On operational properties of quantitative extensions of λ-calculus

Michele ALBERTI

Aix-Marseille Université
Institut de Mathématiques de Marseille

Università di Bologna
Dipartimento di Informatica

Marseille – December 5th, 2014

Semantics

- Reasoning about program behaviours and properties,

Operational Semantics

- Reasoning about program behaviours and properties,
- Abstract descriptions of program executions,

Quantitative Operational Semantics

- Reasoning about program behaviours and properties,
- Abstract descriptions of program executions,
- Provide/Handle fine-grained information about computation.

Some Motivations

- New computational models with quantitative aspects,

Room for fundamental, operationally based investigations.

- Most of research focus is on:
  - Denotational semantics of programming languages,
  - Programming constructs and applications.

Qualitative vs Quantitative Semantics

Non-deterministic computation:

$$\begin{tikzcd}
 & & & \mathtt{O}_1 \\
 & \cdot \arrow[r] & \cdot \arrow[r] \arrow[ur, bend left] \arrow[dr, bend right] & \dots \\
 & & & \mathtt{O}_2 \\
\mathtt{I} \arrow[r] \arrow[uur, bend left] \arrow[dr, bend right] & \cdot \arrow[r] & \cdot \arrow[r] & \dots \\
 & \mathtt{O}_3 & &
\end{tikzcd}$$

Qualitative vs Quantitative Semantics

Non-deterministic computation:

$$
\begin{tikzcd}
 & & & \mathtt{O}_1 \\
 & \cdot \arrow[r] & \cdot \arrow[r] \arrow[ur, bend left] \arrow[dr, bend right] & \ldots \\
 & & & \mathtt{O}_2 \\
\mathtt{I} \arrow[r] \arrow[uur, bend left] \arrow[dr, bend right] & \cdot \arrow[r] & \cdot \arrow[r] & \ldots \\
 & \mathtt{O}_3 & &
\end{tikzcd}
$$

Qualitative is about what.

Qualitative vs Quantitative Semantics

Quantitative semantics as a refinement of qualitative one.

Probabilistic computation:

$$
\begin{tikzcd}
 & & & O_1 \\
 & \cdot \arrow[r] & \cdot \arrow[ur, bend left, "\frac{1}{3}"] \arrow[r, "\frac{1}{3}"] \arrow[dr, bend right, "\frac{1}{3}"'] & \ldots \\
 & & & O_2 \\
I \arrow[uur, bend left, "\frac{2}{3}"] \arrow[r, "\frac{1}{6}"] \arrow[dr, bend right, "\frac{1}{6}"'] & \cdot \arrow[r] & \cdot \arrow[r] & \ldots \\
 & O_3 & &
\end{tikzcd}
$$

Qualitative vs Quantitative Semantics

Quantitative semantics as a refinement of qualitative one.

Probabilistic computation:

$$
\begin{tikzcd}
 & & & \mathtt{O}_1 \\
 & \cdot \arrow[r] & \cdot \arrow[ur, bend left, "\frac{1}{3}"] \arrow[r, "\frac{1}{3}"] \arrow[dr, bend right, "\frac{1}{3}"'] & \dots \\
 & & & \mathtt{O}_2 \\
\mathtt{I} \arrow[uuur, bend left, "\frac{2}{3}"] \arrow[r, "\frac{1}{6}"] \arrow[dr, bend right, "\frac{1}{6}"'] & \cdot \arrow[r] & \cdot \arrow[r] & \dots \\
 & \mathtt{O}_3 & &
\end{tikzcd}
$$

Quantitative is about how {much,many}.

Contributions: Algebraic λ-calculus

- Λ_Σ :untyped Λ with finite R-valued linear combinations,
- Purpose:
  - Framework for {quantum, probabilistic, *etc.* } operational theory.

Contributions: Algebraic λ-calculus

- Λ_Σ :untyped Λ with finite R-valued linear combinations,
- Purpose:
  - Framework for {quantum, probabilistic, *etc.* } operational theory.
- No normal forms in presence of $-1 \in R$
  - No irreducible terms,
  - Every term reduces to every other term (collapse of $\equiv$).

Contributions: Algebraic λ-calculus

- Λ_Σ :untyped Λ with finite R-valued linear combinations,
- Purpose:
  - Framework for {quantum, probabilistic, *etc.* } operational theory.
- No normal forms in presence of $-1 \in \mathrm{R}$
  - No irreducible terms,
  - Every term reduces to every other term (collapse of $\equiv$).

Contributions:

- Normalisation scheme for the whole (untyped) Λ_Σ

  (previously, Vaux for a simply typed setting)
  - First notion of normal form for Λ_Σ,
  - $\veeeq$ does not collapse,
  - Reduction relation characterising these normal forms.
- Factorisation theorem.

Contributions: Probabilistic λ-calculus

- $\Lambda_\oplus$: untyped Λ with probabilistic choice operator,
- Purpose:
  - Program equality in H-O probabilistic functional languages.

Contributions: Probabilistic λ-calculus

- $\Lambda_\oplus$: untyped Λ with probabilistic choice operator,
- Purpose:
  - Program equality in H-O probabilistic functional languages.
- Bisimulation for proving observational equivalence
  - Bisimilarity need to be a congruence relation,
  - Sound w.r.t observational equivalence.

Contributions: Probabilistic λ-calculus

- $\Lambda_{\oplus}$: untyped Λ with probabilistic choice operator,
- Purpose:
  - Program equality in H-O probabilistic functional languages.
- Bisimulation for proving observational equivalence
  - Bisimilarity need to be a congruence relation,
  - Sound w.r.t observational equivalence.

Contributions:

- Probabilistic applicative bisimulation for $\Lambda_{\oplus}$,
- Proof of congruence for probabilistic applicative bisimilarity
  - Along the lines of Howe's method,
  - Non-trivial "disentangling" properties for sets of real numbers.
- Full abstraction on pure λ-terms.

Normal forms for the algebraic λ-calculus

Λ_Σ: Syntax and Operational Semantics [Vau09]

Definition

$$M, N ::= x \ \mid \ \lambda x.M \ \mid \ (M)\,N \ \mid \ \sum_{i=1}^{n} a_i M_i$$

subject to (algebraic) linearity:

1. $\lambda x.\left(\sum_{i=1}^{n} a_i M_i\right) = \sum_{i=1}^{n} a_i \lambda x.M_i$
2. $\left(\sum_{i=1}^{n} a_i M_i\right) N = \sum_{i=1}^{n} a_i\,(M_i)\,N$

Λ_Σ: Syntax and Operational Semantics [Vau09]

Definition (Λ_Σ)

$$\{\text{Simple terms: } \Delta_{\mathrm{R}}\} \qquad s, t \ ::= x \ \mid \ \lambda x.s \ \mid \ (s)\, T$$

$$\{\text{Terms: } \mathrm{R}\langle\Delta_{\mathrm{R}}\rangle\} \qquad S, T \ ::= \textstyle\sum_{i=1}^{n} a_i s_i$$

where $a_i \in \mathrm{R}$ and R a semiring.

