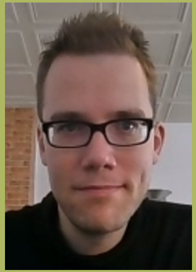
CMCS 2026

Turin

Locally Final Coalgebras in Denotational Semantics



Sergey Goncharov
UNIVERSITY OF BIRMINGHAM

Marco Peressotti
SDU

Stelios Tsampas
SDU

Henning Urbat
FAU

Stefano Volpe
SDU

SDU

April 2026

Contents

Denotational semantics for **higher-order** programming languages are hard and ad-hoc.

Locally final coalgebras make them easily derivable from operational semantics!

Contents

Denotational semantics for **higher-order** programming languages are hard and ad-hoc.

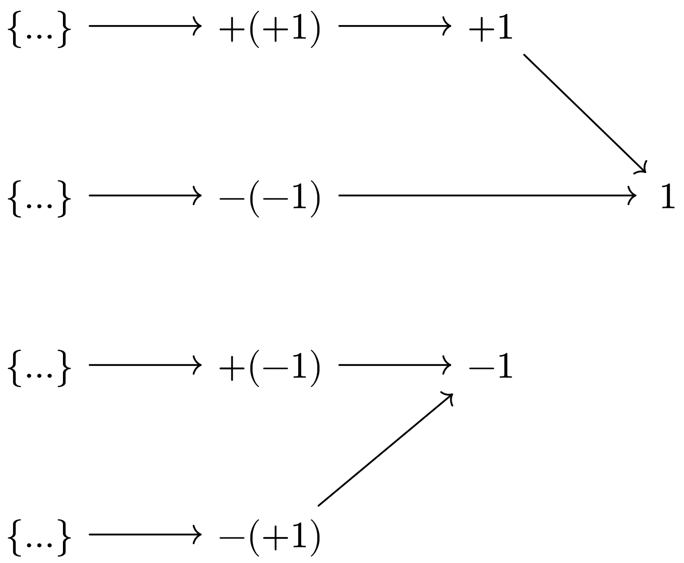
Locally final coalgebras make them easily derivable from operational semantics!

Denotational Semantics

Two Traditions

Operational Semantics

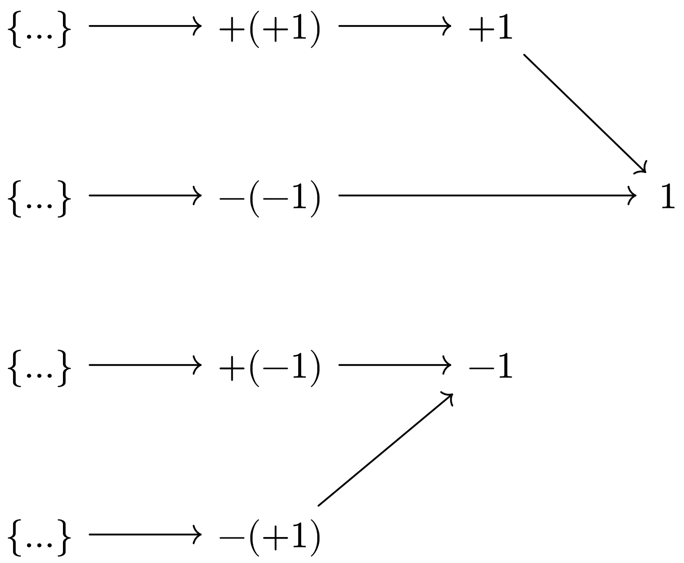
(Some flavour of) transition system! The set of nodes is $\mathcal{L}$, the set of all programs.



Two Traditions

Operational Semantics

(Some flavour of) transition system! The set of nodes is $\mathcal{L}$, the set of all programs.



Denotational Semantics

The way a logician would do it (compositionally, hopefully). A recursive definition inducing a map

$$\begin{aligned} [\![-]\!] &: \mathcal{L} \to \mathbb{B} \\ [\![1]\!] &:= \mathrm{true} \\ [\![+n]\!] &:= [\![n]\!] \\ [\![-n]\!] &:= \neg [\![n]\!]. \end{aligned}$$

