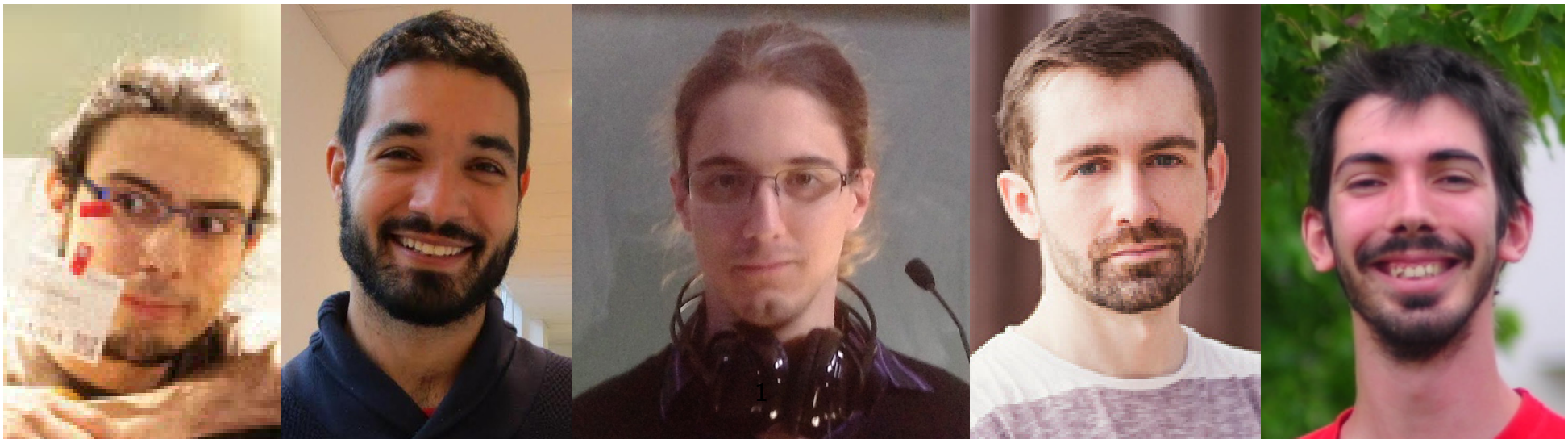
# A preview of a tutorial on polarised L calculi

Kenji Maillard, Étienne Miquey, Xavier Montillet,
Guillaume Munch-Maccagnoni, **Gabriel Scherer**

INRIA (Paris, Nantes, Saclay)

HOPE, September 23th 2018



# Why?

L: polarized versions of the $\lambda\mu\tilde{\mu}$ abstract-machine calculi.

Claim: a good **syntax** to talk about

- classical computation
- mixed-polarity equivalences
- effects
- purity

(Semantics: roughly in line with CBPV models.)

Related works:

- Zeilberger (direct relation to focusing; no syntax)
- CBPV (related models, worse syntax)

# Section 1

## $\mu\tilde{\mu}$ recap

# Abstract machines

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

# Abstract machines

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

$$c ::= \langle t \parallel e \rangle \quad \text{configuration}$$

$$\begin{array}{rll} t, u ::= & & \text{term} \\ & \mid x, y, z & \text{variable} \\ & \mid t\ u & \text{application} \\ & \mid \lambda x.\, t & \lambda\text{-abstraction} \end{array}$$

$$\begin{array}{rll} e, f ::= & & \text{context} \\ & \mid \star & \text{empty} \\ & \mid t \cdot e & \text{application stack} \end{array}$$

# Introducing $\mu$

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

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

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

# Introducing $\mu$

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

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

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

Idea: introduces a more primitive syntax

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

# Introducing $\mu$

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

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

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

Idea: introduces a more primitive syntax

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

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

# Introducing $\mu$

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

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

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

Idea: introduces a more primitive syntax

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

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

$$t\ u \quad \stackrel{\mathsf{def}}{:=} \quad \mu\alpha.\, \langle t \parallel u \cdot \alpha\rangle$$

# Detroducing $\lambda$

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

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

# Detroducing $\lambda$

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

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

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

# Detroducing $\lambda$

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

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

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

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

# Detroducing $\lambda$

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

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

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

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

$$\lambda x.\, t \stackrel{\mathsf{def}}{:=} \mu(x \cdot \alpha).\, \langle t \parallel \alpha \rangle$$

