Operational Techniques for Higher-Order Coalgebras

λ

Henning Urbat

Friedrich-Alexander-Universität Erlangen-Nürnberg

DutchCATS

June 2024

Recent Work

S. Goncharov, S. Milius, L. Schröder, S. Tsampas, H. Urbat:
Towards a Higher-Order Mathematical Operational Semantics.
POPL'23 & JFP Special Issue

H. Urbat, S. Tsampas, S. Goncharov, S. Milius, L. Schröder:
Weak Similarity in Higher-Order Mathematical Operational Semantics.
LICS'23

S. Goncharov, A. Santamaria, L. Schröder, S. Tsampas, H. Urbat:
Logical Predicates in Higher-Order Mathematical Operational Semantics.
FOSSACS'24

S. Goncharov, S. Milius, S. Tsampas, H. Urbat:
Bialgebraic Reasoning on Higher-Order Program Equivalence.
LICS'24

Contextual Equivalence [Morris '68]

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

λ-calculus, Haskell, OCaml, ...

Contextual Preorder

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

iff

for all program contexts $C[\cdot]$:

$$C[p] \text{ terminates } \implies C[q] \text{ terminates.}$$

Contextual Equivalence

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

iff

$$p \lesssim_{\mathrm{ctx}} q \text{ and } q \lesssim_{\mathrm{ctx}} p.$$

☹ Hard to reason about directly $\rightsquigarrow$ need efficient (coinductive) techniques!

Coinductive Proof Techniques for Contextual Preorder

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

Applicative simulations

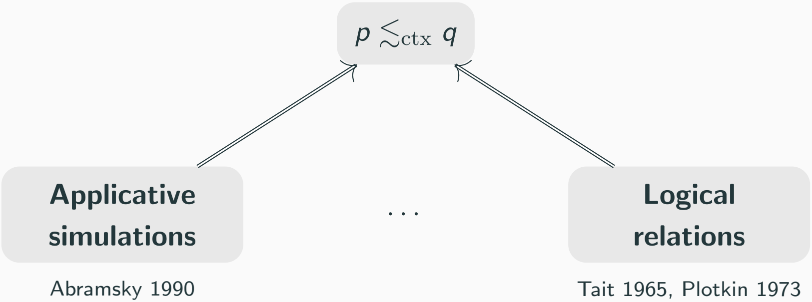
Abramsky 1990

. . .

Logical relations

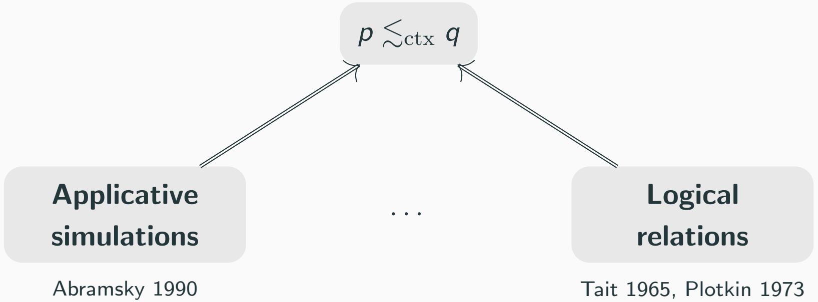
Tait 1965, Plotkin 1973

Coinductive Proof Techniques for Contextual Preorder



- 🙂 Powerful and robust, applicable to a wide variety of languages.
- ☹ Ad hoc - every language needs its own definitions and soundness result!
- ☹ Soundness proofs long, error-prone, boiler-plate!

