Behavioural Metrics for Higher-Order Coalgebras

λ

Henning Urbat

Friedrich-Alexander-Universität Erlangen-Nürnberg

Dagstuhl Workshop
'Behavioural Metrics and Quantitative Logics'

October 2024

Higher-Order Coalgebras

Transition Systems

$$\begin{tikzcd}
 & s \arrow[dl, "a/\frac{1}{2}"'] \arrow[dr, "b/\frac{1}{2}"] & \\
t \arrow[d, "a/1"'] & & u \arrow[dll, "a/\frac{2}{3}"'] \arrow[d, "b/\frac{1}{3}"] \\
v & & w
\end{tikzcd}$$

Behavioural equivalence and metrics

$\rightsquigarrow$

Coalgebras

$$\gamma\colon X \to BX$$

$$(\text{Behaviour type } B\colon\ \mathbb{C} \to \mathbb{C})$$

Behavioural equivalence and metrics

Higher-Order Coalgebras

Transition Systems

$$
\begin{tikzcd}
 & s \arrow[dl, "a/\frac{1}{2}"'] \arrow[dr, "b/\frac{1}{2}"] & \\
t \arrow[d, "a/1"'] & & u \arrow[dll, "a/\frac{2}{3}"'] \arrow[d, "b/\frac{1}{3}"] \\
v & & w
\end{tikzcd}
$$

Behavioural equivalence and metrics

$\rightsquigarrow$

Coalgebras

$$\gamma\colon X \to BX$$

$$(\text{Behaviour type } B\colon\ \mathbb{C} \to \mathbb{C})$$

Behavioural equivalence and metrics

Operational Semantics of HO Languages

$$
\begin{tikzcd}
 & \Omega \oplus \lambda x.t \arrow[dl, Rightarrow, "\frac{1}{2}"'] \arrow[dr, Rightarrow, "\frac{1}{2}"] & \\
\perp & & \lambda x.t \arrow[d, "e \in \Lambda"] \\
 & & t[e/x]
\end{tikzcd}
$$

Behavioural equivalence and metrics

Higher-Order Coalgebras

Transition Systems

$$\begin{tikzcd}
 & s \arrow[dl, "a/\frac{1}{2}"'] \arrow[dr, "b/\frac{1}{2}"] & \\
t \arrow[d, "a/1"'] & & u \arrow[dll, "a/\frac{2}{3}"'] \arrow[d, "b/\frac{1}{3}"] \\
v & & w
\end{tikzcd}$$

Behavioural equivalence and metrics

$\rightsquigarrow$

Coalgebras

$$\gamma\colon X \to BX$$

(Behaviour type $B\colon\ \mathbb{C} \to \mathbb{C}$)

Behavioural equivalence and metrics

Operational Semantics of HO Languages

$$\begin{tikzcd}
 & \Omega \oplus \lambda x.t \arrow[dl, Rightarrow, "\frac{1}{2}"'] \arrow[dr, Rightarrow, "\frac{1}{2}"] & \\
\bot & & \lambda x.t \arrow[d, "e \in \Lambda"] \\
 & & t[e/x]
\end{tikzcd}$$

Behavioural equivalence and metrics

$\rightsquigarrow$

Higher-Order Coalgebras

$$\gamma\colon X \to B(X, X)$$

(Behaviour type $B\colon\ \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

Behavioural equivalence and metrics

Example: λ-Calculus

Syntax: $t, s \ ::= \ x \mid \lambda x.t \mid t\,s$

$+$

value

Big-step operational semantics: $t \Downarrow \lambda x.t'$

$$\frac{}{\lambda x.t \Downarrow \lambda x.t} \qquad \frac{t \Downarrow \lambda x.t' \quad t'[s/x] \Downarrow \lambda x.t''}{t\,s \Downarrow \lambda x.t''}$$

Example: λ-Calculus

Big-step transitions $t \Downarrow \lambda x.t'$

$\rightsquigarrow$

λ-terms

$$\gamma\colon \Lambda \to \{\perp\} + \Lambda^{\Lambda}, \quad \gamma(t) = (e \mapsto t'[e/x]) \text{ if } t \Downarrow \lambda x.t', \quad \gamma(t) = \perp \text{ else.}$$

Example: λ-Calculus

Big-step transitions $t \Downarrow \lambda x.t'$

$\rightsquigarrow$

λ-terms

$$\gamma\colon \Lambda \to \{\bot\} + \Lambda^{\Lambda}, \quad \gamma(t) = (e \mapsto t'[e/x]) \text{ if } t \Downarrow \lambda x.t', \quad \gamma(t) = \bot \text{ else.}$$

- **Determ. labelled transition system** with states Λ and labels Λ.
- **Higher-order coalgebra** $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ for $B(X, Y) = \{\bot\} + Y^X$.

Example: λ-Calculus

Big-step transitions $t \Downarrow \lambda x.t'$

λ-terms

$$\gamma\colon \Lambda \to \{\bot\} + \Lambda^{\Lambda}, \quad \gamma(t) = (e \mapsto t'[e/x]) \text{ if } t \Downarrow \lambda x.t', \quad \gamma(t) = \bot \text{ else.}$$

- **Determ. labelled transition system** with states Λ and labels Λ.
- **Higher-order coalgebra** $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ for $B(X, Y) = \{\bot\} + Y^X$.