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

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

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

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

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

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

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

$$
\begin{array}{rl}
t, u ::= & \\
& \mid x, y, z \\
& \mid \mu\alpha.\, c \\
& \mid \mu(x \cdot \alpha).\, c \\
& \mid (t, u) \\
& \mid \sigma_i\ t
\end{array}
\qquad
\begin{array}{rl}
e, f ::= & \\
& \mid \star, \alpha, \beta \\
& \mid \tilde{\mu}x.\, c \\
& \mid t \cdot e \\
& \mid \tilde{\mu}(x_1, x_2).\, c \\
& \mid \tilde{\mu}[(\sigma_1\ x_1).\, c_1 \mid (\sigma_2\ x_2).\, c_2]
\end{array}
$$

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

$$
\begin{array}{rl}
t, u ::= & \\
& \mid x, y, z \\
& \mid \mu\alpha.\, c \\
& \mid \mu(x \cdot \alpha).\, c \\
& \mid (t, u) \\
& \mid \sigma_i\ t
\end{array}
\qquad
\begin{array}{rl}
e, f ::= & \\
& \mid \star, \alpha, \beta \\
& \mid \tilde{\mu}x.\, c \\
& \mid t \cdot e \\
& \mid \tilde{\mu}(x_1, x_2).\, c \\
& \mid \tilde{\mu}[(\sigma_1\ x_1).\, c_1 \mid (\sigma_2\ x_2).\, c_2] \\
& \mid \pi_i\ e
\end{array}
$$

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

# Other datatypes and $\tilde{\mu}$

$$c ::= \langle t \parallel e \rangle \quad \text{command}$$

$$
\begin{array}{rl}
t, u ::= & \\
& \mid x, y, z \\
& \mid \mu\alpha.\, c \\
& \mid \mu(x \cdot \alpha).\, c \\
& \mid (t, u) \\
& \mid \sigma_i\ t \\
& \mid \mu[(\pi_1\ x_1).\, c_1 \mid (\pi_2\ x_2).\, c_2]
\end{array}
\qquad
\begin{array}{rl}
e, f ::= & \\
& \mid \star, \alpha, \beta \\
& \mid \tilde{\mu}x.\, c \\
& \mid t \cdot e \\
& \mid \tilde{\mu}(x_1, x_2).\, c \\
& \mid \tilde{\mu}[(\sigma_1\ x_1).\, c_1 \mid (\sigma_2\ x_2).\, c_2] \\
& \mid \pi_i\ e
\end{array}
$$

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

# Upside down

$$\text{let } (x_1, x_2) = (\pi_1\ m)\ x\ y \text{ in } t$$

$$\left\langle m \,\middle\|\, \pi_1\ (x \cdot y \cdot \tilde{\mu}(x_1, x_2).\, \langle t \,\|\, \star\rangle)\right\rangle$$

If $t$ is $x_1$, how to instantiate free variables so that the result is 1?

# Program equivalences

The $\eta$-expansion rules are beautiful and regular.
(Thank you, sequent calculus.)

In the $\lambda$-calculus:

$$\begin{array}{rrcl} & (t : A \to B) & \rhd_\eta & \lambda x.\, t\ x \\ \forall C, & C[t : A \otimes B] & \rhd_\eta & \text{let } (x_1, x_2) = t \text{ in } C[(x_1, x_2)] \end{array}$$

In L:

$$\begin{array}{rcl} (t : A \to B) & \rhd_\eta & \mu(x \cdot \alpha).\, \langle t \parallel x \cdot \alpha \rangle \\ (e : A \otimes B) & \rhd_\eta & \tilde{\mu}(x_1, x_2).\, \langle (x_1, x_2) \parallel e \rangle \end{array}$$

# Program equivalences

The $\eta$-expansion rules are beautiful and regular.
(Thank you, sequent calculus.)

In the $\lambda$-calculus:

$$\begin{array}{rrcl} & (t : A \to B) & \rhd_\eta & \lambda x.\, t\ x \\ \forall C, & C[t : A \otimes B] & \rhd_\eta & \text{let } (x_1, x_2) = t \text{ in } C[(x_1, x_2)] \end{array}$$

In L:

$$\begin{array}{rcl} (t : A \to B) & \rhd_\eta & \mu(x \cdot \alpha).\, \langle t \parallel x \cdot \alpha \rangle \\ (e : A \otimes B) & \rhd_\eta & \tilde{\mu}(x_1, x_2).\, \langle (x_1, x_2) \parallel e \rangle \end{array}$$

Killer feature!

