$
$$[\![x^-]\!]_1 \triangleq x$$
$$[\![\mu(x\cdot\alpha).\,c]\!]_1 \triangleq \lambda(x,\alpha).[\![c]\!]$$
$$[\![V_+]\!]_1 \triangleq \lambda k.k\,[\![V_+]\!]_0$$

$$[\![x_+]\!]_0 \triangleq x$$
$$[\![(V,W)]\!]_0 \triangleq ([\![V]\!]_0, [\![W]\!]_0)$$

t_+/e_- t_-/e_+ V_+/S_-

Terms

$$[\![c]\!] : \bot$$

$$[\![A]\!]_2 \triangleq [\![A]\!]_1 \to \bot$$

$$[\![A]\!]_1 \triangleq [\![A]\!]_0 \to \bot$$

$$[\![A \otimes B]\!]_0 \triangleq [\![A]\!]_0 \times [\![A]\!]_0$$
$$\vdots$$

Types

Polarity and CPS

How to define a CPS translation from such a calculus?

Easy: follow the syntax! ☺

$$[\![\langle V \parallel e \rangle^-]\!] \triangleq [\![e]\!]_2\,[\![V]\!]_1 \quad \text{(CbV)}$$
$$[\![\langle t \parallel S \rangle^+]\!] \triangleq [\![t]\!]_2\,[\![S]\!]_1 \quad \text{(CbN)}$$

t_+/e_-

$$[\![\mu^+\alpha.\,c]\!]_2 \triangleq \lambda\alpha.[\![c]\!]$$
$$[\![V]\!]_2 \triangleq \lambda k.k\,[\![V]\!]_0$$

t_-/e_+

$$[\![\mu^-\alpha.\,c]\!]_1 \triangleq \lambda\alpha.[\![c]\!]$$
$$[\![x^-]\!]_1 \triangleq x$$
$$[\![\mu(x\cdot\alpha).\,c]\!]_1 \triangleq \lambda(x,\alpha).[\![c]\!]$$
$$[\![V_+]\!]_1 \triangleq \lambda k.k\,[\![V_+]\!]_0$$

V_+/S_-

$$[\![x_+]\!]_0 \triangleq x$$
$$[\![(V,W)]\!]_0 \triangleq ([\![V]\!]_0, [\![W]\!]_0)$$

Terms

$$[\![c]\!] : \bot$$

$$[\![A]\!]_2 \triangleq [\![A]\!]_1 \to \bot$$

$$[\![A]\!]_1 \triangleq [\![A]\!]_0 \to \bot$$

$$[\![A \otimes B]\!]_0 \triangleq [\![A]\!]_0 \times [\![A]\!]_0$$
$$\vdots$$

Types

Polarity: example

Recall: $t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

Consider $(\lambda x.\, t)\ u$:

$$\langle \lambda x.\, t \parallel\ u\ \cdot \star\rangle^{-}$$

If u has a negative type ($\varepsilon = -$), it is a value: $\triangleright^{*}\ t\ [u/x]$

$\rightsquigarrow$ through the CPS: $[\![\tilde{\mu}^{-} y.\, \langle - \parallel - \rangle^{-}]\!]_2\ [\![u]\!]_1$

If u has a positive type ($\varepsilon = +$) and is not a value, it gets reduced first.

$\rightsquigarrow$ through the CPS: $[\![u]\!]_2\ [\![\tilde{\mu}^{+} y.\, \langle - \parallel - \rangle^{-}]\!]_1$

$P \longrightarrow N$: call-by-value function

$\Downarrow M \longrightarrow N$: call-by-name function

Polarity: example

Recall: $t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

Consider $(\lambda x.\, t)\ u$:

$$\langle \lambda x.\, t \parallel u \cdot \star\rangle^{-}$$

If u has a negative type ($\varepsilon = -$), it is a value: $\triangleright^{*}\ t\,[u/x]$

$\rightsquigarrow$ through the CPS: $[\![\tilde{\mu}^{-} y.\, \langle - \parallel - \rangle^{-}]\!]_2\ [\![u]\!]_1$

If u has a positive type ($\varepsilon = +$) and is not a value, it gets reduced first.

$\rightsquigarrow$ through the CPS: $[\![u]\!]_2\ [\![\tilde{\mu}^{+} y.\, \langle - \parallel - \rangle^{-}]\!]_1$

$P \longrightarrow N$: call-by-value function

$\Downarrow M \longrightarrow N$: call-by-name function

Polarity: example

Recall: $t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

Consider $(\lambda x.\, t)\ u$:

$$\left\langle u \parallel \tilde{\mu}^{\varepsilon} y.\, \langle \lambda x.\, t \parallel\ y\ \cdot \star\rangle^{-}\right\rangle^{\varepsilon}$$

If u has a negative type ($\varepsilon = -$), it is a value: $\triangleright^{*}\ t\,[u/x]$

$\rightsquigarrow$ through the CPS: $[\![\tilde{\mu}^{-} y.\, \langle - \parallel -\rangle^{-}]\!]_2\ [\![u]\!]_1$

If u has a positive type ($\varepsilon = +$) and is not a value, it gets reduced first.

$\rightsquigarrow$ through the CPS: $[\![u]\!]_2\ [\![\tilde{\mu}^{+} y.\, \langle - \parallel -\rangle^{-}]\!]_1$

$P \longrightarrow N$: call-by-value function

$\Downarrow M \longrightarrow N$: call-by-name function

Polarity: example

Recall: $t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

Consider $(\lambda x.\, t)\ u$:

$$\left\langle u \parallel \tilde{\mu}^{\varepsilon} y.\, \langle \lambda x.\, t \parallel\ y\ \cdot \star\rangle^{-}\right\rangle^{\varepsilon}$$

If u has a negative type ($\varepsilon = -$), it is a value: $\triangleright^{*}\ t\,[u/x]$

$\rightsquigarrow$ through the CPS: $[\![\tilde{\mu}^{-} y.\, \langle - \parallel -\rangle^{-}]\!]_2\ [\![u]\!]_1$

If u has a positive type ($\varepsilon = +$) and is not a value, it gets reduced first.

$\rightsquigarrow$ through the CPS: $[\![u]\!]_2\ [\![\tilde{\mu}^{+} y.\, \langle - \parallel -\rangle^{-}]\!]_1$

$P \longrightarrow N$: call-by-value function

$\Downarrow M \longrightarrow N$: call-by-name function

Polarity: example

Recall: $t\ u \triangleq \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

Consider $(\lambda x.\, t)\ u$:

$$\left\langle\ u\ \middle\|\ \tilde{\mu}^{\varepsilon} y.\, \langle \lambda x.\, t \parallel\ y\ \cdot \star\rangle^{-}\right\rangle^{\varepsilon}$$

If u has a negative type ($\varepsilon = -$), it is a value: $\quad \triangleright^{*}\ t\,[u/x]$

$\rightsquigarrow$ through the CPS: $\quad [\![\tilde{\mu}^{-} y.\, \langle - \parallel - \rangle^{-}]\!]_2\ [\![u]\!]_1$

If u has a positive type ($\varepsilon = +$) and is not a value, it gets reduced first.

$\rightsquigarrow$ through the CPS: $\quad [\![u]\!]_2\ [\![\tilde{\mu}^{+} y.\, \langle - \parallel - \rangle^{-}]\!]_1$

$P \longrightarrow N$: call-by-value function

$\Downarrow M \longrightarrow N$: call-by-name function

Monolateral sequents

Involutive negation: *(cf Girard's LC)*

$$\boxed{A = A^{\perp\perp}}$$

+ usual duality laws: $(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}$ $(A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp}$

Sequents: from

$$c : (\Gamma \vdash \Delta) \qquad \Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta$$

to

$$c : (\vdash \Gamma^{\perp}, \Delta) \qquad \vdash \Gamma^{\perp}, \Delta \mid t : A \qquad \vdash \Gamma^{\perp}, \Delta \mid e : A^{\perp}$$

A **single grammar** to describe both expressions and contexts:

$$\begin{array}{lrcl} \textbf{Variables} & & & x, y, z \dots \\ \textbf{Values} & V & ::= & x \mid (V, V') \mid \mu(x, y).c \mid \mu^{-} x.c \\ \textbf{Terms} & t & ::= & \mu^{+} x.c \mid V^{\diamond} \\ \textbf{Commands} & c & ::= & \langle t \parallel V \rangle^{+} \end{array}$$

Monolateral sequents

Involutive negation: *(cf Girard's LC)*

$$\boxed{A = A^{\perp\perp}}$$

+ usual duality laws: $(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}$ $(A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp}$

Sequents: from

$$c : (\Gamma \vdash \Delta) \qquad \Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta$$

to

$$c : (\vdash \Gamma^{\perp}, \Delta) \qquad \vdash \Gamma^{\perp}, \Delta \mid t : A \qquad \vdash \Gamma^{\perp}, \Delta \mid e : A^{\perp}$$

