Ouverture Scientifique

Giulio Manzonetto
giulio.manzonetto@lipn.univ-paris13.fr

Université Sorbonne Paris-Nord,
Laboratoire LIPN

http://lipn.univ-paris13.fr/~manzonetto/

January 5, 2022

Table of contents

The Global Picture

Program Approximation

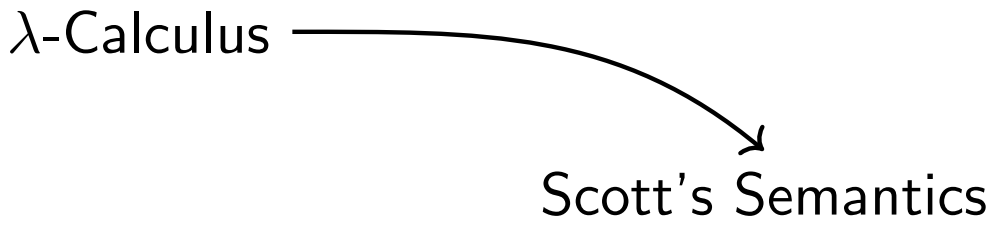
Calculi

Denotational Semantics

λ-Calculus

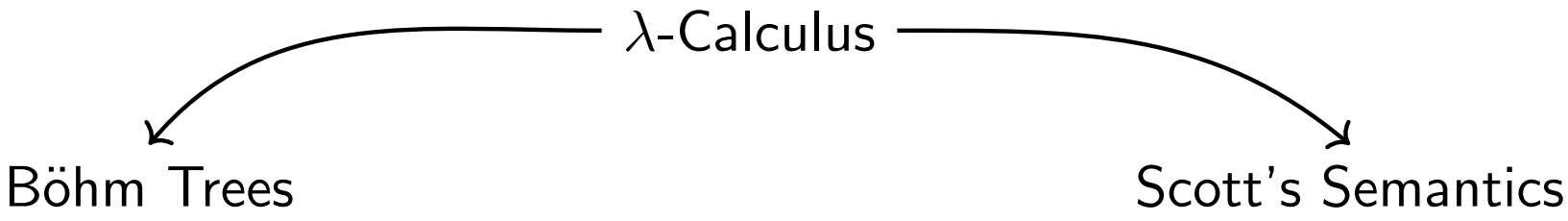
The Global Picture

Program Approximation | Calculi | Denotational Semantics



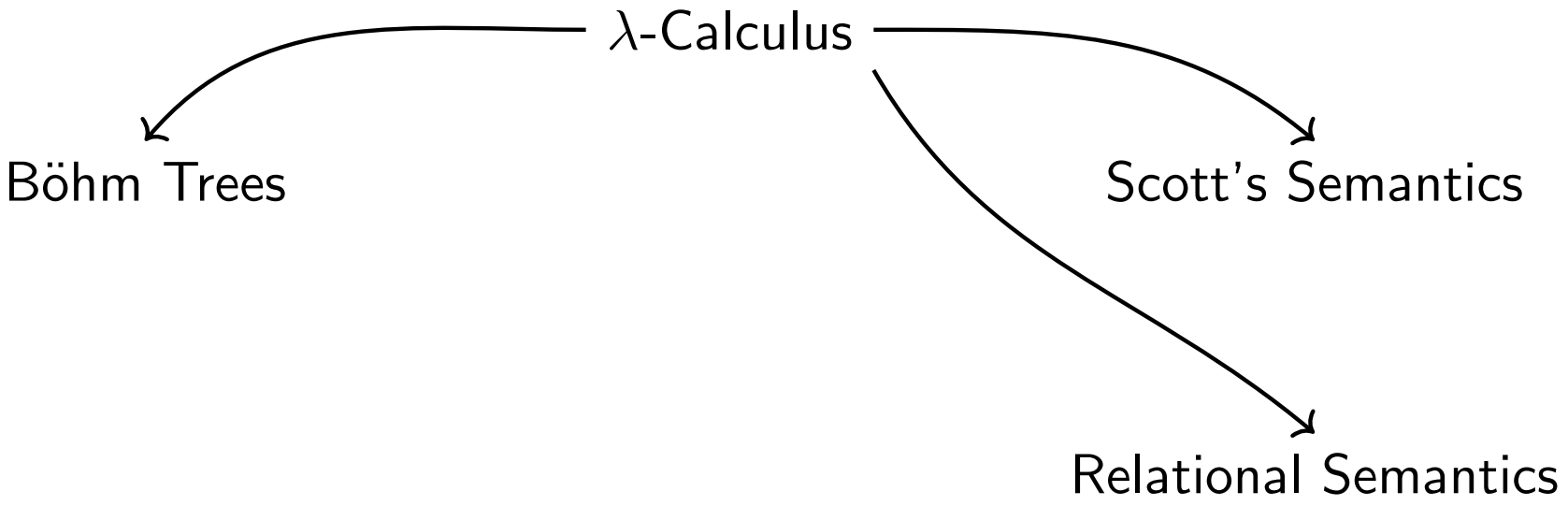
The Global Picture

Program Approximation | Calculi | Denotational Semantics



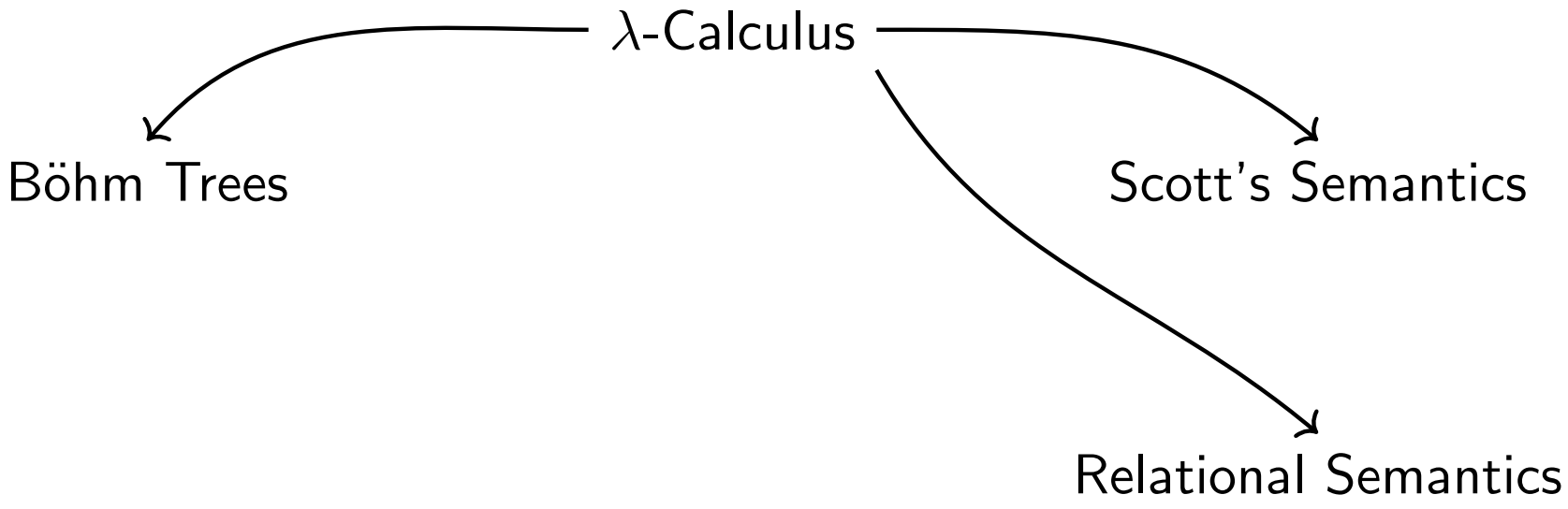
The Global Picture

Program Approximation | Calculi | Denotational Semantics



The Global Picture

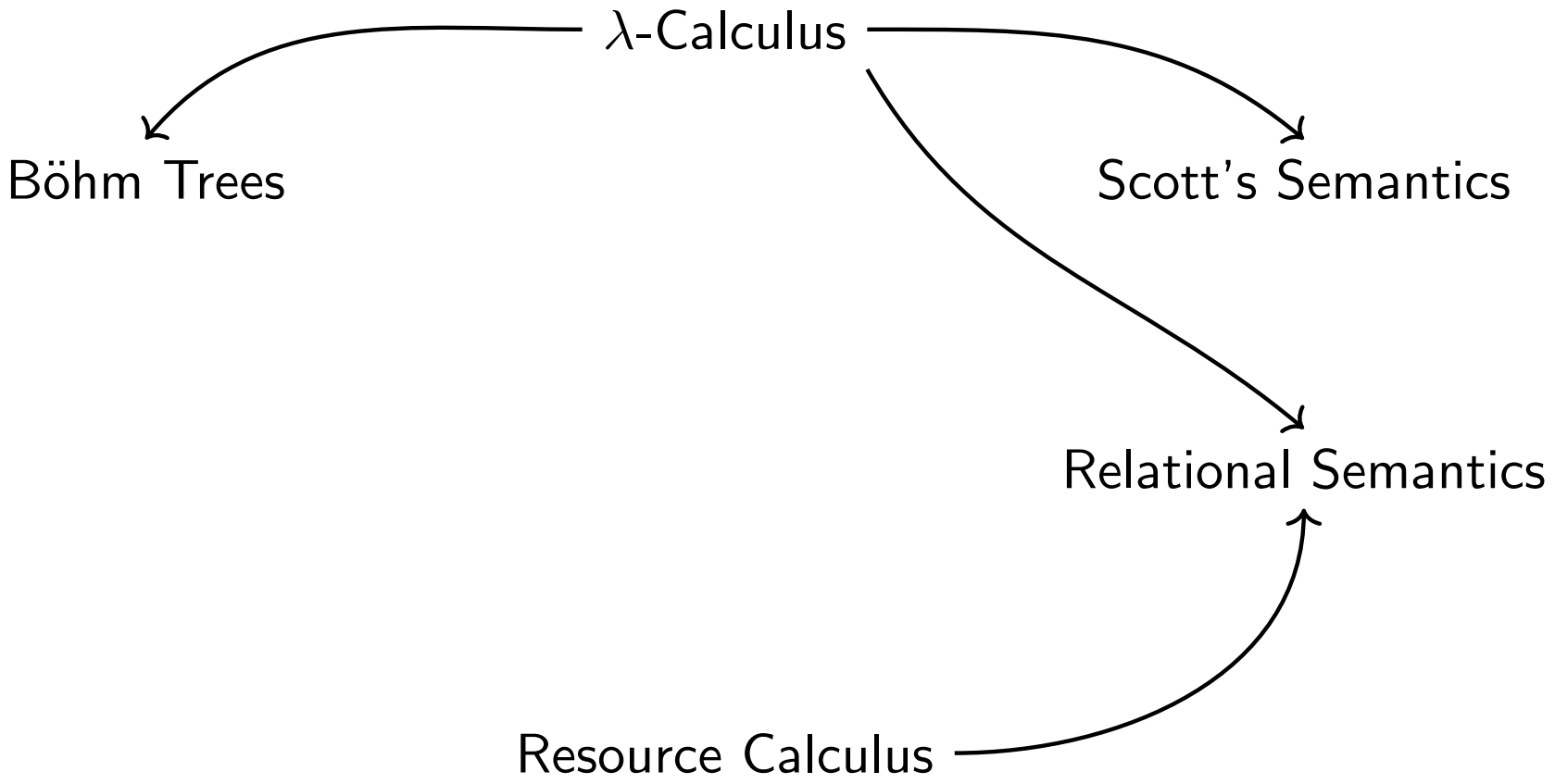
Program Approximation | Calculi | Denotational Semantics



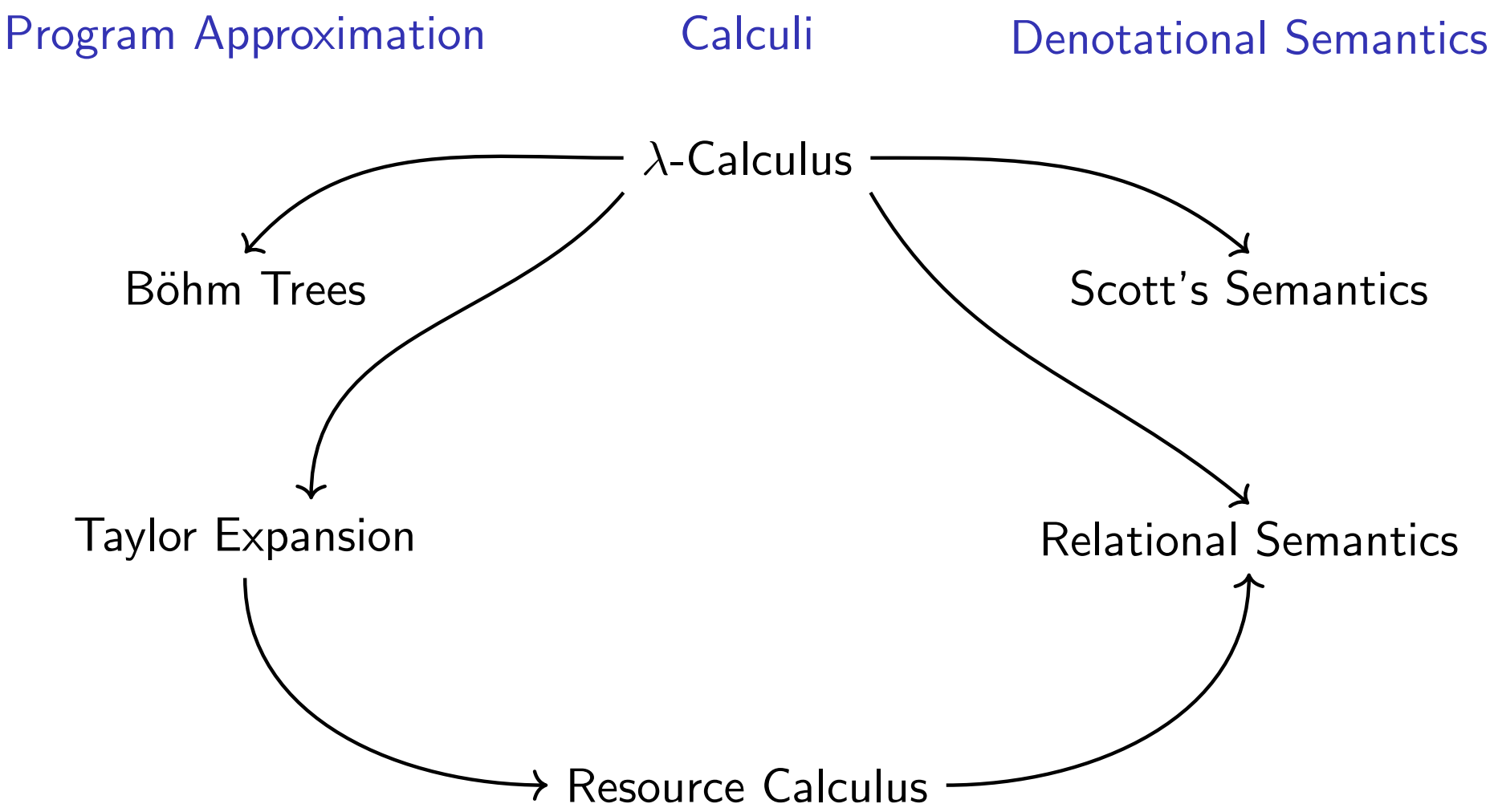
Resource Calculus

The Global Picture

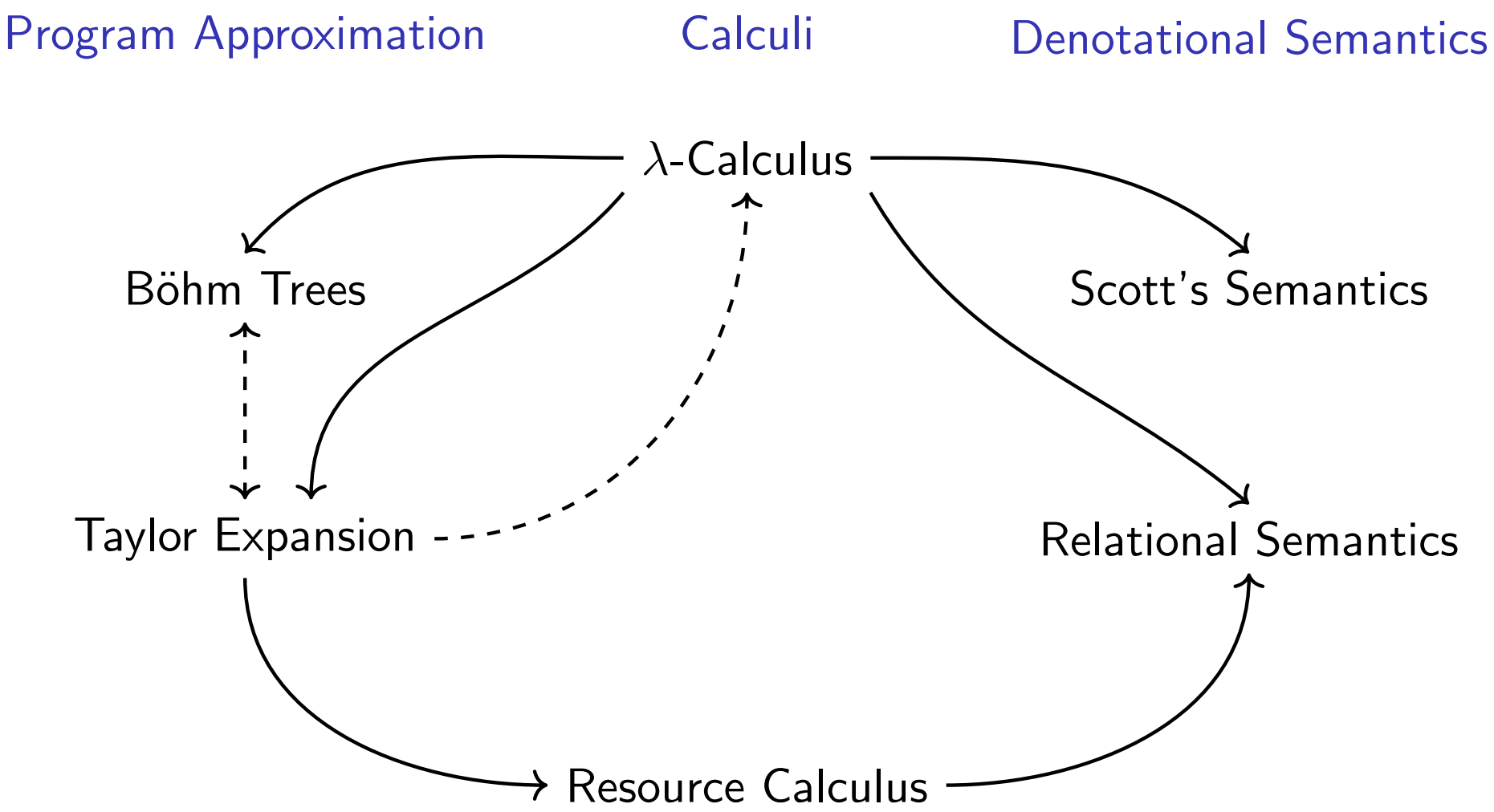
Program Approximation | Calculi | Denotational Semantics



The Global Picture



The Global Picture



Once upon a time...