(Not) Reusing Theory in PL

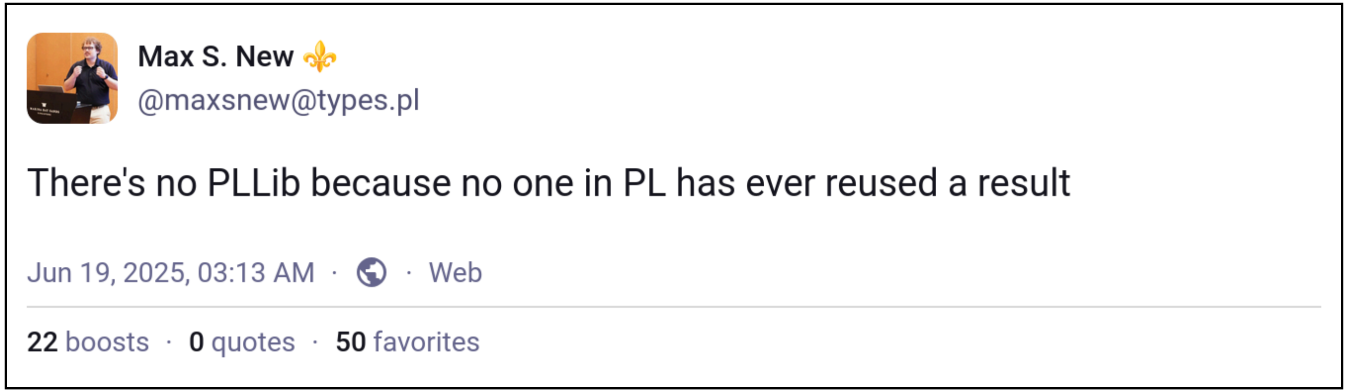


Figure 1: [1] [Online]. Available: https://types.pl/@maxsnew/114707374331592652

(Not) Reusing Theory in PL

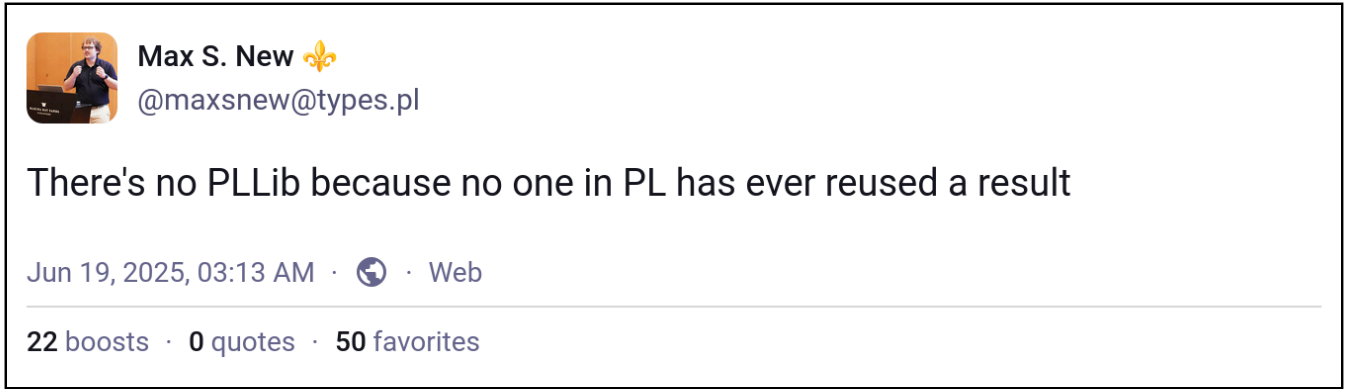


Figure 1: [1] [Online]. Available: https://types.pl/@maxsnew/114707374331592652

Reuse is one of the benefits of **abstraction**.

~~Operational~~ Coalgebraic and ~~Denotational~~ Algebraic Semantics

Operational Semantics

as a **coalgebra** over a behaviour endofunctor $B : \mathscr{C} \to \mathscr{C}$
whose carrier is $\mathscr{L}$, our language object:

$$\gamma : \mathscr{L} \to B\mathscr{L}$$

~~Operational~~ Coalgebraic and ~~Denotational~~ Algebraic Semantics

Operational Semantics

as a **coalgebra** over a behaviour endofunctor $B : \mathcal{C} \to \mathcal{C}$
whose carrier is $\mathcal{L}$, our language object:

$$\gamma : \mathcal{L} \to B\mathcal{L}$$

Also, $\mathcal{L} := \mu\Sigma$ carries an **initial algebra** structure ι.

$$\Sigma(\mu\Sigma) \overset{\iota\cong}{\to} \mu\Sigma \overset{\gamma}{\to} B(\mu\Sigma)$$

~~Operational~~ Coalgebraic and ~~Denotational~~ Algebraic Semantics

Operational Semantics

as a **coalgebra** over a behaviour endofunctor $B : \mathcal{C} \to \mathcal{C}$ whose carrier is $\mathcal{L}$, our language object:

$$\gamma : \mathcal{L} \to B\mathcal{L}$$

Also, $\mathcal{L} := \mu\Sigma$ carries an **initial algebra** structure ι.

