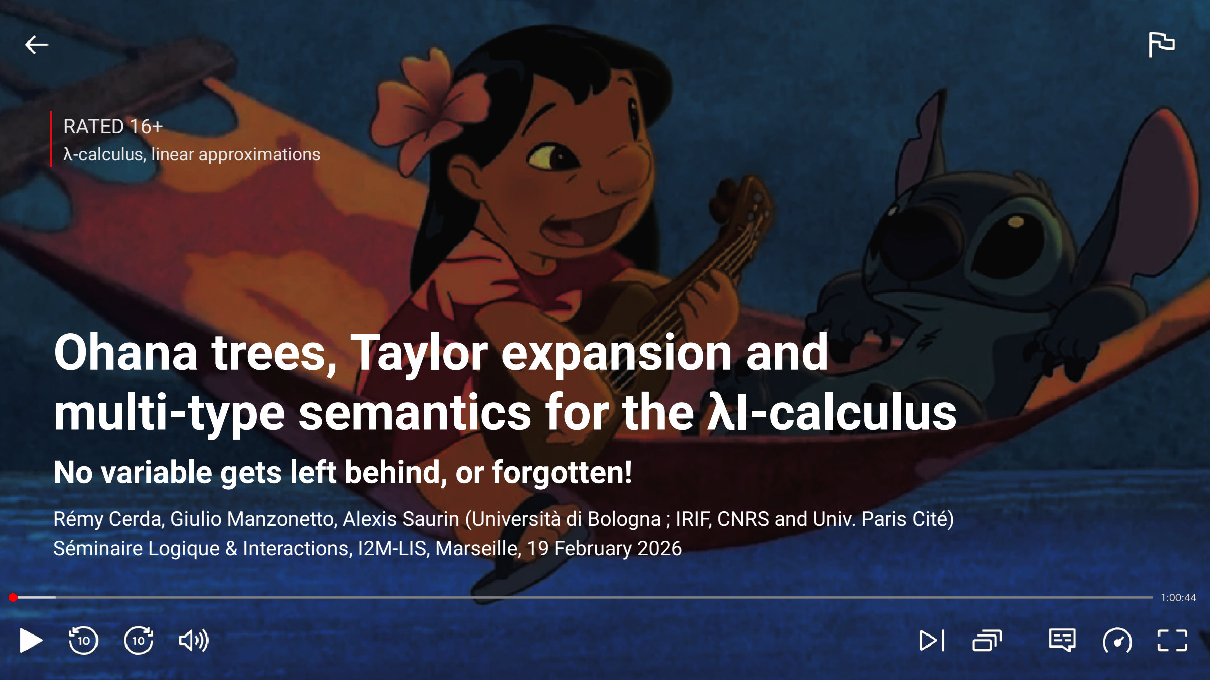


RATED 16+
λ-calculus, linear approximations

Ohana trees, Taylor expansion and multi-type semantics for the λI-calculus

No variable gets left behind, or forgotten!

Rémy Cerda, Giulio Manzonetto, Alexis Saurin (Università di Bologna ; IRIF, CNRS and Univ. Paris Cité)

Séminaire Logique & Interactions, I2M-LIS, Marseille, 19 February 2026

1:00:44

AU MENU AUJOURD'HUI...

λI-theories and Ohana trees

Continuous and linear approximation for Ohana evaluation

A multi-type semantics characterising the theory $\mathcal{O}$ of Ohana trees

Questions you may ask

λI-THEORIES AND OHANA TREES

FROM λ-THEORIES...

λ

A model of the λ-calculus ⇝ A λ-theory

$[\![-]\!]$

$M =_{\mathcal{T}} N$ iff $[\![M]\!] = [\![N]\!]$

FROM λ-THEORIES…

λ

A model of the λ-calculus $\rightsquigarrow$ A λ-theory

$[\![-]\!]$ $\qquad M =_{\mathcal{T}} N$ iff $[\![M]\!] = [\![N]\!]$

A **λ-theory** $\mathcal{T}$ is a set of equalities between λ-terms such that

$$\underbrace{\frac{M =_\beta N}{M =_{\mathcal{T}} N}}_{\text{it contains } \beta\text{-conversion}} \qquad \underbrace{\frac{P =_{\mathcal{T}} P'}{\lambda x.P =_{\mathcal{T}} \lambda x.P'} \qquad \frac{P =_{\mathcal{T}} P' \qquad Q =_{\mathcal{T}} Q'}{PQ =_{\mathcal{T}} P'Q'}}_{\text{it is stable under contexts}}$$

FROM λ-THEORIES…

λ

A model of the λ-calculus $\rightsquigarrow$ A λ-theory $\leftsquigarrow$ A notion of evaluation tree

$M =_{\mathcal{B}} N$ iff $\mathrm{BT}(M) = \mathrm{BT}(N)$

$\mathrm{BT}(-)$

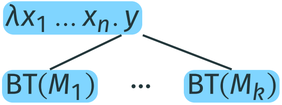
FROM λ-THEORIES...

