

The Quest for Equational Axiomatizations of Parallel Composition: Status and Open Problems

Luca Aceto

*Department of Computer Science
Reykjavík University
103 Reykjavík, Iceland*

*Department of Computer Science
Aalborg University
9220 Aalborg Ø, Denmark*

Wan Fokkink

*Department of Computer Science
Vrije Universiteit Amsterdam
1081 HV Amsterdam, The Netherlands*

Abstract

This essay recounts the story of the quest for equational axiomatizations of parallel composition operators in process description languages, and of similar results in the classic field of formal language theory. Some of the outstanding open problems are also mentioned.

Key words: Concurrency, process algebra, CCS, bisimulation, equational logic, Hennessy's merge, left merge, communication merge.

1 The Story So Far

Since they are designed to allow for the description and analysis of systems of interacting processes, all process description languages contain some form of parallel composition operator (also known as merge) allowing one to put two process terms in parallel with one another. These operators usually interleave the behaviours of their arguments, and allow for some form of synchronization between them. For

¹ Email: luca@ru.is

² Email: wanf@cs.vu.nl

example, Milner’s CCS offers the binary operator $|$, whose intended semantics is described by the following classic rules in Plotkin-style [20]:

$$\frac{x \xrightarrow{\mu} x'}{x | y \xrightarrow{\mu} x' | y} \quad \frac{y \xrightarrow{\mu} y'}{x | y \xrightarrow{\mu} x | y'} \quad \frac{x \xrightarrow{\alpha} x', y \xrightarrow{\bar{\alpha}} y'}{x | y \xrightarrow{\tau} x' | y'} \quad (1)$$

Although the above rules describe the behaviour of the parallel composition operator in very intuitive fashion, the equational characterization of this operator is not straightforward. In their seminal paper [14], Hennessy and Milner offered, amongst a wealth of other classic results, a complete equational axiomatization of bisimulation equivalence [19] over the recursion free fragment of CCS. The axiomatization proposed by Hennessy and Milner dealt with parallel composition using the so-called *expansion law*—a law that, intuitively, allows one to obtain a term describing the initial transitions of the parallel composition of two terms whose initial transitions are known. This law can be expressed as the following equation schema

$$\left(\sum_{i \in I} \mu_i x_i \right) | \left(\sum_{j \in J} \gamma_j y_j \right) = \sum_{i \in I} \mu_i (x_i | y) + \sum_{j \in J} \gamma_j (x | y_j) + \sum_{i \in I, j \in J, \mu_i = \bar{\gamma}_j} \tau (x_i | y_j)$$

(where I and J are two finite index sets, and the μ_i and γ_j are actions), and is nothing but an equational formulation of the aforementioned rules describing the operational semantics of parallel composition.

Despite its natural and simple formulation, the expansion law, however, is an equation schema with a countably infinite number of instances. This raised the question of whether the parallel composition operator could be axiomatized in bisimulation semantics by means of a finite collection of equations. This question was answered positively by Bergstra and Klop, who gave in [3] a finite equational axiomatization of the merge operator in terms of the auxiliary left merge and communication merge operators. Moller showed in [17,18] that strong bisimulation equivalence is not finitely based over CCS and PA without the left merge operator. (The process algebra PA [3] contains a parallel composition operator based on pure interleaving without communication—viz. an operator described by the first two rules in (1)—and the left merge operator.) Thus auxiliary operators are indeed necessary to obtain a finite axiomatization of parallel composition.

In the arguably less well known paper [13], Hennessy proposed an axiomatization of observation congruence [14] over a CCS-like recursion free process language. That axiomatization used an auxiliary operator, denoted $|'$ by Hennessy, that is essentially a combination of the left and communication merge operators as its behaviour is described by the first and the last rule in (1). The proposed axiomatization of observation congruence offered in *op. cit.* is *infinite*, as it used a variant of the expansion theorem from [14]. This led Bergstra and Klop to write in [3, page 118] that:

“It seems that γ does not have a finite equational axiomatization.”

(In *op. cit.* Bergstra and Klop used γ to denote Hennessy’s merge.) That conjecture of Bergstra and Klop’s has been confirmed by Ingolfsdottir, Luttk and us in [2] by showing that, in the presence of two distinct complementary actions, it is impossible to provide a finite axiomatization of the recursion free fragment of CCS modulo bisimulation using Hennessy’s merge operator γ . We believe that this result further reinforces the status of the left merge and the communication merge operators as auxiliary operators in the finite equational characterization of parallel composition in bisimulation semantics.

2 The Future

A possible, albeit very biased, way of trying to predict the future developments along the line of research surveyed above is to state some of the problems we are currently trying to solve.

