

# Impossibility Results in the Equational Logic of Processes

Luca Aceto<sup>1,2</sup>

*Department of Computer Science  
School of Science and Engineering  
Reykjavík University  
103 Reykjavík, Iceland*

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

---

## Abstract

This talk offers a survey of negative results on the existence of finite equational axiomatizations for bisimulation equivalence over fragments of algebraic process calculi.

*Keywords:* Concurrency, process algebra, CCS, ACP, bisimulation, equational logic, complete axiomatizations, Hennessy's merge, left merge, communication merge, priority, interrupt.

---

Process algebras are prototype specification languages for reactive systems—that is, for systems that compute by reacting to stimuli from their environment. They consist of a collection of basic operations for constructing new system descriptions from existing ones, together with some facility for the recursive definition of system behaviours. Well-known examples of such languages are CCS [13], CSP [10] and ACP [6].

One may define intuitively appealing semantic models for a process algebra as quotients of the collection of labelled transition systems modulo some behavioural congruence. *Labelled transition systems* (LTSs) [11] are a fundamental formalism for the description of concurrent computation, which is widely used in light of its flexibility and applicability. In particular, they underlie Plotkin's Structural Operational Semantics [17] and, following Milner's pioneering work on CCS [13], are by now the standard formalism for describing the semantics of various process description languages.

---

<sup>1</sup> The work reported in this paper has been partly supported by the project “The Equational Logic of Parallel Processes” (nr. 060013021) of The Icelandic Research Fund.

<sup>2</sup> Email: [luca@ru.is](mailto:luca@ru.is)

LTSs model processes by explicitly describing their states and their transitions from state to state, together with the actions that produced them. Since this view of process behaviours is very detailed, several notions of behavioural equivalence and preorder have been proposed for LTSs. The aim of such behavioural semantics is to identify those (states of) LTSs that afford the same “observations”, in some appropriate technical sense. The lack of consensus on what constitutes a suitable notion of observable behaviour for reactive systems has led to a large number of proposals for behavioural equivalences for concurrent processes. (See the study [9], where van Glabbeek presents the linear time/branching time spectrum—a lattice of known behavioural equivalences and preorders over LTSs, ordered by inclusion.) In this talk, I shall only focus on the most classic such behavioural equivalence, namely *bisimulation equivalence* [13,16].

Having defined a semantic model for a process algebra in terms of LTSs, it is natural to study its equational theory—that is, the collection of equations that are valid in the given model. The key questions here are:

- Are there “reasonably informative” complete axiom systems for the chosen semantic model? Researchers in concurrency theory often restrict themselves to studying axiom systems that are complete with respect to the collection of valid equations that do not contain occurrences of variables.
- Does the algebra of LTSs modulo the chosen notion of behavioural equivalence afford a finite equational axiomatization?

A complete axiomatization of a behavioural congruence yields a purely syntactic characterization, independent of LTSs and of the actual details of the definition of the chosen behavioural equivalence, of the semantics of the process algebra. This bridge between syntax and semantics plays an important role in both the practice and the theory of process algebras. From the point of view of practice, these proof systems can be used to perform system verifications in a purely syntactic way using general purpose theorem provers or proof checkers, and form the basis of purpose built axiomatic verification tools like, e.g., PAM [12]. A positive answer to the first basic question raised above is therefore not just theoretically pleasing, but has potential practical applications. From the theoretical point of view, complete axiomatizations of behavioural equivalences capture the essence of different notions of semantics for processes in terms of a basic collection of identities, and this often allows one to compare semantics which may have been defined in very different styles and frameworks. A review of existing complete equational axiomatizations for many of the behavioural semantics in van Glabbeek’s spectrum is offered in [9]. The equational axiomatizations offered *ibidem* are over the language BCCSP, a common fragment of Milner’s CCS [13] and Hoare’s CSP [10] suitable for describing finite synchronization trees, and characterize the differences between behavioural semantics in terms of a few revealing axioms.

If the answer to the second basic question mentioned above is negative, then one may resort to expanding the signature with auxiliary operations, thus adding expressive power for the purpose of axiomatizing the equational theory. A classic

example of this line of research is offered by the paper [8]. There Bergstra and Klop showed how to give a finite axiomatization of the language ACP using the auxiliary left and communication merge operators to characterize parallel composition. As shown by Moller [14,15], auxiliary operators are needed to obtain a finite basis for that language because the process algebras CCS and ACP without the auxiliary left merge operator from [7] do not have a finite equational axiomatization modulo bisimulation equivalence.

