Introduction to Higher-Order Mathematical Operational Semantics

Sergey Goncharov

In(tro)duction to Higher-Order Mathematical Operational Semantics

Sergey Goncharov

HO Mathematical Operational Semantics Project

- Goncharov, Milius, Schröder, Tsampas, and Urbat, “Towards a Higher-Order Mathematical Operational Semantics”, POPL 2023
- Urbat, Tsampas, Goncharov, Milius, and Schröder, “Weak Similarity in Higher-Order Mathematical Operational Semantics”, LICS 2023
- Goncharov, Santamaria, Schröder, Tsampas, and Urbat, “Logical Predicates in Higher-Order Mathematical Operational Semantics”, FoSSaCS 2024
- Goncharov, Milius, Tsampas, and Urbat, “Bialgebraic Reasoning on Higher-Order Program Equivalence”, LICS 2024
- Goncharov, Tsampas, and Urbat, “Abstract Operational Methods for Call-by-Push-Value”, POPL 2025

.. and rolling

Context: Applied Category Theory

- Somewhat like
  `https://en.wikipedia.org/wiki/Applied_category_theory`
- More specifically: using category theory as an organization device
- Yet more specially: proving classes of statements at once

Example: Coalgebraic Modal Logic

- **Parameters:** functor F (for transitions), predicate liftings $\heartsuit\colon 2^X \to 2^{FX}$ (for modalities), axioms
- **Results:** soundness, completeness, Hennessy–Milner property, complexity bounds, etc.

Context: Applied Category Theory

- Somewhat like
  `https://en.wikipedia.org/wiki/Applied_category_theory`
- More specifically: using category theory as an organization device
- Yet more specially: proving classes of statements at once

Example: Coalgebraic Modal Logic

- **Parameters:** functor F (for transitions), predicate liftings $\heartsuit\colon 2^X \to 2^{FX}$ (for modalities), axioms
- **Results:** soundness, completeness, Hennessy–Milner property, complexity bounds, etc.

Our **Grand Objective**: develop a similar approach to operational semantics

Semantics: Operational v.s. Denotational

- Operational Semantics (how programs behave?)

$$
\begin{aligned}
s(s(0)) + s(s(0)) &\rightarrow s(s(0) + s(s(0))) \\
&\rightarrow s(s(0 + s(s(0)))) \rightarrow s(s(s(s(0))))
\end{aligned}
$$

- Denotational Semantics (what programs denote?)

$$
[\![s(s(0)) + s(s(0))]\!] = [\![s(s(0))]\!] + [\![s(s(0))]\!] = 2 + 2 = 4
$$

Semantics in Use

Denotational:

- ☺ Compositional by design:

$$[\![p]\!] = [\![q]\!] \Rightarrow [\![C[p]]\!] = [\![C[q]]\!]$$

for any program context C

- ☺ Mathematically rigorous and precise
- ☹ Ease to define: from hard to impossible

Operational:

- ☺ Lightweight and easy to define even for complex languages
- ☹ Nonuniform and fraggile
- ☹ Hard to reason about (because of lack of compositionality)

Overall: not (sufficiently) mathematical

Semantics in Use

Denotational:

- ☺ Compositional by design:

$$[\![p]\!] = [\![q]\!] \Rightarrow [\![C[p]]\!] = [\![C[q]]\!]$$

for any program context C

- ☺ Mathematically rigorous and precise
- ☹ Ease to define: from hard to impossible

Operational:

- ☺ Lightweight and easy to define even for complex languages **!**
- ☹ Nonuniform and fraggile
- ☹ Hard to reason about (because of lack of compositionality)

Overall: not (sufficiently) mathematical

First-Order Abstract GSOS

First-Order Abstract GSOS

Turi and Plotkin's abstraction of GSOS rule format[1]:

- Signature endo-functor Σ
- Behaviour endo-functor B
- GSOS law – natural transformation $\rho_X \colon \Sigma(X \times BX) \to B(\Sigma^\star X)$

Example (Process Algebra):

- $\Sigma = \{\, \mathrm{par}\,/2, \emptyset/0\} \cup \{a.\,(-)/1 \mid a \in A\}$
- $BX = \mathcal{P}(A \times X)$
- GSOS law encodes rules like:

$$\frac{p \xrightarrow{a} p'}{p \;\; \mathrm{par} \;\; q \;\; \xrightarrow{a} \;\; p' \;\; \mathrm{par} \;\; q}$$

[1] Turi and Plotkin, "Towards a Mathematical Operational Semantics".

