

Lifting Non-Finite Axiomatizability Results to Extensions of Process Algebras ^{*}

Luca Aceto¹, Wan Fokkink², Anna Ingólfssdóttir¹, and MohammadReza Mousavi³

¹ School of Computer Science, Reykjavík University,
Kringlan 1, IS-103, Reykjavík, Iceland

² Department of Computer Science, Vrije Universiteit Amsterdam,
NL-1081HV, The Netherlands

³ Department of Computer Science, Eindhoven University of Technology,
NL-5600MB Eindhoven, The Netherlands

Abstract. This paper presents a general technique for obtaining new results pertaining to the non-finite axiomatizability of behavioural (pre)congruences over process algebras from old ones. The proposed technique is based on a variation on the classic idea of reduction mappings. In this setting, such reductions are translations between languages that preserve sound (in)equations and (in)equational proofs over the source language, and reflect families of (in)equations responsible for the non-finite axiomatizability of the target language. The proposed technique is applied to obtain a number of new non-finite axiomatizability theorems in process algebra via reduction to Moller’s celebrated non-finite axiomatizability result for CCS. The limitations of the reduction technique are also studied. In particular, it is shown that prebisimilarity is not finitely based over CCS with the divergent process Ω , but that this result cannot be proved by a reduction to the non-finite axiomatizability of CCS modulo bisimilarity.

1 Introduction

Process algebras, such as the Algebra of Communicating Processes (ACP) [12], the Calculus of Communicating Systems (CCS) [28] and Communicating Sequential Processes (CSP) [23], are prototype languages for the description of reactive systems. Since these languages may be used for describing specifications of process behaviour as well as their implementations, an important ingredient in their theory is a notion of equivalence or approximation between process descriptions. The equivalence between two terms in a process algebra indicates that, although possibly syntactically different, these terms describe essentially the same behaviour. Behavioural equivalences are therefore typically used in the theory of process algebras as the formal yardstick by means of which one can establish the correctness of an implementation with respect to a given specification.

In the light of the algebraic nature of process algebras, a natural question is whether the chosen notion of behavioural equivalence or approximation can be axiomatized by means of a finite, or at least finitely describable, collection of equations. An equational axiomatization characterizes in a nutshell all the valid equivalences that hold in the model of concurrent computation under study, and it is conceptually very satisfactory, as well as aesthetically pleasing, to be able to describe in a purely syntactic fashion all the sound semantic equivalences. Such a syntactic characterization allows one to compare notions of equivalence that may have been defined in very different styles simply by looking at the equations that those equivalences satisfy. Finally, an axiomatization of the relevant notion of equivalence may form the basis for verification tools based on theorem-proving technology [14, 18].

From the theoretical point of view, a fundamental question in the study of algebras of processes is whether they afford a finite (in)equational axiomatization. The first negative results concerning

^{*} The work of Aceto, Ingólfssdóttir and Mousavi has been partially supported by the projects “The Equational Logic of Parallel Processes” (nr. 060013021), “A Unifying Framework for Operational Semantics” (nr. 070030041) and “New Developments in Operational Semantics” (nr. 080039021) of the Icelandic Research Fund.

finite axiomatizability of process algebras go back to the Ph.D. thesis of Faron Moller [29], in which he showed that strong bisimilarity is not finitely based over CCS and over ACP without the left-merge operator. Since then, several other non-finite axiomatizability results have been obtained for a wide collection of very basic process algebras—see, e.g., [5] for a survey of such results.

In general, results concerning (non-)finite axiomatizability are very vulnerable to small changes in, and extensions of, the formalism under study. The addition of a single operator to a non-finitely axiomatizable formalism may make it finitely axiomatizable (e.g., adding the left-merge operator to the synchronization-free subset of CCS [13]). Conversely, the addition of a single operator may ruin the finite axiomatizability of a calculus (e.g., adding parallel composition to the sequential subset of CCS [28, 30]). Also, apparently simple changes to the semantics of process calculi, e.g., adding aspects such as timing, may ruin the (non-)finite axiomatizability results and make their proofs obsolete (e.g., adding timing to synchronization-free CCS with left merge makes it non-finitely axiomatizable, as shown in [9]). Furthermore, proofs of non-finite axiomatizability results in the concurrency-theory literature are extremely delicate and error-prone; they are often rather long, and involve several levels of structural induction and case distinction on the structure of the terms appearing in the equations. Hence, we believe that it would be useful to find some general techniques that can be used to prove non-finite axiomatizability results. Such a general theory would allow one to relate non-finite axiomatizability theorems for different formalisms, and spare researchers (some of) the delicate technical analysis needed to adapt the proofs of such results. Despite some initial proposals, like the one in [3], it is fair to say that such a general theory is missing to date.

In this paper, we present a meta-theorem offering a general technique that can be used to prove non-finite axiomatizability results, and present some of its applications within concurrency theory. In this meta-theorem, we give sufficient criteria to obtain new non-finite axiomatizability results from known ones. The proposed technique is based on a variation on the classic idea of reduction mappings, which underlies the proofs of many classic undecidability results in computability theory and of lower bounds in complexity theory—see, e.g., [37] for a textbook presentation.

The basic idea underlying the reduction-based method we propose in this study is as follows. Assume that we have a language \mathcal{L}_o that we know is not finitely axiomatizable modulo some (pre)congruence \lesssim_o . Typically, such a negative result is shown by exhibiting an infinite family E of sound (in)equations, which no finite sound axiom system can prove. Intuitively, E encapsulates one of the reasons why the (pre)congruence \lesssim_o is hard to axiomatize finitely over \mathcal{L}_o . Suppose now that we wish to prove that some language \mathcal{L}_e is also not finitely axiomatizable modulo some (pre)congruence \lesssim_e . According to the method we propose in this paper, to do so it suffices only to give a mapping from \mathcal{L}_e to \mathcal{L}_o (which we call a *reduction*) that preserves sound (in)equations and (in)equational proofs over the source language, and reflects the family of (in)equations E responsible for the non-finite axiomatizability of the target language. Intuitively, the existence of such a reduction witnesses the fact that the “bad” collection of (in)equations E is also present, in some form, in the source language \mathcal{L}_e , and that if it could be proved from a finite collection of sound (in)equations over the source language, then it could also be shown to hold by means of a finite sound axiom system over the target language. Since, by our assumption, no finite sound axiom system over \mathcal{L}_o can prove E , the existence of the reduction allows us to conclude that \mathcal{L}_e is also not finitely axiomatizable modulo \lesssim_e .

We show the applicability of our reduction-based technique by obtaining several, to our knowledge novel, non-finite axiomatizability results for timed and stochastic process algebras. Namely, we prove non-finite axiomatizability results for the following process algebras modulo their corresponding notions of (pre)congruence:

1. Discrete-time CCS modulo timed bisimilarity [38],
2. Temporal CCS modulo timed bisimilarity [32],
3. ATP modulo timed bisimilarity [35],
4. $TACS^{UT}$ modulo faster-than preorder [24],
5. $TACS^{LT}$ modulo MT-preorder [25],

6. TACS modulo urgent timed bisimilarity [26] and
7. IMC modulo strong Markovian bisimilarity [22].

Our meta-theorem is algebraic in nature and does not rely on any assumption on the specification of the semantics of the languages to which it can be applied. We believe that the general result we present in this study paves the way for several other meta-theorems, once further assumptions are made regarding the underlying models. For example, we expect that, by committing to SOS rules in the style of Plotkin [36] as means of defining the semantics of the formalism, one may invoke existing meta-theorems from the theory of SOS (see, e.g., [10]) to provide sufficient syntactic conditions guaranteeing that the premises of our algebraic meta-theorem hold. A promising future direction of research is to study whether one can apply our meta-theorem in conservative and orthogonal language extensions (in the sense of [11, 17] and [34], respectively).

The paper is organized as follows. In Section 2, we review some preliminary definitions from universal algebra. Section 3 presents our reduction-based technique for proving non-finite axiomatizability results. In Section 4 we apply our approach to obtain seven new non-finite axiomatizability results. In Section 5, we illustrate the limitations of our proof methodology by presenting a non-finite axiomatizability result that cannot be proved using the strategy we employed to obtain the results in Section 4. These limitations can provide sources of inspiration for future improvements on our techniques. Finally, Section 6 concludes the paper and presents some directions for future and ongoing research.

2 Preliminaries

We begin by recalling some basic notions from universal algebra that will be used throughout the paper. We refer the interested reader to, e.g., [21] for more information.