λ

A model of the λ-calculus ⇝ A λ-theory ⇜ A notion of evaluation tree

$M =_{\mathcal{B}} N$ iff $\mathrm{BT}(M) = \mathrm{BT}(N)$ $\mathrm{BT}(-)$

The **Böhm tree** of a λ-term M is defined **coinductively** by



$$\mathrm{BT}(M) := \begin{cases} \lambda x_1 \dots x_n . y \\ \mathrm{BT}(M_1) \quad \cdots \quad \mathrm{BT}(M_k) & \text{if } M \twoheadrightarrow_h \lambda x_1 \dots x_n . y M_1 \cdots M_k, \\ \perp & \text{otherwise.} \end{cases}$$

For example: $\mathrm{BT}(\mathrm{I}) := \mathrm{I}$ $\mathrm{BT}(\Omega) := \perp$ $\mathrm{BT}(\mathrm{Y}) := \lambda f . f(f(f(\dots)))$

From λ-theories… to λI-theories

λ A model of the λ-calculus $\rightsquigarrow$ A λ-theory $\leftsquigarrow$ A notion of evaluation tree

λI

FROM λ-THEORIES... TO λI-THEORIES

λ A model of the λ-calculus $\rightsquigarrow$ A λ-theory $\leftsquigarrow$ A notion of evaluation tree

λI

The **λI-calculus** is the fragment of the λ-calculus **without erasure**. Formally, $\Lambda_\mathsf{I} \coloneqq \bigcup_{X \subseteq_\mathsf{f} \mathcal{V}} \Lambda_\mathsf{I}(X)$, where $\Lambda_\mathsf{I}(X)$ is the set of λI-terms with free variables in X:

$$\frac{}{x \in \Lambda_\mathsf{I}(\{x\})} \qquad \frac{M \in \Lambda_\mathsf{I}(X) \quad x \in X}{\lambda x.M \in \Lambda_\mathsf{I}(X \setminus \{x\})} \qquad \frac{M \in \Lambda_\mathsf{I}(X) \quad N \in \Lambda_\mathsf{I}(Y)}{MN \in \Lambda_\mathsf{I}(X \cup Y)}$$

For example: $\lambda x.x \in \Lambda_\mathsf{I}(\emptyset)$ $\quad \lambda x.xy \in \Lambda_\mathsf{I}(\{y\})$ $\quad \lambda x.y \notin \Lambda_\mathsf{I}$

FROM λ-THEORIES… TO λI-THEORIES

λ	A model of the λ-calculus	⇝	A λ-theory	⇜	A notion of evaluation tree
λI			**???**		

FROM λ-THEORIES… TO λI-THEORIES

λ A model of the λ-calculus ⇝ A λ-theory ⇜ A notion of evaluation tree

λI ???

A **λI-theory** $\mathcal{T}$ is a set of equalities between λ-terms such that

$$\underbrace{\frac{M =_{\beta} N}{M =_{\mathcal{T}} N}}_{\text{it contains }\beta\text{-conversion}} \qquad \underbrace{\frac{P =_{\mathcal{T}} P' \qquad x \in \mathrm{fv}(P) \cap \mathrm{fv}(P')}{\lambda x.P =_{\mathcal{T}} \lambda x.P'} \qquad \frac{P =_{\mathcal{T}} P' \qquad Q =_{\mathcal{T}} Q'}{PQ =_{\mathcal{T}} P'Q'}}_{\text{it is stable under }\lambda I\text{-contexts}}$$

- Every λ-theory restricted to Λ_I is a λI-theory.
- The converse is false, *e.g.* the λI-theory generated by equating all λI-terms without β-nf.

FROM λ-THEORIES... TO λI-THEORIES

λ	A model of the λ-calculus	⇝	A λ-theory	⇜	A notion of evaluation tree
λI			A λI-theory	⇜	**???**

FROM λ-THEORIES… TO λI-THEORIES

λ	A model of the λ-calculus	⇝	A λ-theory	⇜	A notion of evaluation tree
λI			A λI-theory	⇜	**???**

Böhm trees still generate a λI-theory $\mathcal{B}$… but behave poorly wrt. Λ_{I}:

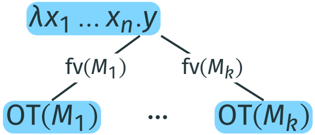
$$M \text{ is a } \lambda\text{I-term} \quad \not\Rightarrow \quad \mathrm{BT}(M) \text{ is an “infinitary } \lambda\text{I-term”}$$

Indeed, abstracted variables may be:

- **left behind** an unsolvable subterm: $\mathrm{BT}(\lambda xy.x(\Omega y)) = \lambda xy.x\bot$.
- **forgotten** along infinite computations:
  if $Mxf \twoheadrightarrow_\beta f(Mxf)$, then $\mathrm{BT}(M) = \lambda xf.f(f(f(\dots)))$.

