Bialgebraic Reasoning on Higher-Order Program Equivalence

λ

Henning Urbat
j.w.w. Sergey Goncharov, Stefan Milius, Stelios Tsampas

Friedrich-Alexander-Universität Erlangen-Nürnberg

LICS 2024

Contextual Equivalence

When are two programs p, q of a higher-order language equivalent?

e.g. λ-calculus

Contextual Equivalence

$$p \approx_{\mathrm{ctx}} q$$

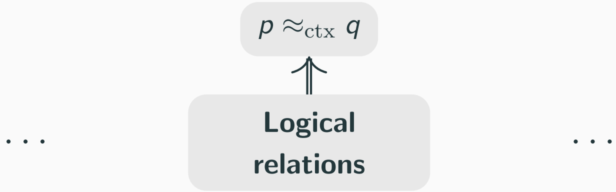
iff

for all contexts $C[\cdot]$:

$C[p]$ terminates $\iff$ $C[q]$ terminates.

Hard to prove directly $\rightsquigarrow$ need efficient proof techniques!

Proving Contextual Equivalence: Logical Relations



$p \approx_{\mathrm{ctx}} q$

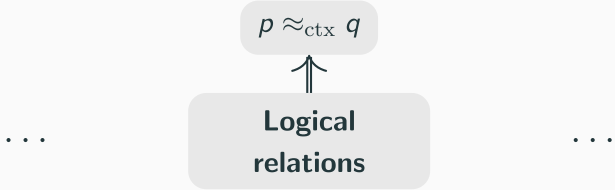
$\Uparrow$

. . .

Logical relations

. . .

Proving Contextual Equivalence: Logical Relations



$p \approx_{\mathrm{ctx}} q$

$\Uparrow$

. . . **Logical relations** . . .

☺ Powerful, robust, widely applicable proof technique.

☹ **Ad hoc:**
Each language needs tailor-made notions + (complex) soundness proof.

Proving Contextual Equivalence: Logical Relations

$p \approx_{\mathrm{ctx}} q$

$\Uparrow$

. . . **Logical relations** . . .

This Talk

Generic, language-independent approach based on

cf. Turi & Plotkin, LICS'97

Higher-Order Abstract GSOS

⇝ abstract bialgebraic theory of higher-order operational semantics.

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Categorical Abstraction

Syntax

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model

$$\gamma\colon \Lambda \to \Lambda + \Lambda^{\Lambda}$$

higher-order LTS on λ-terms

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Categorical Abstraction

Syntax ⟵ **Fiore, Plotkin, Turi, LICS'99**

$$p, q ::= x \mid p\, q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\, q \to p[q/x]} \qquad \frac{p \to p'}{p\, q \to p'\, q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model

$$\gamma \colon \Lambda \to \Lambda + \Lambda^{\Lambda}$$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\, q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\, q \to p[q/x]} \qquad \frac{p \to p'}{p\, q \to p'\, q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model

$$\gamma\colon \Lambda \to \Lambda + \Lambda^{\Lambda}$$

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model

$$\gamma\colon \Lambda \to \Lambda + \Lambda^{\Lambda}$$

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Oper. model ($B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

higher-order coalgebra

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Oper. model ($B\colon \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\, q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\, q \to p[q/x]} \qquad \frac{p \to p'}{p\, q \to p'\, q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-order GSOS law

$$\begin{tikzcd}\Sigma(X \times B(X,Y)) \arrow[d, "\varrho_{X,Y}"] \\ B(X, \Sigma^*(X+Y))\end{tikzcd}$$

dinat. in X, nat. in Y

Oper. model ($B\colon \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-order GSOS law

$$\begin{tikzcd} \Sigma(X \times B(X,Y)) \arrow[d, "\varrho_{X,Y}"] \\ B(X, \Sigma^{*}(X+Y)) \end{tikzcd}$$

dinat. in X, nat. in Y

Oper. model ($B\colon \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-order GSOS law

$$\begin{tikzcd}\Sigma(X \times B(X,Y)) \arrow[d, "\varrho_{X,Y}"] \\ B(X, \Sigma^*(X+Y))\end{tikzcd}$$

dinat. in X, nat. in Y

Oper. model ($B\colon \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\,q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\,q \to p[q/x]} \qquad \frac{p \to p'}{p\,q \to p'\,q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-order GSOS law

$$\begin{tikzcd}
\Sigma(X \times B(X,Y)) \arrow[d, "\varrho_{X,Y}"] \\
B(X, \Sigma^*(X+Y))
\end{tikzcd}$$

dinat. in X, nat. in Y

Oper. model ($B\colon \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Higher-Order Abstract GSOS [POPL'23]

