On the characterization of models of $\mathcal{H}^*$

Flavien BREUVART

PPS, Paris Denis Diderot; LIPN, Paris Nord

CSL-LICS, 2014

Full abstraction

Full abstraction

Denotational and operational congruences coincide :

$$[\![M]\!] = [\![N]\!] \Leftrightarrow M \equiv_o N$$

Denotational congruence $[\![M]\!] = [\![N]\!]$

Congruence induced by the denotational model :

equality of the interpretation

Operational congruence $M \equiv_o N$

Congruence induced by the language :

$$\forall C, \mathcal{O}(C(\!|M|\!)) = \mathcal{O}(C(\!|N|\!))$$

e.g., $\mathcal{O}$ = head-normalization

Many results of full abstraction

[Abramsky & Ong 1993]

[Ouaknine 2002]

[Ehrhard *et. al.* 2014]

[Breuvart 2013]

[Hennessy & Plotkin 1979]

[Manzonetto 2009]

[Joung & Stoughton 1993]

[Hyland & Ong 2000]

[Laird *et. al.* 2011]

[Coppo,Dezani&Zacchi 1987]

[Barendregt 1984]

[Mazza & Ross 2012]

[Harmer & McCusker 1999]

[Wadsworth 1976]

[Laird 1997]

[Abramsky & McCusker 2007]

[Plotkin 1977]

[Cartwright *et.al.* 1994]

[Mazza 2009]

[Milner 1977]

[Berry *et. al.* 1985]

[Hyland 1976]

[Paolini 2003]

[Gouy 1995]

[Bucciarelli *et.al.* 2011]

[Paolini & RoncciDellaRocca 2004]

[Abramsky *et. al.* 1998]

[AJM 2000]

Our question

Previous works

[Milner1977] **Milner's theorem for PCF :**
There is a unique domain fully abstract for PCF
(continuous, extensional and up to iso.)

[Gouy1995] **And for the pure λ-calculus ?** (head reduction)
There exist many non-isomorphic models
fully abstract for $\mathcal{H}^*$

[Manzonetto2009] **Sufficient condition**
Any well-stratified model, *i.e.*, st

$$|f|_{k+1}(|a|_k) = |f(a)|_k \qquad |f|_0(\bot) = |f(\bot)|_0$$

is fully abstract for $\mathcal{H}^*$.

May we characterize the models fully abstract for $\mathcal{H}^*$?

Our answer

The result (informally) :

A K-model D is fully abstract for $\mathcal{H}^*$ iff D is hyperimmune,

i.e., D may contain non-well-founded chains but they are not "accessible" to λ-terms

$$\alpha_1 = a_{1,1} \to \cdots a_{1,i_1} \cdots \to a_{1,g(1)} \to \alpha'_1$$

$$\Cup$$

$$\alpha_2 = a_{2,1} \to \cdots a_{2,i_2} \cdots \to a_{2,g(2)} \to \alpha'_2$$

$$\Cup$$

$$\alpha_3 = a_{3,1} \to \cdots a_{3,i_3} \cdots \to a_{3,g(3)} \to \alpha'_3$$

$$\ddots$$

Table of contents

Extensional K-models

1) Λ-models : Categorical
2) K-models : Definition
3) Examples : D_∞, P_∞, D^*_∞...

The result

1) Hyperimmunity : Definition
2) Intuition : Non-well founded
3) Examples : Well-strat., H^f

The λ-calculi with tests

1) Principle : Model-dep.
2) Intuition : Type inference

Böhm trees

1) Definitions : BT & Approx.
2) Interest : Test & FA

A draft of the proof

1) Hyperimmunity $\Rightarrow$ Full abst. : A counter-example J_g
2) Full abst. $\Rightarrow$ Hyperimmunity : Alternation of (co-)inductions

Table of contents

Models of pure λ-calculus

model of (pure) λ-calculus

Model for typed λ-calculus : Cartesian closed category $(D \Rightarrow E)$.

Model for pure λ-calculus : reflexive object in this ccc :

$$D \underset{app}{\overset{abs}{\rightleftarrows}} D \Rightarrow D \qquad\qquad abs \circ app = id_{D \Rightarrow D}$$

Extensionality

Full Abstraction for $\mathcal{H}^*$ $\Rightarrow$ extensionality (η-equivalence) :

$$app \circ abs = id_D$$

We only consider models respecting the approximation theorem.

Definition of K-models

Extensional K-model [Krivine 1993] :

A preorder $(D, \leq)$ with a bijection "$\to$" from $\mathcal{A}_f(D)^{op} \times D$ to D.

Antichains

$\mathcal{A}_f(D)$ are finite antichains over D, i.e. $a \in \mathcal{A}_f(D)$ if :

$$\forall \alpha, \beta \in a, \quad \alpha \not\leq \beta$$

Order on antichains

The order $\leq$ on D extends to $\mathcal{A}_f(D)$ by $a \leq b$ iff :

$$\forall \alpha \in a, \exists \beta \in b, \quad \alpha \leq \beta$$

We can replace $\mathcal{A}_f(D)$ by $\mathcal{P}_f(D), \mathcal{M}_f(D)$ or intersections

- Sub-class of filter models.
- Contain historical models : D_∞, P_∞, D_∞^*...
- Contain all well-stratified filter models.

