Sensibility, Approximability and Reducibility (Updated)

Flavien BREUVART

LIPN, University Paris 13

HOR: 2018 July 7

General motivations/philosophy

General subject:

Denotational semantics of untype λ-calculus.

Why looking at denotational semantics for untyped λ-calculus?

For Most People:

To show denotational semantics efficiency to treat functional programming related problems.

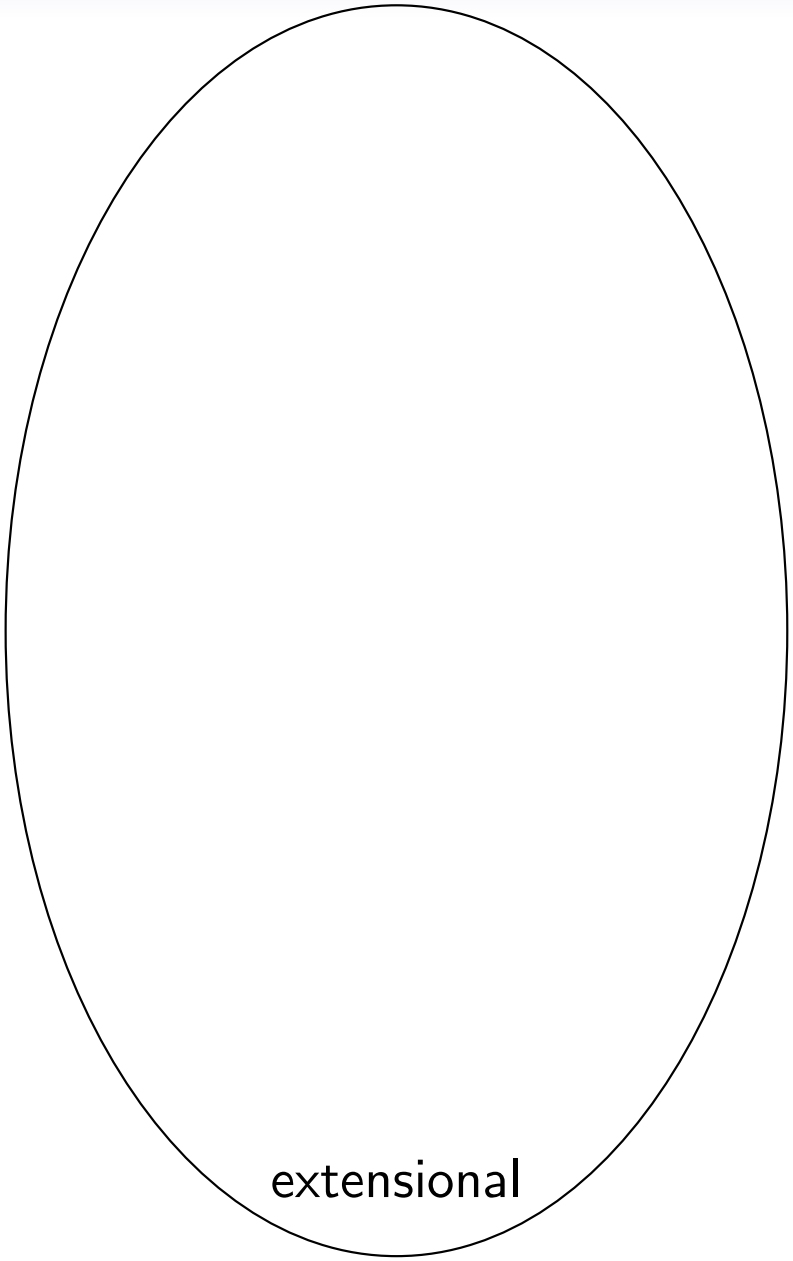
For Me:

~~To catch all Scottemons.~~
To perform a limitation annalysis on our models and technics.

Ultimate objective:

Understand realisability/reducibility at a deeper level.

A map of the land of Λ-models



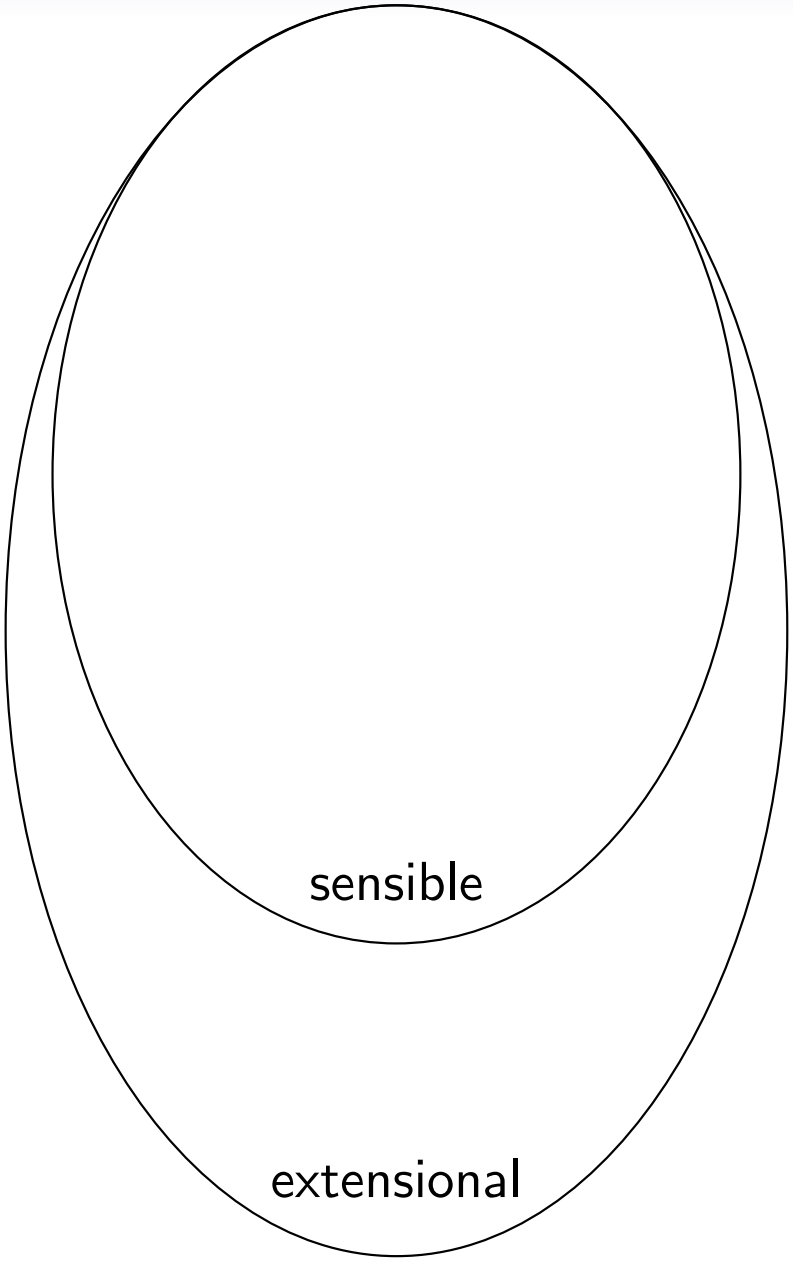
In this talk we only consider extensional models

Extensionality:
$[\![\lambda x.M\ x]\!] = [\![M]\!]$

(In this work, extensionality is used only as an arbitrary constraint forcing more structure to the models.)