A **single grammar** to describe both expressions and contexts:

Variables			$x, y, z \dots$
Values	V	$::=$	$x \mid (V, V') \mid \mu(x, y).c \mid \mu^{-} x.c$
Terms	t	$::=$	$\mu^{+} x.c \mid V^{\diamond}$
Commands	c	$::=$	$\langle t \parallel V \rangle^{+}$

Monolateral sequents

Involutive negation: *(cf Girard's LC)*

$$\boxed{A = A^{\perp\perp}}$$

+ usual duality laws: $(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}$ $(A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp}$

Sequents: from

$$c : (\Gamma \vdash \Delta) \qquad \Gamma \vdash t : A \mid \Delta \qquad \Gamma \mid e : A \vdash \Delta$$

to

$$c : (\vdash \Gamma^{\perp}, \Delta) \qquad \vdash \Gamma^{\perp}, \Delta \mid t : A \qquad \vdash \Gamma^{\perp}, \Delta \mid e : A^{\perp}$$

A **single grammar** to describe both expressions and contexts:

Variables			$x, y, z \ldots$
Values	V	$::=$	$x \mid (V, V') \mid \mu(x, y).c \mid \mu^{-} x.c$
Terms	t	$::=$	$\mu^{+} x.c \mid V^{\diamond}$
Commands	c	$::=$	$\langle t \parallel V \rangle^{+}$

Polarity: summary

We name "**L**" the polarized $\mu\tilde{\mu}$ calcul{us,i}.

A *type-directed* resolution of classical non-confluence.
Lets you mix call-by-value and call-by-name (like CBPV).

Let's make a bolder claim:

The computational interpretation of classical logic.

Constructive!

Pointer: Munch-Maccagnoni's PhD thesis (2013).

Polarity: summary

We name "**L**" the polarized $\mu\tilde{\mu}$ calcul{us,i}.

A *type-directed* resolution of classical non-confluence.
Lets you mix call-by-value and call-by-name (like CBPV).

Let's make a bolder claim:

The computational interpretation of classical logic.

Constructive!

Pointer: Munch-Maccagnoni's PhD thesis (2013).

Polarity: summary

We name "**L**" the polarized $\mu\tilde{\mu}$ calcul{us,i}.

A *type-directed* resolution of classical non-confluence.
Lets you mix call-by-value and call-by-name (like CBPV).

Let's make a bolder claim:

> **The** computational interpretation of classical logic.

Constructive!

Pointer: Munch-Maccagnoni's PhD thesis (2013).

Polarity: summary

We name "**L**" the polarized $\mu\tilde{\mu}$ calcul{us,i}.

A *type-directed* resolution of classical non-confluence.
Lets you mix call-by-value and call-by-name (like CBPV).

Let's make a bolder claim:

The computational interpretation of classical logic.

Constructive!

Pointer: Munch-Maccagnoni's PhD thesis (2013).

Polarity: summary

We name "**L**" the polarized $\mu\tilde{\mu}$ calcul{us,i}.

A *type-directed* resolution of classical non-confluence.
Lets you mix call-by-value and call-by-name (like CBPV).

Let's make a bolder claim:

> **The** computational interpretation of classical logic.

Constructive!

Pointer: Munch-Maccagnoni's PhD thesis (2013).

Let's get back to business

Goal #1

Define a polarised sequent calculus for a dependent type theory

Goal #2

Use it to factorize dependently typed CPS translations

Let's get back to business

Goal #1

Define a polarised sequent calculus for a dependent type theory

Goal #2

Use it to factorize dependently typed CPS translations

Dependent Type Theory

in a Polarised Sequent Calculus

Dependent Type Theory

Expressiveness: we target a type theory featuring:

- cumulative universes hierarchy $\square_i$
- impredicative propositional universe $\mathbb{P}$
- dependent sums $\Sigma(x : A).B$ and products $\Pi(x : A).B$
- Booleans with a dependent elimination principle
- an equality type

(this can be seen as an extension of Luo's **ECC**)

Types:

$$A, B \quad ::= \quad \square_i \mid \mathbb{P} \mid \Sigma(x : A).B \mid \Pi(x : A).B \mid \mathbb{B} \mid t = u$$

ECC type system

Universes

$$\frac{}{\Gamma \vdash \mathbb{P} : \square_0} \qquad \frac{\vdash \Gamma \text{ wf}}{\Gamma \vdash \square_i : \square_{i+1}} \qquad \frac{\Gamma \vdash A : \square_i \quad \Gamma, x : A \vdash B : \mathbb{P}}{\Gamma \vdash \Pi(x : A).B : \mathbb{P}}$$

Dependent types

$$\frac{\Gamma \vdash t : \Pi(x : A).B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[u/x]} \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B\,[t/x]}{\Gamma \vdash (t, u)_{\Sigma(x:A).B} : \Sigma(x : A).B} \qquad \frac{\Gamma \vdash t : \Sigma(x : A).B}{\Gamma \vdash \pi_2(t) : B[\pi_1(t)/x]}$$

$$\frac{\Gamma, x : B \vdash P : \square_i \quad \Gamma \vdash b : \mathbb{B} \quad \Gamma \vdash t : P[\text{true}/x] \quad \Gamma \vdash u : P[\text{false}/x]}{\Gamma \vdash \text{if } b \text{ then } t \text{ else } u : P[b/x]}$$

Subtyping

$$\frac{\Gamma \vdash e : A \quad \Gamma \vdash B : \square_i \quad \Gamma \vdash A \prec B}{\Gamma \vdash e : B} \qquad \frac{\Gamma \vdash A \equiv B}{\Gamma \vdash A \prec B}$$

$$\frac{\Gamma \vdash A \prec B \quad \Gamma \vdash B \prec C}{\Gamma \vdash A \prec C} \qquad \frac{\Gamma \vdash A' \prec A \quad \Gamma \vdash B \prec B'}{\Gamma \vdash \Pi(x : A).B \prec \Pi(x : A').B'}$$

ECC type system

Universes

$$\frac{}{\Gamma \vdash \mathbb{P} : \square_0} \qquad \frac{\vdash \Gamma \text{ wf}}{\Gamma \vdash \square_i : \square_{i+1}} \qquad \frac{\Gamma \vdash A : \square_i \quad \Gamma, x : A \vdash B : \mathbb{P}}{\Gamma \vdash \Pi(x : A).B : \mathbb{P}}$$

Dependent types

$$\frac{\Gamma \vdash t : \Pi(x : A).B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[u/x]} \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B\,[t/x]}{\Gamma \vdash (t, u)_{\Sigma(x:A).B} : \Sigma(x : A).B} \qquad \frac{\Gamma \vdash t : \Sigma(x : A).B}{\Gamma \vdash \pi_2(t) : B[\pi_1(t)/x]}$$

$$\frac{\Gamma, x : B \vdash P : \square_i \quad \Gamma \vdash b : \mathbb{B} \quad \Gamma \vdash t : P[\text{true}/x] \quad \Gamma \vdash u : P[\text{false}/x]}{\Gamma \vdash \text{if } b \text{ then } t \text{ else } u : P[b/x]}$$

Subtyping

$$\frac{\Gamma \vdash e : A \quad \Gamma \vdash B : \square_i \quad \Gamma \vdash A \prec B}{\Gamma \vdash e : B} \qquad \frac{\Gamma \vdash A \equiv B}{\Gamma \vdash A \prec B}$$

$$\frac{\Gamma \vdash A \prec B \quad \Gamma \vdash B \prec C}{\Gamma \vdash A \prec C} \qquad \frac{\Gamma \vdash A' \prec A \quad \Gamma \vdash B \prec B'}{\Gamma \vdash \Pi(x : A).B \prec \Pi(x : A').B'}$$

ECC type system

Universes

$$\frac{}{\Gamma \vdash \mathbb{P} : \square_0} \qquad \frac{\vdash \Gamma \text{ wf}}{\Gamma \vdash \square_i : \square_{i+1}} \qquad \frac{\Gamma \vdash A : \square_i \quad \Gamma, x : A \vdash B : \mathbb{P}}{\Gamma \vdash \Pi(x : A).B : \mathbb{P}}$$

Dependent types

$$\frac{\Gamma \vdash t : \Pi(x : A).B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[u/x]} \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B\,[t/x]}{\Gamma \vdash (t, u)_{\Sigma(x:A).B} : \Sigma(x : A).B} \qquad \frac{\Gamma \vdash t : \Sigma(x : A).B}{\Gamma \vdash \pi_2(t) : B[\pi_1(t)/x]}$$

$$\frac{\Gamma, x : B \vdash P : \square_i \quad \Gamma \vdash b : \mathbb{B} \quad \Gamma \vdash t : P[\text{true}/x] \quad \Gamma \vdash u : P[\text{false}/x]}{\Gamma \vdash \text{if } b \text{ then } t \text{ else } u : P[b/x]}$$

