BRICS

Basic Research in Computer Science

The Saga of the Axiomatization of Parallel Composition

Luca Aceto
Anna Ingólfsdóttir

BRICS Report Series RS-07-11

ISSN 0909-0878 June 2007

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
IT-parken, Aabogade 34
DK–8200 Aarhus N
Denmark

Telephone:	+45 8942 9300
Telefax:	+45 8942 5601
Internet:	BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/07/11/`

The Saga of the Axiomatization of Parallel Composition[⋆]

Luca Aceto and Anna Ingolfsdottir

Department of Computer Science, Reykjavík University
Kringlan 1, 103 Reykjavík, Iceland
luca@ru.is, annai@ru.is

Abstract. This paper surveys some classic and recent results on the finite axiomatizability of bisimilarity over CCS-like languages. It focuses, in particular, on non-finite axiomatizability results stemming from the semantic interplay between parallel composition and nondeterministic choice. The paper also highlights the role that auxiliary operators, such as Bergstra and Klop's left and communication merge and Hennessy's merge operator, play in the search for a finite, equational axiomatization of parallel composition both for classic process algebras and for their real-time extensions.

1 The Problem and its History

Process algebras are prototype description languages for reactive systems that arose from the pioneering work of figures like Bergstra, Hoare, Klop and Milner. Well-known examples of such languages are ACP [18], CCS [44], CSP [40] and Meije [13]. These algebraic description languages for processes differ in the basic collection of operators that they offer for building new process descriptions from existing ones. However, since they are designed to allow for the description and analysis of systems of interacting processes, all these languages contain some form of *parallel composition* (also known as *merge*) operator allowing one to put two process terms in parallel with one another. These operators usually interleave the behaviours of their arguments, and support some form of synchronization between them.

For example, Milner's CCS offers the binary operator $||$, whose intended semantics is described by the following classic rules in the style of Plotkin [49].

$$\frac{x \xrightarrow{\mu} x'}{x \,||\, y \xrightarrow{\mu} x' \,||\, y} \qquad \frac{y \xrightarrow{\mu} y'}{x \,||\, y \xrightarrow{\mu} x \,||\, y'} \qquad \frac{x \xrightarrow{\alpha} x',\ y \xrightarrow{\bar{\alpha}} y'}{x \,||\, y \xrightarrow{\tau} x' \,||\, y'} \tag{1}$$

(In the above rules, the symbol μ stands for an action that a process may perform, α and $\bar{\alpha}$ are two observable actions that may synchronize, and τ is a symbol denoting the result of their synchronization.)

[⋆] This paper is based on joint work with Wan Fokkink, Bas Luttik and MohammadReza Mousavi. The authors were partly supported by the project "The Equational Logic of Parallel Processes" (nr. 060013021) of The Icelandic Research Fund.

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 [39], Hennessy and Milner offered, amongst a wealth of other classic results, a complete equational axiomatization of bisimulation equivalence [48] over the recursion-free fragment of CCS. (See the paper [14] for a more detailed historical account highlighting, e.g., Hans Bekić's early contributions to this field of research.) The axiomatization proposed by Hennessy and Milner in [39] dealt with parallel composition using the so-called *expansion law*—a law that, intuitively, allows one to obtain a term describing explicitly 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

$$\left(\sum_{i\in I}\mu_i x_i\right) \,||\, \left(\sum_{j\in J}\gamma_j y_j\right) = \sum_{i\in I}\mu_i(x_i \,||\, y) + \sum_{j\in J}\gamma_j(x \,||\, y_j) + \sum_{\substack{i\in I, j\in J\\ \mu_i=\overline{\gamma_j}}}\tau(x_i \,||\, y_j) \quad (2)$$

(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 [20] a finite equational axiomatization of the merge operator in terms of the auxiliary left merge and communication merge operators. Moller showed in [46, 47] that bisimulation equivalence is not finitely based over CCS and PA without the left merge operator. (The process algebra PA [20] 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.) These results, which we survey in Section 2, indicate that auxiliary operators are necessary to obtain a finite axiomatization of parallel composition.

Moller's results clarified the role played by the expansion law in the equational axiomatization of parallel composition over CCS and, to the best of our knowledge, were the first negative results on the existence of finite equational axiomatizations for algebras of processes that were presented in the literature. To our mind, the negative results achieved by Moller in his doctoral dissertation removed a psychological barrier by showing that non-finite axiomatizability results could indeed be achieved also in the study of process algebras, and paved the way to the further developments we describe henceforth in this paper.

The contributions our collaborators and we have offered so far to the saga of the axiomatization of parallel composition have been mostly motivated by an attempt to answer the following questions.

1. Are there other "natural" auxiliary operators that can be used, in lieu of Bergstra and Klop's left and communication merge, to achieve a finite equational axiomatization of parallel composition?
2. Do the aforementioned results hold true also for extensions of classic process algebras like CCS with features such as real-time?