The aforementioned results of Moller’s are the earliest examples of non-finite axiomatizability results I am aware of in the setting of process algebra. These results, like all impossibility theorems in general, hold a great fascination for me, and have inspired me to study the expressive power of equational logic in the characterization of behavioural equivalences over fragments of process description languages. In this talk, I shall present a guided tour of the landscape of non-finite axiomatizability results for bisimulation equivalence over fragments of process algebras. In particular, I shall try to motivate the importance of these negative results, to describe the general proof techniques that can be used to establish them, and to introduce a menagerie of operations—for instance, versions of parallel composition and mode transfer operators like interrupt and priority—whose addition to very basic languages for the description of finite process behaviours spoils finite axiomatizability.

The technical contents of this talk will be based upon joint work with Taolue Chen, Wan Fokkink, Anna Ingolfsdottir, Bas Luttik and Sumit Nain presented in the papers [1,2,3,4,5].

## References

- [1] Aceto, L., T. Chen, W. Fokkink and A. Ingolfsdottir, *On the axiomatizability of priority*, Technical Report RS-06-1, BRICS (2006), available from <http://www.brics.dk/RS/06/1/>.
- [2] Aceto, L., W. Fokkink, A. Ingolfsdottir and B. Luttik, *CCS with Hennessy’s merge has no finite equational axiomatization*, Theor. Comput. Sci. **330** (2005), pp. 377–405.
- [3] Aceto, L., W. Fokkink, A. Ingolfsdottir and B. Luttik, *Finite equational bases in process algebra: Results and open questions*, in: A. Middeldorp, V. van Oostrom, F. van Raamsdonk and R. C. de Vrijer, editors, *Processes, Terms and Cycles*, Lecture Notes in Computer Science **3838** (2005), pp. 338–367.
- [4] Aceto, L., W. Fokkink, A. Ingolfsdottir and B. Luttik, *A finite equational base for CCS with left merge and communication merge* (2006), an extended abstract of the paper is available from <http://www.cs.aau.dk/~luca/PAPERS/EqBaseCCS.pdf>.
- [5] Aceto, L., W. Fokkink, A. Ingolfsdottir and S. Nain, *Bisimilarity is not finitely based over BPA with interrupt*, in: J. L. Fiadeiro, N. Harman, M. Roggenbach and J. J. M. M. Rutten, editors, *CALCO*, Lecture Notes in Computer Science **3629** (2005), pp. 52–66.
- [6] Baeten, J. C. and P. Weijland, “Process Algebra,” Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.
- [7] Bergstra, J. and J. W. Klop, *Fixed point semantics in process algebras*, Report IW 206, Mathematisch Centrum, Amsterdam (1982).
- [8] Bergstra, J. and J. W. Klop, *Process algebra for synchronous communication*, Information and Control **60** (1984), pp. 109–137.
- [9] Glabbeek, R. v., *The linear time-branching time spectrum. I. The semantics of concrete, sequential processes*, in: *Handbook of Process Algebra*, North-Holland, Amsterdam, 2001 pp. 3–99.
- [10] Hoare, C., “Communicating Sequential Processes,” Prentice-Hall International, Englewood Cliffs, 1985.

- [11] Keller, R., *Formal verification of parallel programs*, Communications of the ACM **19** (1976), pp. 371–384.
- [12] Lin, H., *An interactive proof tool for process algebras*, in: *9th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science **577** (1992), pp. 617–618.
- [13] Milner, R., “Communication and Concurrency,” Prentice-Hall International, Englewood Cliffs, 1989.
- [14] Moller, F., *The importance of the left merge operator in process algebras*, in: M. Paterson, editor, *Proceedings 17<sup>th</sup> ICALP*, Warwick, Lecture Notes in Computer Science **443** (1990), pp. 752–764.
- [15] Moller, F., *The nonexistence of finite axiomatisations for CCS congruences*, in: *Proceedings 5<sup>th</sup> Annual Symposium on Logic in Computer Science*, Philadelphia, USA (1990), pp. 142–153.
- [16] Park, D., *Concurrency and automata on infinite sequences*, in: P. Deussen, editor, *5<sup>th</sup> GI Conference*, Karlsruhe, Germany, Lecture Notes in Computer Science **104** (1981), pp. 167–183.
- [17] Plotkin, G. D., *A structural approach to operational semantics.*, Journal of Logic and Algebraic Programming **60–61** (2004), pp. 17–139.