Denotational Models

Logique Linéaire et Paradigmes Logiques du Calcul

Year 2023, Part 3, Lecture 3

Giulio Manzonetto
gmanzone@irif.fr

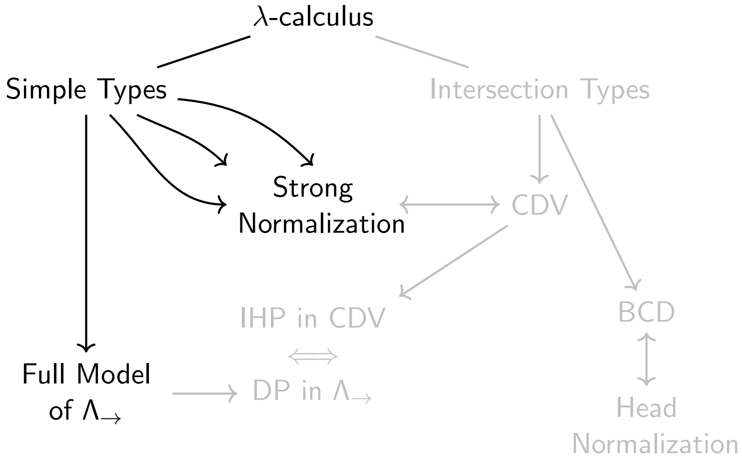
IRIF — Université Paris Cité

http://www.irif.fr/~gmanzone/

January 28, 2024

One year ago
in a galaxy far, far away. . .

Where were we?



Where were we?

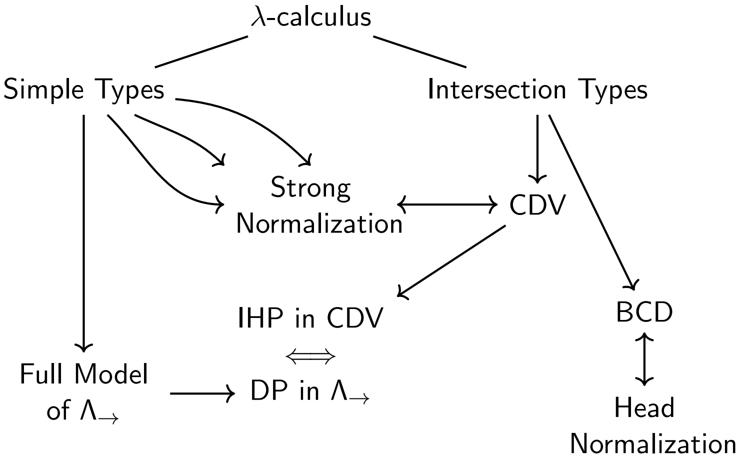


Table of contents

Solvability

A λ-term M is solvable if its closure $\lambda\vec{x}.M$ admits an applicative context $[]P_1 \cdots P_n$ that "send it" to the identity:

$$(\lambda\vec{x}.M)P_1 \cdots P_n =_\beta \mathsf{I}$$

Otherwise, we say that M is unsolvable.

Examples

- $\mathsf{K} = \lambda \mathsf{xy.x}$ is solvable: $\mathsf{KII} \to_\beta (\lambda \mathsf{y.I})\mathsf{I} \to_\beta \mathsf{I}$.
- $M = \lambda x.x\mathsf{I}\Omega$ is solvable: $M\mathsf{K} \to_\beta \mathsf{KI}\Omega \to_\beta (\lambda \mathsf{y.I})\Omega \to_\beta \mathsf{I}$.
- Ω is unsolvable, since it does not interact with any context $[]P$

$$\Omega P \to_\beta \Omega P \to_\beta \Omega P' \to_\beta \cdots$$

Solvability, equivalently

A λ-term M is solvable iff there are $P_1, \ldots, P_n \in \Lambda$ such that

$$(\lambda\vec{x}.M)P_1 \cdots P_n =_\beta H$$

for some hnf H.

Proof. ($\Rightarrow$) Trivial, since I is an hnf.
($\Leftarrow$) Assume $(\lambda\vec{x}.M)\vec{P} =_\beta H$, for some $\vec{P} \in \Lambda$ and H in hnf. Then H has shape (for some $k \geq 0$ and $1 \leq j \leq n > 0$)

$$H = \lambda y_1 \ldots y_n.y_j M_1 \cdots M_k$$

Define $\mathsf{U}^\mathsf{k} = \lambda \mathsf{x}_1 \cdots \mathsf{x}_\mathsf{k}.\mathsf{I}$ and apply it n times:

$$(\lambda\vec{x}.M)\vec{P}\,\vec{\mathsf{U}}^k \twoheadrightarrow_\beta H\,\vec{\mathsf{U}}^k \twoheadrightarrow_\beta \mathsf{U}^\mathsf{k}\mathsf{M}'_1 \cdots \mathsf{M}'_\mathsf{k} \twoheadrightarrow_\beta \mathsf{I} \qquad \square$$

Solvability, equivalently

A λ-term M is solvable iff there are $P_1, \ldots, P_n \in \Lambda$ such that

$$(\lambda\vec{x}.M)P_1 \cdots P_n =_\beta H$$

for some hnf H.

Proof. ($\Rightarrow$) Trivial, since I is an hnf.
($\Leftarrow$) Assume $(\lambda\vec{x}.M)\vec{P} =_\beta H$, for some $\vec{P} \in \Lambda$ and H in hnf. Then H has shape (for some $k \geq 0$ and $1 \leq j \leq n > 0$)

$$H = \lambda y_1 \ldots y_n.y_j M_1 \cdots M_k$$

Define $\mathrm{U}^\mathrm{k} = \lambda \mathrm{x}_1 \cdots \mathrm{x}_\mathrm{k}.\mathsf{I}$ and apply it n times:

$$(\lambda\vec{x}.M)\vec{P}\,\vec{\mathrm{U}}^k \twoheadrightarrow_\beta H\,\vec{\mathrm{U}}^k \twoheadrightarrow_\beta \mathrm{U}^\mathrm{k}\mathrm{M}'_1 \cdots \mathrm{M}'_\mathrm{k} \twoheadrightarrow_\beta \mathsf{I}$$

□

Solvability and head normalization

- A solvable M is capable of interacting with the context $(\lambda\vec{x}.[])\vec{P}$ and eventually one the P_i's goes in head position.
- An unsolvable M leaves its arguments alone, because it always have its own head redex to reduce.

Theorem (Wadsworth'76)

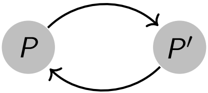
M is solvable if and only if it is head normalizable.

C.P. Wadsworth. The relation between computational and denotational properties for Scott's $\mathcal{D}_\infty$ models of the λ-calculus. SIAM J. Comput. 5,3 (1976).

Classification	Behaviour	Result
normalizable	$P \to P_1 \to P_2 \twoheadrightarrow P_{99} \to 42$	completely defined

$$P \longrightarrow P_1 \longrightarrow P_2 \cdots\cdots\twoheadrightarrow P_{102} \longrightarrow 42$$

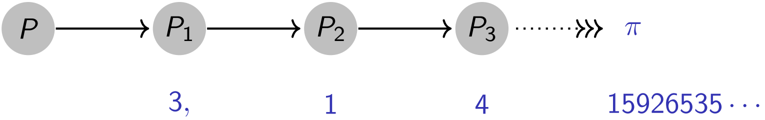
Classification	Behaviour	Result
normalizable	$P \to P_1 \to P_2 \twoheadrightarrow P_{99} \to 42$	completely defined
unsolvable	$P \to P' \to P \twoheadrightarrow_{100} P' \to \cdots$	undefined



Take $W = (\lambda xy.xyy)$, then

$WWW \rightleftarrows (\lambda x.xWW)W$

Classification	Behaviour	Result
normalizable	$P \to P_1 \to P_2 \twoheadrightarrow P_{99} \to 42$	completely defined
unsolvable	$P \to P' \to P \twoheadrightarrow_{100} P' \to \cdots$	undefined
solvable	$P \to o_1 P_1 \to o_1(o_2 P_2)$	stable parts
	$\to o_1(o_2(o_3 P_3)) \to \cdots$	
	$\twoheadrightarrow_\infty o_1(o_2(\cdots o_n \cdots)\cdots)$	(infinitary)



By collecting all stable parts one constructs a possibly infinite tree.