Coinductive Proof Techniques for Contextual Preorder

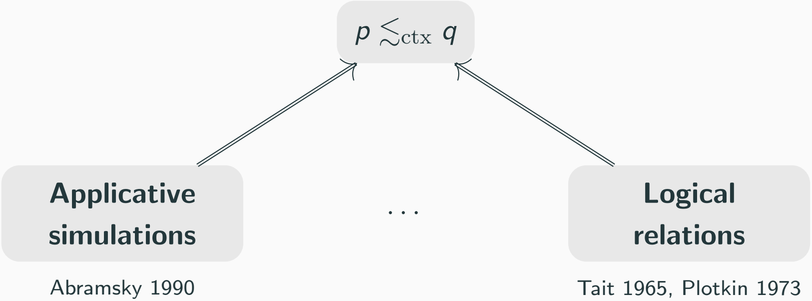


- ☺ Powerful and robust, applicable to a wide variety of languages.
- ☹ Ad hoc - every language needs its own definitions and soundness result!
- ☹ Soundness proofs long, error-prone, boiler-plate!

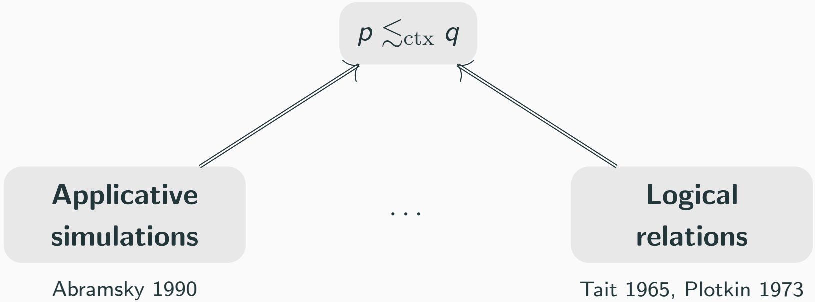
This Talk

A language-independent approach based on **(higher-order) coalgebras**.

Towards an Abstract Theory of Contextual Preorder



Towards an Abstract Theory of Contextual Preorder



Prerequisite for a language-independent approach: an abstract notion of

"higher-order language" and **"operational semantics"**.

This is provided by

Higher-Order Abstract GSOS (extending Turi & Plotkin '97).

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Categorical Abstraction

Syntax

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

Operational rules

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

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

Operational model

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

λ-terms

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Categorical Abstraction

Syntax

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

Operational rules

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

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

Operational model

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

λ-terms

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Categorical Abstraction

Syntax

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

Operational rules

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

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

Operational model

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

λ-terms

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Categorical Abstraction

Syntax

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

Operational rules

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

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

Operational model

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

λ-terms

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

λ-terms

Categorical Abstraction

Syntax [Fiore, Turi & Plotkin '99]

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

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

untyped variable contexts

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

Key observation

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

$$\Sigma X = V + X \times X + \delta X.$$

$V(n) = n$ $\qquad$ $\delta X(n) = X(n+1)$

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational 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]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Behaviour

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

is a **higher-order coalgebra**

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

for the **behaviour bifunctor**

$$B\colon (\mathbf{Set}^{\mathbb{F}})^{\mathsf{op}} \times \mathbf{Set}^{\mathbb{F}} \to \mathbf{Set}^{\mathbb{F}}, \qquad B(X, Y) = Y + Y^X.$$

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

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]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

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]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Higher-Order GSOS law

$$\Sigma(X \times B(X, Y)) \xrightarrow{\varrho_{X,Y}} B(X, \Sigma^*(X + Y))$$

$\varrho_{X,Y}$: 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]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Higher-Order GSOS law

$$\Sigma(X \times B(X, Y)) \xrightarrow{\varrho_{X,Y}} B(X, \Sigma^*(X + Y))$$

dinat. in X, nat. in Y

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

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

Higher-Order Abstract GSOS [POPL'23]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Higher-Order GSOS law

$$\Sigma(X \times B(X, Y)) \xrightarrow{\varrho_{X,Y}} B(X, \Sigma^*(X + Y))$$

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]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Higher-Order GSOS law

$$\Sigma(X \times B(X,Y)) \xrightarrow{\varrho_{X,Y}} B(X, \Sigma^*(X+Y))$$

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]