First-Order Abstract GSOS

Turi and Plotkin's abstraction of GSOS rule format[1]:

- Signature endo-functor Σ
- Behaviour endo-functor B
- GSOS law – natural transformation $\rho_X \colon \Sigma(X \times BX) \to B(\Sigma^\star X)$

Example (Process Algebra):

- $\Sigma = \{\, \mathsf{par}\,/2, \emptyset/0\} \cup \{a.\,(-)/1 \mid a \in A\}$
- $BX = \mathcal{P}(A \times X)$
- GSOS law encodes rules like:

behaviour from B

operation from Σ

$$\frac{p \xrightarrow{a} p'}{p \;\mathsf{par}\; q \xrightarrow{a} p' \;\mathsf{par}\; q}$$

[1] Turi and Plotkin, "Towards a Mathematical Operational Semantics".

First-Order Abstract GSOS

Turi and Plotkin's abstraction of GSOS rule format[1]:

- Signature endo-functor Σ
- Behaviour endo-functor B
- GSOS law – natural transformation $\rho_X \colon \Sigma(X \times BX) \to B(\Sigma^\star X)$

Example (Process Algebra):

- $\Sigma = \{\,\mathsf{par}/2, \emptyset/0\} \cup \{a.\,(-)/1 \mid a \in A\}$
- $BX = \mathcal{P}(A \times X)$
- GSOS law encodes rules like:

$$\frac{p \xrightarrow{a} p'}{p \;\mathsf{par}\; q \xrightarrow{a} p' \;\mathsf{par}\; q}$$

[1] Turi and Plotkin, "Towards a Mathematical Operational Semantics".

First-Orded GSOS

Theory of first-order GSOS takes Σ, B, ρ as input parameters, and produces

- ☺ operational semantics $\gamma\colon \Sigma^\star\emptyset \to B(\Sigma^\star\emptyset)$ (operational model)
- ☺ notion of program equivalence $\sim \subseteq \Sigma^\star\emptyset \times \Sigma^\star\emptyset$ (strong bisimilarity)
- ☺ generic compositionality: $p \sim q \Rightarrow C[p] \sim C[q]$ for any context

But

- ☹ $\sim$ is too fine-grained for programming languages
- ☹ first-order $\subsetneq$ higher-order $\leadsto$ no λ-calculus

First-Orded GSOS

Theory of first-order GSOS takes Σ, B, ρ as input parameters, and produces

- ☺ operational semantics $\gamma \colon \Sigma^\star\emptyset \to B(\Sigma^\star\emptyset)$ (operational model)
- ☺ notion of program equivalence $\sim\ \subseteq \Sigma^\star\emptyset \times \Sigma^\star\emptyset$ (strong bisimilarity)
- ☺ generic compositionality: $p \sim q \Rightarrow C[p] \sim C[q]$ for any context

But

- ☹ $\sim$ is too fine-grained for programming languages
- ☹ first-order $\subsetneq$ higher-order $\rightsquigarrow$ no λ-calculus ⬅

Higher-Order Abstract GSOS

(Call-by-Name, Extended) Combinatory Logic

- I (=$\lambda p.\, p$)
- K (=$\lambda p.\lambda q.\, p$)
- S (=$\lambda p.\lambda q.\lambda r.\, (p \cdot r) \cdot (q \cdot r)$)
- plus S', S'' and K' for partially reduced terms

$$I \xrightarrow{p} p \qquad K \xrightarrow{p} K'(p) \qquad K'(p) \xrightarrow{q} p \qquad S \xrightarrow{p} S'(p)$$

$$S'(p) \xrightarrow{q} S''(p, q) \qquad S''(p, q) \xrightarrow{r} (p \cdot r) \cdot (q \cdot r)$$

$$\frac{p \rightarrow p'}{p \cdot q \rightarrow p' \cdot q} \qquad \frac{p \xrightarrow{q} p'}{p \cdot q \rightarrow p'}$$