Compositionality Theorem [Abramsky '90]

LTS-bisimilarity $\approx$ on (Λ, γ) is a congruence:

$$t \approx s \quad \text{implies} \quad C[t] \approx C[s] \text{ for every context } C[\cdot].$$

Example: λ-Calculus

Big-step transitions $t \Downarrow \lambda x.t'$

$\rightsquigarrow$

λ-terms

$$\gamma\colon \Lambda \to \{\bot\} + \Lambda^{\Lambda}, \quad \gamma(t) = (e \mapsto t'[e/x]) \text{ if } t \Downarrow \lambda x.t', \quad \gamma(t) = \bot \text{ else.}$$

- **Determ. labelled transition system** with states Λ and labels Λ.
- **Higher-order coalgebra** $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ for $B(X,Y) = \{\bot\} + Y^X$.

Compositionality Theorem [Abramsky '90]

LTS-bisimilarity $\approx$ on (Λ, γ) is a congruence:

$$t \approx s \quad \text{implies} \quad C[t] \approx C[s] \text{ for every context } C[\cdot].$$

Proof: Non-trivial (Howe's method)

Example: Affine Probabilistic λ-Calculus

Syntax: $t, s \ ::= \ x \mid \lambda x.t \mid t\,s \mid t \oplus s \mid \Omega$

if $\mathsf{FV}(t) \cap \mathsf{FV}(s) = \emptyset$

probabilistic choice

diverging term

Example: Affine Probabilistic λ-Calculus

Syntax: $t, s \ ::= \ x \mid \lambda x.t \mid t\,s \mid t \oplus s \mid \Omega$

if $\mathsf{FV}(t) \cap \mathsf{FV}(s) = \emptyset$ probabilistic choice diverging term

$+$

subdistribution of values

Big-step operational semantics: $t \Downarrow \varphi = \sum_i p_i \cdot \lambda x.t_i$

$$\frac{}{\lambda x.t \Downarrow 1 \cdot \lambda x.t} \qquad \frac{}{\Omega \Downarrow 0}$$

$$\frac{t \Downarrow \varphi \quad s \Downarrow \psi}{t \oplus s \Downarrow \frac{1}{2} \cdot \varphi + \frac{1}{2} \cdot \psi}$$

$$\frac{t \Downarrow \sum_i p_i \cdot \lambda x.t'_i \quad t'_i[s/x] \Downarrow \sum_j p_{ij} \cdot \lambda x.t''_{ij}}{t\,s \Downarrow \sum_{i,j} p_i \cdot p_{ij} \cdot \lambda x.t''_{ij}}$$

Example: Affine Probabilistic λ-Calculus

Big-step transitions $t \Downarrow \sum_i p_i \cdot \lambda x.t_i$

$$\rightsquigarrow$$

$$\gamma \colon \Lambda \to \mathcal{S}(\Lambda^\Lambda), \quad \gamma(t) = \textstyle\sum_i p_i \cdot (e \mapsto t_i[e/x]) \quad \text{if } t \Downarrow \textstyle\sum_i p_i \cdot \lambda x.t_i.$$

Example: Affine Probabilistic λ-Calculus

Big-step transitions $t \Downarrow \sum_i p_i \cdot \lambda x.t_i$

$$\rightsquigarrow$$

$$\gamma\colon \Lambda \to \mathcal{S}(\Lambda^{\Lambda}), \quad \gamma(t) = \sum_i p_i \cdot (e \mapsto t_i[e/x]) \quad \text{if } t \Downarrow \sum_i p_i \cdot \lambda x.t_i.$$

- **Labelled Markov chain** with states Λ and labels Λ.
- **Higher-order coalgebra** $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ for $B(X, Y) = \mathcal{S}(Y^X)$.

Example: Affine Probabilistic λ-Calculus

Big-step transitions $t \Downarrow \sum_i p_i \cdot \lambda x.t_i$

$$\rightsquigarrow$$

$$\gamma\colon \Lambda \to \mathcal{S}(\Lambda^{\Lambda}), \quad \gamma(t) = \sum_i p_i \cdot (e \mapsto t_i[e/x]) \quad \text{if } t \Downarrow \sum_i p_i \cdot \lambda x.t_i.$$

- **Labelled Markov chain** with states Λ and labels Λ.
- **Higher-order coalgebra** $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ for $B(X, Y) = \mathcal{S}(Y^X)$.

Compositionality Theorem [Crubillé & Dal Lago, LICS '15]

The LMC bisimulation metric d_Λ on (Λ, γ) is a congruence:

$$d_\Lambda(C[t], C[s]) \leq d_\Lambda(t, s) \quad \text{for every context } C[\cdot].$$

fails for non-affine prob. λ-calculus

Towards a General Compositionality Theorem

There are many compositionality theorems for many HO languages.

☹ complex, language-specific, ad hoc ☹

Towards a General Compositionality Theorem

There are many compositionality theorems for many HO languages.

☹ complex, language-specific, ad hoc ☹

Goal: A general, abstract, unifying

Compositionality Theorem for Higher-Order Coalgebras

Suppose that we are given:

- a 'nice' higher-order language (syntax + big-step operational rules);
- the induced higher-order coalgebra $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ on program terms.

Then the bisimulation metric d_Λ on (Λ, γ) is a congruence.

Towards a General Compositionality Theorem

There are many compositionality theorems for many HO languages.