Example: Untyped CBN λ-calculus

Syntax

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

Operational rules

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

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

Operational model

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

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Higher-Order GSOS law

$$\Sigma(X \times B(X,Y)) \xrightarrow{\varrho_{X,Y}} B(X, \Sigma^*(X+Y))$$

$\varrho_{X,Y}$: dinat. in X, nat. in Y

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

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

Higher-Order Abstract GSOS [POPL'23]

Instances:

- Untyped and typed λ-calculi
- Effectful λ-calculi

  e.g. nondeterministic, probabilistic
- Evaluation: CBN, CBV, ...

Categorical Abstraction

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

Initial algebra $\mu\Sigma$

Higher-Order GSOS law

$$\Sigma(X \times B(X,Y)) \xrightarrow{\varrho_{X,Y}} B(X, \Sigma^*(X+Y))$$

$\varrho_{X,Y}$ ← dinat. in X, nat. in Y

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

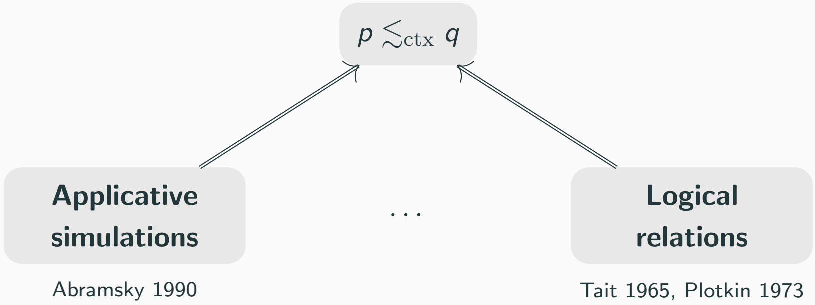
$$\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$$

Abstract Modelling of Operational Semantics

Concrete/Abstract

Concrete	Abstract
1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial algebra $\mu\Sigma$
3. Behaviour type	3. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
4. Operational model	4. $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$
5. Operational rules	5. Higher-order GSOS law

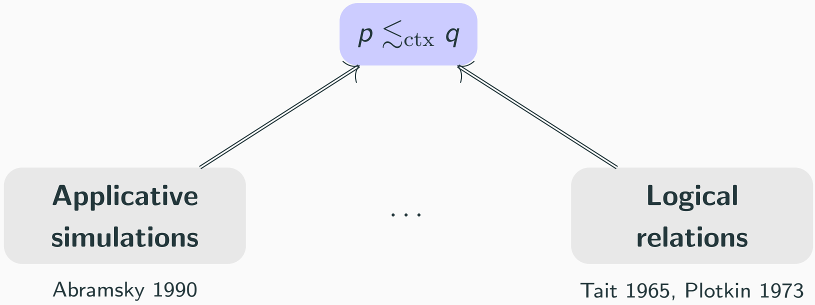
Towards an Abstract Theory of Contextual Preorder



Goal: A language-independent theory of operational techniques based on

Higher-Order Abstract GSOS.

Towards an Abstract Theory of Contextual Preorder



Goal: A language-independent theory of operational techniques based on

Higher-Order Abstract GSOS.

Contextual Preorder

$$p \lesssim_{\mathrm{ctx}} q \quad \text{iff} \quad \forall C[\cdot]\colon\ C[p] \text{ terminates} \implies C[q] \text{ terminates.}$$

A relation R on programs is

- **adequate** if it preserves termination:

$$R(p, q) \quad \text{implies} \quad (p \text{ terminates} \implies q \text{ terminates}).$$

- a **congruence** if it is respected by all language operations.

$$\ldots\text{e.g.} \quad R(p, q) \quad \text{implies} \quad R(\lambda x.\, p, \lambda x.\, q).$$

Observation

The contextual preorder $\lesssim_{\mathrm{ctx}}$ is the greatest adequate congruence.