Subtyping

$$\frac{\Gamma \vdash e : A \quad \Gamma \vdash B : \square_i \quad \Gamma \vdash A \prec B}{\Gamma \vdash e : B} \qquad \frac{\Gamma \vdash A \equiv B}{\Gamma \vdash A \prec B}$$

$$\frac{\Gamma \vdash A \prec B \quad \Gamma \vdash B \prec C}{\Gamma \vdash A \prec C} \qquad \frac{\Gamma \vdash A' \prec A \quad \Gamma \vdash B \prec B'}{\Gamma \vdash \Pi(x : A).B \prec \Pi(x : A').B'}$$

The end of a fairy tale

Recall:

Abstract. We show that a minimal dependent type theory based on Σ-types and equality is degenerated in presence of computational classical logic. By computational classical logic is meant a classical logic derived from a control operator equipped with reduction rules similar to the ones of Felleisen's $\mathcal{C}$ or Parigot's μ operators. As a consequence, formalisms such as Martin-Löf's type theory or the (`Set`-predicative variant of the) Calculus of Inductive Constructions are inconsistent in presence of computational classical logic. Besides, an analysis of the role of the η-rule for control operators through a set-theoretic model of computational classical logic is given.

What about:

- (classical) sequent calculus + dependent types?
- value restriction?

The end of a fairy tale

Recall:

> **ABSTRACT**
>
> We investigate CPS translatability of typed λ-calculi with inductive and coinductive types. We show that tenable Plotkin-style call-by-name CPS translations exist for simply typed λ-calculi with a natural number type and stream types and, more generally, with arbitrary positive inductive and coinductive types. These translations also work in the presence of control operators and generalize for dependently typed calculi where case-like eliminations are only allowed in non-dependent forms. No translation is possible along the same lines for small Σ-types and sum types with dependent case.

What about:

- (classical) sequent calculus + dependent types?
- value restriction?

Sequent calculus | Call-by-value | Π-type

Can this work?

$$\dfrac{\dfrac{\begin{array}{c}\Pi_t\\ \vdots\\ \Gamma, x : A \vdash t : B[x] \mid \Delta\end{array}}{\Gamma \vdash \lambda x.t : \Pi(x : A).B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{array}{c}\Pi_u\\ \vdots\\ \Gamma \vdash u : A \mid \Delta\end{array} \quad \dfrac{\begin{array}{c}\Pi_e\\ \vdots\\ \Gamma \mid e : B[u] \vdash \Delta\end{array}}{\Gamma \mid u \cdot e : \Pi(x : A).B \vdash \Delta}\,(\to_l)}{\langle \lambda x.t \parallel u \cdot e \rangle : (\Gamma \vdash \Delta)}\,(\textsc{Cut})$$

Sequent calculus | Call-by-value | Π-type

Can this work?

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\Pi_t\\ \vdots\\ \Gamma, x : A \vdash t : B[x] \mid \Delta\end{array}}{\Gamma \vdash \lambda x.t : \Pi(x : A).B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{array}{c}\Pi_u\\ \vdots\\ \Gamma \vdash u : A \mid \Delta\end{array} \qquad \dfrac{\begin{array}{c}\Pi_e\\ \vdots\\ \Gamma \mid e : B[u] \vdash \Delta\end{array}}{\Gamma \mid u \cdot e : \Pi(x : A).B \vdash \Delta}\,(\to_l)}{\langle \lambda x.t \parallel u \cdot e \rangle : (\Gamma \vdash \Delta)}\,(\text{Cut})$$

$$\longrightarrow$$

$$\dfrac{\dfrac{\begin{array}{c}\Pi_u\\ \vdots\\ \Gamma \vdash u : A \mid \Delta\end{array} \qquad \dfrac{\dfrac{\Gamma, x : A \vdash t : \cancel{B[x]} \mid \Delta \qquad \Gamma, x : A \mid e : \cancel{B[u]} \vdash \Delta}{\langle t \parallel e \rangle : (\Gamma, x : A \vdash \Delta)}\,\textit{Mismatch}}{\Gamma \mid \tilde{\mu}x.\langle t \parallel e \rangle : A \vdash \Delta}\,(\tilde{\mu})}{\langle u \parallel \tilde{\mu}x.\langle t \parallel e \rangle \rangle : (\Gamma \vdash \Delta)}\,(\text{Cut})$$

Sequent calculus | Call-by-value | Π-type

Can this work? ✓

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\Pi_t\\ \vdots\\ \Gamma, x : A \vdash t : B[x] \mid \Delta\end{array}}{\Gamma \vdash \lambda x.t : \Pi(x : A).B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{array}{c}\Pi_u\\ \vdots\\ \Gamma \vdash u : A \mid \Delta\end{array} \quad \dfrac{\begin{array}{c}\Pi_e\\ \vdots\\ \Gamma \mid e : B[u] \vdash \Delta\end{array}}{\Gamma \mid u \cdot e : \Pi(x : A).B \vdash \Delta}\,(\to_l)}{\langle \lambda x.t \parallel u \cdot e \rangle : (\Gamma \vdash \Delta)}\,(\text{Cut})$$

$$\longrightarrow$$

$$\dfrac{\dfrac{\begin{array}{c}\Pi_u\\ \vdots\\ \Gamma \vdash u : A \mid \Delta\end{array} \qquad \dfrac{\dfrac{\Gamma, x : A \vdash t : B[x] \mid \Delta \quad \Gamma, x : A \mid e : B[u] \vdash \Delta; \{\cdot|t\}\{x|u\}}{\langle t \parallel e \rangle : \Gamma, x : A \vdash \Delta; \{x|u\}}\,(\text{Cut})}{\Gamma \mid \tilde{\mu}x.\langle t \parallel e \rangle : A \vdash \Delta; \{.|u\}}\,(\tilde{\mu})}{\langle u \parallel \tilde{\mu}x.\langle t \parallel e \rangle \rangle : (\Gamma \vdash \Delta); \{\cdot|\cdot\}}\,(\text{Cut})$$

CPS | Call-by-value | Π-type (1/2)

Is it enough?

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

$$[\![u]\!]\,[\![\tilde{\mu}x.\langle t \parallel e\rangle]\!] = \underbrace{[\![u]\!]}_{\neg\neg A}\,(\lambda x.\,\underbrace{[\![t]\!]}_{\neg\neg B(a)}\,\underbrace{[\![e]\!]}_{\neg B(q)})$$

This is quite normal:

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

↬ *we need to do this within the calculus!*

Intuition: $[\![t]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![u]\!]$ *has reduced*

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda x.t \parallel u \cdot e \rangle]\!] \xrightarrow{?} ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

Questions:

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

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda x.t \parallel u \cdot e \rangle]\!] \xrightarrow{?} ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

Questions:

1. Is any u compatible with such a reduction ?

- If u eventually gives a value V:

$$([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!] \to ((\lambda x.[\![t]\!])[\![V]\!])[\![e]\!] \to [\![t]\!][[\![V]\!]/x][\![e]\!] = [\![t[V/x]]\!][\![e]\!] \quad \checkmark$$

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

$$([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!] \to ((\lambda\_.v)\lambda x.[\![t]\!])[\![e]\!] \to v[\![e]\!] \quad \times$$

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda x.t \parallel u \cdot e \rangle]\!] \xrightarrow{?} ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

Questions:

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

- If u eventually gives a value V:

$$([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!] \to ((\lambda x.[\![t]\!])[\![V]\!])[\![e]\!] \to [\![t]\!][[\![V]\!]/x][\![e]\!] = [\![t[V/x]]\!][\![e]\!] \quad \checkmark$$

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

$$([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!] \to ((\lambda\_.v)\lambda x.[\![t]\!])[\![e]\!] \to v[\![e]\!] \quad \times$$

Negative-elimination free (Herbelin'12)

Values + one continuation variable + no application

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda x.t \parallel u \cdot e \rangle]\!] \xrightarrow{?} ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

Questions:

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

Naive attempt:

$$\underbrace{(\underbrace{[\![u]\!]}_{(A\to\bot)\to\bot}\;(\underbrace{\lambda x.[\![t]\!]}_{\Pi(x:A).\neg\neg B(x)}))}_{\text{✗}}\;\underbrace{[\![e]\!]}_{\neg B(u)}$$

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda x.t \parallel u \cdot e \rangle]\!] \xrightarrow{?} ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

Questions:

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

Friedman's trick:

$$\underbrace{\underbrace{(\quad [\![u]\!]}_{\forall R.(A\to R?)\to R?}\quad \underbrace{(\quad \lambda x.[\![t]\!]\quad))}_{\Pi(x:A).\neg\neg B(x)}}_{\neg\neg B}\quad \underbrace{[\![e]\!]}_{\neg B(u)}$$

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda x.t \parallel u \cdot e \rangle]\!] \xrightarrow{?} ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

Questions:

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

Better:

$$\underbrace{\underbrace{(\quad [\![u]\!]}_{\forall R.(\Pi((x{:}A)).R(x))\to R(u)} \quad \underbrace{(\quad \lambda x.[\![t]\!] \quad))}_{\Pi(x{:}A).\neg\neg B(x)}}_{\neg\neg B(u)} \quad \underbrace{[\![e]\!]}_{\neg B(u)}$$

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

Delimited continuations

$$[\![\langle \lambda x.p \parallel u \cdot e \rangle]\!] \longrightarrow ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

So, we're looking for:

$$\langle \lambda x.t \parallel u \cdot e \rangle \overset{u \in \mathsf{NEF}}{\longrightarrow} \left\langle \mu ?\,.\langle u \parallel \tilde{\mu} x.\langle t \parallel ? \rangle \rangle \,\middle\|\, e \right\rangle$$

such that we first reduce $\langle u \parallel \tilde{\mu} x.\langle t \parallel ? \rangle \rangle$.