Λ_Σ: Syntax and Operational Semantics [Vau09]

Definition (Λ_Σ)

$$\{\text{Simple terms: } \Delta_{\mathrm{R}}\} \qquad s, t \ ::= x \ \mid \ \lambda x.s \ \mid \ (s)\, T$$

$$\{\text{Terms: } \mathrm{R}\langle\Delta_{\mathrm{R}}\rangle\} \qquad S, T \ ::= \textstyle\sum_{i=1}^{n} a_i s_i$$

where $a_i \in \mathrm{R}$ and R a semiring.

β-reduction ($\widetilde{\to}$):

- revised β-rule: $(\lambda x.s)\, T \ \to \ s\,[T/x]$,
- extended to linear combinations:

$$au + V \ \widetilde{\to}\ aU' + V \text{ as soon as } a \neq 0 \text{ and } u \to U'.$$

Λ_Σ: Syntax and Operational Semantics [Vau09]

Definition (Λ_Σ)

$$\{\text{Simple terms: } \Delta_R\} \qquad s, t \ ::= x \ \mid \ \lambda x.s \ \mid \ (s)\, T$$

$$\{\text{Terms: } R\langle\Delta_R\rangle\} \qquad S, T \ ::= \textstyle\sum_{i=1}^{n} a_i s_i$$

where $a_i \in R$ and R a semiring.

β-reduction ($\widetilde{\to}$):

- revised β-rule: $(\lambda x.s)\, T \ \to \ s\,[T/x]$,
- extended to linear combinations:

$$au + V \ \widetilde{\to}\ aU' + V \text{ as soon as } a \neq 0 \text{ and } u \to U'.$$

Theorem ([Vau09])

Reduction $\widetilde{\to}$ enjoys confluence.

Factorisation in Pure λ-calculus

Head β-reduction as the essential part of a computation.

Theorem ([Tak95])

Any reduction $s \to^* t$ can be reorganised so that $s \to_{\mathtt{h}}^* \to_{\neg\mathtt{h}}^* t$.

Takahashi's proof: decompose $\Rightarrow$ and then swap reductions.

Factorisation in Pure λ-calculus

Head β-reduction as the essential part of a computation.

Theorem ([Tak95])

Any reduction $s \to^* t$ can be reorganised so that $s \to_{\mathtt{h}}^* \to_{\neg\mathtt{h}}^* t$.

Takahashi's proof: decompose $\Rightarrow$ and then swap reductions.

Theorem ([Bar84])

- *Leftmost Reduction (hence, Head normalisability),*
- *Standardisation.*

Factorisation in Λ_Σ

$\widetilde{\rightarrow_{\mathtt{h}}}$ does not commute with $\widetilde{\rightarrow_{\neg\mathtt{h}}}$.

Factorisation in Λ_Σ

$\widetilde{\to}_{\mathtt{h}}$ does not commute with $\widetilde{\to}_{\neg\mathtt{h}}$.

Function/Argument decomposition

$$\widetilde{\to} = \widetilde{\to}_{\mathtt{f}} \cup \widetilde{\to}_{\mathtt{a}}$$

- $\widetilde{\to}_{\mathtt{a}}$ argument reduction: $\widetilde{\to}$ in argument position,
- $\widetilde{\to}_{\mathtt{f}}$ function reduction: $\widetilde{\to}$ in function position.

Factorisation in Λ_Σ

$\widetilde{\rightarrow}_{\mathtt{h}}$ does not commute with $\widetilde{\rightarrow}_{\neg\mathtt{h}}$.

Function/Argument decomposition

$$\widetilde{\rightarrow} = \widetilde{\rightarrow}_{\mathtt{f}} \cup \widetilde{\rightarrow}_{\mathtt{a}}$$

- $\widetilde{\rightarrow}_{\mathtt{a}}$ argument reduction: $\widetilde{\rightarrow}$ in argument position,
- $\widetilde{\rightarrow}_{\mathtt{f}}$ function reduction: $\widetilde{\rightarrow}$ in function position.

Theorem

Any reduction $S \widetilde{\rightarrow}^{*} T$ can be reorganised so that $S \widetilde{\rightarrow}_{\mathtt{f}}{}^{*} \widetilde{\rightarrow}_{\mathtt{a}}{}^{*} T$.

Proof. Takahashi's with some technical adaptations.

Head Normalisability

Simple terms exhibit the structure of pure λ-terms.

Definition (Head normal form)

Let HNF_R the set of simple head normal forms:

$$\lambda x_1 \ldots \lambda x_m. ((y)\, T_1) \ldots)\, T_n$$

with $m, n \geq 0$ and $T_1, \ldots, T_n \in \mathsf{R}\langle \Delta_\mathsf{R} \rangle$.
Hence, $\mathsf{R}\langle \mathsf{HNF}_\mathsf{R} \rangle$ is the set of head normal forms.

Head Normalisability

Simple terms exhibit the structure of pure λ-terms.

Definition (Head normal form)

Let $\mathrm{HNF}_{\mathrm{R}}$ the set of simple head normal forms:

$$\lambda x_1 \ldots \lambda x_m.\,((y)\, T_1) \ldots)\, T_n$$

with $m, n \geq 0$ and $T_1, \ldots, T_n \in \mathrm{R}\langle \Delta_{\mathrm{R}} \rangle$.
Hence, $\mathrm{R}\langle \mathrm{HNF}_{\mathrm{R}} \rangle$ is the set of head normal forms.

Reduction $\widetilde{\to}_{\mathtt{f}}$ is head normalising:

Theorem

If $S \in \mathrm{P}\langle \Delta_{\mathrm{P}} \rangle$ has head normal form, then $S \; \widetilde{\to}_{\mathtt{f}}^{*} \; \mathrm{HNF}(S)$.

Normalisability (issues)

- Delicate rewriting property in Λ_Σ: it depends on the semiring R.

Normalisability (issues)

- Delicate rewriting property in Λ_Σ: it depends on the semiring R.
- If $-1 \in \mathrm{R}$, so that $1 + (-1) = 0$, reduction collapses.

Proposition [Vau09]

If R is not positive, *i.e.* $a, b \in \mathrm{R}^\bullet$ such that $a + b = 0$, S reduces.

In general, $\mathrm{R}\langle\Delta_\mathrm{R}\rangle$ has no normal forms.