K-models are the reflexive elements of $\text{ScottL}_!$

A model of linear logic : ScottL [Ehrhard2012]

Objects : Posets **Morphisms** : linear fct. $\mathcal{I}(D) \to \mathcal{I}(P)$

Exponential : Finite antichains $!D = \mathcal{A}_f(D)$

$\mathcal{I}(D)$ represents the complete lattice of initial segments over D
a function is said linear if it preserves every sups

The Kleisli category : $\text{ScottL}_!$

Objects : Posets **Morphisms** : continuous fct. $\mathcal{I}(D) \to \mathcal{I}(P)$

Identities : $1_D = id_{\mathcal{I}(D)}$

Composition : the function composition

Cartesian product : $\&_{i\in I} D_i := \{(i,\alpha) | i \in I, \alpha \in A_D\}$

Exponential object : $A \Rightarrow B = \mathcal{A}_f(A)^{op} \times B$

Examples (D_∞)

Scott's D_∞ :

$$D_0 = \{*\}$$
$$D_{n+1} = D_n \cup (\mathcal{A}_f(D_n) \times D_n) - \{(\emptyset, *)\}$$
$$D_\infty = \bigcup_n D_n$$

The order is uniquely generated

$$(a, \alpha) = a \to \alpha \qquad \text{if } a \neq \emptyset \text{ or } \alpha \neq *$$
$$* = \emptyset \to *$$

Examples (D_∞)

Scott's D_∞ :

$$D_0 = \{*\}$$
$$D_{n+1} = D_n \cup (\mathcal{A}_f(D_n) \times D_n) - \{(\emptyset, *)\}$$
$$D_\infty = \bigcup_n D_n$$

$$(a, \alpha) = a \to \alpha \qquad \text{if } a \neq \emptyset \text{ or } \alpha \neq *$$
$$* = \emptyset \to *$$

The order is uniquely generated

Completion

This is an extensional completion : $D_\infty = \overline{D_0}$

D_∞ is fully abstract for $\mathcal{H}^*$
[Hyland 1976, Wadsworth 1976]

Examples (P_∞)

Park's P_∞ :

$$P_0 = \{*\}$$

$$P_{n+1} = P_n \cup (\mathcal{A}_f(P_n) \times P_n) - \{(\{*\}, *)\}$$

$$P_\infty = \bigcup_n P_n$$

$$(a, \alpha) = a \to \alpha \qquad \text{if } a \neq \{*\} \text{ or } \alpha \neq *$$

$$* = \{*\} \to *$$

P_∞ is not fully abstract for $\mathcal{H}^*$

[Park 1976]

Behaviors of D_∞ and P_∞

Hyperimmunity of D_∞

$$(a,\alpha) = a \to \alpha$$
$$\Cup$$
$$(b,\beta) = b \to \beta$$
$$\ddots$$
$$* = \emptyset \to *$$

Non-hyperimmunity of P_∞

$$* = \{*\} \to *$$
$$\Cup$$
$$* = \{*\} \to *$$
$$\Cup$$
$$* = \{*\} \to *$$
$$\ddots$$

Remark

By bijectivity of $\to$, every $\alpha \in D$ is an arrow.

In the λ-calculus, every term is a function

Example ($\overline{\omega}$)

Ordinal completion $\overline{\omega}$:

$E_0 = \mathbb{N}$ $\quad\quad\quad n = \{i \mid i < n\} \rightarrow n$

Hyperimmunity of $\overline{\omega}$

$$n_0 = [0, n_0[\rightarrow \cdots [0, n_0[\cdots \rightarrow n_0$$

$$\Cup\ (n_1 < n_0)$$

$$n_1 = [0, n_1[\rightarrow \cdots [0, n_1[\cdots \rightarrow n_1$$

$$\Cup\ (n_2 < n_1)$$

$$\ddots$$

$$0 = \emptyset \cdots \emptyset \cdots \emptyset \rightarrow 0$$

Example (D_∞^*)

Coppo,Dezani&Zacchi's D_∞^* (or Norm) : :

$N_0 = \{p, q\}$ $\qquad q = \{p\} \to q$ $\qquad p = \{q\} \to p$

D_∞^* is not fully abstract for $\mathcal{H}^*$ (but is for $\mathcal{H}$)
[Coppo,Dezani&Zacchi 1987]

Non-hyperimmunity of D_∞^*

$$p = \{q\} \to p$$
$$\Uplus$$
$$q = \{p\} \to q$$
$$\Uplus$$
$$p = \{q\} \to p$$
$$\ddots$$

Table of contents

Characterization

Hyperimmune models

A K-model D is *hyperimmune* when, $\forall(\alpha_n)_{n\geq 0}, \forall g : \mathbb{N} \to \mathbb{N}$,

if for all n :

$$\alpha_n = a_{n,1} \to \cdots \to a_{n,g(n)} \to \alpha'_n \quad \text{and} \quad \alpha_{n+1} \in \bigcup_{k \leq g(n)} a_{n,k}.$$

then g is not recursive

- Generalization of Manzonetto's stratification.
- Intrinsically logically complex property.
- Link with recursion theory (hyperimmune functions).

Some intuitions

$$\alpha_1 = a_{1,1} \to \cdots a_{1,i_1} \cdots \to a_{1,g(1)} \to \alpha'_1$$

$$\Cup$$

$$\alpha_2 = a_{2,1} \to \cdots a_{2,i_2} \cdots \to a_{2,g(2)} \to \alpha'_2$$

$$\Cup$$

$$\alpha_3 = a_{3,1} \to \cdots a_{3,i_3} \cdots \to a_{3,g(3)} \to \alpha'_3$$

$$\ddots$$

Is D hyperimmune ?

If g is recursive and $g(k) \geq i_k$ for all k, then D is not hyperimmune (even if $(i_n)_n$ may not be recursive).

Examples (well-stratified)

Reminder on well-stratification :

It is the approximation by projections $(|.|_k)_{k\in\mathbb{N}}$ s.t.

$$|f|_{k+1}(x_k) = |f(x)|_k \qquad\qquad |f|_0(\bot) = |f(\bot)|_0$$

(Equivalent presentation of) Well-stratified K-models :

They are extensional completions of (where σ is a permutations)

$$U_0^{A,\sigma} = A \qquad\qquad \forall\alpha \in A, \quad \alpha = \emptyset \to \sigma(\alpha)$$

The completion preserves hyperimmunity

A completion $\bar{E}$ is hyperimmune iff E is.