Delimited continuations:

$$\langle \mu \hat{\mathrm{tp}}.c \parallel e \rangle \longrightarrow \langle \mu \hat{\mathrm{tp}}.c' \parallel e \rangle \qquad (\text{if } c \longrightarrow c')$$
$$\left\langle \mu \hat{\mathrm{tp}}.\langle t \parallel \mathrm{tp} \rangle \,\middle\|\, e \right\rangle \longrightarrow \langle t \parallel e \rangle$$

In other words:

$$u \cdot e \triangleq \lambda x.\left\langle \mu ?\,.\langle u \parallel \tilde{\mu} x.\langle t \parallel ? \rangle \rangle \,\middle\|\, e \right\rangle \qquad (u \in \mathsf{NEF})$$

Delimited continuations

$$[\![\langle \lambda x.p \parallel u \cdot e \rangle]\!] \longrightarrow ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

So, we're looking for:

$$\langle \lambda x.t \parallel u \cdot e \rangle \overset{u \in \mathsf{NEF}}{\longrightarrow} \left\langle \mu ?\,.\langle u \parallel \tilde{\mu} x.\langle t \parallel ? \rangle \rangle \,\middle\|\, e \right\rangle$$

such that we first reduce $\langle u \parallel \tilde{\mu} x.\langle t \parallel ? \rangle \rangle$.

Delimited continuations:

$$\langle \mu \hat{\mathrm{tp}}.c \parallel e \rangle \longrightarrow \langle \mu \hat{\mathrm{tp}}.c' \parallel e \rangle \qquad (\text{if } c \longrightarrow c')$$
$$\left\langle \mu \hat{\mathrm{tp}}.\langle t \parallel \hat{\mathrm{tp}} \rangle \,\middle\|\, e \right\rangle \longrightarrow \langle t \parallel e \rangle$$

In other words:

$$u \cdot e \triangleq \tilde{\mu} y.\left\langle \mu \hat{\mathrm{tp}}.\langle u \parallel \tilde{\mu} z.\langle t \parallel z \cdot \hat{\mathrm{tp}} \rangle \rangle \,\middle\|\, e \right\rangle \qquad (u \in \mathsf{NEF})$$

Delimited continuations

$$[\![\langle \lambda x.p \parallel u \cdot e \rangle]\!] \longrightarrow ([\![u]\!]\,(\lambda x.[\![t]\!]))[\![e]\!]$$

So, we're looking for:

$$\langle \lambda x.t \parallel u \cdot e \rangle \overset{u \in \mathsf{NEF}}{\longrightarrow} \left\langle \mu \hat{\mathrm{tp}}.\langle u \parallel \tilde{\mu} x.\langle t \parallel \hat{\mathrm{tp}} \rangle \rangle \,\middle\|\, e \right\rangle$$

such that we first reduce $\langle u \parallel \tilde{\mu} x.\langle t \parallel \hat{\mathrm{tp}} \rangle \rangle$.

Delimited continuations:

$$\begin{aligned} \langle \mu \hat{\mathrm{tp}}.c \parallel e \rangle &\longrightarrow \langle \mu \hat{\mathrm{tp}}.c' \parallel e \rangle && (\text{if } c \longrightarrow c') \\ \left\langle \mu \hat{\mathrm{tp}}.\langle t \parallel \hat{\mathrm{tp}} \rangle \,\middle\|\, e \right\rangle &\longrightarrow \langle t \parallel e \rangle \end{aligned}$$

In other words:

$$u \cdot e \triangleq \tilde{\mu} y.\left\langle \mu \hat{\mathrm{tp}}.\langle u \parallel \tilde{\mu} z.\langle t \parallel z \cdot \hat{\mathrm{tp}} \rangle \rangle \,\middle\|\, e \right\rangle \qquad (u \in \mathsf{NEF})$$

Call-by-value | Σ-type

Exact same story:

$$\dfrac{\dfrac{x : A \vdash x : A \quad y : \cancel{A[t]}\ \vdash y : \cancel{A[x]}}{x : A, y : B[t] \vdash (x, y) : \Sigma(x : A).B}\ \mathit{Mism.} \qquad \dfrac{\vdots}{\vdash u : B[t]}}{\vdots}$$

$$\dfrac{\dfrac{\vdash t : A \quad \vdash u : B[t]}{\vdash (t, u) : \Sigma(x : A).B} \quad \ldots}{\langle (t, u) \parallel e \rangle} \xrightarrow{?} \left\langle t \,\middle\|\, \tilde{\mu}x.\langle u \parallel \tilde{\mu}y.\langle (x, y) \parallel e \rangle \rangle \right\rangle$$

CPS:

$$[\![(t, u)]\!]_p\, k \triangleq [\![t]\!]_t\, (\lambda x.\ \underbrace{[\![u]\!]_p}_{\neg\neg B[t]}\ \underbrace{(\lambda x.k\,(x, y))}_{\neg B[x]})$$

$[\![u]\!]$ shouldn't be applied to its continuation before $[\![t]\!]$ has reduced

Call-by-value | Σ-type

$[\![u]\!]$ shouldn't be applied to its continuation before $[\![t]\!]$ has reduced

CPS:

$$[\![(t,u)]\!]_p\, k \triangleq \overbrace{[\![u]\!]_p}^{\neg\neg B[t]} \overbrace{(\underbrace{[\![t]\!]_t}_{\forall R.\Pi(x{:}A).R[x]\to R[t]} \quad \underbrace{(\lambda xy.k\,(x,y))}_{\Pi(x{:}A).\neg B[x]})}^{\neg B[t]}$$

Co-delimited continuations:

$$\langle (t,u) \parallel e\rangle \longrightarrow \left\langle u \,\middle\|\, \mu\check{\mathrm{tp}}.\left\langle t \,\middle\|\, \tilde{\mu}x.\left\langle \check{\mathrm{tp}} \,\middle\|\, \tilde{\mu}y.\langle (x,y) \parallel e\rangle\right\rangle\right\rangle\right\rangle$$

or (if $(t \in \textsc{nef})$):

$$(t,u) \triangleq \mu\alpha.\left\langle u \,\middle\|\, \mu\check{\mathrm{tp}}.\left\langle t \,\middle\|\, \tilde{\mu}x.\left\langle \check{\mathrm{tp}} \,\middle\|\, \tilde{\mu}y.\langle (x,y) \parallel \alpha\rangle\right\rangle\right\rangle\right\rangle$$

Recap

So far:

- problems for naive $\Pi(x : A).B$ and $\Sigma(x : A).B$
- both solved with delimited continuations

In terms of L:

$$(t, u) \qquad (t \cdot e)$$

require:

- $t \in \mathrm{NEF}$
- adequate definitions/ζ-rules using delimited continuations

Observation: exactly dual situations $\rightsquigarrow$ monolateral syntax

Good news

The same method applies to Booleans, etc.

Recap

So far:

- problems for naive $\Pi(x : A).B$ and $\Sigma(x : A).B$
- both solved with delimited continuations

In terms of L:

$$(t, u) \qquad (t \cdot e)$$

require:

- $t \in \mathsf{NEF}$
- adequate definitions/ζ-rules using delimited continuations

Observation: exactly dual situations $\rightsquigarrow$ monolateral syntax

Good news

The same method applies to Booleans, etc.