A signature Σ is a set of function symbols f, g, \dots with fixed arities. A function symbol of arity zero is often called a *constant (symbol)*. Given a signature Σ and a set of variables V , terms $t, u, \dots \in \mathcal{T}(\Sigma)$ are constructed inductively (from function symbols and variables) while respecting the arities of the function symbols. (In what follows, whenever we write a term $f(t_1, \dots, t_n)$ we tacitly assume that the arity of f is n .) Closed terms $p, q, \dots \in \mathcal{C}(\Sigma)$ are terms that do not contain variables. We write \equiv for syntactic equality over terms.

A *precongruence* \preceq over $\mathcal{C}(\Sigma)$ is a substitutive preorder over $\mathcal{C}(\Sigma)$ —that is, a preorder over $\mathcal{C}(\Sigma)$ that is preserved by all the function symbols in Σ . A *congruence* \sim over $\mathcal{C}(\Sigma)$ is a substitutive equivalence relation. Each precongruence \preceq over $\mathcal{C}(\Sigma)$ induces a congruence \sim thus: $p \sim q$ iff $p \preceq q \preceq p$.

A (closed) substitution maps variables in V to (closed) terms. For every term t and substitution σ , the term $\sigma(t)$ is obtained by replacing every occurrence of a variable x in t by $\sigma(x)$. Note that $\sigma(t)$ is closed if σ is a closed substitution. We write $[t_1/x_1, \dots, t_n/x_n]$, where the x_i ($1 \leq i \leq n$) are distinct variables, for the substitution mapping each variable x_i to t_i , and acting like the identity function on all the other variables.

Given a relation R over closed terms, for open terms t and u , we define $t R u$ if $\sigma(t) R \sigma(u)$ for each closed substitution σ .

Consider a signature Σ . A set E of equations $t = t'$, where $t, t' \in \mathcal{T}(\Sigma)$, is called an *axiomatization (over $\mathcal{T}(\Sigma)$)*. We write $E \vdash t = t'$ when $t = t'$ is derivable from E by the following set of inference rules.

$$\begin{array}{c}
 \text{(refl)} \frac{}{E \vdash t = t} \quad \text{(trans)} \frac{E \vdash t_0 = t_1 \quad E \vdash t_1 = t_2}{E \vdash t_0 = t_2} \\
 \\
 \text{(cong)} \frac{E \vdash t_1 = t'_1 \quad \dots \quad E \vdash t_n = t'_n}{E \vdash f(t_1, \dots, t_n) = f(t'_1, \dots, t'_n)} \quad \text{(E)} \frac{}{E \vdash \sigma(t) = \sigma(t')} \quad t = t' \in E
 \end{array}$$

(Deduction rule **(cong)** is a rule schema with one instance for each function symbol f in the signature Σ .) For axiomatizations E and E' , we write $E' \vdash E$ when $E' \vdash t = u$ for each $t = u \in E$.

Above, we intentionally did not include the inference rule for symmetry, i.e.,

$$(\text{symm}) \frac{E \vdash t = t'}{E \vdash t' = t} .$$

Excluding **(symm)** does not restrict the applicability of our results by any measure. Any set of equations can be closed under symmetry by simply adding to it a symmetric copy of each equation, and this transformation preserves finiteness. (In what follows, we shall tacitly assume that each equational axiomatization is closed with respect to symmetry.) Furthermore, the omission of the rule for symmetry allows us to deal with axiomatizations for precongruences, which are not necessarily symmetric relations. When working with precongruences, our axiomatizations consist of inequations $t \leq t'$ between terms.

Given a congruence $\sim \subseteq \mathcal{T}(\Sigma) \times \mathcal{T}(\Sigma)$, an equation $t = t'$ is *sound* modulo \sim when $t \sim t'$. An axiomatization is *sound* modulo \sim if each of its equations is sound modulo \sim . An axiomatization E is *complete* modulo \sim if for each sound equation $t = t'$, it holds that $E \vdash t = t'$. E is *ground-complete* modulo \sim if for each *closed* sound equation $p = q$, it holds that $E \vdash p = q$. We say that \sim is *finitely based* over $\mathcal{T}(\Sigma)$ if there is a finite, sound and complete axiomatization for $\mathcal{T}(\Sigma)$ modulo \sim . Similar definitions apply to precongruences and inequational axiomatizations.

3 The Reduction Theorem

Our aim in this section will be to present a general result that will allow us to lift non-finite axiomatizability results from one process algebra to another. Throughout this section, we fix two signatures Σ_o and Σ_e , a common set of variables V and two precongruences \preceq_o and \preceq_e over $\mathcal{T}(\Sigma_o)$ and $\mathcal{T}(\Sigma_e)$, respectively. Intuitively, the signature Σ_o stands for the collection of operations in an original process language for which we already have a non-finite axiomatizability result modulo the precongruence \preceq_o . On the other hand, the signature Σ_e stands for the collection of operations in an extended process language for which we intend to prove a non-finite axiomatizability result modulo the precongruence \preceq_e . Since a congruence is a symmetric precongruence, all the results we present in the remainder of this section apply equally well when any of \preceq_o and \preceq_e is a congruence relation.

Consider a mapping $\hat{\cdot} : \mathcal{T}(\Sigma_e) \rightarrow \mathcal{T}(\Sigma_o)$. For an axiomatization E over $\mathcal{T}(\Sigma_e)$, we define the axiomatization \hat{E} over $\mathcal{T}(\Sigma_o)$ to be $\{\hat{t} \leq \hat{u} \mid t \leq u \in E\}$.

Definition 1. A function $\hat{\cdot} : \mathcal{T}(\Sigma_e) \rightarrow \mathcal{T}(\Sigma_o)$ is a *reduction* from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$, when for all $t, u \in \mathcal{T}(\Sigma_e)$,

1. $t \preceq_e u \Rightarrow \hat{t} \preceq_o \hat{u}$ (that is, $\hat{\cdot}$ preserves sound inequations), and
2. $E \vdash t \leq u \Rightarrow \hat{E} \vdash \hat{t} \leq \hat{u}$, for each axiomatization E over $\mathcal{T}(\Sigma_e)$ (that is, $\hat{\cdot}$ preserves provability).

Definition 2. Let E be an axiomatization over $\mathcal{T}(\Sigma_o)$. A reduction $\hat{\cdot}$ is *E-reflecting*, when for each $t \leq u \in E$, there exists an inequation $t' \leq u'$ over $\mathcal{T}(\Sigma_e)$ that is sound modulo \preceq_e such that $\hat{t}' \equiv t$ and $\hat{u}' \equiv u$. A reduction $\hat{\cdot}$ is called *ground E-reflecting* if for each closed inequation $p \leq q \in E$, there exists a closed inequation $p' \leq q'$ on $\mathcal{T}(\Sigma_e)$ that is sound modulo \preceq_e such that $\hat{p}' \equiv p$ and $\hat{q}' \equiv q$.

We are now ready to state the general tool that we shall use in this paper to lift non-finite axiomatizability results from $\mathcal{T}(\Sigma_o)$ modulo \preceq_o to $\mathcal{T}(\Sigma_e)$ modulo \preceq_e .

Theorem 1. Assume that there is a set of inequations E over $\mathcal{T}(\Sigma_o)$ that is sound modulo \preceq_o and that is not provable from any finite sound axiomatization over $\mathcal{T}(\Sigma_o)$. If there exists an *E-reflecting* reduction from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$, then \preceq_e is not finitely based over $\mathcal{T}(\Sigma_e)$.

Proof. Assume, towards a contradiction, that some finite axiomatization F is sound and complete for $\mathcal{T}(\Sigma_e)$ modulo \lesssim_e . Let $\widehat{\cdot}$ be the E -reflecting reduction given by the proviso of the theorem, and let E' be the corresponding set of sound inequations (modulo \lesssim_e) over $\mathcal{T}(\Sigma_e)$ such that $\widehat{E'} = E$. It follows from the soundness of E' and the completeness of F that $F \vdash E'$. So by item 2 of Definition 1, $\widehat{F} \vdash t \equiv \widehat{t'} \leq \widehat{u'} \equiv u$, for each $t \leq u \in E$. Furthermore, by item 1 of Definition 1 and the soundness of F with respect to \lesssim_e , \widehat{F} is sound modulo \lesssim_o . Thus, there exists a finite sound axiomatization for $\mathcal{T}(\Sigma_o)$ modulo \lesssim_o , namely \widehat{F} , from which E can be derived. This contradicts the hypothesis of the theorem. \square

Remark 1. Let E be a set of inequations over $\mathcal{T}(\Sigma_o)$ that is sound modulo \lesssim_o and that is not provable from any finite sound axiomatization over $\mathcal{T}(\Sigma_o)$. Suppose that $\widehat{\cdot}$ is an E -reflecting reduction from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$. Let E' be the collection of sound inequations over $\mathcal{T}(\Sigma_e)$ such that $\widehat{E'} = E$. The proof of Theorem 1 yields that E' is not provable from any finite axiomatization over $\mathcal{T}(\Sigma_e)$ that is sound modulo \lesssim_e .