Normalisability (issues)

- Delicate rewriting property in Λ_Σ: it depends on the semiring R.
- If $-1 \in \mathsf{R}$, so that $1 + (-1) = 0$, reduction collapses.

Proposition [Vau09]

If R is not positive, *i.e.* $a, b \in \mathsf{R}^\bullet$ such that $a + b = 0$, S reduces.

In general, $\mathsf{R}\langle \Delta_{\mathsf{R}} \rangle$ has no normal forms.

However:

Theorem ([Vau09])

$\mathbb{N}\langle \Delta_{\mathbb{N}} \rangle$ *is conservative with respect to pure* λ*-calculus.*

$\Rightarrow$ (strongly) normalisable terms and normal forms.

The Semiring $\mathbb{P}$

Definition

$\mathbb{P}$ is $\mathbb{N}[\Xi]$: the semiring of polynomials over a set of indeterminates Ξ, with non-negative integer coefficients.

- $\mathbb{P}$ exploitable as representation for every R (*e.g.* $\mathbb{N}, \mathbb{Q}$, *etc.*),
- $\mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$ inherits the good normalisability properties of $\mathbb{N}\langle \Delta_{\mathbb{N}} \rangle$.

The Semiring $\mathbb{P}$

Definition

$\mathbb{P}$ is $\mathbb{N}[\Xi]$: the semiring of polynomials over a set of indeterminates Ξ, with non-negative integer coefficients.

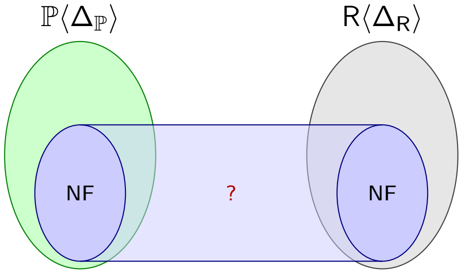
- $\mathbb{P}$ exploitable as representation for every R (*e.g.* $\mathbb{N}, \mathbb{Q}$, *etc.*),
- $\mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$ inherits the good normalisability properties of $\mathbb{N}\langle \Delta_{\mathbb{N}} \rangle$.

Idea

Characterise normal forms in $\mathsf{R}\langle \Delta_{\mathsf{R}} \rangle$ in terms of those in $\mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$.

Abstract Construction

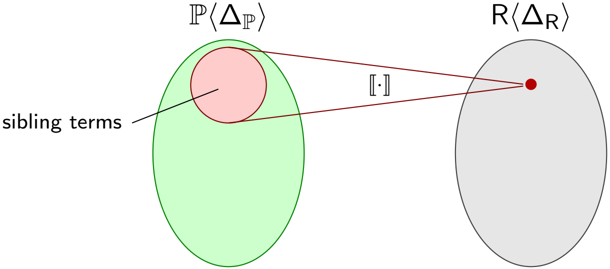
$\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ as representation of any $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$.



How can we relate the two?

Abstract Construction

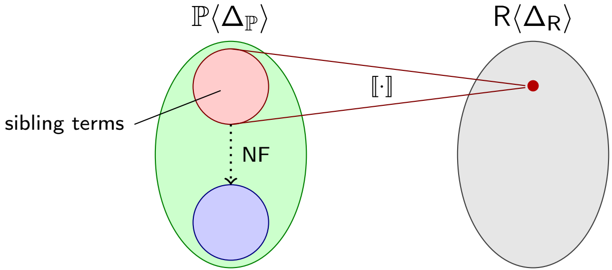
- Term evaluation $[\![\cdot]\!]$: morphism from $\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ to $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$,



Sibling terms are notations for the same term •.

Abstract Construction

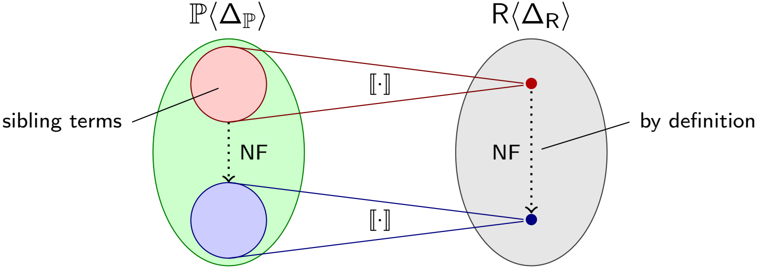
- Term evaluation $[\![\cdot]\!]$: morphism from $\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ to $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$,



$\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ exhibits normal forms.

Abstract Construction

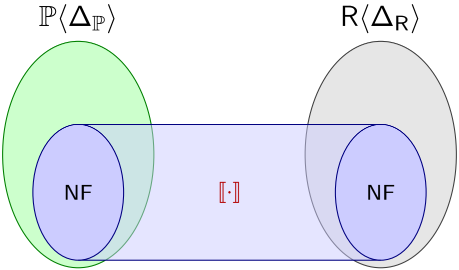
- Term evaluation $[\![\cdot]\!]$: morphism from $\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ to $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$,
- Normal forms in $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$: terms admitting a redex-free writing.



NF(sibling terms) are notations for the same term: $\bullet = \mathrm{NF}(\bullet)$.

Abstract Construction

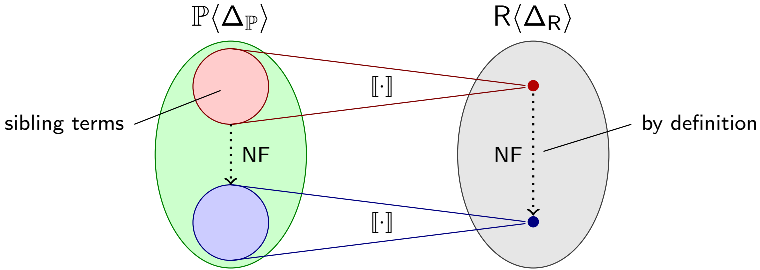
- Term evaluation $[\![\cdot]\!]$: morphism from $\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ to $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$,
- Normal forms in $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$: terms admitting a redex-free writing.



NF of $\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ induces NF of $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$ via $[\![\cdot]\!]$.

Abstract Construction

- Term evaluation $[\![\cdot]\!]$: morphism from $\mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ to $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$,
- Normal forms in $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$: terms admitting a redex-free writing.



Are Sibling Terms NF-Stable?

Strong normalisability:

Theorem

Let $S, T \in \mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ *be* SN*. If* $[\![S]\!] = [\![T]\!]$*, then* $[\![\mathrm{NF}(S)]\!] = [\![\mathrm{NF}(T)]\!]$*.*

Are Sibling Terms NF-Stable?

