Structural Congruences for Bialgebraic Semantics

Jurriaan Rot and Marcello Bonsangue

LIACS — Leiden University
{j.c.rot, m.m.bonsangue}@liacs.leidenuniv.nl

Summary. It was observed by Turi and Plotkin that structural operational semantics can be studied at the level of universal coalgebra, providing specification formats for well-behaved operations on many different types of systems. We extend this framework with non-structural assignment rules which can express, for example, the syntactic format for structural congruences proposed by Mousavi and Reniers. Our main result is that the operational model of such an extended specification is well-behaved, in the sense that bisimilarity is a congruence and that bisimulation-up-to techniques are sound.

Background. Structural operational semantics (SOS) is a framework for defining the semantics of programming languages and calculi in terms of transition system specifications. By imposing syntactic restrictions, one can prove well-behavedness properties of transition systems at the meta-level of their specification. For instance, any specification in the GSOS format [1] has a unique operational model, on which bisimilarity is a congruence.

Traditionally, research in SOS has focused on labelled transition systems as the fundamental model of behaviour. Turi and Plotkin [14] introduced the *bialgebraic* approach to structural operational semantics, where in particular GSOS can be studied at the level of *universal coalgebra* [11]. The theory of coalgebras provides a mathematical framework for the uniform study of many types of state-based systems, including labelled transition systems but also, e.g., (non)-deterministic automata, stream systems and various types of probabilistic and weighted automata. In the coalgebraic framework, there is a canonical notion of bisimilarity, which instantiates to the classical definition of (strong) bisimilarity in the case of labelled transition systems. It is shown in [14] that GSOS specifications can be generalised by certain natural transformations, which are called *abstract GSOS specifications*, and that these correspond to the categorical notion of *distributive laws*. This provides enough structure to prove at this general level that bisimilarity is a congruence. By instantiating the theory to concrete instances, one can then obtain congruence formats for systems such as probabilistic automata, weighted transition systems and streams — see [5] for an overview. Another advantage of abstract GSOS is that bisimulation up to context is "compatible" [10, 9], providing a sound enhancement of the bisimulation proof method which can be combined with other compatible enhancements such as bisimulation up to bisimilarity [12, 8].

Adding assignment rules. In this paper we consider non-structural rules such as the following:

$$\frac{!x \mid x \xrightarrow{a} t}{!x \xrightarrow{a} t} \tag{1}$$

The rule in (1) properly defines the replication operator in CCS[1]: intuitively $!x$ represents $x \mid x \mid x \mid \ldots$, i.e., the infinite parallel composition of x with itself. In fact, the above rule

[1] The simpler rule $\frac{x \to x'}{!x \to !x \mid x'}$ is problematic in the presence of the sum operator [8, 13].

can be seen as assigning the behaviour of the term $!x \mid x$ to the simpler term $!x$, therefore we call it an *assignment rule*. Being inherently non-structural, such an assignment rule cannot directly be embedded in the bialgebraic framework of Turi and Plotkin, where the behaviour of terms is computed inductively. In this paper we show how to interpret assignment rules together with abstract GSOS specifications. As it turns out, this requires the assumption that the functor which represents the type of coalgebra is *ordered* as a complete lattice; for example, in the case of labelled transition systems this order is simply inclusion of sets of pairs (a, x) of a label a and a state x. The operational model on closed terms then is the *least* model such that every transition either (1) can be derived from a rule in the specification, or (2) there is a rule assigning to an operator σ the behaviour of a term t in the model. To ensure the existence of such least models, we restrict to *monotone* abstract GSOS specifications, a generalisation of the *positive GSOS format* for transition systems [3]. Positive GSOS can be seen as the greatest common divisor of GSOS and the tyft/tyxt format.

Our main result is that the interpretation of a monotone abstract GSOS specification together with a set of assignment rules is itself the operational model of another (typically larger) abstract GSOS specification. Like the interpretation of a GSOS specification with assignment rules, we construct this latter specification by fixpoint induction. As a direct consequence of this alternative representation of the interpretation, we obtain that bisimilarity is a congruence and that bisimulation up to context is sound and even compatible — properties that do not follow from bisimilarity being a congruence [8]. As an example application, we obtain the compatibility of bisimulation-up-to techniques for CCS with replication, which so far had to be shown with an ad-hoc argument [8].