Well-stratified models are hyperimmune

for all $\alpha_1 \in A$ and $g : \mathbb{N} \to \mathbb{N}$,

$$\alpha_1 = \emptyset \to \cdots \to \emptyset \to \sigma^n(\alpha_1)$$

False ordinal completion ($\overline{\mathbb{N}}$)

Reminder of the ordinal completion $\overline{\omega}$:

$E_0 = \mathbb{N}$ $\qquad$ $n = [0, n[\to n$

False ordinal completion $\overline{\mathbb{N}}$:

$E_0 = \mathbb{N}$ $\qquad$ $n = [0, n[\to (n+1)$

Norm is not hyperimmune

$$g = (n \mapsto 2)$$

$$(\alpha_n)_n = (n, n, n, ...)$$

$$\begin{aligned} n &= [0, n[\to (n+1) \\ &= [0, n[\to [0, n] \to (n+2) \end{aligned}$$

$$\Cup$$

$$\begin{aligned} n &= [0, n[\to (n+1) \\ &= [0, n[\to [0, n] \to (n+2) \end{aligned}$$

$$\ddots$$

Example (H^f)

The non-well founded H^f :

$$H_0^f = \{*, \alpha_n \mid n\} \quad * = \emptyset \to * \quad \alpha_n = \underbrace{\emptyset \to \cdots \to}_{f(n)} \emptyset \to \{\alpha_{n+1}\} \to *$$

H^f is hyperimmune iff there is g st :

$$\alpha_0 = \emptyset \to \underbrace{\cdots}_{f(0)} \emptyset \to \{\alpha_1\} \to \emptyset \underbrace{\cdots}_{g(0)-f(0)} \to *$$

$$\alpha_1 = \emptyset \to \underbrace{\cdots}_{f(1)} \emptyset \to \{\alpha_2\} \to \emptyset \underbrace{\cdots}_{g(1)-f(1)} \to *$$

$$\alpha_2 = \emptyset \to \underbrace{\cdots}_{f(2)} \emptyset \to \{\alpha_3\} \cdots$$

$$\ddots$$

H^f is hyperimmune iff f is not below any recursive function g.

Our main result

Theorem

For any extensional K-models D, the following are equivalent when D respects approximation theorem :

- D is hyperimmune,
- D is inequationally fully abstract for $\mathcal{H}^*$,
- D is fully abstract for $\mathcal{H}^*$.

Table of contents

Extensional K-models

1) Λ-models : Categorical
2) K-models : Definition
3) Examples : D_∞, P_∞, D_∞^*...

The result

1) Hyperimmunity : Definition
2) Intuition : Non-well founded
3) Examples : Well-strat., H^f

The λ-calculi with tests

1) Principle : Model-dep.
2) Intuition : Type inference

Böhm trees

1) Definitions : BT & Approx.
2) Interest : Test & FA

A draft of the proof

1) Hyperimmunity $\Rightarrow$ Full abst. : A counter-example J_g
2) Full abst. $\Rightarrow$ Hyperimmunity : Alternation of (co-)inductions

$\Lambda_{\tau(D)}$: a model-specific language

$\Lambda_{\tau(D)}$ extends Λ with elements of D

We add the operators $\bar{\epsilon}_\alpha$, $\bar{\tau}_\alpha(Q)$ and $\tau_\alpha(M)$ for all $\alpha \in D$

We have some control over the assertion $\alpha \in [\![M]\!]$

Upto the approxiation theorem, D co-defines all prime algebraic elements :

$$\tau_\alpha(M)\Downarrow \Leftrightarrow \alpha \in [\![M]\!]$$

Full abstraction via deffinissability

Upto the approxiation theorem, D is fully abstract for $\Lambda_{\tau(D)}$ by defining all prime algebraic :

$$\forall \alpha \in D, \quad [\![\bar{\epsilon}_\alpha]\!] = \downarrow \alpha$$

Syntax of tests

2 syntactic kinds

(term) $M, N ::= x \mid \lambda x.M \mid M\ N \mid \sum_{i\leq n} \bar{\tau}_{\alpha_i}(Q_i) \quad , \forall(\alpha_i)_i \in D^n$

(test) $P, Q ::= \sum_{i\leq n} P_i \mid \prod_{i\leq n} P_i \mid \tau_\alpha(M) \quad , \forall \alpha \in D$

Polarised view : tests are processes

$\tau_\alpha(M) \quad \simeq \quad M * \alpha \qquad \bar{\tau}_\alpha(Q) * \pi \quad \simeq \quad Q\cdot(\bar{\alpha} * \pi)$

Reduction strategy (extending head reduction)

$\tau(M)$: infinite application, $\bar{\tau}(Q)$: infinite abstraction,

$\Sigma_i E_i$: may non-determinism, $\Pi_i Q_i$: must non-determinism.

Tests and typing

Type inference procedural for $\vdash \omega : 2$

$\vdash \omega : 2$

Reduction of $\tau_2(\omega)$

$\tau_2(\omega)$

Notations

- $\omega = \lambda x.x\ x$
- D is the completion on $\mathbb{N}$ with $n = [0, n[\rightarrow 0$

Tests and typing

Type inference procedural for $\vdash \omega : 2$

$$\frac{x : \{0,1\} \vdash x\ x : 0}{\vdash \omega : 2} \quad 2 = \{0,1\} \to 0$$

Reduction of $\tau_2(\omega)$

$\tau_2(\omega) \ \to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}})$

Notations

- $\omega = \lambda x.x\ x$
- D is the completion on $\mathbb{N}$ with $n = [0, n[\to 0$

Tests and typing

Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0 \qquad x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0$$