~~Strong~~ Normalisability:

Theorem

Let $S, T \in \mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$ *be* N. *If* $[\![S]\!] = [\![T]\!]$, *then* $[\![\mathsf{NF}(S)]\!] = [\![\mathsf{NF}(T)]\!]$.

Are Sibling Terms NF-Stable?

~~Strong~~ Normalisability:

Theorem

Let $S, T \in \mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ *be* N*. If* $[\![S]\!] = [\![T]\!]$*, then* $[\![\mathrm{NF}(S)]\!] = [\![\mathrm{NF}(T)]\!]$*.*

- Unique normal form:

Definition

Let $T \in \mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ be N with $[\![T]\!] = S \in \mathrm{R}\langle\Delta_{\mathrm{R}}\rangle$: $\mathrm{NF}(S) = [\![\mathrm{NF}(T)]\!]$.

Are Sibling Terms NF-Stable?

~~Strong~~ Normalisability:

Theorem

Let $S, T \in \mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ *be* N. *If* $[\![S]\!] = [\![T]\!]$, *then* $[\![\mathrm{NF}(S)]\!] = [\![\mathrm{NF}(T)]\!]$.

- Unique normal form:

Definition

Let $T \in \mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ be N with $[\![T]\!] = S \in \mathrm{R}\langle\Delta_{\mathrm{R}}\rangle$: $\mathrm{NF}(S) = [\![\mathrm{NF}(T)]\!]$.

- Partial, but consistent, term equivalence:

Definition

$S \veebar T$ if there are normalisable terms $U, V \in \mathbb{P}\langle\Delta_{\mathbb{P}}\rangle$ such that $[\![U]\!] = S$, $[\![V]\!] = T$ and $[\![\mathrm{NF}(U)]\!] = [\![\mathrm{NF}(V)]\!]$.

Canonical Reductions

Is it possible to directly compute these normal forms?

Canonical Reductions

Is it possible to directly compute these normal forms?

- Parallel β-reduction $\Rightarrow$,
- extended to canonical terms only:

$$\sum_{i=1}^{n} a_i s_i \mathrel{\widehat{\Rightarrow}} \sum_{i=1}^{n} a_i T_i \text{ where, } s_i \neq s_j \text{ and } s_i \Rightarrow T_i.$$

Canonical Reductions

Is it possible to directly compute these normal forms?

- Parallel β-reduction $\Rightarrow$,
- extended to canonical terms only:

$$\sum_{i=1}^{n} a_i s_i \mathrel{\widehat{\Rightarrow}} \sum_{i=1}^{n} a_i T_i \text{ where, } s_i \neq s_j \text{ and } s_i \Rightarrow T_i.$$

Theorem

For all normalisable $S \in \mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$, $[\![S]\!] \mathrel{\widehat{\Rightarrow}}^* [\![\mathrm{NF}(S)]\!]$.

Canonical Reductions

Is it possible to directly compute these normal forms?

- Parallel β-reduction $\Rightarrow$,
- extended to canonical terms only:

$$\sum_{i=1}^{n} a_i s_i \mathrel{\widehat{\Rightarrow}} \sum_{i=1}^{n} a_i T_i \text{ where, } s_i \neq s_j \text{ and } s_i \Rightarrow T_i .$$

Theorem

For all normalisable $S \in \mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$, $[\![S]\!] \mathrel{\widehat{\Rightarrow}}^* [\![\mathrm{NF}(S)]\!]$.

- Non-parallel version characterises strong normalisability only,

Canonical Reductions

Is it possible to directly compute these normal forms?

- Parallel β-reduction $\Rightarrow$,
- extended to canonical terms only:

$$\sum_{i=1}^{n} a_i s_i \mathrel{\widehat{\Rightarrow}} \sum_{i=1}^{n} a_i T_i \text{ where, } s_i \neq s_j \text{ and } s_i \Rightarrow T_i.$$

Theorem

For all normalisable $S \in \mathbb{P}\langle \Delta_{\mathbb{P}} \rangle$, $[\![S]\!] \mathrel{\widehat{\Rightarrow}}^* [\![\mathrm{NF}(S)]\!]$.

- Non-parallel version characterises strong normalisability only,
- Relations on canonical terms exhibit bad rewriting properties.

Part II

Coinductive equivalences in a probabilistic scenario

How can we establish
M equivalent to N ?

Higher-Order Program Equivalence

How can we establish
M equivalent to N ?

Context equivalence

As soon as their observable behaviour is the same in every context:

$$\text{if } \mathtt{Obs}(C\langle M\rangle, C\langle N\rangle) \text{ then } M \simeq N.$$

Higher-Order Program Equivalence

How can we establish
M equivalent to N ?

Context equivalence

As soon as their observable behaviour is the same in every context:

$$\text{if } \mathtt{Obs}(C\langle M\rangle, C\langle N\rangle) \text{ then } M \simeq N.$$

Observational equivalences:

- Subsume compositional reasonings (congruence),
- Involve universal quantification (hard),
- Amenable to coinductive formulations (bisimulation).

Lazy λ-calculus as LTS [Abr90]

Definition

$$\{\text{Terms: } \Lambda\} \quad M, N ::= x \mid \lambda x.M \mid (M)\,N$$

$$\{\text{Values: } \mathrm{V}\} \quad V, W ::= \lambda x.M \qquad \text{(closed)}$$

- Lazy operational semantics (Weak CbN):

$$\frac{}{V \Downarrow V}\ \text{(val)} \qquad \frac{M \Downarrow \lambda x.L \quad L\,[N/x] \Downarrow V}{(M)\,N \Downarrow V}\ \text{(term)}$$

Lazy λ-calculus as LTS [Abr90]

Definition

$$\{\text{Terms: } \Lambda\} \quad M, N ::= x \mid \lambda x.M \mid (M)\,N$$

$$\{\text{Values: } \mathrm{V}\} \quad V, W ::= \lambda x.M \qquad \text{(closed)}$$

- Lazy operational semantics (Weak CbN):

$$\frac{}{V \Downarrow V}\ \text{(val)} \qquad \frac{M \Downarrow \lambda x.L \quad L\,[N/x] \Downarrow V}{(M)\,N \Downarrow V}\ \text{(term)}$$