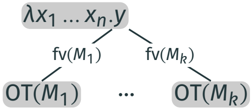
INTRODUCING OHANA TREES

The **Ohana tree** of a λI-term M is defined coinductively by



$$\mathrm{OT}(M) := \begin{cases} \begin{array}{c} \lambda x_1 \dots x_n.y \\ \mathrm{fv}(M_1) \qquad \mathrm{fv}(M_k) \\ \mathrm{OT}(M_1) \quad \cdots \quad \mathrm{OT}(M_k) \end{array} & \text{if } M \twoheadrightarrow_h \lambda x_1 \dots x_n.yM_1 \cdots M_k, \\ \bot_{\mathrm{fv}(M)} & \text{otherwise.} \end{cases}$$

NO VARIABLE LEFT BEHIND!



$\mathrm{OT}(M) :=$

$\lambda x_1 \dots x_n.y$

$\mathrm{fv}(M_1)$ $\mathrm{fv}(M_k)$

$\mathrm{OT}(M_1)$ $\cdots$ $\mathrm{OT}(M_k)$

if $M \twoheadrightarrow_h \lambda x_1 \dots x_n.yM_1 \cdots M_k$,

$\perp_{\mathrm{fv}(M)}$ otherwise.

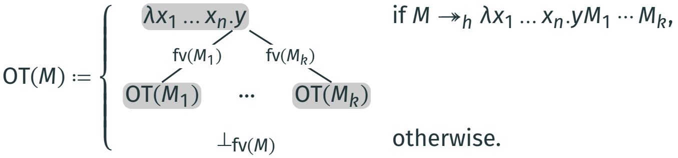
Whereas Böhm trees equate $\lambda x.x(\Omega y)(\Omega z)$ and $\lambda x.x(\Omega z)(\Omega y)$:

$$\mathrm{BT}(\lambda x.x(\Omega x)(\Omega y)) \;=\; \lambda x.x\perp\perp \;=\; \mathrm{BT}(\lambda x.x(\Omega y)(\Omega z))$$

Ohana trees do separate them:

$$\mathrm{OT}(\lambda x.x(\Omega x)(\Omega y)) \;=\; \lambda x.x\perp_{\{x\}}\perp_{\{y\}} \;\neq\; \lambda x.x\perp_{\{y\}}\perp_{\{z\}} \;=\; \mathrm{BT}(\lambda x.x(\Omega y)(\Omega z))$$

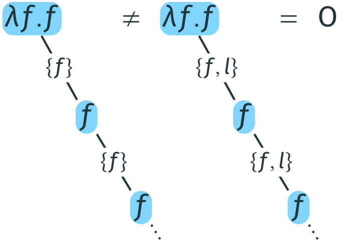
NO VARIABLE FORGOTTEN!



Klop's **Bible fixed-point combinator** is 📖$_l$ $:= \lambda e.\mathrm{BYBel} =_\beta \lambda e.e(\mathrm{BYBel})$.

Whereas Böhm trees equate Y and 📖$_l$: $\mathrm{BT}(\mathrm{Y}) = \lambda f.f(f(\ldots)) = \mathrm{BT}($📖$_l)$,

Ohana trees do separate them: $\mathrm{OT}(\mathrm{Y}) =$ [tree] $\neq$ [tree] $= \mathrm{OT}($📖$_l)$.



OUR FIRST GOAL

The equational theory $\mathcal{O}$ is defined by saying that for all $M, N \in \Lambda_{\mathsf{I}}$, $M =_{\mathcal{O}} N$ whenever $\mathrm{OT}(M) = \mathrm{OT}(N)$.

We want to prove the following

Theorem

$\mathcal{O}$ is a λI-theory.

CONTINUOUS AND LINEAR APPROXIMATION FOR OHANA EVALUATION

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constant $\perp$ to the syntax.
  $\perp$ is “an undefined term”.

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_{\mathsf{f}} \mathcal{V}$) to the syntax.
  $\bot_X$ is “an undefined term whose set of free variables is X”.

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_f \mathcal{V}$) to the syntax.
  $\bot_X$ is "an undefined term whose set of free variables is X".
- Define a **(head) approximation ordering** by

$$\frac{}{\bot \sqsubseteq M} \qquad \frac{\forall i, \quad M_i \sqsubseteq N_i}{\lambda \vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda \vec{x}.yN_1 \cdots N_k}$$

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_f \mathcal{V}$) to the syntax.
  $\bot_X$ is "an undefined term whose set of free variables is X".
- Define a **(head) approximation ordering** by

$$\frac{\mathrm{fv}(M) = X}{\bot_X \sqsubseteq M} \qquad \frac{\forall i, \quad M_i \sqsubseteq N_i \quad \mathrm{fv}(M_i) = \mathrm{fv}(N_i)}{\lambda \vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda \vec{x}.yN_1 \cdots N_k}$$

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_f \mathcal{V}$) to the syntax.
  $\bot_X$ is "an undefined term whose set of free variables is X".
- Define a **(head) approximation ordering** by

$$\frac{\forall i, \qquad M_i \sqsubseteq N_i \qquad \mathrm{fv}(M_i) = \mathrm{fv}(N_i)}{\lambda\vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda\vec{x}.yN_1 \cdots N_k}$$