Example: $S \cdot p \cdot q \cdot r \rightarrow S'(p) \cdot q \cdot r \rightarrow S''(p, q) \cdot r \rightarrow (p \cdot r) \cdot (q \cdot r)$

- This is very similar to original GSOS, but it is not

(Call-by-Name, Extended) Combinatory Logic

- I (=$\lambda p.\, p$)
- K (=$\lambda p.\lambda q.\, p$)
- S (=$\lambda p.\lambda q.\lambda r.\, (p \cdot r) \cdot (q \cdot r)$)
- plus S', S'' and K' for partially reduced terms

$$I \xrightarrow{p} p \qquad K \xrightarrow{p} K'(p) \qquad K'(p) \xrightarrow{q} p \qquad S \xrightarrow{p} S'(p)$$

$$S'(p) \xrightarrow{q} S''(p, q) \qquad S''(p, q) \xrightarrow{r} (p \cdot r) \cdot (q \cdot r)$$

$$\frac{p \to p'}{p \cdot q \to p' \cdot q} \qquad \frac{p \xrightarrow{q} p'}{p \cdot q \to p'} \quad !$$

Example: $S \cdot p \cdot q \cdot r \to S'(p) \cdot q \cdot r \to S''(p, q) \cdot r \to (p \cdot r) \cdot (q \cdot r)$

- This is very similar to original GSOS, but it is not

Higher-Order Abstract GSOS

A higher-order GSOS law consists of

- Signature Σ
- Mixed variance (!) behaviour functor B
- Family of maps $\rho_{X,Y} \colon \Sigma(X \times B(X,Y)) \to B(X, \Sigma^\star(X+Y))$ natural in Y and dinatural in X

Example: For combinatory logic: $B(X,Y) = Y^X + Y$, ρ is induced by rules of operational semantics

☺ Most (but not all!) of Turi and Plotkin's theory caries over

☺ In particular: strong applicative bisimilarity $\sim$ is implied and is a congruence

Strong Applicative Bisimilarity

- Strong applicative bisimilarity is the largest symmetric relation $\sim$ on programs, such that
  1. $p \rightarrow p' \wedge p \sim q \;\; \Rightarrow \;\; \exists q'.\, q \rightarrow q' \wedge p' \sim q'$
  2. $p \xrightarrow{t} p' \wedge p \sim q \;\; \Rightarrow \;\; \exists q'.\, q \xrightarrow{t} q' \wedge p' \sim q'$
- Key expected property – congruence: $p \sim q$ implies $C[p] \sim C[q]$
- Then it is sound method for proving contextual equivalence of p and q (for every C, $C[p]$ and $C[q]$ are "behaviorally equivalent")

Strong Applicative Bisimilarity: Examples

Bisimilar Terms:

- $\Omega_0 = (SII)(SII)$, $\Omega_1 = (I(SII))(I(SII))$, $\Omega_2 = (II(SII))(I(SII))$, etc.
- Then $\Omega_0 = (SII)(SII) \to (S'(I)I)(SII) \to (S''(I,I))(SII) \to \Omega_1$
- Analogously, $\Omega_n \to^\star \Omega_{n+1}$
- Proof that $\Omega_n \sim \Omega_{n+1}$:
  - $\mathcal{R} = \{(P,Q) \mid \Omega_0 \to^\star P, \Omega_0 \to^\star Q\}$ is bisimulation,
  - hence $(\Omega_n, \Omega_m) \in \mathcal{R} \subseteq \sim$

Non-Bisimilar Terms: Generally, for $P \to Q$, $P \not\sim Q$

E.g. $K'(I)I \to I$, but $I \not\to$

Proving Congruence

- ☹ Structural induction won't do: given $p \sim q$, $S'(p) \sim S'(q)$ would require $S''(q, t) \sim S''(p, t)$
- ☹ Proving that

  $$\widehat{\sim} = \{(C[p], C[q]) \mid p \sim q\}$$

  is bisimulation won't do, for $S''(s, t)p \to (sp)(tp)$
- ☹ Proving that $\{(C[\overline{p}], C[\overline{q}]) \mid p_i \sim q_i\}$ is bisimulation won't do: $p_1 p_2 \sim q_1 q_2$ may require $p' \sim q'$ where $p_1 \xrightarrow{p_2} p'$ and $q_1 \xrightarrow{q_2} q'$
- ☺ Worked approach: bisimulation up-to congruence $\iff$ $\widehat{\sim}^{\star}$ is a bisimulation. Hence $\widehat{\sim} \subseteq \widehat{\sim}^{\star} \subseteq \sim$