Reduction of $\tau_2(\omega)$

$$\begin{aligned} \tau_2(\omega) &\to \tau_0(\bar\epsilon_{\{0,1\}}\ \bar\epsilon_{\{0,1\}}) \\ &\to \tau_0(\bar\epsilon_0\ \bar\epsilon_{\{0,1\}}) + \tau_0(\bar\epsilon_1\ \bar\epsilon_{\{0,1\}}) \end{aligned}$$

Notations

- $\omega = \lambda x.x\ x$
- D is the completion on $\mathbb{N}$ with $n = [0, n[\to 0$

Tests and typing

Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{\dfrac{\cdots}{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0} \qquad \dfrac{x : \{0,1\} \vdash x : 0 \qquad 1 \leq 0}{x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}\ 1 = 0{\to}0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\}{\to}0$$

Reduction of $\tau_2(\omega)$

$$\begin{aligned}
\tau_2(\omega) &\to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_1\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\tau}_0(\tau_0(\bar{\epsilon}_{\{0,1\}})))
\end{aligned}$$

Notations

- $\omega = \lambda x.x\ x$
- D is the completion on $\mathbb{N}$ with $n = [0, n[\to 0$

Tests and typing

Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{\dfrac{\dots}{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0} \qquad \dfrac{x : \{0,1\} \vdash x : 0 \qquad \overline{1 \leq 0}}{x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}\ 1 = 0 \to 0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0$$

Reduction of $\tau_2(\omega)$

$$\begin{aligned}
\tau_2(\omega) &\to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_1\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\tau}_0(\tau_0(\bar{\epsilon}_{\{0,1\}}))) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_{\{0,1\}})
\end{aligned}$$

Notations

- $\omega = \lambda x.x\ x$
- D is the completion on $\mathbb{N}$ with $n = [0, n[\to 0$

Tests and typing

Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{\dfrac{\cdots}{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0} \qquad \dfrac{\dfrac{}{x : \{0,1\} \vdash x : 0} \quad \dfrac{}{1 \leq 0}}{x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}\ 1 = 0 \to 0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0$$

Reduction of $\tau_2(\omega)$

$$\begin{aligned}
\tau_2(\omega) &\to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_1\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\tau}_0(\tau_0(\bar{\epsilon}_{\{0,1\}}))) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \epsilon
\end{aligned}$$

Notations

- $\omega = \lambda x.x\ x$
- D is the completion on $\mathbb{N}$ with $n = [0, n[\to 0$

Infinite derivation

$$\tau_p(J\ \bar{\epsilon}_p)$$

$$x : p \vdash J\ x : p$$

Notations

- $J = Y\ (\lambda uxy.x\ (u\ y))$,
- Y is a fixpoint.
  e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

Model : D^*_∞

Completion of $\{p, q\}$ with :

$$p = \{q\} \rightarrow p$$
$$q = \{p\} \rightarrow q$$

Infinite derivation

$$\tau_p(J\ \bar{\epsilon}_p)\ \rightarrow^*\ \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y))$$

$$\frac{x : p \vdash \lambda y.x\ (J\ y) : p}{x : p \vdash J\ x : p}\ J\ x \rightarrow_h^* \lambda y.x\ (J\ y)$$

Notations

- $J = Y\ (\lambda uxy.x\ (u\ y))$,
- Y is a fixpoint.
  e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

Model : D_∞^*

Completion of $\{p, q\}$ with :

$$p = \{q\} \rightarrow p$$
$$q = \{p\} \rightarrow q$$

Infinite derivation

$$\tau_p(J\ \bar{\epsilon}_p) \to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y)) \to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q))$$

$$\dfrac{\dfrac{x : p, y : q \vdash x\ (J\ y) : p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

Notations

- $J = Y\ (\lambda uxy.x\ (u\ y))$,
- Y is a fixpoint.

  e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

Model : D_∞^*

Completion of $\{p, q\}$ with :

$$p = \{q\}{\to}p$$
$$q = \{p\}{\to}q$$

Infinite derivation

$$\dfrac{\dfrac{\dfrac{y : q \vdash J\ y : q \qquad p \leq p}{x : p, y : q \vdash x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

$$\begin{aligned} \tau_p(J\ \bar{\epsilon}_p) &\to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y)) \\ &\to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q)) \\ &\to \tau_p(\bar{\tau}_p(\tau_q(J\ \bar{\epsilon}_q))) \end{aligned}$$

Notations

- $J = Y\ (\lambda uxy.x\ (u\ y))$,
- Y is a fixpoint.
  e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

Model : D_∞^*

Completion of $\{p, q\}$ with :

$$p = \{q\}{\to}p$$
$$q = \{p\}{\to}q$$

Infinite derivation

$$\dfrac{\dfrac{\dfrac{y:q\vdash J\ y:q \qquad \overline{p\leq p}}{x:p,y:q\vdash x\ (J\ y):p}\ p=q\to p}{x:p\vdash \lambda y.x\ (J\ y):p}\ p=q\to p}{x:p\vdash J\ x:p}\ J\ x\to_h^* \lambda y.x\ (J\ y)$$

$$\begin{aligned}
\tau_p(J\ \bar{\epsilon}_p) &\to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y))\\
&\to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q))\\
&\to \tau_p(\bar{\tau}_p(\tau_q(J\ \bar{\epsilon}_q)))\\
&\to \tau_q(J\ \bar{\epsilon}_q)
\end{aligned}$$

Notations

- $J = Y\ (\lambda uxy.x\ (u\ y))$,
- Y is a fixpoint.
  e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

Model : D^*_∞

Completion of $\{p, q\}$ with :

$$p = \{q\}\to p$$
$$q = \{p\}\to q$$

Infinite derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dots}{y : q \vdash J\ y : q} \qquad \dfrac{}{p \leq p}}{x : p, y : q \vdash x\ (J\ y) : p}\ p = q \to p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q \to p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

$$\begin{aligned}
\tau_p(J\ \bar{\epsilon}_p) &\to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y)) \\
&\to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q)) \\
&\to \tau_p(\bar{\tau}_p(\tau_q(J\ \bar{\epsilon}_q))) \\
&\to \tau_q(J\ \bar{\epsilon}_q) \\
&\to \dots
\end{aligned}$$