The untyped λ-calculus (Church, 1932)

Based on a primitive notion of function.

$$M, N \ ::= \ x \mid \lambda x.M \mid MN$$

Computation becomes substitution

$$(\lambda x.M)N \rightarrow_\beta M\{N/x\}$$

Today:

λ-calculus is still at the core of all functional programming languages.

The Theory of Program Approximation

How to handle complex programs?

Denotational Semantics

- Model = abstract mathematical structure.
- Define a program interpretation satisfying compositionality.

$$[\![MN]\!] = [\![M]\!] \bullet [\![N]\!].$$

Systems of Approximants

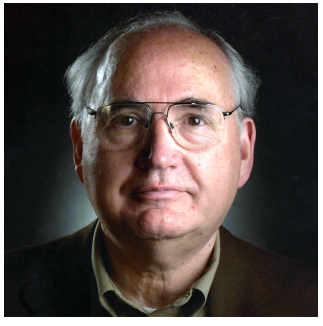
- Decompose a program into elementary "bricks"
- Retrieve the whole program behaviour performing some "limit" of its approximants.

$$M = \bigvee \{A \mid A \text{ is an approximant of } M\}$$

Denotational vs Operational Models

Continuous Semantics (Scott, 1969)

$\mathcal{D}_\infty$: First denotational model of λ-calculus.



Böhm tree semantics (Barendregt, 1977)

Tree-like representation for program execution.

"Syntactic model" of λ-calculus.

The Crucial Point — How to Handle Recursion?

Scott's continuity

"A finite portion of the output of a program must be generated by a finite portion of its input."

Kleene Fixed Point Theorem

Let $\mathcal{D} = (D, \leq, \bot)$ be a domain. Every Scott-continuous function

$$f : \mathcal{D} \to \mathcal{D}$$

has a least fixed point $\mathrm{lfp}(f)$ that can be calculated as follows:

$$\mathrm{lfp}(f) = \bigvee_{n \in \mathsf{N}} f^n(\bot)$$

Example - The factorial

The following higher-order program is not recursive:

```
fun f →
  fun n →
    if n = 0
      then
        1
      else
        n * (f (n − 1))
```

But its least fixed point gives the factorial.

Example - Fixed Point Combinators

In λ-calculus, one can define fixed point combinators:

$$\mathsf{Y} = \lambda \mathsf{f}.(\lambda \mathsf{x}.\mathsf{f}(\mathsf{xx}))(\lambda \mathsf{x}.\mathsf{f}(\mathsf{xx}))$$

Lemma. For all $M \in \Lambda$, $\mathsf{YM} =_\beta \mathsf{M}(\mathsf{YM})$.

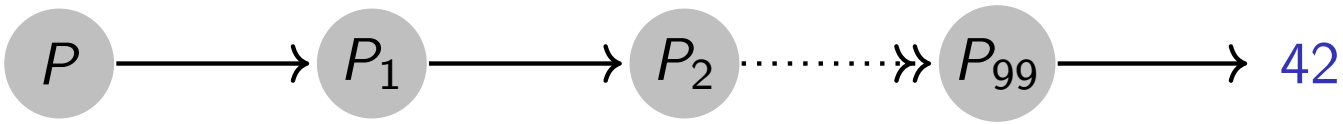
$$\begin{array}{rcl} \mathsf{YM} & = & (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))M \\ & \to_\beta & (\lambda x.M(xx))(\lambda x.M(xx)) \\ & \to_\beta & M((\lambda x.M(xx))(\lambda x.M(xx))) \\ & {}_\beta\!\leftarrow & M(\mathsf{YM}) \end{array}$$

Exercise

- Check that YM is a fixed point of M.
- Show that, for all $n \geq 0$, $\mathsf{Yf} =_\beta \mathsf{f}^\mathsf{n}(\mathsf{Yf})$.

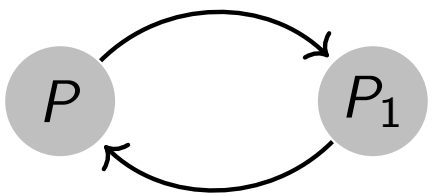
Possible behaviours of a program

Classification	Behaviour	Result
normalizable	$P \to P_1 \to P_2 \twoheadrightarrow_{97} P_{99} \to 42$	completely defined



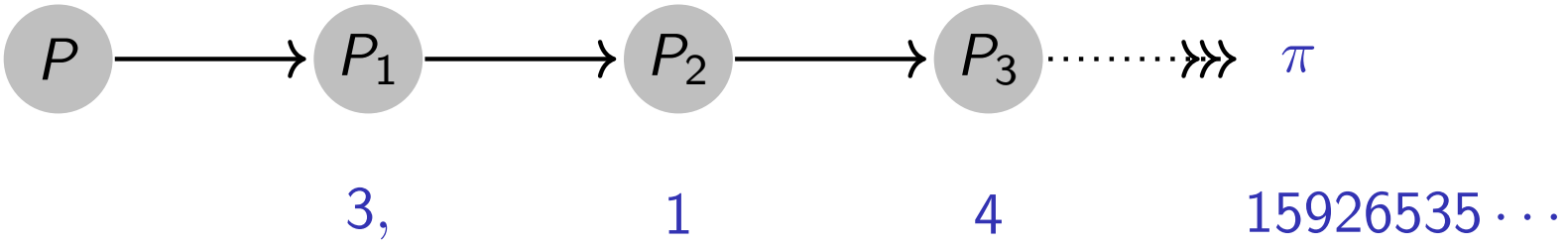
Possible behaviours of a program

Classification	Behaviour	Result
normalizable	$P \to P_1 \to P_2 \twoheadrightarrow_{97} P_{99} \to 42$	completely defined
unsolvable	$P \to P_1 \to P \twoheadrightarrow_{97} P_1 \to \cdots$	undefined



Possible behaviours of a program

Classification	Behaviour	Result
normalizable	$P \to P_1 \to P_2 \twoheadrightarrow_{97} P_{99} \to 42$	completely defined
unsolvable	$P \to P_1 \to P \twoheadrightarrow_{97} P_1 \to \cdots$	undefined
solvable	$P \to o_1 P_1 \to o_1(o_2 P_2) \to \cdots$ $\twoheadrightarrow_\infty o_1(o_2(o_3(\cdots o_n))\cdots)$	stable parts (infinitary)



The Böhm Tree Semantics

Given a λ-term M, its Böhm tree $\mathrm{BT}(M)$ is defined as follows:

- If M is completely undefined (unsolvable), then

$$\mathrm{BT}(M) = \bot,$$

- Otherwise $M \twoheadrightarrow_\beta \lambda x_1 \ldots x_n.y\, M_1 \cdots M_k$ and

$$\mathrm{BT}(M) = \lambda x_1 \ldots x_n.y$$

$$\mathrm{BT}(M_1) \quad \cdots \quad \mathrm{BT}(M_k)$$

The Böhm Tree Semantics

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

Example - Fixed point combinator Y

$$
\begin{array}{lll}
\mathsf{Y} & = & \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) \\
& \to_\beta & \lambda f.f((\lambda x.f(xx))(\lambda x.f(xx))) \\
& \to_\beta & \lambda f.f(f((\lambda x.f(xx))(\lambda x.f(xx)))) \\
& \to_\beta & \lambda f.f(f(f((\lambda x.f(xx))(\lambda x.f(xx))))) \\
& \twoheadrightarrow_\beta & \lambda f.f^n((\lambda x.f(xx))(\lambda x.f(xx))) \\
& \twoheadrightarrow_\beta & \cdots
\end{array}
$$

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

Digression — Böhm Trees as Coinductive Data-Types

Böhm-like trees are coinductively defined by:

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

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, $\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4. \ldots$

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

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)

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

Theorem (Kennaway et Al.)

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

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)

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

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?

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

$$A, A_i \ ::= \ \bot \mid \lambda x_1 \dots x_n.y \, A_1 \cdots A_k$$

- $\bot$ represents the undefined.
- the preorder $\sqsubseteq$ is generated by $\bot \sqsubseteq M$, for all $M \in \Lambda_\bot$.

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

$$\mathcal{A}(M) = \{A \in \mathcal{A} \mid \exists N \in \Lambda \,.\, M \twoheadrightarrow_\beta N \text{ and } A \sqsubseteq N\}$$

Moreover

$$\mathcal{B} \vdash M = N \iff \mathcal{A}(M) = \mathcal{A}(N)$$

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

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

$$A, A_i \ ::= \ \bot \mid \lambda x_1 \ldots x_n.y \, A_1 \cdots A_k$$

- $\bot$ represents the undefined.
- the preorder $\sqsubseteq$ is generated by $\bot \sqsubseteq M$, for all $M \in \Lambda_\bot$.

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

$$\mathcal{A}(M) = \{A \in \mathcal{A} \mid \exists N \in \Lambda \,.\, M \twoheadrightarrow_\beta N \text{ and } A \sqsubseteq N\}$$

Moreover

$$\mathcal{B} \vdash M = N \iff \mathcal{A}(M) = \mathcal{A}(N)$$

The 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

$\mathrm{BT}(\mathrm{Y})$

$=$

$\lambda f.f$

|

f

|

f

|

f

|

⋮

The Approximation Theorem

For all $M \in \Lambda$,

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

Examples:

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

Example

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

The Approximation Theorem

For all $M \in \Lambda$,

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

Examples:

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

Example

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

The Approximation Theorem

For all $M \in \Lambda$,

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

Examples:

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

Example

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

$=$

$\lambda f.f$

$|$

f

$|$

$\bot$

The Approximation Theorem

For all $M \in \Lambda$,

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

Examples:

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

Example

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

$=$

$\lambda f.f$

$|$

f

$|$

f

$|$

$\bot$

The Approximation Theorem

For all $M \in \Lambda$,

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

Examples:

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

Example

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

$=$

$\lambda f.f$

$|$

f

$|$

f

$|$

f

$|$

$\bot$

The Approximation Theorem

For all $M \in \Lambda$,

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

Examples:

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

Example

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

$=$

$\lambda f.f$

|

f

|

f

|

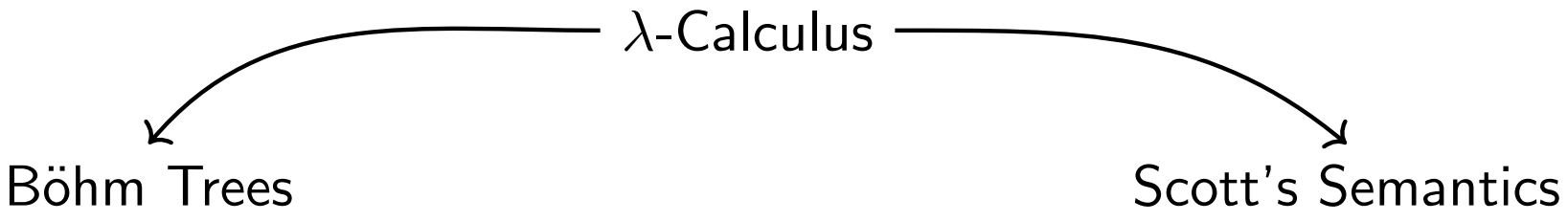
f

|

$\vdots$

The Global Picture

Program Approximation | Calculi | Denotational Semantics

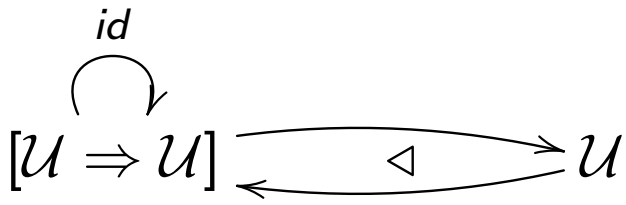


Relational Semantics

Denotational Semantics

What is a model of λ-calculus?

A reflexive object $\mathcal{U}$ in a Cartesian closed category C, i.e.,



- $[\mathcal{U} \Rightarrow \mathcal{U}] =$ exponential object.
- $\triangleleft$, intuitively, $\subseteq$

A λ-term M such that $\mathrm{FV}(M) \subseteq \vec{x}$ is interpreted as a morphism

$$|M|_{\vec{x}} : \mathcal{U}^{\vec{x}} \to \mathcal{U}$$

The relational semantics MRel

The category MRel is the coKleisli of Rel over the comonad $\mathcal{M}_f(-)$ of finite multisets.

A. Bucciarelli, T. Ehrhard, G. Manzonetto:
Not Enough Points Is Enough. CSL 2007: 298-312

The relational semantics MRel

The category MRel is the coKleisli of Rel over the comonad $\mathcal{M}_f(-)$ of finite multisets.

MRel, a concrete description:

- Objects: sets $A, B, C, \ldots$
- Arrow $f : A \to B$: any relation $f \subseteq \mathcal{M}_f(A) \times B$
- Identity: $Id_A = \{([\alpha], \alpha) \mid \alpha \in A\}$
- composition of $A \xrightarrow{f} B \xrightarrow{g} C$

$$f; g = \{(a_1 + \cdots + a_k, \gamma) \quad | \quad \begin{array}{l} \exists \beta_1, \ldots, \beta_k \in B \,.\, (a_i, \beta_i) \in f \\ ([\beta_1, \ldots, \beta_k], \gamma) \in g \end{array} \}$$

A. Bucciarelli, T. Ehrhard, G. Manzonetto:
Not Enough Points Is Enough. CSL 2007: 298-312

Theorem MRel is cartesian closed:

- (Countable) products: $A \,\&\, B := A \uplus B$ disjoint union.
- Exponential object: $[A \Rightarrow B] := \mathcal{M}_f(A) \times B$
- Seely isomorphisms: $\mathcal{M}_f(A \,\&\, B) = \mathcal{M}_f(A) \times \mathcal{M}_f(B)$

To construct a reflexive object $\mathcal{D}$, it is enough to have an injection

$$\iota : \mathcal{M}_f(D) \times D \to D$$

The resulting interpretation of M w.r.t. $x_1, \ldots, x_n \supseteq \mathrm{fv}(M)$ is

$$|M|_{\vec{x}} \subseteq \mathcal{M}_f(D)^{\vec{x}} \times D$$

Nowadays relational models are often presented via "non-idempotent" intersection type systems.

Theorem MRel is cartesian closed:

- (Countable) products: $A \,\&\, B := A \uplus B$ disjoint union.
- Exponential object: $[A \Rightarrow B] := \mathcal{M}_f(A) \times B$
- Seely isomorphisms: $\mathcal{M}_f(A \,\&\, B) = \mathcal{M}_f(A) \times \mathcal{M}_f(B)$

To construct a reflexive object $\mathcal{D}$, it is enough to have an injection

$$\iota : \mathcal{M}_f(D) \times D \to D$$

The resulting interpretation of M w.r.t. $x_1, \ldots, x_n \supseteq \mathrm{fv}(M)$ is

$$|M|_{\vec{x}} \subseteq \mathcal{M}_f(D)^{\vec{x}} \times D$$

Nowadays relational models are often presented via "non-idempotent" intersection type systems.