Contextual Preorder for Functor Algebras

Recall: A **congruence** on an algebra $\Sigma A \xrightarrow{a} A$ is a subalgebra $R \rightarrowtail A \times A$.

Definition (Abstract Contextual Preorder)

Given a preorder $O \rightarrowtail \mu\Sigma \times \mu\Sigma$ of **observations**, let

$$\lesssim^{O}_{\mathrm{ctx}} \;\rightarrowtail\; \mu\Sigma \times \mu\Sigma$$

be the greatest congruence on $\mu\Sigma$ contained in O.

exists if $\mathbb{C}$ cocomplete + well-powered + extensive and Σ finitary

Example (λ-calculus)

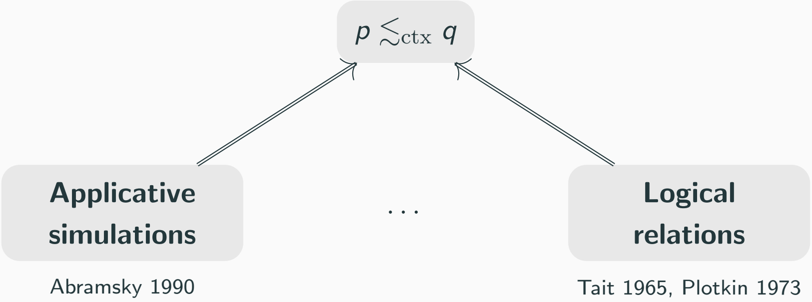
$\lesssim^{O}_{\mathrm{ctx}} = \lesssim_{\mathrm{ctx}}$ for $O = \{\, (p,q) \mid p \text{ terminates } \implies q \text{ terminates} \,\}$.

Abstract Modelling of Operational Semantics

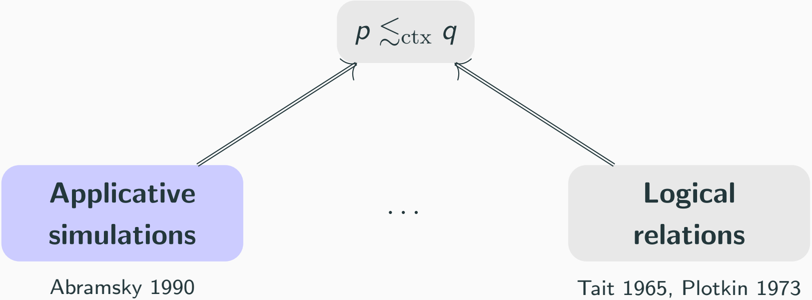
Concrete/Abstract

Concrete	Abstract
1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial algebra $\mu\Sigma$
3. Behaviour type	3. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
4. Operational model	4. $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$
5. Operational rules	5. Higher-order GSOS law
6. Contextual preorder	6. $\lesssim^{O}_{\mathrm{ctx}}$ $(O \rightarrowtail \mu\Sigma \times \mu\Sigma)$

Towards an Abstract Theory of Contextual Preorder



Towards an Abstract Theory of Contextual Preorder



Applicative Simulations (Untyped CBN λ-Calculus)

An **applicative simulation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

$$p \to p' \implies \exists q'.\, q \to^{\star} q' \wedge R(p', q')$$
$$p = \lambda x.p' \implies \exists q'.\, q \to^{\star} \lambda x.q' \wedge \forall e \in \Lambda.\ R(p'[e/x], q'[e/x]).$$

("Related functions send the same input to related outputs")

Equivalently: R is a weak simulation on the LTS $\gamma \colon \Lambda \to \Lambda + \Lambda^{\Lambda}$.

Applicative Simulations (Untyped CBN λ-Calculus)

An **applicative simulation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p,q)$,

$$p \to p' \implies \exists q'.\, q \to^\star q' \;\wedge\; R(p',q')$$
$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall e \in \Lambda.\; R(p'[e/x], q'[e/x]).$$

("Related functions send the same input to related outputs")

Equivalently: R is a weak simulation on the LTS $\gamma\colon \Lambda \to \Lambda + \Lambda^\Lambda$.