# Type system...

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

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

# Type system...

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

$c ::= \langle t \parallel e \rangle$ $\quad \boxed{c : \Gamma \vdash \Delta}$ $\quad \boxed{\Gamma \vdash t : A \mid \Delta}$ $\quad \boxed{\Gamma \mid e : A \vdash \Delta}$

# Type system...

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

$$
c ::= \langle t \parallel e \rangle \qquad \boxed{c : \Gamma \vdash \Delta} \qquad \boxed{\Gamma \vdash t : A \mid \Delta} \qquad \boxed{\Gamma \mid e : A \vdash \Delta}
$$

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

# Type system...

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

$c ::= \langle t \parallel e \rangle$ $\quad \boxed{c : \Gamma \vdash \Delta}$ $\quad \boxed{\Gamma \vdash t : A \mid \Delta}$ $\quad \boxed{\Gamma \mid e : A \vdash \Delta}$

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

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

# Type system...

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

$$c ::= \langle t \parallel e \rangle \qquad \boxed{c : \Gamma \vdash \Delta} \qquad \boxed{\Gamma \vdash t : A \mid \Delta} \qquad \boxed{\Gamma \mid e : A \vdash \Delta}$$

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

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

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

# Type system...

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

$c ::= \langle t \parallel e \rangle$ $\quad \boxed{c : \Gamma \vdash \Delta}$ $\quad \boxed{\Gamma \vdash t : A \mid \Delta}$ $\quad \boxed{\Gamma \mid e : A \vdash \Delta}$

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

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

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

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

# 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.)

$$\mathsf{call/cc}(f) \quad \stackrel{\mathsf{def}}{:=}$$

# 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.)

$$\mathsf{call/cc}(f) \quad \stackrel{\mathsf{def}}{:=} \quad \mu\alpha.\left\langle f \,\middle\|\, (\mu(x \cdot \beta).\langle x \,\|\, \alpha\rangle) \cdot \alpha \right\rangle$$

# 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.)

$$\mathsf{call/cc}(f) \quad \stackrel{\mathsf{def}}{:=} \quad \mu\alpha.\left\langle f \parallel (\mu(x \cdot \beta).\langle x \parallel \alpha\rangle) \cdot \alpha\right\rangle$$

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

$$t\ u \stackrel{\mathsf{def}}{:=} \mu\alpha.\langle t \parallel u \cdot \alpha\rangle \qquad \lambda x.\, t \stackrel{\mathsf{def}}{:=} \mu(x \cdot \alpha).\langle t \parallel \alpha\rangle$$

$$t\ u \stackrel{\mathsf{def}}{:=} \mu\star.\langle t \parallel u \cdot \star\rangle \qquad \lambda x.\, t \stackrel{\mathsf{def}}{:=} \mu(x \cdot \star).\langle t \parallel \star\rangle$$

# Confluence problem

Computational Classical logic: hard.

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle$$

# Confluence problem

Computational Classical logic: hard.

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle \quad \triangleright \quad c_2\,[\mu\alpha.\, c_1/x]$$

# Confluence problem

Computational Classical logic: hard.

$$c_1\,[\tilde{\mu}x.\,c_2/\alpha] \quad \triangleleft \quad \langle \mu\alpha.\,c_1 \parallel \tilde{\mu}x.\,c_2 \rangle \quad \triangleright \quad c_2\,[\mu\alpha.\,c_1/x]$$

# Confluence problem

Computational Classical logic: hard.

$$c_1\,[\tilde{\mu}x.\,c_2/\alpha] \quad \triangleleft \quad \langle \mu\alpha.\,c_1 \parallel \tilde{\mu}x.\,c_2 \rangle \quad \triangleright \quad c_2\,[\mu\alpha.\,c_1/x]$$

$$\langle x \parallel \alpha \rangle \quad \triangleleft \quad \left\langle \mu\_.\,\langle x \parallel \alpha \rangle \,\middle\|\, \tilde{\mu}\_.\,\langle y \parallel \beta \rangle \right\rangle \quad \triangleright \quad \langle y \parallel \beta \rangle$$

# Confluence problem

Computational Classical logic: hard.

$$c_1\,[\tilde{\mu}x.\,c_2/\alpha] \quad \triangleleft \quad \langle \mu\alpha.\,c_1 \parallel \tilde{\mu}x.\,c_2 \rangle \quad \triangleright \quad c_2\,[\mu\alpha.\,c_1/x]$$

$$\langle x \parallel \alpha \rangle \quad \triangleleft \quad \left\langle \mu\_.\,\langle x \parallel \alpha \rangle \,\middle\|\, \tilde{\mu}\_.\,\langle y \parallel \beta \rangle \right\rangle \quad \triangleright \quad \langle y \parallel \beta \rangle$$