Proving Congruence

- ☹ Structural induction won't do: given $p \sim q$, $S'(p) \sim S'(q)$ would require $S''(q,t) \sim S''(p,t)$
- ☹ Proving that

$$\widehat{\sim} = \{(C[p], C[q]) \mid p \sim q\}$$

  is bisimulation won't do, for $S''(s,t)p \to (sp)(tp)$
- ☹ Proving that $\{(C[\overline{p}], C[\overline{q}]) \mid p_i \sim q_i\}$ is bisimulation won't do: $p_1 p_2 \sim q_1 q_2$ may require $p' \sim q'$ where $p_1 \xrightarrow{p_2} p'$ and $q_1 \xrightarrow{q_2} q'$
- ☺ Worked approach: bisimulation up-to congruence $\iff$ $\widehat{\sim}^\star$ is a bisimulation. Hence $\widehat{\sim} \subseteq \widehat{\sim}^\star \subseteq \sim$

Key insight of HO-MOS: This generalizes to: regular categories, mono-preserving B and "finitary" Σ

!

Lambda-Calculus

- Standard rules

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

- We take: $\mathsf{C} = \mathsf{Set}^{\mathbb{F}}$ ($\mathbb{F}$ – category of finite cardinals)

$$\Sigma\colon \mathsf{C} \to \mathsf{C} \qquad\qquad \Sigma X = V + \delta X + X \times X$$

$$B\colon \mathsf{C}^{\mathsf{op}} \times \mathsf{C} \to \mathsf{C} \qquad B(X, Y) = \langle\!\langle X, Y \rangle\!\rangle \times (Y + Y^X + 1)$$

$V(n) = \{n\}$ (variables), $(\delta X)(n) = X(n+1)$, $\langle\!\langle X, Y \rangle\!\rangle(n) = \mathsf{C}(X^n, Y)$

- $(\Sigma^\star\emptyset)(n) \cong \lambda$-terms over n free variables[2]

[2] Fiore, Plotkin, and Turi, "Abstract Syntax and Variable Binding".

Lambda-Calculus

- Standard rules

"substitution" behavior

"reduction" behavior

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

- We take: $\mathsf{C} = \mathsf{Set}^{\mathbb{F}}$ ($\mathbb{F}$ – category of finite cardinals)

$$\Sigma\colon \mathsf{C} \to \mathsf{C} \qquad \Sigma X = V + \delta X + X \times X$$

$$B\colon \mathsf{C}^{\mathrm{op}} \times \mathsf{C} \to \mathsf{C} \qquad B(X, Y) = \langle\!\langle X, Y \rangle\!\rangle \times (Y + Y^X + 1)$$

$V(n) = \{n\}$ (variables), $(\delta X)(n) = X(n+1)$, $\langle\!\langle X, Y \rangle\!\rangle(n) = \mathsf{C}(X^n, Y)$

- $(\Sigma^\star \emptyset)(n) \cong \lambda$-terms over n free variables[2]

[2] Fiore, Plotkin, and Turi, "Abstract Syntax and Variable Binding".

Further Advances

First-Orded GSOS [Recall]

Theory of first-order GSOS takes Σ, B, ρ as input parameters, and produces

- ☺ operational semantics $\gamma\colon \Sigma^\star\emptyset \to B(\Sigma^\star\emptyset)$ (operational model)
- ☺ notion of program equivalence $\sim\, \subseteq \Sigma^\star\emptyset \times \Sigma^\star\emptyset$ (strong bisimilarity)
- ☺ generic compositionality: $p \sim q \Rightarrow C[p] \sim C[q]$ for any context

But

- ☹ $\sim$ is too fine-grained for programming languages ←
- ☹ first-order $\subsetneq$ higher-order $\rightsquigarrow$ no λ-calculus