$$\Sigma(\mu\Sigma) \overset{\iota\cong}{\to} \mu\Sigma \overset{\gamma}{\to} B(\mu\Sigma)$$

Denotational Semantics

as an **algebra** over a syntax endofunctor $\Sigma : \mathcal{C} \to \mathcal{C}$ whose carrier is Z, the chosen denotational domain:

$$a : \Sigma Z \to Z$$

~~Operational~~ Coalgebraic and ~~Denotational~~ Algebraic Semantics

Operational Semantics

as a **coalgebra** over a behaviour endofunctor $B : \mathcal{C} \to \mathcal{C}$ whose carrier is $\mathcal{L}$, our language object:

$$\gamma : \mathcal{L} \to B\mathcal{L}$$

Also, $\mathcal{L} := \mu\Sigma$ carries an **initial algebra** structure ι.

$$\Sigma(\mu\Sigma) \overset{\iota\cong}{\to} \mu\Sigma \overset{\gamma}{\to} B(\mu\Sigma)$$

Denotational Semantics

as an **algebra** over a syntax endofunctor $\Sigma : \mathcal{C} \to \mathcal{C}$ whose carrier is Z, the chosen denotational domain:

$$a : \Sigma Z \to Z$$

Also, if we coinductively defined $Z := \nu B$, it carries a **final coalgebra** structure ζ.

$$\Sigma(\nu B) \overset{a}{\to} \nu B \overset{\zeta\cong}{\to} B(\nu B)$$

What's the pattern?

Abstract GSOS (Grand Structured Op. Sem.)

Assume $\mathcal{C}$ has binary products, Σ has an algebraically free monad $\Sigma^\star$, and that we encoded the derivation system for the operational model as a natural transformation

$$\begin{array}{cc} \dfrac{\cdots \quad \cdots}{\cdots} & \dfrac{\cdots}{\cdots} \\[1em] \dfrac{\cdots \quad \cdots \quad \cdots}{\cdots} & \dfrac{\cdots \quad \cdots}{\cdots} \end{array} \quad \rightsquigarrow \quad (\rho_X : \Sigma(X \times BX) \to B\Sigma^\star X)_{X \in \mathrm{Ob}(\mathcal{C})}.$$

Abstract GSOS (Grand Structured Op. Sem.)

Assume $\mathcal{C}$ has binary products, Σ has an algebraically free monad $\Sigma^\star$, and that we encoded the derivation system for the operational model as a natural transformation

$$\begin{array}{cc} \dfrac{\cdots \quad \cdots}{\cdots} & \dfrac{\cdots}{\cdots} \\[2ex] \dfrac{\cdots \quad \cdots \quad \cdots}{\cdots} & \dfrac{\cdots \quad \cdots}{\cdots} \end{array} \quad \rightsquigarrow \quad (\rho_X : \Sigma(X \times BX) \to B\Sigma^\star X)_{X \in \mathrm{Ob}(\mathcal{C})}.$$

Then, ρ-*bialgebras* are pairs (g, h) of a Σ-coalgebra g and a B-coalgebra h with a shared carrier making

$$\begin{array}{ccccc} \Sigma X & \xrightarrow{\quad g \quad} & X & \xrightarrow{\quad h \quad} & BX \\ \Big\downarrow{\scriptstyle \Sigma\langle \mathrm{id}_X, h\rangle} & & & & \Big\downarrow{\scriptstyle B\hat{g}} \\ \Sigma(X \times BX) & & \xrightarrow{\qquad \rho_X \qquad} & & B\Sigma^\star X \end{array}$$

commute. We take these as models: algebra-coalgebra pairs respecting the derivation system.

Abstract GSOS (Grand Structured Op. Sem.) (2)

ρ-bialgebras form a category, where morphisms are maps which are algebra and coalgebra morphisms at the same time.

Abstract GSOS (Grand Structured Op. Sem.) (2)

ρ-bialgebras form a category, where morphisms are maps which are algebra and coalgebra morphisms at the same time.

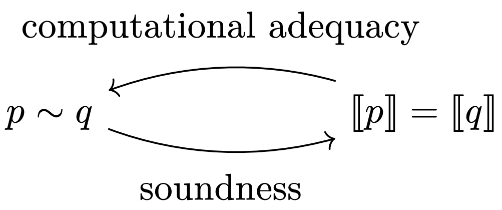
D. Turi and G. Plotkin [2] proved the existence of an **initial** and of a **final object**, and that they are the **operational** and **denotational models** respectively.

Abstract GSOS (Grand Structured Op. Sem.) (2)

ρ-bialgebras form a category, where morphisms are maps which are algebra and coalgebra morphisms at the same time.