Playing with fixed point combinators

Let

$$\mathsf{Y} = \lambda \mathsf{f}.\Delta_{\mathsf{f}}\Delta_{\mathsf{f}},$$

with $\Delta_f = \lambda x.f(xx)$.

- Y is head-normalizing, but we can keep reducing it:

  $$\mathsf{Y} \twoheadrightarrow_\beta \lambda \mathsf{f}.\mathsf{f}(\Delta_{\mathsf{f}}\Delta_{\mathsf{f}}) \twoheadrightarrow_\beta \lambda \mathsf{f}.\mathsf{f}(\mathsf{f}(\Delta_{\mathsf{f}}\Delta_{\mathsf{f}})) \twoheadrightarrow_\beta \lambda \mathsf{f}.\mathsf{f}^{\mathsf{n}}(\Delta_{\mathsf{f}}\Delta_{\mathsf{f}}) \twoheadrightarrow_\beta \cdots$$

  The portion $\lambda f.f(f(\cdots))$ is a stable part of the output.

- YK is not head-normalizing:

  $$\mathsf{YK} \twoheadrightarrow_\beta \mathsf{K}(\Delta_{\mathsf{K}}\Delta_{\mathsf{K}}) \twoheadrightarrow_\beta \lambda \mathsf{x}_1.\Delta_{\mathsf{K}}\Delta_{\mathsf{K}} \twoheadrightarrow_\beta \lambda \mathsf{x}_1 \ldots \mathsf{x}_{\mathsf{n}}.\Delta_{\mathsf{K}}\Delta_{\mathsf{K}} \twoheadrightarrow_\beta \cdots$$

  The portion $\lambda x_1 \ldots x_n.$ is a stable part of the output, but it does not contribute to the production of a hnf. YK unsolvable.

Playing with fixed point combinators

Let

$$\mathsf{Y} = \lambda \mathsf{f}.\Delta_\mathsf{f}\Delta_\mathsf{f},$$

with $\Delta_f = \lambda x.f(xx)$.

- Y is head-normalizing, but we can keep reducing it:

  $$\mathsf{Y} \twoheadrightarrow_\beta \lambda \mathsf{f}.\mathsf{f}(\Delta_\mathsf{f}\Delta_\mathsf{f}) \twoheadrightarrow_\beta \lambda \mathsf{f}.\mathsf{f}(\mathsf{f}(\Delta_\mathsf{f}\Delta_\mathsf{f})) \twoheadrightarrow_\beta \lambda \mathsf{f}.\mathsf{f}^\mathsf{n}(\Delta_\mathsf{f}\Delta_\mathsf{f}) \twoheadrightarrow_\beta \cdots$$

  The portion $\lambda f.f(f(\cdots))$ is a stable part of the output.
- YK is not head-normalizing:

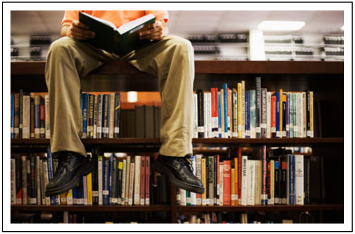
  $$\mathsf{YK} \twoheadrightarrow_\beta \mathsf{K}(\Delta_\mathsf{K}\Delta_\mathsf{K}) \twoheadrightarrow_\beta \lambda \mathsf{x}_1.\Delta_\mathsf{K}\Delta_\mathsf{K} \twoheadrightarrow_\beta \lambda \mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n}.\Delta_\mathsf{K}\Delta_\mathsf{K} \twoheadrightarrow_\beta \cdots$$

  The portion $\lambda x_1 \ldots x_n.$ is a stable part of the output, but it does not contribute to the production of a hnf. YK unsolvable.

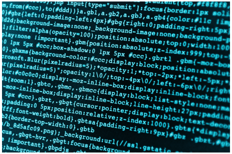
The interest of semantics

Coder

Semanticist



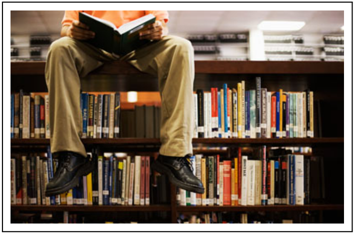
P



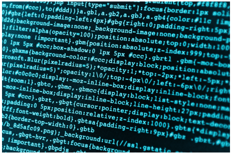
The interest of semantics

Coder

Semanticist



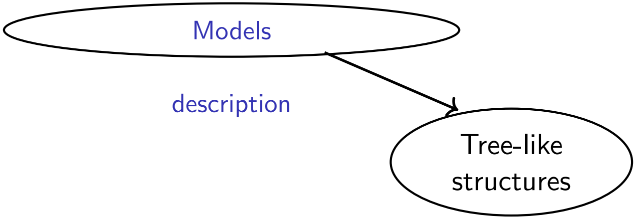
P



$$\mathcal{M} \models P = \perp$$

The Böhm Tree Semantics

Syntactic Models



The (possibly infinitary) behaviour of a λ-term
is modelled as a (possibly infinite) tree

The type free lambda calculus. (1977).
H. Barendregt. Handbook of Mathematical Logic, volume 90 of Studies in Logic and the Foundations of Mathematics.

The Böhm Tree Semantics (Barendregt '77)

Given a program M, its Böhm tree $\mathrm{BT}(M)$ is defined by:

- If M is unsolvable, then $\mathrm{BT}(M) = \bot$,
  where $\bot$ is a constant representing the undefined.
- Otherwise $M \twoheadrightarrow_\beta \lambda x_1 \ldots x_n.y\, M_1 \cdots M_k$ and

$$\mathrm{BT}(M) = \begin{array}{ccc} & \lambda x_1 \ldots x_n.y & \\ \swarrow & & \searrow \\ \mathrm{BT}(M_1) & \cdots & \mathrm{BT}(M_k) \end{array}$$

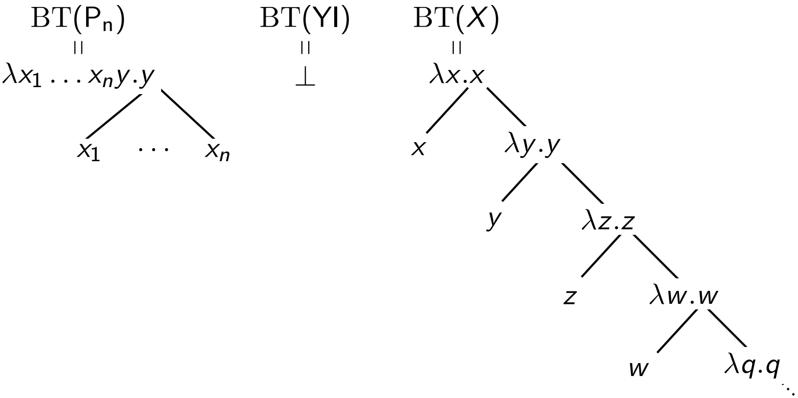
This induces an equivalence on λ-terms:

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

Example

$$\begin{array}{c} \mathrm{BT}(\mathrm{Y}) \\ \shortparallel \\ \lambda f.f \\ | \\ f \\ | \\ f \\ | \\ f \\ | \\ \vdots \end{array}$$

Böhm trees - Examples



where

- $\mathsf{P_n} = \lambda \mathsf{x_1} \ldots \mathsf{x_n y.y x_1} \cdots \mathsf{x_n}$
- $X = \mathsf{Y}(\lambda \mathsf{yx.xxy})$

The Böhm tree semantics is “infinitary”

There are λ-terms M, N with the same Böhm tree, that cannot be equated by any “finite” reduction.