Weak Bisimilarity

Weak Applicative (Bi)similarity

- Weak transitions: let $\Rightarrow$ be $\rightarrow^\star$, $\overset{t}{\Rightarrow}$ be $(\Rightarrow \cdot \overset{t}{\rightarrow})$
- Weak applicative similarity – largest such $\lesssim$ that
  1. $t \rightarrow t'$ implies $s \Rightarrow s'$ and $t' \lesssim s'$ for some s'
  2. $t \overset{r}{\rightarrow} t'$ implies $s \overset{r}{\Rightarrow} s'$ and $t' \lesssim s'$ for some s'
- Weak applicative bisimilarity $\approx = \lesssim \cap \gtrsim$

Example: $f \lesssim S \cdot (K \cdot I) \cdot f$ (analogue of $f \lesssim \lambda x.\, fx$), but

$S \cdot (K \cdot I) \cdot \Omega \not\lesssim \Omega$, because

$$S \cdot (K \cdot I) \cdot \Omega \overset{t}{\rightarrow} (K \cdot I \cdot t)(\Omega \cdot t) \quad \text{but} \quad \Omega \rightarrow \Omega \rightarrow \ldots$$

Key Property: pre-congruence of $\lesssim$

Proof Idea: Bisimulation up-to Howe's closure (Howe's method)

Howe's Method

For a relation $\mathcal{R} \subseteq \Sigma^\star\emptyset \times \Sigma^\star\emptyset$, Howe's closure $\mathcal{R}^H$ is generated by

$$\frac{}{t \; \mathcal{R}^H \; t} \qquad\qquad \frac{t_1 \; \mathcal{R}^H \; t_1' \quad \dots \quad t_n \; \mathcal{R}^H \; t_n' \quad f(t_1', \dots, t_n') \; \mathcal{R} \; s}{f(t_1, \dots, t_n) \; \mathcal{R}^H \; s}$$

! This is crucially weaker than saying that $\mathcal{R}^H$ is closure of $\mathcal{R}$ under contexts and transitivity

Key property: $\approx^H$ is a bisimulation

→ $\widehat{\approx} \subseteq \approx^H \subseteq \approx$

→ $\approx$ is congruence

Analogously: $\lesssim$ is pre-congruence

Howe's Method Abstractly

- Together with $\gamma\colon \Sigma^\star\emptyset \to B(\Sigma^\star\emptyset, \Sigma^\star\emptyset)$ for strong transitions "$\to$", we require new parameter

$$\widetilde{\gamma}\colon \Sigma^\star\emptyset \to B(\Sigma^\star\emptyset, \Sigma^\star\emptyset)$$

  for weak transitions "$\Rightarrow$"

- We need liftings to the category of relations:

$$\begin{array}{ccc} \mathsf{Rel}_{\mathsf{C}} & \xrightarrow{\overline{\Sigma}} & \mathsf{Rel}_{\mathsf{C}} \\ {\scriptstyle |-|}\downarrow & & \downarrow{\scriptstyle |-|} \\ \mathsf{C} & \xrightarrow{\Sigma} & \mathsf{C} \end{array} \qquad \begin{array}{ccc} \mathsf{Rel}_{\mathsf{C}}^{\mathsf{op}} \times \mathsf{Rel}_{\mathsf{C}} & \xrightarrow{\overline{B}} & \mathsf{Rel}_{\mathsf{C}} \\ {\scriptstyle |-|^{\mathsf{op}} \times |-|}\downarrow & & \downarrow{\scriptstyle |-|} \\ \mathsf{C}^{\mathsf{op}} \times \mathsf{C} & \xrightarrow{B} & \mathsf{C} \end{array}$$

- .. and some other assumptions

Result: $\lesssim$ (i.e. largest relation R, such that $R \leqslant (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(\Delta, R)]$) is a congruence (i.e. $\overline{\Sigma}(\lesssim) \leqslant (\iota \times \iota)^{-1}[\lesssim]$)