Key:
◯ : set of models
respecting some property

A map of the land of Λ-models



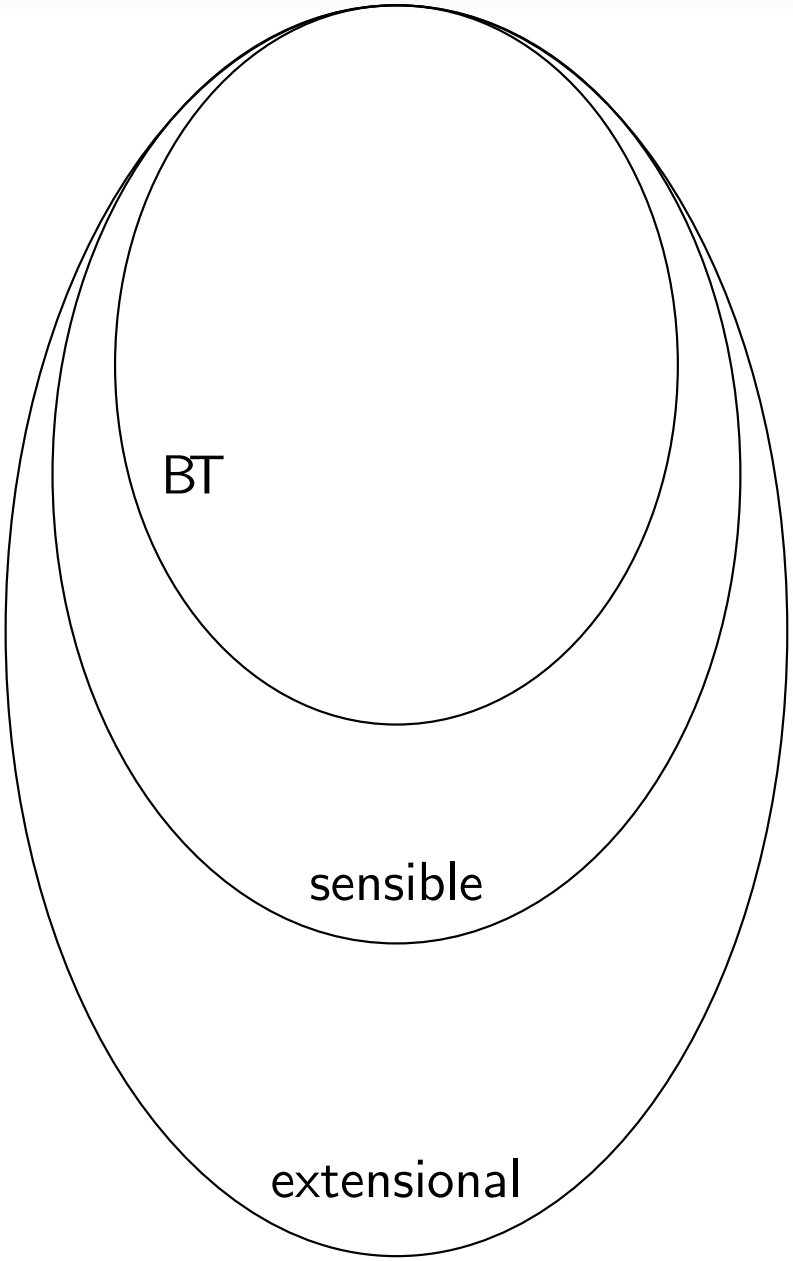
Sensibility:
M, N head diverges $\Rightarrow [\![M]\!] = [\![N]\!]$

The most fundamental and least understood property of λ-models.

(warning: there is sensible non-extensional models, but we are not considering them in this talk.)