The above theorem gives us a general technique to lift non-finite axiomatizability results from a language $\mathcal{T}(\Sigma_o)$ modulo \lesssim_o to a language $\mathcal{T}(\Sigma_e)$ modulo \lesssim_e . Indeed, suppose that we know that a precongruence \lesssim_o is not finitely based over $\mathcal{T}(\Sigma_o)$. Typically, such a negative result is shown by exhibiting an infinite collection E of sound inequations that cannot be proved from any finite sound axiomatization over Σ_o . (See, e.g., [2, 4–6, 9, 15, 16, 29, 31] and the references therein.) In the light of the above theorem, to show that \lesssim_e is not finitely based over $\mathcal{T}(\Sigma_e)$ it suffices only to exhibit an E -reflecting reduction from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$.

As the examples we present in Section 4 will show, Theorem 1, albeit not technically complex, is widely applicable. In all our applications of Theorem 1, the reduction from Σ_e to Σ_o is defined inductively on the structure of terms. Since such “structural” reductions play an important role in the remainder of the paper, we now proceed to define them precisely and to prove a very useful property such reductions afford.

Definition 3. A mapping $\widehat{\cdot} : \mathcal{T}(\Sigma_e) \rightarrow \mathcal{T}(\Sigma_o)$ is structural if

1. it is the identity function over variables, i.e., $\widehat{x} \equiv x$ for each $x \in V$,
2. it does not introduce new variables, i.e., $\text{vars}(f(\widehat{x_1}, \dots, \widehat{x_n})) \subseteq \{x_1, \dots, x_n\}$, for each $f \in \Sigma_e$ and sequence of distinct $x_1, \dots, x_n \in V$, and
3. it is defined compositionally, i.e., $f(\widehat{t_1}, \dots, \widehat{t_n}) \equiv f(\widehat{x_1}, \dots, \widehat{x_n}) [\widehat{t_1}/x_1, \dots, \widehat{t_n}/x_n]$, for each $f \in \Sigma_e$, and sequences of distinct $x_1, \dots, x_n \in V$ and of $t_1, \dots, t_n \in \mathcal{T}(\Sigma_e)$.

We note that $f(\widehat{y_1}, \dots, \widehat{y_n}) \equiv f(\widehat{x_1}, \dots, \widehat{x_n}) [y_1/x_1, \dots, y_n/x_n]$, by conditions 1 and 3 in the definition above. Moreover, it is easy to see that, whenever $\widehat{\cdot}$ is structural, $\text{vars}(\widehat{t}) \subseteq \text{vars}(t)$, for each $t \in \mathcal{T}(\Sigma_e)$.

Structural mappings afford the following crucial property, which describes their interplay with substitutions and is akin to the classic “substitution lemma” from denotational semantics—see, e.g., [19]. In the statement of the subsequent lemma, for each substitution σ over Σ_e we use $\widehat{\sigma}$ to denote the substitution over Σ_o mapping each variable x to $\widehat{\sigma(x)}$.

Lemma 1. Let $\widehat{\cdot} : \mathcal{T}(\Sigma_e) \rightarrow \mathcal{T}(\Sigma_o)$ be a structural mapping. Then $\widehat{\sigma(t)} \equiv \widehat{\sigma}(\widehat{t})$, for each term $t \in \mathcal{T}(\Sigma_e)$ and each substitution σ over Σ_e .

Proof. By structural induction on t . Condition 1 in Definition 3 is used to handle the case $t \equiv x$ for some variable x . The case $t \equiv f(t_1, \dots, t_n)$ for some $f \in \Sigma_e$ and $t_1, \dots, t_n \in \mathcal{T}(\Sigma_e)$ is dealt with using induction and conditions 2–3. \square

Remark 2. Note that the above lemma would fail if structural substitutions were not required to satisfy condition 2 of Definition 3. To see this, consider, for instance, the term $t \equiv f(x)$, and assume that $\widehat{f(x)} \equiv x + y$. Then, since $\widehat{\cdot}$ satisfies the third condition in Definition 3,

$$\widehat{\sigma(t)} \equiv \widehat{f(x)}[\widehat{\sigma(x)}/x] \equiv (x + y)[\widehat{\sigma(x)}/x] \equiv \widehat{\sigma(x)} + y .$$

On the other hand,

$$\widehat{\sigma}(\widehat{t}) \equiv \widehat{\sigma}(x + y) \equiv \widehat{\sigma}(x) + \widehat{\sigma}(y) .$$

If $\widehat{\sigma}(y)$ is different from y , then the terms $\widehat{\sigma}(x) + y$ and $\widehat{\sigma}(x) + \widehat{\sigma}(y)$ are not equal.

The following theorem shows that, if the reduction is structural, one can dispense with proving item 2 of Definition 1. Since each reduction we consider in this paper is structural, this result eases our applications of Theorem 1 considerably.

Theorem 2. *A structural mapping satisfies item 2 of Definition 1.*

Proof. By an induction on the depth of the proof of the statement $E \vdash t = u$. We distinguish cases based on the last inference rule applied to derive $t = u$ from E . The case for **(refl)** is trivial. The case for **(trans)** follows from the induction hypothesis. The case for **(cong)** is handled using condition 3 in Definition 3. Finally, the case for **(E)** follows easily from Lemma 1 using the definition of \widehat{E} . \square

Ground completeness. If the collection of equations E mentioned in the statement of Theorem 1 is closed, then one can prove impossibility of a finite ground-complete axiomatization of \lesssim_e over $\mathcal{T}(\Sigma_e)$, which is a stronger result than Theorem 1.

Theorem 3. *Assume that there is a set of closed equations E that is sound modulo \lesssim_o , and that is not provable from any finite axiomatization over $\mathcal{T}(\Sigma_o)$ that is sound modulo \lesssim_o . If there exists a ground E -reflecting reduction from Σ_e to Σ_o , then there exists no sound and ground-complete finite axiomatization for \lesssim_e over $\mathcal{T}(\Sigma_e)$.*

Proof. The proof is almost identical to the proof of Theorem 1. All appearances of “complete” need to be replaced by “ground-complete”, all terms need to be replaced by closed terms, and “ E -reflecting” is to be replaced by “ground E -reflecting”. \square

For structural reductions whose source is a language over a signature that contains at least one constant, in order to apply Theorem 3 it suffices to show that the reduction is E -reflecting by the following theorem. Thus, if the collection of equations E is closed and the reduction is structural, one can readily obtain impossibility of a finite ground-complete axiomatization without any further work (by showing that the premises of Theorem 1 hold).

Theorem 4. *An E -reflecting structural reduction $\widehat{\cdot}$ is also ground E -reflecting, provided that the signature Σ_e contains at least one constant symbol.*

Proof. We need to show that if $p \leq q \in E$ is sound modulo \lesssim_o , then there exist closed terms $p', q' \in \mathcal{C}(\Sigma_e)$ such that $p' \lesssim_e q'$, $\widehat{p}' \equiv p$ and $\widehat{q}' \equiv q$. To this end, assume that $p \leq q \in E$. The reduction $\widehat{\cdot}$ is E -reflecting by the proviso of the theorem, and thus there exist two (possibly open) terms $t, u \in \mathcal{T}(\Sigma_e)$ such that $t \lesssim_e u$, $\widehat{t} \equiv p$ and $\widehat{u} \equiv q$. Take an arbitrary closed substitution $\sigma : V \rightarrow \mathcal{C}(\Sigma_e)$. (Such a substitution exists because, by the proviso of the theorem, Σ_e contains at least one constant symbol.) It holds that $\sigma(t) \lesssim_e \sigma(u)$. If we show that $\widehat{\sigma(t)} \equiv \widehat{t} \equiv p$ and $\widehat{\sigma(u)} \equiv \widehat{u} \equiv q$, then the theorem follows.

To see that $\widehat{\sigma(t)} \equiv \widehat{t} \equiv p$, simply observe that, using Lemma 1 and the assumption that p is closed,

$$\widehat{\sigma(t)} \equiv \widehat{\sigma}(\widehat{t}) \equiv \widehat{\sigma}(p) \equiv p .$$

Since $\widehat{\sigma(u)} \equiv \widehat{u} \equiv q$ also holds by a similar argument, the proof is complete. \square

Remark 3. The proviso that the signature Σ_e contains at least one constant symbol is necessary for the previous theorem to hold. Consider, for instance, a signature Σ_e that only contains a function symbol f of arity one. Let Σ_o consist only of the constant symbol c . As congruences \sim_e and \sim_o , consider the universal relations over the sets of Σ_e - and Σ_o -terms, respectively. Let E be the axiomatization consisting only of the equation $c = c$.