Step-Indexing Logical Relations

Step-Indexing, Concretely

The step-indexed logical relation $\mathcal{L}$ for combinatory logic is the inductively defined family $(\mathcal{L}^\alpha \subseteq \Sigma^\star\emptyset \times \Sigma^\star\emptyset)_{\alpha \leqslant \omega}$:

$$\mathcal{L}^0 = \top, \qquad \mathcal{L}^{n+1} = \mathcal{L}^n \cap \mathcal{E}(\mathcal{L}^n) \cap \mathcal{V}(\mathcal{L}^n, \mathcal{L}^n), \qquad \mathcal{L}^\omega = \bigcap_{n<\omega} \mathcal{L}^n$$

where $\mathcal{E}$ and $\mathcal{V}$ are relation transformers:

$$\mathcal{E}(R) = \{(t,s) \mid \text{if } t \to t' \text{ then } \exists s'.\, s \Rightarrow s' \wedge R(t',s')\}$$

$$\mathcal{V}(Q,R) = \{(t,s) \mid \text{for all } r_1, r_2,\ Q(r_1,r_2), \text{ if } t \xrightarrow{r_1} t' \text{ then } \exists s'.\, s \overset{r_2}{\Rightarrow} s' \wedge R(t',s')\}$$

As a slogan: "related programs applied to related arguments produce related results"

Step-Indexing: Properties

- $\mathcal{L}^\omega$ is a fixpoint $\mathcal{L}^\omega = \mathcal{L}^\omega \cap \mathcal{E}(\mathcal{L}^\omega) \cap \mathcal{V}(\mathcal{L}^\omega, \mathcal{L}^\omega)$
- In first-order case we would reduce to the familiar fixpoint theory and Kleene/Knaster-Tarski theorems, but because of higher-order, we generally do not (!)
- Every $\mathcal{L}^\alpha$ is a congruence
- $\mathcal{L}^\omega$ is sound for contextual preorder

Bottom Line: Step-indexing is another sound method for proving contextual equivalence

Step-Indexing, Abstractly

Under similar (but weaker) assumptions, as for weak applicative bisimilarity,

1. There is an abstract (ordinal-indexed) logical relation $(\mathcal{L}^\alpha)_\alpha$
2. The limit $\mathcal{L}^\gamma = \bigcap_\alpha \mathcal{L}^\alpha$ exists
3. Every $\mathcal{L}^\alpha$ is a congruence
4. $\mathcal{L}^\gamma$ is sound for contextual equivalence

Conclusions

✓ We broke the barrier between (co-)algebraic methods and higher-order semantics

? A lot to be done:

- Call-by-value
- Big-step semantics (and equivalence thereof)
- Other behavioral equivalence (e.g. trace equivalence)
- Metric, probabilistic, quantialic, fibrational generalizations
- Modelling polymorphic languages
- Modelling effectful languages
- $\mathcal{L}^{\nu}$ is included in applicative bisimilarity. When they are equal?

Thank You for Your Attention!

Higher-Order Abstract GSOS

Categorical Framework for Higher-Order Operational Semantics

Language

Signature ≅ Endofunctor $\Sigma\colon \mathbf{C} \to \mathbf{C}$ on category $\mathbf{C}$, e.g.:

- $\mathbf{C} = \mathbf{Set}$, $\Sigma = \{0/0, a_i/0, +/2, \cdot/2\}$
- $\mathbf{C}$ = "nominal sets", $\Sigma X = \mathbb{A} + [\mathbb{A}]X + X \times X$

Behaviour

Behaviour = Mixed-variance functor $B\colon \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \to \mathbf{C}$, e.g.:

- $B(X,Y) = Y^X + Y$ (deterministic)
- $B(X,Y) = \mathscr{P}_\omega(Y^X + Y)$ (non-deterministic)

HO GSOS Specification

$$\frac{t \xrightarrow{a} t' \quad s \xrightarrow{\bar{a}} s'}{t \| s \xrightarrow{\tau} t' \| s'} \qquad \frac{t \xrightarrow{s} t'}{t\,s \longrightarrow t'} \qquad \frac{}{(\lambda x.\, f)s \longrightarrow f[s/x]}$$

$\Longrightarrow$ **Distributivity** ρ of Σ over B

HO Bialgebraic Semantics

Transition semantics is a unique solution $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$:

$$\begin{array}{ccccc}
\Sigma\mu\Sigma & \xrightarrow{\iota} & \mu\Sigma & \xrightarrow{\gamma} & B(\mu\Sigma,\mu\Sigma) \\
\downarrow{\scriptstyle \Sigma\langle \mathrm{id},\gamma\rangle} & & & & \uparrow{\scriptstyle B(\mathrm{id},\hat{\iota})} \\
\Sigma(\mu\Sigma \times B(\mu\Sigma,\mu\Sigma)) & \xrightarrow{\rho} & B(\mu\Sigma, \Sigma^\star(\mu\Sigma + \mu\Sigma)) & \xrightarrow{B(\mathrm{id},\Sigma^\star\nabla)} & B(\mu\Sigma, \Sigma^\star\mu\Sigma)
\end{array}$$