Untyped CBN λ-calculus

Syntax ($\Sigma\colon \mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{F}}$)

$$p, q ::= x \mid p\, q \mid \lambda x.p$$

Operational rules

$$\frac{}{(\lambda x.p)\, q \to p[q/x]} \qquad \frac{p \to p'}{p\, q \to p'\, q}$$

$$\frac{}{\lambda x.p \xrightarrow{q} p[q/x]}$$

Oper. model ($B(X,Y) = Y + Y^X$)

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

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-order GSOS law

$$\begin{tikzcd}
\Sigma(X \times B(X,Y)) \arrow[d, "\varrho_{X,Y}"] \\
B(X, \Sigma^*(X+Y))
\end{tikzcd}$$

dinat. in X, nat. in Y

Oper. model ($B\colon \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Higher-Order Abstract GSOS [POPL'23]

Instances:

- Untyped, typed
- CBN, CBV, CBPV
- Computational effects: nondeterminism, probabilities

Categorical Abstraction

Syntax ($\Sigma\colon \mathbb{C} \to \mathbb{C}$)

Initial algebra $\mu\Sigma$

Higher-order GSOS law

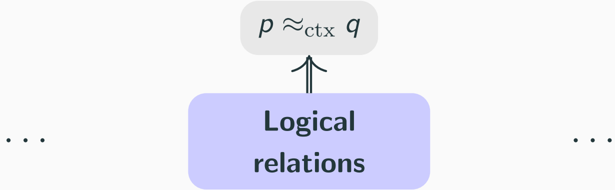
$$\begin{tikzcd}
\Sigma(X \times B(X,Y)) \arrow[d, "\varrho_{X,Y}"] \\
B(X, \Sigma^*(X+Y))
\end{tikzcd}$$

← dinat. in X, nat. in Y

Oper. model ($B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$)

$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Proving Contextual Equivalence: Logical Relations



$p \approx_{\mathrm{ctx}} q$

$\Uparrow$

. . .

Logical relations

. . .

This Talk

Generic, language-independent approach based on

Higher-Order Abstract GSOS

$\rightsquigarrow$ abstract bialgebraic theory of higher-order operational semantics.

Logical Relations (Untyped CBN λ-Calculus)

Step-indexed logical relation $\mathcal{L}_n \rightarrowtail \Lambda \times \Lambda \ (n \in \mathbb{N})$

$\mathcal{L}_0(p,q)$ always, and $\mathcal{L}_{n+1}(p,q)$ iff $\mathcal{L}_n(p,q)$ and

$$\begin{array}{lcl} p \to p' & \Longrightarrow & \exists q'.\, q \to^\star q' \ \wedge\ \mathcal{L}_n(p',q') \\ p = \lambda x.p' & \Longrightarrow & \exists q'.\, q \to^\star \lambda x.q' \ \wedge\ \forall \mathcal{L}_n(d,e).\ \mathcal{L}_n(p'[d/x], q'[e/x]). \end{array}$$

("Related functions send related inputs to related outputs")

Logical Relations (Untyped CBN λ-Calculus)

Step-indexed logical relation $\mathcal{L}_n \rightarrowtail \Lambda \times \Lambda \ (n \in \mathbb{N})$

$\mathcal{L}_0(p,q)$ always, and $\mathcal{L}_{n+1}(p,q)$ iff $\mathcal{L}_n(p,q)$ and

$$\begin{array}{lcl} p \to p' & \Longrightarrow & \exists q'.\, q \to^\star q' \ \wedge\ \mathcal{L}_n(p',q') \\ p = \lambda x.p' & \Longrightarrow & \exists q'.\, q \to^\star \lambda x.q' \ \wedge\ \forall \mathcal{L}_n(d,e).\ \mathcal{L}_n(p'[d/x], q'[e/x]). \end{array}$$

("Related functions send related inputs to related outputs")

Soundness Theorem

$[\forall n.\, \mathcal{L}_n(p,q) \wedge \mathcal{L}_n(q,p)] \implies p \approx_{\mathrm{ctx}} q.$

$$\Uparrow$$

Congruence Theorem

Each $\mathcal{L}_n \rightarrowtail \Lambda \times \Lambda$ is a congruence, i.e. respected by language operations.