Define the mapping $\widehat{\cdot}$ to be the identity function over variables and let $\widehat{f(t)} = c$ for each term t over the signature Σ_e . We have that:

1. $\widehat{\cdot}$ is E -reflecting and
2. $\widehat{\cdot}$ is structural. (This is because $\widehat{f(t)} = c = \widehat{f(x)}[\widehat{t}/x]$, for each variable x and term t .)

However, $\widehat{\cdot}$ is not ground E -reflecting because there are no closed terms over Σ_e .

The set of basic equations that we shall use throughout the rest of this paper in our applications of Theorem 1 is closed and, furthermore, all our reductions are structural; thus, all the impossibility results we present in the subsequent section hold for ground-complete as well as complete axiomatizations.

4 Applications

In this section, we take a well-known non-finite axiomatizability result in the setting of process algebra due to Moller [29, 30], and use Theorem 1 to establish other, to the best of our knowledge novel, non-finite axiomatizability results for several notions of behavioural (pre)congruences over other process algebras. A brief comparison between the full proof of the original result in [29, 30] and those based on Theorem 1 presented in the remainder of this section reveals that our proofs are substantially more concise and simpler than direct proofs. This is despite the fact that the calculi and notions of (pre)congruence treated henceforth are more sophisticated than the ones treated in [29, 30].

4.1 Basic Theory

Consider the subset of CCS [28] with the following syntax.

$$P ::= \mathbf{0} \mid a.P \mid P + P \mid P \parallel P$$

Note that here $a.P$ stands for *one unary operator* (action-prefixing with one particular action a) and not, as it is customary, for a collection of unary operators. Henceforth, we denote the signature of the above-mentioned calculus by Σ_o since that fragment of CCS will be the target language in all the applications of Theorem 1 to follow.

The operational semantics of the calculus above is given by the following SOS rules.

$$\begin{array}{lll}
 \text{(a)} \frac{}{a.x \xrightarrow{a} x} & \text{(c0)} \frac{x_0 \xrightarrow{a} y}{x_0 + x_1 \xrightarrow{a} y} & \text{(c1)} \frac{x_1 \xrightarrow{a} y}{x_0 + x_1 \xrightarrow{a} y} \\
 \text{(p0)} \frac{x_0 \xrightarrow{a} y_0}{x_0 \parallel x_1 \xrightarrow{a} y_0 \parallel x_1} & \text{(p1)} \frac{x_1 \xrightarrow{a} y_1}{x_0 \parallel x_1 \xrightarrow{a} x_0 \parallel y_1} &
 \end{array}$$

Note that since there is only one action (and no co-action) in our signature, the standard SOS rule for communication in CCS can be safely omitted.

Definition 4. A symmetric relation $R \subseteq \mathcal{C}(\Sigma_o) \times \mathcal{C}(\Sigma_o)$ is a strong bisimulation when for all $(p, q) \in R$ and $p' \in \mathcal{C}(\Sigma_o)$, if $p \xrightarrow{a} p'$ then there exists a q' such that $q \xrightarrow{a} q'$ and $(p', q') \in R$. Two closed terms p and q are strongly bisimilar (or just bisimilar), denoted by $p \leftrightarrow_b q$, when there exists a strong bisimulation R such that $(p, q) \in R$.

Moller showed in [29, 30] that strong bisimilarity affords no finite ground-complete axiomatization over the above calculus. His negative result was a corollary of a statement to the effect that the following set of closed equations (which are sound modulo strong bisimilarity), denoted henceforth by \mathcal{M} , cannot be derived from any finite set of sound axioms over the signature Σ_o :

$$\{a^1 \parallel (a^1 + a^2 + \dots + a^n) = a.(a^1 + a^2 + \dots + a^n) + a^2 + a^3 + \dots + a^{n+1} \mid n \geq 1\} ,$$

where, for each $i \geq 1$,

$$a^i = \underbrace{a \dots a}_{i \text{ times}} . \mathbf{0} .$$

Theorem 5 (Moller [29, 30]). *There is no finite axiomatization E over the signature Σ_o that is sound modulo strong bisimilarity and proves all the equations in \mathcal{M} .*

Remark 4. Theorem 5 applies regardless of the choice of action a . In particular, it holds true when τ is taken to be the only action in the signature of CCS. We shall make use of this observation in our developments in Section 4.7.

In the remainder of this section, we use Theorems 1 and 5 to obtain other non-finite axiomatizability results, with the aforementioned fragment of CCS as the target language for our reductions. In order to make the paper self-contained, we present the syntax, operational semantics and a notion of behavioural equivalence or preorder for each of the languages we consider in what follows. However, we refer the reader to the original literature for motivation and examples.

4.2 Discrete-time CCS and Timed Bisimilarity

Timed CCS is a timed extension of CCS proposed by Wang Yi [38]. In [9], we proved some non-finite axiomatizability results for Timed CCS modulo timed bisimilarity under the assumption that the underlying time domain satisfy a density property, and left open whether those results carry over to the discrete-time fragment of Timed CCS (referred to as DiTCCS in what follows). In this section, we instantiate our reduction theorem to show that a finite sound and ground-complete axiomatization for DiTCCS modulo timed bisimilarity does not exist.

Let A be a set of actions that contains the action a . Following Milner, we write \bar{A} for the set of complementary actions $\{\bar{b} \mid b \in A\}$, and assume that $\bar{\bar{\alpha}} = \alpha$ for each $\alpha \in A \cup \bar{A}$.

The syntax of DiTCCS is given below:

$$P ::= \mathbf{0} \mid \mu.P \mid \epsilon(d).P \mid P + P \mid P \parallel P \ ,$$

where $\mu.P$ is a set of unary operators, one for each $\mu \in A \cup \bar{A} \cup \{\tau\}$, and $\epsilon(d).P$ is a set of unary operators, one for each $d \in \mathbb{N} = \{1, 2, \dots\}$. In this subsection, we refer to the signature of DiTCCS as Σ_e since we use this language as our source language in applying Theorem 1.

Remark 5. In a discrete-time setting, it would be enough to consider the fragment of DiTCCS that only contains the delay-prefixing operator $\epsilon(1).-$. Indeed, modulo any reasonable notion of equivalence for that calculus, one can express an arbitrary delay prefixing $\epsilon(d).P$, with $d \in \mathbb{N}$, thus:

$$\epsilon(d).P = \underbrace{\epsilon(1) \dots \epsilon(1)}_{d \text{ times}}.P \ .$$

The non-finite axiomatizability result we present below holds true also for the language that only contains the delay-prefixing operator $\epsilon(1).-$.

The operational semantics of DiTCCS is given by the set of SOS rules in Table 1, where $\alpha \in A \cup \bar{A}$, $\mu \in A \cup \bar{A} \cup \{\tau\}$ and $d, e \in \mathbb{N}$. Those rules define transitions between closed DiTCCS terms. The side condition in rule **(tp)** on Table 1 uses the *timed sort* $\text{Sort}_d(p)$, where p is a closed DiTCCS term and $d \in \mathbb{N}$, which is defined thus:

$$\text{Sort}_d(p) = \{\alpha \in A \cup \bar{A} \mid p \xrightarrow{\epsilon(e)} p' \xrightarrow{\alpha} \text{ for some } p' \text{ and } e < d\} \ .$$

(The timed sort of a process can be defined structurally as in [38, Definition 4.1].) For example, the side condition prevents the process $\epsilon(1).a.\mathbf{0} \parallel \bar{a}.\mathbf{0}$ from delaying for two time units.

The notion of equivalence over DiTCCS we shall consider in what follows is *timed bisimilarity*.