Key:
◯ : set of models
respecting some property

A map of the land of Λ-models

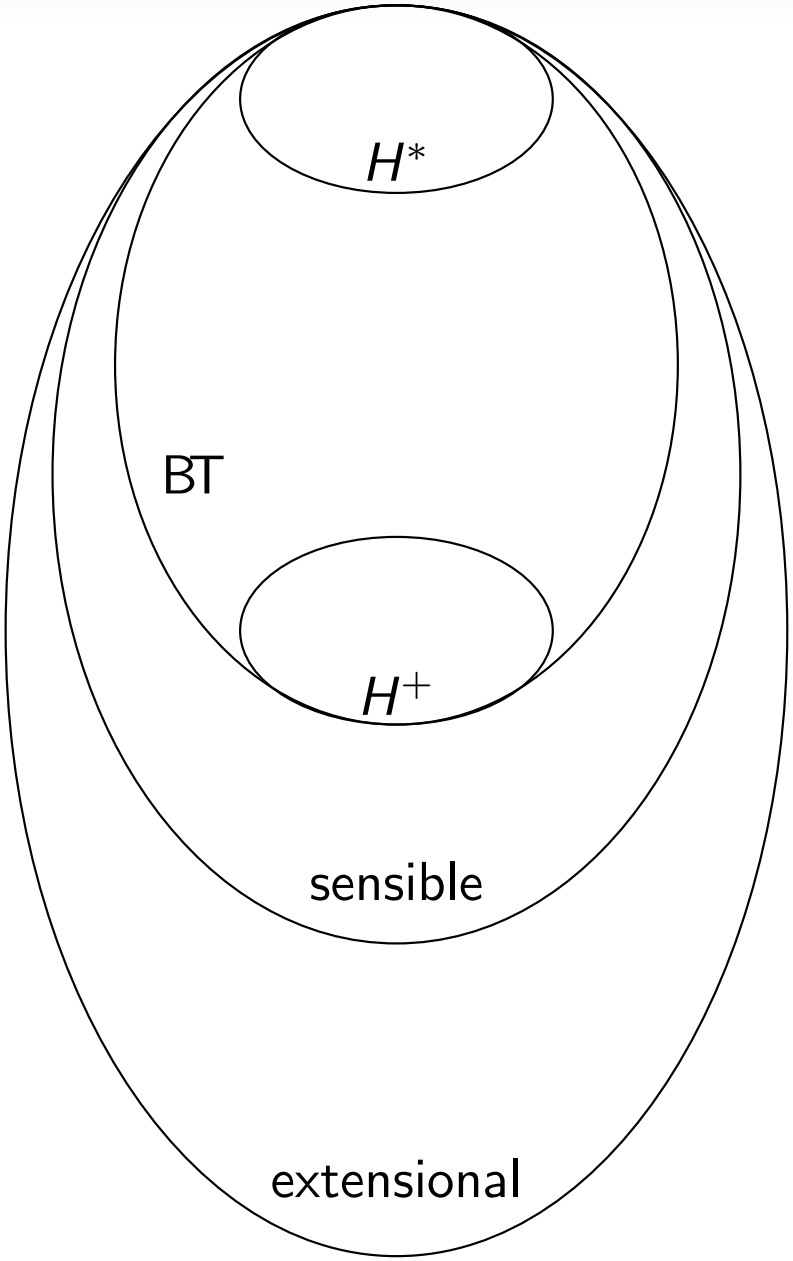


BT, or adequat for Böhm Trees :
$BT(M) = BT(N) \Rightarrow [\![M]\!] = [\![N]\!]$

where $BT(M)$ is the set of Böhm approximants of M
(given by the application-free truncatures of reducts from M)

Key:
◯ : set of models
respecting some property

A map of the land of Λ-models



corresponding to specific λ-theory
$\mathcal{T} \vdash M \equiv N \Leftrightarrow [\![M]\!] = [\![N]\!]$

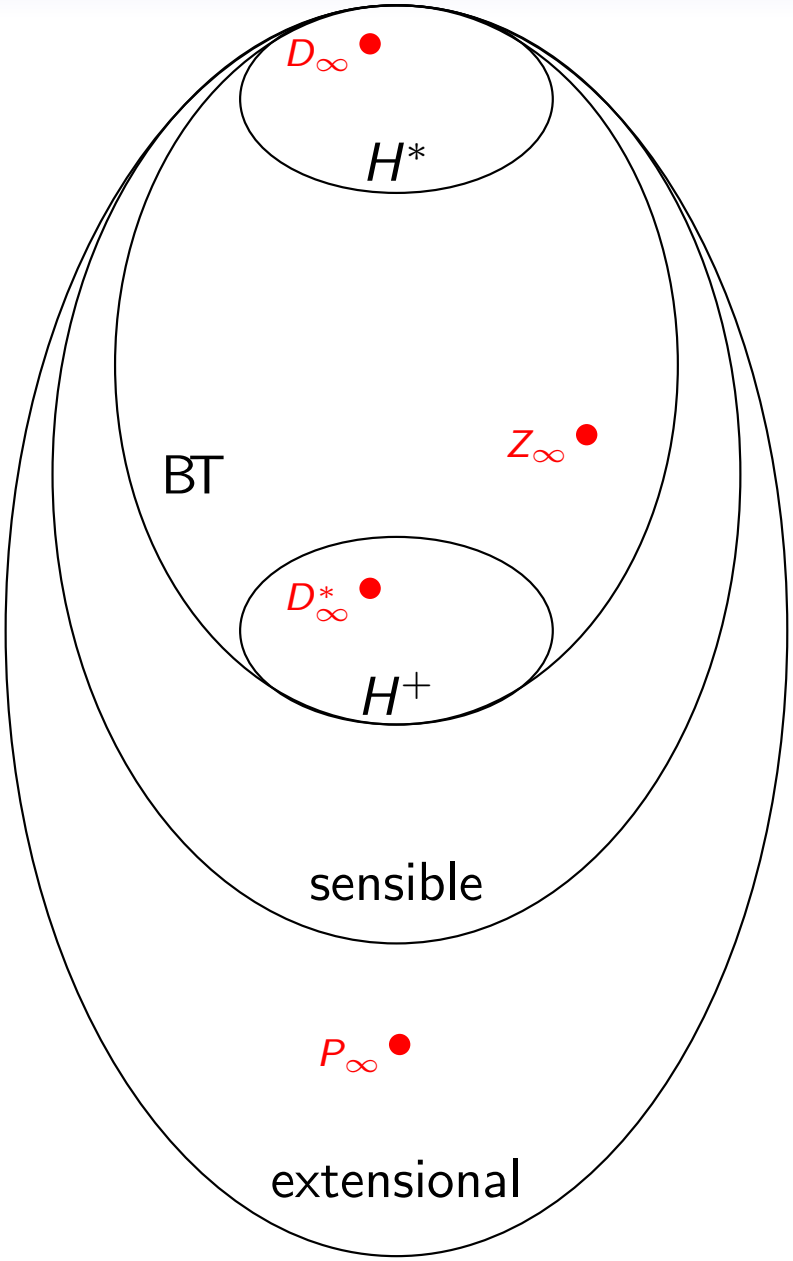
λ-theory: Congruence on lambda terms respecting β

e.g.: H^+ and H^*, the coarsest and leanest extensional BT theories

There is infinitly many non-isomorphic models coresponding to the same λ-theory

Key:
◯ : set of models
respecting some property

A map of the land of Λ-models



Some examples of known models

Scott's D_∞ : first model proven fuly abstract for H^*

CDZ's D^*_∞ : first model proven fuly abstract for H^+

Park's P_∞ : standard example of non-sensible model.