As far as the former motivating question is concerned, the literature on process algebra offers at least one alternative proposal to the use of the left and communication merge operators. In the paper [38], which we believe is not so well known as it deserves to be, Hennessy proposed an axiomatization of observation congruence [39] and split-2 congruence over a CCS-like recursion-free process language. (It is worth noting for the sake of historical accuracy that the results reported in [38] were actually obtained in 1981; see the preprint [36].) Those axiomatizations 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). Apart from having soundness problems (see the reference [2] for a general discussion of this problem, and corrected proofs of Hennessy's results), the proposed axiomatization of observation congruence offered in [38] is *infinite*, as it uses a variant of the expansion law from [39]. This led Bergstra and Klop to write in [20, page 118] that:

> "It seems that γ does not have a finite equational axiomatization."

(In [20] Bergstra and Klop used γ to denote Hennessy's merge.) In Section 3, we will present an answer to this conjecture of Bergstra and Klop's 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 equivalence using Hennessy's merge operator $|\!\!/$. We believe that this result, which was originally proved in [6], 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. Interestingly, as shown in [8], in sharp contrast to the situation in standard bisimulation semantics, CCS with Hennessy's merge *can be finitely axiomatized* modulo split-2 bisimulation equivalence [33, 38]. (Split-2 bisimilarity is defined like standard bisimilarity, but is based on the assumption that action occurrences have a beginning and an ending, and that these events may be observed.) This shows that, in sharp contrast to the results offered in [45, 46], "reasonable congruences" finer than standard bisimulation equivalence can be finitely axiomatized over CCS using Hennessy's merge as the single auxiliary operation—compare with the non-finite axiomatizability results for these congruences offered in [45, 46].

It is also natural to ask oneself whether the aforementioned non-finite axiomatizability results hold true also for extensions of the basic CCS calculus with features such as real-time. In Section 4, we review some negative results, originally shown in [12], on the finite axiomatizability of timed bisimilarity over Yi's timed CCS [52, 53]. In particular, we prove that timed bisimilarity is not finitely based both for single-sorted and two-sorted presentations of timed CCS. We further strengthen this result by showing that, unlike in the setting of CCS, adding the untimed or the timed left merge operator to the syntax and semantics of timed CCS does not solve the axiomatizability problem. To our mind, these results indicate that the expressive power that is gained by adding to CCS linguistic features suitable for the description of timing-based behaviours substantially complicates the equational theory of the resulting algebras of processes.

We feel that there are still many chapters to be written in the saga of the study of the equational logic of parallel composition, and we list a few open problems and directions of ongoing research throughout this paper.

Related Work in Concurrency and Formal Language Theory The equational characterization of different versions of the parallel composition operator is a classic topic in the theory of computation. In particular, the process algebraic literature abounds with results on equational axiomatizations of various notions of behavioural equivalence or preorder over languages incorporating some notion of parallel composition—see, e.g., the textbooks [18, 30, 37, 44] and the classic papers [20, 39, 43] for general references. Early ω-complete axiomatizations are offered in [35, 45]. More recently, Fokkink and Luttik have shown in [31] that the process algebra PA [20] affords an ω-complete axiomatization that is finite if so is the underlying set of actions. As shown in [9], the same holds true for the fragment of CCS without recursion, relabelling and restriction extended with the left and communication merge operators. The readers will find a survey of recent results on the equational logic of processes in [7], and further non-finite axiomatizability results for rather basic process algebras in, e.g., [4, 10].

An analysis of the reasons why operators like the left merge and the communication merge are equationally well behaved in bisimulation semantics has led to general algorithms for the generation of (finite) equational axiomatizations for behavioural equivalences from various types of transition system specifications—see, e.g., [1, 3, 15] and the references in [11] for further details.

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 close relationship with process theory.

In [51], 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 [34, 50]—a model of concurrency based on partial orders whose algebraic aspects have been investigated by Gischer in [32]—, and proved that the equational theory of series-parallel pomsets coincides with that of languages over concatenation and shuffle. The argument adopted by Tschantz in his proof 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, respectively. Tschantz's technique of coding pomsets with languages homomorphically was further extended in the papers [22, 24, 25] to deal with several other operators, infinite pomsets and infinitary languages, as well as sets of pomsets. The axiomatizations by Gischer and Tschantz have later been extended in [25, 29] 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 [29], 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 [46, 47] are paralleled in the world of formal language theory by those offered in [21, 23, 28]. 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 [28] 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 [23] 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.

2 Background

The core process algebra that we shall consider henceforth in this paper is a fragment of Milner's CCS. This language, which will be referred to simply as CCS, is given by the following grammar:

$$t ::= x \quad | \quad \mathbf{0} \quad | \quad at \quad | \quad \bar{a}t \quad | \quad \tau t \quad | \quad t+t \quad | \quad t \,||\, t \enspace ,$$

where x is a variable drawn from a countably infinite set V, a is an action, and $\bar{a}$ is its complement. We assume that the actions a and $\bar{a}$ are distinct. Following Milner [44], the action symbol τ will result from the synchronized occurrence of the complementary actions a and $\bar{a}$. We let $\mu \in \{a, \bar{a}, \tau\}$ and $\alpha \in \{a, \bar{a}\}$. (We remark, in passing, that this small collection of actions suffices to prove all the negative results we survey in this study. All the positive results we shall present in what follows hold for arbitrary finite sets of actions.) As usual, we postulate that $\bar{\bar{a}} = a$. We shall use the meta-variables t, u to range over process terms. The *size* of a term is the number of operator symbols in it. A process term is *closed* if it does not contain any variables. Closed terms will be typically denoted by p, q.