1. Take a λ-term M satisfying (Its definition? Exercise!):

$$M \twoheadrightarrow_\beta \lambda zx.x(Mz)$$

2. Take a variable y. Then, $\mathrm{BT}(My) = \begin{array}{c}\lambda x.x\\ |\\ \mathrm{BT}(My)\end{array} = \begin{array}{c}\lambda x.x\\ |\\ \lambda x.x\\ |\\ \vdots\end{array}$

3. For $y \neq z$, we have $My \neq_\beta Mz$
but $\mathrm{BT}(My) = \mathrm{BT}(Mz)$.

Digression — Böhm Trees as Coinductive Data-Types

Böhm-like trees are coinductively defined by:

$$T ::=_{\text{co-ind}} \bot \mid \lambda x_1 \dots x_n.y T_1 \cdots T_k$$

Intuition

- Start from the set of all possibly infinite labelles trees.
- Throw away all trees that do not satisfy the above rules.
- E.g., $\bot\bot$, infinitely branching trees, $\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4. \dots$.

Inductive grammar	$\cong$	least fixed point
Co-inductive grammar	$\cong$	greatest fixed point

Question: do you see a non λ-definable Böhm-like tree?

Digression — Böhm Trees as normal forms

The λ^∞-calculus:

$$(\Lambda^\infty) \qquad M, N ::=_{\text{co-ind}} \perp \mid x \mid \lambda x.M \mid MN$$

with β-reduction and $\perp$-reductions: $\perp M \to_\perp \perp$ and $\lambda x.\perp \to_\perp \perp$.

Reduction sequences may now have transfinite length α (ordinal)

$$M_0 \to_{\beta\perp} M_1 \to_{\beta\perp} \cdots M_\omega \to_{\beta\perp} M_{\omega+1} \to_{\beta\perp} \cdots \twoheadrightarrow_{\beta\perp} M_\alpha$$

Theorem (Kennaway et Al.)

1. *The λ^∞-calculus is confluent.*
2. *The λ^∞-calculus enjoys strong normalization.*
3. *For all finite $M \in \Lambda$, $M \twoheadrightarrow_{\beta\perp} \mathrm{BT}(M)$.*

Digression — Böhm Trees as normal forms

The λ^∞-calculus:

$$(\Lambda^\infty) \qquad M, N ::=_{\text{co-ind}} \bot \mid x \mid \lambda x.M \mid MN$$

with β-reduction and $\bot$-reductions: $\bot M \to_\bot \bot$ and $\lambda x.\bot \to_\bot \bot$.

Reduction sequences may now have transfinite length α (ordinal)

$$M_0 \to_{\beta\bot} M_1 \to_{\beta\bot} \cdots M_\omega \to_{\beta\bot} M_{\omega+1} \to_{\beta\bot} \cdots \twoheadrightarrow\!\!\!\!\!\rightarrow_{\beta\bot} M_\alpha$$

- R. Kennaway, J.W. Klop, M. R. Sleep, F.-J. de Vries: Infinitary Lambda Calculus. Theor. Comput. Sci. 175(1): 93-125 (1997)
- Łukasz Czajka: A new coinductive confluence proof for infinitary λ-calculus. Log. Methods Comput. Sci. 16(1) (2020)

That was scary. . . can we go back to induction?

- Finite trees are pieces of “output” that can be obtained in a finite amount of time.
- Böhm trees are naturally ordered, as follows:

$$
\begin{array}{ccccccccc}
\perp & \sqsubseteq & \lambda f.f & \sqsubseteq & \lambda f.f & \sqsubseteq & \lambda f.f & \sqsubseteq & \lambda f.f \\
 & & | & & | & & | & & | \\
 & & \perp & & f & & f & & f \\
 & & & & | & & | & & | \\
 & & & & \perp & & f & & f \\
 & & & & & & | & & | \\
 & & & & & & \perp & & f \\
 & & & & & & & & | \\
 & & & & & & & & \vdots
\end{array}
$$

Lambda terms approximations

- The set $\Lambda_\perp$ of $\lambda\perp$-terms is defined as follows:

$$P, Q \ ::= \ \perp \mid x \mid PQ \mid \lambda x.P$$

- Intuively, the constant $\perp$ represents an unsolvable (Ω).
- $\to_\beta$ extends in the obvious way:

$$(\lambda x.P)Q \to_\beta P[x := Q]$$

- The $\perp$-reduction is generated by:

$$\perp M \to_\perp \perp, \qquad\qquad \lambda x.\perp \to_\perp \perp$$

Ordering $\lambda\bot$-terms

The ordering $\sqsubseteq$ on $\lambda\bot$-terms is generated by:

$$\frac{}{\bot \sqsubseteq P} \qquad \frac{P \sqsubseteq P' \quad Q \sqsubseteq Q'}{PQ \sqsubseteq P'Q'} \qquad \frac{P \sqsubseteq P'}{\lambda x.P \sqsubseteq \lambda x.P'}$$

The corresponding "sup" $P \sqcup Q$ is inductively given by:

$$\begin{array}{rcl} \bot \sqcup P & = & P \sqcup \bot = P \\ (PQ) \sqcup (P'Q') & = & (P \sqcup P')(Q \sqcup Q') \\ (\lambda x.P) \sqcup (\lambda x.P') & = & \lambda x.(P \sqcup P') \end{array}$$

(It is well-defined on "compatible" elements only.)

Finite Approximants

The set $\mathcal{A}$ of finite approximants is defined as follows:

$$(\mathcal{A}) \qquad A, A_i \;::=\; \perp \mid \lambda x_1 \ldots x_n . y\, A_1 \cdots A_k$$

Characterization. For $P \in \Lambda_\perp$, the following are equivalent:

1. $P \in \mathcal{A}$,
2. P is normal w.r.t. $\to_{\beta\perp}$.

Direct approximation

The direct approximant $\omega(M) \in \mathcal{A}$ of $M \in \Lambda$ is inductively defined:

$$\begin{aligned} \omega(\lambda x_1 \ldots x_n.yM_1 \cdots M_k) &= \lambda x_1 \ldots x_n.y\omega(M_1) \cdots \omega(M_k), \\ \omega(\lambda x_1 \ldots x_n.(\lambda y.P)QM_1 \cdots M_k) &= \bot \end{aligned}$$

Lemma. $M \to_\beta N$ implies $\omega(M) \sqsubseteq \omega(N)$.

Proof. By structural induction on M. There are two cases:

1. M has a head redex. Trivial since $\omega(M) = \bot$.
2. $M = \lambda\vec{x}.yM_1M_2 \cdots M_k$, $N = \lambda\vec{x}.yM_1'M_2 \cdots M_k$, with $M_1 \to_\beta M_1'$. From the induction hypothesis $\omega(M_1) \sqsubseteq \omega(M_1')$, thus the conclusion follows. $\square$

Finite approximants of a λ-term

The set of finite approximants of a λ-term M is defined by:

$$
\begin{aligned}
\mathcal{A}(M) &\stackrel{(1)}{=} \{\omega(N) \mid N =_\beta M\}, \text{ equivalently*,} \\
&\stackrel{(2)}{=} \{A \in \mathcal{A} \mid \exists N \in \Lambda \,.\, M \twoheadrightarrow_\beta N \text{ and } A \sqsubseteq N\} \\
&\neq \{\omega(N) \mid M \twoheadrightarrow_\beta N\}
\end{aligned}
$$

* Some work is needed to prove the equivalence (1) $\iff$ (2).

Lemma. If $M =_\beta N$ then $\mathcal{A}(M) = \mathcal{A}(N)$.

Trivial with (1), non-trivial with (2). [Hint: use confluence!] $\square$

Finite approximants of a λ-term

The set of finite approximants of a λ-term M is defined by:

$$\begin{aligned}\mathcal{A}(M) &\overset{(1)}{=} \{\omega(N) \mid N =_\beta M\}, \text{ equivalently*}, \\ &\overset{(2)}{=} \{A \in \mathcal{A} \mid \exists N \in \Lambda \,.\, M \twoheadrightarrow_\beta N \text{ and } A \sqsubseteq N\} \\ &\neq \{\omega(N) \mid M \twoheadrightarrow_\beta N\}\end{aligned}$$

* Some work is needed to prove the equivalence (1) $\iff$ (2).