☹ complex, language-specific, ad hoc ☹

Goal: A general, abstract, unifying

Compositionality Theorem for Higher-Order Coalgebras

Suppose that we are given:

- a 'nice' higher-order language (syntax + big-step operational rules);
- the induced higher-order coalgebra $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ on program terms.

Then the bisimulation metric d_Λ on (Λ, γ) is a congruence.

Related: Higher-Order Abstract GSOS

Theory of behavioural equivalence based on small-step semantics.

$\subseteq$ {**Goncharov, Milius, Schröder, Tsampas, Urbat**}: **LICS '23 '24 & POPL '23, '25**

Towards a General Compositionality Theorem

There are many compositionality theorems for many HO languages.

☹ complex, language-specific, ad hoc ☹

Goal: A general, abstract, unifying

Compositionality Theorem for Higher-Order Coalgebras

Suppose that we are given:

- a 'nice' higher-order language (syntax + big-step operational rules);
- the induced higher-order coalgebra $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ on program terms.

Then the bisimulation metric d_Λ on (Λ, γ) is a congruence.

Related: Higher-Order Abstract GSOS

Theory of behavioural equivalence based on small-step semantics.

$\subseteq$ {**Goncharov, Milius, Schröder, Tsampas, Urbat**}: **LICS '23 '24 & POPL '23, '25**

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

1. Syntax	1.
2. Program terms	2.
3. Congruence	3.
4. Behaviour type	4.
5. Bisimulation metric	5.
6. Big-step rules	6.
7. Operational model	7.

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial Σ-algebra Λ
3. Congruence	3.
4. Behaviour type	4.
5. Bisimulation metric	5.
6. Big-step rules	6.
7. Operational model	7.

Syntax, Abstractly

$$\Sigma\colon \mathbb{C} \to \mathbb{C}$$

Algebra

$$\Sigma A \xrightarrow{\alpha} A$$

Initial algebra

$$\begin{tikzcd} \Sigma\Lambda \arrow[r, "\iota"] \arrow[d, "\Sigma h"'] & \Lambda \arrow[d, dashed, "h"] \\ \Sigma A \arrow[r, "\alpha"] & A \end{tikzcd}$$

Free algebra on X

$$\begin{tikzcd} \Sigma\Sigma^* X \arrow[r, "\iota_X"] \arrow[d, "\Sigma h"'] & \Sigma^* X \arrow[d, dashed, "h"] & X \arrow[l, "\eta"'] \arrow[dl, "h_0"] \\ \Sigma A \arrow[r, "\alpha"] & A & \end{tikzcd}$$

Example: Polynomial functor $\Sigma X = \coprod_i X^{n_i}$ on Set

Algebra: set A with operations $f_i\colon A^{n_i} \to A$ $(i \in I)$

Initial algebra Λ: algebra of closed Σ-terms

Free algebra $\Sigma^* X$: algebra of Σ-terms in variables from X

Example: λ-Calculus [Fiore, Plotkin & Turi, LICS '99]

$$t, s ::= x \mid \lambda x.t \mid t\, s$$

$$\mathbb{C} = \mathbf{Set}^{\mathbb{F}} \qquad (\mathbb{F} = \text{finite cardinals and functions}).$$

$n \;\hat{=}\;$ untyped variable context $x_1, \ldots, x_n$

$$\ldots\text{e.g.} \quad \Lambda \in \mathbf{Set}^{\mathbb{F}}, \quad \Lambda(n) = \{\, \lambda\text{-terms in context } x_1, \ldots, x_n \,\}.$$

Example: λ-Calculus [Fiore, Plotkin & Turi, LICS '99]

$$t, s ::= x \mid \lambda x.t \mid t\, s$$

$$\mathbb{C} = \mathbf{Set}^{\mathbb{F}} \qquad (\mathbb{F} = \text{finite cardinals and functions}).$$

$n \;\hat{=}\;$ untyped variable context $x_1, \ldots, x_n$

$\ldots$ e.g. $\quad \Lambda \in \mathbf{Set}^{\mathbb{F}}, \quad \Lambda(n) = \{\, \lambda\text{-terms in context } x_1, \ldots, x_n \,\}.$

Key observation

Λ carries the initial algebra of the endofunctor $\Sigma\colon \mathbf{Set}^{\mathbb{F}} \to \mathbf{Set}^{\mathbb{F}}$,

$$\Sigma X(n) = \underbrace{n}_{x} + \underbrace{X(n+1)}_{\lambda x.\ t} + \underbrace{X(n) \times X(n)}_{t\, s}.$$

Example: λ-Calculus [Fiore, Plotkin & Turi, LICS '99]

$$t, s ::= x \mid \lambda x.t \mid t\, s$$

$$\mathbb{C} = \mathbf{Set}^{\mathbb{F}} \qquad (\mathbb{F} = \text{finite cardinals and functions}).$$

$n \;\hat{=}\;$ untyped variable context $x_1, \ldots, x_n$

…e.g. $\Lambda \in \mathbf{Set}^{\mathbb{F}}$, $\Lambda(n) = \{\, \lambda\text{-terms in context } x_1, \ldots, x_n \,\}$.

Key observation

Λ carries the initial algebra of the endofunctor $\Sigma\colon \mathbf{Set}^{\mathbb{F}} \to \mathbf{Set}^{\mathbb{F}}$,

$$\Sigma X(n) = \underbrace{n}_{x} + \underbrace{X(n+1)}_{\lambda x.\ t} + \underbrace{X(n) \times X(n)}_{t\, s}.$$

From now on: Pretend that $\Sigma\colon \mathbf{Set} \to \mathbf{Set}$ and $\Lambda \in \mathbf{Set}$.