Arbitrary resolution of the critical pair:

- prefer reducing the term: call-by-value
- prefer reducing the co-term: call-by-name

# Confluence problem: values and co-values

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle$$

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

# Confluence problem: values and co-values

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle$$

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

Prefer reducing the term: $\langle V \parallel \tilde{\mu}x.\, c \rangle \rhd_{\tilde{\mu}} c\,[V/x]$

# Confluence problem: values and co-values

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle$$

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

Prefer reducing the term: $\langle V \parallel \tilde{\mu}x.\, c \rangle \rhd_{\tilde{\mu}} c\,[V/x]$

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

# Confluence problem: values and co-values

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle$$

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

Prefer reducing the term: $\langle V \parallel \tilde{\mu}x.\, c \rangle \rhd_{\tilde{\mu}} c\,[V/x]$

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

Prefer reducing the co-term: $\langle \mu\alpha.\, c \parallel S \rangle \rhd_{\tilde{\mu}} c\,[S/\alpha]$

# Confluence problem: values and co-values

$$\langle \mu\alpha.\, c_1 \parallel \tilde{\mu}x.\, c_2 \rangle$$

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

Prefer reducing the term: $\langle V \parallel \tilde{\mu}x.\, c \rangle \rhd_{\tilde{\mu}} c\,[V/x]$

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

Prefer reducing the co-term: $\langle \mu\alpha.\, c \parallel S \rangle \rhd_{\tilde{\mu}} c\,[S/\alpha]$

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

# Pointers

The duality of computation (Curien and Herbelin, 2000) introduced the syntax (with $\lambda$)

Tutorial on computational classical logic and the sequent calculus (Downen and Ariola, 2018) (in JFP)

Examples of PL research it inspired:

- Copatterns: Programming Infinite Structures by Observations
  Abel, Pientka, Thibodeau, and Setzer (2013)
- Structures for Structural Recursion
  Downen, Johnson-Freyd, and Ariola (2015)
- A classical sequent calculus with dependent types
  Miquey (2017)
- Deciding equivalence with sums and the empty type
  Scherer (2017)

# Section 2

# A bit of polarity

Call-by-value,name restrictions: **global** confluence solutions.

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

Call-by-value,name restrictions: **global** confluence solutions.

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

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

$$\begin{array}{rcllllll} P, Q & ::= & X^+, Y^+ & \mid P \otimes Q & \mid P \oplus Q & \mid \ldots & \mid \langle N \rangle^+ \\ M, N & ::= & X^-, Y^- & \mid P \to N & \mid M \times N & \mid \ldots & \mid \langle P \rangle^- \end{array}$$

Call-by-value,name restrictions: **global** confluence solutions.

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

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

$$\begin{array}{rcllllll} P, Q & ::= & X^+, Y^+ & \mid P \otimes Q & \mid P \oplus Q & \mid \ldots & \mid \langle N \rangle^+ \\ M, N & ::= & X^-, Y^- & \mid P \to N & \mid M \times N & \mid \ldots & \mid \langle P \rangle^- \end{array}$$

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

Call-by-value,name restrictions: **global** confluence solutions.

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

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

$$\begin{array}{rcllllll} P, Q & ::= & X^+, Y^+ & \mid P \otimes Q & \mid P \oplus Q & \mid \ldots & \mid \langle N \rangle^+ \\ M, N & ::= & X^-, Y^- & \mid P \to N & \mid M \times N & \mid \ldots & \mid \langle P \rangle^- \end{array}$$

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

$$\begin{array}{rcll} c & ::= & \langle V \parallel e \rangle^- & \mid \langle t \parallel S \rangle^+ \end{array}$$

Call-by-value,name restrictions: **global** confluence solutions.

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

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

$$\begin{array}{rcllllll} P, Q & ::= & X^+, Y^+ & \mid P \otimes Q & \mid P \oplus Q & \mid \ldots & \mid \langle N \rangle^+ \\ M, N & ::= & X^-, Y^- & \mid P \to N & \mid M \times N & \mid \ldots & \mid \langle P \rangle^- \end{array}$$

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

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

Call-by-value,name restrictions: **global** confluence solutions.