Relational Models as Type Systems

Let C be a set of constants. Define types ($\mathbb{T}$) and multi-types ($\mathbb{T}^!$)

$$
\begin{array}{llll}
(\mathbb{T}_C) & \alpha, \beta & ::= & a \mid \sigma \to \alpha \\
(\mathbb{T}^!_C) & \sigma, \tau & ::= & [\alpha_1, \ldots, \alpha_n] \qquad (n \geq 0)
\end{array}
$$

- multi-types are only present at the left hand-sight of a arrow
- the empty multi-type is denoted by [].
- [] is not a type.

F. Breuvart, G. Manzonetto, D. Ruoppolo: Relational Graph Models at Work. Log. Methods Comput. Sci. 14(3) (2018)

Environments

An environment Γ is a function

$$\Gamma : \mathit{Var} \to \mathbb{T}^!$$

whose support $\mathrm{supp}(\Gamma) = \{x \mid \Gamma(x) \neq []\}$ is finite.

Given envoronments Γ_1, Γ_2, define:

$$(\Gamma_1 + \Gamma_2)(x) = \Gamma_1(x) + \Gamma_2(x)$$

Notation. $x_1 : \sigma_1, \ldots, x_n : \sigma_n$ represents the environment

$$\Gamma(y) = \begin{cases} \sigma_i & \text{if } y = x_i, \\ [], & \text{if } y \notin \vec{x}. \end{cases}$$

The inference rules

$$\frac{}{x : [\alpha] \vdash x : \alpha} \qquad \frac{\Gamma, x : \sigma \vdash M : \alpha}{\Gamma \vdash \lambda x.M : \sigma \to \alpha} \qquad \frac{\Gamma \vdash M : \alpha \quad \alpha \simeq \beta}{\Gamma \vdash M : \beta}$$

$$\frac{\Gamma_0 \vdash M : [\beta_1, \ldots, \beta_n] \to \alpha \quad \Gamma_1 \vdash N : \beta_1 \quad \cdots \quad \Gamma_n \vdash N : \beta_n}{\sum_{i=0}^{n} \Gamma_i \vdash MN : \alpha}$$

where $\simeq$ is any equivalence over $\mathbb{T}$ (resp. $\mathbb{T}^!$) satisfying

$$\sigma \to \alpha \simeq \tau \to \beta \iff \sigma \simeq \tau \text{ and } \alpha \simeq \beta$$

$$[\alpha_1, \ldots, \alpha_n] \simeq [\beta_1, \ldots, \beta_m] \iff m = n \text{ and } \forall i \,.\, \alpha_i \simeq \beta_i$$

A model $\mathcal{D}$ is uniquely determined by $\langle C, \simeq \rangle$.

Properties

Assume

$$M \to_\beta N$$

Then

- Subject Reduction

$$\Gamma \vdash M : \alpha \quad \Rightarrow \quad \Gamma \vdash N : \alpha$$

- Subject Expansion

$$\Gamma \vdash N : \alpha \quad \Rightarrow \quad \Gamma \vdash M : \alpha$$

Examples I

We have:

$$\dfrac{\dfrac{\overline{x : [[\alpha] \to \beta] \vdash x : [\alpha] \to \beta} \quad \overline{x : [\alpha] \vdash x : \alpha}}{x : [\alpha, [\alpha] \to \beta] \vdash xx : \beta}}{\vdash \lambda x.xx : [\alpha, [\alpha] \to \beta] \to \beta}$$

but Ω is not typable! (Exercise)

More generally:

- All head normal forms are typable (wlog, $y \notin \vec{x}$):

$$y : [[]^k \to \alpha] \vdash \lambda x_1 \dots x_n.yM_1 \cdots M_k : []^n \to \alpha$$

- By subject expansion, all solvable are typable.
- No unsolvable is typable.

Examples I

We have:

$$\dfrac{\dfrac{\overline{x : [[\alpha] \to \beta] \vdash x : [\alpha] \to \beta} \quad \overline{x : [\alpha] \vdash x : \alpha}}{x : [\alpha, [\alpha] \to \beta] \vdash xx : \beta}}{\vdash \lambda x.xx : [\alpha, [\alpha] \to \beta] \to \beta}$$

but Ω is not typable! (Exercise)

More generally:

- All head normal forms are typable (wlog, $y \notin \vec{x}$):

$$y : [[]^k \to \alpha] \vdash \lambda x_1 \dots x_n.yM_1 \cdots M_k : []^n \to \alpha$$

- By subject expansion, all solvable are typable.
- No unsolvable is typable.

Examples II

Consider the model having 1 atomic type $C = \{\varepsilon\}$ and the equivalence $\simeq$ on $\mathbb{T}_\varepsilon$ generated by

$$\varepsilon \simeq [] \to \varepsilon$$

Let J be a combinator satisfying

$$\mathsf{J}x \twoheadrightarrow_\beta \lambda z_0.x(\mathsf{J}z_0) \twoheadrightarrow_\beta \lambda z_0.x(\lambda z_1.z_0(\mathsf{J}z_1)) \twoheadrightarrow_\beta \cdots$$

Then, we have:

$$\vdash \mathsf{J} : [[] \to \varepsilon] \to [] \to \varepsilon$$

Examples II

Consider the model having 1 atomic type $C = \{\varepsilon\}$ and the equivalence $\simeq$ on $\mathbb{T}_\varepsilon$ generated by

$$\varepsilon \simeq [] \to \varepsilon$$

Let J be a combinator satisfying

$$\mathsf{J}x \twoheadrightarrow_\beta \lambda z_0.x(\mathsf{J}z_0) \twoheadrightarrow_\beta \lambda z_0.x(\lambda z_1.z_0(\mathsf{J}z_1)) \twoheadrightarrow_\beta \cdots$$

Then, we have:

$$\dfrac{\dfrac{x : [[] \to \varepsilon] \vdash x : [] \to \varepsilon}{x : [[] \to \varepsilon], z_0 : [] \vdash x(\mathsf{J}z_0) : \varepsilon}}{\vdash \mathsf{J} =_\beta \lambda x z_0.x(\mathsf{J}z_0) : [[] \to \varepsilon] \to [] \to \varepsilon}$$

Examples II

Consider the model having 1 atomic type $C = \{\varepsilon\}$ and the equivalence $\simeq$ on $\mathbb{T}_\varepsilon$ generated by

$$\varepsilon \simeq [] \to \varepsilon$$

Let J be a combinator satisfying

$$\mathsf{J}x \twoheadrightarrow_\beta \lambda z_0.x(\mathsf{J}z_0) \twoheadrightarrow_\beta \lambda z_0.x(\lambda z_1.z_0(\mathsf{J}z_1)) \twoheadrightarrow_\beta \cdots$$

Then, we have:

$$\dfrac{\dfrac{x : [[] \to \varepsilon] \vdash x : [] \to \varepsilon}{x : [[] \to \varepsilon], z_0 : [] \vdash x(\mathsf{J}z_0) : \varepsilon}}{\vdash \mathsf{J} =_\beta \lambda x z_0.x(\mathsf{J}z_0) : [[] \to \varepsilon] \to [] \to \varepsilon \simeq [\varepsilon] \to \varepsilon}$$

So, in this model $[\![\mathsf{J}]\!] = [\![\mathsf{I}]\!] = \{\beta \mid \beta \simeq [\alpha] \to \alpha, \alpha \in \mathbb{T}_\varepsilon\}$

Interpretation

Define the interpretation of M in $\mathcal{D}$ as:

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

Write $\mathcal{D} \models M = N$ whenever $[\![M]\!] = [\![N]\!]$.

Theorem (Soundness)

For all $M, N \in \Lambda$

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

Proof. It follows from subject expansion + subject reduction.

Typed vs Untyped occurrences

In the following derivation, the subterm Ω is not typed:

$$\dfrac{\dfrac{x : [[] \to \alpha] \vdash x : [] \to \alpha}{x : [[] \to \alpha] \vdash x\Omega : \alpha}}{\lambda x.x\Omega : [[] \to \alpha] \to \alpha}$$

This derivation is in typed normal form.

On the contrary, the redex occurrence $\mathsf{I}x$ is typed in:

$$\dfrac{\dfrac{x : [[\alpha] \to \alpha] \vdash x : [\alpha] \to \alpha \qquad \dfrac{\vdash \mathsf{I} : [\alpha] \to \alpha \quad x : [\alpha] \vdash x : \alpha}{x : [\alpha] \vdash \mathsf{I}x : \alpha}}{x : [[\alpha] \to \alpha, \alpha] \vdash x\Omega : \alpha}}{\lambda x.x(\mathsf{I}x) : [[\alpha] \to \alpha, \alpha] \to \alpha}$$

Typed vs Untyped occurrences

In the following derivation, the subterm Ω is not typed:

$$\dfrac{\dfrac{x : [[] \to \alpha] \vdash x : [] \to \alpha}{x : [[] \to \alpha] \vdash x\Omega : \alpha}}{\lambda x.x\Omega : [[] \to \alpha] \to \alpha}$$

This derivation is in typed normal form.

On the contrary, the redex occurrence $\mathsf{I}x$ is typed in:

$$\dfrac{\dfrac{x : [[\alpha] \to \alpha] \vdash x : [\alpha] \to \alpha \qquad \dfrac{\vdash \mathsf{I} : [\alpha] \to \alpha \quad x : [\alpha] \vdash x : \alpha}{x : [\alpha] \vdash \mathsf{I}x : \alpha}}{x : [[\alpha] \to \alpha, \alpha] \vdash x\Omega : \alpha}}{\lambda x.x(\mathsf{I}x) : [[\alpha] \to \alpha, \alpha] \to \alpha}$$

Weighted Subject Reduction

If the redex contracted in $M \to_\beta N$ is typed in

$$\frac{\Pi}{\Gamma \vdash M : \alpha}$$

then, there exists a derivation Π' of

$$\frac{\Pi'}{\Gamma \vdash N : \alpha}$$

such that $|\Pi| < |\Pi'|$.

Let Π be a derivation of $\Gamma \vdash M : \alpha$ in typed nf

Associate an approximant $A_\Pi \in \mathcal{A}$ s.t. $\Gamma \vdash A_\Pi : \alpha$ and $A_\Pi \sqsubseteq M$ by

$$x : [\alpha] \vdash x : \alpha \qquad \Rightarrow A_\Pi = x$$

$$\frac{\dfrac{\Pi'}{\Gamma, x : \sigma \vdash M : \alpha}}{\Gamma \vdash \lambda x.M : \sigma \to \alpha} \qquad \Rightarrow A_\Pi = \lambda x.A_{\Pi'}$$

$$\frac{\dfrac{\Pi'}{\Gamma \vdash M : \alpha} \quad \alpha \simeq \beta}{\Gamma \vdash M : \beta} \qquad \Rightarrow A_\Pi = A_{\Pi'}$$

$$\frac{\dfrac{\Pi_0}{\Gamma_0 \vdash M : [\beta_1, \ldots, \beta_n] \to \alpha} \quad \dfrac{\Pi_i}{\Gamma_i \vdash N : \beta_i}}{\sum_{i=0}^{n} \Gamma_i \vdash MN : \alpha} \qquad \Rightarrow A_\Pi = A_{\Pi_0}\left(\bigsqcup_{i=1}^{n} A_{\Pi_i}\right)$$

Note that $\bigsqcup_{i=1}^{0} A_i = \bot$

The Approximation Theorem

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

$$\Gamma \vdash M : \alpha \iff \exists A \in \mathcal{A}(M) \,.\, \Gamma \vdash M : \alpha$$

Proof. ($\Rightarrow$) Let Π be a derivation of $\Gamma \vdash M : \alpha$ not in typed nf.

- Then, by the Weighted Subject Reduction,

$$M = M_0 \to_\beta M_1 \twoheadrightarrow_\beta M_n = N$$

  and there exists a derivation Π' of $\Gamma \vdash N : \alpha$ in typed nf.
- Therefore, $\Gamma \vdash A_{\Pi'} : \alpha$ with $A_{\Pi'} \sqsubseteq N$.
- Since $M \twoheadrightarrow_\beta N$ and $A_{\Pi'} \sqsubseteq N$, we conclude $A_{\Pi'} \in \mathcal{A}(M)$

($\Leftarrow$) Easy. □

 A. Bucciarelli, D. Kesner, S. Ronchi Della Rocca: Inhabitation for non-idempotent Intersection types. LMCS. 14(3) 2018

The Approximation Theorem and Its Consequences

Theorem (Approximation Theorem)

For $M \in \Lambda$, we have

$$[\![M]\!] = \bigcup_{A \in \mathcal{A}(M)} [\![A]\!]$$

Corollary 1. M is unsolvable $\iff [\![M]\!] = \emptyset$

- M unsolvable $\Rightarrow \mathcal{A}(M) = \bot$. So, $[\![M]\!] = [\![\bot]\!] = \emptyset$.
- M solvable $\Rightarrow$ M typable $\Rightarrow [\![M]\!] \neq \emptyset$

Corollary 2. $\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad [\![M]\!] = [\![N]\!]$.

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

Comparison

Filter Models

- Intersection type systems with weakening
- solvable $\iff$ typable $\neq \omega$
- Approximation Theorem Reducibility candidates

Relational Models

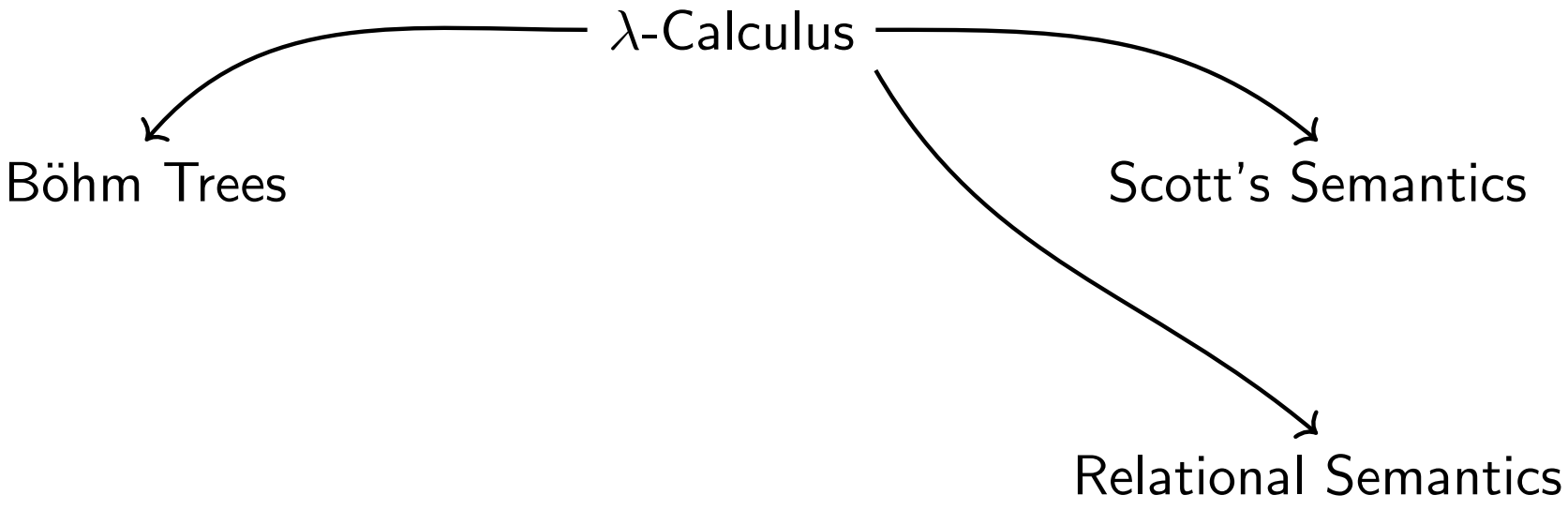
- tensor type systems without weakening
- solvable $\iff$ typable
- Approximation Theorem Easy induction.

Impredicative techniques vs quantitative proofs

Why?

The Global Picture

Program Approximation | Calculi | Denotational Semantics



Resource Calculus

The Resource Calculus

Lambda Calculus is not resource conscious

In one step of β-reduction

$$(\lambda x.M)N \to_\beta M\{N/x\}$$

the argument N can be:

- Erased: $(\lambda xy.y)N \to_\beta \lambda y.y$
- Duplicated: $(\lambda x.xx)N \to_\beta NN$
- Copied an arbitrary number of times:

$$(\lambda fz.f(f(\cdots f(z))))N \to_\beta \lambda z.N(N(\cdots N(z)))$$

Linear Logic is resource sensitive

Linear Logic decomposes the intuitionistic arrow

$$A \to B \qquad \text{as} \qquad !A \multimap B$$

and this suggests that one step of β-reduction

$$(\lambda x.M)N \to_\beta M\{N/x\}$$

should be decomposable into more elementary steps.

Linear Lambda Calculus is ~~stupid~~ extremely basic

The naïve calculus arising from linearity

$$\lambda x.M \quad \Rightarrow \quad x \text{ occurs exactly once in } M$$

is not very interesting from the operational point of view.

$$\lambda xyz.xyz,\ \lambda xyz.yzx,\ \lambda xyz.zxy,\ \lambda xyz.x(\lambda f.yf)(\lambda g.gz), \ldots$$

It can be interesting from a combinatorial perspective:

 Noam Zeilberger. Counting isomorphism classes of β-normal linear lambda terms. arXiv:1509.07596 (2015)

 Noam Zeilberger: Linear lambda terms as invariants of rooted trivalent maps. J. Funct. Program. 26: e21 (2016)

The Resource Calculus

It is a resource sensitive version of λ-calculus where

- variables can occur multiple times in its programs,
- resources cannot be erased nor copied during the reduction.