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

Concrete	Abstract
1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial Σ-algebra Λ
3. Congruence	3. $\overline{\Sigma}$-congruence
4. Behaviour type	4.
5. Bisimulation metric	5.
6. Big-step rules	6.
7. Operational model	7.

Congruences, Abstractly [Hermida & Jacobs '98]

FRel = fuzzy relations $(X, d \colon X \times X \to [0,1])$ and nonexpansive maps

$$\begin{tikzcd} \mathbf{FRel} \arrow[r, "\overline{\Sigma}"] \arrow[d, "U"'] & \mathbf{FRel} \arrow[d, "U"] \\ \mathbf{Set} \arrow[r, "\Sigma"] & \mathbf{Set} \end{tikzcd}$$

A $\overline{\Sigma}$-congruence on an algebra $\alpha \colon \Sigma A \to A$ is a fuzzy relation d on A s. th.

$$\alpha \colon \overline{\Sigma}(A, d) \to (A, d) \qquad \text{is nonexpansive.}$$

Example: Polynomial functor $\Sigma X = X \times X$ on Set

$\overline{\Sigma}(X, d) = (X \times X, \overline{d})$ where $\overline{d}((x, y), (x', y')) = d(x, x') + d(y, y')$.

$$d \text{ congruence on } (A, \oplus) \iff d(a \oplus b, a' \oplus b') \leq d(a, a') + d(b, b').$$

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial Σ-algebra Λ
3. Congruence	3. $\overline{\Sigma}$-congruence
4. Behaviour type	4. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
5. Bisimulation metric	5. $\overline{B}$-bisimulation metric
6. Big-step rules	6.
7. Operational model	7.

Bisimulation Metrics, Abstractly

cf. [Hermida & Jacobs '98], [Baldan, Bonchi, Kerstan & König '13]

$$
\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{B}"] \arrow[d, "U^{\mathrm{op}} \times U"'] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "B"] & \mathbf{Set}
\end{tikzcd}
$$

A fuzzy $\overline{B}$-bisimulation on $\gamma\colon C \to B(C, C)$ is a fuzzy rel. d on C s. th.

$$\gamma\colon (C, d) \to \overline{B}((C, d_=), (C, d)) \qquad \text{is nonexpansive.}$$

discrete metric

Fuzzy $\bar{B}$-bisimilarity $d_\wedge$ is the greatest fuzzy bisimulation.

$d \sqsubseteq d' \iff \forall t, s.\, d(t, s) \geq d'(t, s)$

Note: $d_\wedge$ is a metric under mild assumptions on the lifting $\overline{B}$.

Example: Probabilistic λ-Calculus

$$B(X,Y) = \mathcal{S}(Y^X) = \mathcal{D}(1+Y^X)$$

subdistributions

distributions

Wasserstein lifting

$$\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{\mathrm{Hom}}"] \arrow[d, "U^{\mathrm{op}} \times U"'] \arrow[rrr, bend left, "\overline{B}"] & \mathbf{FRel} \arrow[r, "1+(-)"] \arrow[d, "U"] & \mathbf{FRel} \arrow[r, "\overline{\mathcal{D}}"] \arrow[d, "U"] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "\mathrm{Hom}"] \arrow[rrr, bend right, "B"'] & \mathbf{Set} \arrow[r, "1+(-)"] & \mathbf{Set} \arrow[r, "\mathcal{D}"] & \mathbf{Set}
\end{tikzcd}$$

Example: Probabilistic λ-Calculus

$$B(X,Y) = \mathcal{S}(Y^X) = \mathcal{D}(1 + Y^X)$$

subdistributions distributions

Wasserstein lifting

$$\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{\mathrm{Hom}}"] \arrow[d, "U^{\mathrm{op}} \times U"'] \arrow[rrrr, bend left, "\overline{B}"] & \mathbf{FRel} \arrow[r, "1+(-)"] \arrow[d, "U"] & \mathbf{FRel} \arrow[r, "\overline{\mathcal{D}}"] \arrow[d, "U"] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "\mathrm{Hom}"] \arrow[rrr, bend right, "B"'] & \mathbf{Set} \arrow[r, "1+(-)"] & \mathbf{Set} \arrow[r, "\mathcal{D}"] & \mathbf{Set}
\end{tikzcd}$$

$$\overline{\mathcal{D}}((X, d_X)) = (\mathcal{D}X, d),$$

$$d(\varphi_1, \varphi_2) = \inf\{ \sum_{x,x'} d_X(x,x') \cdot \varphi(x,x') \mid \varphi \in \mathcal{D}(X \times X),\ \mathcal{D}\pi_i(\varphi) = \varphi_i \}$$

Example: Probabilistic λ-Calculus

$$B(X, Y) = \mathcal{S}(Y^X) = \mathcal{D}(1 + Y^X)$$

subdistributions   distributions

Wasserstein lifting

$$\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{\mathrm{Hom}}"] \arrow[d, "U^{\mathrm{op}} \times U"'] \arrow[rrr, bend left, "\overline{B}"] & \mathbf{FRel} \arrow[r, "1+(-)"] \arrow[d, "U"] & \mathbf{FRel} \arrow[r, "\overline{\mathcal{D}}"] \arrow[d, "U"] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "\mathrm{Hom}"] \arrow[rrr, bend right, "B"'] & \mathbf{Set} \arrow[r, "1+(-)"] & \mathbf{Set} \arrow[r, "\mathcal{D}"] & \mathbf{Set}
\end{tikzcd}$$