Soundness Theorem

Applicative similarity $\lesssim_{\mathrm{app}}$ is an adequate congruence. Hence

$$p \lesssim_{\mathrm{app}} q \qquad \text{implies} \qquad p \lesssim_{\mathrm{ctx}} q.$$

Proof: Difficult (Howe's method).

Applicative Simulations (Untyped CBN λ-Calculus)

An **applicative simulation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

$$p \to p' \implies \exists q'.\, q \to^\star q' \;\wedge\; R(p', q')$$
$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall e \in \Lambda.\ R(p'[e/x], q'[e/x]).$$

("Related functions send the same input to related outputs")

Equivalently: R is a weak simulation on the LTS $\gamma \colon \Lambda \to \Lambda + \Lambda^\Lambda$.

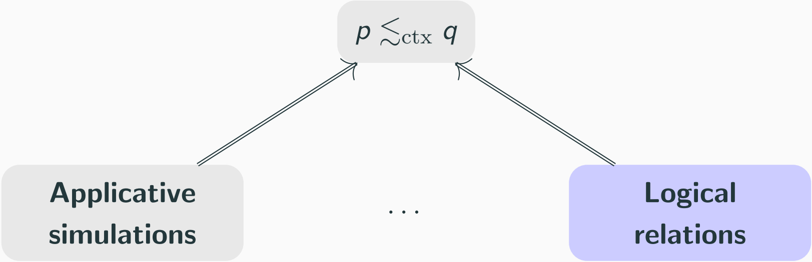
Soundness Theorem

Applicative similarity $\lesssim_{\mathrm{app}}$ is an adequate congruence. Hence

$$p \lesssim_{\mathrm{app}} q \qquad \text{implies} \qquad p \lesssim_{\mathrm{ctx}} q.$$

Proof: Difficult (Howe's method). **Generalization to HO GSOS?**

Towards an Abstract Theory of Contextual Preorder



Abramsky 1990

Tait 1965, Plotkin 1973

Logical Relations (Untyped CBN λ-Calculus)

A **logical relation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

$$p \to p' \implies \exists q'.\, q \to^\star q' \;\wedge\; R(p', q')$$

$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall R(d, e).\; R(p'[d/x], q'[e/x]).$$

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

Logical Relations (Untyped CBN λ-Calculus)

A **logical relation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

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

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

Construction: Step-indexed logical relation $\mathcal{L} = \bigcap_n \mathcal{L}_n$

$\mathcal{L}_0 = \Lambda \times \Lambda$ and $\mathcal{L}_{n+1}(p, q)$ iff

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

Logical Relations (Untyped CBN λ-Calculus)

A **logical relation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

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

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

Construction: Step-indexed logical relation $\mathcal{L} = \bigcap_n \mathcal{L}_n$

$\mathcal{L}_0 = \Lambda \times \Lambda$ and $\mathcal{L}_{n+1}(p, q)$ iff

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

Soundness Theorem

$\mathcal{L}$ is a logical relation, and an adequate congruence.

Proof: Easier than for $\lesssim_{\text{app}}$, but tedious.

Logical Relations (Untyped CBN λ-Calculus)

A **logical relation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

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

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

Construction: **Step-indexed logical relation** $\mathcal{L} = \bigcap_n \mathcal{L}_n$