Introduced in


T. Ehrhard, L. Regnier: The differential lambda-calculus. Theor. Comput. Sci. 309(1-3): 1-41 (2003)

More understandable syntax in


M. Pagani, P. Tranquilli: Parallel Reduction in Resource Lambda-Calculus. APLAS 2009: 226-242

Ancestor


G. Boudol: The Lambda-Calculus with Multiplicities. CONCUR 1993: 1-6

Disclaimer

Today I will present the so-called:

"Finitary Resource Calculus"

aka, the promotion-free fragment of the Full Resource Calculus.

The previous articles concern the Full Resource Calculus:

- Same basic concepts, but
- More complicated, because of
  - interaction between linear and replicable resources
- Also, more expressive.

Its syntax

Syntactic categories:

Terms	s, t, u	$::=$	$x \mid \lambda x.t \mid tb$	Λ^r
Bags	b	$::=$	$[t_1, \ldots, t_n],\ \text{for}\ n \geq 0,$	Λ^b
Formal sums	$\mathbb{S}, \mathbb{T}, \mathbb{U}$	$::=$	$0 \mid t + \mathbb{T}$	$\mathsf{N}\langle\Lambda^r\rangle$

Intuitively

- Terms are the protagonists of our calculus.
- Bags are multisets of linear resources.
- Sums represent non-deterministic choice between terms

$$s + t \to s \qquad\qquad s + t \to t$$

except that the choice is never actually made.

Assumptions

On formal sums

- The operator $+$ is associative and commutative.
- As usual, we write $\sum_{i=1}^{k} t_i = t_1 + \cdots + t_k$

Bags are multisets represented in multiplicative notation.

- 1 is the empty bag.
- $b_1 \cdot b_2$ represents the multiset union of b_1 and b_2.
- Structural induction on bags, becomes:
  - 1, base case.
  - $[t] \cdot b$, induction step.

Sums of bags $\mathbb{B} \in \mathrm{N}\langle\Lambda^b\rangle$ are unimportant, but useful if you want the induction to pass through.

All constructors are linear

In this context

$$\text{“linearity”} \simeq \text{commutation with sums}$$

Notation.
For sums in $\mathsf{N}\langle\Lambda^r\rangle$, we introduce a syntactic sugar:

$$\begin{array}{lcl} \lambda x.\sum_{i=1}^n t_i & := & \sum_{i=1}^n \lambda x.t_i \\ (\sum_{i=1}^n t_i)b & := & \sum_{i=1}^n t_i b \\ t(\sum_{i=1}^n b_i) & := & \sum_{i=1}^n t b_i \end{array}$$

In other words, sums can always be pushed to surface.

Remark. A subterm 0 annihilates the whole term:

$$\lambda x.0 = t0 = 0b = 0 \qquad \text{and} \qquad [0]\cdot b = 0$$

All constructors are linear

In this context

$$\text{“linearity”} \simeq \text{commutation with sums}$$

Notation.

For sums in $\mathsf{N}\langle\Lambda^r\rangle$, we introduce a syntactic sugar:

$$\begin{array}{lcl} \lambda x.\sum_{i=1}^n t_i & := & \sum_{i=1}^n \lambda x.t_i \\ (\sum_{i=1}^n t_i)b & := & \sum_{i=1}^n t_i b \\ t(\sum_{i=1}^n b_i) & := & \sum_{i=1}^n t b_i \end{array}$$

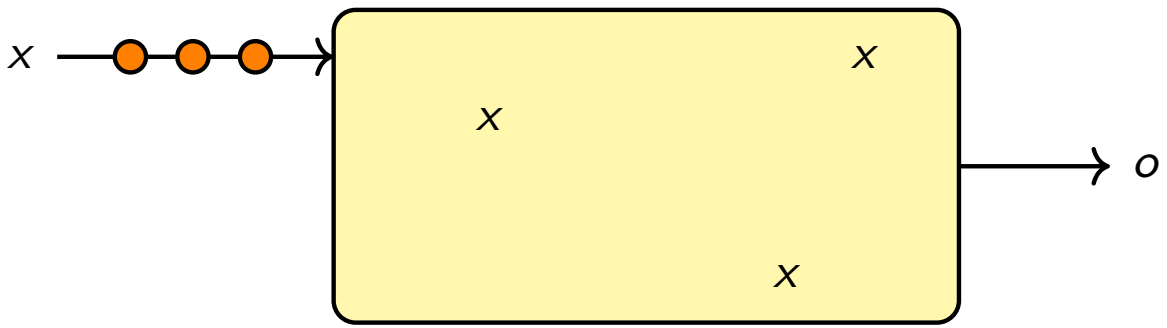
In other words, sums can always be pushed to surface.

Remark. A subterm 0 annihilates the whole term:

$$\lambda x.0 = t0 = 0b = 0 \qquad \text{and} \qquad [0]\cdot b = 0$$

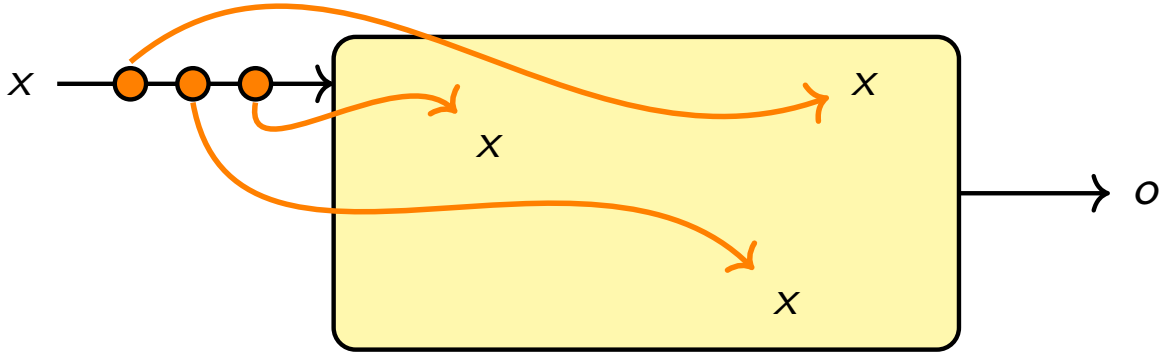
Its operational semantics – the idea

$$(\lambda x.t)[s_1, s_2, s_3] \rightarrow \ ?$$



Its operational semantics – the idea

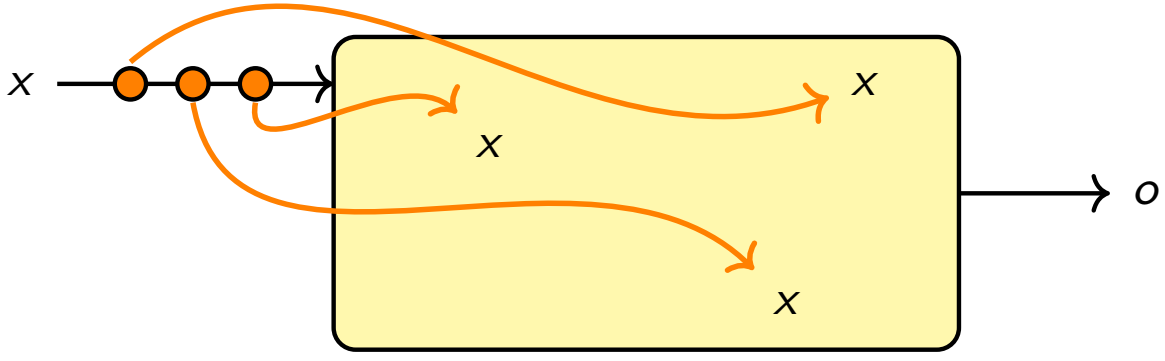
$$(\lambda x.t)[s_1, s_2, s_3] \to t\langle s_1/x_1, s_2/x_2, s_3/x_3\rangle$$



$\deg_x(x)$	$=$	1	$\deg_x(y)$	$=$	0
$\deg_x(\lambda y.t)$	$=$	$\deg_x(t)$	$\deg_x(tb)$	$=$	$\deg_x(t) + \deg_x(b)$
$\deg_x(1)$	$=$	0	$\deg_x([t] \cdot b)$	$=$	$\deg_x(t) + \deg_x(b)$

Its operational semantics – the idea

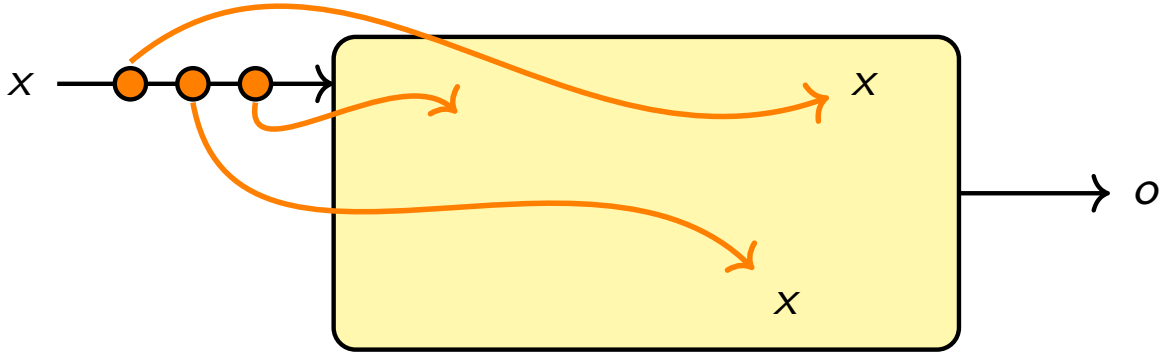
$$(\lambda x.t)[s_1, s_2, s_3] \to \sum_{\sigma \in \mathfrak{S}_3} t\langle s_1/x_{\sigma(1)}, s_2/x_{\sigma(2)}, s_3/x_{\sigma(3)}\rangle$$



$\deg_x(x)$	$=$	1	$\deg_x(y)$	$=$	0
$\deg_x(\lambda y.t)$	$=$	$\deg_x(t)$	$\deg_x(tb)$	$=$	$\deg_x(t) + \deg_x(b)$
$\deg_x(1)$	$=$	0	$\deg_x([t] \cdot b)$	$=$	$\deg_x(t) + \deg_x(b)$

Its operational semantics – the idea

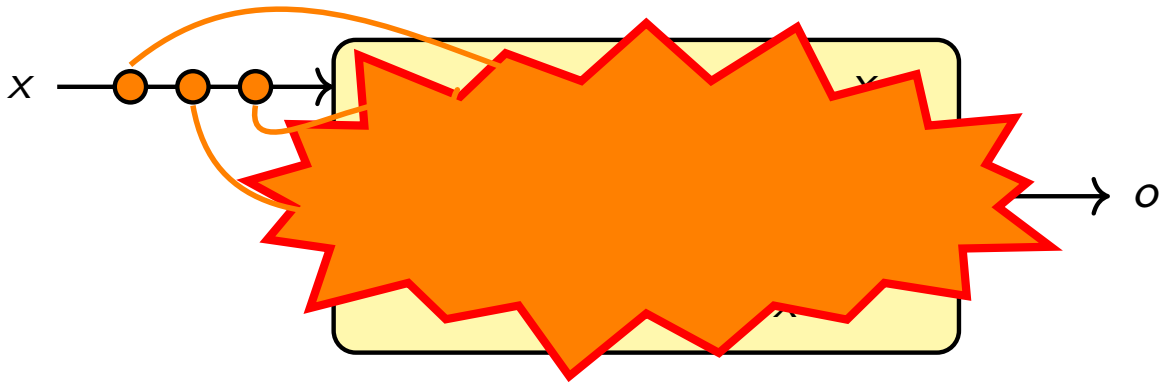
$$(\lambda x.t)[s_1, s_2, s_3] \rightarrow \ ?$$



$\deg_x(x)$	$=$	1	$\deg_x(y)$	$=$	0
$\deg_x(\lambda y.t)$	$=$	$\deg_x(t)$	$\deg_x(tb)$	$=$	$\deg_x(t) + \deg_x(b)$
$\deg_x(1)$	$=$	0	$\deg_x([t] \cdot b)$	$=$	$\deg_x(t) + \deg_x(b)$

Its operational semantics – the idea

$$(\lambda x.t)[s_1, s_2, s_3] \to 0$$



$\deg_x(x)$	$=$	1	$\deg_x(y)$	$=$	0
$\deg_x(\lambda y.t)$	$=$	$\deg_x(t)$	$\deg_x(tb)$	$=$	$\deg_x(t) + \deg_x(b)$
$\deg_x(1)$	$=$	0	$\deg_x([t] \cdot b)$	$=$	$\deg_x(t) + \deg_x(b)$

Linear substitution

For $s, t \in \Lambda^r$, define $t\langle s/x\rangle \in \mathsf{N}\langle\Lambda^r\rangle$ aka the

linear substitution of s for one occurrence of x in t

$$
\begin{aligned}
y\langle s/x\rangle &= \begin{cases} s, & \text{if } x = y, \\ 0, & \text{otherwise,} \end{cases} \\
(\lambda y.t)\langle s/x\rangle &= \lambda y.t\langle s/x\rangle \qquad (\text{wlog. } x \neq y) \\
(tb)\langle s/x\rangle &= t\langle s/x\rangle b + t(b\langle s/x\rangle)
\end{aligned}
$$

on bags:

$$
\begin{aligned}
1\langle s/x\rangle &= 0 \\
[t]\langle s/x\rangle &= [t\langle s/x\rangle] \\
(b_1 \cdot b_2)\langle s/x\rangle &= b_1\langle s/x\rangle \cdot b_2 + b_1 \cdot (b_2\langle s/x\rangle)
\end{aligned}
$$

Linear substitution

For $s, t \in \Lambda^r$, define $t\langle s/x\rangle \in \mathsf{N}\langle\Lambda^r\rangle$ aka the

linear substitution of s for one occurrence of x in t

$$
\begin{aligned}
y\langle s/x\rangle &= \begin{cases} s, & \text{if } x = y, \\ 0, & \text{otherwise,} \end{cases} \\
(\lambda y.t)\langle s/x\rangle &= \lambda y.t\langle s/x\rangle \qquad (\text{wlog. } x \neq y) \\
(tb)\langle s/x\rangle &= t\langle s/x\rangle b + t(b\langle s/x\rangle)
\end{aligned}
$$

on bags:

$$
\begin{aligned}
1\langle s/x\rangle &= 0 \\
[t]\langle s/x\rangle &= [t\langle s/x\rangle] \\
(b_1 \cdot b_2)\langle s/x\rangle &= b_1\langle s/x\rangle \cdot b_2 + b_1 \cdot (b_2\langle s/x\rangle)
\end{aligned}
$$

Commutation of linear substitutions

Lemma (Schwartz's Lemma)

For $s, t, u \in \Lambda^r$, and variables x, y such that $y \notin \mathrm{fv}(s)$, we have

$$t\langle u/y\rangle\langle s/x\rangle = t\langle s/x\rangle\langle u/y\rangle + t\langle u\langle s/x\rangle/y\rangle.$$

Example. Take $t := x[y, x]$, $u := \mathsf{T}[\mathsf{x}]$ and $s = \mathsf{F}$. Then

$$\begin{array}{lcl}
(x[y,x])\langle \mathsf{T}[\mathsf{x}]/\mathsf{y}\rangle\langle \mathsf{F}/\mathsf{x}\rangle & = & (x[\mathsf{T}[\mathsf{x}],\mathsf{x}])\langle \mathsf{F}/\mathsf{x}\rangle \\
 & = & \mathsf{F}[\mathsf{T}[\mathsf{x}],\mathsf{x}] + \mathsf{x}[\mathsf{T}[\mathsf{F}],\mathsf{x}] + \mathsf{x}[\mathsf{T}[\mathsf{x}],\mathsf{F}]
\end{array}$$

On the other side:

$$\begin{array}{lcl}
(x[y,x])\langle \mathsf{F}/\mathsf{x}\rangle\langle \mathsf{T}[\mathsf{x}]/\mathsf{y}\rangle & + & (x[y,x])\langle \mathsf{T}[\mathsf{x}]\langle \mathsf{F}/\mathsf{x}\rangle/\mathsf{y}\rangle = \\
(\mathsf{F}[\mathsf{y},\mathsf{x}] + \mathsf{x}[\mathsf{y},\mathsf{F}])\langle \mathsf{T}[\mathsf{x}]/\mathsf{y}\rangle & + & (x[y,x])\langle \mathsf{T}[\mathsf{F}]/\mathsf{y}\rangle = \\
\mathsf{F}[\mathsf{T}[\mathsf{x}],\mathsf{x}] + \quad \mathsf{x}[\mathsf{T}[\mathsf{x}],\mathsf{F}] & + & x[\mathsf{T}[\mathsf{F}],\mathsf{x}]
\end{array}$$

Commutation of linear substitutions

Lemma (Schwartz's Lemma)

For $s, t, u \in \Lambda^r$, and variables x, y such that $y \notin \mathrm{fv}(s)$, we have

$$t\langle u/y\rangle\langle s/x\rangle = t\langle s/x\rangle\langle u/y\rangle + t\langle u\langle s/x\rangle/y\rangle.$$

