





Higher-Order Mathematical Operational Semantics

Sergey Goncharov  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stefan Milius  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Lutz Schröder  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stelios Tsampas  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Henning Urbat  

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract

We present a higher-order extension of Turi and Plotkin’s abstract GSOS framework that retains the key feature of the latter: for every language whose operational rules are represented by a higher-order GSOS law, strong bisimilarity on the canonical operational model is a congruence with respect to the operations of the language. We further extend this result to weak (bi-)similarity, for which a categorical account of Howe’s method is developed. It encompasses, for instance, Abramsky’s classical compositionality theorem for applicative similarity in the untyped λ -calculus. In addition, we give first steps of a theory of logical relations at the level of higher-order abstract GSOS.

2012 ACM Subject Classification Theory of computation \rightarrow Categorical semantics

Keywords and phrases Abstract GSOS, lambda-calculus, applicative bisimilarity, bialgebra

Digital Object Identifier 10.4230/LIPIcs.CALCO.2023.24

Category Early Ideas

Abstract GSOS. Turi and Plotkin’s *Mathematical Operational Semantics* [6] provides an elegant categorical approach to modelling the operational semantics of process and programming languages, and elucidates when and why such semantics are well-behaved. In this framework the operational rules of a language are represented as a distributive law of a monad over a comonad in a suitable category. An important example is that of *GSOS laws*, viz. natural transformations of the form

$$\rho_X: \Sigma(X \times BX) \rightarrow B\Sigma^*X,$$

for endofunctors $\Sigma, B: \mathcal{C} \rightarrow \mathcal{C}$ determining the *syntax* and *behaviour* of the language at hand and the free (term) monad Σ^* generated by Σ . A GSOS law is thought of representing a set of inductive transition rules that specify how programs are run. For example, the choice of $\mathcal{C} = \mathbf{Set}$ and $BX = (\mathcal{P}_f X)^L$, where \mathcal{P}_f is the finite powerset functor and L a set of transition labels, leads to the well-known GSOS rule format for specifying labelled transition systems. To every GSOS law ρ one can canonically associate an *operational model* given by a coalgebra $\gamma: \mu\Sigma \rightarrow B(\mu\Sigma)$ on the initial algebra $\mu\Sigma$ (the object of programs), and dually a *denotational model* given by an algebra $\alpha: \Sigma(\nu B) \rightarrow \nu B$ on the final coalgebra νB (the object of abstract behaviours). In fact, γ and α respectively extend to an *initial* and *final* ρ -bialgebra. These universal properties entail an important well-behavedness feature: every language modelled by a GSOS law is *compositional*, that is, strong bisimilarity on its



© Sergey Goncharov, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat;
licensed under Creative Commons License CC-BY 4.0

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).

Editors: Paolo Baldan and Valeria de Paiva; Article No. 24; pp. 24:1–24:3



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

operational model is a congruence w.r.t. the operations of the language. Compositionality greatly simplifies reasoning, as it means that the behaviour of a program is determined by the behaviour of its subprograms.

Higher-Order Abstract GSOS. In the past 25 years the abstract GSOS framework has been successfully instantiated to a wealth of first-order languages, even fairly complex ones such as CCS or the π -calculus. However, its application to *higher-order* languages, including the λ -calculus, has been a long-standing open problem. The difficulty lies in the phenomenon that higher-order programs represent both computation and data, which means that the behaviour of a generic set X of programs should be described by a function space like $BX = X^X$. Since X occurs both co- and contravariantly, the map $X \mapsto BX$ is not functorial, hence abstract GSOS does not apply.

In recent work [5] we introduced *higher-order abstract GSOS*, an extension of Turi and Plotkin's original framework designed to overcome this issue. The core idea is to replace the behaviour endofunctor $B: \mathcal{C} \rightarrow \mathcal{C}$ by a behaviour *bifunctor* $B: \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{C}$ of mixed variance, and GSOS laws by *higher-order GSOS laws*, given by families of morphisms

$$\rho_{X,Y}: \Sigma(X \times B(X,Y)) \rightarrow B(X, \Sigma^*(X+Y))$$

natural in $Y \in \mathcal{C}$ and *dinatural* in $X \in \mathcal{C}$. For example, combinatory logics like Curry's SKI-calculus can be modelled by a higher-order GSOS law of a polynomial functor Σ over the behaviour bifunctor $B(X,Y) = Y + Y^X$ on $\mathcal{C} = \mathbf{Set}$. For the untyped λ -calculus, we take the presheaf category $\mathcal{C} = \mathbf{Set}^{\mathbb{F}}$ (where \mathbb{F} is the category of finite sets), a syntax endofunctor Σ whose initial algebra is the presheaf of λ -terms, and a bifunctor B modelling β -reductions and the substitution structure of λ -terms, building on earlier ideas by Fiore et al. [3].