In the remainder of this paper, we let a^0 denote $\mathbf{0}$, and a^{m+1} denote $a(a^m)$. We sometimes simply write a in lieu of a^1.

$$\frac{}{\mu x \xrightarrow{\mu} x} \qquad \frac{x \xrightarrow{\mu} x'}{x + y \xrightarrow{\mu} x'} \qquad \frac{y \xrightarrow{\mu} y'}{x + y \xrightarrow{\mu} y'}$$

$$\frac{x \xrightarrow{\mu} x'}{x \,||\, y \xrightarrow{\mu} x' \,||\, y} \qquad \frac{y \xrightarrow{\mu} y'}{x \,||\, y \xrightarrow{\mu} x \,||\, y'} \qquad \frac{x \xrightarrow{\alpha} x',\ y \xrightarrow{\bar{\alpha}} y'}{x \,||\, y \xrightarrow{\tau} x' \,||\, y'}$$

Table 1. SOS Rules for the CCS Operators ($\mu \in \{a, \bar{a}, \tau\}$ and $\alpha \in \{a, \bar{a}\}$)

The SOS rules for the above language are standard, and may be found in Table 1. These transition rules give rise to transitions between closed terms. The operational semantics for our language, and for all its extensions that we shall introduce in the remainder of this paper, is thus given by a labelled transition system [42] whose states are closed terms, and whose labelled transitions are those that are provable using the rules that are relevant for the language under consideration.

In this paper, we shall consider our core language and all its extensions modulo bisimulation equivalence [44, 48].

Definition 1. *Bisimulation equivalence (also sometimes referred to as bisimilarity), denoted by* $\underline{\leftrightarrow}$*, is the largest symmetric relation over closed terms such that whenever* $p \,\underline{\leftrightarrow}\, q$ *and* $p \xrightarrow{\mu} p'$*, then there is a transition* $q \xrightarrow{\mu} q'$ *with* $p' \,\underline{\leftrightarrow}\, q'$*.*

If $p \,\underline{\leftrightarrow}\, q$*, then we say that* p *and* q *are bisimilar.*

It is well known that, as its name suggests, bisimulation equivalence is indeed an equivalence relation (see, e.g., the references [44, 48]). Since the SOS rules in Table 1 (and all of the other rules we shall introduce in the remainder of this paper) are in de Simone's format [27], bisimulation equivalence is a congruence.

Bisimulation equivalence is extended to arbitrary terms in the standard way.

2.1 Classic Results on Equational Axiomatizations

An *axiom system* is a collection of equations $t \approx u$, where t and u are terms. An equation $t \approx u$ is derivable from an axiom system E if it can be proved from the axioms in E using the rules of equational logic (viz. reflexivity, symmetry, transitivity, substitution and closure under contexts). An equation $t \approx u$ is *sound* with respect to $\underline{\leftrightarrow}$ iff $t \,\underline{\leftrightarrow}\, u$. An axiom system is sound with respect to $\underline{\leftrightarrow}$ iff so is each of its equations. For example, the axiom system in Table 2 is sound. In what follows, we use a *summation* $\sum_{i \in \{1,\ldots,k\}} t_i$ to denote $t_1 + \cdots + t_k$, where the empty sum represents $\mathbf{0}$.

An axiom system E is a *complete axiomatization* of $\underline{\leftrightarrow}$ over (some extension of) CCS if E is sound with respect to $\underline{\leftrightarrow}$, and proves all of the equations over the language that are sound with respect to $\underline{\leftrightarrow}$. If E is sound with respect to $\underline{\leftrightarrow}$, and proves all of the *closed* equations over the language that are sound with respect to $\underline{\leftrightarrow}$, then we say that E is *ground complete*.

The study of equational axiomatizations of bisimilarity over process algebras was initiated by Hennessy and Milner, who proved the following classic result.

A1	$x + y \approx y + x$
A2	$(x + y) + z \approx x + (y + z)$
A3	$x + x \approx x$
A4	$x + \mathbf{0} \approx x$

Table 2. Some Axioms for Bisimilarity

Theorem 1 (Hennessy and Milner [39]). *The axiom system consisting of equations* A1–A4 *and all of the instances of (2) is ground complete for bisimilarity over* CCS*.*

Since the equation schema (2) has infinitely many instances, the above theorem 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 [20] a finite ground-complete axiomatization of the merge operator in terms of the auxiliary left merge and communication merge operators. The operational rules for these operators are

$$\frac{x \xrightarrow{\mu} x'}{x \mathbin{\lfloor\!\lfloor} y \xrightarrow{\mu} x' \,||\, y} \qquad \frac{x \xrightarrow{\alpha} x',\ y \xrightarrow{\bar{\alpha}} y'}{x \,|\, y \xrightarrow{\tau} x' \,||\, y'}$$

where $\lfloor\!\lfloor$ and $|$ stand for the left and communication merge operators, respectively.