Example. Take $t := x[y, x]$, $u := \mathsf{T}[\mathsf{x}]$ and $s = \mathsf{F}$. Then

$$\begin{array}{lcl}
(x[y,x])\langle \mathsf{T}[\mathsf{x}]/\mathsf{y}\rangle\langle \mathsf{F}/\mathsf{x}\rangle & = & (x[\mathsf{T}[\mathsf{x}],\mathsf{x}])\langle \mathsf{F}/\mathsf{x}\rangle \\
 & = & \mathsf{F}[\mathsf{T}[\mathsf{x}],\mathsf{x}] + \mathsf{x}[\mathsf{T}[\mathsf{F}],\mathsf{x}] + \mathsf{x}[\mathsf{T}[\mathsf{x}],\mathsf{F}] \\
\text{On the other side:} & & \\
(x[y,x])\langle \mathsf{F}/\mathsf{x}\rangle\langle \mathsf{T}[\mathsf{x}]/\mathsf{y}\rangle & + & (x[y,x])\langle \mathsf{T}[\mathsf{x}]\langle \mathsf{F}/\mathsf{x}\rangle/\mathsf{y}\rangle = \\
(\mathsf{F}[\mathsf{y},\mathsf{x}] + \mathsf{x}[\mathsf{y},\mathsf{F}])\langle \mathsf{T}[\mathsf{x}]/\mathsf{y}\rangle & + & (x[y,x])\langle \mathsf{T}[\mathsf{F}]/\mathsf{y}\rangle = \\
\mathsf{F}[\mathsf{T}[\mathsf{x}],\mathsf{x}] + \quad \mathsf{x}[\mathsf{T}[\mathsf{x}],\mathsf{F}] & + & x[\mathsf{T}[\mathsf{F}],\mathsf{x}]
\end{array}$$

Its operational semantics

Baby-step. Define $\to_{\mathsf{b}} \subseteq \Lambda^r \times \mathsf{N}\langle\Lambda^r\rangle$ by

$$(\lambda x.t)([s]\cdot b) \to_{\mathsf{b}} (\lambda x.t\langle s/x\rangle)b$$

$$(\lambda x.t)1 \to_{\mathsf{b}} \begin{cases} t, & \text{if } x \notin \mathsf{fv}(t), \\ 0, & \text{otherwise.} \end{cases}$$

Normal step. Define $\to_{\mathsf{r}} \subseteq \Lambda^r \times \mathsf{N}\langle\Lambda^r\rangle$ by

$$(\lambda x.t)[s_1,\dots,s_n] \to_{\mathsf{r}} \begin{cases} \sum_{\sigma\in\mathfrak{G}_n} t\{s_1/x_{\sigma(1)},\dots,s_n/x_{\sigma(n)}\}, & \text{if } \deg_x(t) = n, \\ 0, & \text{otherwise.} \end{cases}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathrm{N}\langle\Lambda^r\rangle \times \mathrm{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Baby-steps:

$$\begin{array}{rl} & (\lambda xy.x[y,y])[a][b,b] \\ \to_{\mathsf{b}} & (\lambda xy.a[y,y])1[b,b] \\ \to_{\mathsf{b}} & (\lambda y.a[y,y])[b,b] \end{array}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathsf{N}\langle\Lambda^r\rangle \times \mathsf{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Baby-steps:

$$\begin{aligned} & (\lambda xy.x[y,y])[a][b,b] \\ \to_{\mathsf{b}} \quad & (\lambda xy.a[y,y])1[b,b] \\ \to_{\mathsf{b}} \quad & (\lambda y.a[y,y])[b,b] \\ \to_{\mathsf{b}} \quad & (\lambda y.a[b,y])[b] + (\lambda y.a[y,b])[b] = 2.(\lambda y.a[b,y])[b] \end{aligned}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathrm{N}\langle\Lambda^r\rangle \times \mathrm{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Baby-steps:

$$\begin{aligned}
& (\lambda xy.x[y,y])[a][b,b] \\
\to_{\mathrm{b}} \quad & (\lambda xy.a[y,y])1[b,b] \\
\to_{\mathrm{b}} \quad & (\lambda y.a[y,y])[b,b] \\
\to_{\mathrm{b}} \quad & (\lambda y.a[b,y])[b] + (\lambda y.a[y,b])[b] = 2.(\lambda y.a[b,y])[b] \\
\to_{\mathrm{b}} \quad & (\lambda y.a[b,b])1 + (\lambda y.a[b,y])[b]
\end{aligned}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathrm{N}\langle\Lambda^r\rangle \times \mathrm{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Baby-steps:

$$\begin{aligned}
& (\lambda xy.x[y,y])[a][b,b] \\
\to_{\mathsf{b}} \quad & (\lambda xy.a[y,y])1[b,b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[y,y])[b,b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[b,y])[b] + (\lambda y.a[y,b])[b] = 2.(\lambda y.a[b,y])[b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[b,b])1 + (\lambda y.a[b,y])[b] \\
\to_{\mathsf{b}} \quad & 2.(\lambda y.a[b,b])1
\end{aligned}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathrm{N}\langle\Lambda^r\rangle \times \mathrm{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Baby-steps:

$$\begin{aligned}
& (\lambda xy.x[y,y])[a][b,b] \\
\to_{\mathsf{b}} \quad & (\lambda xy.a[y,y])1[b,b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[y,y])[b,b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[b,y])[b] + (\lambda y.a[y,b])[b] = 2.(\lambda y.a[b,y])[b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[b,b])1 + (\lambda y.a[b,y])[b] \\
\to_{\mathsf{b}} \quad & 2.(\lambda y.a[b,b])1 \\
\overset{2}{\to}_{\mathsf{b}} \quad & 2.a[b,b] = a[b,b] + a[b,b]
\end{aligned}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathrm{N}\langle\Lambda^r\rangle \times \mathrm{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Baby-steps:

$$\begin{aligned}
& (\lambda xy.x[y,y])[a][b,b] \\
\to_{\mathsf{b}} \quad & (\lambda xy.a[y,y])1[b,b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[y,y])[b,b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[b,y])[b] + (\lambda y.a[y,b])[b] = 2.(\lambda y.a[b,y])[b] \\
\to_{\mathsf{b}} \quad & (\lambda y.a[b,b])1 + (\lambda y.a[b,y])[b] \\
\to_{\mathsf{b}} \quad & 2.(\lambda y.a[b,b])1 \\
\overset{2}{\to}_{\mathsf{b}} \quad & 2.a[b,b] = a[b,b] + a[b,b]
\end{aligned}$$

Examples of reductions

Both notions of reduction extend to $\to \subseteq \mathsf{N}\langle\Lambda^r\rangle \times \mathsf{N}\langle\Lambda^r\rangle$ by

$$t \to t' \quad \Rightarrow \quad t + \mathbb{T} \to t' + \mathbb{T}$$

- Normal-steps:

$$\begin{array}{lll} (\lambda xy.x[y,y])[a][b,b] & \to_{\mathsf{r}} & (\lambda y.a[y,y])[b,b] \\ & \to_{\mathsf{r}} & 2.a[b,b] \end{array}$$

Theorem (Equivalence baby $\leftrightarrow$ normal reductions)

- *If* $t \to_{\mathsf{r}} \mathbb{T}$ *then* $t \twoheadrightarrow_{\mathsf{b}} \mathbb{T}$.
- *For* $\mathbb{T} = t_1 + \cdots + t_n$ *with each* t_i *in normal form, we have:*

$$t \twoheadrightarrow_{\mathsf{b}} \mathbb{T} \iff t \twoheadrightarrow_{\mathsf{r}} \mathbb{T}$$

Main Properties – Strong Normalization

Theorem

The resource calculus is strongly normalizing (SN).

Proof.

Define:

- the size $\#t \in \mathbb{N}$ of a term t, as you imagine.
- the size of a sum as
  $\#(t_1 + \cdots + t_n) = [\#t_1, \ldots, \#t_n] \in \mathcal{M}_f(\mathbb{N})$

Check that $\#$ decreases along a reduction

$$(\lambda x.t)[s_1, \ldots, s_n] \to_b \begin{cases} t\langle s_1/x_1, \ldots, s_n/x_n\rangle \ ? & \deg_x(t) = n \\ 0 \ \checkmark & \textit{otherwise} \end{cases}$$

w.r.t. the multiset ordering $<_m$ ($\Rightarrow$ at the blackboard!) □

Main Properties – Strong Normalization

Theorem

The resource calculus is strongly normalizing (SN).

Proof.

Define:

- the size $\#t \in \mathbb{N}$ of a term t, as you imagine.
- the size of a sum as
  $\#(t_1 + \cdots + t_n) = [\#t_1, \ldots, \#t_n] \in \mathcal{M}_f(\mathbb{N})$

Check that $\#$ decreases along a reduction

$$(\lambda x.t)[s_1, \ldots, s_n] \to_b \begin{cases} t\langle s_1/x_1, \ldots, s_n/x_n\rangle \ ? & \deg_x(t) = n \\ 0 \ \checkmark & \textit{otherwise} \end{cases}$$

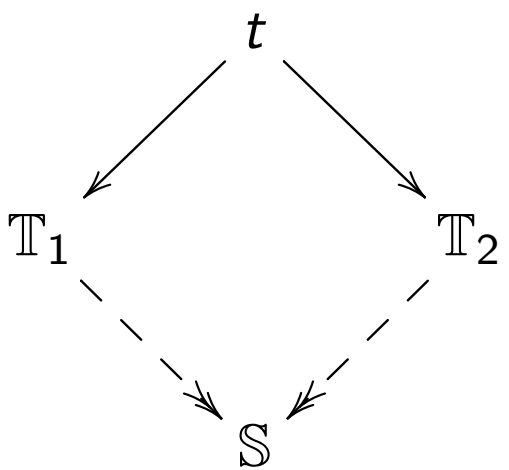
w.r.t. the multiset ordering $<_m$ ($\Rightarrow$ at the blackboard!) □

Main Properties – Confluence

Theorem

The resource calculus is confluent.

Proof. The resource calculus is *locally confluent*, i.e., $t \to_{\mathsf{r}} \mathbb{T}_1$ and $t \to_{\mathsf{r}} \mathbb{T}_2$ imply $\exists \mathbb{S}$ such that $\mathbb{T}_1 \twoheadrightarrow_{\mathsf{r}} \mathbb{S}$ and $\mathbb{T}_2 \twoheadrightarrow_{\mathsf{r}} \mathbb{S}$.



Conclude by applying:

Newmann's Lemma. A term rewriting system is confluent if it is SN and locally confluent. □

Main Properties – Linearity

- Starvation:

$$(\lambda x.x[x])[\lambda x.x[x], \lambda x.x[x]] \to_{\mathsf{r}} 2.(\lambda x.x[x])[\lambda x.x[x]] \to_{\mathsf{r}} 0$$

- Surfeit:

$$(\lambda fgx.f[g[x]])[h][b, c] \to_{\mathsf{r}} (\lambda gx.h[g[x]])[b, c] \to_{\mathsf{r}} 0$$

- Non-determinism:

$$(\lambda x.x[x])[f, g] \to_{\mathsf{r}} f[g] + g[f]$$

Main Properties – Summary

The Resource Calculus

$$\begin{array}{rcll} t & ::= & x \mid \lambda x.t \mid t\, b & \\ b & ::= & [t_1, \ldots, t_n] & \text{where } n \geq 0 \\ \mathbb{T} & ::= & t_1 + \cdots + t_n & \end{array}$$

Reduction:

$(\lambda x.t)[s_1, \ldots, s_n] \twoheadrightarrow_r \mathbb{T} \neq 0 \quad \Rightarrow$ t must use each s_i exactly once in the reduction to a value.

$t \twoheadrightarrow_r c\langle\!|0|\!\rangle = 0 \quad \Leftarrow$ otherwise, the whole program t becomes an empty program 0.

Main Properties

Strong Normalization:	Trivial, because there is no duplication.	✓
Confluence:	Locally confluent + strongly normalizing.	✓
Linearity:	Nothing gets erased in a non-zero reduction.	✓

Its relational semantics

The relational semantics allows to model Λ^r as well.

$$\frac{}{x : [\alpha] \vdash x : \alpha} \qquad \frac{\Gamma, x : \sigma \vdash t : \alpha}{\Gamma \vdash \lambda x.t : \sigma \to \alpha} \qquad \frac{\Gamma \vdash t : \alpha \quad \alpha \simeq \beta}{\Gamma \vdash t : \beta}$$

$$\frac{\Gamma_0 \vdash t : [\beta_1, \ldots, \beta_n] \to \alpha \quad \Gamma_1 \vdash s_1 : \beta_1 \quad \cdots \quad \Gamma_n \vdash s_n : \beta_n}{\sum_{i=0}^{n} \Gamma_i \vdash t[s_1, \ldots, s_n] : \alpha}$$

There are no untyped subterms!

The interpretation of t in $\mathcal{R}$ is given by:

$$\begin{aligned} [\![t]\!] &= \{(\Gamma, \alpha) \mid \Gamma \vdash t : \alpha\} \\ [\![\textstyle\sum_i t_i]\!] &= \textstyle\bigcup_i [\![t_i]\!] \end{aligned}$$

Theorem (Soundness)

- $\mathbb{T} =_r \mathbb{S} \quad \Rightarrow \quad [\![\mathbb{T}]\!] = [\![\mathbb{S}]\!]$

Its expressive power

Virtually unexistent...

(I bet) One cannot even encode an adequate numeral system:

- numerals: $\underline{1}, \underline{2}, \underline{3}, \ldots$
- operations: successor, predecessor, Test-on-Zero, etc.

(While the Full Resource Calculus contains as subsystems:

- The finitary resource calculus,
- The λ-calculus, whence it is Turing-complete,
- The non-deterministic λ-calculus)

Its expressive power

Virtually unexistent...

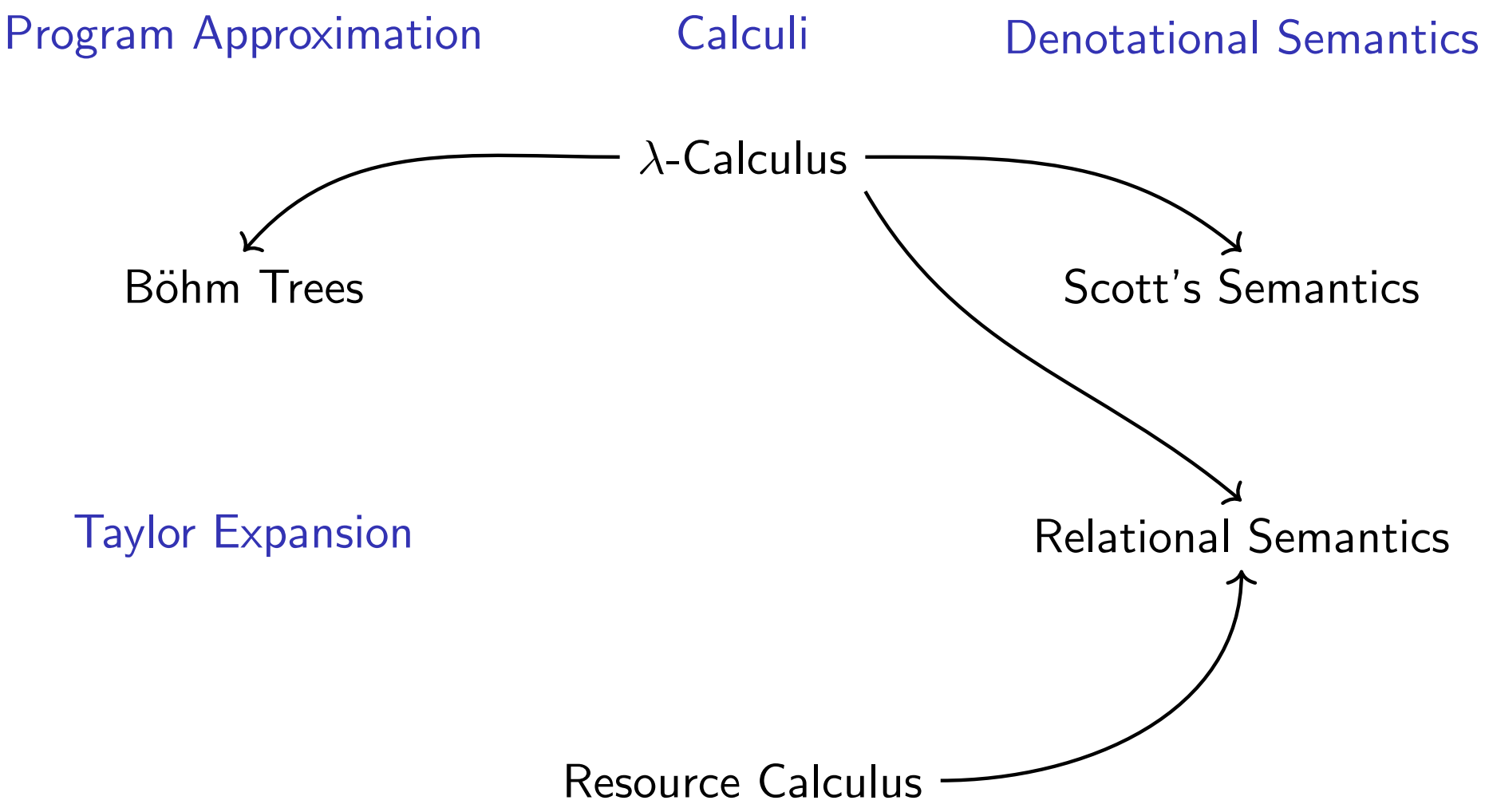
(I bet) One cannot even encode an adequate numeral system:

- numerals: $\underline{1}, \underline{2}, \underline{3}, \ldots$
- operations: successor, predecessor, Test-on-Zero, etc.