- Labelled Transition System:

$$\begin{array}{ccc} \textbf{Terms} & & \textbf{Values} \\ M & \xrightarrow{\quad eval \quad} & V \\ L\,[N/x] & \xleftarrow{\quad N \quad} & \lambda x.L \end{array}$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$M \;\; \mathcal{R} \;\; N$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$\begin{tikzcd} & & & \lambda x.L \\ M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N & \end{tikzcd}$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$\begin{tikzcd}
 & & & \lambda x.L \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & \\
 & & & \lambda x.P
\end{tikzcd}$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$
\begin{tikzcd}
 & & & \lambda x.L \arrow[drr, bend left, "Q"] & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & & L[Q/x] \\
 & & & \lambda x.P & &
\end{tikzcd}
$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$\begin{tikzcd}
 & \lambda x.L \arrow[dr, bend left, "Q"] & & \\
M \mathrel{\mathcal{R}} N \arrow[ur, bend left, "eval"] \arrow[dr, bend right, "eval"'] & & L\,[Q/x] & P\,[Q/x] \\
 & \lambda x.P \arrow[urr, bend right, "Q"'] & &
\end{tikzcd}$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$
\begin{tikzcd}
 & & & \lambda x.L \arrow[dr, bend left, "Q"] & & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & L[Q/x] & \mathcal{R} & P[Q/x] \\
 & & & \lambda x.P \arrow[urrr, bend right, "Q"'] & & &
\end{tikzcd}
$$

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$
\begin{tikzcd}
 & & & \lambda x.L \arrow[dr, bend left, "Q"] & & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & L\,[Q/x] & \mathcal{R} & P\,[Q/x] \\
 & & & \lambda x.P \arrow[urrr, bend right, "Q"'] & & &
\end{tikzcd}
$$

Bisimulation: $\mathcal{R}$ and $\mathcal{R}^{op}$ are simulations.

- Similarity $\lesssim$: union of all simulations,
- Bisimilarity $\sim$: union of all bisimulations.

Applicative (Bi)Simulation [Abr90]

When is $\mathcal{R}$ a simulation?

$$\begin{tikzcd}
 & & & \lambda x.L \arrow[dr, bend left, "Q"] & & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & L[Q/x] & \mathcal{R} & P[Q/x] \\
 & & & \lambda x.P \arrow[urrr, bend right, "Q"'] & & &
\end{tikzcd}$$

Bisimulation: $\mathcal{R}$ and $\mathcal{R}^{op}$ are simulations.

- Similarity $\lesssim$: union of all simulations,
- Bisimilarity $\sim$: union of all bisimulations.

Theorem (Applicative bisimilarity is context equivalence)

$$M \sim N \iff M \simeq N$$

$\Lambda_{\oplus}$: Syntax and Probabilistic Operational Semantics

Same language of terms of the non-deterministic λ-calculus [dP95].

Definition

$$\{\text{Terms: } \Lambda_{\oplus}\} \qquad M, N ::= x \;\mid\; \lambda x.M \;\mid\; (M)\,N \;\mid\; M \oplus N$$

$$\{\text{Values: } V\Lambda_{\oplus}\} \quad V, W ::= \lambda x.M \qquad \text{(closed)}$$

$\Lambda_\oplus$: Syntax and Probabilistic Operational Semantics

Same language of terms of the non-deterministic λ-calculus [dP95].

Definition

$$\{\text{Terms: } \Lambda_\oplus\} \qquad M, N ::= x \mid \lambda x.M \mid (M)\,N \mid M \oplus N$$

$$\{\text{Values: } V\Lambda_\oplus\} \quad V, W ::= \lambda x.M \qquad \text{(closed)}$$

Non-deterministic meaning of a term:

- A set of values

$$M \mapsto \{V, W\},$$

- Potentially infinite (Θ and $\oplus$).

$\Lambda_\oplus$: Syntax and Probabilistic Operational Semantics

Same language of terms of the non-deterministic λ-calculus [dP95].

Definition

$$\{\text{Terms: } \Lambda_\oplus\} \qquad M, N ::= x \mid \lambda x.M \mid (M)\,N \mid M \oplus N$$

$$\{\text{Values: } \mathsf{V}\Lambda_\oplus\} \quad V, W ::= \lambda x.M \qquad \text{(closed)}$$

Probabilistic meaning of a term:

- A value distribution $\mathscr{D} : \mathsf{V}\Lambda_\oplus \to \mathbb{R}_{[0,1]}$

$$M \mapsto \{V^p, W^q\},$$

- Bounded by 1: $\sum_{V \in \mathsf{V}\Lambda_\oplus} \mathscr{D}(V) \leq 1$.

$\Lambda_\oplus$: Syntax and Probabilistic Operational Semantics

Same language of terms of the non-deterministic λ-calculus [dP95].

Definition

$$\{\text{Terms: } \Lambda_\oplus\} \qquad M, N ::= x \mid \lambda x.M \mid (M)\,N \mid M \oplus N$$

$$\{\text{Values: } \mathsf{V}\Lambda_\oplus\} \quad V, W ::= \lambda x.M \qquad (\text{closed})$$

Probabilistic meaning of a term:

- A value distribution $\mathscr{D} : \mathsf{V}\Lambda_\oplus \to \mathbb{R}_{[0,1]}$

$$M \mapsto \{V^p, W^q\},$$

- Bounded by 1: $\sum_{V \in \mathsf{V}\Lambda_\oplus} \mathscr{D}(V) \leq 1$.

Problem: no finitary operational semantics suffices.

$\Lambda_\oplus$: Syntax and Probabilistic Operational Semantics

DL and Zorzi's idea: limit of approximated value distributions.

$\Lambda_\oplus$: Syntax and Probabilistic Operational Semantics

DL and Zorzi's idea: limit of approximated value distributions.

Approximation CbN operational semantics [DLZ12]:

$$\frac{}{M \Downarrow \emptyset}\ \text{(be)} \qquad \frac{}{V \Downarrow \{V^1\}}\ \text{(bv)} \qquad \frac{M \Downarrow \mathscr{D} \quad N \Downarrow \mathscr{E}}{M \oplus N \Downarrow \frac{1}{2}\mathscr{D} + \frac{1}{2}\mathscr{E}}\ \text{(bs)}$$

$$\frac{M \Downarrow \mathscr{D} \quad \{P\,[N/x] \Downarrow \mathscr{E}_{P,N}\}_{\lambda x.P \in \mathsf{Supp}(\mathscr{D})}}{(M)\,N \Downarrow \sum_{\lambda x.P \in \mathsf{Supp}(\mathscr{D})} \mathscr{D}(\lambda x.P) \cdot \mathscr{E}_{P,N}}\ \text{(ba)}$$

$\Lambda_\oplus$: Syntax and Probabilistic Operational Semantics

DL and Zorzi's idea: limit of approximated value distributions.