But, are auxiliary operators necessary to obtain a finite equational axiomatization of bisimilarity over the language CCS? This question remained unanswered for about a decade until Moller proved the following seminal result in his doctoral dissertation.

Theorem 2 (Moller [45, 47]). *Bisimilarity has no finite, (ground-)complete equational axiomatization over* CCS*.*

Thus auxiliary operators are indeed necessary to obtain a finite axiomatization of parallel composition, and the expansion law cannot be replaced by a finite collection of sound equations.

Moller's proof of the theorem above is based on the observation that, since $||$ does not distribute over $+$, no finite, sound axiom system over CCS can be powerful enough to "expand" the initial behaviour of a term of the form $a \,||\, p$ when p has a sufficiently large number of initial transitions leading to non-bisimilar terms. It follows that no finite collection of sound axioms is as powerful as the expansion law (2). Technically, Moller showed that, when n is greater than the size of the largest term in a finite, sound axiom system E over the language CCS, E cannot prove the sound equation

$$a \,||\, \sum_{i=1}^{n} a^i \approx a(\sum_{i=1}^{n} a^i) + \sum_{i=2}^{n+1} a^i \ .$$

Note that, up to bisimilarity, the right-hand side of the above equation expresses "syntactically" the collection of initial transitions of the term on the left-hand side.

Remark 1. Theorem 2 holds true for each "reasonable congruence" over CCS. A congruence is "reasonable" in the sense of Moller if it is included in bisimilarity and satisfies the family of equations Red_n presented in [45, page 111].

3 The Role of Hennessy's Merge

Theorem 2 shows that one cannot hope to achieve a finite (ground-)complete axiomatization for bisimilarity over CCS without recourse to auxiliary operators. Moreover, the work by Bergstra and Klop presented in [20] tells us that a finite ground-complete axiomatization can be obtained at the price of adding the left and communication merge operators to the language. (In fact, as shown in [9], for any finite set of actions the resulting language also affords a finite *complete* axiomatization modulo bisimilarity.) A natural question to ask at this point is whether one can obtain a finite equational axiomatization of bisimilarity over CCS extended with some auxiliary binary operator other than those proposed by Bergstra and Klop. An independent proposal, put forward by Hennessy in [36, 38], is to add the auxiliary operator $|\!\!/$ with the following SOS rules to the signature for CCS.

$$\frac{x \xrightarrow{\mu} x'}{x \mathbin{|\!\!/} y \xrightarrow{\mu} x' \,||\, y} \qquad \frac{x \xrightarrow{\alpha} x',\; y \xrightarrow{\bar{\alpha}} y'}{x \mathbin{|\!\!/} y \xrightarrow{\tau} x' \,||\, y'}$$

Note that the above operator is essentially a combination of the left and communication merge operators. We denote the resulting language by CCS_H.

Does bisimilarity afford a finite equational axiomatization over CCS_H? In [20, page 118], Bergstra and Klop conjectured a negative answer to the above question. Their conjecture was finally confirmed about twenty years later by the following theorem.

Theorem 3 (Aceto, Fokkink, Ingolfsdottir and Luttik [6]). *Bisimulation equivalence admits no finite (ground-)complete equational axiomatization over the language* CCS_H.

The aforementioned negative result holds in a very strong form. Indeed, we prove that no finite collection of equations over CCS_H that are sound with respect to bisimulation equivalence can prove all of the sound closed equalities of the form

$$e_n : \quad a \mathbin{|\!\!/} p_n \approx ap_n + \sum_{i=0}^{n} \tau a^i \quad (n \geq 0) \ ,$$

where the terms p_n are defined thus:

$$p_n = \sum_{i=0}^{n} \bar{a}a^i \quad (n \geq 0) \ .$$

The proof of Theorem 3 is given along proof-theoretic lines that have their roots in Moller's proof of Theorem 2. However, the presence of possible synchronizations in the terms used in the family of equations e_n is necessary for our result, and requires careful attention in our proof. (Indeed, in the absence of synchronization, Hennessy's merge reduces to Bergstra and Klop's left merge operator, and thus affords a finite equational axiomatization.) In particular, the infinite family of equations e_n and our arguments based upon it exploit the inability of any finite axiom system E that is sound with respect to bisimulation equivalence to "expand" the synchronization behaviour of terms of the form $p \mathbin{|\!/} q$, for terms q that, like the terms p_n above eventually do, have a number of inequivalent "summands" that is larger than the maximum size of the terms mentioned in equations in E. As in the original arguments of Moller's, the root of this problem can be traced back to the fact that, since $|\!/$ distributes with respect to the choice operator $+$ in the first argument but *not* in the second, no finite collection of equations can express the interplay between interleaving and communication that underlies the semantics of Hennessy's merge.