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

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

$$\begin{array}{rcllllll} P, Q & ::= & X^+, Y^+ & \mid P \otimes Q & \mid P \oplus Q & \mid \ldots & \mid \langle N \rangle^+ \\ M, N & ::= & X^-, Y^- & \mid P \to N & \mid M \times N & \mid \ldots & \mid \langle P \rangle^- \end{array}$$

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

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

# Polarized reduction

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

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

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

# Strong values

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

# Strong values

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

# 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.

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

- Add an automatic reduction with arbitrary order: Wadler's $\zeta$-rules.

$$\langle (t, u) \parallel S \rangle^+ \rhd^{t \notin V}_{\zeta.\mathsf{pair}.1} \left\langle t \,\middle\|\, \tilde{\mu}x.\, \langle (x, u) \parallel S \rangle^+ \right\rangle$$

# Polarity: example

Recall: $t\ u \stackrel{\text{def}}{:=} \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

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

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

# Polarity: example

Recall: $t\ u \stackrel{\text{def}}{:=} \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

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

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

# Polarity: example

Recall: $t\ u \stackrel{\text{def}}{:=} \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

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

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

# Polarity: example

Recall: $t\ u \stackrel{\text{def}}{:=} \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

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

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

If $u$ has a negative type, it is a value: $\rhd^{*}\ t\,[u/x]$
If $u$ has a positive type and is not a value, it gets reduced first.

# Polarity: example

Recall: $t\ u \stackrel{\text{def}}{:=} \mu\alpha.\langle t \parallel u \cdot \alpha\rangle$

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

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

If $u$ has a negative type, it is a value: $\rhd^{*}\ t\,[u/x]$
If $u$ has a positive type and is not a value, it gets reduced first.

$P \to N$: call-by-value function
$\langle M \rangle^{+} \to N$: call-by-name function

# 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).

# 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:

# 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.

# 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!

# 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).

# Section 3

# Effect examples

(Inspired by Paul's nice CBPV examples)

# How to add a primitive effect to L?

Head reduction can be seen as a function

$$(\triangleright) : \mathsf{Com} \to \mathsf{Com}$$

written as a relation: $c_1 \triangleright c_2$ for $(\triangleright\, c_1) = c_2$.
(normal commands reduce into themselves)

# How to add a primitive effect to L?

Head reduction can be seen as a function

$$(\triangleright) : \mathsf{Com} \to \mathsf{Com}$$

written as a relation: $c_1 \triangleright c_2$ for $(\triangleright\, c_1) = c_2$.
(normal commands reduce into themselves)

Now pick an effect $T$ as a monad on Set, and extend

$$(\triangleright_T) : \mathsf{Com} \to T(\mathsf{Com})$$

# Printing

Output literals $o \in \mathsf{O}$.

$$T(X) \stackrel{\text{def}}{:=} \mathsf{O}^* \times X$$

# Printing

Output literals $o \in \mathsf{O}$.

$$T(X) \stackrel{\text{def}}{:=} \mathsf{O}^* \times X$$

$$c ::= \cdots \mid \mathsf{print}(o).\, c$$

# Printing

Output literals $o \in \mathsf{O}$.

$$T(X) \stackrel{\text{def}}{:=} \mathsf{O}^* \times X$$

$$c ::= \cdots \mid \mathsf{print}(o).\, c$$

$$\mathsf{print}(o).\, c \rhd_T ([o], c)$$

# Printing

Output literals $o \in \mathsf{O}$.

$$T(X) \stackrel{\text{def}}{:=} \mathsf{O}^* \times X$$

$$c ::= \cdots \mid \mathsf{print}(o).\, c$$

$$\mathsf{print}(o).\, c \rhd_T ([o], c)$$

Repeated reductions, Kleisli composition:

$$\mathsf{print}(1).\, \mathsf{print}(2).\, c \quad \rhd_T \quad ([1], \mathsf{print}(2).\, c) \quad \rhd_T \quad ([1, 2], c)$$

# Global state

Fixed state type $S$.

$$T(X) \stackrel{\text{def}}{:=} S \to (S \times X)$$

# Global state

Fixed state type $S$.

$$T(X) \stackrel{\text{def}}{:=} S \to (S \times X)$$

$$c ::= \cdots \mid \mathsf{get}(x).\, c \mid \mathsf{set}(t).\, c$$

# Global state

Fixed state type $S$.

$$T(X) \stackrel{\text{def}}{:=} S \to (S \times X)$$

$$c ::= \cdots \mid \mathsf{get}(x).\, c \mid \mathsf{set}(t).\, c$$

$$\begin{array}{lll} \mathsf{get}(x).\, c & \rhd_T & (V \mapsto (V, c\,[V/x])) \\ \mathsf{set}(V).\, c & \rhd_T & (\_ \mapsto (V, c)) \end{array}$$