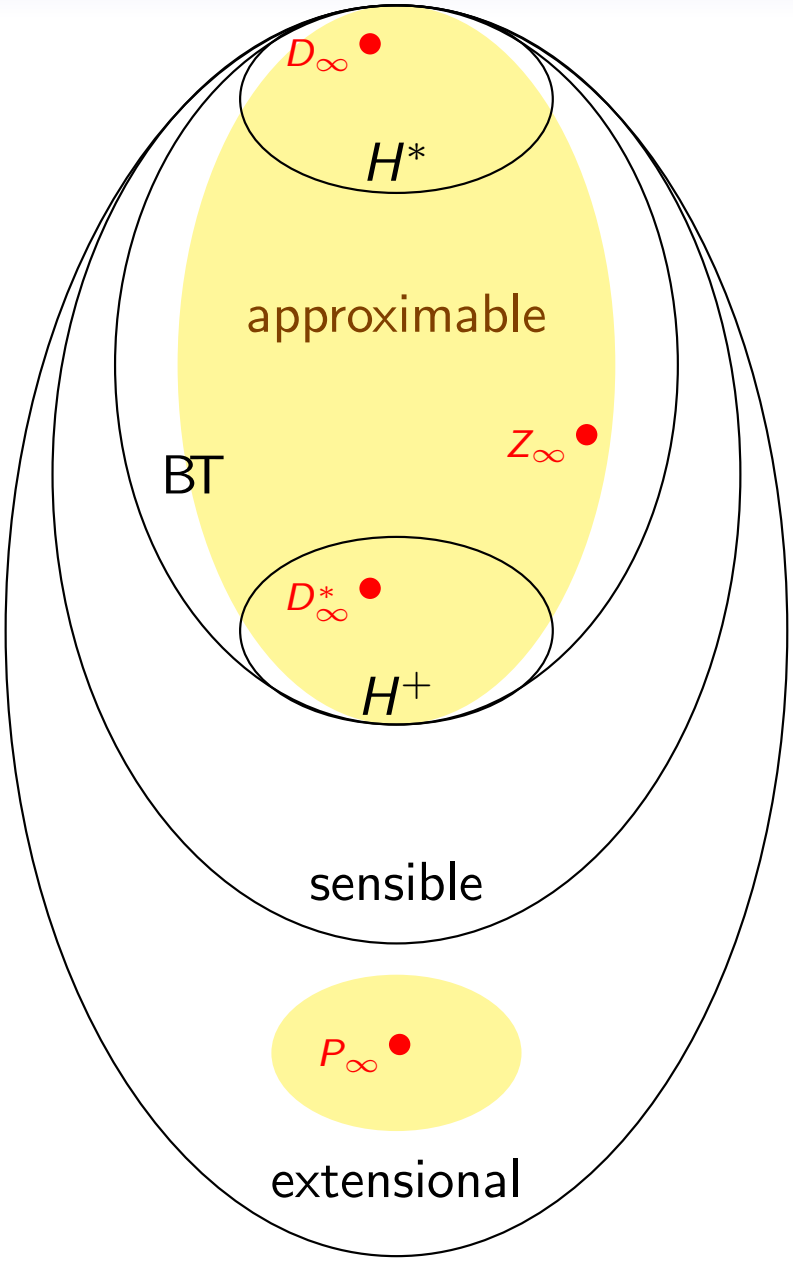
Park's Z_∞ : a coinductive model we will come back to.

Key:

◯ : set of models respecting some property

• : Particular models

A map of the land of Λ-models



Question 1:
Where are situated the models that we have been studied ?

Our Answer: All studied models so far fall in two categories, approximable models and non-sensible models obtained by forcing.

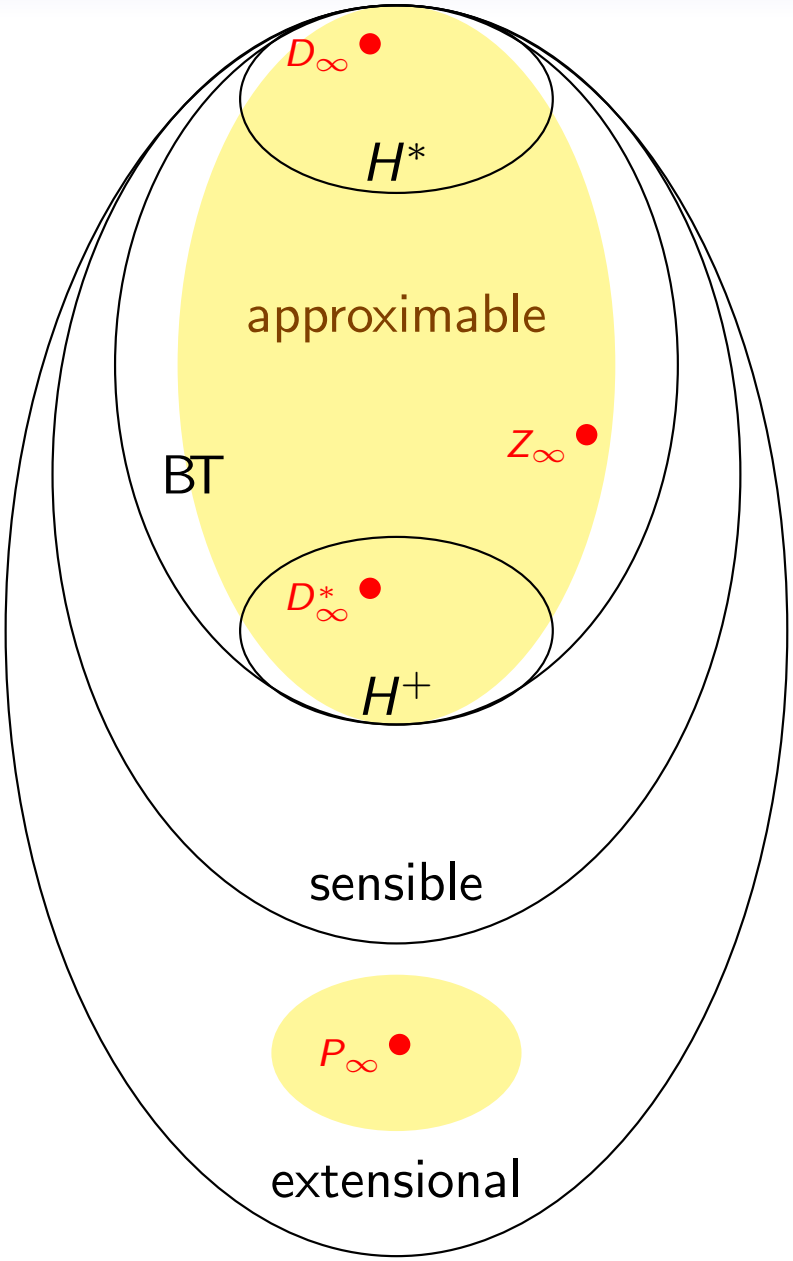
(The second are left out, but they are very interesting, in marticular tey can equate easy terms)

Key:
◯ : set of models respecting some property

• : Particular models

▭ : models we know a bit

A map of the land of Λ-models



Approximability
(or approximation theorem):

$$[\![M]\!] = \bigcup_{t \in BT(M)} [\![t]\!]$$

Notice that approximable implies BT, but, a priori, BT does not imply approximable

BT, or adequat for Böhm Trees :
$BT(M) = BT(N) \Rightarrow [\![M]\!] = [\![\mathrm{N}]\!]$