Definition 5. *A symmetric relation $R \subseteq \mathcal{C}(\Sigma_e) \times \mathcal{C}(\Sigma_e)$ is a timed bisimulation when for all $(p, q) \in R$, $\chi \in A \cup \bar{A} \cup \{\tau\} \cup \{\epsilon(d) \mid d \in \mathbb{N}\}$ and $p' \in \mathcal{C}(\Sigma_e)$, if $p \xrightarrow{\chi} p'$ then there exists a $q' \in \mathcal{C}(\Sigma_e)$ such that $q \xrightarrow{\chi} q'$ and $(p', q') \in R$. Two closed terms p and q are timed bisimilar, denoted by $p \xleftrightarrow{t} q$, when there exists a timed bisimulation R such that $(p, q) \in R$.*

$$\begin{array}{ccc}
(\mathbf{tn}) \frac{}{\mathbf{0} \xrightarrow{\epsilon(d)} \mathbf{0}} & (\mathbf{a}) \frac{}{\mu.x \xrightarrow{\mu} x} & (\mathbf{ta}) \frac{}{\alpha.x \xrightarrow{\epsilon(d)} \alpha.x} \\
(\mathbf{td0}) \frac{}{\epsilon(d).x \xrightarrow{\epsilon(d)} x} & (\mathbf{td1}) \frac{}{\epsilon(d+e).x \xrightarrow{\epsilon(d)} \epsilon(e).x} & (\mathbf{td2}) \frac{x \xrightarrow{\epsilon(e)} y}{\epsilon(d).x \xrightarrow{\epsilon(d+e)} y} \\
(\mathbf{c0}) \frac{x_0 \xrightarrow{\mu} y}{x_0 + x_1 \xrightarrow{\mu} y} & (\mathbf{c1}) \frac{x_1 \xrightarrow{\mu} y}{x_0 + x_1 \xrightarrow{\mu} y} & (\mathbf{tc}) \frac{x_0 \xrightarrow{\epsilon(d)} y_0 \quad x_1 \xrightarrow{\epsilon(d)} y_1}{x_0 + x_1 \xrightarrow{\epsilon(d)} y_0 + y_1} \\
(\mathbf{p0}) \frac{x_0 \xrightarrow{\mu} y_0}{x_0 \parallel x_1 \xrightarrow{\mu} y_0 \parallel x_1} & (\mathbf{p1}) \frac{x_1 \xrightarrow{\mu} y_1}{x_0 \parallel x_1 \xrightarrow{\mu} x_0 \parallel y_1} & (\mathbf{p2}) \frac{x_0 \xrightarrow{\alpha} y_0 \quad x_1 \xrightarrow{\bar{\alpha}} y_1}{x_0 \parallel x_1 \xrightarrow{\tau} y_0 \parallel y_1} \\
(\mathbf{tp}) \frac{x_0 \xrightarrow{\epsilon(d)} y_0 \quad x_1 \xrightarrow{\epsilon(d)} y_1}{x_0 \parallel x_1 \xrightarrow{\epsilon(d)} y_0 \parallel y_1} \quad \text{Sort}_d(x_0) \cap \overline{\text{Sort}_d(x_1)} = \emptyset
\end{array}$$

Table 1. SOS Rules for DiTCCS

It is well known that \leftrightarrow_t is a congruence over DiTCCS; see, e.g., [38, Theorem 5.1], where the congruence result is stated for dense-time Timed CCS.

Theorem 6. *DiTCCS affords no finite ground-complete axiomatization modulo \leftrightarrow_t .*

In the remainder of this subsection, we prove the above result using Theorem 1. To this end, we begin by defining the following translation $\widehat{\cdot} : \mathcal{T}(\Sigma_e) \rightarrow \mathcal{T}(\Sigma_o)$.

$$\begin{array}{l}
\widehat{\mathbf{0}} = \mathbf{0} \quad \widehat{x} = x \quad \widehat{\mu.p} = \begin{cases} a.\widehat{p} & \text{if } \mu = a, \\ \mathbf{0} & \text{if } \mu \neq a. \end{cases} \\
\widehat{\epsilon(d).p} = \mathbf{0} \quad \widehat{p+q} = \widehat{p} + \widehat{q} \quad \widehat{p \parallel q} = \widehat{p} \parallel \widehat{q}
\end{array}$$

Lemma 2. *The mapping $\widehat{\cdot}$ defined above is structural.*

Consider now the set of Moller's equations \mathcal{M} , which are sound over CCS modulo bisimilarity. In order to prove that timed bisimilarity is not finitely based over DiTCCS, by Theorem 1 it suffices only to show the following statements:

1. $t \leftrightarrow_t u \Rightarrow \widehat{t} \leftrightarrow_b \widehat{u}$, for each $t, u \in \mathcal{T}(\Sigma_e)$, and
2. $\widehat{\cdot}$ is \mathcal{M} -reflecting.

Note that, for each axiomatization E over the signature of DiTCCS,

$$E \vdash t = u \Rightarrow \widehat{E} \vdash \widehat{t} = \widehat{u}$$

holds by Theorem 2 since $\widehat{\cdot}$ is structural (Lemma 2). Therefore, once we prove the two statements above, Theorem 6 indeed follows as a corollary of Theorem 1.

Next, we give the proofs of the above two statements.

1. Proof of $t \leftrightarrow_t u \Rightarrow \widehat{t} \leftrightarrow_b \widehat{u}$.

In order to prove this statement, it suffices to show that the relation

$$R = \{(\sigma(\widehat{t}), \sigma(\widehat{u})) \mid t \leftrightarrow_t u \wedge \sigma : V \rightarrow \mathcal{C}(\Sigma_o)\}$$

is a bisimulation. To this end, observe, first of all, that R is symmetric. In order to prove that R satisfies the transfer property in Definition 4, we shall make use of the following two claims, whose proof will be given later.

- (a) For all $p \in \mathcal{C}(\Sigma_e)$ and $p' \in \mathcal{C}(\Sigma_o)$, if $\widehat{p} \xrightarrow{a} p'$ with respect to the operational semantics of CCS, then there exists some $p'' \in \mathcal{C}(\Sigma_e)$ such that $p \xrightarrow{a} p''$, with respect to the operational semantics of DiTCCS, and $\widehat{p''} \equiv p'$.
- (b) For all $p, p' \in \mathcal{C}(\Sigma_e)$, if $p \xrightarrow{a} p'$ with respect to the operational semantics of DiTCCS, then $\widehat{p} \xrightarrow{a} \widehat{p'}$ with respect to the operational semantics of CCS.

Assume now that $\sigma(\widehat{t}) R \sigma(\widehat{u})$ and $\sigma(\widehat{t}) \xrightarrow{a} p'_0$. By Lemmas 1 and 2, $\sigma(\widehat{t}) \equiv \widehat{\sigma(t)}$. It follows from item 1a above that $\sigma(t) \xrightarrow{a} p_0$, for some p_0 such that $\widehat{p_0} \equiv p'_0$. Furthermore, as t and u are timed bisimilar, $\sigma(u) \xrightarrow{a} p_1$, for some p_1 such that $p_0 \leftrightarrow_t p_1$. From item 1b and Lemmas 1–2, we have that $\sigma(\widehat{u}) \equiv \widehat{\sigma(u)} \xrightarrow{a} \widehat{p_1}$ and, by the definition of R , we may conclude that $p'_0 = \widehat{p_0} R \widehat{p_1}$, which was to be shown.

In order to complete the proof of this statement, we are therefore left to show items 1a and 1b. This we now proceed to do.

- (a) Proof of item 1a.

We prove this claim by an induction on the structure of p .

- Assume that $p \equiv \mathbf{0}$. This is impossible since then $\widehat{p} \equiv \mathbf{0}$ cannot make an a -transition.
- Assume that $p \equiv \mu.p_0$. Then p must be of the form $a.p_0$ (in order for \widehat{p} to make an a -transition) and thus, $\widehat{p} = a.\widehat{p_0} \xrightarrow{a} \widehat{p_0} = p'$. The claim then follows since $a.p_0 \xrightarrow{a} p_0$.
- Assume that $p \equiv \epsilon(d).p_0$. This is impossible since then \widehat{p} would not be able to make an a -transition.
- Assume that $p \equiv p_0 + p_1$. Then $\widehat{p} \equiv \widehat{p_0} + \widehat{p_1}$. Suppose, without loss of generality, that the transition $\widehat{p_0} + \widehat{p_1} \xrightarrow{a} p'$ is due to an application of rule **(c0)**; thus, $\widehat{p_0} \xrightarrow{a} p'$. It then follows from the induction hypothesis that $p_0 \xrightarrow{a} p''$ for some p'' such that $\widehat{p''} \equiv p'$. By applying deduction rule **(c0)**, we obtain $p \equiv p_0 + p_1 \xrightarrow{a} p''$.
- The case $p \equiv p_0 \parallel p_1$ is similar to the one above and is therefore omitted.

- (b) Proof of item 1b.

By an induction on the depth of the proof for $p \xrightarrow{a} p'$. We distinguish the following cases based on the last deduction rule applied to obtain $p \xrightarrow{a} p'$.