D. Turi and G. Plotkin [2] proved the existence of an **initial** and of a **final object**, and that they are the **operational** and **denotational models** respectively.

Both are compositional. Computational adequacy holds. If B preserves weak pullbacks, so does soundness.



$$\text{full abstraction} = \text{soundness} + \text{computational adequacy}$$

Higher-order Abstract GSOS… (1)

Higher-order behaviour requires a mixed-variance behaviour functor, $B : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$ (e.g., $B(X, Y) = Y + Y^X$ in a category with binary products and exponentials).

Higher-order Abstract GSOS... (1)

Higher-order behaviour requires a mixed-variance behaviour functor, $B : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$ (e.g., $B(X, Y) = Y + Y^X$ in a category with binary products and exponentials).

Assume $\mathcal{C}$ has binary products, Σ has an initial algebra $\mu\Sigma \to \Sigma(\mu\Sigma)$, as well of an algebraically free monad $\Sigma^\star$, and that we have

$$\left(\rho_{\boldsymbol{X},Y} : \Sigma(X \times B(\boldsymbol{X}, Y)) \to B(\boldsymbol{X}, \Sigma^\star(\boldsymbol{X} + Y))\right)_{\boldsymbol{X},Y \in \mathrm{ob}(C)}$$

dinatural in X and natural in Y.

Higher-order Abstract GSOS... (1)

Higher-order behaviour requires a mixed-variance behaviour functor, $B : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathscr{C}$ (e.g., $B(X, Y) = Y + Y^X$ in a category with binary products and exponentials).

Assume $\mathscr{C}$ has binary products, Σ has an initial algebra $\mu\Sigma \to \Sigma(\mu\Sigma)$, as well of an algebraically free monad $\Sigma^\star$, and that we have

$$\left(\rho_{\mathbf{X},Y} : \Sigma(X \times B(\mathbf{X}, Y)) \to B(\mathbf{X}, \Sigma^\star(\mathbf{X} + Y))\right)_{\mathbf{X},Y \in \mathrm{ob}(C)}$$

dinatural in X and natural in Y. Now, ρ-bialgebras are easily adapted [3], while a morphism is now a map $f : X \to Y$ s.t.

$$\begin{array}{ccccccc}
\Sigma X & \xrightarrow{g_X} & X & \xrightarrow{h_X} & B(X,X) & & \\
 & & & & & \searrow^{B(X,f)} & \\
\downarrow{\scriptstyle \Sigma f} & & \downarrow{\scriptstyle f} & & & & B(X,Y) \\
 & & & & & \nearrow_{B(f,X)} & \\
\Sigma Y & \xrightarrow{g_Y} & Y & \xrightarrow{h_Y} & B(Y,Y) & &
\end{array}$$

commutes.

Higher-order Abstract GSOS… (2)

Operational…

S. Goncharov, S. Milius, L. Schröder, S. Tsampas, and H. Urbat [3] proved the existence of an **initial ρ-bialgebra**, and that it is a **compositional operational model**.

Higher-order Abstract GSOS… (2)

Operational…

S. Goncharov, S. Milius, L. Schröder, S. Tsampas, and H. Urbat [3] proved the existence of an **initial ρ-bialgebra**, and that it is a **compositional operational model**.

…But Not Denotational (yet)

A final (diagonal) higher-order coalgebra does not always exist, nor it always extends to a final ρ-bialgebra…

…Now with Denotational Semantics!

All ρ-bialgebras seen as denotational semantics are such that their algebra $\Sigma A \stackrel{a}{\to} A$ is:

…Now with Denotational Semantics!

All ρ-bialgebras seen as denotational semantics are such that their algebra $\Sigma A \xrightarrow{a} A$ is:
$\rightarrow$ **compositional** by construction (denotation map defined as a map from initial model);

…Now with Denotational Semantics!

All ρ-bialgebras seen as denotational semantics are such that their algebra $\Sigma A \xrightarrow{a} A$ is:

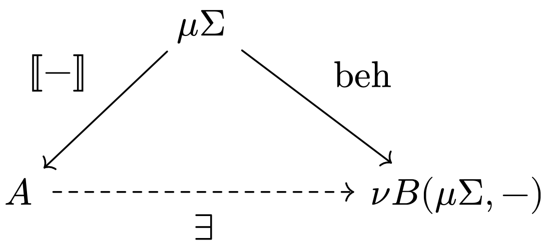
→ **compositional** by construction (denotation map defined as a map from initial model);

→ **computationally adequate** (behavioural equivalence being strong applicative bisimilarity),

…Now with Denotational Semantics!