Lemma. If $M =_\beta N$ then $\mathcal{A}(M) = \mathcal{A}(N)$.

Trivial with (1), non-trivial with (2). [Hint: use confluence!] $\square$

Lemma. The set $\mathcal{A}(M)$ is an ideal w.r.t. $\sqsubseteq$:

1. $\bot \in \mathcal{A}(M)$;
2. if $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 \sqcup A_2 \in \mathcal{A}(M)$;
3. downward closed: $A_1 \sqsubseteq A_2 \in \mathcal{A}(M) \quad \Rightarrow \quad A_1 \in \mathcal{A}(M)$.

Proof. With Definition 2, (1) and (3) become trivial.

Lemma. The set $\mathcal{A}(M)$ is an ideal w.r.t. $\sqsubseteq$:

1. $\bot \in \mathcal{A}(M)$;
2. if $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 \sqcup A_2 \in \mathcal{A}(M)$;
3. downward closed: $A_1 \sqsubseteq A_2 \in \mathcal{A}(M) \quad \Rightarrow \quad A_1 \in \mathcal{A}(M)$.

Proof. With Definition 2, (1) and (3) become trivial.
2) If $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 = \omega(N_1)$ and $A_2 = \omega(N_2)$ for some

$$N_1 \qquad =_\beta \qquad M \qquad =_\beta \qquad N_2$$

Lemma. The set $\mathcal{A}(M)$ is an ideal w.r.t. $\sqsubseteq$:

1. $\perp \in \mathcal{A}(M)$;
2. if $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 \sqcup A_2 \in \mathcal{A}(M)$;
3. downward closed: $A_1 \sqsubseteq A_2 \in \mathcal{A}(M) \quad \Rightarrow \quad A_1 \in \mathcal{A}(M)$.

Proof. With Definition 2, (1) and (3) become trivial.
2) If $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 = \omega(N_1)$ and $A_2 = \omega(N_2)$ for some

$$\begin{tikzcd}
N_1 \arrow[dr, two heads] & =_\beta & M \arrow[dl, two heads] \arrow[dr, two heads] & =_\beta & N_2 \arrow[dl, two heads] \\
 & R_1 \arrow[dr, two heads] & & R_2 \arrow[dl, two heads] & \\
 & & R & &
\end{tikzcd}$$

Conclude since direct approximants increase along reduction. □

Lemma. The set $\mathcal{A}(M)$ is an ideal w.r.t. $\sqsubseteq$:

1. $\perp \in \mathcal{A}(M)$;
2. if $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 \sqcup A_2 \in \mathcal{A}(M)$;
3. downward closed: $A_1 \sqsubseteq A_2 \in \mathcal{A}(M) \quad \Rightarrow \quad A_1 \in \mathcal{A}(M)$.

Proof. With Definition 2, (1) and (3) become trivial.
2) If $A_1, A_2 \in \mathcal{A}(M)$ then $A_1 = \omega(N_1)$ and $A_2 = \omega(N_2)$ for some

$$\begin{tikzcd}
N_1 \arrow[dr] & =_\beta & M \arrow[dl] \arrow[dr] & =_\beta & N_2 \arrow[dl] \\
& R_1 \arrow[dr] & & R_2 \arrow[dl] & \\
& & R & &
\end{tikzcd}$$

Conclude since direct approximants increase along reduction. $\square$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Examples:

- $\mathcal{A}(\Omega) = \{\bot\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathrm{Y}) = \{ \bot, \lambda f.f\bot, \lambda f.f(f\bot), \lambda f.f(f(f\bot)), \dots, \lambda f.f^n(\bot), \dots \}$

Example

$$\mathrm{BT}(\mathrm{Y}) = \begin{array}{c} \lambda f.f \\ | \\ f \\ | \\ f \\ | \\ f \\ | \\ \vdots \end{array}$$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Example

$$\mathrm{BT}(\mathsf{Y}) = \perp$$

Examples:

- $\mathcal{A}(\Omega) = \{\perp\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathsf{Y}) = \{ \perp, \lambda f.f\perp, \lambda f.f(f\perp), \lambda f.f(f(f\perp)), \ldots, \lambda f.f^n(\perp), \ldots \}$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Examples:

- $\mathcal{A}(\Omega) = \{\bot\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathrm{Y}) = \{ \bot, \lambda f.f\bot, \lambda f.f(f\bot), \lambda f.f(f(f\bot)), \ldots, \lambda f.f^n(\bot), \ldots \}$

Example

$$\begin{array}{c} \mathrm{BT}(\mathrm{Y}) \\ \shortparallel \\ \lambda f.f \\ | \\ \bot \end{array}$$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Examples:

- $\mathcal{A}(\Omega) = \{\bot\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathrm{Y}) = \{ \bot, \lambda f.f\bot, \lambda f.f(f\bot), \lambda f.f(f(f\bot)), \ldots, \lambda f.f^n(\bot), \ldots \}$

Example

$$\begin{array}{c} \mathrm{BT}(\mathrm{Y}) \\ \shortparallel \\ \lambda f.f \\ | \\ f \\ | \\ \bot \end{array}$$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Examples:

- $\mathcal{A}(\Omega) = \{\bot\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathrm{Y}) = \{ \bot, \lambda f.f\bot, \lambda f.f(f\bot), \lambda f.f(f(f\bot)), \dots, \lambda f.f^n(\bot), \dots \}$

Example

$$\begin{array}{c} \mathrm{BT}(\mathrm{Y}) \\ \shortparallel \\ \lambda f.f \\ | \\ f \\ | \\ f \\ | \\ \bot \end{array}$$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Examples:

- $\mathcal{A}(\Omega) = \{\bot\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathrm{Y}) = \{ \bot, \lambda f.f\bot, \lambda f.f(f\bot), \lambda f.f(f(f\bot)), \ldots, \lambda f.f^n(\bot), \ldots \}$

Example

$$\begin{array}{c} \mathrm{BT}(\mathrm{Y}) \\ \shortparallel \\ \lambda f.f \\ | \\ f \\ | \\ f \\ | \\ f \\ | \\ \vdots \end{array}$$

The Syntactic Approximation Theorem

For all $M \in \Lambda$,

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Examples:

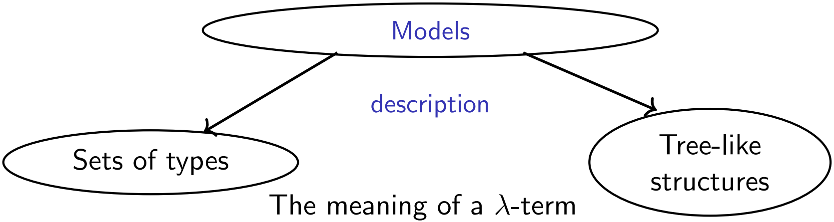
- $\mathcal{A}(\Omega) = \{\perp\}$, for $\Omega = (\lambda x.xx)(\lambda x.xx)$,
- $\mathcal{A}(\mathrm{Y}) = \{ \perp, \lambda f.f\perp, \lambda f.f(f\perp), \lambda f.f(f(f\perp)), \ldots, \lambda f.f^n(\perp), \ldots \}$

Example

$$\begin{array}{c} \mathrm{BT}(\mathrm{Y}) \\ \shortparallel \\ \lambda f.f \\ | \\ f \\ | \\ f \\ | \\ f \\ | \\ \vdots \end{array}$$

Filter Models

Denotational Models



The meaning of a λ-term
is given by the set of its types

A filter λ-model and the completeness of type assignment.
H. Barendregt, M. Coppo, M. Dezani-Ciancaglini.
J. Symb. Log. 48(4): 931-940 (1983)

BCD Types:

$\mathbb{A}_\omega$:	$\omega, \alpha, \beta, \dots$	countable set of atoms
$\mathbb{T}_\wedge$:	$\sigma, \tau ::= \omega \mid \alpha \mid \sigma \to \tau \mid \sigma \wedge \tau$	intersection types