- Define **(head) approximants**:

$$\mathcal{A} \quad \ni \quad A, B, \ldots \quad := \quad \bot \quad | \quad \lambda x_1 \ldots x_n.yA_1 \cdots A_k$$

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_f \mathcal{V}$) to the syntax.
  $\bot_X$ is “an undefined term whose set of free variables is X”.
- Define a **(head) approximation ordering** by

$$\frac{\forall i, \qquad M_i \sqsubseteq N_i \qquad \mathrm{fv}(M_i) = \mathrm{fv}(N_i)}{\lambda\vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda\vec{x}.yN_1 \cdots N_k}$$

- Define **(head) approximants**:

$$\mathcal{A}_{\mathrm{m}}(X) \quad \ni \quad A, B, \ldots \quad := \quad \bot_X \quad | \quad \lambda x_1 \ldots x_n.yA_1 \cdots A_k \quad \text{s.t. } (\ldots)$$

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_{\mathrm{f}} \mathcal{V}$) to the syntax.
  $\bot_X$ is "an undefined term whose set of free variables is X".
- Define a **(head) approximation ordering** by

$$\frac{\forall i, \qquad M_i \sqsubseteq N_i \qquad \mathrm{fv}(M_i) = \mathrm{fv}(N_i)}{\lambda \vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda \vec{x}.yN_1 \cdots N_k}$$

- Define **(head) approximants**:

$$\mathcal{A}_{\mathrm{m}}(X) \quad \ni \quad A, B, \ldots \quad := \quad \bot_X \quad | \quad \lambda x_1 \ldots x_n.yA_1 \cdots A_k \quad \text{s.t. } (\ldots)$$

  and obtain the approximants of a λI-term:

$$\mathrm{App}_{\mathrm{m}}(M) := \{A \in \mathcal{A} \mid \exists M \twoheadrightarrow_\beta N,\ A \sqsubseteq N\}.$$

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_{X}$ (for $X \subseteq_{\mathrm{f}} \mathcal{V}$) to the syntax.
  $\bot_X$ is “an undefined term whose set of free variables is X”.
- Define a **(head) approximation ordering** by

$$\frac{\forall i, \quad M_i \sqsubseteq N_i \qquad \mathrm{fv}(M_i) = \mathrm{fv}(N_i)}{\lambda\vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda\vec{x}.yN_1 \cdots N_k}$$

- Define **(head) approximants**:

$$\mathcal{A}_{\mathrm{m}}(X) \quad \ni \quad A, B, \ldots \quad ::= \quad \bot_X \quad | \quad \lambda x_1 \ldots x_n.yA_1 \cdots A_k \quad \text{s.t. } (\ldots)$$

  and obtain the approximants of a λI-term:

$$\mathrm{App}_{\mathrm{m}}(M) := \{A \in \mathcal{A} \mid \exists M \twoheadrightarrow_\beta N,\ A \sqsubseteq N\}.$$

- **Continuous approximation theorem:** $\mathrm{BT}(M) = \bigsqcup \mathrm{App}_{\mathrm{m}}(M)$

THE CONTINUOUS APPROXIMATION: IT WORKS AS USUAL

- Add constants $\bot_X$ (for $X \subseteq_{\mathrm{f}} \mathcal{V}$) to the syntax.
  $\bot_X$ is "an undefined term whose set of free variables is X".
- Define a **(head) approximation ordering** by

$$\frac{\forall i, \quad M_i \sqsubseteq N_i \qquad \mathrm{fv}(M_i) = \mathrm{fv}(N_i)}{\lambda\vec{x}.yM_1 \cdots M_k \sqsubseteq \lambda\vec{x}.yN_1 \cdots N_k}$$

- Define **(head) approximants**:

$$\mathcal{A}_{\mathrm{m}}(X) \quad \ni \quad A, B, \dots \quad ::= \quad \bot_X \quad | \quad \lambda x_1 \dots x_n.yA_1 \cdots A_k \quad \text{s.t. } (\dots)$$

  and obtain the approximants of a λI-term:

$$\mathrm{App}_{\mathrm{m}}(M) := \{A \in \mathcal{A}_{\mathrm{m}} \mid \exists M \twoheadrightarrow_\beta N,\ A \sqsubseteq N\}.$$

- **Continuous approximation theorem:** $\quad \mathrm{OT}(M) = \bigsqcup \mathrm{App}_{\mathrm{m}}(M)$

THE LINEAR APPROXIMATION (ORIGINAL ED.)

What you may usually call (multi)linear approximation:

Take a function:

$$f(x)$$

to a formal sum of n-linear approximations:

$$\sum_{n\in\mathbf{N}} \frac{1}{n!} \times \frac{\partial f(t)}{\partial t^n} \cdot x^n$$

via an operation of **Taylor expansion**.

THE LINEAR APPROXIMATION (ORIGINAL ED.)