Notations

- $J = Y\ (\lambda uxy.x\ (u\ y))$,
- Y is a fixpoint.
  e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

Model : D_∞^*

Completion of $\{p, q\}$ with :

$$p = \{q\} \to p$$
$$q = \{p\} \to q$$

Procedural of $\tau_\alpha(M)$

$$\Gamma \vdash M : \alpha$$

Procedural of $\tau_\alpha(M)$

$$\frac{\Gamma \vdash \lambda x_1 ... x_n . x_k \ N_1 \cdots N_m : \alpha}{\Gamma \vdash M : \alpha} \ M \to_h^* \lambda x_1 ... x_n . x_k \ N_1 \cdots N_m$$

Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\Gamma \vdash \lambda x_1 ... x_n . x_k \;\; N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash \lambda x_1 ... x_n . x_k \;\; N_1 \cdots N_m : \alpha} \; \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha} \; M \to_h^* \lambda x_1 ... x_n . x_k \;\; N_1 \cdots N_m$$

Procedural of $\tau_\alpha(M)$

$$\cfrac{\cfrac{\cfrac{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m$$

Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m$$

Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m$$

Procedural of $\tau_\alpha(M)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma' \vdash N_i : \gamma_i \qquad \alpha' \leq \beta'}{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \forall i, \forall \gamma_i \in b_i}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m
$$

Procedural of $\tau_\alpha(M)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dots}{\Gamma' \vdash N_i : \gamma_i} \qquad \alpha' \leq \beta'}{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \forall i, \forall \gamma_i \in b_i}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 \dots x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\dfrac{\Gamma \vdash \lambda x_1 \dots x_n . x_k\ N_1 \cdots N_m : \alpha}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 \dots x_n . x_k\ N_1 \cdots N_m}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'
$$

Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dots}{\Gamma' \vdash N_i : \gamma_i} \quad \alpha' \leq \beta'}{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \forall i, \forall \gamma_i \in b_i}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m$$

4 possible failures

- M diverges
- $a_k = \emptyset$
- $\alpha' \not\leq \beta'$
- infinite derivation

Consistence

This procedure succeeds iff $\alpha \in [\![M]\!]^\Gamma$

by the approximation theorem's hypothesis

$\Lambda_{\tau(D)}$

The λ-calculus with D-tests internalizes this reduction :

$\tau_\alpha(M)\Downarrow \Leftrightarrow \alpha \in [\![M]\!]$

Table of contents

Böhm trees

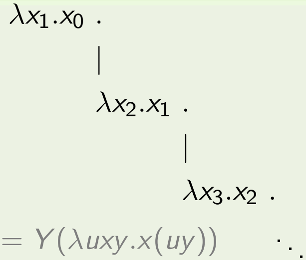
Definition of the Böhm tree $\mathbf{BT}(M)$ of a term M

$\mathbf{BT}$ is a coinductive structure defined by :

- If M head diverges, $\mathbf{BT}(M) = \Omega$,
- if $M \to_h^* \lambda x_1 ... x_n . y \ N_1 \cdots N_k$ then

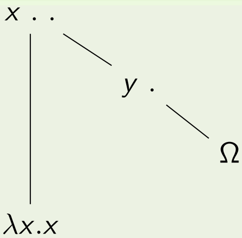
$$\mathbf{BT}(M) = \lambda x_1 ... x_n . y \ \mathbf{BT}(N_1) \cdots \mathbf{BT}(N_k).$$

Example : $\mathbf{BT}(J \ x_0)$



$J = Y(\lambda uxy.x(uy))$

Example : $\mathbf{BT}(x \ (II) \ (y \ YI))$



Approximant

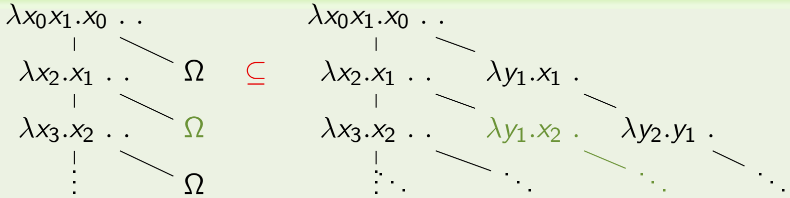
Definition of the approximation $M \subseteq_{\mathbf{BT}} N$

M is an approximant of N if $\mathbf{BT}(M)$ is a subtree of $\mathbf{BT}(N)$ with truncated edges replaced by Ω's. It if the largest relation st. :

- $\Omega \subseteq V$ for all V
- If for all $i \leq k$, $U_i \subseteq V_i$, then

$$(\lambda x_1 ... x_n . y\ U_1 \cdots U_k) \subseteq (\lambda x_1 ... x_n . y\ V_1 \cdots V_k).$$

Example $Y(\lambda uxy.x(uy)\Omega) \subseteq_{\mathbf{BT}} Y(\lambda uxy.x(uy)(Jx))$



Approximation theorem

Definition of the property

It is when terms are interpreted as the sup of the interpretation of its finite approximants (approximants with finite Böhm trees) :

$$[\![M]\!]_D = \bigcup_{\substack{\mathbf{BT}(N)\subseteq\mathbf{BT}(M)\\ \mathbf{BT}(N)\ \text{finite}}} [\![N]\!]_D$$

Conjecture

Any fully abstract K-model respect the approximation theorem.

Böhm trees and tests

Typing of $\vdash J : \mu \to \mu$

$$\vdash J : \mu \to \mu$$

Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

Derivation of $\tau_{\mu \to \mu}(J)$

$\tau_{\mu \to \mu}(J)$

Böhm tree

$J^{\mu \to \mu}$.