All ρ-bialgebras seen as denotational semantics are such that their algebra $\Sigma A \xrightarrow{a} A$ is:

- **compositional** by construction (denotation map defined as a map from initial model);
- **computationally adequate** (behavioural equivalence being strong applicative bisimilarity), i.e.

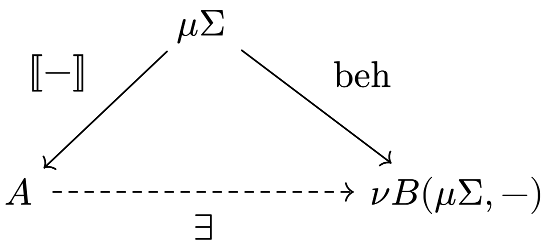


  commutes, where
  - $[\![-]\!]$ is the map from the initial Σ-algebra to a;
  - beh is the map from the coalgebra of our initial model to the final $B(\mu\Sigma, -)$-coalgebra.

…Now with Denotational Semantics!

All ρ-bialgebras seen as denotational semantics are such that their algebra $\Sigma A \xrightarrow{a} A$ is:

- → **compositional** by construction (denotation map defined as a map from initial model);
- → **computationally adequate** (behavioural equivalence being strong applicative bisimilarity), i.e.



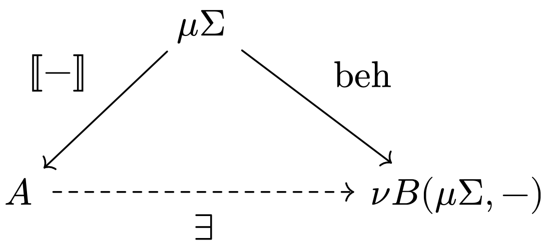
  commutes, where
  - → $[\![-]\!]$ is the map from the initial Σ-algebra to a;
  - → beh is the map from the coalgebra of our initial model to the final $B(\mu\Sigma, -)$-coalgebra.

A map $Z \to B(Z, Z)$ that is a final $B(Z, -)$-coalgebra is called a *locally final coalgebra*.

…Now with Denotational Semantics!

All ρ-bialgebras seen as denotational semantics are such that their algebra $\Sigma A \xrightarrow{a} A$ is:

- **compositional** by construction (denotation map defined as a map from initial model);
- **computationally adequate** (behavioural equivalence being strong applicative bisimilarity), i.e.



commutes, where

- $[\![-]\!]$ is the map from the initial Σ-algebra to a;
- beh is the map from the coalgebra of our initial model to the final $B(\mu\Sigma, -)$-coalgebra.

A map $Z \to B(Z, Z)$ that is a final $B(Z, -)$-coalgebra is called a *locally final coalgebra*. If ρ is relatively flat, every locally final coalgebra extends to a ρ-bialgebra! How to find a locally final coalebra?

Contents

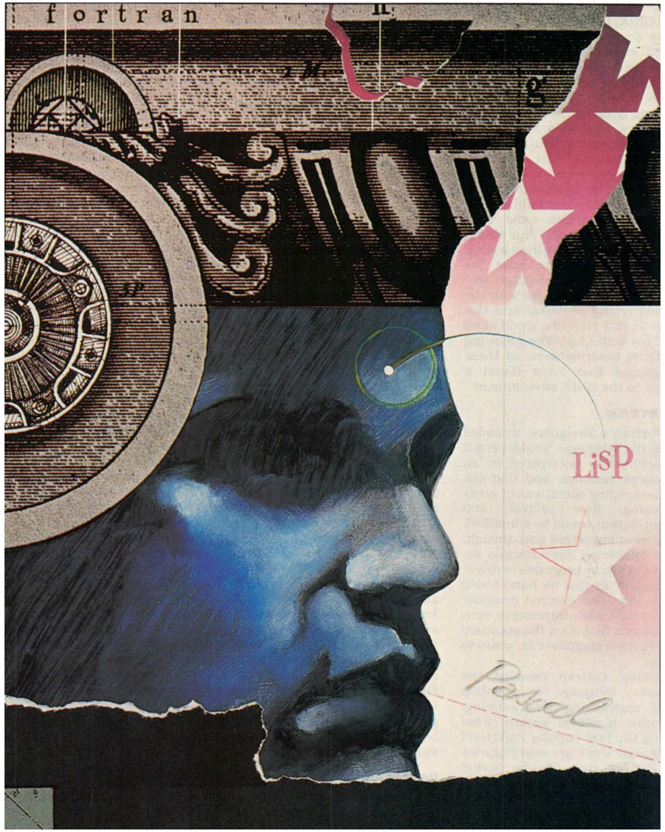


Figure 3: [4] Unknown artist, *UNIX REVIEW cover*. 1985.