Derivation Rules:

$$x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash_\wedge x_i : \sigma_i \quad (ax) \qquad\qquad \Gamma \vdash_\wedge M : \omega \quad (U)$$

$$\frac{\Gamma \vdash_\wedge M : \tau \to \sigma \quad \Gamma \vdash_\wedge N : \tau}{\Gamma \vdash_\wedge MN : \sigma}\ (\to_E) \qquad \frac{\Gamma, x : \sigma \vdash_\wedge M : \tau}{\Gamma \vdash_\wedge \lambda x.M : \sigma \to \tau}\ (\to_I)$$

$$\frac{\Gamma \vdash_\wedge M : \sigma \quad \Gamma \vdash_\wedge M : \tau}{\Gamma \vdash_\wedge M : \sigma \wedge \tau}\ (\wedge_I) \qquad \frac{\Gamma \vdash_\wedge M : \sigma \quad \sigma \leq \tau}{\Gamma \vdash_\wedge M : \tau}\ (\leq)$$

Subtyping:

$$\sigma \leq \sigma \ (\text{refl}) \qquad \sigma \wedge \tau \leq \sigma \ (\text{incl}_L) \qquad \sigma \wedge \tau \leq \tau \ (\text{incl}_R) \qquad \sigma \leq \omega \ (\text{top})$$

$$(\sigma \to \tau) \wedge (\sigma \to \tau') \leq \sigma \to (\tau \wedge \tau') \quad (\to_\wedge) \qquad\qquad \omega \leq \sigma \to \omega \ (\text{arr}_\omega)$$

$$\frac{\sigma \leq \gamma \quad \gamma \leq \tau}{\sigma \leq \tau}\ (\text{trans}) \qquad \frac{\sigma \leq \tau \quad \sigma \leq \tau'}{\sigma \leq \tau \wedge \tau'}\ (\text{glb}) \qquad \frac{\sigma' \leq \sigma \quad \tau \leq \tau'}{\sigma \to \tau \leq \sigma' \to \tau'}\ (\to)$$

Filters of types

We study the collection

$$\mathcal{F} = \{F \subseteq \mathbb{T}_\wedge \mid F \text{ is a filter of types } \}$$

- A subset $F \subseteq \mathbb{T}_\wedge$ is a filter of types if:
  1. $\omega \in F$,
  2. $\sigma, \tau \in F$ implies $\sigma \wedge \tau \in F$,
  3. $\sigma \in F$ and $\sigma \leq \tau$ imply $\tau \in F$.
- The principal filter generated by $\sigma \in \mathbb{T}_\wedge$ is given by:

$$\sigma\uparrow = \{\tau \in \mathbb{T}_\wedge \mid \sigma \leq \tau\} \in \mathcal{F}$$

Interpretation in the filter model BCD

Given $M \in \Lambda$ and a valuation $\rho : \Lambda \to \mathcal{F}$, define the interpretation of M w.r.t. ρ by induction:

$$
\begin{aligned}
[\![x]\!]^{\mathcal{F}}_{\rho} &= \rho(x) \\
[\![\lambda x.M]\!]^{\mathcal{F}}_{\rho} &= \{\sigma \to \tau \mid \tau \in [\![M]\!]^{\mathcal{F}}_{\rho[x:=\sigma\uparrow]}\}\uparrow \\
[\![MN]\!]^{\mathcal{F}}_{\rho} &= \{\tau \mid \exists \sigma \in [\![N]\!]^{\mathcal{F}}_{\rho} . \sigma \to \tau \in [\![M]\!]^{\mathcal{F}}_{\rho}\}
\end{aligned}
$$

Characterization. For $M \in \Lambda$, we have

$$[\![M]\!]^{\mathcal{F}}_{\rho} = \{\sigma \in \mathbb{T}_{\wedge} \mid \Gamma \vdash_{\wedge} M : \sigma\}$$

where $\rho(x) = \tau\uparrow$, for all $x : \tau \in \Gamma$.

Interpretation in the filter model BCD

Given $M \in \Lambda$ and a valuation $\rho : \Lambda \to \mathcal{F}$, define the interpretation of M w.r.t. ρ by induction:

$$\begin{aligned} [\![x]\!]^{\mathcal{F}}_{\rho} &= \rho(x) \\ [\![\lambda x.M]\!]^{\mathcal{F}}_{\rho} &= \{\sigma \to \tau \mid \tau \in [\![M]\!]^{\mathcal{F}}_{\rho[x:=\sigma\uparrow]}\}\uparrow \\ [\![MN]\!]^{\mathcal{F}}_{\rho} &= \{\tau \mid \exists \sigma \in [\![N]\!]^{\mathcal{F}}_{\rho} \,.\, \sigma \to \tau \in [\![M]\!]^{\mathcal{F}}_{\rho}\} \end{aligned}$$

Characterization. For $M \in \Lambda$, we have

$$[\![M]\!]^{\mathcal{F}}_{\rho} = \{\sigma \in \mathbb{T}_{\wedge} \mid \Gamma \vdash_{\wedge} M : \sigma\}$$

where $\rho(x) = \tau\uparrow$, for all $x : \tau \in \Gamma$.

Interpretation - examples

Remark. If M is closed, then $\forall \rho, \rho' \,.\, [\![M]\!]_\rho = [\![M]\!]_{\rho'}$ (forget the ρ!).

Exemples:

- $[\![\Omega]\!] = \{\sigma \mid \sigma \simeq \omega\}$ (since $\vdash_\wedge \Omega : \omega$)
- $[\![\mathsf{I}]\!] = \{\sigma \to \sigma \mid \sigma \in \mathbb{T}_\wedge\}{\uparrow}$
- $[\![\lambda f.f\Omega]\!] = \{(\omega \to \sigma) \to \sigma \mid \sigma \in \mathbb{T}_\wedge\}{\uparrow}$
- $[\![\lambda x.xx]\!] = \{(\sigma \wedge (\sigma \to \tau)) \to \tau \mid \sigma, \tau \in \mathbb{T}_\wedge\}{\uparrow}$
- $[\![\mathsf{K}]\!] = \{\sigma \to \tau \to \sigma \mid \sigma, \tau \in \mathbb{T}_\wedge\}{\uparrow}$
- Spoiler alert. For all U unsolvable, we have $(\forall \rho)$:

$$[\![U]\!]_\rho = \{\sigma \mid \sigma \simeq \omega\}$$

The theory induced by the model

Every model induces a notion of equality between λ-terms:

$$\mathcal{F} \models M = N \text{ if and only if } \forall \rho \,.\, [\![M]\!]_\rho = [\![N]\!]_\rho$$

Main properties

Soundness:

$$M =_\beta N \quad \Rightarrow \quad \mathcal{F} \models M = N$$

(By the characterization + SR + SE)

Contextuality:

$$\begin{array}{rcl} \mathcal{F} \models M = M' & \Rightarrow & \mathcal{F} \models \lambda x.M = \lambda x.M' \\ \mathcal{F} \models M = M' \text{ and } \mathcal{F} \models N = N' & \Rightarrow & \mathcal{F} \models MN = M'N' \end{array}$$

The theory induced by the model

Every model induces a notion of equality between λ-terms:

$$\mathcal{F} \models M = N \text{ if and only if } \forall \rho \,.\, [\![M]\!]_\rho = [\![N]\!]_\rho$$

Main properties

Soundness:

$$M =_\beta N \quad \Rightarrow \quad \mathcal{F} \models M = N$$

(By the characterization + SR + SE)

Contextuality:

$$\begin{array}{rcl} \mathcal{F} \models M = M' & \Rightarrow & \mathcal{F} \models \lambda x.M = \lambda x.M' \\ \mathcal{F} \models M = M' \text{ and } \mathcal{F} \models N = N' & \Rightarrow & \mathcal{F} \models MN = M'N' \end{array}$$

Lambda terms sharing the same interpretation

Examples:

- $\mathcal{F} \models \mathsf{KI}\Omega = \mathsf{I}$, as $\mathsf{KI}\Omega =_\beta \mathsf{I}$.
- $\mathcal{F} \models \Omega = \mathsf{YI}$, as they are both unsolvable. Thus:
- $\mathcal{F} \models \lambda x.x\Omega = \lambda x.x(\mathsf{YI})$, by contextuality.
- For all fpc's Z, we have $\mathcal{F} \models Z = \mathsf{Y}$.