Our Theorem 3 is the counterpart of Moller's Theorem 2 over the language CCS_H. As we recalled in Remark 1, Moller's non-finite axiomatizability result for CCS holds for each "reasonable" congruence. It is therefore natural to ask ourselves whether each "reasonable" congruence is not finitely based over CCS_H too. The following result shows that, in sharp contrast to the situation in standard bisimulation semantics, the language CCS_H *can be finitely axiomatized* modulo split-2 bisimulation equivalence [36, 38], and therefore that, modulo this non-interleaving equivalence, the use of Hennessy's merge suffices to yield a finite axiomatization of the parallel composition operation.

Theorem 4 (Aceto, Fokkink, Ingolfsdottir and Luttik [8]). *Split-2 bisimilarity affords a finite ground-complete equational axiomatization over the language* CCS_H*.*

The above result hints at the possibility that non-interleaving equivalences like split-2 bisimilarity may be finitely axiomatizable using a single binary auxiliary operator. Whether a similar result holds true for standard bisimilarity remains open. We conjecture that the use of *two* binary auxiliary operators is 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 auxiliary operators proposed by Bergstra and Klop. Preliminary work on the confirmation of some form of this conjecture is under way [5].

4 The Influence of Time

So far in this paper we have presented, mostly negative, results on the finite axiomatizability of notions of bisimilarity over variations on Milner's CCS. Over the years, several extensions of CCS with, e.g., time, probabilities and priority have been presented in the literature. However, to the best of our knowledge, the question whether the aforementioned negative results hold true also for these extensions of classic process algebras like CCS has not received much attention in the research literature. In what follows, we discuss some impossibility results in the equational logic of timed bisimilarity over a fragment of Yi's timed CCS (TCCS) [52, 53], which is one of the best-known timed extension of Milner's CCS.

One of the first design decisions to be taken when developing a language for the description of timing-based behaviours is what structure to use to model time. Since we are interested in studying the equational theory of TCCS modulo bisimilarity, rather than selecting a single mathematical structure, such as the natural numbers or the non-negative rationals or reals, to represent time, we feel that it is more satisfying to adopt an axiomatic approach. We will therefore axiomatize a class of mathematical models of time for which our negative results hold. The non-negative rationals and the non-negative reals will be specific instances of our axiomatic framework, amongst others.

Following [41], we define a monoid $(X, +, 0)$ to be:

- *left-cancellative* iff $(x + y = x + z) \Rightarrow (y = z)$, and
- *anti-symmetric* iff $(x + y = 0) \Rightarrow (x = y = 0)$.

We define a partial order on X as $x \leq y$ iff $x + z = y$ for some $z \in X$. A *time domain* is a left-cancellative anti-symmetric monoid $(D, +, 0)$ such that $\leq$ is a total order. A time domain is *non-trivial* if D contains at least two elements. Note that every non-trivial time domain does not have a largest element, and is therefore infinite. A time domain has 0 as *cluster point* iff for each $d \in D$ such that $d \neq 0$ there is a $d' \in D$ such that $0 < d' < d$. In what follows, we assume that our time domain, denoted henceforth by D, is non-trivial and has 0 as cluster point.

Syntactically, we consider the language TCCS that is obtained by adding to the signature of CCS delay prefixing operators of the form $\epsilon(d).\_$, where d is a non-zero element of a time domain D. *In what follows, we only consider action prefixing operators of the form at and parallel composition without synchronization.*

The operational semantics for closed TCCS terms is based on two types of transition relations: $\xrightarrow{a}$ for action transitions and $\xrightarrow{d}$, where $d \in D$, for time-delay transitions. Action transitions are defined by the rules in Table 1, whereas the Plotkin-style rules defining delay transitions are given below.

$$\frac{}{\mathbf{0} \xrightarrow{d} \mathbf{0}} \qquad \frac{}{ax \xrightarrow{d} ax}$$

$$\frac{}{\epsilon(d).x \xrightarrow{d} x} \qquad \frac{}{\epsilon(d+e).x \xrightarrow{d} \epsilon(e).x} \qquad \frac{x \xrightarrow{e} y}{\epsilon(d).x \xrightarrow{d+e} y}$$

$$\frac{x_0 \xrightarrow{d} y_0 \quad x_1 \xrightarrow{d} y_1}{x_0 + x_1 \xrightarrow{\epsilon(d)} y_0 + y_1} \qquad \frac{x_0 \xrightarrow{d} y_0 \quad x_1 \xrightarrow{d} y_1}{x_0 \,||\, x_1 \xrightarrow{d} y_0 \,||\, y_1}$$

The notion of equivalence over TCCS that we are interested in is *timed bisimilarity*. This is defined exactly as in Definition 1, with the proviso that the meta-variable μ now ranges over time delays as well as actions. For example, $\mathbf{0}$ and $\epsilon(d).\mathbf{0}$ are timed bisimilar for each d, and so are a and $a + \epsilon(d).a$. On the other hand, a and $\epsilon(d).a$ are not timed bisimilar because the former term affords an a-labelled transition whereas the latter does not. (Intuitively, the latter term has to wait for d units of time before being able to perform the action a.)

It is natural to wonder whether TCCS affords a finite (ground-)complete axiomatization modulo timed bisimilarity. Before addressing this question, let us remark that one can take two different approaches to formalizing the syntax of TCCS in a term algebra.