(While the Full Resource Calculus contains as subsystems:

- The finitary resource calculus,
- The λ-calculus, whence it is Turing-complete,
- The non-deterministic λ-calculus)

The Global Picture

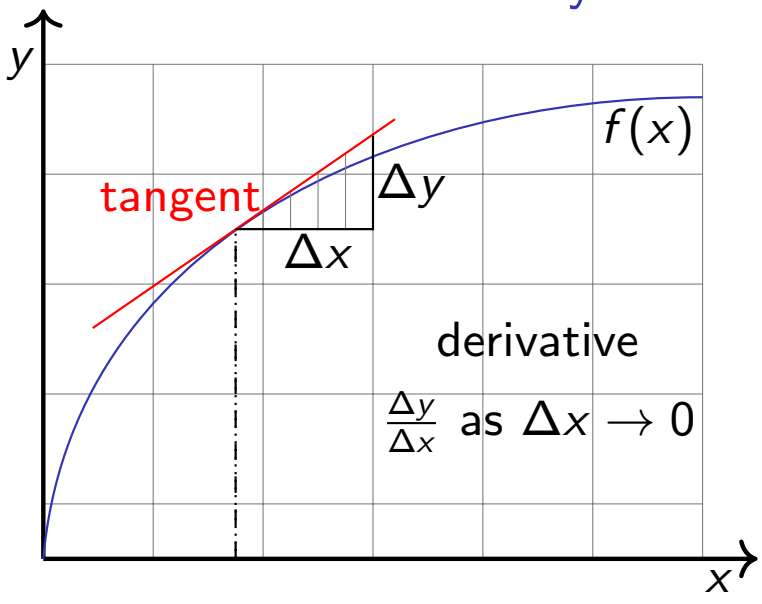


Is the Resource Calculus of any interest?

$$t\langle s/x\rangle \quad \cong \quad \frac{\partial t}{\partial x} \cdot s$$

Towards a differential theory of program approximations

Mathematical Analysis

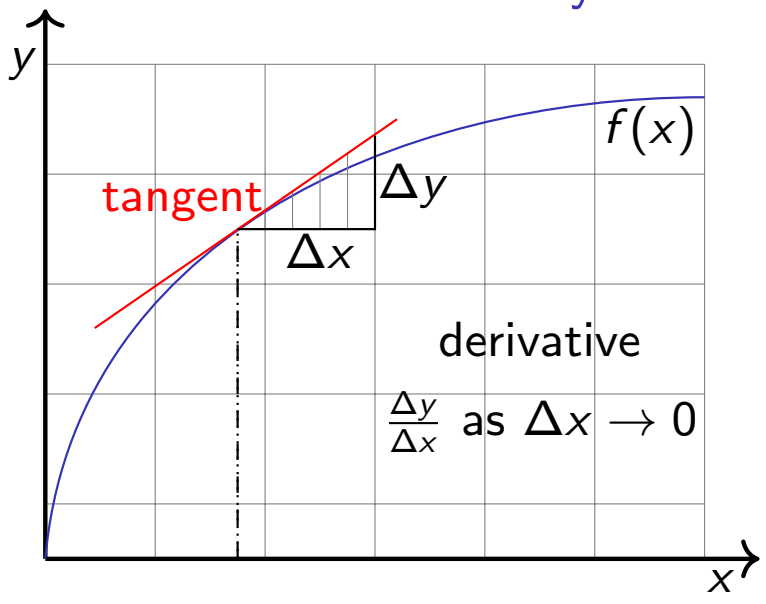


Taylor expansion

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Towards a differential theory of program approximations

Mathematical Analysis



Taylor expansion

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Theory of Programming Languages

The differential λ-calculus

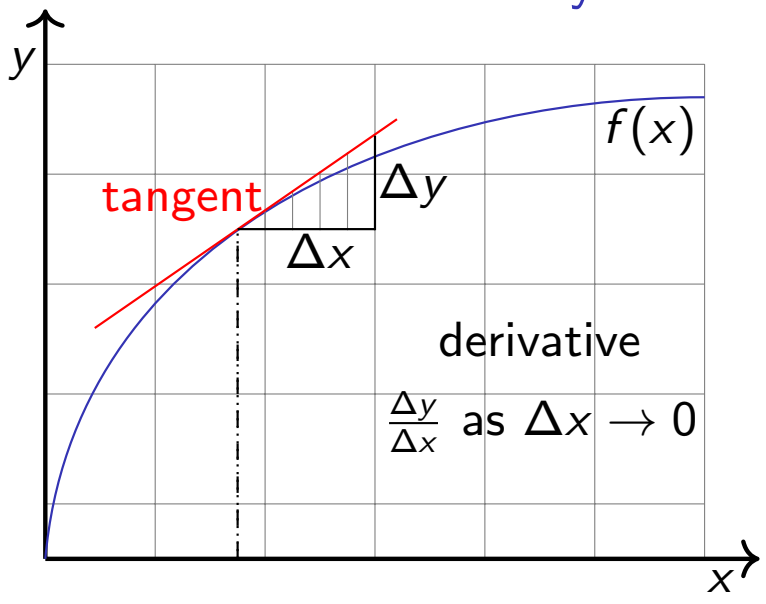
$$D(\lambda x.M) \cdot N \to$$

$$\lambda x.\left(\frac{\partial M}{\partial x} \cdot N\right)$$

linear substitution of N
for one occurrence of x in M

Towards a differential theory of program approximations

Mathematical Analysis



Theory of Programming Languages

The differential λ-calculus

$$D(\lambda x.M) \cdot N \to$$

$$\lambda x.\left(\frac{\partial M}{\partial x} \cdot N\right)$$

linear substitution of N
for one occurrence of x in M

Taylor expansion

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Taylor expansion $\mathcal{T}(-)$

$$P\,x = \sum_{n=0}^{\infty} \frac{1}{n!}\left(\mathrm{D}^n(P) \cdot (x,\ldots,x)\right)0$$

The ambitious goal

Replace the theory of program approximation based on

continuity and Böhm trees

with the theory of

resource consumption based on Taylor expansion.

Resource calculus = Target language of Taylor expansion

Taylor Expansion

$$\mathcal{T}(-) : \quad \Lambda \quad \to \quad \text{power series of resource approximants}$$

$$\begin{aligned}
\mathcal{T}(x) &= x \\
\mathcal{T}(\lambda x.M) &= \lambda x.\mathcal{T}(M) \\
\mathcal{T}(MN) &= \sum_{k\in\mathbb{N}} \frac{1}{k!}\mathcal{T}(M)[\underbrace{\mathcal{T}(N),\ldots,\mathcal{T}(N)}_{k \text{ times}}]
\end{aligned}$$

Examples

- $\mathcal{T}(\mathsf{I}) = \{\lambda x.x\}$,
- $\mathcal{T}(\Delta) = \{\lambda x.x1, \lambda x.x[x], \lambda x.x[x,x], \lambda x.x[x,x,x], \ldots\}$,
  $= \{\lambda x.x[n.x] \mid n \geq 0\}$,
- $\mathcal{T}(\Omega) = \{(\lambda x.x[n.x])[\lambda x.x[n_1.x], \ldots, \lambda x.x[n_k.x]] \mid n, k, n_i \geq 0\}$.

Taylor Expansion

$\mathcal{T}(-) : \quad \Lambda \quad \rightarrow \quad$ sets of resource approximants

$$\begin{aligned}
\mathcal{T}(x) &= \{x\} \\
\mathcal{T}(\lambda x.M) &= \{\lambda x.t \mid t \in \mathcal{T}(M)\} \\
\mathcal{T}(MN) &= \bigcup_{k \in \mathbb{N}} \{t[s_1, \ldots, s_k] \mid t \in \mathcal{T}(M), s_1, \ldots, s_k \in \mathcal{T}(N)\}
\end{aligned}$$

Examples

- $\mathcal{T}(\mathsf{I}) = \{\lambda x.x\}$,
- $\mathcal{T}(\Delta) = \{\lambda x.x1, \lambda x.x[x], \lambda x.x[x,x], \lambda x.x[x,x,x], \ldots\}$,
  $= \{\lambda x.x[n.x] \mid n \geq 0\}$,
- $\mathcal{T}(\Omega) = \{(\lambda x.x[n.x])[\lambda x.x[n_1.x], \ldots, \lambda x.x[n_k.x]] \mid n, k, n_i \geq 0\}$.

Relational semantics — behind the scenes

This explains the inductive proof of the Approximation Theorem!

The Real Approximation Theorem For all $M \in \Lambda$,

$$\Gamma \vdash M : \alpha \iff \exists t \in \mathcal{T}(M) \,.\, \Gamma \vdash t : \alpha$$

Therefore $[\![M]\!] = \bigcup_{t \in \mathcal{T}(M)} [\![t]\!]$.

Relational semantics — behind the scenes

This explains the inductive proof of the Approximation Theorem!

The Real Approximation Theorem For all $M \in \Lambda$,

$$\Gamma \vdash M : \alpha \iff \exists t \in \mathcal{T}(M)\,.\,\Gamma \vdash t : \alpha$$

$$\begin{array}{ccccccccccc}
M = M_0 & \longrightarrow & M_1 & \longrightarrow & M_2 & \longrightarrow & M_3 & \longrightarrow\!\!\!\!\rightarrow & M_n & \longrightarrow & \cdots \\
\mathcal{T}(M_0) & & \mathcal{T}(M_1) & & \mathcal{T}(M_2) & & \mathcal{T}(M_3) & & \mathcal{T}(M_n) & & \\
\Psi & & \cup & & \cup & & \cup & & \cup & & \\
\exists t_0 & \longrightarrow\!\!\!\!\rightarrow & t_1 + \mathbb{T}_1 & \twoheadrightarrow & t_2 + \mathbb{T}_2 & \twoheadrightarrow & t_3 + \mathbb{T}_3 & \longrightarrow\!\!\!\!\rightarrow & t_n + \mathbb{T}_n & = \mathsf{nf}(t) & \\
\Gamma \vdash t_0 : \alpha & & & & & & & & \Gamma \vdash t_n : \alpha & &
\end{array}$$

Therefore $[\![M]\!] = \bigcup_{t \in \mathcal{T}(M)} [\![t]\!]$.

The Dynamics of Taylor Expansion

The Taylor expansion is a "static" operation

- $M = x \quad \Rightarrow \quad t \in \mathcal{T}(M)$ has the shape of a variable,
- $M = \lambda x.N \quad \Rightarrow \quad t \in \mathcal{T}(M)$ has the shape of an abstraction,
- $M = PQ \quad \Rightarrow \quad t \in \mathcal{T}(M)$ has the shape of an application,
- $M = (\lambda x.P)Q \quad \Rightarrow \quad t \in \mathcal{T}(M)$ has the shape of a redex.

As the Resource Calculus enjoys SN, we can define:

$$\mathrm{NF}(\mathcal{T}(M)) = \bigcup \{\mathrm{nf}(t) \mid t \in \mathcal{T}(M)\}$$

Normalizing the Taylor Expansion

$$\mathrm{NF}(\mathcal{T}(\Omega)) = \emptyset$$

- For every $t \in \mathcal{T}(\Omega)$, check $t \twoheadrightarrow_{\mathsf{r}} 0$.
- Conclude $\mathcal{T}(\Omega) = \emptyset$.

More generally:

$$M \text{ is unsolvable} \quad \Rightarrow \quad \mathrm{NF}(\mathcal{T}(M)) = \emptyset$$

(We'll prove it later)

Normalizing the Taylor Expansion

$$\mathrm{NF}(\mathcal{T}(\Omega)) = \emptyset$$

- For every $t \in \mathcal{T}(\Omega)$, check $t \twoheadrightarrow_{\mathsf{r}} 0$.
- Conclude $\mathcal{T}(\Omega) = \emptyset$.

More generally:

$$M \text{ is unsolvable} \quad \Rightarrow \quad \mathrm{NF}(\mathcal{T}(M)) = \emptyset$$

(We'll prove it later)

Normalizing the Taylor Expansion

$$\mathrm{NF}(\mathcal{T}(\mathsf{Y})) = ?$$

Normalizing the Taylor Expansion

$$\mathrm{NF}(\mathcal{T}(\mathsf{Y})) = ?$$

- Mmm...

Let us look at its shape:

$$\mathsf{Y} = \lambda \mathsf{f}.(\lambda \mathsf{x}.\mathsf{f}(\mathsf{x}\mathsf{x}))(\lambda \mathsf{x}.\mathsf{f}(\mathsf{x}\mathsf{x}))$$

Normalizing the Taylor Expansion

$$\mathrm{NF}(\mathcal{T}(\mathsf{Y})) = ?$$

- Mmm...

Let us look at its shape:

$$\mathsf{Y} = \lambda \mathsf{f}.(\lambda \mathsf{x}.\mathsf{f}(\mathsf{xx}))(\lambda \mathsf{x}.\mathsf{f}(\mathsf{xx}))$$

- Mmm...

Problem! That's a toughy...

We know how to compute its Böhm tree

$$\mathrm{BT}(\mathsf{Y}) = \lambda \mathsf{f}.\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\cdots))))$$

since

$$\mathcal{A}(\mathsf{Y}) = \{\lambda \mathsf{f}.\mathsf{f}\bot, \lambda \mathsf{f}.\mathsf{f}(\mathsf{f}\bot), \lambda \mathsf{f}.\mathsf{f}(\mathsf{f}(\mathsf{f}\bot)), \lambda \mathsf{f}.\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}\bot))), \dots\}$$

Can we Taylor expand a Böhm tree?

For an approximant A, define:

$$\begin{aligned}
\mathcal{T}(\bot) &= \emptyset, \\
\mathcal{T}(x) &= \{x\}, \\
\mathcal{T}(\lambda x.A) &= \{\lambda x.t \mid t \in \mathcal{T}(A)\}, \\
\mathcal{T}(A_1 A_2) &= \bigcup_{k \in \mathbb{N}} \left\{ t[s_1, \ldots, s_k] \mid t \in \mathcal{T}(A_1), s_1, \ldots, s_k \in \mathcal{T}(A_2) \right\}.
\end{aligned}$$

Then, we can simply define

$$\mathcal{T}(\mathrm{BT}(M)) = \bigcup_{A \in \mathcal{A}(M)} \mathcal{T}(A)$$

Commutation Taylor / Böhm

Theorem (Ehrhard & Regnier 2003)

For every λ-term M, we have:

$$\mathrm{NF}(\mathcal{T}(M)) = \mathcal{T}(\mathrm{BT}(M))$$

Thanks! $\mathcal{T}(\mathrm{BT}(\mathsf{Y})) = \{\lambda \mathsf{f}.\mathsf{f}1, \lambda \mathsf{f}.\mathsf{f}[\lambda \mathsf{f}.\mathsf{f}1], \lambda \mathsf{f}.\mathsf{f}[\lambda \mathsf{f}.\mathsf{f}1, \lambda \mathsf{f}.\mathsf{f}1], \dots\}$

Commutation Taylor / Böhm

Theorem (Ehrhard & Regnier 2003)

For every λ-term M, we have:

$$\mathrm{NF}(\mathcal{T}(M)) = \mathcal{T}(\mathrm{BT}(M))$$

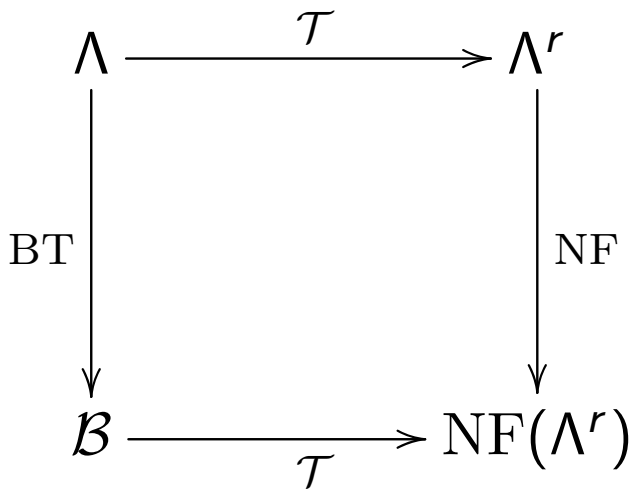
Corollary 1

$$M \text{ is unsolvable} \iff \mathrm{NF}(\mathcal{T}(M)) = \emptyset$$

Corollary 2

$$\mathrm{BT}(M) = \mathrm{BT}(N) \iff \mathrm{NF}(\mathcal{T}(M)) = \mathrm{NF}(\mathcal{T}(N))$$

Taylor Expansion vs Böhm Trees



Advantages:

1. Approximants are closed under application.
2. Enjoy Strong Normalization + Linearity.
3. Generalizable to the mainstream languages.

Disadvantage:

1. lots of indices arise from the linearization.