Interpreting $\lambda\perp$-terms

We can extend the type system BCD to $\Lambda_\perp$ by adding this rule:

$$\frac{}{\Gamma \vdash_\wedge \perp : \omega}\ (\perp)$$

And therefore the interpretation of $\lambda\perp$-terms:

$$[\![\perp]\!]_\rho = \{\sigma \mid \sigma \simeq \omega\}$$

Everything works fine because $\perp$ and Ω have the same properties.

Typing the direct approximant

Lemma. For all $M \in \Lambda$:

$$\Gamma \vdash \omega(M) : \sigma \quad \Rightarrow \quad \Gamma \vdash M : \sigma$$

Proof. By induction on a derivation of $\Gamma \vdash \omega(M) : \sigma$.
Case $\omega(M) = \bot$. Then $\sigma \simeq \omega$, whence $\Gamma \vdash_\wedge M : \omega$.
Case $\omega(M) = y\omega(M_1) \cdots \omega(M_k)$, since $M = yM_1 \cdots M_k$.
By Inversion Lemma II, applied k times, we get:

$$\Gamma \vdash_\wedge y : \tau_1 \to \cdots \to \tau_k \to \sigma$$

$$\Gamma \vdash_\wedge \omega(M_1) : \tau_1 \qquad \cdots \qquad \Gamma \vdash_\wedge \omega(M_k) : \tau_k .$$

By IH, we get $\Gamma \vdash_\wedge M_i : \tau_i$, from which $\Gamma \vdash_\wedge yM_1 \cdots M_k : \sigma$ follows.

To be continued. . .

Typing the direct approximant

Lemma. For all $M \in \Lambda$:

$$\Gamma \vdash \omega(M) : \sigma \quad \Rightarrow \quad \Gamma \vdash M : \sigma$$

Proof. Case $\omega(M) = \lambda x.\omega(M')$, with $M = \lambda x.M'$. By Inversion Lemma I, there are $\tau_1, \ldots, \tau_k, \gamma_1, \ldots, \gamma_k \in \mathbb{T}_\wedge$ such that

$$\begin{array}{l} (\tau_1 \to \gamma_1) \wedge \cdots \wedge (\tau_k \to \gamma_k) \leq \sigma \\ \forall i \in \{1, \ldots, k\} \,.\, \Gamma, x : \tau_i \vdash_\wedge \omega(M') : \gamma_i. \end{array}$$

By IH, we get $\Gamma, x : \tau_i \vdash_\wedge M' : \gamma_i$. We conclude:

$$\dfrac{\dfrac{\Gamma \vdash_\wedge \lambda x.M' : \tau_1 \to \gamma_1 \quad \cdots \quad \Gamma \vdash_\wedge \lambda x.M' : \tau_k \to \gamma_k}{\Gamma \vdash_\wedge \lambda x.M' : \wedge_i(\tau_i \to \gamma_i)} \qquad \wedge_i(\tau_i \to \gamma_i) \leq \sigma}{\Gamma \vdash_\wedge \lambda x.M : \sigma}$$

Semantic Approximation Theorem

Semantic Approximation Theorem. For $M \in \Lambda$.

$$[\![M]\!]_\rho = \bigcup\{[\![A]\!]_\rho \mid A \in \mathcal{A}(M)\}$$

This is what we are going to prove.

Corollary. If $\mathrm{BT}(M) = \mathrm{BT}(N)$ then $\mathcal{F} \models M = N$.

Proof. Assume $\mathrm{BT}(M) = \mathrm{BT}(N)$. Then also $\mathcal{A}(M) = \mathcal{A}(N)$.

$$\begin{array}{rll} [\![M]\!]_\rho = & \bigcup_{A \in \mathcal{A}(M)} [\![A]\!]_\rho, & \text{by Approximation Theorem,} \\ & \bigcup_{A \in \mathcal{A}(N)} [\![A]\!]_\rho = [\![N]\!]_\rho & \square \end{array}$$

Semantic Approximation Theorem

Semantic Approximation Theorem. For $M \in \Lambda$.

$$[\![M]\!]_\rho = \bigcup \{ [\![A]\!]_\rho \mid A \in \mathcal{A}(M) \}$$

This is what we are going to prove.

Corollary. If $\mathrm{BT}(M) = \mathrm{BT}(N)$ then $\mathcal{F} \models M = N$.

Proof. Assume $\mathrm{BT}(M) = \mathrm{BT}(N)$. Then also $\mathcal{A}(M) = \mathcal{A}(N)$.

$$\begin{aligned} [\![M]\!]_\rho = \; & \textstyle\bigcup_{A \in \mathcal{A}(M)} [\![A]\!]_\rho, \quad \text{by Approximation Theorem,} \\ & \textstyle\bigcup_{A \in \mathcal{A}(N)} [\![A]\!]_\rho = [\![N]\!]_\rho \end{aligned}$$

$\square$

Logical Approximation Theorem

Theorem

$$\Gamma \vdash_{\wedge} M : \sigma \iff \exists P \in \mathcal{A}(M) \,.\, \Gamma \vdash_{\wedge} P : \sigma$$

Proof. ($\Leftarrow$) Assume $\Gamma \vdash_{\wedge} P : \sigma$, for some $P \in \mathcal{A}(M)$.

- By Definition 1, there exists $N =_{\beta} M$ such that $P = \omega(N)$.
- We have seen that $\Gamma \vdash_{\wedge} \omega(N) : \sigma$ implies $\Gamma \vdash_{\wedge} N : \sigma$.
- By SR+SE+confluence, typing is invariant under $=_{\beta}$.
- Conclude $\Gamma \vdash_{\wedge} M : \sigma$.

($\Rightarrow$) Mmm... it does not seem so easy. Ideas?

Of course!

The property that we want for the λ-terms of type σ in Γ:

$$\mathcal{P}_\Gamma(\sigma) = \{M \in \Lambda \mid \exists A \in \mathcal{A}(M) \,.\, \Gamma \vdash_\wedge A : \sigma\}$$

Define the interpretation of $\sigma \in \mathbb{T}_\wedge$ in Γ as follows:

$$\begin{array}{lcl} |\alpha|_\Gamma & = & \mathcal{P}_\Gamma(\alpha), \text{ for } \alpha \in \mathbb{A}_\omega, \\ |\sigma \to \tau|_\Gamma & = & \{M \mid \forall \Gamma', \forall N \in |\sigma|_{\Gamma'} \,.\, MN \in |\tau|_{\Gamma \wedge \Gamma'}\} \\ |\sigma \wedge \tau|_\Gamma & = & |\sigma|_\Gamma \cap |\tau|_\Gamma \end{array}$$

Remark. Recall that $\perp \in \mathcal{A}(M)$ and $\Gamma \vdash_\wedge M : \omega$, for every M. So

$$\sigma \simeq \omega \text{ implies } |\sigma|_\Gamma = \Lambda$$

In particular $|\sigma \wedge \omega|_\Gamma = |\sigma|_\Gamma$, which is consistent with $\sigma \wedge \omega \simeq \sigma$.

Of course!

The property that we want for the λ-terms of type σ in Γ:

$$\mathcal{P}_\Gamma(\sigma) = \{M \in \Lambda \mid \exists A \in \mathcal{A}(M) \,.\, \Gamma \vdash_\wedge A : \sigma\}$$

Define the interpretation of $\sigma \in \mathbb{T}_\wedge$ in Γ as follows:

$$\begin{array}{lcl} |\alpha|_\Gamma & = & \mathcal{P}_\Gamma(\alpha), \text{ for } \alpha \in \mathbb{A}_\omega, \\ |\sigma \to \tau|_\Gamma & = & \bigcap_{\Gamma'} \left(|\sigma|_{\Gamma'} \Rightarrow |\tau|_{\Gamma \wedge \Gamma'}\right) \\ |\sigma \wedge \tau|_\Gamma & = & |\sigma|_\Gamma \cap |\tau|_\Gamma \end{array}$$

Remark. Recall that $\perp \in \mathcal{A}(M)$ and $\Gamma \vdash_\wedge M : \omega$, for every M. So

$$\sigma \simeq \omega \text{ implies } |\sigma|_\Gamma = \Lambda$$

In particular $|\sigma \wedge \omega|_\Gamma = |\sigma|_\Gamma$, which is consistent with $\sigma \wedge \omega \simeq \sigma$.