Recap

So far:

- problems for naive $\Pi(x : A).B$ and $\Sigma(x : A).B$
- both solved with delimited continuations

In terms of L:

$$(t, u) \qquad (t \cdot e)$$

require:

- $t \in \mathsf{NEF}$
- adequate definitions/ζ-rules using delimited continuations

Observation: exactly dual situations $\rightsquigarrow$ monolateral syntax

Good news

The same method applies to Booleans, etc.

Our proposal: L_{dep}

$\mathsf{L}_{\mathsf{dep}}$

Monolateral syntax

Atoms	C	::=	$x \mid \mathbb{B} \mid \mathbb{P} \mid \square_i \mid t = u$
Types$^+$	P	::=	$C \mid A \otimes x.B \mid \Downarrow A$
Types$^-$	N	::=	$C^\perp \mid A \parr x.B \mid \Uparrow A$
Types	A	::=	$P \mid N$
Values	V	::=	$x \mid A \mid V \otimes_A V' \mid \mathsf{true} \mid \mathsf{false} \mid \mathtt{refl}$
		$\mid$	$\mu(x \otimes_A \alpha).c \mid \mu^- x.c \mid \mu[c_1 \mid c_2] \mid \mu{=}c \mid \hat{\mu} c$
Terms	t	::=	$\mu^+ x.c \mid \text{ˆ} \mid V^\diamond$
Commands	c	::=	$\langle t \parallel V \rangle^+$

Expressiveness

$\mathsf{L}_{\mathsf{dep}}$

Monolateral syntax

Atoms	C	::=	$x \mid \mathbb{B} \mid \mathbb{P} \mid \square_i \mid t = u$
Types$^+$	P	::=	$C \mid A \otimes x.B \mid \Downarrow A$
Types$^-$	N	::=	$C^\perp \mid A \parr x.B \mid \Uparrow A$
Types	A	::=	$P \mid N$
Values	V	::=	$x \mid A \mid V \otimes_A V' \mid \mathsf{true} \mid \mathsf{false} \mid \mathtt{refl}$
		$\mid$	$\mu(x \otimes_A \alpha).c \mid \mu^- x.c \mid \mu[c_1 \mid c_2] \mid \mu{=}c \mid \hat{\mu} c$
Terms	t	::=	$\mu^+ x.c \mid \text{ˆ} \mid V^\diamond$
Commands	c	::=	$\langle t \parallel V \rangle^+$

Expressiveness

1) Negative commands:

$$\langle V \parallel t \rangle^- \triangleq \langle t \parallel V \rangle^+$$

$\mathsf{L}_{\mathsf{dep}}$

Monolateral syntax

Atoms	C	::=	$x \mid \mathbb{B} \mid \mathbb{P} \mid \square_i \mid t = u$
Types$^+$	P	::=	$C \mid A \otimes x.B \mid \Downarrow A$
Types$^-$	N	::=	$C^\perp \mid A \parr x.B \mid \Uparrow A$
Types	A	::=	$P \mid N$
Values	V	::=	$x \mid A \mid V \otimes_A V' \mid \mathsf{true} \mid \mathsf{false} \mid \mathtt{refl}$
		$\mid$	$\mu(x \otimes_A \alpha).c \mid \mu^- x.c \mid \mu[c_1 \mid c_2] \mid \mu{=}c \mid \hat{\mu} c$
Terms	t	::=	$\mu^+ x.c \mid \text{ˆ} \mid V^\diamond$
Commands	c	::=	$\langle t \parallel V \rangle^+$

Expressiveness

2) Types:

$$\Sigma(x : A).B := A \otimes x.B \qquad \Pi(x : A).B := A^\perp \parr x.B$$

$\mathbf{L}_{\mathbf{dep}}$

Monolateral syntax

Atoms	C	::=	$x \mid \mathbb{B} \mid \mathbb{P} \mid \square_i \mid t = u$
Types$^+$	P	::=	$C \mid A \otimes x.B \mid \Downarrow A$
Types$^-$	N	::=	$C^\perp \mid A \parr x.B \mid \Uparrow A$
Types	A	::=	$P \mid N$
Values	V	::=	$x \mid A \mid V \otimes_A V' \mid \mathsf{true} \mid \mathsf{false} \mid \mathtt{refl}$
		$\mid$	$\mu(x \otimes_A \alpha).c \mid \mu^- x.c \mid \mu[c_1 \mid c_2] \mid \mu{=}c \mid \hat{\mu} c$
Terms	t	::=	$\mu^+ x.c \mid \text{ˆ} \mid V^\diamond$
Commands	c	::=	$\langle t \parallel V \rangle^+$

Expressiveness

3) Dependent types:

$$T(b) = \mu^+ x.\left\langle b \,\middle\|\, \mu[\langle \mathbb{P} \parallel x\rangle^+ \mid \langle \mathbb{B} \parallel x\rangle^+]\right\rangle^+$$

such that $T(\mathsf{true}) \equiv \mathbb{P}$ and $T(\mathsf{false}) \equiv \mathbb{B}$

Reductions

$$
\begin{array}{lll}
(\mu^+) & \langle \mu^+ x.c \parallel V \rangle^+ & \rhd_{\mathsf{R}} \quad c[V/x] \\
(\mu^-) & \langle V \parallel \mu^- x.c \rangle^+ & \rhd_{\mathsf{R}} \quad c[V/x] \\
(\mu \otimes) & \langle (V \otimes_A V') \parallel \mu(x \otimes_A y).c \rangle^+ & \rhd_{\mathsf{R}} \quad c[V/x, V'/y] \\
(\mu{=}) & \langle \mathtt{refl} \parallel \mu{=}c \rangle^+ & \rhd_{\mathsf{R}} \quad c \\
(\mu^1_{\mathbb{B}}) & \langle \mathsf{true} \parallel \mu[c_1 \mid c_2] \rangle^+ & \rhd_{\mathsf{R}} \quad c_1 \\
(\mu^2_{\mathbb{B}}) & \langle \mathsf{false} \parallel \mu[c_1 \mid c_2] \rangle^+ & \rhd_{\mathsf{R}} \quad c_2 \\
(\hat{\mu}) & \hat{\mu}\langle \hat{} \parallel V \rangle^+ & \rhd_{\mathsf{R}} \quad V
\end{array}
$$

Pairs:

$$t \otimes u \quad := \quad \mu^+ z.\left\langle t \,\middle\|\, \mu^- x.\left\langle u \,\middle\|\, \mu^- y.\langle x \otimes_{\_} y \parallel z \rangle^+ \right\rangle^+ \right\rangle^+$$

Dependent pairs: $(t \in \mathsf{NEF})$

$$t \otimes_A u \quad := \quad \mu^+ z.\left\langle u \,\middle\|\, \hat{\mu}\left\langle t \,\middle\|\, \mu^- x.\left\langle \mu^- y.\langle z \parallel x \otimes_A y \rangle^- \,\middle\|\, \hat{} \right\rangle^- \right\rangle^+ \right\rangle^+$$

Reductions

$$
\begin{array}{lll}
(\mu^+) & \langle \mu^+ x.c \parallel V \rangle^+ & \rhd_{\mathsf{R}} \quad c[V/x] \\
(\mu^-) & \langle V \parallel \mu^- x.c \rangle^+ & \rhd_{\mathsf{R}} \quad c[V/x] \\
(\mu \otimes) & \langle (V \otimes_A V') \parallel \mu(x \otimes_A y).c \rangle^+ & \rhd_{\mathsf{R}} \quad c[V/x, V'/y] \\
(\mu{=}) & \langle \mathtt{refl} \parallel \mu{=}c \rangle^+ & \rhd_{\mathsf{R}} \quad c \\
(\mu^1_{\mathbb{B}}) & \langle \mathsf{true} \parallel \mu[c_1 \mid c_2] \rangle^+ & \rhd_{\mathsf{R}} \quad c_1 \\
(\mu^2_{\mathbb{B}}) & \langle \mathsf{false} \parallel \mu[c_1 \mid c_2] \rangle^+ & \rhd_{\mathsf{R}} \quad c_2 \\
(\hat{\mu}) & \hat{\mu}\langle \hat{} \parallel V \rangle^+ & \rhd_{\mathsf{R}} \quad V
\end{array}
$$

Pairs:

$$t \otimes u \quad := \quad \mu^+ z.\left\langle t \,\middle\|\, \mu^- x.\left\langle u \,\middle\|\, \mu^- y.\langle x \otimes_{\_} y \parallel z \rangle^+ \right\rangle^+ \right\rangle^+$$

Dependent pairs: $(t \in \mathsf{NEF})$

$$t \otimes_A u \quad := \quad \mu^+ z.\left\langle u \,\middle\|\, \hat{\mu}\left\langle t \,\middle\|\, \mu^- x.\left\langle \mu^- y.\langle z \parallel x \otimes_A y \rangle^- \,\middle\|\, \hat{} \right\rangle^- \right\rangle^+ \right\rangle^+$$

Type system

Two typing modes:

- **regular** for non-dependent derivations
- **dependent** otw., delimited by $\hat{\mu}/\hat{}$

Type system

Two typing modes:

- **regular** for non-dependent derivations
- **dependent** otw., delimited by $\hat{\mu}/\hat{}$

$$\frac{\vdash \Gamma \mid t : P \quad \vdash \Gamma \mid V : P^{\perp}}{\langle t \parallel V \rangle^{+} : (\vdash \Gamma)}$$

$$\frac{c : (\vdash \Gamma, x : N)}{\vdash \Gamma \mid \mu^{-} x.c : N}$$

$$\frac{c : (\vdash_{N} \Gamma)}{\vdash \Gamma \mid \hat{\mu} c : N}$$

Regular mode

$$\frac{\vdash \Gamma \mid t : P \quad \vdash_{B[\bullet]} \Gamma \mid V : P^{\perp}}{\langle t \parallel V \rangle^{+} : (\vdash_{B[t]} \Gamma)} \; (t \in \mathsf{NEF})$$

$$\frac{c : (\vdash_{B[x]} \Gamma, x : N)}{\vdash_{B[\bullet]} \Gamma \mid \mu^{-} x.c : N}$$

$$\frac{\bullet \notin B}{\vdash_{B} \Gamma \mid \hat{} : B^{\perp}}$$

Dependent mode

Type system

Two typing modes:

- **regular** for non-dependent derivations
- **dependent** otw., delimited by $\hat{\mu}/\hat{}$

Example: $T(b)$ s.t. $T(\text{true}) \equiv \mathbb{P}$ and $T(\text{false}) \equiv \mathbb{B}$
Intuitively:

$$b : \mathbb{B} \vdash \text{if } b \text{ then } \Pi(X : \mathbb{P}).X \text{ else true} : T(b)$$

In our setting:

$$\vdash b : \mathbb{B}^{\perp} \mid \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{} \rangle^{-} \mid \langle \text{true} \parallel \hat{} \rangle^{-}]\right\rangle^{+} : T(b)$$

Type system

Two typing modes:

- **regular** for non-dependent derivations
- **dependent** otw., delimited by $\hat{\mu}/\hat{}$

Example: $T(b)$ s.t. $T(\text{true}) \equiv \mathbb{P}$ and $T(\text{false}) \equiv \mathbb{B}$
Intuitively:

$$b : \mathbb{B} \vdash \text{if } b \text{ then } \Pi(X : \mathbb{P}).X \text{ else true} : T(b)$$

In our setting:

$$\vdash b : \mathbb{B}^{\perp} \mid \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{} \rangle^{-} \mid \langle \text{true} \parallel \hat{} \rangle^{-}]\right\rangle^{+} : T(b)$$

Type system

Two typing modes:

- **regular** for non-dependent derivations
- **dependent** otw., delimited by $\hat{\mu}/\hat{}$

Example: $T(b)$ s.t. $T(\text{true}) \equiv \mathbb{P}$ and $T(\text{false}) \equiv \mathbb{B}$
Intuitively:

$$b : \mathbb{B} \vdash \text{if } b \text{ then } \Pi(X : \mathbb{P}).X \text{ else true} : T(b)$$

In our setting:

$$\dfrac{\vdash b : \mathbb{B}^{\perp} \mid b : \mathbb{B} \qquad \dfrac{\dfrac{\vdash\mid \Pi(X : \mathbb{P}).X : T(\text{true})}{\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} : (\vdash_{T(\text{true})})} \qquad \dfrac{\vdash\mid \text{true} : T(\text{false})}{\langle \text{true} \parallel \hat{}\rangle^{-} : (\vdash_{T(\text{false})})}}{\vdash_{T(\bullet)} b : \mathbb{B}^{\perp} \mid \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} \mid \langle \text{true} \parallel \hat{}\rangle^{-}] : \mathbb{B}^{\perp}}}{\dfrac{\left\langle b \,\middle\|\, \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} \mid \langle \text{true} \parallel \hat{}\rangle^{-}] \right\rangle^{+} : (\vdash_{T(b)} b : \mathbb{B}^{\perp})}{\vdash b : \mathbb{B}^{\perp} \mid \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} \mid \langle \text{true} \parallel \hat{}\rangle^{-}] \right\rangle^{+} : T(b)}}$$

Type system

Two typing modes:

- **regular** for non-dependent derivations
- **dependent** otw., delimited by $\hat{\mu}/\hat{}$

Example: $T(b)$ s.t. $T(\text{true}) \equiv \mathbb{P}$ and $T(\text{false}) \equiv \mathbb{B}$
Intuitively:

$$b : \mathbb{B} \vdash \text{if } b \text{ then } \Pi(X : \mathbb{P}).X \text{ else true} : T(b)$$

In our setting:

$$\dfrac{\vdash b : \mathbb{B}^{\perp} \mid b : \mathbb{B} \qquad \dfrac{\dfrac{\vdash\mid \Pi(X : \mathbb{P}).X : T(\text{true})}{\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} : (\vdash_{T(\text{true})})} \qquad \dfrac{\vdash\mid \text{true} : T(\text{false})}{\langle \text{true} \parallel \hat{}\rangle^{-} : (\vdash_{T(\text{false})})}}{\vdash_{T(\bullet)} b : \mathbb{B}^{\perp} \mid \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} \mid \langle \text{true} \parallel \hat{}\rangle^{-}] : \mathbb{B}^{\perp}}}{\dfrac{\left\langle b \,\middle\|\, \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} \mid \langle \text{true} \parallel \hat{}\rangle^{-}] \right\rangle^{+} : (\vdash_{T(b)} b : \mathbb{B}^{\perp})}{\vdash b : \mathbb{B}^{\perp} \mid \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle \Pi(X : \mathbb{P}).X \parallel \hat{}\rangle^{-} \mid \langle \text{true} \parallel \hat{}\rangle^{-}] \right\rangle^{+} : T(b)}}$$

CPS translation

Two translations:

1. $t : A$ in regular mode translated to $[t]^{\neg} : [\![A]\!]^{\neg}$ (negative transl.)
2. $t : A$ in dependent mode translated to $[t] : [\![A]\!]^{\mathrm{d}}[t]_v$, where:
   - $[t]_v$ is the value that t is equivalent to through the CPS
   - $[\![A]\!]^{\mathrm{d}}$ uses a parametric and dependent return type:

$$[\![A]\!]^{\mathrm{d}} \equiv \lambda y : [\![A]\!]_v.\Pi(R : [\![A]\!]_v \to \mathbb{P}).((\Pi(x : \mathbb{B}).R\,x) \to R\,y))$$

CPS translation

Two translations:

1. $t : A$ in regular mode translated to $[t]^{\neg} : [\![A]\!]^{\neg}$ (negative transl.)
2. $t : A$ in dependent mode translated to $[t] : [\![A]\!]^{\mathrm{d}}[t]_v$, where:
   - $[t]_v$ is the value that t is equivalent to through the CPS
   - $[\![A]\!]^{\mathrm{d}}$ uses a parametric and dependent return type:

$$[\![A]\!]^{\mathrm{d}} \equiv \lambda y : [\![A]\!]_v.\Pi(R : [\![A]\!]_v \to \mathbb{P}).((\Pi(x : \mathbb{B}).R\,x) \to R\,y))$$

Indeed, by parametricity we have:

Fact

If $\Gamma \vdash t : \Pi(R : A \to \mathbb{P}).(\Pi(x : A).R\,x \to R\,u)$ then $t\,R\,k = k\,u$.

Example:

$$\begin{aligned}[\mathrm{true}]^{\mathrm{d}} &:= \lambda(R : \mathbb{B} \to \mathbb{P}).\lambda(k : \Pi(x : \mathbb{B}).R\,x).k\,\mathrm{true} \\ &: \Pi(R : \mathbb{B} \to \mathbb{P}).((\Pi(x : \mathbb{B}).R\,x) \to R\,\mathrm{true}))\end{aligned}$$

CPS translation

Two translations:

1. $t : A$ in regular mode translated to $[t]^{\neg} : [\![A]\!]^{\neg}$ (negative transl.)
2. $t : A$ in dependent mode translated to $[t] : [\![A]\!]^{\mathrm{d}}[t]_v$, where:
   - $[t]_v$ is the value that t is equivalent to through the CPS
   - $[\![A]\!]^{\mathrm{d}}$ uses a parametric and dependent return type:

$$[\![A]\!]^{\mathrm{d}} \equiv \lambda y : [\![A]\!]_v.\Pi(R : [\![A]\!]_v \to \mathbb{P}).((\Pi(x : \mathbb{B}).R\,x) \to R\,y))$$

Indeed, by parametricity we have:

Fact

If $\Gamma \vdash t : \Pi(R : A \to \mathbb{P}).(\Pi(x : A).R\,x \to R\,u)$ then $t\,R\,k = k\,u$.