Locally Final Coalgebras

A Metric Setting for Fixpoints

The Setting

By Lambek, locally final coalgebras are particular fixpoints of B seen as a recursive equation[1]. Can we adapt previous fixpoint theory?

[1]In $\mathrm{CPO}_\perp$, we would say “domain equation”.

A Metric Setting for Fixpoints

The Setting

By Lambek, locally final coalgebras are particular fixpoints of B seen as a recursive equation[1]. Can we adapt previous fixpoint theory?

Assume $\mathcal{C}$ to be an M-category, i.e. a category enriched over complete, bounded metric spaces satisfying the ultrametric equality:

$$d(x,z) \leq \max(d(x,y), d(y,z)).$$

[1] In $\mathrm{CPO}_\perp$, we would say “domain equation”.

A Metric Setting for Fixpoints

The Setting

By Lambek, locally final coalgebras are particular fixpoints of B seen as a recursive equation[1]. Can we adapt previous fixpoint theory?

Assume $\mathcal{C}$ to be an M-category, i.e. a category enriched over complete, bounded metric spaces satisfying the ultrametric equality:

$$d(x, z) \leq \max(d(x, y), d(y, z)).$$

We say that $F : \mathcal{C} \to \mathcal{C}$ is *locally contractive* (*locally non-expansive*) if, for each $X, Y \in \mathrm{ob}(\mathcal{C})$, function $F_{X,Y}$ is contractive with the same contractivity factor (*non-expansive*).

[1] In $\mathrm{CPO}_\perp$, we would say “domain equation”.

A Metric Setting for Fixpoints

The Setting

By Lambek, locally final coalgebras are particular fixpoints of B seen as a recursive equation[1]. Can we adapt previous fixpoint theory?

Assume $\mathcal{C}$ to be an M-category, i.e. a category enriched over complete, bounded metric spaces satisfying the ultrametric equality:

$$d(x, z) \leq \max(d(x, y), d(y, z)).$$

We say that $F : \mathcal{C} \to \mathcal{C}$ is *locally contractive* (*locally non-expansive*) if, for each $X, Y \in \mathrm{ob}(\mathcal{C})$, function $F_{X,Y}$ is contractive with the same contractivity factor (*non-expansive*).

The Fixpoint Theorem

(Slight rephrasing of) L. Birkedal, K. Støvring, and J. Thamsborg [5]:

If $\mathcal{C}$ has a terminal object and inverse limits of increasing Cauchy towers, every locally contractive functor F such that there exists a map $1 \to F1$ has a fixpoint.

[1] In $\mathrm{CPO}_\perp$, we would say "domain equation".

A Metric Setting for Fixpoints

The Setting

By Lambek, locally final coalgebras are particular fixpoints of B seen as a recursive equation[1]. Can we adapt previous fixpoint theory?

Assume $\mathcal{C}$ to be an M-category, i.e. a category enriched over complete, bounded metric spaces satisfying the ultrametric equality:

$$d(x, z) \leq \max(d(x, y), d(y, z)).$$

We say that $F : \mathcal{C} \to \mathcal{C}$ is *locally contractive* (*locally non-expansive*) if, for each $X, Y \in \mathrm{ob}(\mathcal{C})$, function $F_{X,Y}$ is contractive with the same contractivity factor (*non-expansive*).

The Fixpoint Theorem

(Slight rephrasing of) L. Birkedal, K. Støvring, and J. Thamsborg [5]:

If $\mathcal{C}$ has a terminal object and inverse limits of increasing Cauchy towers, every locally contractive functor F such that there exists a map $1 \to F1$ has a fixpoint.

If all hom-sets are inhabited, the fixpoint is unique.

[1] In $\mathrm{CPO}_\perp$, we would say "domain equation".

Iterative Construction

Adámek's Final Sequence

If $\mathcal{C}$ has a terminal object and limits of chains (i.e., α^{op}-chains for all ordinals α), we can start from 1 and repeatedly apply a fixed $H : \mathcal{C} \to \mathcal{C}$ to construct the *final sequence*, a (large) diagram $D : \mathrm{Ord}^{\mathrm{op}} \to \mathcal{C}$.

Iterative Construction

Adámek's Final Sequence

If $\mathcal{C}$ has a terminal object and limits of chains (i.e., α^{op}-chains for all ordinals α), we can start from 1 and repeatedly apply a fixed $H : \mathcal{C} \to \mathcal{C}$ to construct the *final sequence*, a (large) diagram $D : \mathrm{Ord}^{\mathrm{op}} \to \mathcal{C}$.