Two steps

We need to prove that all typable terms are approximable:

$$\Gamma \vdash_\wedge M : \sigma \quad \Rightarrow \quad M \in \mathcal{P}_\Gamma(\sigma)$$

The proof is split into 2 steps. Step 1:

$$\Gamma \vdash_\wedge M : \sigma \quad \Rightarrow \quad M \in |\sigma|_\Gamma \tag{1}$$

Step 2:

$$M \in |\sigma|_\Gamma \quad \Rightarrow \quad M \in \mathcal{P}_\Gamma(\sigma) \tag{2}$$

Auxiliary lemmas

Lemma (AUX0). $M \in |\sigma|_\Gamma$ and $M =_\beta N$ implies $N \in |\sigma|_\Gamma$.

Proof sketch. By induction on σ.

Lemma (AUX1). For $M \in \Lambda$ and $z \notin \mathrm{FV}(M)$, we have:

$$Mz \in \mathcal{P}_{\Gamma, z:\tau}(\sigma) \quad \Rightarrow \quad M \in \mathcal{P}_\Gamma(\tau \to \sigma)$$

Proof sketch. Let $P \in \mathcal{A}(Mz)$ such that $\Gamma, z : \tau \vdash_\wedge P : \sigma$. By cases on P and M, show that $\exists P' \in \mathcal{A}(M)$ such that $\Gamma \vdash_\wedge P' : \tau \to \sigma$.

Lemma (AUX2). $\sigma \leq \tau$ implies $|\sigma|_\Gamma \subseteq |\tau|_\Gamma$.

Proof sketch. By induction on a derivation of $\sigma \leq \tau$.

Step 2 comes first

Main Lemma.

1. $xM_1 \cdots M_n \in \mathcal{P}_\Gamma(\sigma)$ implies $xM_1 \cdots M_n \in |\sigma|_\Gamma$.
2. $|\sigma|_\Gamma \subseteq \mathcal{P}_\Gamma(\sigma)$.

Proof. By simultaneous induction (IH1, IH2 = ind. hyp.)
1. Case $\sigma \simeq \omega$. Trivial since $|\omega|_\Gamma = \Lambda$.
Case $\sigma = \tau_1 \wedge \tau_2$. Easy.
Case $\sigma = \tau_1 \to \tau_2$ ($\not\simeq \omega$). Let $x\vec{M} \in \mathcal{P}_\Gamma(\sigma)$. Then

$$\exists P \in \mathcal{A}(xM_1 \cdots M_n) \,.\, \Gamma \vdash_\wedge P : \tau_1 \to \tau_2$$

with $A_i \in \mathcal{A}(M_i)$.

Step 2 comes first

Main Lemma.

1. $xM_1 \cdots M_n \in \mathcal{P}_\Gamma(\sigma)$ implies $xM_1 \cdots M_n \in |\sigma|_\Gamma$.
2. $|\sigma|_\Gamma \subseteq \mathcal{P}_\Gamma(\sigma)$.

Proof. By simultaneous induction (IH1, IH2 = ind. hyp.)
1. Case $\sigma \simeq \omega$. Trivial since $|\omega|_\Gamma = \Lambda$.
Case $\sigma = \tau_1 \wedge \tau_2$. Easy.
Case $\sigma = \tau_1 \to \tau_2$ $(\not\simeq \omega)$. Let $x\vec{M} \in \mathcal{P}_\Gamma(\sigma)$. Then

$$\exists P \in \mathcal{A}(xM_1 \cdots M_n) \,.\, \Gamma \vdash_\wedge P = xA_1 \cdots A_n : \tau_1 \to \tau_2$$

with $A_i \in \mathcal{A}(M_i)$. By IH2, $\forall N \in |\tau_1|_{\Gamma'} \quad \Rightarrow \quad N \in \mathcal{P}_{\Gamma'}(\sigma)$, i.e.
$\exists A_{n+1} \in \mathcal{A}(N)$ such that $\quad \Gamma' \vdash_\wedge A_{n+1} : \tau_1$.

Step 2 comes first

Main Lemma.

1. $xM_1 \cdots M_n \in \mathcal{P}_\Gamma(\sigma)$ implies $xM_1 \cdots M_n \in |\sigma|_\Gamma$.
2. $|\sigma|_\Gamma \subseteq \mathcal{P}_\Gamma(\sigma)$.

Proof. By simultaneous induction (IH1, IH2 = ind. hyp.)
1. Case $\sigma \simeq \omega$. Trivial since $|\omega|_\Gamma = \Lambda$.
Case $\sigma = \tau_1 \wedge \tau_2$. Easy.
Case $\sigma = \tau_1 \to \tau_2$ ($\not\simeq \omega$). Let $x\vec{M} \in \mathcal{P}_\Gamma(\sigma)$. Then

$$\exists P \in \mathcal{A}(xM_1 \cdots M_n)\,.\,\Gamma \vdash_\wedge P = xA_1 \cdots A_n : \tau_1 \to \tau_2$$

with $A_i \in \mathcal{A}(M_i)$. By IH2, $\forall N \in |\tau_1|_{\Gamma'} \quad \Rightarrow \quad N \in \mathcal{P}_{\Gamma'}(\sigma)$, i.e.
$\exists A_{n+1} \in \mathcal{A}(N)$ such that $\quad \Gamma' \vdash_\wedge A_{n+1} : \tau_1$.

Step 2 comes first

Main Lemma.

1. $xM_1 \cdots M_n \in \mathcal{P}_\Gamma(\sigma)$ implies $xM_1 \cdots M_n \in |\sigma|_\Gamma$.
2. $|\sigma|_\Gamma \subseteq \mathcal{P}_\Gamma(\sigma)$.

Proof. By simultaneous induction (IH1, IH2 = ind. hyp.)
1. Case $\sigma \simeq \omega$. Trivial since $|\omega|_\Gamma = \Lambda$.
Case $\sigma = \tau_1 \wedge \tau_2$. Easy.
Case $\sigma = \tau_1 \to \tau_2$ ($\not\simeq \omega$). Let $x\vec{M} \in \mathcal{P}_\Gamma(\sigma)$. Then

$$\exists P \in \mathcal{A}(xM_1 \cdots M_n) \,.\, \Gamma \wedge \Gamma' \vdash_\wedge P = xA_1 \cdots A_n : \tau_1 \to \tau_2$$

with $A_i \in \mathcal{A}(M_i)$. By IH2, $\forall N \in |\tau_1|_{\Gamma'} \quad \Rightarrow \quad N \in \mathcal{P}_{\Gamma'}(\sigma)$, i.e. $\exists A_{n+1} \in \mathcal{A}(N)$ such that $\Gamma \wedge \Gamma' \vdash_\wedge A_{n+1} : \tau_1$. By $(\to_E)$,

$$\Gamma \wedge \Gamma' \vdash_\wedge xA_1 \cdots A_{n+1} : \tau_2 \;\Rightarrow^{\mathrm{IH1}}\; x\vec{M}N \in |\tau_2|_{\Gamma \wedge \Gamma'} \text{ (conclude)}$$

Step 2 comes first

Main Lemma.

1. $xM_1 \cdots M_n \in \mathcal{P}_\Gamma(\sigma)$ implies $xM_1 \cdots M_n \in |\sigma|_\Gamma$.
2. $|\sigma|_\Gamma \subseteq \mathcal{P}_\Gamma(\sigma)$.