# Partiality

$$T(X) \stackrel{\text{def}}{:=} 1 + X$$

$$c ::= \cdots \mid \mathsf{fail}().$$

$$\mathsf{fail}(). \rhd_T \mathsf{left}()$$

# Effects and polarity: as it should

"(print 1; (x, y))":

$$\mu^{+}\alpha.\,\mathsf{print}(1).\,\langle (x, y) \parallel \alpha \rangle^{+}$$

# Effects and polarity: as it should

"(print 1; (x, y))":

$$\mu^{+}\alpha.\,\mathsf{print}(1).\,\langle (x, y) \parallel \alpha \rangle^{+}$$

Not a value, will perform the effect first.

# Effects and polarity: as it should

"(print 1; (x, y))":

$$\mu^{+}\alpha.\,\mathsf{print}(1).\,\langle (x, y) \parallel \alpha \rangle^{+}$$

Not a value, will perform the effect first.

"(print 1; $\lambda x.\, x$)":

$$\mu^{-}\alpha.\,\mathsf{print}(1).\,\langle \lambda x.\, x \parallel \alpha \rangle^{-}$$

Value, will block effects until the function is forced.

# Effects and polarity: as it should

"(print 1; (x, y))":

$$\mu^{+}\alpha.\,\mathsf{print}(1).\,\langle (x, y) \parallel \alpha\rangle^{+}$$

Not a value, will perform the effect first.

"(print 1; $\lambda x.\, x$)":

$$\mu^{-}\alpha.\,\mathsf{print}(1).\,\langle \lambda x.\, x \parallel \alpha\rangle^{-}$$

Value, will block effects until the function is forced.

$$\begin{array}{rcl} (t : A \to B) & \rhd_{\eta} & \mu(x \cdot \alpha).\,\langle t \parallel x \cdot \alpha\rangle \\ (e : A \otimes B) & \rhd_{\eta} & \tilde{\mu}(x_1, x_2).\,\langle (x_1, x_2) \parallel e\rangle \end{array}$$

# Effects and polarity: as it should

"(print 1; (x, y))":

$$\mu^{+}\alpha.\,\mathsf{print}(1).\,\langle (x, y) \parallel \alpha\rangle^{+}$$

Not a value, will perform the effect first.

"(print 1; $\lambda x.\, x$)":

$$\mu^{-}\alpha.\,\mathsf{print}(1).\,\langle \lambda x.\, x \parallel \alpha\rangle^{-}$$

Value, will block effects until the function is forced.

$$\begin{array}{rcl} (t : A \to B) & \rhd_{\eta} & \mu(x \cdot \alpha).\,\langle t \parallel x \cdot \alpha\rangle \\ (e : A \otimes B) & \rhd_{\eta} & \tilde{\mu}(x_1, x_2).\,\langle (x_1, x_2) \parallel e\rangle \end{array}$$

We already had the right $\eta$-laws for effects.

# Section 4

# The symmetry of purity

# Linear and thunkable

Symmetric notions of purity from **substitution properties**.

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable.

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.
Closed by anti-reduction: $\mu^{+}\alpha.\, \langle V \parallel \alpha \rangle$ is thunkable.

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.
Closed by anti-reduction: $\mu^{+}\alpha.\, \langle V \parallel \alpha \rangle$ is thunkable.

Not thunkable: $t \stackrel{\text{def}}{:=} \mu^{+}\alpha.\, \mathsf{print}(1).\, \langle x \parallel \alpha \rangle$.

$$c\,[t/x] \not\simeq \langle t \parallel \tilde{\mu} x.\, c \rangle$$

For example:

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.
Closed by anti-reduction: $\mu^{+}\alpha.\, \langle V \parallel \alpha \rangle$ is thunkable.

Not thunkable: $t \stackrel{\text{def}}{:=} \mu^{+}\alpha.\, \mathsf{print}(1).\, \langle x \parallel \alpha \rangle$.

$$c\,[t/x] \not\simeq \langle t \parallel \tilde{\mu} x.\, c \rangle$$

For example:

$\mathsf{print}(2).\, c'$ (non-commutative)

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.
Closed by anti-reduction: $\mu^{+}\alpha.\, \langle V \parallel \alpha \rangle$ is thunkable.