Sufficient Conditions for Convergence (One is Enough)

→ H preserves limits of α^{op}-chains for some α;

→ $\mathcal{C}$ is locally presentable, H preserves monomorphisms and colimits of α-chains of strong monos for some regular cardinal α.

Iterative Construction

Adámek's Final Sequence

If $\mathcal{C}$ has a terminal object and limits of chains (i.e., α^{op}-chains for all ordinals α), we can start from 1 and repeatedly apply a fixed $H : \mathcal{C} \to \mathcal{C}$ to construct the *final sequence*, a (large) diagram $D : \mathrm{Ord}^{\mathrm{op}} \to \mathcal{C}$.

Sufficient Conditions for Convergence (One is Enough)

→ H preserves limits of α^{op}-chains for some α;

→ $\mathcal{C}$ is locally presentable, H preserves monomorphisms and colimits of α-chains of strong monos for some regular cardinal α.

Stable M-Category

M-category $\mathcal{C}$ is stable if, for every parallel pair $f, g : A \to B$ and jointly monic family $(h_i : B \to C_i)_{i \in I}$,

$$d_{A,B}(f, g) = \sup_{i \in I} d_{A,C_i}(h_i \circ f, h_i \circ g)$$

($\geq$ always holds by non-expansivity of $\circ$).

Existence and Uniqueness of Locally Final Coalgebras

Let $\mathcal{C}$ be a stable M-category with a terminal object and limits of chains. Let $B : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$ be such that:

- B is locally contractive in the first argument, locally non-expansive in the second;
- there is a map $1 \to B(1, 1)$;
- for each $X \in \mathrm{ob}(\mathcal{C})$, the final sequence of $B(X, -)$ converges.

Existence and Uniqueness of Locally Final Coalgebras

Let $\mathcal{C}$ be a stable M-category with a terminal object and limits of chains. Let $B : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$ be such that:

- B is locally contractive in the first argument, locally non-expansive in the second;
- there is a map $1 \to B(1, 1)$;
- for each $X \in \mathrm{ob}(\mathcal{C})$, the final sequence of $B(X, -)$ converges.

Then, B has a locally final coalgebra. This is unique up to isomorphism if all hom-sets of $\mathcal{C}$ are inhabited.

Existence and Uniqueness of Locally Final Coalgebras

Let $\mathcal{C}$ be a stable M-category with a terminal object and limits of chains. Let $B : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$ be such that:

- B is locally contractive in the first argument, locally non-expansive in the second;
- there is a map $1 \to B(1, 1)$;
- for each $X \in \mathrm{ob}(\mathcal{C})$, the final sequence of $B(X, -)$ converges.

Then, B has a locally final coalgebra. This is unique up to isomorphism if all hom-sets of $\mathcal{C}$ are inhabited.

Proof Idea

Convergence for each X yields a contravariant functor

$$F : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$$
$$F(X) := \nu B(X, -).$$

We are looking for a fixpoint of F, so we can use the fixpoint theorem for both existence and uniqueness.

Contents

Figure 4: [6] Cooksey-Talbott Studio, “Beyond PCs: The Unix System Integrates the Office.” 1985.

Conclusions

Summary

Issue

How can we easily derive denotational semantics from operational semantics of higher-order languages without reinventing the wheel?

Summary

Issue

How can we easily derive denotational semantics from operational semantics of higher-order languages without reinventing the wheel?

Contributions

- → A construction of a computationally adequate, compositional **categorical model from any locally final coalgebra** (assuming relative flatness only).
- → A theory of **existence and uniqueness of locally final coalgebras** via (ultra)metric enrichments.

Future Work

Big-Step Operational Semantics

Our ρ can only encode small-step rules (HO-GSOS), so we get strong applicative bisimilarity. What about big-step rules (AHOS)? We could get *weak* applicative bisimilarity.

Future Work

Big-Step Operational Semantics

Our ρ can only encode small-step rules (HO-GSOS), so we get strong applicative bisimilarity. What about big-step rules (AHOS)? We could get *weak* applicative bisimilarity.

Other Enrichment-Based Fixpoint Theories

→ Enrichment over CPOs (domain theory!)

→ Enrichment over sheaves over a fixed, complete Heyting algebra with a well-founded basis (transfinite step-indexing!)

Takeaway

Denotational semantics for **higher-order** programming languages are hard and ad-hoc.

Locally final coalgebras make them easily derivable from operational semantics!

Takeaway

Denotational semantics for **higher-order** programming languages are hard and ad-hoc.

Locally final coalgebras make them easily derivable from operational semantics!

Thank you!