Approximation CbN operational semantics [DLZ12]:

$$\frac{}{M \Downarrow \emptyset}\ (\mathsf{be}) \qquad \frac{}{V \Downarrow \{V^1\}}\ (\mathsf{bv}) \qquad \frac{M \Downarrow \mathscr{D} \quad N \Downarrow \mathscr{E}}{M \oplus N \Downarrow \frac{1}{2}\mathscr{D} + \frac{1}{2}\mathscr{E}}\ (\mathsf{bs})$$

$$\frac{M \Downarrow \mathscr{D} \quad \{P\,[N/x] \Downarrow \mathscr{E}_{P,N}\}_{\lambda x.P \in \mathsf{Supp}(\mathscr{D})}}{(M)\,N \Downarrow \sum_{\lambda x.P \in \mathsf{Supp}(\mathscr{D})} \mathscr{D}(\lambda x.P) \cdot \mathscr{E}_{P,N}}\ (\mathsf{ba})$$

Definition (Semantics)

CbN semantics of $M \in \Lambda_\oplus$: $[\![M]\!] = \sup_{M \Downarrow \mathscr{D}} \mathscr{D}$.

Probabilistic (Bi)Simulation [LS91]

- Probabilistic LTS: Labelled Markov Chains,

Probabilistic (Bi)Simulation [LS91]

- Probabilistic LTS: Labelled Markov Chains,

Definition

A probabilistic bisimulation is an equivalence relation $\mathcal{R}$ on $\mathcal{S}$ such that $s\,\mathcal{R}\,t$ implies: for every $\ell \in \mathcal{L}$ and $\mathtt{E} \in \mathcal{S}/\mathcal{R}$,

$$\mathcal{P}(s, \ell, \mathtt{E}) = \mathcal{P}(t, \ell, \mathtt{E}).$$

Probabilistic simuation is required to be a preorder.

Probabilistic (Bi)Simulation [LS91]

- Probabilistic LTS: Labelled Markov Chains,

Definition

A probabilistic bisimulation is an equivalence relation $\mathcal{R}$ on $\mathcal{S}$ such that $s\,\mathcal{R}\,t$ implies: for every $\ell \in \mathcal{L}$ and $\mathtt{E} \in \mathcal{S}/\mathcal{R}$,

$$\mathcal{P}(s, \ell, \mathtt{E}) = \mathcal{P}(t, \ell, \mathtt{E}).$$

Probabilistic simuation is required to be a preorder.

Similarity ($\lesssim$) and Bisimilarity ($\sim$) can always be formed.

Proposition (Bisimilarity is similarity equivalence)

$$\sim \; = \; \lesssim \cap \lesssim^{op}$$

$\Lambda_{\oplus}$ as a Labelled Markov Chain

- States: terms and values,

$\Lambda_\oplus$ as a Labelled Markov Chain

- States: terms and values,
- Transitions:
  - M evaluates to V with $[\![M]\!]\,(V)$ probability,

$$
\begin{tikzcd}
\textbf{Terms} & \textbf{Values} \\
 & \vdots \\
 & W \\
M \arrow[ur, bend left, "{eval,\ [\![M]\!]\,(W)}"] \arrow[r, "{eval,\ [\![M]\!]\,(V)}"] \arrow[dr, bend right, "{eval,\ [\![M]\!]\,(X)}"'] & V \\
 & X \\
 & \vdots
\end{tikzcd}
$$

$\Lambda_{\oplus}$ as a Labelled Markov Chain

- States: terms and values,
- Transitions:
  - M evaluates to V with $[\![M]\!](V)$ probability,
  - Values get a term in input.

Terms **Values**

$$L[N/x] \xleftarrow{N, 1} \lambda x.L$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$\lambda x.M \;\; \mathcal{R} \;\; \lambda x.N$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$
\begin{tikzcd}
\lambda x.M \arrow[rrrr, bend left, "L"] & \mathcal{R} & \lambda x.N \arrow[rrrr, bend right, "L"'] & & M[L/x] & \mathcal{R} & N[L/x]
\end{tikzcd}
$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$\begin{tikzcd} \lambda x.M \arrow[rrr, bend left, "L"] & \mathcal{R} & \lambda x.N \arrow[rrr, bend right, "L"'] & M\,[L/x] & \mathcal{R} & N\,[L/x] \end{tikzcd}$$

- on terms: for every $\mathtt{E} \in \mathsf{V}\Lambda_{\oplus}/\mathcal{R}$,

$$M \;\; \mathcal{R} \;\; N$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$
\begin{tikzcd}
\lambda x.M \arrow[rrrr, bend left, "L"] & \mathcal{R} & \lambda x.N \arrow[rrrr, bend right, "L"'] & & M[L/x] & \mathcal{R} & N[L/x]
\end{tikzcd}
$$

- on terms: for every $\mathtt{E} \in \mathsf{V}\Lambda_{\oplus}/\mathcal{R}$,

$$
\begin{tikzcd}
 & & & [\![M]\!] \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N &
\end{tikzcd}
$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$\begin{tikzcd}
\lambda x.M \arrow[rrrr, bend left, "L"] & \mathcal{R} & \lambda x.N \arrow[rrrr, bend right, "L"'] & & M[L/x] & \mathcal{R} & N[L/x]
\end{tikzcd}$$

- on terms: for every $\mathtt{E} \in \mathsf{V}\Lambda_{\oplus}/\mathcal{R}$,

$$\begin{tikzcd}
 & & & \llbracket M \rrbracket \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & \\
 & & & \llbracket N \rrbracket
\end{tikzcd}$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$\begin{tikzcd}
\lambda x.M \arrow[rrr, bend left, "L"] & \mathcal{R} & \lambda x.N \arrow[rrr, bend right, "L"'] & M\,[L/x] & \mathcal{R} & N\,[L/x]
\end{tikzcd}$$

- on terms: for every $\mathtt{E} \in V\Lambda_{\oplus}/\mathcal{R}$,

$$\begin{tikzcd}
 & & & \llbracket M \rrbracket \arrow[rd, bend left] & \\
M \arrow[rrru, bend left, "eval"] & \mathcal{R} & N \arrow[rd, bend right, "eval"'] & & \llbracket M \rrbracket\,(\mathtt{E}) \\
 & & & \llbracket N \rrbracket &
\end{tikzcd}$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$
\begin{tikzcd}
\lambda x.M \arrow[rrrr, bend left, "L"] & \mathcal{R} & \lambda x.N \arrow[rrrr, bend right, "L"'] & & M\,[L/x] & \mathcal{R} & N\,[L/x]
\end{tikzcd}
$$

- on terms: for every $\mathtt{E} \in V\Lambda_{\oplus}/\mathcal{R}$,

$$
\begin{tikzcd}
 & & & \llbracket M \rrbracket \arrow[dr, bend left] & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & \llbracket M \rrbracket\,(\mathtt{E}) & \llbracket N \rrbracket\,(\mathtt{E}) \\
 & & & \llbracket N \rrbracket \arrow[urr, bend right] & &
\end{tikzcd}
$$