Example: Probabilistic λ-Calculus

$$B(X, Y) = \mathcal{S}(Y^X) = \mathcal{D}(1 + Y^X)$$

subdistributions distributions

Wasserstein lifting

$$\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{\mathrm{Hom}}"] \arrow[d, "U^{\mathrm{op}} \times U"'] \arrow[rrr, bend left, "\overline{B}"] & \mathbf{FRel} \arrow[r, "1+(-)"] \arrow[d, "U"] & \mathbf{FRel} \arrow[r, "\overline{\mathcal{D}}"] \arrow[d, "U"] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "\mathrm{Hom}"] \arrow[rrr, bend right, "B"'] & \mathbf{Set} \arrow[r, "1+(-)"] & \mathbf{Set} \arrow[r, "\mathcal{D}"] & \mathbf{Set}
\end{tikzcd}$$

$$\overline{\mathrm{Hom}}((X, d_X), (Y, d_Y)) = (Y^X, d),$$

$$d(f, g) = \inf\{\, \varepsilon \geq 0 \mid \forall x, x'.\, d_Y(f(x), g(x')) \leq d_X(x, x') + \varepsilon \,\}$$

Note: f nonexpansive iff $d(f, f) = 0$.

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial Σ-algebra Λ
3. Congruence	3. $\overline{\Sigma}$-congruence
4. Behaviour type	4. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
5. Bisimulation metric	5. $\overline{B}$-bisimulation metric
6. Big-step rules	6. Abstract Big-Step Spec.
7. Operational model	7.

Big-Step Rules, Abstractly

$$\text{Syntax } \Sigma\colon \mathbb{C} \to \mathbb{C} \quad + \quad \text{Behaviour } B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$$

An Abstract Big-Step Specification (ABSS) is a morphism

$$\varrho\colon \Sigma B^\infty(\Lambda, \Lambda) \to B(\Lambda, \Sigma^*\Lambda)$$

- Λ: initial Σ-algebra
- $\Sigma^*\Lambda$: free Σ-algebra on Λ ('terms over terms')
- $B^\infty(\Lambda, \Lambda)$: cofree $B(\Lambda, -)$-coalgebra on Λ ('iterated behaviours')

$$C \xrightarrow{\gamma} B(\Lambda, C) \xrightarrow{B(\Lambda,\gamma)} B(\Lambda, B(\Lambda, C)) \xrightarrow{B(\Lambda,B(\Lambda,\gamma))} \cdots$$

Big-Step Rules, Abstractly

$$\text{Syntax } \Sigma\colon\ \mathbb{C} \to \mathbb{C} \quad + \quad \text{Behaviour } B\colon\ \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$$

An Abstract Big-Step Specification (ABSS) is a morphism

$$\varrho\colon \Sigma B^{\infty}(\Lambda, \Lambda) \to B(\Lambda, \Sigma^{*}\Lambda)$$

Intuition: ϱ encodes big-step operational rules into a function.

$$\frac{t \Downarrow \lambda x.t' \quad t'[s/x] \Downarrow \lambda x.t''}{t\, s \Downarrow \lambda x.t''}$$

Big-Step Rules, Abstractly

$$\text{Syntax } \Sigma \colon \mathbb{C} \to \mathbb{C} \quad + \quad \text{Behaviour } B \colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$$

An Abstract Big-Step Specification (ABSS) is a morphism

$$\varrho \colon \Sigma B^{\infty}(\Lambda, \Lambda) \to B(\Lambda, \Sigma^{*}\Lambda)$$

Intuition: ϱ encodes big-step operational rules into a function.

$$\frac{t \Downarrow \lambda x.t' \qquad t'[s/x] \Downarrow \lambda x.t''}{t\ s \Downarrow \lambda x.t''}$$

Big-Step Rules, Abstractly

$$\text{Syntax } \Sigma\colon\ \mathbb{C} \to \mathbb{C} \quad + \quad \text{Behaviour } B\colon\ \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$$

An Abstract Big-Step Specification (ABSS) is a morphism

$$\varrho\colon \Sigma B^{\infty}(\Lambda, \Lambda) \to B(\Lambda, \Sigma^{*}\Lambda)$$

Intuition: ϱ encodes big-step operational rules into a function.

$$\frac{t \Downarrow \lambda x.t' \qquad t'[s/x] \Downarrow \lambda x.t''}{t\, s \Downarrow \lambda x.t''}$$

Big-Step Rules, Abstractly

$$\text{Syntax } \Sigma\colon\ \mathbb{C} \to \mathbb{C} \quad + \quad \text{Behaviour } B\colon\ \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$$

An Abstract Big-Step Specification (ABSS) is a morphism

$$\varrho\colon \Sigma B^{\infty}(\Lambda, \Lambda) \to B(\Lambda, \Sigma^{*}\Lambda)$$

Intuition: ϱ encodes big-step operational rules into a function.

$$\frac{t \Downarrow \lambda x.t' \quad t'[s/x] \Downarrow \lambda x.t''}{t\, s \Downarrow \lambda x.t''}$$