1. The first approach is to use a single-sorted algebra with the only available sort representing processes. Then $\epsilon(d).\_$ is a set of unary operators, one for each $d \in D$.
2. The other approach is to take two different sorts, one for time and one for processes, denoted by $\mathbb{T}$ and $\mathbb{P}$, respectively. Then, $\epsilon(\_)$ is a single function symbol with arity $\mathbb{T} \times \mathbb{P} \to \mathbb{P}$.

If we decide to follow the first approach then, since our time domain is infinite, we are immediately led to observe that no finite collection of sound equations can prove all of the valid equalities of the form $\mathbf{0} \approx \epsilon(d).\mathbf{0}$. As a corollary of this observation, we obtain the following result.

Theorem 5 (Aceto, Ingolfsdottir and Mousavi [12]). *Timed bisimilarity over single-sorted* TCCS *has no finite (ground-)complete axiomatization.*

The lesson to be drawn from the above result is that, in the presence of an infinite time domain, when studying the equational theory of TCCS, it is much more natural to consider a two-sorted presentation of the calculus. However, even in a two-sorted setting, we are still faced with the inability of any finite sound axiom system to capture the interplay between interleaving and nondeterminism, which underlies Theorem 2. Therefore, by carefully adapting the proof of Moller's result, we obtain the following theorem.

Theorem 6 (Aceto, Ingolfsdottir and Mousavi [12]). *Timed bisimilarity over two-sorted* TCCS *has no finite (ground-)complete axiomatization.*

As shown by Bergstra and Klop in [19], in the setting of classic CCS and in the absence of synchronization one can finitely axiomatize bisimilarity by adding the left merge operator to the syntax for CCS. It is therefore natural to ask ourselves whether a finite axiomatization of timed bisimilarity over the fragment of TCCS we consider in this study can be obtained by adding some version of the left merge operator to the syntax for TCCS. Our order of business will now be to show that, unlike in the setting of Milner's CCS, even adding two variations on the left merge operator does not improve the situation with respect to axiomatizability.

We begin by noting that adding the classic left merge operator proposed by Bergstra and Klop to the syntax of TCCS does not lead to a finitely axiomatizable theory.

Theorem 7 (Aceto, Ingolfsdottir and Mousavi [12]). *Timed bisimilarity over two-sorted* TCCS *extended with Bergstra and Klop's left merge operator has no finite (ground-)complete axiomatization.*

Following the tradition of Bergstra and Klop, the left merge operator was given a timed semantics in [17] as follows.

$$\frac{x_0 \xrightarrow{a} y_0}{x_0 \mathbin{|\!\lfloor} x_1 \xrightarrow{a} y_0 \,||\, x_1} \qquad \frac{x_0 \xrightarrow{d} y_0 \quad x_1 \xrightarrow{d} y_1}{x_0 \mathbin{|\!\lfloor} x_1 \xrightarrow{d} y_0 \mathbin{|\!\lfloor} y_1}$$

This timed left merge operator enjoys most of the axioms for the classic left merge operator. However, this operator does not help in obtaining a finite ground-complete axiomatization for TCCS modulo bisimilarity either.

Theorem 8 (Aceto, Ingolfsdottir and Mousavi [12]). *Two-sorted* TCCS *extended with the timed left merge operator affords no finite (ground-)complete axiomatization modulo timed bisimilarity.*

Intuitively, the reason for the above-mentioned negative result is that the axiom

$$(ax) \mathbin{|\!\lfloor} y \approx a(x \,||\, y) \ \ ,$$

which is sound in the untimed setting, is in general unsound over TCCS. For example, consider the process $a \mathbin{|\!\lfloor} \epsilon(d).a$; after making a time delay of length d, it results in $a \mathbin{|\!\lfloor} a$, which is capable of performing two consecutive a-transitions. On the other hand, $a(\mathbf{0} \,||\, \epsilon(d).a)$ after a time delay of length d remains the same process and can only perform one a-transition, since the second a-transition still has to wait for d time units before becoming enabled.

However, the above axiom *is sound* for a class of TCCS processes whose behaviour, modulo timed bisimilarity, does not change by delaying. For instance, we have that

$$a \mathbin{|\!\lfloor} \left(\sum_{i=1}^{n} a \left(\sum_{j=1}^{i} a^j \right) \right) \leftrightarrow a \left(\sum_{i=1}^{n} a \left(\sum_{j=1}^{i} a^j \right) \right)$$

for each $n \geq 0$. However, no finite sound axiom system can prove all of the above equalities, and therefore cannot be complete.

Remark 2. All of the impossibility results presented in this section also hold for conditional equations of the form $P \Rightarrow t \approx u$, where P is an arbitrary predicate over the time domain, and t, u are TCCS terms.

In the case of two-sorted TCCS, our proofs make use of the fact that the time domain has 0 as a cluster point. However, we conjecture that discrete-time TCCS, or its extension with (timed) left merge, is not finitely axiomatizable modulo timed bisimilarity either. Work on a proof of this conjecture is ongoing.