$\mathcal{L}_0 = \Lambda \times \Lambda$ and $\mathcal{L}_{n+1}(p, q)$ iff

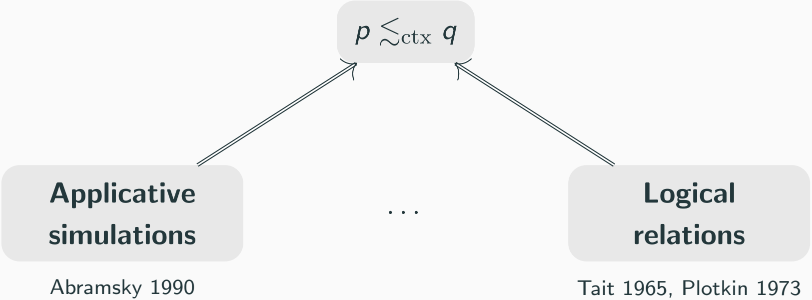
$$p \to p' \implies \exists q'.\, q \to^\star q' \,\wedge\, \mathcal{L}_n(p', q')$$
$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \,\wedge\, \forall \mathcal{L}_n(d, e).\, \mathcal{L}_n(p'[d/x], q'[e/x]).$$

Soundness Theorem

$\mathcal{L}$ is a logical relation, and an adequate congruence.

Proof: Easier than for $\lesssim_{\mathrm{app}}$, but tedious. **Generalization to HO GSOS?**

Towards an Abstract Theory of Contextual Preorder



Now: Categorical abstraction via **relation liftings**.

Relation Liftings

$\mathbf{Rel}(\mathbb{C})$: 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

$$\begin{array}{ccc} \mathbf{Rel}(\mathbb{C})^{\mathsf{op}} \times \mathbf{Rel}(\mathbb{C}) & \xrightarrow{\overline{B}} & \mathbf{Rel}(\mathbb{C}) \\ \downarrow & & \downarrow \\ \mathbb{C}^{\mathsf{op}} \times \mathbb{C} & \xrightarrow{B} & \mathbb{C} \end{array}$$

Relation Liftings

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

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

$$\begin{array}{ccc}
\mathbf{Rel}(\mathbb{C})^{\mathrm{op}} \times \mathbf{Rel}(\mathbb{C}) & \xrightarrow{\overline{B}} & \mathbf{Rel}(\mathbb{C}) \\
\downarrow & & \downarrow \\
\mathbb{C}^{\mathrm{op}} \times \mathbb{C} & \xrightarrow{\quad B \quad} & \mathbb{C}
\end{array}$$

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 (Untyped CBN λ-Calculus)

A **logical relation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

$$p \to p' \implies \exists q'.\, q \to^\star q' \;\wedge\; R(p', q')$$

$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall R(d, e).\; R(p'[d/x], q'[e/x]).$$

Equivalently: $R \leq (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(R, R)]$

"$\to$" "$\to^\star$"

Logical Relations (Untyped CBN λ-Calculus)

A **logical relation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

$$p \to p' \implies \exists q'.\, q \to^\star q' \;\wedge\; R(p', q')$$
$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall R(d, e).\; R(p'[d/x], q'[e/x]).$$

Equivalently: $R \leq (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(R, R)]$

"$\to$" "$\to^\star$"

Construction: Step-indexed logical relation $\mathcal{L} = \bigcap_n \mathcal{L}_n$

$\mathcal{L}_0 = \Lambda \times \Lambda$ and $\mathcal{L}_{n+1}(p, q)$ iff

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

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

Applicative Simulations (Untyped CBN λ-Calculus)

An **applicative simulation** $R \rightarrowtail \Lambda \times \Lambda$ satisfies, for $R(p, q)$,

$$p \to p' \implies \exists q'.\, q \to^\star q' \;\wedge\; R(p', q')$$

$$p = \lambda x.p' \implies \exists q'.\, q \to^\star \lambda x.q' \;\wedge\; \forall e \in \Lambda.\; R(p'[e/x], q'[e/x]).$$