Generic Strong HO Bisimulation

Coalgebraic notion of **strong applicative bisimilarity** $\sim$ on initial Σ-algebra $\mu\Sigma$ (=algebra of programs) as pullback

$$\begin{array}{ccc}
\sim & \longrightarrow & \mu\Sigma \\
\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{coit}\,\gamma} \\
\mu\Sigma & \xrightarrow{\mathrm{coit}\,\gamma} & \nu\gamma.B(\mu\Sigma,\gamma)
\end{array}$$

Bialgebras Form a Category

If $f : A \to A'$ and $g : A' \to A''$ are ρ-bialgebra morphisms then so is the composition $g \cdot f$, for the diagram

$$
\begin{array}{ccccccc}
A & & & \xrightarrow{\quad c \quad} & & & B(A, A) \\
\downarrow{\scriptstyle f} & & & & & & \downarrow{\scriptstyle B(A,f)} \\
A' & & \xrightarrow{\quad c' \quad} & & B(A', A') & \xrightarrow{B(f,A')} & B(A, A') \\
\downarrow{\scriptstyle g} & & & & \downarrow{\scriptstyle B(A',g)} & & \downarrow{\scriptstyle B(A,g)} \\
A'' & \xrightarrow{c''} & B(A'', A'') & \xrightarrow{B(g,A'')} & B(A', A'') & \xrightarrow{B(f,A'')} & B(A, A'')
\end{array}
$$

obviously commutes.

Lambda-Calculus

- Operational semantics rules

$$\frac{s \to s'}{s\ t \to s'\ t} \qquad \frac{}{(\lambda x.s)\ t \to s[t/x]}$$

- $\mathsf{C} = \mathsf{Set}^{\mathbb{F}}$, where $\mathbb{F}$ is the category of finite cardinals

$$\Sigma \colon \mathsf{C} \to \mathsf{C}, \qquad \Sigma X = V + \delta X + X \times X,$$

$$B \colon \mathsf{C}^{\mathsf{op}} \times \mathsf{C} \to \mathsf{C}, \qquad B(X, Y) = \langle\!\langle X, Y \rangle\!\rangle \times (Y + Y^X + 1)$$

  where Y^X is exponent in $\mathsf{Set}^{\mathbb{F}}$, V is the presheaf of variables $\mathsf{Set}^{\mathbb{F}}(n) = n$, $(\delta X)(n) = X(n+1)$, $\langle\!\langle X, Y \rangle\!\rangle(n) = \mathsf{Set}^{\mathbb{F}}(X^n, Y)$

- $\mu\Sigma$ is the presheaf $\Lambda \in \mathsf{Set}^{\mathbb{F}}$ of λ-terms over n free variables
- H/O GSOS law is pointed:

  $\rho_{X,Y} \colon \Sigma(jX \times B(jX, Y)) \to B(jX, \Sigma^\star(jX + Y))$[3]

[3] Fiore, Plotkin, and Turi, "Abstract Syntax and Variable Binding".

Dinaturality

A dinatural transformation from $F\colon \mathsf{C}^{\mathrm{op}} \times \mathsf{C} \to \mathsf{D}$ to $G\colon \mathsf{C}^{\mathrm{op}} \times \mathsf{C} \to \mathsf{D}$ is a family $(\sigma_X\colon F(X,X) \to G(X,X))_{X \in \mathsf{C}}$, such that

$$\begin{array}{ccccccc} & & F(X,X) & \xrightarrow{\sigma_X} & G(X,X) & & \\ & \overset{F(f,X)}{\nearrow} & & & & \overset{G(X,f)}{\searrow} & \\ F(Y,X) & & & & & & G(X,Y) \\ & \underset{F(X,f)}{\searrow} & & & & \underset{G(f,Y)}{\nearrow} & \\ & & F(Y,Y) & \xrightarrow{\sigma_Y} & G(Y,Y) & & \end{array}$$

for every $f\colon X \to Y$

Example: apply: $X^Y \times Y \to X$

Proving the "η-Law"