Logical Relations (Untyped CBN λ-Calculus)

Step-indexed logical relation $\mathcal{L}_n \rightarrowtail \Lambda \times \Lambda \ (n \in \mathbb{N})$

$\mathcal{L}_0(p, q)$ always, and $\mathcal{L}_{n+1}(p, q)$ iff $\mathcal{L}_n(p, q)$ and

$$p \to p' \implies \exists q'.\, q \to^{\star} q' \ \wedge\ \mathcal{L}_n(p', q')$$

$$p = \lambda x.p' \implies \exists q'.\, q \to^{\star} \lambda x.q' \ \wedge\ \forall \mathcal{L}_n(d, e).\ \mathcal{L}_n(p'[d/x], q'[e/x]).$$

("Related functions send related inputs to related outputs")

Soundness Theorem

$[\forall n.\, \mathcal{L}_n(p, q) \wedge \mathcal{L}_n(q, p)] \implies p \approx_{\text{ctx}} q.$

$$\Uparrow$$

Congruence Theorem

Each $\mathcal{L}_n \rightarrowtail \Lambda \times \Lambda$ is a congruence, i.e. respected by language operations.

Key idea: Categorical abstraction via **relation liftings!**

Relation Liftings

Rel$(\mathbb{C})$: Cat. of relations $R \rightarrowtail X \times X$ and relation-preserving morphisms

A **relation lifting** of $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$ is a bifunctor $\overline{B}$ such that

$$\begin{tikzcd}
\mathbf{Rel}(\mathbb{C})^{\mathsf{op}} \times \mathbf{Rel}(\mathbb{C}) \arrow[r, "\overline{B}"] \arrow[d] & \mathbf{Rel}(\mathbb{C}) \arrow[d] \\
\mathbb{C}^{\mathsf{op}} \times \mathbb{C} \arrow[r, "B"] & \mathbb{C}
\end{tikzcd}$$

Relation Liftings

Rel($\mathbb{C}$): Cat. of relations $R \rightarrowtail X \times X$ and relation-preserving morphisms

A **relation lifting** of $B\colon\ \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$ is a bifunctor $\overline{B}$ such that

$$\begin{tikzcd}
\mathbf{Rel}(\mathbb{C})^{\mathsf{op}} \times \mathbf{Rel}(\mathbb{C}) \arrow[r, "\overline{B}"] \arrow[d] & \mathbf{Rel}(\mathbb{C}) \arrow[d] \\
\mathbb{C}^{\mathsf{op}} \times \mathbb{C} \arrow[r, "B"] & \mathbb{C}
\end{tikzcd}$$

Example: $B(X, Y) = Y^X$ on Set

$$(R \subseteq X \times X,\ S \subseteq Y \times Y) \quad \mapsto \quad \overline{B}(R, S) \subseteq Y^X \times Y^X$$

where

$$\overline{B}(R, S)(f, g) \quad \text{iff} \quad \forall x, x'.\, R(x, x') \implies S(fx, gx')$$

Logical Relations via Relation Liftings

Step-indexed logical relation $\mathcal{L}_n \rightarrowtail \Lambda \times \Lambda \; (n \in \mathbb{N})$

$\mathcal{L}_0(p, q)$ always, and $\mathcal{L}_{n+1}(p, q)$ iff $\mathcal{L}_n(p, q)$ and

$$
\begin{aligned}
p \to p' \quad &\Longrightarrow \quad \exists q'.\, q \to^\star q' \;\wedge\; \mathcal{L}_n(p', q') \\
p = \lambda x.p' \quad &\Longrightarrow \quad \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall \mathcal{L}_n(d, e).\; \mathcal{L}_n(p'[d/x], q'[e/x]).
\end{aligned}
$$

("Related functions send related inputs to related outputs")

Equivalently: $\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(\mathcal{L}_n, \mathcal{L}_n)]$

$\to$ $\quad \to^\star$ $\quad$ relation lifting

$$\begin{tikzcd}
\mathbf{Rel}(\mathbb{C})^{\mathrm{op}} \times \mathbf{Rel}(\mathbb{C}) \arrow[r, "\overline{B}"] \arrow[d] & \mathbf{Rel}(\mathbb{C}) \arrow[d] \\
\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \arrow[r, "B"] & \mathbb{C}
\end{tikzcd}$$