What you will now call (multi)linear approximation:

Take a λ-term:

$$x \quad | \quad \lambda x.M \quad | \quad MN$$

to a formal sum of “multilinear λ-terms” (aka **resource λ-terms**):

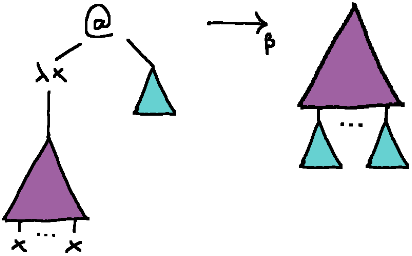
$$x \quad | \quad \lambda x.s \quad | \quad s[t_1, \dots, t_n]$$

via an operation of **Taylor expansion**. [Ehrhard & Regnier ’08]

THE LINEAR APPROXIMATION (ORIGINAL ED.)

We are able to simulate β-reduction of λ-terms:

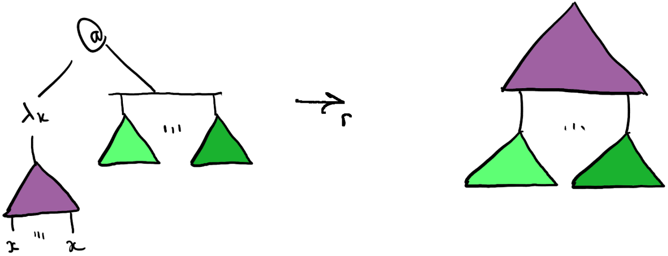
$$(\lambda x.M)N \quad \to_\beta \quad M[N/x]$$



THE LINEAR APPROXIMATION (ORIGINAL ED.)

We are able to simulate β-reduction of λ-terms...

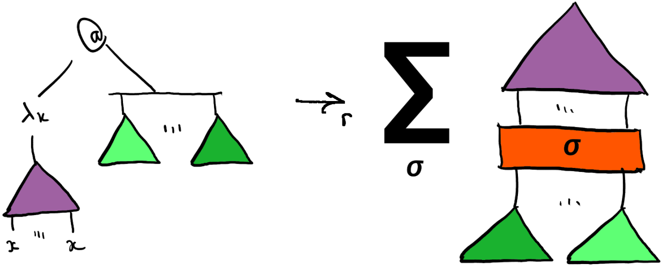
using multilinear β-reduction of resource λ-terms:



THE LINEAR APPROXIMATION (ORIGINAL ED.)

We are able to simulate β-reduction of λ-terms...

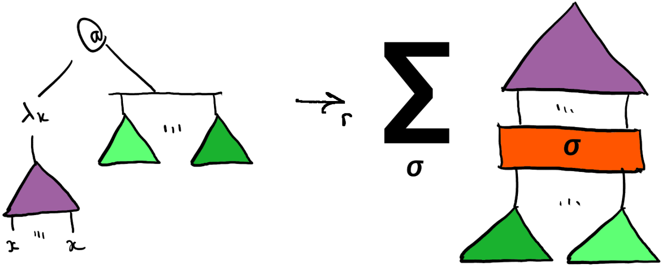
using multilinear β-reduction of resource λ-terms:



THE LINEAR APPROXIMATION (ORIGINAL ED.)

We are able to simulate β-reduction of λ-terms...

using multilinear β-reduction of resource λ-terms:



And in the usual setting we obtain the celebrated

Commutation theorem: $\mathrm{nf}(\mathcal{T}(M)) = \mathcal{T}(\mathrm{BT}(M))$. [Ehrhard & Regnier '06]

THE LINEAR APPROXIMATION (OHANA ED.) — λI-RESOURCE TERMS

In our setting, the resource λ-calculus is “fibered” over finite sets of variables:

$$\frac{}{x \in \Delta_{\mathrm{I}}(\{x\})} \qquad \frac{s \in \Delta_{\mathrm{I}}(X) \qquad x \in X}{\lambda x.s \in \Delta_{\mathrm{I}}(X - x)}$$

$$\frac{s \in \Delta_{\mathrm{I}}(X)}{s[\,]_Y \in \Delta_{\mathrm{I}}(X \cup Y)} \qquad \frac{s \in \Delta_{\mathrm{I}}(X) \qquad t_1, \dots, t_{n+1} \in \Delta_{\mathrm{I}}(Y)}{s[t_1, \dots, t_{n+1}] \in \Delta_{\mathrm{I}}(X \cup Y)}$$

As usual we define (fibered) sums $\mathbf{s}, \mathbf{t} \in \mathbf{N}[\Delta_{\mathrm{I}}(X)]$...