Structural congruences. A further contribution of this work consists in combining *structural congruences* [6, 7] with the bialgebraic framework using assignment rules. Structural congruences were introduced in the operational semantics of the π-calculus in [6]. The basic idea is that SOS specifications are extended with *equations* on terms, which are then linked by a special deduction rule. This rule essentially states that if two processes are equated by the congruence generated by the set of equations, then they can perform the same transitions. Prototypical examples are the specification of the parallel operator by combining a single rule with commutativity, and the specification of the replication operator by an equation, both shown below:

$$\frac{x \xrightarrow{a} x'}{x \mid y \xrightarrow{a} x' \mid y} \qquad x \mid y = y \mid x \qquad\qquad !x = !x \mid x \tag{2}$$

In [7] Mousavi and Reniers show that SOS rules with structural congruences can be interpreted in different but equivalent ways. They exhibit very simple examples of equations and SOS rules for which bisimilarity is not a congruence, even when the SOS rules are in the tyft (or the GSOS) format. As a solution to this problem they introduce a restricted format for equations, called cfsc, for which bisimilarity is a congruence when combined with tyft specifications.

In the present work we show how to interpret structural congruences at the general level of coalgebras, in terms of an operational model on closed terms. We prove that when the equations are in the cfsc format then they can be encoded by assignment rules, in such a way that their respective interpretations coincide up to bisimilarity. Consequently, not only is bisimilarity a congruence for monotone abstract GSOS combined with cfsc equations, but also bisimulation up to context and bisimilarity is compatible.

Related work. The main work on structural congruences [7] focuses on labelled transition systems, whereas our work considers the more general notion of coalgebras. As for transition systems, the basic rule format in [7] is tyft/tyxt[2], which is strictly more general than positive GSOS since it allows lookahead. However, while [7] proves congruence of bisimilarity this does not imply the compatibility (or even soundness) of bisimulation up to context [8], which we obtain in the present work (and is in fact problematic in the presence of lookahead).

In the bialgebraic setting, Klin [4] showed that by moving to CPPO-enriched categories, one can interpret recursive constructs which have a similar form as our assignment rules. Technically our approach is different, allowing us to stay in the familiar category of sets, and apply the coalgebraic bisimulation-up-to techniques which are based in this category. Further, in [4] each operator is either specified by an equation or by operational rules, disallowing a specification such as that of the parallel operator in (2).

In [2] it is shown how to obtain a distributive law for a monad which is obtained as the quotient of another monad by imposing equations on terms, under the condition that the distributive law respects the equations. However, this condition requires that the equations already hold semantically, which is fundamentally different from the present work where we define behaviour by imposing equations on an operational specification.

Acknowledgements. We would like to thank Daniel Gebler, Bartek Klin and Jan Rutten for helpful suggestions and discussions. Our research has been funded by the Netherlands Organisation for Scientific Research (NWO), CoRE project, dossier number: 612.063.920.

References

[1] B. Bloom, S. Istrail, and A. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, 1995.

[2] M. M. Bonsangue, H. H. Hansen, A. Kurz, and J. Rot. Presenting distributive laws. In R. Heckel and S. Milius, editors, *CALCO*, volume 8089 of *LNCS*, pages 95–109. Springer, 2013.

[3] M. Fiore and S. Staton. Positive structural operational semantics and monotone distributive laws. In *CMCS Short Contributions*, page 8, 2010.

[4] B. Klin. Adding recursive constructs to bialgebraic semantics. *JLAP*, 60-61:259–286, 2004.

[5] B. Klin. Bialgebras for structural operational semantics: An introduction. *TCS*, 412(38):5043–5069, 2011.

[6] R. Milner. Functions as processes. *MSCS*, 2(2):119–141, 1992.

[7] M. R. Mousavi and M. A. Reniers. Congruence for structural congruences. In V. Sassone, editor, *FoSSaCS*, volume 3441 of *LNCS*, pages 47–62. Springer, 2005.

[8] D. Pous and D. Sangiorgi. Enhancements of the bisimulation proof method. In *Advanced Topics in Bisimulation and Coinduction*, pages 233–289. CUP, 2012.

[9] J. Rot, F. Bonchi, M.M. Bonsangue, D. Pous, J.J.M.M. Rutten, and A. Silva. Enhanced coalgebraic bisimulation. `http://www.liacs.nl/~jrot/up-to.pdf`.

[10] J. Rot, M. Bonsangue, and J. Rutten. Coalgebraic bisimulation-up-to. In P. van Emde Boas, F. Groen, G. Italiano, J. Nawrocki, and H. Sack, editors, *SOFSEM*, volume 7741 of *LNCS*, pages 369–381. Springer, 2013.

[11] J. Rutten. Universal coalgebra: a theory of systems. *TCS*, 249(1):3–80, 2000.

[12] D. Sangiorgi. On the bisimulation proof method. *MSCS*, 8(5):447–479, 1998.

[13] D. Sangiorgi and D. Walker. *The Pi-Calculus - a theory of mobile processes*. CUP, 2001.

[14] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *LICS*, pages 280–291. IEEE Computer Society, 1997.

[2] In [7] it is sketched how to extend the results to the ntyft/ntyxt, which involves however a rather complicated integration of the cfsc format with the notion of stable model.