Böhm trees and tests

Typing of $\vdash J : \mu{\to}\mu$

$$\frac{\vdash \lambda xy.x\ (J\ y) : \mu{\to}\mu}{\vdash J : \mu{\to}\mu}$$

Böhm tree

$$(\lambda xy.x\ (J\ y))^{\mu\to\mu}\ .$$

Completion of $\{*, \mu\}$

$$* = \emptyset{\to}* \qquad \mu = \{*\}{\to}*$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\tau_{\mu\to\mu}(J)\ \to^*\ \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y))$$

Böhm trees and tests

Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{x:\mu, y:*, \vdash x\ (J\ y) : *}{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}}{\vdash J : \mu \to \mu}$$

Böhm tree

$\lambda x^{\mu} y^{*}.(x\ (J\ y))^{*}$.

Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned} \tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\ &\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \end{aligned}$$

Böhm trees and tests

Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{y : * \vdash J\ y : * \qquad * \leq *}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}}{\vdash J : \mu \to \mu}$$

Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

Derivation of $\tau_{\mu \to \mu}(J)$

$$\begin{aligned} \tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\ &\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*)) \\ &\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*))) \end{aligned}$$

Böhm tree

$$\lambda x^\mu y^*.x^\mu\ (J\ y)^*\ .$$

Böhm trees and tests

Typing of $\vdash J : \mu\to\mu$

$$\dfrac{\dfrac{\dfrac{y : * \vdash J\ y : * \qquad \overline{* \leq *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu\to\mu}}{\vdash J : \mu\to\mu}$$

Completion of $\{*, \mu\}$

$$* = \emptyset\to* \qquad \mu = \{*\}\to*$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}\tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\ &\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*)) \\ &\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*))) \\ &\to \tau_*(J\ \bar{\epsilon}_*)\end{aligned}$$

Böhm tree

$$\lambda x^\mu y^*.x^\mu\ (J\ y)^*\ .$$

Böhm trees and tests

Typing of $\vdash J : \mu\to\mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{y : * \vdash \lambda z.y\ (J\ z) : *}{y : * \vdash J\ y : *} \qquad \dfrac{}{* \leq *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu\to\mu}}{\vdash J : \mu\to\mu}$$

Böhm tree

$$\lambda x^{\mu} y^{*}.x^{\mu}$$
$$|$$
$$(\lambda z.y\ (J\ z))^{*}$$

Completion of $\{*, \mu\}$

$$* = \emptyset\to* \qquad \mu = \{*\}\to*$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\epsilon}_{*\ (Jy)})
\end{aligned}$$

Böhm trees and tests

Typing of $\vdash J : \mu\to\mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{y:*,z:\emptyset \vdash y\ (J\ z) : *}{y:*\vdash \lambda z.y\ (J\ z) : *}}{y:*\vdash J\ y : *} \quad \dfrac{}{* \leq *}}{x:\mu, y:*, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu\to\mu}}{\vdash J : \mu\to\mu}$$

Completion of $\{*, \mu\}$

$$* = \emptyset\to* \qquad \mu = \{*\}\to*$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*)) \\
&\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*))) \\
&\to \tau_*(J\ \bar{\epsilon}_*) \\
&\to^* \tau_*(\lambda y.\bar{\epsilon}_{*}\ {}_{(Jy)}) \\
&\to \tau_*(\bar{\epsilon}_*\ (J\Omega))
\end{aligned}$$

Böhm tree

$$\lambda x^\mu y^*.x^\mu\ .$$
$$|$$
$$\lambda z^\emptyset.(y\ (J\ z))^*$$

Böhm trees and tests

Typing of $\vdash J : \mu{\to}\mu$

$$* \leq *$$

$$y : *, z : \emptyset \vdash y\ (J\ z) : *$$

$$y : * \vdash \lambda z.y\ (J\ z) : *$$

$$y : * \vdash J\ y : * \qquad * \leq *$$

$$x : \mu, y : *, \vdash x\ (J\ y) : *$$

$$\vdash \lambda xy.x\ (J\ y) : \mu{\to}\mu$$

$$\vdash J : \mu{\to}\mu$$

Böhm tree

$$\lambda x^{\mu} y^{*}.x^{\mu}\ .$$

$$|$$

$$\lambda z^{\emptyset}.y^{*}\ (J\ z)^{\emptyset}$$

Completion of $\{*, \mu\}$

$$* = \emptyset{\to}* \qquad \mu = \{*\}{\to}*$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\boldsymbol{\epsilon}}_{*\ (Jy)}) \\
&\to \tau_{*}(\bar{\boldsymbol{\epsilon}}_{*}\ (J\Omega)) \\
&\to \tau_{*}(\bar{\boldsymbol{\epsilon}}_{*})
\end{aligned}$$

Böhm trees and tests

Typing of $\vdash J : \mu{\to}\mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{* \leq *}}{y : *, z : \emptyset \vdash y\ (J\ z) : *}}{y : * \vdash \lambda z.y\ (J\ z) : *}}{y : * \vdash J\ y : *} \quad \overline{* \leq *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu{\to}\mu}}{\vdash J : \mu{\to}\mu}$$

Böhm tree

$$\lambda x^{\mu} y^{*}.x^{\mu}\ .$$

$$|$$

$$\lambda z^{\emptyset}.y^{*}\ (J\ z)^{\emptyset}$$

Completion of $\{*, \mu\}$

$$* = \emptyset{\to}* \qquad \mu = \{*\}{\to}*$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\boldsymbol{\epsilon}}_{*\ (Jy)}) \\
&\to \tau_{*}(\bar{\boldsymbol{\epsilon}}_{*}\ (J\Omega)) \\
&\to \tau_{*}(\bar{\boldsymbol{\epsilon}}_{*}) \\
&\to \epsilon
\end{aligned}$$