Probabilistic Applicative (Bi)Simulation

$\mathcal{R}$ bisimulation whenever equivalence relation and

- on values: for every L,

$$
\begin{tikzcd}
\lambda x.M & \mathcal{R} & \lambda x.N \arrow[rrr, bend right, "L"'] & M\,[L/x] \arrow[from=1-1, bend left, "L"] & \mathcal{R} & N\,[L/x]
\end{tikzcd}
$$

- on terms: for every $\mathtt{E} \in \mathsf{V}\Lambda_{\oplus}/\mathcal{R}$,

$$
\begin{tikzcd}
 & & & \llbracket M \rrbracket \arrow[dr, bend left] & & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & \llbracket M \rrbracket\,(\mathtt{E}) & = & \llbracket N \rrbracket\,(\mathtt{E}) \\
 & & & \llbracket N \rrbracket \arrow[urrr, bend right] & & &
\end{tikzcd}
$$

Context Equivalence vs. Bisimilarity

Definition

Contexts $C\Lambda_{\oplus}$:

$$C ::= \langle\cdot\rangle \;\mid\; \lambda x.C \;\mid\; (C)\,M \;\mid\; (M)\,C \;\mid\; C \oplus M \;\mid\; M \oplus C$$

Context equivalence: $M \simeq N \iff \sum [\![C\langle M\rangle]\!] = \sum [\![C\langle N\rangle]\!]$.

Is $\sim$ included in $\simeq$?

Context Equivalence vs. Bisimilarity

Definition

Contexts $C\Lambda_{\oplus}$:

$$C ::= \langle\cdot\rangle \;\mid\; \lambda x.C \;\mid\; (C)\,M \;\mid\; (M)\,C \;\mid\; C \oplus M \;\mid\; M \oplus C$$

Context equivalence: $M \simeq N \iff \sum [\![C\langle M\rangle]\!] = \sum [\![C\langle N\rangle]\!]$.

It is sufficient to show $\sim$ congruence:

$$\begin{aligned} M \sim N &\implies C\langle M\rangle \sim C\langle N\rangle \\ &\implies \sum [\![C\langle M\rangle]\!] = \sum [\![C\langle N\rangle]\!] \\ &\implies M \simeq N. \end{aligned}$$

Context Equivalence vs. Bisimilarity

Definition

Contexts $C\Lambda_{\oplus}$:

$$C ::= \langle\cdot\rangle \mid \lambda x.C \mid (C)\,M \mid (M)\,C \mid C \oplus M \mid M \oplus C$$

Context equivalence: $M \simeq N \iff \sum [\![C\langle M\rangle]\!] = \sum [\![C\langle N\rangle]\!]$.

It is sufficient to show $\sim$ congruence:

$$\begin{aligned} M \sim N &\implies C\langle M\rangle \sim C\langle N\rangle \\ &\implies \sum [\![C\langle M\rangle]\!] = \sum [\![C\langle N\rangle]\!] \\ &\implies M \simeq N. \end{aligned}$$

How to prove it? Direct proof fails due to application.

Howe's Technique [Pit11]

- Howe's lifting: $\lesssim$ preorder $\Rightarrow \lesssim^H$ precongruence and $\lesssim \subseteq \lesssim^H$,

$$\lesssim \qquad\qquad \lesssim^H$$

Howe's Technique [Pit11]

- Howe's lifting: $\lesssim$ preorder $\Rightarrow \lesssim^H$ precongruence and $\lesssim \subseteq \lesssim^H$,

$$\begin{tikzcd} \lesssim \arrow[rr, bend left, "\subseteq"] & & \lesssim^H \end{tikzcd}$$

Howe's Technique [Pit11]

- Howe's lifting: $\lesssim$ preorder $\Rightarrow$ $\lesssim^H$ precongruence and $\lesssim \subseteq \lesssim^H$,
- Key Lemma: $\lesssim^H$ is a simulation $\Rightarrow$ $\lesssim^H \subseteq \lesssim$ (to be proved!).

$$
\begin{tikzcd}
\lesssim \arrow[rr, bend left, "\subseteq"] & & \lesssim^H \arrow[ll, bend left, dotted, "\supseteq"]
\end{tikzcd}
$$

Key Lemma

Key Lemma ($\lesssim^H$ is a simulation)

If $M \lesssim^H N$, then for every $X \subseteq \Lambda_{\oplus}(x)$ it holds that
$$[\![M]\!]\,(\lambda x.X) \leq [\![N]\!]\,(\lambda x.(\lesssim^H(X))).$$

Proof sketch.

- For all $\mathscr{D}$ such that $M \Downarrow \mathscr{D}$, $\mathscr{D}(\lambda x.X) \leq [\![N]\!]\,(\lambda x.(\lesssim^H(X)))$,
- By induction on the (finite) derivation of $M \Downarrow \mathscr{D}$:
  - Value and probabilistic choice cases are easy,
  - Application case is hard: it requires probability assignments can always be disentangled (via max-flow min-cut theorem).

Key Lemma

Key Lemma ($\precsim^H$ is a simulation)

If $M \precsim^H N$, then for every $X \subseteq \Lambda_{\oplus}(x)$ it holds that

$$[\![M]\!]\,(\lambda x.X) \leq [\![N]\!]\,(\lambda x.(\precsim^H(X))).$$

Proof sketch.

- For all $\mathscr{D}$ such that $M \Downarrow \mathscr{D}$, $\mathscr{D}(\lambda x.X) \leq [\![N]\!]\,(\lambda x.(\precsim^H(X)))$,
- By induction on the (finite) derivation of $M \Downarrow \mathscr{D}$:
  - Value and probabilistic choice cases are easy,
  - Application case is hard: it requires probability assignments can always be disentangled (via max-flow min-cut theorem).

$\sim$ congruence, since $\sim = \precsim \cap \precsim^{op}$ and $\precsim$ precongruence:

- $\precsim \subseteq \precsim^H$ (Howe's lifting),
- $\precsim \supseteq \precsim^H$ (Key Lemma).

On Full Abstraction

- $\sim$ is sound w.r.t. $\simeq$.
- Is $\sim$ also complete?

On Full Abstraction

- $\sim$ is sound w.r.t. $\simeq$.
- Is $\sim$ also complete? No, at least in CbN.

Counterexample. Consider:

$$M = \lambda x.\lambda y.(\Omega \oplus \mathbf{I}) \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.\mathbf{I})$$

Since $\mathbf{I} \not\sim \Omega$, then

$$\lambda y.\Omega \not\sim \lambda y.\mathbf{I} \not\sim \lambda y.(\Omega \oplus \mathbf{I})$$

which implies $M \not\sim N$.