Open Problem 1 We believe that a natural question to ask at this point is whether there is a single *binary* operator that preserves bisimulation equivalence, and whose addition to the recursion free fragment of CCS allows for the finite equational axiomatization of parallel composition—see [1, Problem 8]. We conjecture that no such operator exists, and that the use of *two* auxiliary operators is therefore necessary to achieve a finite axiomatization of parallel composition in bisimulation semantics. This result would offer the definitive justification we seek for the canonical standing of the operators proposed by Bergstra and Klop. Work on the confirmation of some form of this conjecture is under way, and we hope to report on it elsewhere in the near future. At this moment, it is not even clear to us how the general form of this conjecture could be established. How does one show that no single binary operation can be used to give a finite axiomatization of parallel composition in bisimulation semantics? Most likely there are powerful results from universal algebra and equational logic that are unknown to us and could be brought to bear on this line of work, but several literature reviews and enquiries to universal algebra mailing lists have not unearthed any answer yet.

The positive results mentioned in the previous section all deal with axiom systems that are complete when restricted to terms that contain no occurrences of variables. Much less is known regarding equational axiomatizations of behavioural equivalence over process languages with parallel composition operators that are ω -complete. Early ω -complete axiomatizations are offered in [12,16]. More recently, Fokkink and Luttk have shown in [10] that the process algebra PA [3] affords an ω -complete axiomatization that is finite if the underlying set of actions is finite.

Open Problem 2 Find ω -complete axiomatizations for bisimilarity over process algebras involving parallel composition with synchronization, e.g., for ACP.

The negative results mentioned in Sect. 1 have all been established in the setting of strong bisimulation semantics. Perhaps surprisingly, much less is known in the setting of congruences that abstract from internal steps in process behaviours. For example, is observation congruence finitely axiomatizable over the recursion free

fragment of CCS? The answer is, of course, negative, but we are still missing a proof of this fact! This leads us to state:

Open Problem 3 Prove that observation congruence has no finite equational axiomatization over the recursion, relabelling and restriction free version of CCS. Indeed, as conjectured by van Glabbeek in a recent posting on the Concurrency Mailing list, this may hold in a much stronger form. Namely, one might attempt to prove that this negative result holds true for all extensions of that language with any finite collection of GSOS operations. (Note that, in the setting of observation congruence, the operational semantics of the left and communication merge operators uses look-ahead. Therefore these two operators are *not* GSOS operations.)

Many open problems still remain, specifically in the search for ω -complete axiomatizations for rich process description languages, but the margins of this paper are too small to list them all.

3 The Heritage of Formal Language Theory

Parallel composition appears as the shuffle operator in the time-honoured theory of formal languages. Not surprisingly, the equational theory of shuffle has received considerable attention in the literature. Here we limit ourselves to mentioning some results that have a special relationship with process theory.

In [22], Tschantz offered a finite equational axiomatization of the theory of languages over concatenation and shuffle, solving an open problem raised by Pratt. In proving this result he essentially rediscovered the concept of pomset [21]—a model of concurrency based on partial orders whose algebraic aspects have been investigated by Gischer in [11]—and proved that the equational theory of series-parallel pomsets coincides with that of languages over concatenation and shuffle. The argument adopted by Tschantz was based on the observation that series-parallel pomsets may be coded by a suitable homomorphism into languages, where the series and parallel composition operators on pomsets are modelled by the concatenation and shuffle operators on languages. Tschantz’s technique of coding pomsets with languages homomorphically was further extended in the papers [5,7] to deal with several other operators, infinite pomsets and infinitary languages, and sets of pomsets. The axiomatizations by Gischer and Tschantz have later been extended in [9] to a two-sorted language with ω powers of the concatenation and parallel composition operators. The axiomatization of the algebra of pomsets resulting from the addition of these iteration operators is, however, necessarily infinite because, as shown in *op. cit.* no finite collection of equations can capture all the sound equalities involving them.

The results of Moller’s on the non-finite axiomatizability of bisimulation equivalence over the recursion free fragment of CCS and PA without the left merge operator given in [17,18] are paralleled in the world of formal language theory by those offered in [4,6,8]. In the first of those references, Bloom and Ésik proved that the valid inequations in the algebra of languages equipped with concatenation and

shuffle have no finite basis. Ésik and Bertol showed in [8] that the equational theory of union, concatenation and shuffle over languages has no finite first-order axiomatization relative to the collection of all valid inequations that hold for concatenation and shuffle. Hence the combination of some form of parallel composition, sequencing and choice is hard to characterize equationally both in the theory of languages and in that of processes. Moreover, Bloom and Ésik have shown in [6] that the variety of all languages over a finite alphabet ordered by inclusion with the operators of concatenation and shuffle, and a constant denoting the singleton language containing only the empty word is not finitely axiomatizable by first-order sentences that are valid in the equational theory of languages over concatenation, union and shuffle.