Proof. By simultaneous induction (IH1, IH2 = ind. hyp.)
1. Case $\sigma \simeq \omega$. Trivial since $|\omega|_\Gamma = \Lambda$.
Case $\sigma = \tau_1 \wedge \tau_2$. Easy.
Case $\sigma = \tau_1 \to \tau_2$ ($\not\simeq \omega$). Let $x\vec{M} \in \mathcal{P}_\Gamma(\sigma)$. Then

$$\exists P \in \mathcal{A}(xM_1 \cdots M_n)\,.\,\Gamma \wedge \Gamma' \vdash_\wedge P = xA_1 \cdots A_n : \tau_1 \to \tau_2$$

with $A_i \in \mathcal{A}(M_i)$. By IH2, $\forall N \in |\tau_1|_{\Gamma'} \quad \Rightarrow \quad N \in \mathcal{P}_{\Gamma'}(\sigma)$, i.e. $\exists A_{n+1} \in \mathcal{A}(N)$ such that $\Gamma \wedge \Gamma' \vdash_\wedge A_{n+1} : \tau_1$. By ($\to_E$),

$$\Gamma \wedge \Gamma' \vdash_\wedge xA_1 \cdots A_{n+1} : \tau_2 \;\stackrel{\text{IH1}}{\Rightarrow}\; x\vec{M}N \in |\tau_2|_{\Gamma\wedge\Gamma'} \text{ (conclude)}$$

Step 2 comes first

Main Lemma.

1. $xM_1 \cdots M_n \in \mathcal{P}_\Gamma(\sigma)$ implies $xM_1 \cdots M_n \in |\sigma|_\Gamma$.
2. $|\sigma|_\Gamma \subseteq \mathcal{P}_\Gamma(\sigma)$.

Proof. 2. Case $\sigma \simeq \omega$. Trivial since $\mathcal{P}_\Gamma(\omega) = \Lambda$.
Case $\sigma = \tau_1 \wedge \tau_2$. By rule $(\wedge_I)$ and IH2.
Case $\sigma = \tau_1 \to \tau_2$. Take $M \in |\tau_1 \to \tau_2|_\Gamma$ and z a fresh variable.
Since $z \in |\tau_1|_{z:\tau_1}$, by IH1, we have

$$
\begin{array}{ccc}
M \in |\tau_1 \to \tau_2|_\Gamma \text{ and } z \in |\tau_1|_{z:\tau_1} & \Rightarrow & Mz \in |\tau_2|_{\Gamma, z:\tau_2} \\
\text{(by IH2)} & \Rightarrow & Mz \in \mathcal{P}_{\Gamma, z:\tau_1}(\tau_2) \\
\text{(AUX1)} & \Rightarrow & M \in \mathcal{P}_\Gamma(\tau_1 \to \tau_2)
\end{array}
$$

□

Step 1 comes later

Main Theorem. Let $M \in \Lambda$, with $\mathrm{FV}(M) \subseteq \{x_1, \ldots, x_n\}$.
Let $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$ and $\Gamma_1, \ldots, \Gamma_n$ be type environments.
For all $N_1 \in |\tau_1|_{\Gamma_1}, \ldots, N_n \in |\tau_n|_{\Gamma_n}$, we have

$$\Gamma \vdash_\wedge M : \sigma \quad \Rightarrow \quad M[x_1 := N_1, \ldots, x_n := N_n] \in |\sigma|_{\Gamma \wedge \Gamma_1 \wedge \cdots \wedge \Gamma_n}$$

Proof. By induction on the derivation of $\Gamma \vdash_\wedge M : \sigma$.

We only see this interesting case: $\dfrac{\Gamma, y : \sigma \vdash_\wedge M : \tau}{\Gamma \vdash_\wedge \lambda y.M : \sigma \to \tau}\ (\to_I)$

By IH, for all $N_i \in |\tau_i|_{\Gamma_i}$, $X \in |\sigma|_{\Gamma_{n+1}}$

$$M[\vec{x} := \vec{N}, y := X] \in |\tau|_{\Gamma \wedge (\bigwedge_{j=1}^{n+1} \Gamma_j)}$$

Step 1 comes later

Main Theorem. Let $M \in \Lambda$, with $\mathrm{FV}(M) \subseteq \{x_1, \ldots, x_n\}$.
Let $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$ and $\Gamma_1, \ldots, \Gamma_n$ be type environments.
For all $N_1 \in |\tau_1|_{\Gamma_1}, \ldots, N_n \in |\tau_n|_{\Gamma_n}$, we have

$$\Gamma \vdash_\wedge M : \sigma \quad \Rightarrow \quad M[x_1 := N_1, \ldots, x_n := N_n] \in |\sigma|_{\Gamma \wedge \Gamma_1 \wedge \cdots \wedge \Gamma_n}$$

Proof. By induction on the derivation of $\Gamma \vdash_\wedge M : \sigma$.

We only see this interesting case: $\dfrac{\Gamma, y : \sigma \vdash_\wedge M : \tau}{\Gamma \vdash_\wedge \lambda y.M : \sigma \to \tau}\ (\to_I)$

By IH, for all $N_i \in |\tau_i|_{\Gamma_i}$, $X \in |\sigma|_{\Gamma_{n+1}}$

$$(\lambda y.M[\vec{x} := \vec{N}])X \to_\beta M[\vec{x} := \vec{N}, y := X] \in |\tau|_{\Gamma \wedge (\bigwedge_{j=1}^{n+1} \Gamma_j)}$$

By (AUX0).

Step 1 comes later

Main Theorem. Let $M \in \Lambda$, with $\mathrm{FV}(M) \subseteq \{x_1, \ldots, x_n\}$.
Let $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$ and $\Gamma_1, \ldots, \Gamma_n$ be type environments.
For all $N_1 \in |\tau_1|_{\Gamma_1}, \ldots, N_n \in |\tau_n|_{\Gamma_n}$, we have

$$\Gamma \vdash_\wedge M : \sigma \quad \Rightarrow \quad M[x_1 := N_1, \ldots, x_n := N_n] \in |\sigma|_{\Gamma \wedge \Gamma_1 \wedge \cdots \wedge \Gamma_n}$$

Proof. By induction on the derivation of $\Gamma \vdash_\wedge M : \sigma$.

We only see this interesting case: $\dfrac{\Gamma, y : \sigma \vdash_\wedge M : \tau}{\Gamma \vdash_\wedge \lambda y.M : \sigma \to \tau}\ (\to_I)$

By IH, for all $N_i \in |\tau_i|_{\Gamma_i}$, $X \in |\sigma|_{\Gamma_{n+1}}$

$$(\lambda y.M[\vec{x} := \vec{N}])X \to_\beta M[\vec{x} := \vec{N}, y := X] \in |\tau|_{\Gamma \wedge (\bigwedge_{j=1}^{n+1} \Gamma_j)}$$

By (AUX0). By definition, $\lambda y.M[\vec{x} := \vec{N}] \in |\sigma \to \tau|_{\Gamma \wedge (\bigwedge_{i=1}^{n} \Gamma_i)}$. $\square$

Summing up...

Logical Approximation Theorem. For $M \in \Lambda$.

$$\Gamma \vdash_\wedge M : \sigma \iff \exists P \in \mathcal{A}(M) \,.\, \Gamma \vdash_\wedge P : \sigma$$

Semantic Approximation Theorem. For $M \in \Lambda$.

$$[\![M]\!]_\rho = \bigcup \{ [\![A]\!]_\rho \mid A \in \mathcal{A}(M) \}$$

Corollary. The theory of the filter model BCD includes BT:

$$\mathcal{B} \subseteq \mathrm{Th}(\mathcal{F}^{\mathsf{BCD}})$$

Bibliography (aka, where should I study?)

- Böhm tree semantics: where you want. E.g.
  H. Barendregt. The type free lambda calculus. (1977).
- Filter models:
  H. Barendregt, W. Dekkers, R. Statman. Lambda Calculus with Types. Perspectives in logic, Cambridge University Press 2013, ISBN 978-0-521-76614-2, pp. I-XXII, 1-833
- Approximation theorem:
  M. Dezani-Ciancaglini, F. Honsell and Y. Motohama. Approximation Theorems for Intersection Type Systems.


ANY
QUESTIONS
?