- (a) In this case, p is of the form $a.p_0$ and $p' \equiv p_0$. Thus, using to the same deduction rule in the semantics of CCS, we have $\widehat{p} \equiv a.\widehat{p_0} \xrightarrow{a} \widehat{p_0}$.
- (c0)** Then $p \equiv p_0 + p_1$ and $p_0 \xrightarrow{a} p'$ by a shorter inference. It follows from the induction hypothesis that $\widehat{p_0} \xrightarrow{a} \widehat{p'}$ and, using rule **(c0)** in the semantics of CCS, we infer that $\widehat{p_0} + \widehat{p_1} \xrightarrow{a} \widehat{p'}$. Furthermore, by the definition of $\widehat{\cdot}$, we have that $\widehat{p} \equiv \widehat{p_0} + \widehat{p_1}$.

The cases for deduction rules **(c1)**, **(p0)** and **(p1)** are similar to the case of **(c0)**.

The proof of the first statement is now complete.

2. Proof of the fact that $\widehat{\cdot}$ is \mathcal{M} -reflecting.

We show that all axioms in \mathcal{M} are sound modulo \leftrightarrow_t . Since $\widehat{\cdot}$ is the identity over CCS terms, the statement then follows immediately. To this end, we prove the following two claims.

- (a) For each $p \in \mathcal{C}(\Sigma_o)$ and positive integer d , $p \xrightarrow{\epsilon(d)} p'$ iff $p \equiv p'$. By an induction on the structure of p . The cases for $\mathbf{0}$ and $a.p_0$ follow from deduction rules **(tn)** and **(ta)**, respectively. The cases for $p_0 + p_1$ and $p_0 \parallel p_1$ follow from the induction hypothesis, and **(tc)** and **(tp)**, respectively.
- (b) For each $p, q \in \mathcal{C}(\Sigma_o)$, if $p \leftrightarrow_b q$ then $p \leftrightarrow_t q$.

We show that \leftrightarrow_b is a timed bisimulation. To this end, note, first of all, that the relation \leftrightarrow_b is symmetric. Assume now that $p \xrightarrow{a} p'$ and $p \leftrightarrow_b q$. Since \leftrightarrow_b is a bisimulation, it follows that $q \xrightarrow{a} q'$ (with respect to the semantics of CCS, and thus of DiTCCS using the same deduction rules) for some q' such that $p' \leftrightarrow_b q'$, and we are done. That delay transitions of p may be matched by q follows trivially from the previous item.

Since all the provisos of Theorem 1 are met, Theorem 6 follows.

4.3 Temporal CCS

In the paper [32], Moller and Tofts proposed another timed extension of Milner’s CCS, which they called *Temporal Calculus of Communicating Systems* (referred to as TCCS_{MT} in what follows to avoid any confusion with Wang Yi’s Timed CCS), and studied its semantics theory modulo timed bisimilarity. Our order of business in this section will be to use our reduction-based method to show that timed bisimilarity affords no finite ground-complete axiomatization over TCCS_{MT} .

For our purposes in this section, TCCS_{MT} is the language generated by the following grammar:

$$P ::= \mathbf{0} \mid \mu.P \mid (d).P \mid \delta.P \mid P + P \mid P \oplus P \mid P \parallel P ,$$

where $\mu.P$ is a set of unary operators, one for each $\mu \in A \cup \bar{A} \cup \{\tau\}$, and $(d).P$ is a set of unary operators, one for each positive integer d . The intuition underlying each of the operators in the signature of TCCS_{MT} is carefully described in [32, Pages 402–403]. For the sake of clarity, however, we find it useful to mention that:

- process terms of the form $\mathbf{0}$ or $a.p$ cannot delay, unlike in DiTCCS ;
- $(d).p$ behaves exactly like $\epsilon(d).p$ in DiTCCS ;
- $\delta.p$ describes a process which behaves like p , but is willing to wait any amount of time before doing so; and
- $p \oplus q$ is a “weak choice” between p and q . The choice between p and q is made upon performance of an action from either of the two processes, or at the occurrence of a time delay which can only be performed by one of the processes. By way of example, as $a.p$ cannot delay, a process of the form $a.p \oplus (1).\mathbf{0}$ will be transformed into $\mathbf{0}$ after a delay of one time unit.

In order to define the operational semantics of the weak choice operator, the Plotkin-style rules for that operator from [32] make use of the function $\text{maxdelay}()$, which associates a non-negative integer or ω with each closed TCCS_{MT} term. The function $\text{maxdelay}()$ is defined by structural induction on terms as follows:

$$\begin{aligned} \text{maxdelay}(\mathbf{0}) &= \text{maxdelay}(\mu.p) = 0 & \text{maxdelay}(\delta.p) &= \omega \\ \text{maxdelay}(p + q) &= \text{maxdelay}(p \parallel q) = \min(\text{maxdelay}(p), \text{maxdelay}(q)) \\ \text{maxdelay}(p \oplus q) &= \max(\text{maxdelay}(p), \text{maxdelay}(q)) . \end{aligned}$$

The operational semantics of closed TCCS_{MT} terms is given by means of two types of transitions, namely actions transitions $\xrightarrow{\mu}$ with $\mu \in A \cup \bar{A} \cup \{\tau\}$ and delay transitions $\xrightarrow{\epsilon(d)}$, with $d \in \mathbb{N}$. The transition relations $\xrightarrow{\mu}$ are defined as for DiTCCS , with the proviso that

- $(d).p$ has no outgoing action transitions,
- $p \oplus q$ has the same outgoing action transitions as $p + q$, and
- the action transitions of $\delta.p$ are exactly those of p —i.e., they are those provable using the rules

$$\frac{x \xrightarrow{\mu} y}{\delta.x \xrightarrow{\mu} y} \quad (\mu \in A \cup \bar{A} \cup \{\tau\}) .$$

On the other hand, the transition relations $\xrightarrow{\epsilon(d)}$ are the least relations satisfying the rules on Table 2. Closed TCCS_{MT} terms are considered modulo timed bisimilarity \leftrightarrow_t (as defined in Definition 5). Timed bisimilarity is a congruence over TCCS_{MT} as shown in [32, Proposition 3.4].

Theorem 7. *TCCS_{MT} affords no finite ground-complete axiomatization modulo \leftrightarrow_t .*

In the remainder of this subsection, we prove the above result using Theorem 1. To this end, we begin by defining the following translation $\widehat{\cdot}$ from open TCCS_{MT} terms to open CCS terms.

$$\begin{aligned} \widehat{\mathbf{0}} &= \mathbf{0} & \widehat{x} &= x & \widehat{\delta.p} &= \widehat{p} \\ \widehat{a.p} &= a.\widehat{p} & \widehat{\mu.p} &= \mathbf{0} \text{ for } \mu \neq a & \widehat{(d).p} &= \mathbf{0} \\ \widehat{p + q} &= \widehat{p} + \widehat{q} & \widehat{p \oplus q} &= \widehat{p} + \widehat{q} & \widehat{p \parallel q} &= \widehat{p} \parallel \widehat{q} \end{aligned}$$

$\overline{\delta.x \xrightarrow{\epsilon(d)} \delta.x}$	$\overline{(d).x \xrightarrow{\epsilon(d)} x}$	$\overline{(d+e).x \xrightarrow{\epsilon(d)} (e).x}$	$\frac{x \xrightarrow{\epsilon(e)} y}{(d).x \xrightarrow{\epsilon(d+e)} y}$
$\frac{x_0 \xrightarrow{\epsilon(d)} y_0 \quad x_1 \xrightarrow{\epsilon(d)} y_1}{x_0 \oplus x_1 \xrightarrow{\epsilon(d)} y_0 \oplus y_1}$	$\frac{x_0 \xrightarrow{\epsilon(d)} y_0 \quad \text{maxdelay}(x_1) < d}{x_0 \oplus x_1 \xrightarrow{\epsilon(d)} y_0}$	$\frac{x_1 \xrightarrow{\epsilon(d)} y_1 \quad \text{maxdelay}(x_0) < d}{x_0 \oplus x_1 \xrightarrow{\epsilon(d)} y_1}$	
$\frac{x_0 \xrightarrow{\epsilon(d)} y_0 \quad x_1 \xrightarrow{\epsilon(d)} y_1}{x_0 + x_1 \xrightarrow{\epsilon(d)} y_0 + y_1}$	$\frac{x_0 \xrightarrow{\epsilon(d)} y_0 \quad x_1 \xrightarrow{\epsilon(d)} y_1}{x_0 \parallel x_1 \xrightarrow{\epsilon(d)} y_0 \parallel y_1}$		

Table 2. Rules defining the delay transitions $\xrightarrow{\epsilon(d)}$ over TCCS_{MT} ($d \in \mathbb{N}$)

Lemma 3. *The mapping $\widehat{\cdot}$ defined above is structural.*

Consider now the set of Moller's equations \mathcal{M} , which are sound over CCS modulo bisimilarity. In order to prove that timed bisimilarity is not finitely based over TCCS_{MT} , by Theorem 1 it suffices only to show the following statements:

1. $t \xleftrightarrow{a} u$ implies $\widehat{t} \xleftrightarrow{a} \widehat{u}$, for all TCCS_{MT} terms t, u , and
2. $\widehat{\cdot}$ is \mathcal{M} -reflecting.