Establishing results of comparable elegance and strength in the setting of concurrency theory will be a challenge that we hope some members of our research community will meet.

References

- [1] L. ACETO, *Some of my favourite results in classic process algebra*, BRICS Report NS-03-2, BRICS, Department of Computer Science, Aalborg University, September 2003.
- [2] L. ACETO, W. FOKKINK, A. INGOLFSDOTTIR, AND B. LUTTIK, *CCS with Hennessy's merge has no finite equational axiomatization*, Theoretical Comput. Sci., 330 (2005), pp. 377–405.
- [3] J. BERGSTRA AND J. W. KLOP, *Process algebra for synchronous communication*, Information and Control, 60 (1984), pp. 109–137.
- [4] S. L. BLOOM AND Z. ÉSIK, *Nonfinite axiomatizability of shuffle inequalities*, in Proceedings of TAPSOFT'95: Theory and Practice of Software Development, 6th International Joint Conference CAAP/FASE, Aarhus, Denmark, May 22–26, 1995, P. D. Mosses, M. Nielsen, and M. I. Schwartzbach, eds., vol. 915 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 318–333.
- [5] ———, *Free shuffle algebras in language varieties*, Theoret. Comput. Sci., 163 (1996), pp. 55–98.
- [6] ———, *Axiomatizing shuffle and concatenation in languages*, Inform. and Comput., 139 (1997), pp. 62–91.
- [7] ———, *Varieties generated by languages with poset operations*, Math. Structures Comput. Sci., 7 (1997), pp. 701–713.
- [8] Z. ÉSIK AND M. BERTOL, *Nonfinite axiomatizability of the equational theory of shuffle*, Acta Inform., 35 (1998), pp. 505–539.
- [9] Z. ÉSIK AND S. OKAWA, *Series and parallel operations on pomsets*, in Proceedings of Foundations of Software Technology and Theoretical Computer Science (Chennai,

- 1999), vol. 1738 of Lecture Notes in Comput. Sci., Springer-Verlag, Berlin, 1999, pp. 316–328.
- [10] W. FOKKINK AND B. LUTTIK, *An omega-complete equational specification of interleaving*, in Proceedings 27th Colloquium on Automata, Languages and Programming—ICALP'00, Geneva, U. Montanari, J. Rolinn, and E. Welzl, eds., vol. 1853 of Lecture Notes in Computer Science, Springer-Verlag, July 2000, pp. 729–743.
- [11] J. L. GISCHER, *The equational theory of pomsets*, Theoretical Comput. Sci., 61 (1988), pp. 199–224.
- [12] J. F. GROOTE, *A new strategy for proving ω -completeness with applications in process algebra*, in Proceedings CONCUR 90, Amsterdam, J. Baeten and J. Klop, eds., vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 314–331.
- [13] M. HENNESSY, *Axiomatising finite concurrent processes*, SIAM J. Comput., 17 (1988), pp. 997–1017.
- [14] M. HENNESSY AND R. MILNER, *Algebraic laws for nondeterminism and concurrency*, J. Assoc. Comput. Mach., 32 (1985), pp. 137–161.
- [15] R. MILNER, *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989.
- [16] F. MOLLER, *Axioms for Concurrency*, PhD thesis, Department of Computer Science, University of Edinburgh, July 1989. Report CST-59-89. Also published as ECS-LFCS-89-84.
- [17] ———, *The importance of the left merge operator in process algebras*, in Proceedings 17th ICALP, Warwick, M. Paterson, ed., vol. 443 of Lecture Notes in Computer Science, Springer-Verlag, July 1990, pp. 752–764.
- [18] ———, *The nonexistence of finite axiomatisations for CCS congruences*, in Proceedings 5th Annual Symposium on Logic in Computer Science, Philadelphia, USA, IEEE Computer Society Press, 1990, pp. 142–153.
- [19] D. Park, *Concurrency and automata on infinite sequences*, in: P. Deussen (Ed.), 5th GI Conference, Karlsruhe, Germany, Vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.
- [20] G. PLOTKIN, *A structural approach to operational semantics*, Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.
- [21] V. PRATT, *Modeling concurrency with partial orders*, International Journal of Parallel Programming, 15 (1986), pp. 33–71.
- [22] S. T. TSCHANTZ, *Languages under concatenation and shuffling*, Mathematical Structures in Computer Science, 4 (1994), pp. 505–511.