... and we extend the syntax by linearity, *e.g.*

$$(s + \mathbf{s})\bar{t} := s\bar{t} + \mathbf{s}\bar{t} \qquad \mathbf{0}_X \bar{t} := \mathbf{0}_{X \cup \mathrm{fv}(\bar{t})}.$$

THE LINEAR APPROXIMATION (OHANA ED.) — OPERATIONAL SEMANTICS

(For Taylor expansion nerds) here's what changes in the multilinear substitution:

$$x\langle \bar{u}/x\rangle := \begin{cases} u & \text{if } \bar{u} = [u] \\ \mathbf{0}_{\mathrm{fv}(\bar{u})} & \text{otherwise} \end{cases} \qquad y\langle \bar{u}/x\rangle := \begin{cases} y & \text{if } \bar{u} = []_{\mathrm{fv}(\bar{u})} \\ \mathbf{0}_{\{y\}} & \text{otherwise} \end{cases}$$

$$[]_X\langle \bar{u}/x\rangle := \begin{cases} []_{X\{\mathrm{fv}(\bar{u})/x\}} & \text{if } \bar{u} = []_{\mathrm{fv}(\bar{u})} \\ \mathbf{0}_{X\{\mathrm{fv}(\bar{u})/x\}} & \text{otherwise} \end{cases}$$

Resource reduction $\to_{\mathrm{r}}$ is generated as usual by

$$(\lambda x.s)\bar{t} \to_{\mathrm{r}} s\langle \bar{t}/x\rangle$$

and lifting to contexts and to finite sums in $\mathbf{N}[\Delta_{\mathrm{l}}(X)]$.

Theorem (sanity check)

$\to_{\mathrm{r}}$ is confluent and strongly normalising.

THE LINEAR APPROXIMATION (OHANA ED.) — TAYLOR APPROXIMATION

Usually,

$$\begin{aligned} \mathcal{T}(x) &:= \{x\} \\ \mathcal{T}(\lambda x.M) &:= \{\lambda x.s \mid s \in \mathcal{T}(M)\} \\ \mathcal{T}(MN) &:= \{s[t_1, \ldots, t_n] \mid s \in \mathcal{T}(M), t_i \in \mathcal{T}(N)\}. \end{aligned}$$

Now the Taylor expansion is also fibered:

$$\mathcal{T}_{\mathrm{m}}(M) := (\mathrm{fv}(M), \mathcal{T}^{\bullet}_{\mathrm{m}}(M))$$

where $\mathcal{T}^{\bullet}_{\mathrm{m}}(M)$ is the set of all λI-resource approximants of M.

In particular $\mathcal{T}_{\mathrm{m}}(\Omega x) = (\{x\}, \emptyset)$: Taylor expansion distinguishes Ωx and Ωy...

THE LINEAR APPROXIMATION (OHANA ED.) — THE APPROXIMATION THEOREM

We extend the construction to Ohana trees:

$$\mathcal{T}_{\mathrm{m}}(\mathrm{OT}(M)) := \bigcup^{\bullet}_{A \in \mathrm{App}_{\mathrm{m}}(M)} \mathcal{T}_{\mathrm{m}}(A) \qquad \text{with } \begin{cases} \mathcal{T}_{\mathrm{m}}(\bot_X) & := & (X, \emptyset) \\ \mathcal{T}^{!}_{\mathrm{m}}(\bot_X) & := & \{1_X\}. \end{cases}$$

and we finally obtain a

Commutation theorem

$\mathrm{nf}(\mathcal{T}_{\mathrm{m}}(M)) = \mathcal{T}_{\mathrm{m}}(\mathrm{OT}(M))$.

WE REACH OUR FIRST GOAL!

Theorem

$\mathcal{O}$ is a λI-theory.

Proof sektch. We use the fact that:

$$\begin{aligned} M =_{\mathcal{O}} N &\Leftrightarrow \mathrm{OT}(M) = \mathrm{OT}(N) \\ &\Leftrightarrow \mathcal{T}_{\mathrm{m}}(\mathrm{OT}(M)) = \mathcal{T}_{\mathrm{m}}(\mathrm{OT}(N)) \\ &\Leftrightarrow \mathrm{nf}(\mathcal{T}_{\mathrm{m}}(M)) = \mathrm{nf}(\mathcal{T}_{\mathrm{m}}(N)). \end{aligned}$$

- $\mathcal{O}$ contains $=_\beta$.
- Compatibility with abstraction: if $M =_{\mathcal{O}} N$ and $x \in \mathrm{fv}(M) \cap \mathrm{fv}(N)$ then

$$\mathrm{nf}(\mathcal{T}_{\mathrm{m}}(\lambda x.M)) = \lambda x.\mathrm{nf}(\mathcal{T}_{\mathrm{m}}(M)) = \lambda x.\mathrm{nf}(\mathcal{T}_{\mathrm{m}}(N)) = \mathrm{nf}(\mathcal{T}_{\mathrm{m}}(\lambda x.N)).$$

- Compatibility with application: same... ☐

A MULTI-TYPE SEMANTICS CHARACTERISING THE THEORY $\mathcal{O}$ OF OHANA TREES

WHAT HAPPENED SO FAR...