References

1. L. Aceto. Deriving complete inference systems for a class of GSOS languages generating regular behaviours. In B. Jonsson and J. Parrow, editors, *Proceedings CONCUR 94,* Uppsala, Sweden, volume 836 of *Lecture Notes in Computer Science*, pages 449–464. Springer-Verlag, 1994.
2. L. Aceto. On "Axiomatising finite concurrent processes". *SIAM J. Comput.*, 23(4):852–863, 1994.
3. L. Aceto, B. Bloom, and F. Vaandrager. Turning SOS rules into equations. *Information and Computation*, 111(1):1–52, 1994.
4. L. Aceto, T. Chen, W. Fokkink, and A. Ingolfsdottir. On the axiomatizability of priority. In Bugliesi et al. [26], pages 480–491.
5. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. Are two binary operators necessary to finitely axiomatize parallel composition? In preparation.
6. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. CCS with Hennessy's merge has no finite equational axiomatization. *Theoretical Comput. Sci.*, 330(3):377–405, 2005.
7. L. Aceto, 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: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday*, volume 3838 of *Lecture Notes in Computer Science*, pages 338–367. Springer-Verlag, 2005.
8. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. Split-2 bisimilarity has a finite axiomatization over CCS with Hennessy's merge. *Logical Methods in Computer Science*, 1(1):1–12, 2005.
9. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. A finite equational base for CCS with left merge and communication merge. In Bugliesi et al. [26], pages 492–503.
10. L. Aceto, W. Fokkink, A. Ingolfsdottir, and S. Nain. Bisimilarity is not finitely based over BPA with interrupt. *Theoretical Comput. Sci.*, 366(1–2):60–81, 2006.
11. L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, pages 197–292. North-Holland, 2001.
12. L. Aceto, A. Ingolfsdottir, and M. Mousavi. Impossibility results for the equational theory of timed CCS. In *Proceedings of the 2nd Conference on Algebra and Coalgebra in Computer Science*, Lecture Notes in Computer Science, Bergen, Norway, 2007. Springer-Verlag.
13. D. Austry and G. Boudol. Algèbre de processus et synchronisations. *Theoretical Comput. Sci.*, 30(1):91–131, 1984.
14. J. Baeten. A brief history of process algebra. *Theoretical Comput. Sci.*, 335(2–3):131–146, 2005.
15. J. Baeten and E. de Vink. Axiomatizing GSOS with termination. In H. Alt and A. Ferreira, editors, *Proceedings of STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science, Antibes-Juan les Pins, France, March 14–16, 2002*, volume 2285 of *Lecture Notes in Computer Science*, pages 583–595. Springer-Verlag, 2002.
16. J. Baeten and J. W. Klop, editors. *Proceedings CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.
17. J. Baeten and C. A. Middelburg. *Process Algebra with Timing*. Monographs in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2002.

18. J. Baeten and P. Weijland. *Process Algebra*. Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.
19. J. Bergstra and J. W. Klop. Fixed point semantics in process algebras. Report IW 206, Mathematisch Centrum, Amsterdam, 1982.
20. J. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1/3):109–137, 1984.
21. S. L. Bloom and Z. Ésik. Nonfinite axiomatizability of shuffle inequalities. In P. D. Mosses, M. Nielsen, and M. I. Schwartzbach, editors, *Proceedings of TAPSOFT'95: Theory and Practice of Software Development, 6th International Joint Conference CAAP/FASE, Aarhus, Denmark, May 22–26, 1995*, volume 915 of *Lecture Notes in Computer Science*, pages 318–333. Springer-Verlag, 1995.
22. S. L. Bloom and Z. Ésik. Free shuffle algebras in language varieties. *Theoret. Comput. Sci.*, 163(1-2):55–98, 1996.
23. S. L. Bloom and Z. Ésik. Axiomatizing shuffle and concatenation in languages. *Inform. and Comput.*, 139(1):62–91, 1997.
24. S. L. Bloom and Z. Ésik. Varieties generated by languages with poset operations. *Math. Structures Comput. Sci.*, 7(6):701–713, 1997.
25. S. L. Bloom and Z. Ésik. Shuffle binoids. *RAIRO Inform. Théor. Appl.*, 32(4-6):175–198, 1998.
26. M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors. *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.
27. R. de Simone. Higher-level synchronising devices in Meije–SCCS. *Theoretical Comput. Sci.*, 37:245–267, 1985.
28. Z. Ésik and M. Bertol. Nonfinite axiomatizability of the equational theory of shuffle. *Acta Inform.*, 35(6):505–539, 1998.
29. Z. Ésik and S. Okawa. Series and parallel operations on pomsets. In *Proceedings of Foundations of Software Technology and Theoretical Computer Science (Chennai, 1999)*, volume 1738 of *Lecture Notes in Comput. Sci.*, pages 316–328. Springer-Verlag, Berlin, 1999.
30. W. Fokkink. *Introduction to Process Algebra*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000.
31. W. Fokkink and B. Luttik. An omega-complete equational specification of interleaving. In U. Montanari, J. Rolinn, and E. Welzl, editors, *Proceedings 27th Colloquium on Automata, Languages and Programming—ICALP'00, Geneva*, volume 1853 of *Lecture Notes in Computer Science*, pages 729–743. Springer-Verlag, July 2000.
32. J. L. Gischer. The equational theory of pomsets. *Theoretical Comput. Sci.*, 61:199–224, 1988.
33. R. van Glabbeek and F. Vaandrager. Petri net models for algebraic theories of concurrency. In J. de Bakker, A. Nijman, and P. Treleaven, editors, *Proceedings PARLE conference,* Eindhoven, *Vol. II (Parallel Languages)*, volume 259 of *Lecture Notes in Computer Science*, pages 224–242. Springer-Verlag, 1987.
34. J. Grabowski. On partial languages. *Fundamenta Informaticae*, IV(2):427–498, 1981.
35. J. F. Groote. A new strategy for proving ω–completeness with applications in process algebra. In Baeten and Klop [16], pages 314–331.
36. M. Hennessy. On the relationship between time and interleaving. Preprint, CMA, Centre de Mathématiques Appliquées, Ecole des Mines de Paris, 1981.