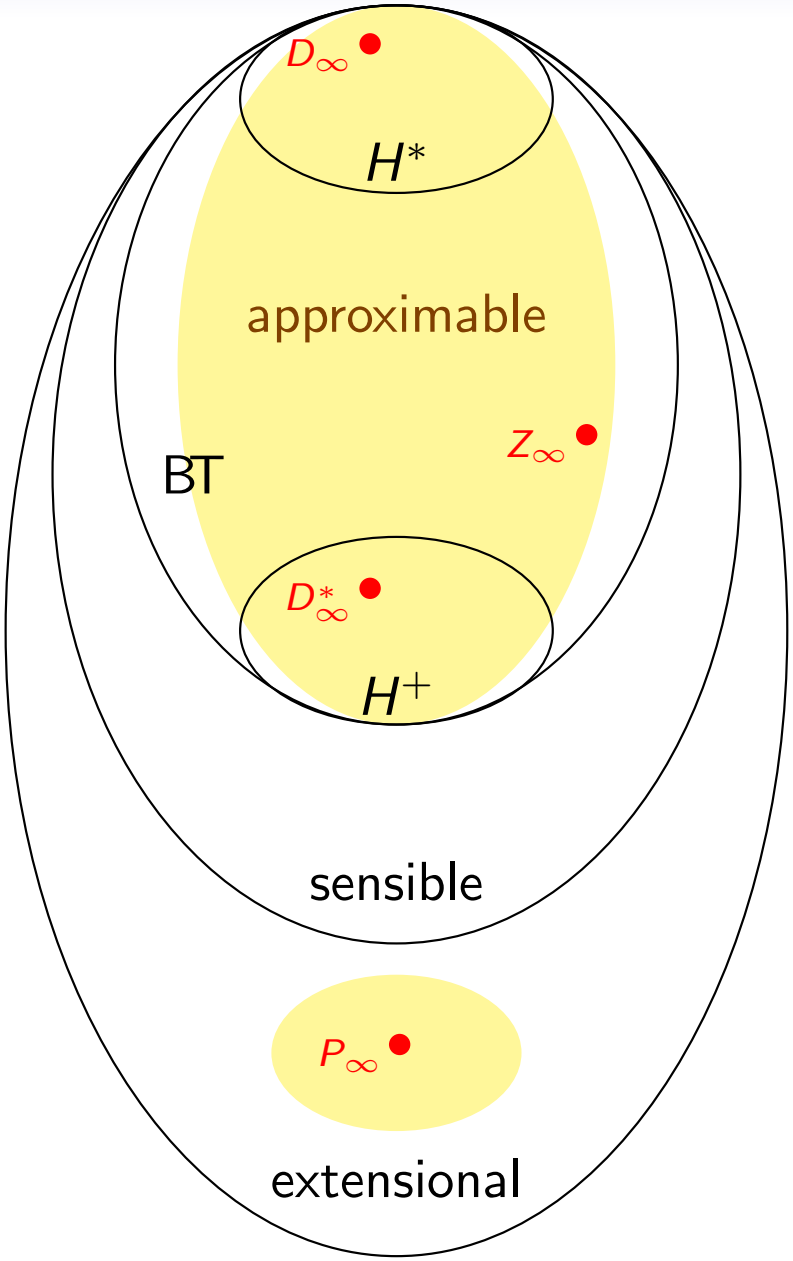
Key:

◯ : set of models respecting some property

● : Particular models

▭ : models we know a bit

A map of the land of Λ-models



Question 2:
Why did we not find any sensible but non approximable model ?

Our Answer: Because standard proofs of sensiblility can be extended to proof of approximability.

(We will see that approximability corresponds to sensibility of an extension of the λ-calculus called λ-calculus with tests.)

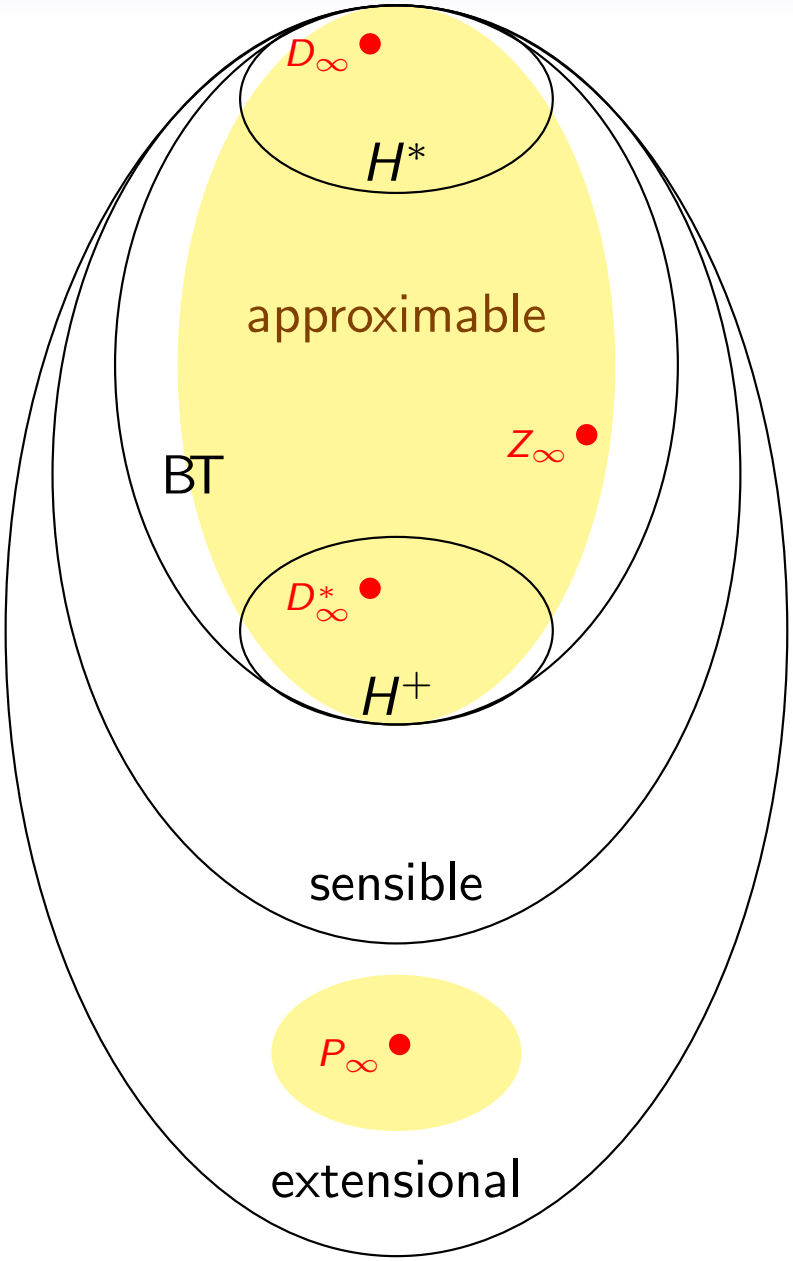
Key:

◯ : set of models respecting some property

● : Particular models

▭ : models we know a bit

A map of the land of Λ-models



An other argument is that approximable models are easier to deal with

[B.14] Characterization of approximable H^* models

[B,Manzoneto,Polonsky,Ruopulo.16] Characterization of approximable H^+ models

Key:

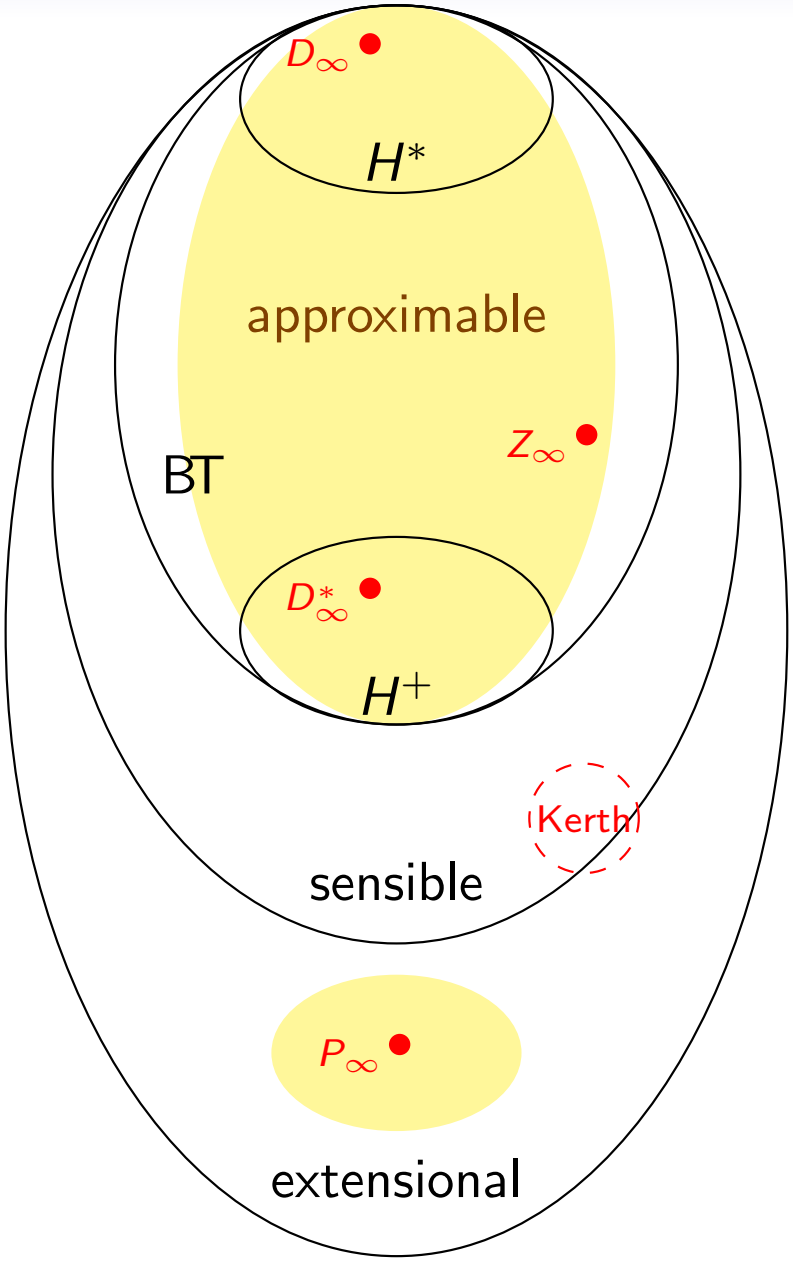
◯ : set of models respecting some property

● : Particular models

▒ : models we know a bit

▒ : limit we can characterize

A map of the land of Λ-models



Question 3: Is there any sensible non-approximable models ?

(This question only make sens for a particular class of models.)

Previous answer :

- Relational models are all approximable
- Kerth gave a non BT model conjectured sensible.

Key:

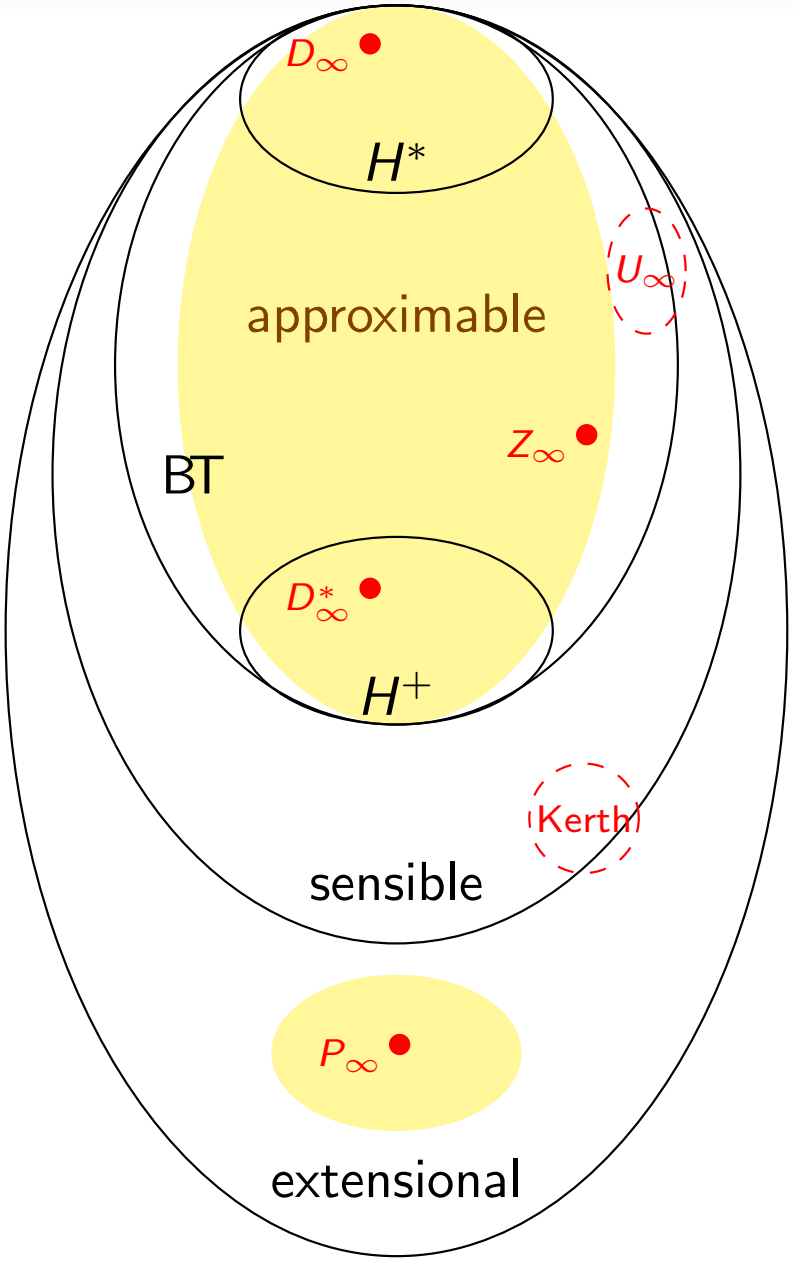
◯ : set of models respecting some property

● : Particular models

: models we know a bit

: limit we can characterize

A map of the land of Λ-models



Our answer (not in the paper)

Non well founded approximable models: An extension of previous proofs but still working on approximable models.