- Recall:

$$t \overset{r}{\Rightarrow} s = (\exists t'.\, t \Rightarrow t' \wedge t' \overset{r}{\longrightarrow} s) \vee (\exists t'.\, t \Rightarrow t' \wedge s = t'\ r)$$

$$\mathcal{L}^{n+1} = \mathcal{L}^{n} \cap \mathcal{E}(\mathcal{L}^{n}) \cap \mathcal{V}(\mathcal{L}^{n}, \mathcal{L}^{n})$$

$$\mathcal{E}(\mathcal{L}^{n}) = \{(t, s) \mid \text{if } t \to t' \text{ then } \exists s'.\, s \Rightarrow s' \wedge \mathcal{L}^{n}(t', s')\}$$

$$\mathcal{V}(\mathcal{L}^{n}, \mathcal{L}^{n}) = \{(t, s) \mid \text{for all } r_1, r_2,\ \mathcal{L}^{n}(r_1, r_2),$$

$$\text{if } t \overset{r_1}{\longrightarrow} t' \text{ then } \exists s'.\, s \overset{r_2}{\Rightarrow} s' \wedge \mathcal{L}^{n}(t', s')\}$$

- Proof of $\mathcal{L}^{n}(S \cdot (K \cdot I) \cdot f, f)$ by induction on n, in particular:

$$\begin{array}{ccccccccc}
S \cdot (K \cdot I) \cdot f & \to & S'(K \cdot I) \cdot f & \to & S''(K \cdot I, f) & \overset{t}{\to} & (K \cdot I \cdot t) \cdot (f \cdot t) & \overset{\star}{\to} & f \cdot t \\
\mid \mathcal{L}^{n} & & \mid \mathcal{L}^{n-1} & & \mid \mathcal{L}^{n-2} & & \mid \mathcal{L}^{n-3} & & \mid \mathcal{L}^{n-6} \\
f & \Longrightarrow & f & \Longrightarrow & f & \overset{t'}{\Longrightarrow} & f \cdot t' & \Longrightarrow & f \cdot t'
\end{array}$$

References I

- Fiore, Marcelo P., Gordon D. Plotkin, and Daniele Turi. “Abstract Syntax and Variable Binding”. In: *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*. IEEE Computer Society, 1999, pp. 193–202. URL: `https://doi.org/10.1109/LICS.1999.782615`.
- Goncharov, Sergey, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat. “Towards a Higher-Order Mathematical Operational Semantics”. In: Proc. ACM Program. Lang. 7 (2023), pp. 632–658. DOI: `10.1145/3571215`.
- Goncharov, Sergey, Stefan Milius, Stelios Tsampas, and Henning Urbat. “Bialgebraic Reasoning on Higher-Order Program Equivalence”. In: *LICS*. 2024, pp. 1–13.

References II

Goncharov, Sergey, Alessio Santamaria, Lutz Schröder, Stelios Tsampas, and Henning Urbat. “Logical Predicates in Higher-Order Mathematical Operational Semantics”. In: *Foundations of Software Science and Computation Structures - 27th International Conference, FoSSaCS 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part II*. Ed. by Naoki Kobayashi and James Worrell. Vol. 14575. Lecture Notes in Computer Science. Springer, 2024, pp. 47–69. DOI: 10.1007/978-3-031-57231-9_3. URL: https://doi.org/10.1007/978-3-031-57231-9_3.

References III

- Goncharov, Sergey, Stelios Tsampas, and Henning Urbat. "Abstract Operational Methods for Call-by-Push-Value". In: Proc. ACM Program. Lang. (2025). accepted.
- Turi, D. and G. Plotkin. "Towards a Mathematical Operational Semantics". In: *Logic in Computer Science*. IEEE. 1997, pp. 280–291.
- Urbat, Henning, Stelios Tsampas, Sergey Goncharov, Stefan Milius, and Lutz Schröder. "Weak Similarity in Higher-Order Mathematical Operational Semantics". In: *LICS*. 2023, pp. 1–13. DOI: 10.1109/LICS56636.2023.10175706. URL: https://doi.org/10.1109/LICS56636.2023.10175706.