Example:

$$\begin{array}{rcl}[\mathrm{true}]^{\mathrm{d}} & := & \lambda(R : \mathbb{B} \to \mathbb{P}).\lambda(k : \Pi(x : \mathbb{B}).R\,x)).k\,\mathrm{true} \\ & : & \Pi(R : \mathbb{B} \to \mathbb{P}).((\Pi(x : \mathbb{B}).R\,x) \to R\,\mathrm{true}))\end{array}$$

The importance of return types

Soundness

- If $\vdash_{N[\bullet]} \Gamma \mid t : A^{\perp}$ then $[\![\Gamma]\!]^{\mathrm{d}}_{v} \vdash [t]^{\mathrm{d}} : \Pi(x : [\![A]\!]^{\mathrm{d}}_{v}).[\![N[x]]\!]^{\neg}$
- If $c : (\vdash_{N} \Gamma)$ then $[\![\Gamma]\!]^{\mathrm{d}}_{v} \vdash [c]^{\mathrm{d}} : [\![N]\!]^{\neg}$.
- ...

Example:

Through the CPS:

$$[\text{if } b \text{ then } t \text{ else } u : T(b)]^{\mathrm{d}}$$
$$= \lambda(k : \neg[\![T(b)]\!]^{\neg}_{v}).\big([b]^{\mathrm{d}}\, R\, \underbrace{(\text{if } x \text{ then } [t]^{\mathrm{d}}\,(R\,\text{true}) \text{ else } [u]^{\mathrm{d}}\,(R\,\text{false}))}_{\Pi(x:\mathbb{B}).R\,x}\big)\, k$$

where $R := \lambda(x : \mathbb{B}).[\![T(x)]\!]^{\neg}$ is indeed of type $\mathbb{B} \to \mathbb{P}$.

The importance of return types

Soundness

- If $\vdash_{N[\bullet]} \Gamma \mid t : A^{\perp}$ then $[\![\Gamma]\!]^{\mathrm{d}}_{v} \vdash [t]^{\mathrm{d}} : \Pi(x : [\![A]\!]^{\mathrm{d}}_{v}).[\![N[x]]\!]^{\neg}$
- If $c : (\vdash_{N} \Gamma)$ then $[\![\Gamma]\!]^{\mathrm{d}}_{v} \vdash [c]^{\mathrm{d}} : [\![N]\!]^{\neg}$.
- ...

Example:

$$\frac{\vdash b : \mathbb{B} \quad \vdash t : T(\mathrm{true}) \quad \vdash u : T(\mathrm{false})}{\vdash \mathrm{if}\ b\ \mathrm{then}\ t\ \mathrm{else}\ u : T(b)}$$

Through the CPS:

$$[\mathrm{if}\ b\ \mathrm{then}\ t\ \mathrm{else}\ u : T(b)]^{\mathrm{d}}$$
$$= \lambda(k : \neg[\![T(b)]\!]^{\neg}_{v}).\big([b]^{\mathrm{d}}\ R\ \underbrace{(\mathrm{if}\ x\ \mathrm{then}\ [t]^{\mathrm{d}}\ (R\,\mathrm{true})\ \mathrm{else}\ [u]^{\mathrm{d}}\ (R\,\mathrm{false}))}_{\Pi(x:\mathbb{B}).R\,x}\big)\ k$$

where $R := \lambda(x : \mathbb{B}).[\![T(x)]\!]^{\neg}$ is indeed of type $\mathbb{B} \to \mathbb{P}$.

The importance of return types

Soundness

- If $\vdash_{N[\bullet]} \Gamma \mid t : A^{\perp}$ then $[\![\Gamma]\!]^{\mathrm{d}}_{v} \vdash [t]^{\mathrm{d}} : \Pi(x : [\![A]\!]^{\mathrm{d}}_{v}).[\![N[x]]\!]^{\neg}$
- If $c : (\vdash_{N} \Gamma)$ then $[\![\Gamma]\!]^{\mathrm{d}}_{v} \vdash [c]^{\mathrm{d}} : [\![N]\!]^{\neg}$.
- ...

Example:

$$\dfrac{\dfrac{\vdash \cdot \mid b : \mathbb{B} \qquad \vdash_{T(\bullet)} \mid \mu[\langle t \parallel \hat{} \rangle^{-} \mid \langle u \parallel \hat{} \rangle^{-}] : \mathbb{B}^{\perp}}{\left\langle b \,\middle\|\, \mu[\langle t \parallel \hat{} \rangle^{-} \mid \langle u \parallel \hat{} \rangle^{-}] \right\rangle^{+} : (\vdash_{T(b)} \cdot)}}{\vdash \cdot \mid \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle t \parallel \hat{} \rangle^{-} \mid \langle u \parallel \hat{} \rangle^{-}] \right\rangle^{+} : T(b)}$$

$$\dfrac{\dfrac{\vdash \cdot \mid t : T(\mathrm{true})}{\langle t \parallel \hat{} \rangle^{-} : (\vdash_{T(\mathrm{true})})} \qquad \dfrac{\vdash \cdot \mid u : T(\mathrm{false})}{\langle u \parallel \hat{} \rangle^{-} : (\vdash_{T(\mathrm{false})})}}{\vdash_{T(\bullet)} \mid \mu[\langle t \parallel \hat{} \rangle^{-} \mid \langle u \parallel \hat{} \rangle^{-}] : \mathbb{B}^{\perp}}$$

Through the CPS:

$$[\text{if } b \text{ then } t \text{ else } u : T(b)]^{\mathrm{d}}$$
$$= \lambda(k : \neg[\![T(b)]\!]^{\neg}_{v}).\big([b]^{\mathrm{d}}\, R\, \underbrace{(\text{if } x \text{ then } [t]^{\mathrm{d}}\, (R\,\mathrm{true}) \text{ else } [u]^{\mathrm{d}}\, (R\,\mathrm{false}))}_{\Pi(x:\mathbb{B}).R\,x}\big)\, k$$

where $R := \lambda(x : \mathbb{B}).[\![T(x)]\!]^{\neg}$ is indeed of type $\mathbb{B} \to \mathbb{P}$.

Universes

$$\frac{}{\vdash \square_i : \square_{i+1}}$$

General structure:

$$\begin{array}{l} \vdash t : A \\ \vdash V : A \end{array} \quad \rightsquigarrow \quad \begin{array}{l} \vdash [t] : [\![A]\!] \\ \vdash [V]_v : [\![A]\!]_v \end{array}$$

Negative translation :

Value type	$[\![\square_i]\!]_v^\neg$	$\triangleq \square_i^\neg$	$[\![A]\!]_v^{\mathrm{d}} = \ldots$
Term type	$[\![\square_i]\!]^\neg$	$\triangleq \mathcal{F}^\neg(\square_i^\neg)$	$[\![A]\!]^{\mathrm{d}} = \mathcal{F}^{\mathrm{d}}([\![A]\!]_v^{\mathrm{d}})$
Value	$[\square_i]_v^\neg$	$\triangleq (\square_i^\neg, \mathcal{F}^\neg(\square_i^\neg))$	$[A]_v^{\mathrm{d}} = ([\![A]\!]_v^{\mathrm{d}}, [\![A]\!]^{\mathrm{d}})$

Universes

$$\frac{}{\vdash \square_i : \square_{i+1}}$$

General structure:

$$\begin{array}{l} \vdash t : A \\ \vdash V : A \end{array} \quad \rightsquigarrow \quad \begin{array}{l} \vdash [t] : [\![A]\!] \\ \vdash [V]_v : [\![A]\!]_v \end{array}$$

Negative translation :

$$\begin{array}{rcl} \mathcal{F}^{\neg}(A) & := & \neg\neg A \\ \square_i^{\neg} & := & \Sigma(A : \square_i).\mathbb{P} \end{array}$$

Value type	$[\![\square_i]\!]_v^{\neg}$	$\triangleq \square_i^{\neg}$	$[\![A]\!]_v^{\mathrm{d}} = \ldots$
Term type	$[\![\square_i]\!]^{\neg}$	$\triangleq \mathcal{F}^{\neg}(\square_i^{\neg})$	$[\![A]\!]^{\mathrm{d}} = \mathcal{F}^{\mathrm{d}}([\![A]\!]_v^{\mathrm{d}})$
Value	$[\square_i]_v^{\neg}$	$\triangleq (\square_i^{\neg}, \mathcal{F}^{\neg}(\square_i^{\neg}))$	$[A]_v^{\mathrm{d}} = ([\![A]\!]_v^{\mathrm{d}}, [\![A]\!]^{\mathrm{d}})$