A non-approximable sensible model: U_∞ is proven non approximable and sensible, it may even be H^*.

Key:

◯ : set of models respecting some property

● : Particular models

: models we know a bit

: limit we can characterize

Our tools: Tests

Our Subject: Realisability

U_∞

Conclusion

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:

$$\begin{tikzcd} D \arrow[r, bend left, "\mathrm{abs}"] & D \Rightarrow D \arrow[l, bend left, "\mathrm{app}"] \end{tikzcd}$$

$$abs \circ app = id_{D \Rightarrow D}$$

Definition of Filter models

Distributive extensional filter models (DEFiM):

Pointed meet-semilattice $D = (|D|, \wedge, \leq, \omega)$,

Closure: binary operation $\rightarrow$ on D such that

$$\gamma \rightarrow \delta \geq_D \bigwedge_i \alpha_i \rightarrow \beta_i \qquad \Leftrightarrow \qquad \delta \geq_D \bigwedge_{\{i \mid \gamma \leq \alpha_i\}} \beta_i,$$

Extensionality: $\mathbf{ext}_D : D \rightarrow \mathcal{P}_f(D \times D)$ such that $\alpha = \bigwedge_{(\beta,\gamma) \in \mathbf{ext}_D(\alpha)} \beta \rightarrow \gamma$,

Distributivity: $\forall \alpha \geq \beta \wedge \gamma,\ \exists \beta' \geq_D \beta,\ \exists \gamma' \geq_D \gamma,\ \ \alpha = \beta' \wedge \gamma'$.

- Stone dual of Scott domains,
- Contain historical models: D_∞, P_∞, D^*_∞...
- Can be seen as intersection type systems.

Examples (D_∞)

Scott's D_∞ :

$$E_0 = \{*\}$$
$$E_{n+1} = E_n \cup (D_n \times D_n) - \{(\emptyset, *), (e, \emptyset) \mid e \in D\}$$
$$D_n = \mathcal{A}_f(E_n)$$
$$D_\infty = \bigcup_n D_n$$

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

$$\mathbf{ext}_D a = a \qquad a \wedge b = a \cup b \qquad \omega = \emptyset$$

The order is uniquely generated

Completion

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

Correseponds to intersection types generated by $*$ and the equation $* = \emptyset \to *$.

Examples (P_∞)

Park's P_∞ :

$$P_0 = \{*\}$$
$$P_{n+1} = P_n \cup (\mathcal{A}_f(P_n) \times P_n) - \{(\emptyset, *), (e, \emptyset) \mid e \in D\}$$
$$P_\infty = \bigcup_n P_n$$

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

Zilions of Examples

Extentional completion of simple generators :

D^*_∞ :	$\{p \leq q\}$	$p = q \to p$	$q = p \to q$
N_∞ :	$\mathbb{N}$	$\alpha_{n+1} = \alpha_n \to \alpha_n$	$\alpha_0 = \omega \to \alpha_0$
Z_∞ :	$\mathbb{Z}$	$\alpha_{n+1} = \alpha_n \to \alpha_n$	
U_∞ :	$\{*, \gamma\}$	$* = \omega \to *$	$\gamma = (\gamma \wedge (\omega \to * \to *)) \to *$

Quotient of type systems, exemple of coinductive types :

$$\alpha, \beta := \alpha \to \beta \mid X \mid \alpha \wedge \beta \mid \omega \mid \nu X.(\alpha \to \beta)$$

restricted to close and positive types and quotiented by axioms of DEFiM

Zilions of Examples

Extentional completion of simple generators :

D^*_∞ :	$\{p \leq q\}$	$p = q \to p$	$q = p \to q$
N_∞ :	$\mathbb{N}$	$\alpha_{n+1} = \alpha_n \to \alpha_n$	$\alpha_0 = \omega \to \alpha_0$
Z_∞ :	$\mathbb{Z}$	$\alpha_{n+1} = \alpha_n \to \alpha_n$	
U_∞ :	$\{*, \gamma\}$	$* = \omega \to *$	$\gamma = (\gamma \wedge (\omega \to * \to *)) \to *$

Quotient of type systems, exemple of coinductive types :

$$\alpha, \beta := \alpha \to \beta \mid X \mid \alpha \wedge \beta \mid \omega \mid \nu X.(\alpha \to \beta)$$

restricted to close and positive types and quotiented by axioms of DEFiM

No good formalism...

We will work with the firsts

Complements on filter models

- pointed meet-semilattice $D = (|D|, \wedge, \leq, \omega)$,
- closure: binary operation $\rightarrow$ on D such that

$$\gamma \rightarrow \delta \geq_D \bigwedge_i \alpha_i \rightarrow \beta_i \qquad \Leftrightarrow \qquad \delta \geq_D \bigwedge_{\{i \mid \gamma \leq \alpha_i\}} \beta_i.$$

- extensionality: $\mathbf{ext}_D : D \rightarrow \mathcal{P}_f(D \times D)$ such that

$$\alpha = \bigwedge_{(\beta,\gamma) \in \mathbf{ext}_D(\alpha)} \beta \rightarrow \gamma$$

$$\bigwedge_{(\beta',\gamma') \in I} (\beta' \rightarrow \gamma') \notin Im(\rightarrow) \qquad \text{and} \qquad \alpha \neq \bigwedge_{(\beta',\gamma') \in \mathbf{ext}_D(\alpha) - (\beta,\gamma)} \beta' \rightarrow \gamma'$$

In particular $(\beta, \omega) \in \mathbf{ext}_D(\alpha)$ implies $\alpha = \omega$, moreover $\mathbf{ext}_D(\omega) = \{(\beta, \omega)\}$ for some arbitrary β since $\beta \rightarrow \omega = \omega$.
Unfortunately, the choice of the function $\mathbf{ext}_D$ is generally not unique or even canonical. In order remove any influence from this choice, we restrict our study to <u>distributive</u> filter models.
A filter model D is distributive whenever any $\alpha \geq \beta \wedge \gamma$ is accessible in the sense that there exists a decomposition $\alpha = \beta' \wedge \gamma'$ such that $\beta' \geq_D \beta$ and $\gamma' \geq_D \gamma$.