Böhm trees and tests

Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{* \leq *}}{y : *, z : \emptyset \vdash y\ (J\ z) : *}}{y : * \vdash \lambda z.y\ (J\ z) : *}}{y : * \vdash J\ y : *} \qquad \dfrac{}{* \leq *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\dfrac{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}{\vdash J : \mu \to \mu}}$$

Böhm tree

$$\lambda x^{\mu} y^{*}.x^{\mu}\ .$$

$$|$$

$$\lambda z^{\emptyset}.y^{*}\ \Omega^{\emptyset}$$

Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\epsilon}_{*\ (Jy)}) \\
&\to \tau_{*}(\bar{\epsilon}_{*}\ (J\Omega)) \\
&\to \tau_{*}(\bar{\epsilon}_{*}) \\
&\to \epsilon
\end{aligned}$$

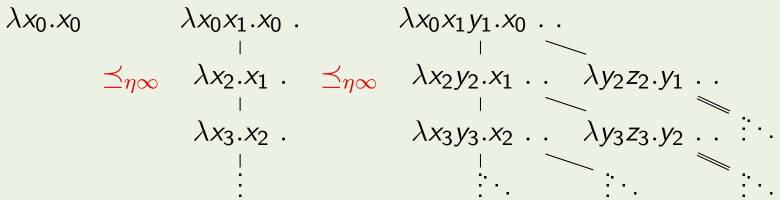
$\eta\infty$-order

Definition of the order $\preceq_{\eta\infty}$

We said $M \preceq_{\eta\infty} N$ if **BT**(N) is the result of infinitely many η-expansions on **BT**(M). Or co-inductively $M \preceq_{\eta\infty} N$ if :

- either M and N diverges,
- or $N \to_h^* \lambda x_1 ... x_n.y\ N_1 \cdots N_k$ and there is $\lambda x_1 ... x_n.y\ M_1 \cdots M_k \succeq_\eta M$ such that $N_i \preceq_{\eta\infty} M_i$ for all $i \leq k$.

Example : $I \ \preceq_{\eta\infty}\ J\ \preceq_{\eta\infty}\ Y\ (\lambda uxyz.x\ (uy)\ (uz))$



Full abstraction and $\eta\infty$-order

Theorem [Barendregt]

$M \equiv_o N$ iff they are $\eta\infty$-bounded.

Example : $J \;\; \equiv_o \;\; Y\ (\lambda uxyz.x\ y\ (uz))$

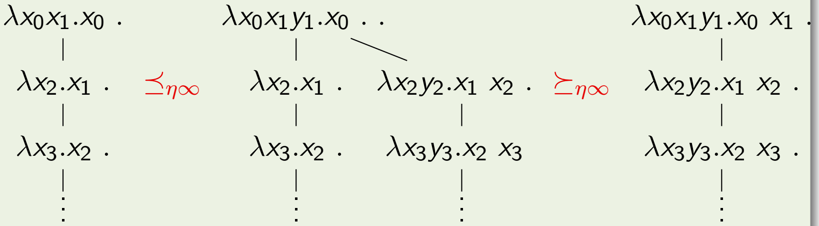


Table of contents

Full abstraction imply Hyperimmunity

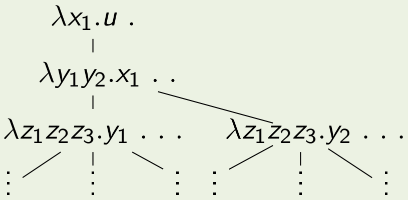
Lemma : The $\eta\infty$-extension of I along g

For any recursive g, there is a term J_g such that :

$$\mathbf{BT}(J_g\ u) = \lambda x_1 \ldots x_{g(1)}.u\ \mathbf{BT}(J_h\ x_1)\ \mathbf{BT}(J_h\ x_{g(1)})$$

With $h = n \mapsto g(n+1)$

$J_{id}\ u$



$J_g \equiv_o I$

Because $I \preceq_{\eta\infty} J_g$:
By coinduction.

Non-hyperim. $\Rightarrow [\![I]\!] \neq [\![J_g]\!]$

If $(g,(\alpha_n)_n)$ refutes hyperim.,
then $\alpha_0 \to \alpha_0 \notin [\![J_g]\!]$

Full abstraction imply Hyperimmunity

$$\alpha_1 = a_{1,1} \to \cdots a_{1,i_1} \cdots \to a_{1,g(1)} \to \alpha'_1$$

$$\Cup$$

$$\alpha_2 = a_{2,1} \to \cdots a_{2,i_2} \cdots \to a_{2,g(2)} \to \alpha'_2$$

$$\Cup$$

$$\alpha_3 = a_{3,1} \to \cdots a_{3,i_3} \cdots \to a_{3,g(3)} \to \alpha'_3$$

$$\ddots$$

$$J_g\, u = \lambda x_{1,1} \ldots x_{1,g(1)}.u \;\; \cdot 1 \;\; \cdots\cdots \;\; \cdot i_1 \;\; \cdots\cdots \;\; \cdot g(1)$$

$$\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;|\; \vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots$$

$$\lambda x_{2,1} \ldots x_{2,g(2)}.x_{1,i_1} \;\; \cdot 1 \;\; \cdots\cdots \;\; \cdot i_2 \;\; \cdots\cdots \;\; \cdot g(2)$$

$$\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;|\; \vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots \;\vdots$$

$$\lambda x_{3,1} \ldots x_{3,g(3)}.x_{2,i_2} \;\; \cdot 1 \;\; \cdots\cdots \;\; \cdot i_3 \;\; \cdots\cdots \;\; \cdot g(3)$$