Universes

$$\frac{}{\vdash \square_i : \square_{i+1}}$$

General structure:

$$\begin{array}{l} \vdash t : A \\ \vdash V : A \end{array} \quad \rightsquigarrow \quad \begin{array}{l} \vdash [t] : [\![A]\!] \\ \vdash [V]_v : [\![A]\!]_v \end{array}$$

Dependent translation:

$$\begin{array}{rcl} \mathcal{F}^{\mathrm{d}}(A) & \triangleq & \lambda(z : A).\Pi(R : A \to \mathbb{P}).\Pi(x : A).R\,x \to R\,z \\ \square_i^{\mathrm{d}} & \triangleq & \Sigma(A : \square_i).A \to \mathbb{P} \end{array}$$

Value type	$[\![\square_i]\!]_v^{\mathrm{d}} \triangleq \square_i^{\mathrm{d}}$		$[\![A]\!]_v^{\mathrm{d}} = \ldots$
Term type	$[\![\square_i]\!]^{\mathrm{d}} \triangleq \mathcal{F}^{\mathrm{d}}(\square_i^{\mathrm{d}})$		$[\![A]\!]^{\mathrm{d}} = \mathcal{F}^{\mathrm{d}}([\![A]\!]_v^{\mathrm{d}})$
Value	$[\square_i]_v^{\mathrm{d}} \triangleq (\square_i^{\mathrm{d}}, \mathcal{F}^{\mathrm{d}}(\square_i^{\mathrm{d}}))$		$[A]_v^{\mathrm{d}} = ([\![A]\!]_v^{\mathrm{d}}, [\![A]\!]^{\mathrm{d}})$

ECC

CbV **ECC** embeds into $\mathbf{L}_{\mathbf{dep}}$

Types:

$$\begin{array}{rcl@{\qquad}rcl} \Sigma(x:A).B & \triangleq & A \otimes x.B & \mathbb{B} & \triangleq & \mathbb{B} \\ \Pi(x:A).B & \triangleq & A^{\perp} \parr x.B & \square_i & \triangleq & \square_i \end{array}$$

Terms:

$$\begin{array}{rcl} (t,u)_{\Sigma(x:A).B} & \triangleq & \mu^{+}z.\left\langle u \,\middle\|\, \hat{\mu}\left\langle t \,\middle\|\, \mu^{-}x.\left\langle \mu^{-}y.\langle z \parallel x \otimes_A y\rangle^{-} \parallel \star \right\rangle^{-} \right\rangle^{+} \right\rangle^{+} \\ \text{if } b \text{ then } t \text{ else } u & \triangleq & \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle t \parallel \star\rangle^{-} \mid \langle u \parallel \star\rangle^{-}] \right\rangle^{+} \end{array}$$

Incidentally:

ECC

CbV **ECC** embeds into $\mathbf{L}_{\mathbf{dep}}$

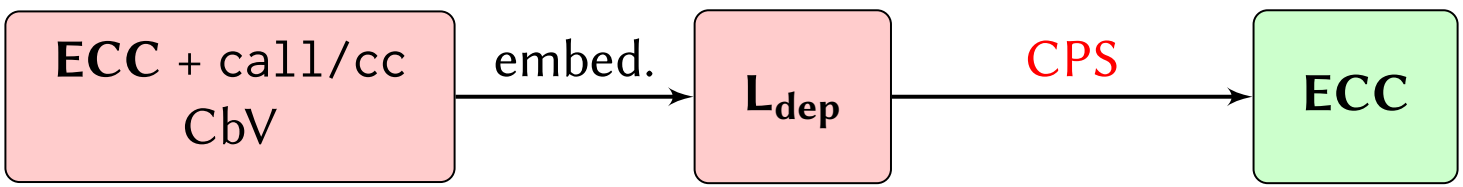
Types:

$$\Sigma(x:A).B \triangleq A \otimes x.B \qquad \mathbb{B} \triangleq \mathbb{B}$$
$$\Pi(x:A).B \triangleq A^{\perp} \parr x.B \qquad \square_i \triangleq \square_i$$

Terms:

$$(t,u)_{\Sigma(x:A).B} \triangleq \mu^{+}z.\left\langle u \,\middle\|\, \hat{\mu}\left\langle t \,\middle\|\, \mu^{-}x.\left\langle \mu^{-}y.\langle z \parallel x \otimes_A y\rangle^{-} \,\middle\|\, \star\right\rangle^{-}\right\rangle^{+}\right\rangle^{+}$$
$$\text{if } b \text{ then } t \text{ else } u \triangleq \hat{\mu}\left\langle b \,\middle\|\, \mu[\langle t \parallel \star\rangle^{-} \mid \langle u \parallel \star\rangle^{-}]\right\rangle^{+}$$

Incidentally:



Pure case

Through CPS translations:

Every intuitionistic term eventually returns a value

(a.k.a. every term is thunkable)

Result

We obtain a CPS from CbV ECC to ECC.

Improve over previous results (M., Bowman et al.):

unrestricted, dependently typed, no extra assumptions!

Check `www.irif.fr/~emiquey/content/CPS_ECC.v`

Pure case

Through CPS translations:

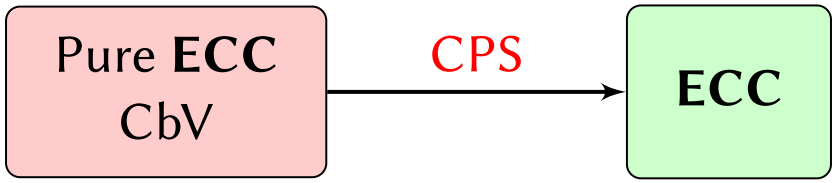
Every intuitionistic term eventually returns a value

(a.k.a. every term is thunkable)

Result

We obtain a CPS from CbV ECC to ECC.

Improve over previous results (M., Bowman et al.):



unrestricted, dependently typed, no extra assumptions!

Check www.irif.fr/~emiquey/content/CPS_ECC.v

Pure case

Through CPS translations:

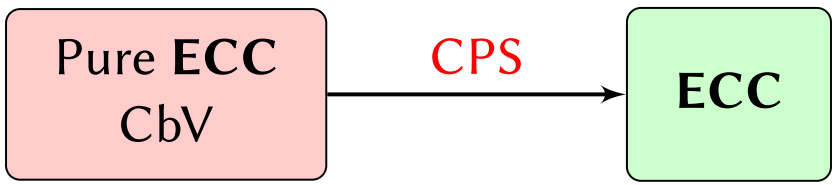
Every intuitionistic term eventually returns a value

(a.k.a. every term is thunkable)

Result

We obtain a CPS from CbV ECC to ECC.

Improve over previous results (M., Bowman et al.):



unrestricted, dependently typed, no extra assumptions!

Check `www.irif.fr/~emiquey/content/CPS_ECC.v`

Conclusion

Contributions:

1. $\mathbf{L_{dep}}$, dependent polarised sequent calculus
2. CPS translation to **ECC**
3. embedding of **ECC**+`call/cc`
4. unrestricted CPS in the intuitionistic case

Further work:

- relax restriction to thunkable terms
- by-name CPS translation
- second-class continuations (cf. Cong *et al.*'19)
- different effects (cf. Pédrot-Tabareau'19)

Conclusion

Contributions:

1. L_{dep}, dependent polarised sequent calculus
2. CPS translation to **ECC**
3. embedding of **ECC**+`call/cc`
4. unrestricted CPS in the intuitionistic case

Further work:

- relax restriction to thunkable terms
- by-name CPS translation
- second-class continuations (cf. Cong *et al.*'19)
- different effects (cf. Pédrot-Tabareau'19)

Conclusion

Contributions:

1. L_{dep}, dependent polarised sequent calculus
2. CPS translation to **ECC**
3. embedding of **ECC**+`call/cc`
4. unrestricted CPS in the intuitionistic case

Further work:

- relax restriction to thunkable terms
- by-name CPS translation
- second-class continuations (cf. Cong *et al.*'19)
- different effects (cf. Pédrot-Tabareau'19)

Conclusion

Contributions:

1. L_{dep}, dependent polarised sequent calculus
2. CPS translation to **ECC**
3. embedding of **ECC**+`call/cc`
4. unrestricted CPS in the intuitionistic case

Further work:

- relax restriction to thunkable terms
- by-name CPS translation
- second-class continuations (cf. Cong *et al.*'19)
- different effects (cf. Pédrot-Tabareau'19)

Conclusion

Contributions:

1. L_{dep}, dependent polarised sequent calculus
2. CPS translation to **ECC**
3. embedding of **ECC**+`call/cc`
4. unrestricted CPS in the intuitionistic case

Further work:

- relax restriction to thunkable terms
- by-name CPS translation
- second-class continuations (cf. Cong *et al.*'19)
- different effects (cf. Pédrot-Tabareau'19)

Thank you for you attention.