Böhm trees

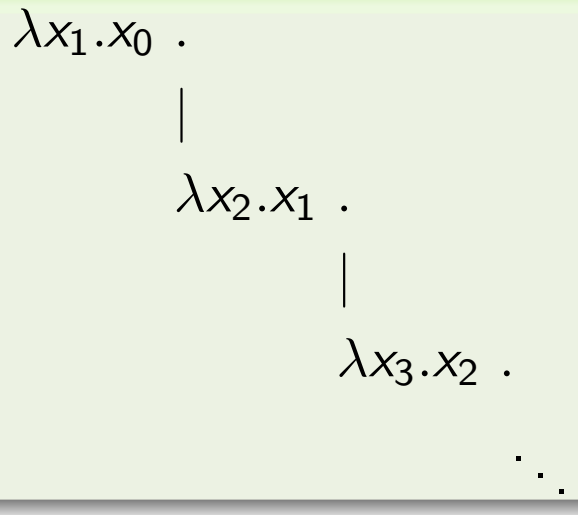
Definition of the Böhm tree $\boldsymbol{BT}(M)$ of a term M

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

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

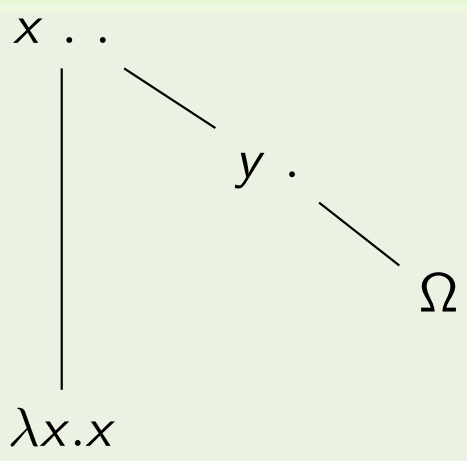
$$\boldsymbol{BT}(M) = \lambda x_1 ... x_n . y\ \boldsymbol{BT}(N_1) \cdots \boldsymbol{BT}(N_k).$$

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



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

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



Approximant

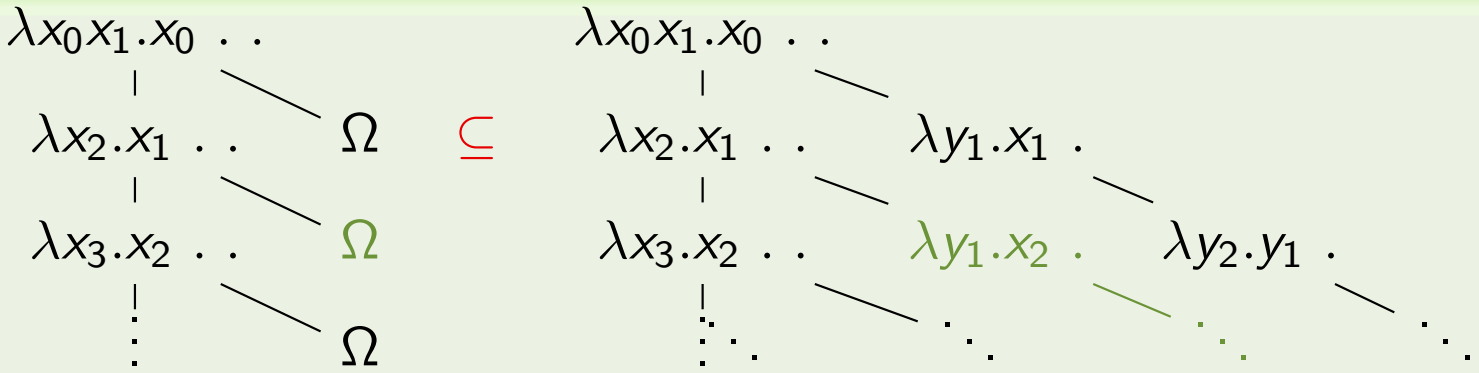
Definition of the approximation $M \subseteq_{\boldsymbol{BT}} N$

M is an approximant of N if $\boldsymbol{BT}(M)$ is a subtree of $\boldsymbol{BT}(N)$ 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 $Y(\lambda uxy.x(uy)\Omega)\ \subseteq_{\boldsymbol{BT}}\ Y(\lambda uxy.x(uy)(Jx))$



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):

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

Conjecture

Any fully abstract K-model respect the approximation theorem.

$\Lambda_{\tau(D)}$: 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:

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

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

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

...

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

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

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

...

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

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

Polarised view: tests are processes

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

Reduction strategy (extending head reduction)

$\tau(M)$: infinite application,

$\Sigma_i E_i$: may non-determinism,

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

$\Pi_i Q_i$: 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 $n = [0, n[\to 0$

Tests and typing

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

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

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 $\vdash \omega : 2$

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

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 $\vdash \omega : 2$

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

Reduction of $\tau_2(\omega)$

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

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 $\vdash \omega : 2$

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

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:

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

Infinite derivation

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

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

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

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

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

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

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

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

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 $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

Procedural of $\tau_\alpha(M)$

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

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:

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

Böhm trees and tests

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

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

Böhm tree

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

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

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

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

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

Böhm 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 $\vdash J : \mu \to \mu$

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

Böhm tree

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

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

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

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

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

Böhm trees and tests

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

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

Böhm tree

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

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

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

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

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

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 $\{*, \mu\}$

$$* = \emptyset \to *$$

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

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

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

Böhm tree

$$
\begin{array}{l}
\lambda x^{\mu} y^{*}.x^{\mu}\ . \\
\qquad\qquad | \\
\qquad\qquad \lambda z^{\emptyset}.(y\ (J\ z))^{*}
\end{array}
$$

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 trees and tests

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

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

Böhm tree

$$
\begin{array}{l}
\lambda x^{\mu} y^{*}.x^{\mu}\ . \\
\qquad\qquad | \\
\qquad\qquad \lambda z^{\emptyset}.y^{*}\ (J\ z)^{\emptyset}
\end{array}
$$

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

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

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

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

Böhm trees and tests

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

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

Böhm tree

$$
\begin{array}{l}
\lambda x^{\mu} y^{*}.x^{\mu}\ . \\
\qquad\qquad | \\
\qquad\qquad \lambda z^{\emptyset}.y^{*}\ (J\ z)^{\emptyset}
\end{array}
$$

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

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

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

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

Böhm trees and tests

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

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

Böhm tree

$$
\begin{array}{l}
\lambda x^{\mu} y^{*}.x^{\mu}\ . \\
\qquad\qquad | \\
\qquad\qquad \lambda z^{\emptyset}.y^{*}\ \Omega^{\emptyset}
\end{array}
$$

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

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

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

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