Bisimilarity Is Not Finitely Based
over BPA with Interrupt

Luca Aceto\textsuperscript{1,3}, Wan Fokkink\textsuperscript{4}, Anna Ingolfsdottir\textsuperscript{1,2}, and Sumit Nain\textsuperscript{1}

\textsuperscript{1} Brics (Basic Research in Computer Science), Centre of the Danish National Research Foundation, Department of Computer Science, Aalborg University, Fr. Bajersvej 7B, 9220 Aalborg Ø, Denmark \{luca, annai, nain\}@cs.aau.dk

\textsuperscript{2} Department of Computer Science, University of Iceland, 107 Reykjavík, Iceland annaining@hi.is

\textsuperscript{3} School of Computer Science, Reykjavík University, Ofanleiti 2, 103 Reykjavík, Iceland luca@ru.is

\textsuperscript{4} Vrije Universiteit Amsterdam, Department of Computer Science, Section Theoretical Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands wanf@cs.vu.nl

Abstract. This paper shows that bisimulation equivalence does not afford a finite equational axiomatization over the language obtained by enriching Bergstra and Klop’s Basic Process Algebra with the interrupt operator. Moreover, it is shown that the collection of closed equations over this language is also not finitely based. In sharp contrast to these results, the collection of closed equations over the language BPA enriched with the disrupt operator is proven to be finitely based.

1 Introduction

Programming and specification languages often include constructs to specify mode switches (see, e.g., [8,11,23,24,26]). Indeed, some form of mode transfer in computation appears in the time-honoured theory of operating systems in the guise of, e.g., interrupts, in programming languages as exceptions, and in the behaviour of control programs and embedded systems as discrete “mode switches” triggered by changes in the state of their environment.

In light of the ubiquitous nature of mode changes in computation, it is not surprising that classic process description languages either include primitive operators to describe mode changes—for example, LOTOS [15,23] offers the so-called disruption operator—or have been extended with variations on mode transfer operators. For instance, examples of such operators that may be added to CCS are discussed by Milner in [25, pp. 192–193], and the reference [17] offers some discussion of the benefits of adding one of those, viz. the checkpointing operator, to that language.

In the setting of Basic Process Algebra (BPA), as introduced by Bergstra and Klop in [12], some of these extensions, and their relative expressiveness, have been discussed in the early paper [11]. That preprint of Bergstra’s has later been revised and extended.
Bisimilarity Is Not Finitely Based over BPA with Interrupt 53

There, Baeten and Bergstra study the equational theory and expressiveness of BPA$\delta$ (the extension of BPA with a constant $\delta$ to describe “deadlock”) enriched with two mode transfer operators, viz. the disrupt and interrupt operators. In particular, they offer an equational axiomatization of bisimulation equivalence \cite{25, 29} over the resulting extension of the language BPA$\delta$. This axiomatization is finite, if so is the underlying set of actions—a state of affairs that is most pleasing for process algebraists.

However, the axiomatization of bisimulation equivalence offered by Baeten and Bergstra in \cite{7} relies on the use of four auxiliary operators—two per mode transfer operator. Although the use of auxiliary operators in the axiomatization of behavioral equivalences over process description languages has been well established since Bergstra and Klop’s axiomatization of parallel composition using the left and communication merge operators \cite{13}, to our mind, a result like the aforementioned one always begs the question whether the use of auxiliary operators is necessary to obtain a finite axiomatization of bisimulation equivalence.

For the case of parallel composition, Moller showed in \cite{27, 28} that strong bisimulation equivalence is not finitely based over CCS \cite{25} and PA \cite{13} without the left merge operator. (The process algebra PA \cite{13} contains a parallel composition operator based on pure interleaving without communication, and the left merge operator.) Thus auxiliary operators are necessary to obtain a finite axiomatization of parallel composition. But, is the use of auxiliary operators necessary to give a finite axiomatization of bisimulation equivalence over the language BPA enriched with the mode transfer operators studied by Baeten and Bergstra in \cite{7}?

We address the above natural question in this paper. In particular, we focus on BPA enriched with the interrupt operator. Intuitively, “$p$ interrupted by $q$” describes a process that normally behaves like $p$. However, at each point of the computation before $p$ terminates, $q$ can interrupt it, and begin its execution. If this happens, $p$ resumes its computation upon termination of $q$.

We show that, in the presence of two distinct actions, bisimulation equivalence is not finitely based over BPA with the interrupt operator. Moreover, we prove that the collection of closed equations over this language is also not finitely based. This result provides evidence that the use of auxiliary operators in the technical developments presented in \cite{7} is indeed necessary in order to obtain a finite axiomatization of bisimulation equivalence.

Our main result adds the interrupt operator to the list of operators whose addition to a process algebra spoils finite axiomatizability modulo bisimulation equivalence; see, e.g., \cite{4, 14, 16, 20, 30, 31} for other examples of non-finite axiomatizability results over process algebras, and some of their precursors in the setting of formal language theory. Of special relevance for concurrency theory are the aforementioned results of Moller’s to the effect that the process algebras CCS and PA without the auxiliary left merge operator from \cite{12} do not have a finite equational axiomatization modulo bisimulation equivalence \cite{27, 28}. Recently, in collaboration with Luttik, the first three authors have shown in \cite{5} that the process algebra obtained by adding Hennessy’s merge operator from \cite{22} to CCS does not have a finite equational axiomatization modulo bisimulation equivalence. This result is in sharp contrast with a theorem established by Fokkink and Luttik in \cite{18} to the effect that the process algebra PA \cite{13} affords an $\omega$-complete ax-