Classic results

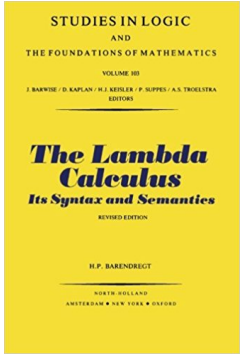
Scott's Continuity

topological argument

Contextuality of BTs

Genericity Lemma

Chapter 14



Berry's Stability

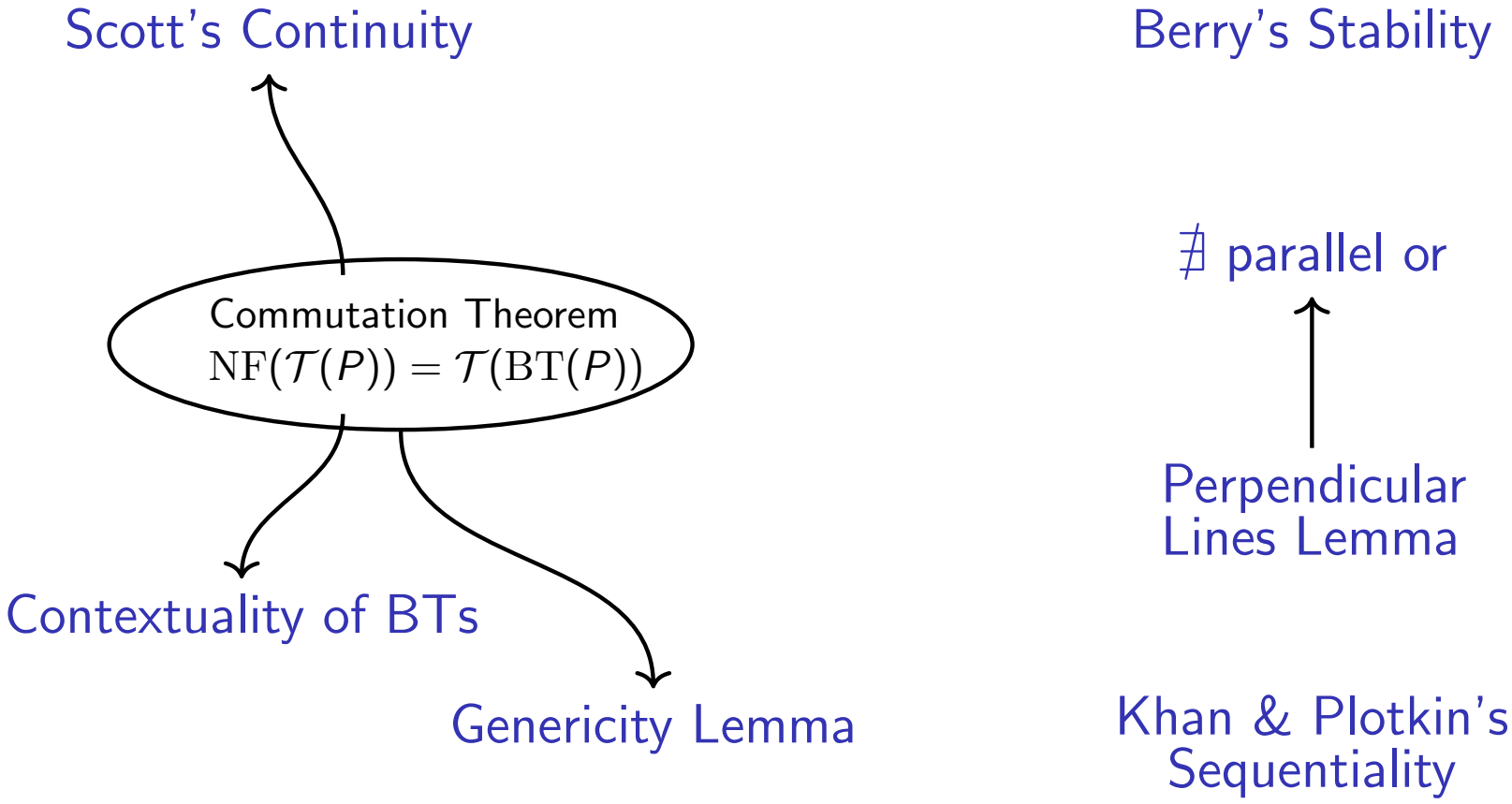
$\nexists$ parallel or

Perpendicular Lines Lemma

Khan & Plotkin's Sequentiality

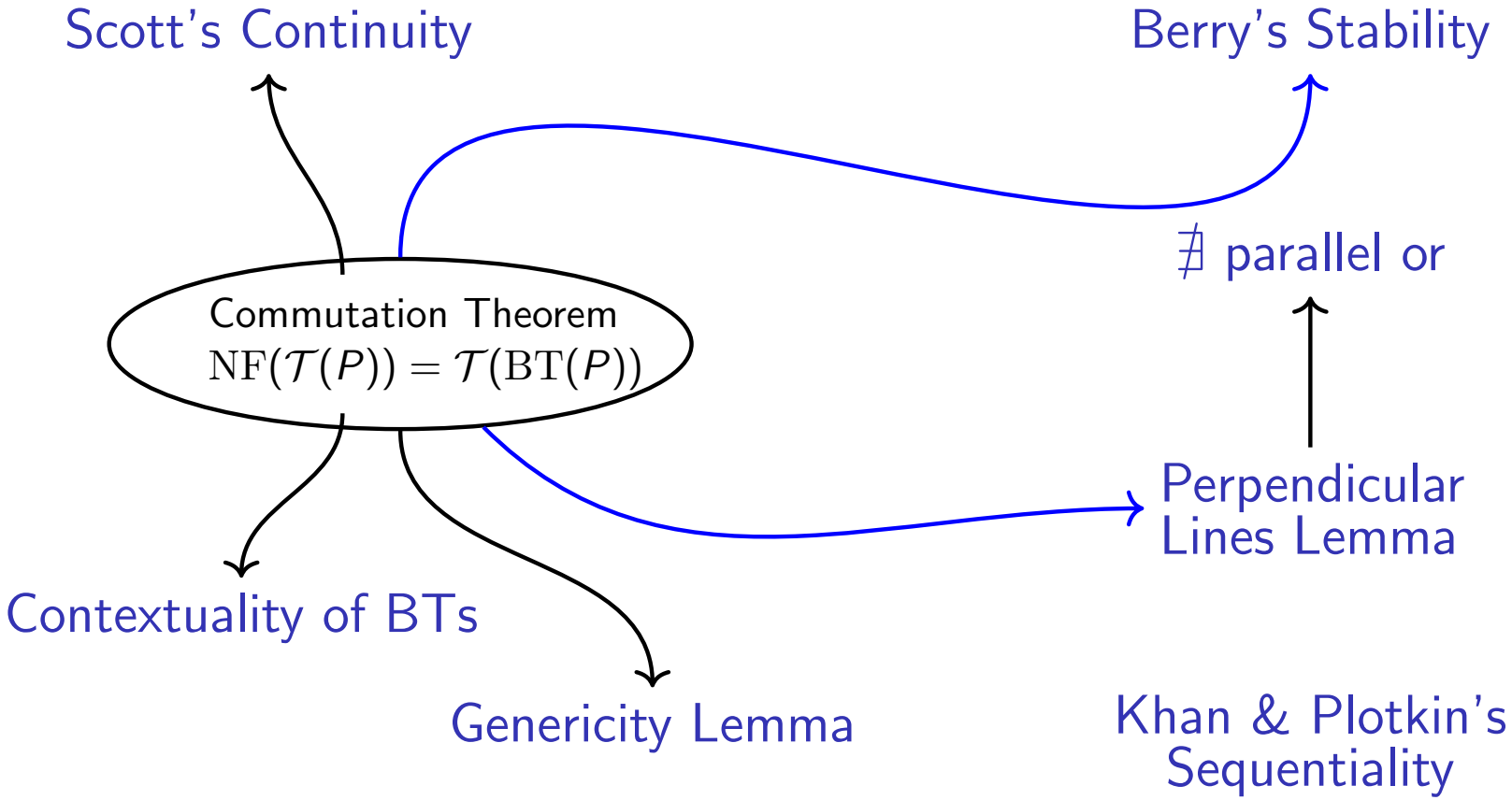
D. Barbarossa and G. Manzonetto. Taylor Subsumes Scott, Berry, Kahn and Plotkin. PACMPL Vol. 4, pp. 1:1-1:23, 2020.

Classic results with simpler inductive proofs



D. Barbarossa and G. Manzonetto. Taylor Subsumes Scott, Berry, Kahn and Plotkin. PACMPL Vol. 4, pp. 1:1-1:23, 2020.

Classic results with simpler inductive proofs



D. Barbarossa and G. Manzonetto. Taylor Subsumes Scott, Berry, Kahn and Plotkin. PACMPL Vol. 4, pp. 1:1-1:23, 2020.

A proof of context closure via Taylor Expansion

$$\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad \forall C[] \,.\, \mathrm{BT}(C[M]) = \mathrm{BT}(C[N])$$

Proof. By Corollary 2, assuming

$$\mathrm{NF}(\mathcal{T}(M)) \subseteq \mathrm{NF}(\mathcal{T}(N))$$

we have to prove:

$$\mathrm{NF}(\mathcal{T}(C[M])) \subseteq \mathrm{NF}(\mathcal{T}(C[N]))$$

Proceed by structural induction on $C[]$.

Cases $C[] = x$ and $C[] = []$. Trivial.

A proof of context closure via Taylor Expansion

$$\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad \forall C[] \,.\, \mathrm{BT}(C[M]) = \mathrm{BT}(C[N])$$

Proof. By Corollary 2, assuming

$$\mathrm{NF}(\mathcal{T}(M)) \subseteq \mathrm{NF}(\mathcal{T}(N))$$

we have to prove:

$$\mathrm{NF}(\mathcal{T}(C[M])) \subseteq \mathrm{NF}(\mathcal{T}(C[N]))$$

Proceed by structural induction on $C[] = \lambda x.C'[]$.

For $t \in \mathrm{NF}(\mathcal{T}(\lambda x.C'[M]))$, $\exists \lambda x.t' \in \mathcal{T}(\lambda x.C'[M])$ such that $\lambda x.t' \twoheadrightarrow_{\mathsf{r}} \lambda x.\mathrm{nf}(t') \ni t$. By IH, we get

$$\mathrm{nf}(t') \subseteq \mathrm{NF}(\mathcal{T}(C'[M])) \subseteq \mathrm{NF}(\mathcal{T}(C'[N]))$$

whence $t \in \lambda x.\mathrm{nf}(t') \subseteq \mathrm{NF}(\mathcal{T}(\lambda x.C'[N])))$.

A proof of context closure via Taylor Expansion

$$\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad \forall C[] \,.\, \mathrm{BT}(C[M]) = \mathrm{BT}(C[N])$$

Proof. The only interesting case is application: $C[] = (C_1[])\,(C_2[])$.

Take $t \in \mathrm{NF}(\mathcal{T}(C[M]))$, then $\exists t' \in \mathcal{T}((C_1[M])(C_2[M]))$ such that

$$t' = \qquad\qquad s_1[u_1, \ldots, u_k] \longrightarrow\!\!\!\!\!\rightarrow t + \mathbb{T}$$

A proof of context closure via Taylor Expansion

$$\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad \forall C[] \,.\, \mathrm{BT}(C[M]) = \mathrm{BT}(C[N])$$

Proof. The only interesting case is application: $C[] = (C_1[])\,(C_2[])$.

Take $t \in \mathrm{NF}(\mathcal{T}(C[M]))$, then $\exists t' \in \mathcal{T}((C_1[M])(C_2[M]))$ such that

$$t' = \qquad\qquad s_1[u_1, \ldots, u_k] \longrightarrow\!\!\!\!\!\rightarrow t + \mathbb{T}$$

with $s_1 \in \mathcal{T}(C_1[M])$

and $u_1, \ldots, u_k \in \mathcal{T}(C_2[M])$.

A proof of context closure via Taylor Expansion

$$\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad \forall C[] \,.\, \mathrm{BT}(C[M]) = \mathrm{BT}(C[N])$$

Proof. The only interesting case is application: $C[] = (C_1[])\,(C_2[])$.

Take $t \in \mathrm{NF}(\mathcal{T}(C[M]))$, then $\exists t' \in \mathcal{T}((C_1[M])(C_2[M]))$ such that

$$t' = \qquad s_1[u_1, \ldots, u_k] \twoheadrightarrow t + \mathbb{T}$$
$$\Big\downarrow \qquad \nearrow$$
$$\mathrm{nf}(s_1)[\mathrm{nf}(u_1), \ldots, \mathrm{nf}(u_k)]$$

with $\mathrm{nf}(s_1) \in \mathrm{NF}(\mathcal{T}(C_1[M]))$

and $\mathrm{nf}(u_1), \ldots, \mathrm{nf}(u_k) \in \mathrm{NF}(\mathcal{T}(C_2[M]))$

A proof of context closure via Taylor Expansion

$$\mathrm{BT}(M) = \mathrm{BT}(N) \quad \Rightarrow \quad \forall C[] \,.\, \mathrm{BT}(C[M]) = \mathrm{BT}(C[N])$$

Proof. The only interesting case is application: $C[] = (C_1[])\,(C_2[])$.

Take $t \in \mathrm{NF}(\mathcal{T}(C[M]))$, then $\exists t' \in \mathcal{T}((C_1[M])(C_2[M]))$ such that

$$t' = \qquad s_1[u_1, \ldots, u_k] \twoheadrightarrow t + \mathbb{T}$$
$$\downarrow \qquad \nearrow$$
$$\mathrm{nf}(s_1)[\mathrm{nf}(u_1), \ldots, \mathrm{nf}(u_k)]$$

with $\mathrm{nf}(s_1) \in \mathrm{NF}(\mathcal{T}(C_1[M])) = \mathrm{NF}(\mathcal{T}(C_1[N]))$

and $\mathrm{nf}(u_1), \ldots, \mathrm{nf}(u_k) \in \mathrm{NF}(\mathcal{T}(C_2[M])) = \mathrm{NF}(\mathcal{T}(C_2[N]))$.

We conclude that $t \in \mathrm{NF}(\mathcal{T}(C[N]))$. $\square$

Taylor approximants are "just like" Böhm's ones

A resource term t is called

- linearized if every bag in t has cardinality 1.
- affined if every bag in t has cardinality at most 1.

Every affined normal $t \in \Lambda^r$ can be sent to an approximant $|t| \in \mathcal{A}$:

$$\begin{aligned} |x| &= x, \\ |\lambda x.t| &= \lambda x.|t|, \\ |s[t]| &= |s|\,|t|, \\ |s[]| &= |s|\,\bot. \end{aligned}$$

Taylor approximants are "just like" Böhm's ones

A resource term t is called

- linearized if every bag in t has cardinality 1.
- affined if every bag in t has cardinality at most 1.

Every approximant $A \neq \bot$, can be sent to an affined $A^\circ \in \Lambda^r$:

$$\begin{aligned} x^\circ &= x, \\ (\lambda x.A)^\circ &= \lambda x.A^\circ, \\ (A_1 A_2)^\circ &= A_1^\circ[A_2^\circ], \\ (A\bot)^\circ &= A^\circ[]. \end{aligned}$$

Properties

- For all $A \in \mathcal{A} - \{\bot\}$ and $t \in \Lambda^r$, we have:

$$|P^\circ| = P$$

and

$$|t|^\circ = t.$$

- For all M there exists a unique linearized t such that

$$t \in \mathrm{NF}(\mathcal{T}(M)) \iff M \text{ is } \beta\text{-normalizable.}$$

In this case, we have $\mathrm{nf}_\beta(M) = |t|$.

The Genericity Lemma

Let M unsolvable. $C[M]$ has a β-nf $\Rightarrow$ $\forall N\,.\,C[M] =_\beta C[N]$.

Standard proof: Topological method.

Compactification points in the tree topology are precisely the unsolvables.

Several proofs in the literature:

Masako Takahashi: A Simple Proof of the Genericity Lemma. Logic, Language and Computation 1994: 117-118

Jan Kuper: Proving the Genericity Lemma by Leftmost Reduction is Simple. RTA 1995: 271-278

The Genericity Lemma

Let M unsolvable. $C[M]$ has a β-nf $\quad\Rightarrow\quad$ $\forall N\,.\,C[M] =_\beta C[N]$.

Proof. As $C[M]$ normalizable, $\exists! t \in \mathrm{NF}(\mathcal{T}(C[M]))$ linearized s.t.

$$|t| = \mathrm{nf}(C[M])$$

So, there exist $t' \in \mathcal{T}(C[M])$ such that:

$$t' = \qquad\qquad c(\!|s_1, \ldots, s_k|\!) \longrightarrow\!\!\!\!\!\rightarrow t + \mathbb{T}$$

for some $c(\!|-|\!) \in \mathcal{T}(C[-])$ and $s_1, \ldots, s_k \in \mathcal{T}(M)$.

The Genericity Lemma

Let M unsolvable. $C[M]$ has a β-nf $\quad\Rightarrow\quad$ $\forall N\,.\,C[M] =_\beta C[N]$.