Generalizing the first-order case, every higher-order GSOS law ρ induces an operational model $\gamma: \mu\Sigma \rightarrow B(\mu\Sigma, \mu\Sigma)$, which extends to an initial higher-order ρ -bialgebra. However, in sharp contrast to the first-order case, a final bialgebra usually fails to exist. Nonetheless, higher-order GSOS laws admit a compositional semantics: under mild assumptions on \mathcal{C} , Σ and B , strong bisimilarity on the operational model is a congruence. For example, for the λ -calculus, the operational model extends the transition system on λ -terms given by β -reductions, and strong coalgebraic bisimilarity amounts to strong applicative bisimilarity.

With these foundations at hand, a number of interesting directions open up. In the following we outline a few results, insights, and goals of our ongoing research.

Weak Bisimilarity. While the above compositionality result applies to strong bisimilarity, notions of behavioural equivalence for higher-order languages are typically forms of *weak* bisimilarity where computation steps (e.g. β -reductions) are unobservable and only function applications are deemed relevant. A prime example is Abramsky's *applicative bisimilarity* [1] for the λ -calculus. Proving congruence results for weak bisimilarity is known to be challenging for first-order languages and even more so for higher-order ones, where tailor-made proof techniques such as Howe's method are needed. We have recently established such a result in the generality of higher-order abstract GSOS [7]: weak (bi-)similarity on the operational model of a higher-order GSOS law is a congruence provided that its associated weak model forms a *lax* higher-order bialgebra. This generalizes a corresponding result for first-order abstract GSOS [2]. Our theorem holds in all categories \mathcal{C} where relations are sufficiently well-behaved, e.g. in all (co-)complete, well-powered and locally distributive categories. Its proof is substantially more complex than in the strong case and requires the development of several new techniques of independent interest, including an abstract categorical version

of Howe’s method and the construction of relation liftings of behaviour bifunctors. As an instance of the theorem we recover, e.g., an important property of the λ -calculus originally proved by Abramsky [1]: applicative bisimilarity is a congruence, and hence provides a sound and complete coinductive proof method for contextual equivalence of λ -terms.

A current aim is generalizing the above results from bisimilarity to behavioural distances, e.g. for probabilistic λ -calculi [4], using liftings of bifunctors to quantale-valued relations.

Logical Relations. Besides Howe’s method, another important operational technique for reasoning about higher-order languages is given by *logical relations*; for instance, they yield an efficient proof of strong normalization for the simply typed λ -calculus. The idea is as follows: on the set $\mu\Sigma$ of programs one forms a predicate (or multi-ary relation) P that implies the property of interest, e.g. normalization, and is compatible with function application, that is, if a program computes a function $f: \mu\Sigma \rightarrow \mu\Sigma$, then f respects P . One then shows by structural induction that every program lies in P , whence $P = \mu\Sigma$. In practice logical relations are usually invented ad hoc, but their generic flavour allows for a more systematic approach based on higher-order abstract GSOS. In fact, the “ f respects P ” assertion may be neatly explained using bifunctorial relation liftings, and the generic parts of the structural induction come for free for languages modelled by a higher-order GSOS law. We expect that this approach can greatly reduce the proof obligations for arguments using logical relations.

References

- 1 S. Abramsky. The lazy λ -calculus. In *Research topics in Functional Programming*, pages 65–117. Addison Wesley, 1990.
- 2 Filippo Bonchi, Daniela Petrişan, Damien Pous, and Jurriaan Rot. Lax bialgebras and up-to techniques for weak bisimulations. In *CONCUR’15*, pages 240–253, 2015.
- 3 Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *LICS’99*, pages 193–202. IEEE, 1999.
- 4 Francesco Gavazzo. Quantitative behavioural reasoning for higher-order effectful programs: Applicative distances. In *LICS’18*, pages 452–461. ACM, 2018.
- 5 Sergey Goncharov, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat. Towards a higher-order mathematical operational semantics. In *POPL’23*, volume 7, pages 632–658. ACM, 2023. [arXiv:2210.13387](#).
- 6 Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *LICS’97*, pages 280–291. IEEE, 1997.
- 7 Henning Urbat, Stelios Tsampas, Sergey Goncharov, Stefan Milius, and Lutz Schröder. Weak similarity in higher-order mathematical operational semantics. In *LICS’23*. IEEE, 2023. [arXiv:2302.08200](#).