Big-Step Rules, Abstractly

$$\text{Syntax } \Sigma\colon\ \mathbb{C} \to \mathbb{C} \quad + \quad \text{Behaviour } B\colon\ \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$$

An Abstract Big-Step Specification (ABSS) is a morphism

$$\varrho\colon \Sigma B^{\infty}(\Lambda, \Lambda) \to B(\Lambda, \Sigma^{*}\Lambda)$$

Intuition: ϱ encodes big-step operational rules into a function.

$$\frac{t \Downarrow \lambda x.t' \quad t'[s/x] \Downarrow \lambda x.t''}{t\,s \Downarrow \lambda x.t''}$$

Related: Abstract GSOS [Turi & Plotkin, LICS '97], HO Abstract GSOS

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

Concrete	Abstract
1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial Σ-algebra Λ
3. Congruence	3. $\overline{\Sigma}$-congruence
4. Behaviour type	4. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
5. Bisimulation metric	5. $\overline{B}$-bisimulation metric
6. Big-step rules	6. Abstract Big-Step Spec.
7. Operational model	7. $\gamma = \bigvee_n \gamma_n \colon \Lambda \to B(\Lambda, \Lambda)$

Abstract Big-Step Specification: Operational Model

ABSS $\varrho\colon \Sigma B^{\infty}(\Lambda,\Lambda) \to B(\Lambda,\Sigma^{*}\Lambda)$

$$\rightsquigarrow$$

Higher-order coalgebra $\gamma\colon \Lambda \to B(\Lambda,\Lambda)$

Abstract Big-Step Specification: Operational Model

$$\text{ABSS} \quad \varrho \colon \Sigma B^{\infty}(\Lambda,\Lambda) \to B(\Lambda,\Sigma^{*}\Lambda)$$

$$\Lambda \xrightarrow{\gamma_0} B(\Lambda,\Lambda)$$

$$\bot = \gamma_0$$

Intuition:

- γ_0: no information – e.g. $\gamma_0 \colon \Lambda \to \mathcal{S}(\Lambda^{\Lambda})$, $t \mapsto 0$.
-

Abstract Big-Step Specification: Operational Model

ABSS $\varrho\colon \Sigma B^{\infty}(\Lambda,\Lambda) \to B(\Lambda,\Sigma^*\Lambda)$

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}_0"'] & \Lambda \arrow[r, "\gamma_1"] & B(\Lambda,\Lambda) \\
\Sigma B^{\infty}(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^*\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}$$

$\bot = \gamma_0 \ \leq \ \gamma_1$

Intuition:

- γ_1: all $t \Downarrow \varphi$ provable from the rules with a proof tree of height 1.
-

Abstract Big-Step Specification: Operational Model

$$\text{ABSS} \quad \varrho\colon \Sigma B^\infty(\Lambda,\Lambda) \to B(\Lambda, \Sigma^*\Lambda)$$

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}_1"'] & \Lambda \arrow[r, "\gamma_2"] & B(\Lambda,\Lambda) \\
\Sigma B^\infty(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^*\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}$$

$$\bot = \gamma_0 \;\leq\; \gamma_1 \;\leq\; \gamma_2$$

Intuition:

- γ_2: all $t \Downarrow \varphi$ provable from the rules with a proof tree of height 2.
-

Abstract Big-Step Specification: Operational Model

$$\text{ABSS} \quad \varrho\colon \Sigma B^{\infty}(\Lambda,\Lambda) \to B(\Lambda,\Sigma^{*}\Lambda)$$

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}_n"'] & \Lambda \arrow[r, "\gamma_{n+1}"] & B(\Lambda,\Lambda) \\
\Sigma B^{\infty}(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^{*}\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}$$

$$\bot = \gamma_0 \;\le\; \gamma_1 \;\le\; \gamma_2 \;\le\; \cdots \;\le\; \gamma_n \;\le\; \gamma_{n+1}$$

Intuition:

- γ_n: all $t \Downarrow \varphi$ provable from the rules with a proof tree of height n.
-

Abstract Big-Step Specification: Operational Model

ABSS $\varrho \colon \Sigma B^{\infty}(\Lambda, \Lambda) \to B(\Lambda, \Sigma^{*}\Lambda)$

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}"'] & \Lambda \arrow[r, "\gamma"] & B(\Lambda,\Lambda) \\
\Sigma B^{\infty}(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^{*}\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}$$

$$\perp = \gamma_0 \leq \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_n \leq \gamma_{n+1} \leq \cdots \leq \gamma = \bigvee_n \gamma_n$$

Intuition:

- γ_n: all $t \Downarrow \varphi$ provable from the rules with a proof tree of height n.
- γ: all $t \Downarrow \varphi$ provable from the rules.

Abstract Modelling of Higher-Order Languages

Concrete/Abstract

1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial Σ-algebra Λ
3. Congruence	3. $\overline{\Sigma}$-congruence
4. Behaviour type	4. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
5. Bisimulation metric	5. $\overline{B}$-bisimulation metric
6. Big-step rules	6. Abstract Big-Step Spec.
7. Operational model	7. $\gamma = \bigvee_n \gamma_n \colon \Lambda \to B(\Lambda, \Lambda)$

Towards a General Compositionality Theorem

Goal: A general, abstract, unifying

Compositionality Theorem for Higher-Order Coalgebras

Suppose that we are given:

- a 'nice' higher-order language (syntax + big-step operational rules);
- the induced higher-order coalgebra $\gamma\colon \Lambda \to B(\Lambda, \Lambda)$ on program terms.