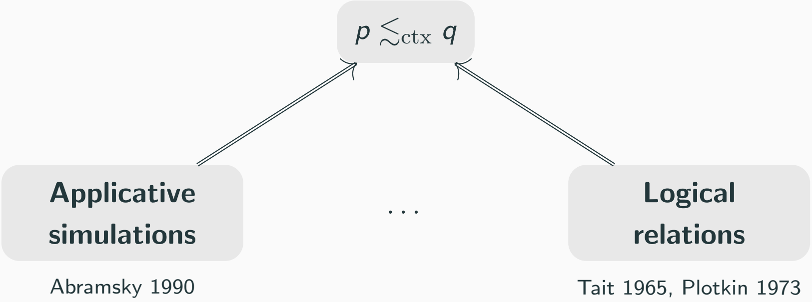
Equivalently: $R \leq (\gamma \times \widetilde{\gamma})^{-1}[\overline{B}(\Delta, R)]$.

Abstract Modelling of Operational Semantics

Concrete/Abstract

Concrete	Abstract
1. Syntax	1. $\Sigma\colon \mathbb{C} \to \mathbb{C}$
2. Program terms	2. Initial algebra $\mu\Sigma$
3. Behaviour type	3. $B\colon \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathbb{C}$
4. Operational model	4. $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$
5. Operational rules	5. Higher-order GSOS law
6. Contextual preorder	6. $\lesssim^{O}_{\mathrm{ctx}}$ $(O \rightarrowtail \mu\Sigma \times \mu\Sigma)$
7. Applicative simulation	7. $R \le (\gamma \times \widetilde{\gamma})^{-1}\overline{B}(\Delta, R)$
8. Logical relation	8. $R \le (\gamma \times \widetilde{\gamma})^{-1}\overline{B}(R, R)$
9. Step-indexed log. relation	9. $\mathcal{L}_{n+1} = (\gamma \times \widetilde{\gamma})^{-1}\overline{B}(\mathcal{L}_n, \mathcal{L}_n)$

Towards an Abstract Theory of Contextual Preorder



Now: An abstract **congruence result**!

Main Result

Congruence Theorem for HO Abstract GSOS [LICS '23, '24]

- Applicative similarity on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.
- The logical relation $\mathcal{L}$ on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.

if

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

Main Result

Congruence Theorem for HO Abstract GSOS [LICS '23, '24]

- Applicative similarity on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.
- The logical relation $\mathcal{L}$ on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.

if

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

$$
\begin{array}{ccccc}
\Sigma(\mu\Sigma) & \xrightarrow{\quad\cong\quad} & \mu\Sigma & \xrightarrow{\quad\widetilde{\gamma}\quad} & B(\mu\Sigma, \mu\Sigma) \\
\Big\downarrow{\scriptstyle\langle \mathrm{id}, \widetilde{\gamma}\rangle} & & \rotatebox{-90}{$\succeq$} & & \Big\uparrow{\scriptstyle B(\mathrm{id}, \hat{\iota})} \\
\Sigma(\mu\Sigma \times B(\mu\Sigma, \mu\Sigma)) & \xrightarrow{\varrho_{\mu\Sigma,\mu\Sigma}} & B(\mu\Sigma, \Sigma^*(\mu\Sigma + \mu\Sigma)) & \xrightarrow{B(\mathrm{id}, \Sigma^* \nabla)} & B(\mu\Sigma, \Sigma^* \mu\Sigma)
\end{array}
$$

cf. Bonchi, Petrişan, Pous, Rot '15

Main Result

Congruence Theorem for HO Abstract GSOS [LICS '23, '24]

- Applicative similarity on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.
- The logical relation $\mathcal{L}$ on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.

if

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

rules remain sound for weak transitions

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

Main Result

Congruence Theorem for HO Abstract GSOS [LICS '23, '24]

- Applicative similarity on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.
- The logical relation $\mathcal{L}$ on $\gamma\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$ is a congruence.

if

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

Take-home message:

- The two most popular higher-order operational techniques work for the same abstract reason.
- The lax-bialgebra condition isolates the language-specific core of their congruence properties – and is usually easy to check.

Conclusion and Perspectives