Not thunkable: $t \stackrel{\text{def}}{:=} \mu^{+}\alpha.\, \mathsf{print}(1).\, \langle x \parallel \alpha \rangle$.

$$c\,[t/x] \not\simeq \langle t \parallel \tilde{\mu} x.\, c \rangle$$

For example:

$$\left\langle x \,\middle\|\, \tilde{\mu} x_1.\, \langle x \parallel \tilde{\mu} x_2.\, \rangle c' \right\rangle \qquad \text{(duplicating)}$$

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.
Closed by anti-reduction: $\mu^{+}\alpha.\, \langle V \parallel \alpha \rangle$ is thunkable.

Not thunkable: $t \stackrel{\text{def}}{:=} \mu^{+}\alpha.\, \mathsf{print}(1).\, \langle x \parallel \alpha \rangle$.

$$c\,[t/x] \not\simeq \langle t \parallel \tilde{\mu} x.\, c \rangle$$

For example:

$$\langle y \parallel \star \rangle \qquad \text{(erasing)}$$

# Thunkable terms

$t$ **thunkable** ("$t$ is pure") $\stackrel{\text{def}}{:=}$

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

Remark: all $V$ are thunkable. Semantic notion of value.
Closed by anti-reduction: $\mu^{+}\alpha.\, \langle V \parallel \alpha \rangle$ is thunkable.

Not thunkable: $t \stackrel{\text{def}}{:=} \mu^{+}\alpha.\, \mathsf{print}(1).\, \langle x \parallel \alpha \rangle$.

$$c\,[t/x] \not\simeq \langle t \parallel \tilde{\mu} x.\, c \rangle$$

For example:

$$\langle y \parallel \star \rangle \qquad \text{(erasing)}$$

Thunkable: observationally pure commands

$$\mu\alpha.\, \mathsf{flip}(x).\, \left\langle x \,\middle\|\, \tilde{\mu}[(\sigma_1\ \_).\, \langle V \parallel \star \rangle \mid (\sigma_2\ \_).\, \langle V \parallel \star \rangle] \right\rangle$$

# Linear co-terms

$e$ **linear** ("$e$ is strict") $\stackrel{\text{def}}{:=}$

$$\forall c[\alpha], \quad c\,[e/\alpha] \simeq \langle \mu\alpha.\, c \parallel e \rangle$$

All $S$ are linear. Semantic notion of co-value.

Not linear: $e \stackrel{\text{def}}{:=} \tilde{\mu}^{-} x.\, \langle y \parallel \alpha \rangle$

$$c\,[e/\alpha] \not\simeq \langle \mu\alpha.\, c \parallel e \rangle$$

(take $c[\alpha] \stackrel{\text{def}}{:=} \mathsf{fail}().$ )

# Purity and polarity

Consider a one-one command

$$c[y, \beta] : (y : A) \vdash \beta : B$$

$c[y, \beta]$ thunkable $\stackrel{\text{def}}{:=}$ $\mu\beta.\, c[y, \beta]$ thunkable.

$c[y, \beta]$ linear $\stackrel{\text{def}}{:=}$ $\tilde{\mu} y.\, c[y, \beta]$ linear.

# Purity and polarity

Consider a one-one command

$$c[y, \beta] : (y : A) \vdash \beta : B$$

$$c[y,\beta] \text{ thunkable} \quad \stackrel{\text{def}}{:=} \quad \mu\beta.\, c[y,\beta] \text{ thunkable.}$$

$$c[y,\beta] \text{ linear} \quad \stackrel{\text{def}}{:=} \quad \tilde{\mu} y.\, c[y,\beta] \text{ linear.}$$

$$\mu\beta.\, c[y,\beta] \quad \text{thunkable}$$

$$\stackrel{\text{def}}{:=}$$

$$\forall c'[x], \qquad c'\,[\mu\beta.\, c[y,\beta]/x] \quad \simeq \quad \langle \mu\beta.\, c[y,\beta] \parallel \tilde{\mu} x.\, c'[x] \rangle$$

# Purity and polarity

Consider a one-one command

$$c[y, \beta] : (y : A) \vdash \beta : B$$

$$c[y,\beta] \text{ thunkable} \quad \stackrel{\text{def}}{:=} \quad \mu\beta.\, c[y,\beta] \text{ thunkable.}$$

$$c[y,\beta] \text{ linear} \quad \stackrel{\text{def}}{:=} \quad \tilde{\mu} y.\, c[y,\beta] \text{ linear.}$$

$$\mu\beta.\, c[y,\beta] \quad \text{thunkable}$$

$$\stackrel{\text{def}}{:=}$$

$$\forall c'[x], \qquad c'\,[\mu\beta.\, c[y,\beta]/x] \quad \simeq \quad \langle \mu\beta.\, c[y,\beta] \parallel \tilde{\mu} x.\, c'[x] \rangle$$

All $(y : A) \vdash \beta : N$ are thunkable.