Sergey **Goncharov**

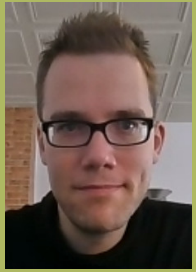
UNIVERSITY OF BIRMINGHAM

Marco **Peressotti**

SDU

Stelios **Tsampas**

SDU



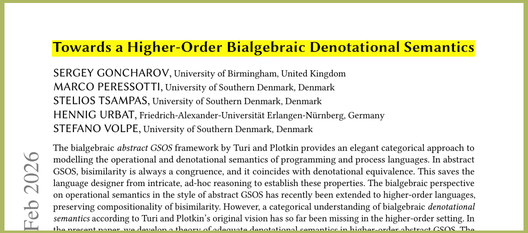
Henning **Urbat**

FAU

Stefano **Volpe**

SDU

arxiv.org/abs/2602.18295



Towards a Higher-Order Bialgebraic Denotational Semantics

SERGEY GONCHAROV, University of Birmingham, United Kingdom
MARCO PERESSOTTI, University of Southern Denmark, Denmark
STELIOS TSAMPAS, University of Southern Denmark, Denmark
HENNIG URBAT, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
STEFANO VOLPE, University of Southern Denmark, Denmark

Feb 2026

The bialgebraic *abstract GSOS* framework by Turi and Plotkin provides an elegant categorical approach to modelling the operational and denotational semantics of programming and process languages. In abstract GSOS, bisimilarity is always a congruence, and it coincides with denotational equivalence. This saves the language designer from intricate, ad-hoc reasoning to establish these properties. The bialgebraic perspective on operational semantics in the style of abstract GSOS has recently been extended to higher-order languages, preserving compositionality of bisimilarity. However, a categorical understanding of bialgebraic *denotational semantics* according to Turi and Plotkin's original vision has so far been missing in the higher-order setting. In the present paper, we develop a theory of adequate denotational semantics in higher-order abstract GSOS. The

Thank you!

References

[1] [Online]. Available: https://types.pl/@maxsnew/114707374331592652

[2] D. Turi and G. Plotkin, “Towards a mathematical operational semantics,” in *Twelfth Annual IEEE Symposium on Logic in Computer Science*, IEEE Comput. Soc, 1997, pp. 280–291. doi: 10.1109/lics.1997.614955.

[3] S. Goncharov, S. Milius, L. Schröder, S. Tsampas, and H. Urbat, “Towards a Higher-Order Mathematical Operational Semantics,” *Proceedings of the ACM on Programming Languages*, pp. 632–658, 2023, doi: 10.1145/3571215.

[4] Unknown artist, *UNIX REVIEW cover*. 1985.

[5] L. Birkedal, K. Støvring, and J. Thamsborg, “The category-theoretic solution of recursive metric-space equations,” *Theoretical Computer Science*, vol. 411, no. 47, pp. 4102–4122, 2010, doi: 10.1016/j.tcs.2010.07.010.

[6] Cooksey-Talbott Studio, “Beyond PCs: The Unix System Integrates the Office.” 1985.

[7] S. Goncharov, M. Peressotti, S. Tsampas, H. Urbat, and S. Volpe, “Towards a Higher-Order Bialgebraic Denotational Semantics,” 2026. [Online]. Available: https://arxiv.org/abs/2602.18295

Relative Flatness

We assume a mild condition on ρ which, in derivation systems, has the following interpretation:

Constructors from signature Σ can be well-ordered so that, if $f \in \Sigma$ is assigned rank j, term $f(...)$ transitions into terms of strictly lower terms, except for their head symbol, which may have rank $\leq j$.

Idea: ban builtin Y combinator and whatnot.

Relating the Two Semantics

Logic

Syntax and semantics:

completeness: $\Gamma \vdash \varphi \leftarrow \Gamma \models \varphi$

soundness: $\Gamma \vdash \varphi \rightarrow \Gamma \models \varphi$

$$\text{adequacy} = \text{soundness} + \text{completeness}$$

Relating the Two Semantics

Logic

Syntax and semantics:

$$\Gamma \vdash \varphi \;\underset{\text{soundness}}{\overset{\text{completeness}}{\leftrightarrows}}\; \Gamma \models \varphi$$

$$\text{adequacy} = \text{soundness} + \text{completeness}$$

Programming Languages (PL)

“Syntactic” semantics (with fixed behavioural equivalence $\sim$) and “traditional” semantics:

$$p \sim q \;\underset{\text{soundness}}{\overset{\text{computational adequacy}}{\leftrightarrows}}\; [\![p]\!] = [\![q]\!]$$

$$\text{full abstraction} = \text{soundness} + \text{computational adequacy}$$