Logical Relations in Higher-Order Abstract GSOS

$$\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(\mathcal{L}_n, \mathcal{L}_n)]$$

$\to$ $\to^\star$ relation lifting

Logical Relations in Higher-Order Abstract GSOS

$$\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(\mathcal{L}_n, \mathcal{L}_n)]$$

$\to$ $\to^\star$ relation lifting

Congruence Theorem for Operational Model $\gamma \colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

Each $\mathcal{L}_n \rightarrowtail \mu\Sigma \times \mu\Sigma$ is a congruence

if

the weak operational model $\widetilde{\gamma}$ is a **lax higher-order bialgebra**.

rules remain sound for weak transitions

$$\frac{p \to p'}{p\, q \to p'\, q} \quad \rightsquigarrow \quad \frac{p \to^\star p'}{p\, q \to^\star p'\, q}$$

cf. Bonchi, Petrişan, Pous, Rot, CONCUR'15

Logical Relations in Higher-Order Abstract GSOS

$$\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\underset{\to}{\gamma} \times \underset{\to^\star}{\widetilde{\gamma}})^{-1}[\underset{\text{relation lifting}}{\overline{B}}(\mathcal{L}_n, \mathcal{L}_n)]$$

Congruence Theorem for Operational Model $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

Each $\mathcal{L}_n \rightarrowtail \mu\Sigma \times \mu\Sigma$ is a congruence

if

the weak operational model $\widetilde{\gamma}$ is a **lax higher-order bialgebra**.

Logical Relations in Higher-Order Abstract GSOS

$$\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\underset{\to}{\gamma} \times \underset{\to^\star}{\widetilde{\gamma}})^{-1}[\underset{\text{relation lifting}}{\overline{B}}(\mathcal{L}_n, \mathcal{L}_n)]$$

Congruence Theorem for Operational Model $\gamma \colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

Each $\mathcal{L}_n \rightarrowtail \mu\Sigma \times \mu\Sigma$ is a congruence

if

the weak operational model $\widetilde{\gamma}$ is a **lax higher-order bialgebra**.

- **Corollary:** Soundness for abstract contextual equivalence (see paper).

Logical Relations in Higher-Order Abstract GSOS

$$\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(\mathcal{L}_n, \mathcal{L}_n)]$$

$\to$ $\to^\star$ relation lifting

Congruence Theorem for Operational Model $\gamma \colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

Each $\mathcal{L}_n \rightarrowtail \mu\Sigma \times \mu\Sigma$ is a congruence

if

the weak operational model $\widetilde{\gamma}$ is a **lax higher-order bialgebra**.

- **Corollary:** Soundness for abstract contextual equivalence (see paper).
- **Lax bialg. condition:** isolates language-specific core + easy to check.

Logical Relations in Higher-Order Abstract GSOS

$$\mathcal{L}_{n+1} = \mathcal{L}_n \wedge (\underset{\to}{\gamma} \times \underset{\to^\star}{\widetilde{\gamma}})^{-1}[\underset{\text{relation lifting}}{\overline{B}}(\mathcal{L}_n, \mathcal{L}_n)]$$

Congruence Theorem for Operational Model $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

Each $\mathcal{L}_n \rightarrowtail \mu\Sigma \times \mu\Sigma$ is a congruence

if

the weak operational model $\widetilde{\gamma}$ is a **lax higher-order bialgebra**.

- **Corollary:** Soundness for abstract contextual equivalence (see paper).
- **Lax bialg. condition:** isolates language-specific core + easy to check.
- **Related:** Soundness of applicative similarity, Howe's method [LICS'23].

Perspectives [→ Poster Session]

- **Higher-Order Abstract GSOS** (POPL'23, JFP)
  - Reasoning Methods
    - **Logical Relations** (FoSSaCS'24, LICS'24)
    - **Howe's Method** (LICS'23)
    - Environmental Bisimulations
  - Secure Compilation
    - Secure Compilation Criteria
    - Fully abstract compilation
    - Maps of HO GSOS laws
  - Computational Effects
    - Probabilistic, Behavioural Distances
    - Exceptions
    - Side-effects
  - Mechanization
    - User interface
    - Proofs