$$\ddots$$

Full abstraction imply Hyperimmunity

$$(J_g\, u^{\alpha_1})^{\alpha_1} = \lambda x_{1,1}^{a_{1,1}} \ldots x_{1,g(1)}^{a_{1,g(1)}}.(u^{\alpha_1}\ .^{a_{1,1}} \cdots\cdots .^{a_{1,i_1}} \cdots\cdots .^{a_{1,g(1)}})^{\alpha'_1}$$

$$\vdots \qquad | \qquad \vdots$$

$$\lambda x_{2,1}^{a_{2,1}} \ldots x_{2,g(2)}^{a_{1,g(2)}}.(x_{1,1}^{\alpha_2}\ .^{a_{2,1}} \cdots\cdots .^{a_{2,i_2}} \cdots\cdots .^{a_{2,g(2)}})^{\alpha'_2}$$

$$\vdots \qquad | \qquad \vdots$$

$$\lambda x_{3,1}^{a_{3,1}} \ldots x_{3,g(3)}^{a_{3,g(3)}}.(x_{2,i_2}^{\alpha_3}\ .^{a_{3,1}} \cdots\cdots .^{a_{3,i_3}} \cdots\cdots .^{a_{3,g(3)}})^{\alpha'_3}$$

$$\vdots$$

Hyperimmunity imply full abstraction

If D is hyperimmune and respect the approximation theorem then :

Adequation

$$[\![M]\!] \subseteq [\![N]\!] \quad \Rightarrow \quad M \sqsubseteq_o N$$

Completeness

$$\begin{aligned} [\![M]\!] \not\subseteq [\![N]\!] \quad &\Rightarrow \quad \exists \alpha \in [\![M]\!] - [\![N]\!] \\ &\Rightarrow \quad \exists \alpha, \tau_\alpha(M)\Downarrow \text{ and } \tau_\alpha(N)\Uparrow \\ &\Rightarrow \quad M \not\sqsubseteq_o N \end{aligned}$$

Key lemma : $M = \lambda x.x$

$$\tau_{\alpha\to\alpha}(N)\Uparrow \;\Rightarrow\; \lambda x.x \not\sqsubseteq_o N$$

Proof : if $\lambda x.x \sqsubseteq_o N$ not hyp : co-inductively construct $(\alpha_n)_n$

Principal proof

$$\tau_\alpha(M)\Downarrow,\ \tau_\alpha(N)\Uparrow \quad \Rightarrow \quad M \not\sqsubseteq_o N$$

Proof : By induction on the derivation of $\tau_\alpha(M)$.

Conclusion and questions

Theorem

For all extensional K-models D, the following are equivalent when it respects approximation theorem :

- D is hyperimmune,
- D is inequationally fully abstract for $\mathcal{H}^*$,
- D is fully abstract for $\mathcal{H}^*$.

- Is that a consequence of the expressiveness of K-models ?
  $\rightsquigarrow$ There is $2^{\aleph_0}$ λ-theories between $\mathcal{H}$ and $\mathcal{H}^*$...
- A link with the range property for $\mathcal{H}^*$?
  $\rightsquigarrow$ The uses of recursivity is reminiscent of [Polonsky 2012]

Goal of tests

We need some control over the assertion $\alpha \in [\![M]\!]$

$$\tau_\alpha(M)\Downarrow \Leftrightarrow \alpha \in [\![M]\!]$$

We want to define prime elements of D

$$[\![\bar{\epsilon}_\alpha]\!] = \downarrow \alpha$$

...

$$\tau_\alpha(M[\bar{\epsilon}_{a_i}/x_i \mid i])\Downarrow \quad \Leftrightarrow \quad (a_1 \ldots a_n, \alpha) \in [\![M]\!]^{x_1 \ldots x_n}$$

$$[\![\bar{\tau}_\alpha(Q)]\!]^{x_1 \ldots x_n} = \downarrow (a_1 \ldots a_n, \alpha) \quad \Leftrightarrow \quad Q[\bar{\epsilon}_{a_i}/x_i \mid i]\Downarrow$$

Goal of tests

We need some control over the assertion $\alpha \in [\![M]\!]$

$$\tau_\alpha(M)\Downarrow \Leftrightarrow \alpha \in [\![M]\!]$$

We want to define prime elements of D

$$[\![\bar{\tau}_\alpha(Q)]\!] = \emptyset \quad \text{if } Q\Uparrow$$
$$[\![\bar{\tau}_\alpha(Q)]\!] = \downarrow\alpha \text{ if } Q\Downarrow$$

...

$$\tau_\alpha(M[\bar{\epsilon}_{a_i}/x_i \mid i])\Downarrow \quad \Leftrightarrow \quad (a_1 ... a_n, \alpha) \in [\![M]\!]^{x_1 ... x_n}$$
$$[\![\bar{\tau}_\alpha(Q)]\!]^{x_1 ... x_n} = \downarrow (a_1 ... a_n, \alpha) \quad \Leftrightarrow \quad Q[\bar{\epsilon}_{a_i}/x_i \mid i]\Downarrow$$

Hyperimmunity imply full abstraction (alternative proof)

Two interpretations of Böhm trees

Böhm trees are basically infinite terms, thus can be interpreted in D like terms modulo a fixpoint, in particular :

- by a least fixpoint : the inductive interpretation $[\![U]\!]_{ind} = \bigcup_{\substack{V \subseteq U \\ V \text{ finite}}} [\![V]\!]$,
- by a greatest fixpoint : the coinductive interpretation.

Example in D^*_∞ : $p \to p \in [\![\mathbf{BT}(J)]\!]_{coind}$

—*shadow*

—*shadow*

Coapproximation property

$[\![M]\!] = [\![\mathbf{BT}(M)]\!]_{coind}$

Step 1

Coapproximation
$+$ extensionality
$\Rightarrow$ full abstraction

Step 2

Approximation
$+$ Hyperimmunity
$\Rightarrow$ Coapproximation