Note that, for all TCCS_{MT} terms t, u and axiomatization E ,

$$E \vdash t = u \Rightarrow \widehat{E} \vdash \widehat{t} = \widehat{u}$$

holds by Theorem 2 since $\widehat{\cdot}$ is structural (Lemma 3). Therefore, once we prove the two statements above, Theorem 7 indeed follows as a corollary of Theorem 1.

We establish the two statements above in turn. The following lemma will be useful.

Lemma 4.

1. Assume that $\widehat{p} \xrightarrow{a} r$ holds with respect to the operational semantics of CCS for some closed TCCS_{MT} term p and CCS term r . Then $p \xrightarrow{a} p'$ holds with respect to the operational semantics of TCCS_{MT} for some closed TCCS_{MT} term p' such that $\widehat{p'} = r$.
2. If $p \xrightarrow{a} p'$ holds with respect to the operational semantics of TCCS_{MT} for some closed TCCS_{MT} terms p, p' then $\widehat{p} \xrightarrow{a} \widehat{p'}$ holds with respect to the operational semantics of CCS.

We are now ready to show that $\widehat{\cdot}$ preserves sound equations.

Proposition 1. $t \xleftrightarrow{a} u$ implies $\widehat{t} \xleftrightarrow{a} \widehat{u}$, for all TCCS_{MT} terms t, u .

Proof. It suffices to show that the relation

$$R = \{(\widehat{p}, \widehat{q}) \mid p \xleftrightarrow{a} q, \text{ with } p, q \text{ closed } \text{TCCS}_{\text{MT}} \text{ terms}\}$$

is a strong bisimulation. Indeed, assuming that R is a strong bisimulation, we can show the proposition as follows.

Suppose that $t \xleftrightarrow{a} u$ holds for some TCCS_{MT} terms t, u . Let σ be a closed CCS substitution. We shall argue that $\sigma(\widehat{t}) \xleftrightarrow{a} \sigma(\widehat{u})$ holds. This follows because

- $\sigma(\widehat{t}) = \widehat{\sigma(t)}$ and $\sigma(\widehat{u}) = \widehat{\sigma(u)}$ (by Lemma 1, as $\widehat{\cdot}$ is structural and $\sigma = \widehat{\sigma}$), and
- $\widehat{\sigma(t)} \xleftrightarrow{a} \widehat{\sigma(u)}$ (since $\widehat{\sigma(t)} R \widehat{\sigma(u)}$ and R is a strong bisimulation).

So we are left to show that R is indeed a strong bisimulation. This can be easily checked using Lemma 4. \square

To complete the proof of Theorem 7, we now show that $\hat{\cdot}$ is \mathcal{M} -reflecting. Since $\hat{\cdot}$ is the identity function over CCS terms, it suffices to prove the following result. (Note that, since CCS is a reduct of the language TCCS_{MT} , it makes sense to consider CCS terms modulo \leftrightarrow_t .)

Proposition 2. *The relations \leftrightarrow_t and \leftrightarrow_b coincide over CCS terms.*

Proof. The relation \leftrightarrow_t is included in \leftrightarrow_b over the collection of CCS terms by Proposition 1. The converse inclusion follows because \leftrightarrow_b is a timed bisimulation. This can be shown using Lemma 4 and observing that $p \xrightarrow{\epsilon^{(d)}}$ holds for each closed CCS term p and positive integer d . \square

Since all the provisos of Theorem 1 are met by our reduction, Theorem 7 follows.

4.4 ATP and Timed Bisimilarity

In [35] Nicollin and Sifakis defined the Algebra of Timed Processes (ATP), which has the following syntax.

$$P ::= \delta \mid \mu.P \mid P \oplus P \mid P \mid P \parallel P$$

Deadlock is represented by δ in ATP and is akin to $\mathbf{0}$ in DiTCCS, i.e., it can only delay. In the semantics of ATP, only unit delays are present and are denoted by $\overset{\chi}{\rightarrow}$. To be consistent with the rest of our presentation, we denote such transitions with $\overset{\epsilon^{(1)}}{\rightarrow}$. Thus, the only deduction rule for δ is the following.

$$\overline{\delta \overset{\epsilon^{(1)}}{\rightarrow} \delta}$$

Action prefixing is denoted by $\mu.P$; a process of that form can only perform action μ and turn into P in doing so, i.e., it is not delayable. Nondeterministic choice is denoted by $P \oplus Q$ and has a semantics that is identical to that of $+$ in DiTCCS and Temporal CCS (and is thus different from Temporal CCS's \oplus). The semantics of the unit-delay operator $_$ allows for two possible transitions: either the first argument takes an action, thereby taking over control for the rest of the execution, or the process delays for a single unit of time and, as a result, the second argument takes over control for the rest of the computation. This semantics is captured by the following two deduction rules.

$$\frac{x_0 \xrightarrow{\mu} y_0}{[x_0](x_1) \xrightarrow{\mu} y_0} \quad \frac{}{[x_0](x_1) \overset{\epsilon^{(1)}}{\rightarrow} x_1}$$

Parallel composition in ATP behaves like the same operator in Temporal CCS. In particular, the deduction rule for the unit-delay transition of a parallel composition is as follows.

$$\frac{x_0 \overset{\epsilon^{(1)}}{\rightarrow} y_0 \quad x_1 \overset{\epsilon^{(1)}}{\rightarrow} y_1}{x_0 \parallel x_1 \overset{\epsilon^{(1)}}{\rightarrow} y_0 \parallel y_1}$$

In the remainder of this subsection, we denote the signature of ATP by Σ_e . The notion of equivalence used for ATP is the specialization of the notion of timed bisimilarity (given in Definition 5) to this calculus, which we denote by \leftrightarrow_t .

Theorem 8. *ATP affords no finite ground-complete axiomatization modulo \leftrightarrow_t .*

We prove the above theorem using our reduction method. To this end, we define the following structural reduction from ATP to CCS.

$$\begin{array}{l} \widehat{\delta} = \mathbf{0} \quad \widehat{x} = x \quad \widehat{a.p} = a.\widehat{p} \quad \widehat{\mu.p} = \mathbf{0} \text{ for } \mu \neq a \\ \widehat{p \oplus q} = \widehat{p} + \widehat{q} \quad \widehat{[p](q)} = \widehat{p} \quad \widehat{p \parallel q} = \widehat{p} \parallel \widehat{q} \end{array}$$

To show that the above reduction is \mathcal{M} -reflecting, observe the following two facts.

1. For each process $p \in \mathcal{C}(\Sigma_o)$, it holds that $p \equiv \widehat{p}_\delta$, where p_δ is the ATP term obtained by replacing in p each occurrence of $\mathbf{0}$ with δ and $+$ with \oplus .
2. For each $p, q \in \mathcal{C}(\Sigma_o)$ if $p \leftrightarrow_b q$, then $p \leftrightarrow_t q$. This claim follows from the following two facts, which hold for all $p, q \in \mathcal{C}(\Sigma_o)$:
 - $p \xrightarrow{a} q$ iff $p_\delta \xrightarrow{a} q_\delta$;
 - $p_\delta \xrightarrow{\epsilon(1)} r$ iff $p_\delta \leftrightarrow_t \delta$ and $r \equiv p_\delta$, for each $r \in \mathcal{C}(\Sigma_e)$.

We are therefore left to prove that our reduction preserves valid equations. For the rest of the proof, we need the following two auxiliary lemmas.

Lemma 5. *For all $p, p' \in \mathcal{C}(\Sigma_e)$, if $p \xrightarrow{a} p'$, then $\widehat{p} \xrightarrow{a} \widehat{p}'$.*

Proof. By induction on the proof of the transition $p \xrightarrow{a} p'$. We proceed by a case analysis on the last rule used in the proof.