Then the bisimulation metric d_Λ on (Λ, γ) is a congruence.

Compositionality Theorem for Bisimulation Metrics

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}"'] & \Lambda \arrow[r, "\gamma=\bigvee_n \gamma_n"] \arrow[d, phantom, "\text{ABSS + model}"] & B(\Lambda,\Lambda) \\
\Sigma B^{\infty}(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^*\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{FRel} \arrow[r, "\overline{\Sigma}"] \arrow[d, "U"'] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set} \arrow[r, "\Sigma"] & \mathbf{Set}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{B}"] \arrow[d, "U^{\mathrm{op}}\times U"'] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "B"] & \mathbf{Set}
\end{tikzcd}$$

Compositionality Theorem for Higher-Order Coalgebras

The $\overline{B}$-bisimulation metric on $\gamma\colon \Lambda \to B(\Lambda,\Lambda)$ is a $\overline{\Sigma}$-congruence

if

a few mild conditions on $\overline{\Sigma}$, $\overline{B}$, $\leq$ **&** the ABSS ϱ is liftable.

Liftable ABSS

ABSS ϱ liftable: For **every** fuzzy relation $d\colon \Lambda \times \Lambda \to [0,1]$,

$$\varrho\colon \overline{\Sigma}\,\overline{B}^{\infty}((\Lambda, d), (\Lambda, d)) \to \overline{B}((\Lambda, d), \overline{\Sigma}^{*}(\Lambda, d)) \qquad \text{is nonexpansive.}$$

Intuition: Rules are nonexpansive.

$$\frac{t \Downarrow \sum_i p_i \cdot \lambda x.t'_i \quad t'_i[s/x] \Downarrow \sum_j p_{ij} \cdot \lambda x.t''_{ij}}{t\, s \Downarrow \sum_{i,j} p_i \cdot p_{ij} \cdot \lambda x.t''_{ij}}$$

$$\Uparrow$$

$\mu\colon \overline{\mathcal{D}}\,\overline{\mathcal{D}}(X, d) \to \overline{\mathcal{D}}(X, d)$ nonexpansive.

Compositionality Theorem for Bisimulation Metrics

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}"'] & \Lambda \arrow[r, "\gamma = \bigvee_n \gamma_n"] \arrow[d, phantom, "\text{ABSS + model}"] & B(\Lambda,\Lambda) \\
\Sigma B^{\infty}(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^{*}\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{FRel} \arrow[r, "\overline{\Sigma}"] \arrow[d, "U"'] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set} \arrow[r, "\Sigma"] & \mathbf{Set}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{FRel}^{\mathrm{op}} \times \mathbf{FRel} \arrow[r, "\overline{B}"] \arrow[d, "U^{\mathrm{op}} \times U"'] & \mathbf{FRel} \arrow[d, "U"] \\
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "B"] & \mathbf{Set}
\end{tikzcd}$$

Compositionality Theorem for Higher-Order Coalgebras

The $\overline{B}$-bisimulation metric on $\gamma\colon \Lambda \to B(\Lambda,\Lambda)$ is a $\overline{\Sigma}$-congruence

if

a few mild conditions on $\overline{\Sigma}$, $\overline{B}$, $\leq$ **&** the ABSS ϱ is liftable.

- ☺ **Liftability isolates the language-specific core of compositionality!**
- ☺ **In applications: not difficult to verify!**

Compositionality Theorem for Fibrational Bisimulations

The Compositionality Theorem generalizes

from $\begin{array}{c}\textbf{FRel}\\ \big\downarrow{\scriptstyle U}\\ \textbf{Set}\end{array}$ to **InQtl**-fibrations $\begin{array}{c}\mathbb{E}\\ \big\downarrow{\scriptstyle p}\\ \mathbb{B}\end{array}$

InQtl-fibration: every fiber $\mathbb{E}_X$ is an involutive unital quantale

$$(\,\mathbb{E}_X,\,\sqsubseteq,\,\cdot,\,1,\,(-)^\circ\,)$$

abstracting composition $\cdot$ and reversal $(-)^\circ$ of fuzzy relations.

Compositionality Theorem for Fibrational Bisimulations

$$\begin{tikzcd}
\Sigma\Lambda \arrow[r, "\iota", "\cong"'] \arrow[d, "\Sigma\hat{\gamma}"'] & \Lambda \arrow[r, "\gamma = \bigvee_n \gamma_n"] \arrow[d, phantom, "\text{ABSS} + \text{model}"] & B(\Lambda,\Lambda) \\
\Sigma B^{\infty}(\Lambda,\Lambda) \arrow[rr, "\varrho"] & & B(\Lambda,\Sigma^*\Lambda) \arrow[u, "B(\Lambda,\hat{\iota})"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{E} \arrow[r, "\overline{\Sigma}"] \arrow[d, "p"'] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{B} \arrow[r, "\Sigma"] & \mathbb{B}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{E}^{\mathsf{op}} \times \mathbb{E} \arrow[r, "\overline{B}"] \arrow[d, "p^{\mathsf{op}} \times p"'] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{B}^{\mathsf{op}} \times \mathbb{B} \arrow[r, "B"] & \mathbb{B}
\end{tikzcd}$$

Compositionality Theorem for Higher-Order Coalgebras

The $\overline{B}$-bisimilarity object on $\gamma\colon \Lambda \to B(\Lambda,\Lambda)$ is a $\overline{\Sigma}$-congruence

if

a few mild conditions on $\overline{\Sigma}$, $\overline{B}$, $\leq$ **&** the ABSS ϱ is liftable.