37. M. Hennessy. *Algebraic Theory of Processes.* MIT Press, Cambridge, Massachusetts, 1988.
38. M. Hennessy. Axiomatising finite concurrent processes. *SIAM J. Comput.*, 17(5):997–1017, 1988.
39. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.
40. C. Hoare. *Communicating Sequential Processes.* Prentice-Hall International, Englewood Cliffs, 1985.
41. A. Jeffrey, S. Schneider, and F. Vaandrager. A comparison of additivity axioms in timed transition systems. Report CS-R9366, CWI, Amsterdam, 1993.
42. R. Keller. Formal verification of parallel programs. *Commun. ACM*, 19(7):371–384, 1976.
43. R. Milner. Flowgraphs and flow algebras. *J. ACM*, 26(4):794–818, 1979.
44. R. Milner. *Communication and Concurrency.* Prentice-Hall International, Englewood Cliffs, 1989.
45. 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.
46. F. Moller. The importance of the left merge operator in process algebras. In M. Paterson, editor, *Proceedings* 17^{th} *ICALP,* Warwick, volume 443 of *Lecture Notes in Computer Science*, pages 752–764. Springer-Verlag, July 1990.
47. F. Moller. The nonexistence of finite axiomatisations for CCS congruences. In *Proceedings* 5^{th} *Annual Symposium on Logic in Computer Science,* Philadelphia, USA, pages 142–153. IEEE Computer Society Press, 1990.
48. D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, Karlsruhe, Germany, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.
49. G. D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60–61:17–139, 2004.
50. V. Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986.
51. S. T. Tschantz. Languages under concatenation and shuffling. *Mathematical Structures in Computer Science*, 4(4):505–511, 1994.
52. W. Yi. Real-time behaviour of asynchronous agents. In Baeten and Klop [16], pages 502–520.
53. W. Yi. *A Calculus of Real Time Systems.* PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 1991.

Recent BRICS Report Series Publications

RS-07-11 Luca Aceto and Anna Ingólfsdóttir. *The Saga of the Axiomatization of Parallel Composition*. June 2007. 15 pp. To appear in the Proceedings of CONCUR 2007, the 18th International Conference on Concurrency Theory (Lisbon, Portugal, September 4–7, 2007), Lecture Notes in Computer Science, Springer-Verlag, 2007.

RS-07-10 Claus Brabrand, Robert Giegerich, and Anders Møller. *Analyzing Ambiguity of Context-Free Grammars*. May 2007. 17 pp. Full version of paper presented at CIAA '07.

RS-07-9 Janus Dam Nielsen and Michael I. Schwartzbach. *The SMCL Language Specification*. March 2007.

RS-07-8 Olivier Danvy and Kevin Millikin. *A Simple Application of Lightweight Fusion to Proving the Equivalence of Abstract Machines*. March 2007. ii+6 pp.

RS-07-7 Olivier Danvy and Kevin Millikin. *Refunctionalization at Work*. March 2007. ii+16 pp. Invited talk at the 8th International Conference on Mathematics of Program Construction, MPC '06.

RS-07-6 Olivier Danvy, Kevin Millikin, and Lasse R. Nielsen. *On One-Pass CPS Transformations*. March 2007. ii+19 pp. Theoretical Pearl to appear in the *Journal of Functional Programming*. Revised version of BRICS RS-02-3.

RS-07-5 Luca Aceto, Silvio Capobianco, and Anna Ingólfsdóttir. *On the Existence of a Finite Base for Complete Trace Equivalence over BPA with Interrupt*. February 2007. 26 pp.

RS-07-4 Kristian Støvring and Søren B. Lassen. *A Complete, Co-Inductive Syntactic Theory of Sequential Control and State*. February 2007. 36 pp. Appears in the proceedings of POPL 2007, p. 161–172.

RS-07-3 Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir. *Ready To Preorder: Get Your BCCSP Axiomatization for Free!* February 2007. 37 pp.

RS-07-2 Luca Aceto and Anna Ingólfsdóttir. *Characteristic Formulae: From Automata to Logic*. January 2007. 18 pp.

RS-07-1 Daniel Andersson. *HIROIMONO is NP-complete*. January 2007. 8 pp.