Proof. As $C[M]$ normalizable, $\exists! t \in \mathrm{NF}(\mathcal{T}(C[M]))$ linearized s.t.

$$|t| = \mathrm{nf}(C[M])$$

So, there exist $t' \in \mathcal{T}(C[M])$ such that:

$$t' = \qquad c(\!|s_1, \ldots, s_k|\!) \twoheadrightarrow t + \mathbb{T}$$
$$\Big\downarrow \qquad \nearrow$$
$$c(\!|\mathrm{nf}(s_1), \ldots, \mathrm{nf}(s_k)|\!)$$

for some $c(\!|-|\!) \in \mathcal{T}(C[-])$ and $s_1, \ldots, s_k \in \mathcal{T}(M)$. (By Confluence and Strong Normalization.)

The Genericity Lemma

Let M unsolvable. $C[M]$ has a β-nf $\quad\Rightarrow\quad$ $\forall N\,.\,C[M] =_\beta C[N]$.

Proof. As $C[M]$ normalizable, $\exists! t \in \mathrm{NF}(\mathcal{T}(C[M]))$ linearized s.t.

$$|t| = \mathrm{nf}(C[M])$$

So, there exist $t' \in \mathcal{T}(C[M])$ such that:

$$t' = \qquad \begin{array}{ccc} c(\!|s_1,\ldots,s_k|\!) & \twoheadrightarrow & t + \mathbb{T} \\ \downarrow & \nearrow & \\ c(\!|0,\ldots,0|\!) & & \end{array}$$

for some $c(\!|-|\!) \in \mathcal{T}(C[-])$ and $s_1,\ldots,s_k \in \mathcal{T}(M)$. Now, M unsolvable entails $\mathrm{nf}(s_i) = 0$.

The Genericity Lemma

Let M unsolvable. $C[M]$ has a β-nf $\quad\Rightarrow\quad$ $\forall N\,.\,C[M] =_\beta C[N]$.

Proof. As $C[M]$ normalizable, $\exists! t \in \mathrm{NF}(\mathcal{T}(C[M]))$ linearized s.t.

$$|t| = \mathrm{nf}(C[M])$$

So, there exist $t' \in \mathcal{T}(C[M])$ such that:

$$t' = \qquad \begin{array}{ccc} c(\!|s_1,\ldots,s_k|\!) & \twoheadrightarrow & t + \mathbb{T} \\ \downarrow & \nearrow & \\ c(\!|0,\ldots,0|\!) & & \end{array}$$

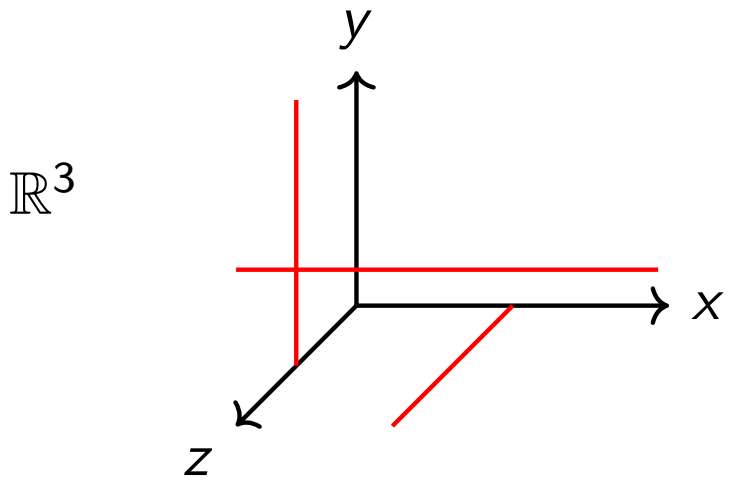
for some $c(\!|-|\!) \in \mathcal{T}(C[-])$ and $s_1,\ldots,s_k \in \mathcal{T}(M)$. Now, M unsolvable entails $\mathrm{nf}(s_i) = 0$. Thus, $(\!|-|\!)$ cannot occur in $c(\!|-|\!)$ so we get:

$$c(\!|s_1,\ldots,s_k|\!) \in \mathcal{T}(C[N]) \quad\Rightarrow\quad t \in \mathrm{NF}(\mathcal{T}(C[N]))$$

and since t is linearized we obtain $\mathrm{nf}_\beta(C[N]) = |t|$. □

Perpendicular Lines Lemma

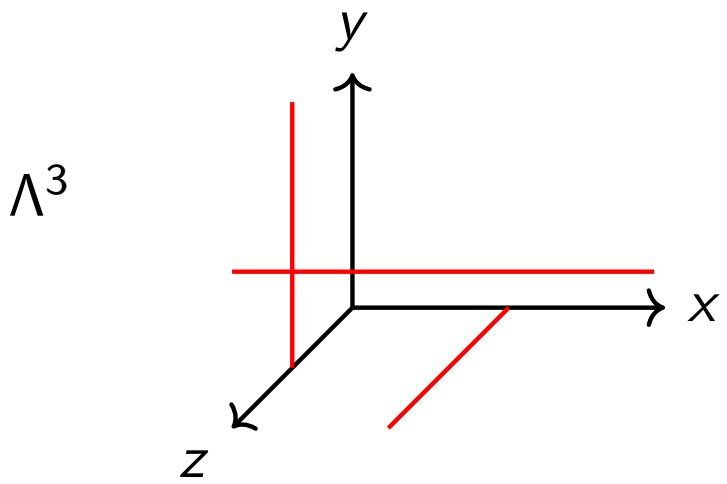
PLL: If a context $C[-_1, \ldots, -_n] : \Lambda^n \to \Lambda$ is constant on n perpendicular lines, then it must be constant everywhere.



$$\ell_1 = \{(x, 1, 2) \mid x \in \mathbb{R}\},$$
$$\ell_2 = \{(0, y, 1) \mid y \in \mathbb{R}\},$$
$$\ell_3 = \{(1, 0, z) \mid z \in \mathbb{R}\}.$$

Perpendicular Lines Lemma

PLL: If a context $C[-_1, \ldots, -_n] : \Lambda^n \to \Lambda$ is constant on n perpendicular lines, then it must be constant everywhere.



$$\ell_1 = \{(X, \lambda x.x, \lambda xy.x) \mid X \in \Lambda\},$$
$$\ell_2 = \{(\lambda xy.y, Y, \lambda x.x) \mid Y \in \Lambda\},$$
$$\ell_3 = \{(\lambda x.x, \lambda xy.x, Z) \mid Z \in \Lambda\}.$$

Known results

Perpendicular Lines Lemma	β	$\mathcal{B}$
open term model	✓	✓
closed term model	✗	?

- $\mathcal{M}(\mathcal{B}) \models$ PLL, Barendregt's Book 1982,
  Proof technique: Sequentiality.
- $\mathcal{M}^o(\mathcal{B}) \models$ PLL?
- $\mathcal{M}^o(\beta) \not\models$ PLL, by Barendregt & Statman 1999.
  Proof: Counterexample via Plotkin's terms.
- $\mathcal{M}(\beta) \models$ PLL, by De Vrijer & Endrullis 2008.
  Proof: via Reduction under Substitution.

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

In $\mathcal{B}$ a context $C[-]$ can be constant for several reasons:

1. C does not contain the hole in the first place (the trivial case);
2. the hole is erased during its reduction;
3. the hole is "hidden" behind an unsolvable;
4. the hole is never erased but "pushed into infinity".

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

An approximant $c \in \mathcal{T}(C[-])$ such that $\mathrm{nf}(c) \neq 0$ can be constant for only one reason:

1. c does not contain the hole in the first place (the trivial case);
2. the hole is erased during its reduction ;
3. the hole is "hidden" behind an unsolvable;
4. the hole is never erased but "pushed into infinity".

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

An approximant $c \in \mathcal{T}(C[-])$ such that $\mathrm{nf}(c) \neq 0$ can be constant for only one reason:

1. c does not contain the hole in the first place (the trivial case);
2. ~~the hole is erased during its reduction~~ (linearity);
3. the hole is "hidden" behind an unsolvable;
4. the hole is never erased but "pushed into infinity".

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

An approximant $c \in \mathcal{T}(C[-])$ such that $\mathrm{nf}(c) \neq 0$ can be constant for only one reason:

1. c does not contain the hole in the first place (the trivial case);
2. ~~the hole is erased during its reduction~~ (linearity);
3. ~~the hole is "hidden" behind an unsolvable~~ (SN);
4. the hole is never erased but "pushed into infinity".

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

An approximant $c \in \mathcal{T}(C[-])$ such that $\mathrm{nf}(c) \neq 0$ can be constant for only one reason:

1. c does not contain the hole in the first place (the trivial case);
2. ~~the hole is erased during its reduction~~ (linearity);
3. ~~the hole is "hidden" behind an unsolvable~~ (SN);
4. ~~the hole is never erased but "pushed into infinity"~~ (finiteness).

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

Claim.

$\forall c \in \mathcal{T}(C[-_1, \ldots, -_n]),\ \mathrm{nf}(c) \neq 0 \Rightarrow c$ cannot contain any hole.

By induction on the size of c, using all the properties above.

Perpendicular Lines Lemma

$$\forall Z \begin{cases} C[Z, M_{12}, \ldots\ldots, M_{1n}] & =_{\mathcal{B}} & N_1 \\ C[M_{21}, Z, \ldots\ldots, M_{2n}] & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ C[M_{n1}, \ldots, M_{n(n-1)}, Z] & =_{\mathcal{B}} & N_n \end{cases}$$

$$\Downarrow$$

$$\forall \vec{Z} \,.\, C[Z_1, \ldots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \cdots =_{\mathcal{B}} N_n$$

Our proof does not need open terms!

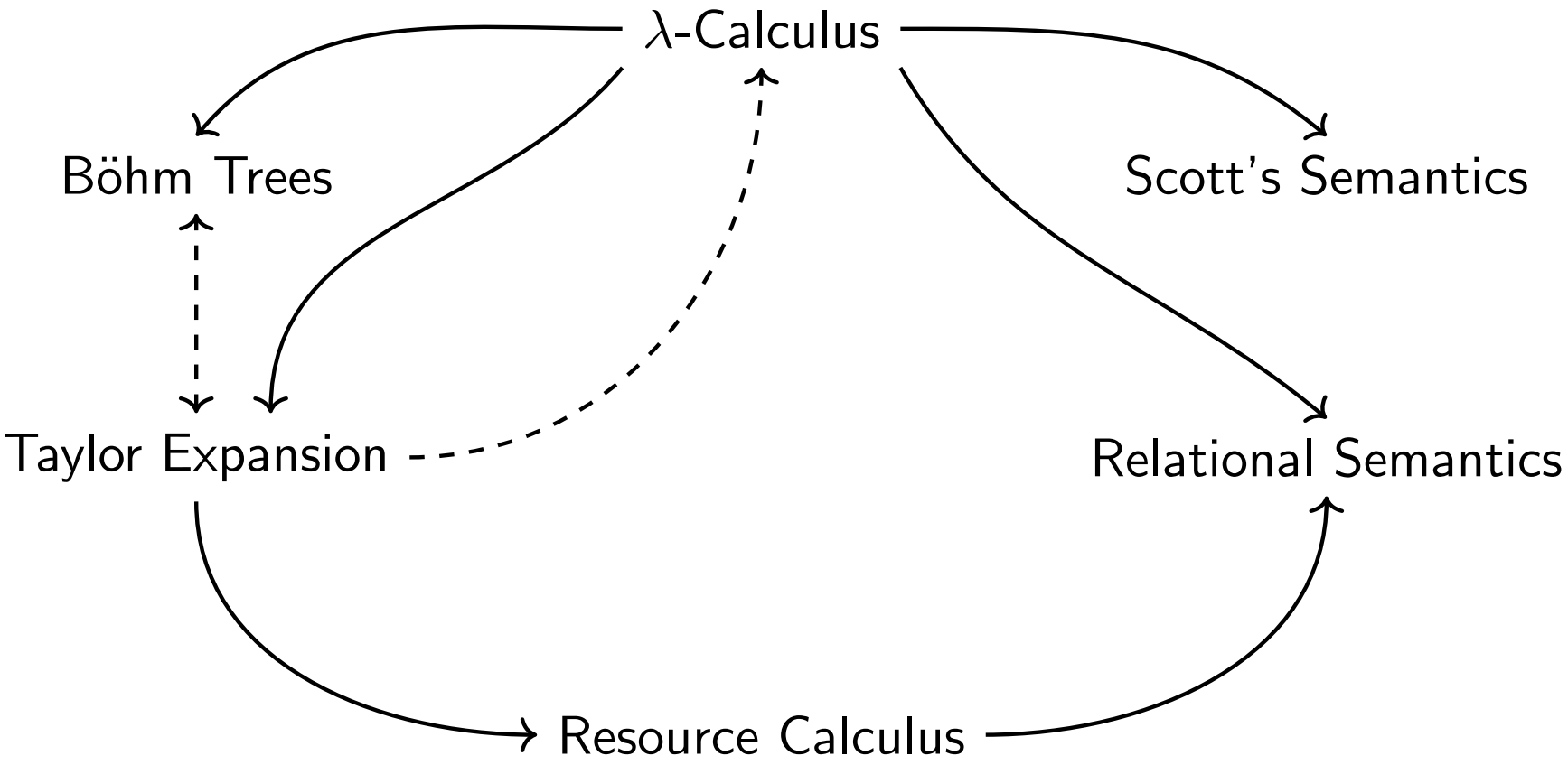
$\mathcal{M}^o(\mathcal{B}) \models \mathsf{PLL}$ ✓

This completes the picture!

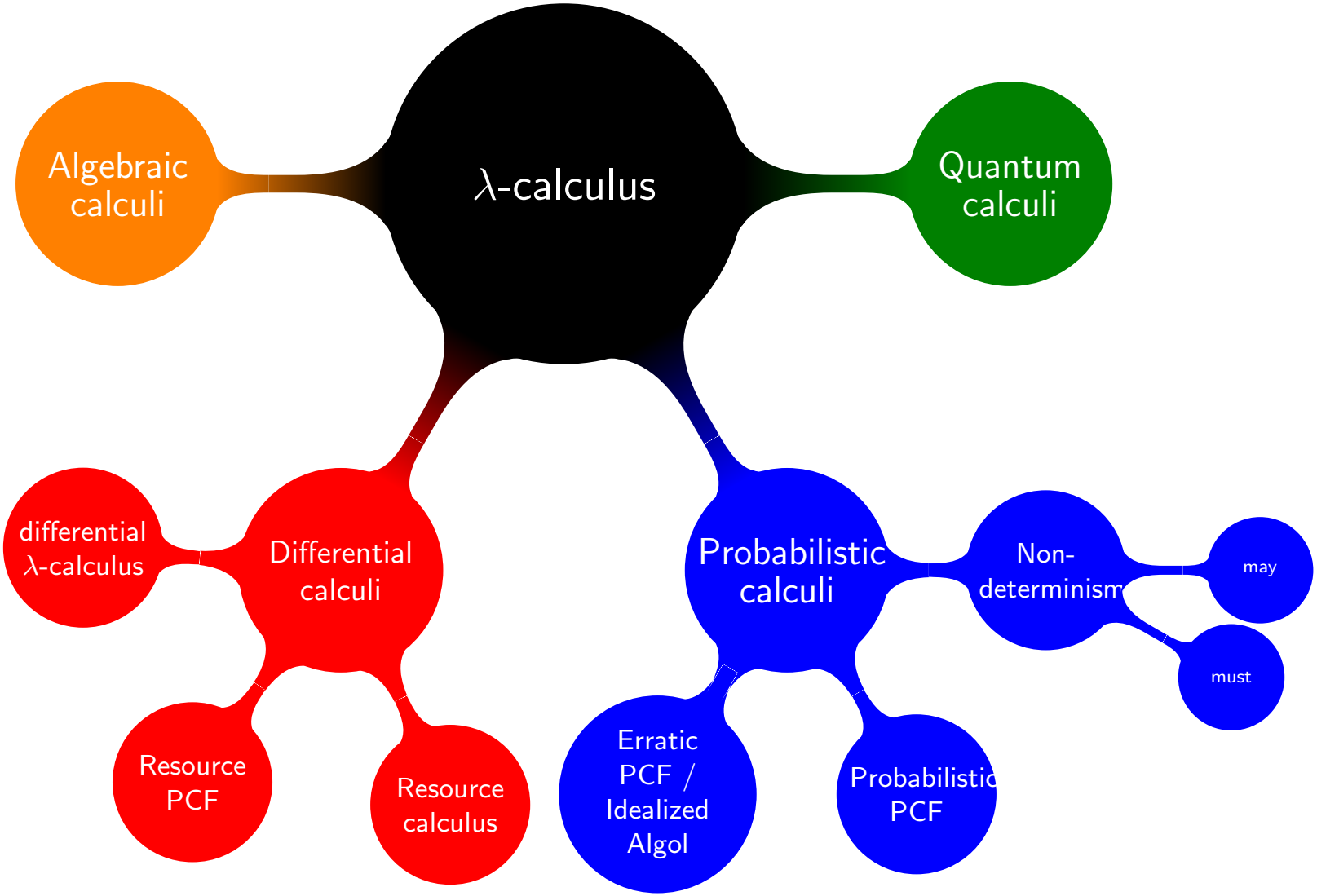
Perpendicular Lines Lemma	β	$\mathcal{B}$
open term model	✓	✓
closed term model	✗	✓

The Global Picture

Program Approximation | Calculi | Denotational Semantics



Advantage: These techniques scale to many languages



Thanks!