Proof: Fibrational generalization of Howe's method.

Applications

Compositionality Theorems for Bisimulations/Bisimulation Metrics

- Deterministic λ-calculus [Abramsky '90]
- Probabilistic λ-calculus [Crubillé & Dal Lago, LICS' 15]

Applications

Compositionality Theorems for Bisimulations/Bisimulation Metrics

- Deterministic λ-calculus [Abramsky '90]
- Probabilistic λ-calculus [Crubillé & Dal Lago, LICS' 15]
- Nondeterministic λ-calculus
  $\mathcal{S} \quad \rightsquigarrow \quad \mathcal{P}$ (power set monad with Hausdorff lifting $\overline{\mathcal{P}}$ to **FRel**)

Applications

Compositionality Theorems for Bisimulations/Bisimulation Metrics

- Deterministic λ-calculus [Abramsky '90]
- Probabilistic λ-calculus [Crubillé & Dal Lago, LICS' 15]
- Nondeterministic λ-calculus
  $\mathcal{S} \rightsquigarrow \mathcal{P}$ (power set monad with Hausdorff lifting $\overline{\mathcal{P}}$ to **FRel**)
- Nondeterministic probabilistic λ-calculus
  $\mathcal{S} \rightsquigarrow \mathcal{C}$ (convex sets of distributions monad)

Applications

Compositionality Theorems for Bisimulations/Bisimulation Metrics

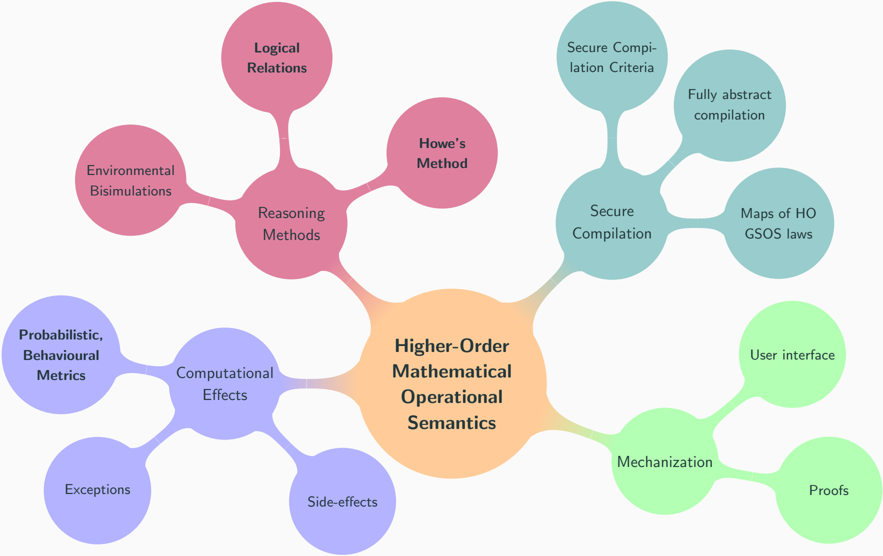
- Deterministic λ-calculus [Abramsky '90]
- Probabilistic λ-calculus [Crubillé & Dal Lago, LICS' 15]
- Nondeterministic λ-calculus
  $\mathcal{S} \rightsquigarrow \mathcal{P}$ (power set monad with Hausdorff lifting $\overline{\mathcal{P}}$ to **FRel**)
- Nondeterministic probabilistic λ-calculus
  $\mathcal{S} \rightsquigarrow \mathcal{C}$ (convex sets of distributions monad)
- Effectful λ-calculus (?)
  $\mathcal{S} \rightsquigarrow \mathbb{T}$ (monad on **Set** with a lifting $\overline{\mathbb{T}}$ to **FRel**)

Applications

Compositionality Theorems for Bisimulations/Bisimulation Metrics

- Deterministic λ-calculus [Abramsky '90]
- Probabilistic λ-calculus [Crubillé & Dal Lago, LICS' 15]
- Nondeterministic λ-calculus
  $\mathcal{S} \quad \rightsquigarrow \quad \mathcal{P}$ (power set monad with Hausdorff lifting $\overline{\mathcal{P}}$ to **FRel**)
- Nondeterministic probabilistic λ-calculus
  $\mathcal{S} \quad \rightsquigarrow \quad \mathcal{C}$ (convex sets of distributions monad)
- Effectful λ-calculus (?)
  $\mathcal{S} \quad \rightsquigarrow \quad \mathbb{T}$ (monad on **Set** with a lifting $\overline{\mathbb{T}}$ to **FRel**)
- Continuous probabilistic λ-calculus (??)
  $\mathcal{S} \quad \rightsquigarrow \quad \mathcal{G}$ (Giry monad on nice measurable spaces)

Big Picture



Logical Relations
Howe's Method
Environmental Bisimulations
Reasoning Methods
Secure Compilation Criteria
Fully abstract compilation
Secure Compilation
Maps of HO GSOS laws
Probabilistic, Behavioural Metrics
Computational Effects
Higher-Order Mathematical Operational Semantics
User interface
Mechanization
Proofs
Exceptions
Side-effects