# Purity and polarity

Consider a one-one command

$$c[y, \beta] : (y : A) \vdash \beta : B$$

$$c[y,\beta] \text{ thunkable} \stackrel{\text{def}}{:=} \mu\beta.\, c[y,\beta] \text{ thunkable}.$$
$$c[y,\beta] \text{ linear} \stackrel{\text{def}}{:=} \tilde{\mu} y.\, c[y,\beta] \text{ linear}.$$

$$\mu\beta.\, c[y,\beta] \quad \text{thunkable}$$

$$\stackrel{\text{def}}{:=}$$

$$\forall c'[x], \qquad c'\,[\mu\beta.\, c[y,\beta]/x] \quad \simeq \quad \langle \mu\beta.\, c[y,\beta] \parallel \tilde{\mu} x.\, c'[x] \rangle$$

All $(y : A) \vdash \beta : N$ are thunkable.
All $(y : P) \vdash \beta : B$ are linear.

# Section 5

# A whiff of categories

One-one commands $P \vdash Q$ form a category.
One-one commands $M \vdash N$ form a category.

General one-one commands $A \vdash B$ do **not** form a category:
composition may be non-associative: $A \stackrel{f}{\to} P \stackrel{g}{\to} N \stackrel{h}{\to} B$

# Adjunction(s)

Two adjunctions?

- $\langle \_ \rangle^- \dashv \langle \_ \rangle^+$
- $\langle \_ \rangle^+ \dashv \langle \_ \rangle^-$

# Adjunction(s)

Two adjunctions?

- $\langle \_ \rangle^- \dashv \langle \_ \rangle^+$
- $\langle \_ \rangle^+ \dashv \langle \_ \rangle^-$

$$\frac{\langle P \rangle^- \overset{\mathsf{lin}}{\to} N}{\dfrac{P \overset{\mathsf{lin+thunk}}{\to} N}{P \overset{\mathsf{thunk}}{\to} \langle N \rangle^-}} \qquad \frac{\langle N \rangle^+ \to P}{\dfrac{N \to P}{N \to \langle P \rangle^-}}$$

The first is between pure commands (both linear and thunkable).
Minimal effects.

# Adjunction(s)

Two adjunctions?

- $\langle \_ \rangle^- \dashv \langle \_ \rangle^+$
- $\langle \_ \rangle^+ \dashv \langle \_ \rangle^-$

$$\frac{\langle P \rangle^- \xrightarrow{\mathsf{lin}} N}{\dfrac{P \xrightarrow{\mathsf{lin+thunk}} N}{P \xrightarrow{\mathsf{thunk}} \langle N \rangle^-}} \qquad \frac{\langle N \rangle^+ \to P}{\dfrac{N \to P}{N \to \langle P \rangle^-}}$$

The first is between pure commands (both linear and thunkable).
Minimal effects.

The second is between all commands.
Polarised effects.

# Conclusion

L: polarized versions of the $\lambda\mu\tilde{\mu}$ abstract-machine calculi.

Future pointer: our upcoming tutorial!

Claim: a good **syntax** to talk about

- classical computation
- mixed-polarity equivalences
- effects
- purity

Thanks!

Andreas Abel, Brigitte Pientka, David Thibodeau, and Anton Setzer. Copatterns: Programming infinite structures by observations. In **POPL**, 2013. URL http://www.cse.chalmers.se/~abela/popl13.pdf.

Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In **ICFP**, 2000. URL http://pauillac.inria.fr/~herbelin/publis/icfp-CuHer00-duality+errata.pdf.

Paul Downen and Zena Ariola. Tutorial on computational classical logic and the sequent calculus. In **JFP**, 2018. URL http://ix.cs.uoregon.edu/~pdownen/publications/sequent-intro.pdf.

Paul Downen, Philip Johnson-Freyd, and Zena M. Ariola. Structures for structural recursion. In **ICFP**, 2015. URL http://ix.cs.uoregon.edu/~pdownen/publications/structure-recursion-extended.pdf.

Étienne Miquey. A classical sequent calculus with dependent types. In **ESOP**, 2017.

Gabriel Scherer. Deciding equivalence with sums and the empty type. In **POPL**, 2017.