λ A model of the λ-calculus $\rightsquigarrow$ A λ-theory $\leftsquigarrow$ A notion of evaluation tree

$M =_{\mathcal{B}} N$ iff $\mathrm{BT}(M) = \mathrm{BT}(N)$ $\mathrm{BT}(-)$

λI A λI-theory $\leftsquigarrow$ A notion of evaluation tree

$M =_{\mathcal{O}} N$ iff $\mathrm{OT}(M) = \mathrm{OT}(N)$ $\mathrm{OT}(-)$

WHAT HAPPENED SO FAR… AND WHAT’S GOING ON

λ	A model of the λ-calculus	$\rightsquigarrow$	A λ-theory	$\leftsquigarrow$	A notion of evaluation tree
			$M =_{\mathcal{B}} N$ iff $\mathrm{BT}(M) = \mathrm{BT}(N)$		$\mathrm{BT}(-)$
λI	???	$\rightsquigarrow$	A λI-theory	$\leftsquigarrow$	A notion of evaluation tree
			$M =_{\mathcal{O}} N$ iff $\mathrm{OT}(M) = \mathrm{OT}(N)$		$\mathrm{OT}(-)$

Is there a λI-model whose theory is $\mathcal{O}$?

Well, we don’t even know what a λI-model is…

… but we can try to keep applying the usual recipe:

- turn our linear approximation into a multi-type system,
- build the corresponding relational model.

A MULTI-TYPE SYSTEM CHARACTERISING $\mathcal{O}$

Types:

$$\begin{array}{rcrcll}
?\mathbf{T} & \ni & \sigma, \tau, \ldots & := & \mathbf{0}_X \mid \alpha & X \subseteq_{\mathsf{f}} \mathcal{V} \\
\mathbf{T} & \ni & \alpha, \beta, \ldots & := & * \mid \bar{\alpha} \multimap \beta & * \in \mathcal{A} \\
!\mathbf{T} & \ni & \bar{\alpha}, \bar{\beta}, \ldots & := & []_X \mid [\alpha_0, \ldots, \alpha_n] & X \subseteq_{\mathsf{f}} \mathcal{V},\ n \in \mathbf{N}
\end{array}$$

Contexts:

- Γ maps (all) variables to multisets of types (it's the usual one),
- Δ maps (only) the free variables of the typed term to sets of variables (intuitively, x that will be substituted with N is mapped to $\mathrm{fv}(N)$).

Judgements:

$$\Gamma ; \Delta \vdash M : \sigma \qquad\qquad \Gamma ; \Delta \vdash^{!} M : \bar{\alpha}$$

A MULTI-TYPE SYSTEM CHARACTERISING $\mathcal{O}$

$$\frac{\operatorname{dom}(\Delta) = \operatorname{fv}(M)}{\,;\Delta \vdash M : \mathbf{0}_{\cup \operatorname{im}(\Delta)}}\ (\mathbf{0})$$

$$\frac{}{x : [\alpha] \,;\, x : X \vdash x : \alpha}\ (ax) \qquad \frac{\Gamma_0 \,;\, \Delta_0 \vdash M : \bar{\alpha} \multimap \beta \qquad \Gamma_1 \,;\, \Delta_1 \vdash^{!} N : \bar{\alpha} \qquad \Delta_0 \asymp \Delta_1}{\Gamma_0 + \Gamma_1 \,;\, \Delta_0 \vee \Delta_1 \vdash MN : \beta}\ (@)$$

$$\frac{\Gamma, x : [\,] \,;\, \Delta, x : X \vdash M : \beta}{\Gamma \,;\, \Delta \vdash \lambda x.M : [\,]_X \multimap \beta}\ (\lambda^0) \qquad \frac{\Gamma, x : [\alpha_0, \dots, \alpha_n] \,;\, \Delta, x : X \vdash M : \beta}{\Gamma \,;\, \Delta \vdash \lambda x.M : [\alpha_0, \dots, \alpha_n] \multimap \beta}\ (\lambda^+)$$

$$\frac{\operatorname{dom}(\Delta) = \operatorname{fv}(M)}{\,;\Delta \vdash^{!} M : [\,]_{\cup \operatorname{im}(\Delta)}}\ (!^0) \qquad \frac{\Gamma_0 \,;\, \Delta \vdash M : \alpha_0 \qquad \dots \qquad \Gamma_n \,;\, \Delta \vdash M : \alpha_n}{\Gamma_0 + \dots + \Gamma_n \,;\, \Delta \vdash^{!} M : [\alpha_0, \dots, \alpha_n]}\ (!^+)$$

A MULTI-TYPE SYSTEM CHARACTERISING $\mathcal{O}$

The typing system enjoys **subject reduction** and **subject expansion**. Good news!

A MULTI-TYPE SYSTEM CHARACTERISING $\mathcal{O}$

The typing system enjoys **subject reduction** and **subject expansion**. Good news!

In addition, if one defines

$$[\![M]\!] := \{(\Gamma, \Delta, \sigma) \mid \Gamma \,;\, \Delta \vdash M : \sigma\}$$

we obtain the following

Theorem

For all $M, N \in \Lambda_{\mathsf{I}}$, $[\![M]\!] = [\![N]\!]$ iff. $M =_{\mathcal{O}} N$.