- Same counterexample to completeness in non-deterministic Λ,

On Full Abstraction

- $\sim$ is sound w.r.t. $\simeq$.
- Is $\sim$ also complete? No, at least in CbN.

Counterexample. Consider:

$$M = \lambda x.\lambda y.(\Omega \oplus \mathbf{I}) \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.\mathbf{I})$$

Since $\mathbf{I} \not\sim \Omega$, then

$$\lambda y.\Omega \not\sim \lambda y.\mathbf{I} \not\sim \lambda y.(\Omega \oplus \mathbf{I})$$

which implies $M \not\sim N$.

- Same counterexample to completeness in non-deterministic Λ,
- Nonetheless, $M \simeq N$ (via CIU-equivalence).

Full Abstraction on Pure λ-terms

What about $\sim$ vs. $\simeq$ on pure, deterministic, λ-terms?

Full Abstraction on Pure λ-terms

What about $\sim$ vs. $\simeq$ on pure, deterministic, λ-terms?

- Lévy-Longo trees are the lazy variant of Böhm trees:
  - $LT(\lambda x.\Omega) = \lambda x.\bot$ whereas $LT(\Omega) = \bot$,

Full Abstraction on Pure λ-terms

What about $\sim$ vs. $\simeq$ on pure, deterministic, λ-terms?

- Lévy-Longo trees are the lazy variant of Böhm trees:
  - $LT(\lambda x.\Omega) = \lambda x.\bot$ whereas $LT(\Omega) = \bot$,
  - Induce the finest equivalence in a lazy regime.

Definition (Lévy-Longo equality)

$$M =_{\mathsf{LL}} N \iff LT(M) = LT(N).$$

Full Abstraction on Pure λ-terms

What about $\sim$ vs. $\simeq$ on pure, deterministic, λ-terms?

- Lévy-Longo trees are the lazy variant of Böhm trees:
  - $LT(\lambda x.\Omega) = \lambda x.\bot$ whereas $LT(\Omega) = \bot$,
  - Induce the finest equivalence in a lazy regime.

Definition (Lévy-Longo equality)

$$M =_{\mathsf{LL}} N \iff LT(M) = LT(N).$$

- $\simeq \subseteq =_{\mathsf{LL}}$ via a variation of Böhm-out technique,

Full Abstraction on Pure λ-terms

What about $\sim$ vs. $\simeq$ on pure, deterministic, λ-terms?

- Lévy-Longo trees are the lazy variant of Böhm trees:
  - $LT(\lambda x.\Omega) = \lambda x.\bot$ whereas $LT(\Omega) = \bot$,
  - Induce the finest equivalence in a lazy regime.

Definition (Lévy-Longo equality)

$$M =_{\mathsf{LL}} N \iff LT(M) = LT(N).$$

- $\simeq \subseteq =_{\mathsf{LL}}$ via a variation of Böhm-out technique,
- $=_{\mathsf{LL}} \subseteq \sim$ via a notion of strict applicative bisimulation.

Full Abstraction on Pure λ-terms

What about $\sim$ vs. $\simeq$ on pure, deterministic, λ-terms?

- Lévy-Longo trees are the lazy variant of Böhm trees:
    - $LT(\lambda x.\Omega) = \lambda x.\bot$ whereas $LT(\Omega) = \bot$,
    - Induce the finest equivalence in a lazy regime.

Definition (Lévy-Longo equality)

$$M =_{\mathsf{LL}} N \iff LT(M) = LT(N).$$

- $\simeq \subseteq =_{\mathsf{LL}}$ via a variation of Böhm-out technique,
- $=_{\mathsf{LL}} \subseteq \sim$ via a notion of strict applicative bisimulation.

Theorem

On pure λ-terms: $\sim$, $\simeq$ and $=_{\mathsf{LL}}$ all coincide.

Thanks!

(Q & A*)* time.

Labelled Markov Chains

- Probabilistic labelled transition sytems,
- Discrete state space and time.

Definition

A Labelled Markov Chain is a triple $(\mathcal{S}, \mathcal{L}, \mathcal{P})$ such that:

- $\mathcal{S}$ is a countable set of states;
- $\mathcal{L}$ is set of labels;
- $\mathcal{P}$ is a transition probability matrix, i.e. a function $\mathcal{P} : \mathcal{S} \times \mathcal{L} \times \mathcal{S} \to \mathbb{R}_{[0,1]}$ such that for every $s \in \mathcal{S}$ and $\ell \in \mathcal{L}$:

$$\mathcal{P}(s, \ell, \mathcal{S}) = \sum_{t \in \mathcal{S}} \mathcal{P}(s, \ell, t) \leq 1.$$

Howe's Technique [Pit11]

- Construct $\mathcal{R}^H$ from $\mathcal{R}$ such that:
  - $\mathcal{R}^H$ is a precongruence, whenever $\mathcal{R}$ is a preorder,
  - $\mathcal{R} \subseteq \mathcal{R}^H$ and (via Key Lemma) $\mathcal{R} \supseteq \mathcal{R}^H$.

Howe's Technique [Pit11]

- Construct $\mathcal{R}^H$ from $\mathcal{R}$ such that:
  - $\mathcal{R}^H$ is a precongruence, whenever $\mathcal{R}$ is a preorder,
  - $\mathcal{R} \subseteq \mathcal{R}^H$ and (via Key Lemma) $\mathcal{R} \supseteq \mathcal{R}^H$.
- Howe's lifting for $\Lambda_\oplus$:

$$\frac{\overline{x} \vdash x \,\mathcal{R}\, M}{\overline{x} \vdash x \,\mathcal{R}^H\, M} \qquad \frac{\overline{x} \cup \{x\} \vdash M \,\mathcal{R}^H\, L \qquad \overline{x} \vdash \lambda x.L \,\mathcal{R}\, N \qquad x \notin \overline{x}}{\overline{x} \vdash \lambda x.M \,\mathcal{R}^H\, N}$$

$$\frac{\overline{x} \vdash M \,\mathcal{R}^H\, P \qquad \overline{x} \vdash N \,\mathcal{R}^H\, Q \qquad \overline{x} \vdash (P)\,Q \,\mathcal{R}\, L}{\overline{x} \vdash (M)\,N \,\mathcal{R}^H\, L}$$

$$\frac{\overline{x} \vdash M \,\mathcal{R}^H\, P \qquad \overline{x} \vdash N \,\mathcal{R}^H\, Q \qquad \overline{x} \vdash P \oplus Q \,\mathcal{R}\, L}{\overline{x} \vdash M \oplus N \,\mathcal{R}^H\, L}$$