WHAT ABOUT A λI-MODEL?

It is not completely obvious how to extract a λI-model from this typing system.

WHAT ABOUT A λI-MODEL?

It is not completely obvious how to extract a λI-model from this typing system.

Neither is it obvious what a λI-model is!

- There's a proposition [Jacobs '93] with two instances (strict cpo's and some relational model with non-empty multisets),
  but it doesn't seem to be general enough for a model with theory $\mathcal{O}$ to exist.

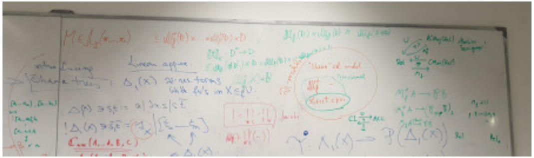
WHAT ABOUT A λI-MODEL?

It is not completely obvious how to extract a λI-model from this typing system.

Neither is it obvious what a λI-model is!

- There's a proposition [Jacobs '93] with two instances (strict cpo's and some relational model with non-empty multisets),
  but it doesn't seem to be general enough for a model with theory $\mathcal{O}$ to exist.
- It should be something more general than a λ-model,
  our candidate: a cartesian closed multicategory with contractions.

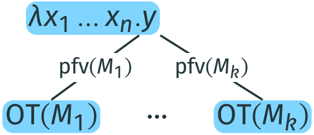
But don't ask me more, all the notes we have look like this:



QUESTIONS YOU MAY ASK

OHANA TREES FOR THE FULL λ-CALCULUS?

By the way, the Ohana tree of a λ-term M can be defined coinductively by



$$\mathrm{OT}(M) := \begin{cases} \begin{array}{ccc} & \lambda x_1 \dots x_n.y & \\ \mathrm{pfv}(M_1) & & \mathrm{pfv}(M_k) \\ \mathrm{OT}(M_1) & \cdots & \mathrm{OT}(M_k) \end{array} & \text{if } M \twoheadrightarrow_h \lambda x_1 \dots x_n.yM_1 \cdots M_k, \\ \bot_{\mathrm{pfv}(M)} & \text{otherwise.} \end{cases}$$

where

$$\mathrm{pfv}(M) := \{x \in \mathcal{V} \mid \forall M \twoheadrightarrow_\beta N,\ x \in \mathrm{fv}(N)\}.$$

But Ohana tree equality does **not** induce a λ-theory. ☹

May Ohana trees induce an interesting observational equivalence?

WHY SHOULD WE CARE?

- **It's fun.** (Right?)

WHY SHOULD WE CARE?

- **It's fun.** (Right?)
- Since the λ-calculus is an (equationally) conservative extension of the λI-calculus, **λI-theories allow to β-separate λ-terms** (in the λ-calculus).

THE DOUBLE FIXED-POINT COMBINATOR CONJECTURE

Sometimes in λ-calculus we want to **separate** two λ-terms M and N, *i.e.* prove that they are not β(η)-equivalent.

An example (open) problem.

- Take $\delta := \lambda yx.x(yx)$. Then: [Böhm, van der Mey]
  - Y is a fpc $\iff$ $\delta Y =_\beta Y$ (hence is a fpc too)
  - Y is a fpc $\implies$ $Y\delta$ is a fpc
- Starting with $Y_0 :=$ your preferred fpc, define $Y_{n+1} := Y_n\delta$.
  For $Y_0 :=$ Curry's fpc, all the terms of the sequence are β-distinct. [Klop '07]
  And in general? It's an open question.
- More generally, **is there a fpc Y s.t. $Y =_\beta Y\delta$?** [Statman '93]
  Conjecture: such a "double fixed-point combinator" doesn't exist.

ATTACKING A SEPARATION PROBLEM

Standard ideas:

- Find a separating context (as in Böhm's theorem), *i.e.* $C[\ ]$ such that $C[M] \to^* x$ and $C[N] \to^* y$.
- (...)
- Find a separating λ-theory (or λ-model), *i.e.* $=_{\mathcal{T}}$ such that $M \neq_{\mathcal{T}} N$ can be proved.

As for the double fpc problem:

- The Ohana theory doesn't help...
- ... but suggest some ideas! One may design evaluation trees such that in

  $$Y\delta f =_\beta \delta(Y\delta)f =_\beta f(Y\delta f) =_\beta \dots$$

  something like the subterm $Y\delta f$ is remembered at infinity.

WHY SHOULD WE CARE?

- **It's fun.** (Right?)
- Since the λ-calculus is an (equationally) conservative extension of the λI-calculus, **λI-theories allow to β-separate λ-terms** (in the λ-calculus).
- **Long-term goal: to infinity and beyond.**

  Our formalism accounts for free variables pushed to infinity.
  It is a first step: what about pushing whole subterms to infinity?
  - This suggests working with **transfinite terms** and investigate the associated rewriting.
  - It is not clear what evaluation trees and semantics may look like in such a jungle. (But we have some preliminary ideas.)

Thanks for your attention!