- Assume that $p \equiv a.p' \xrightarrow{a} p'$. Trivial.
- Assume that $p \equiv p_0 \oplus p_1 \xrightarrow{a} p'$. Without loss of generality, we can assume that $p_0 \xrightarrow{a} p'$. By the induction hypothesis we have $\widehat{p}_0 \xrightarrow{a} \widehat{p}'$. Thus, $\widehat{p} \equiv \widehat{p}_0 + \widehat{p}_1 \xrightarrow{a} \widehat{p}'$.
- Assume that $p \equiv [p_0](p_1) \xrightarrow{a} p'$. Then it follows from the semantics of unit delay operator that $p_0 \xrightarrow{a} p'$. By the induction hypothesis, we have $\widehat{p}_0 \xrightarrow{a} \widehat{p}'$. Then $\widehat{p} \equiv \widehat{p}_0 \xrightarrow{a} \widehat{p}'$.
- Assume that $p \equiv p_0 \parallel p_1 \xrightarrow{a} p'$. Without loss of generality, we can assume that there is some p'_0 such that $p_0 \xrightarrow{a} p'_0$ and $p' \equiv p'_0 \parallel p_1$. By the induction hypothesis, we have $\widehat{p}_0 \xrightarrow{a} \widehat{p}'_0$. Then, $\widehat{p} \equiv \widehat{p}_0 \parallel \widehat{p}_1 \xrightarrow{a} \widehat{p}'_0 \parallel \widehat{p}_1 \equiv \widehat{p}'$. \square

Lemma 6. *Assume that $p \in \mathcal{C}(\Sigma_e)$ and $q \in \mathcal{C}(\Sigma_o)$. If $\widehat{p} \xrightarrow{a} q$, then $p \xrightarrow{a} p'$, for some $p' \in \mathcal{C}(\Sigma_e)$ such that $\widehat{p}' \equiv q$.*

Proof. By induction on the structure of $p \in \mathcal{C}(\Sigma_e)$. The claim is vacuous if $p \equiv \delta$ or $p \equiv \mu.p'$ with $\mu \neq a$. If $p \equiv a.p'$, then $q \equiv \widehat{p}'$ and the lemma follows since $p \xrightarrow{a} p'$. If $p \equiv [p_0](p_1)$, then $\widehat{p} \equiv \widehat{p}_0 \xrightarrow{a} q$. By the induction hypothesis, $p_0 \xrightarrow{a} p'$, for some p' such that $\widehat{p}' \equiv q$. It follows from the semantics of the unit delay operator that $p \equiv [p_0](p_1) \xrightarrow{a} p'$ and we are done. The cases $p \equiv p_0 \oplus p_1$ and $p \equiv p_0 \parallel p_1$ are similar to those in the previous proof of this claim for DiTCCS. \square

Since $\widehat{\cdot}$ is structural, to complete the proof of Theorem 8, we are only left to show the following proposition.

Proposition 3. *For all $p, q \in \mathcal{C}(\Sigma_e)$, if $p \leftrightarrow_t q$ then $p \leftrightarrow_b q$.*

Proof. We show that $\{(\widehat{p}, \widehat{q}) \mid p \leftrightarrow_t q \wedge p, q \in \mathcal{C}(\Sigma_e)\}$ is a bisimulation. Note that R is symmetric due to the symmetry of \leftrightarrow_t . Assume that $\widehat{p} R \widehat{q}$ and $\widehat{p} \xrightarrow{a} r$ for some $r \in \mathcal{C}(\Sigma_o)$.

By Lemma 6, we have that $p \xrightarrow{a} p'$ for some p' such that $\widehat{p}' \equiv r$. Since $(\widehat{p}, \widehat{q}) \in R$, we have that $p \leftrightarrow_t q$; therefore, it follows from $p \xrightarrow{a} p'$ that there exists some q' such that $q \xrightarrow{a} q'$ and $p' \leftrightarrow_t q'$. By Lemma 5, we have that $\widehat{q} \xrightarrow{a} \widehat{q}'$. It follows from the definition of R that $r \equiv \widehat{p}' R \widehat{q}'$, and we are done. \square

Since all the provisos of Theorem 1 are met by our reduction, Theorem 8 follows.

4.5 $TACS^{UT}$ and Faster-Than Preorder

Another discrete-time extension of CCS, called the calculus of Timed Asynchronous Communicating Systems ($TACS^{UT}$), is presented in [24]. $TACS^{UT}$ is meant to be a calculus for the analysis of the worst-case timing behaviour of reactive systems.

The syntax of $TACS^{UT}$ is given below. (In the following grammar, to be consistent with our presentation of DiTCCS, we use $\epsilon(1).P$, instead of the original notation $\sigma.P$, for a unit delay. Moreover, the meta-variable μ ranges over $A \cup \overline{A} \cup \{\tau\}$ as in the grammar for DiTCCS.)

$$P ::= \mathbf{0} \mid \underline{\mu}.P \mid \underline{\epsilon(1)}.P \mid P + P \mid P \parallel P$$

In this subsection, we refer to the signature of $TACS^{UT}$ as Σ_e since we use this language as our source language in applying Theorem 1.

The operational semantics of $\mathbf{0}$ and $+$ is the same as that of their counterparts in DiTCCS—see Table 1—, but in this setting $d = 1$ is the only possible time delay. The semantics of $\underline{\mu}.$ and $\underline{\epsilon(1)}.$ are specified by the following rules, where $\alpha \in A \cup \bar{A}$.

$$\begin{array}{c} \text{(dd0)} \frac{}{\underline{\epsilon(1)}.x \xrightarrow{\epsilon(1)} x} \quad \text{(dd1)} \frac{x \xrightarrow{\mu} y}{\underline{\epsilon(1)}.x \xrightarrow{\mu} y} \quad \text{(a)} \frac{}{\underline{\mu}.x \xrightarrow{\mu} x} \quad \text{(ta)} \frac{}{\underline{\alpha}.x \xrightarrow{\epsilon(1)} \underline{\alpha}.x} \end{array}$$

In the light of the first two rules above, $\underline{\epsilon(1)}.P$ indicates a delay of *at most one time unit* before the execution of P . Hence, unlike in DiTCCS, $\underline{\epsilon(1)}.a.\mathbf{0} \xrightarrow{a} \mathbf{0}$ holds. Note, however, that action a is *not* urgent in $\underline{\epsilon(1)}.a.\mathbf{0}$ since its occurrence may be delayed by one time unit in any context because the action is in the scope of a delay-prefixing operator. This is formalized in the following definition.

Definition 6. *The set of urgent initial actions of a process p , denoted by $\mathcal{U}(p)$, is inductively defined as follows.*

$$\begin{aligned} \mathcal{U}(\mathbf{0}) &= \emptyset & \mathcal{U}(\underline{\mu}.p) &= \{\mu\} & \mathcal{U}(\underline{\epsilon(1)}.p) &= \emptyset \\ \mathcal{U}(p + q) &= \mathcal{U}(p) \cup \mathcal{U}(q) & \mathcal{U}(p \parallel q) &= \mathcal{U}(p) \cup \mathcal{U}(q) \cup \{\tau \mid \mathcal{U}(p) \cap \overline{\mathcal{U}(q)} \neq \emptyset\} \end{aligned}$$

For instance, $\mathcal{U}(\bar{a}.\mathbf{0} \parallel \underline{\epsilon(1)}.a.\mathbf{0}) = \{\bar{a}\}$ and $\mathcal{U}(\bar{a}.\mathbf{0} \parallel a.\mathbf{0}) = \{\bar{a}, a, \tau\}$.

The SOS rules for \parallel are like those for DiTCCS in Table 1, but rule **(tp)** is replaced by the following one.

$$\frac{x_0 \xrightarrow{\epsilon(1)} y_0 \quad x_1 \xrightarrow{\epsilon(1)} y_1}{x_0 \parallel x_1 \xrightarrow{\epsilon(1)} y_0 \parallel y_1} \quad \tau \notin \mathcal{U}(x_0 \parallel x_1)$$

Definition 7. *The faster-than preorder is the largest relation \sqsupseteq over closed $TACS^{UT}$ terms satisfying the following conditions for each two process p, q such that $p \sqsupseteq q$:*

1. $\forall_{p'} p \xrightarrow{\mu} p' \Rightarrow \exists_{q'} q \xrightarrow{\mu} q' \wedge p' \sqsupseteq q'$,
2. $\forall_{q'} q \xrightarrow{\mu} q' \Rightarrow \exists_{p'} p \xrightarrow{\mu} p' \wedge p' \sqsupseteq q'$ and
3. $\forall_{p'} p \xrightarrow{\epsilon(1)} p' \Rightarrow \mathcal{U}(q) \subseteq \mathcal{U}(p) \wedge \exists_{q'} (q \xrightarrow{\epsilon(1)} q' \wedge p' \sqsupseteq q')$.

Intuitively, $p \sqsupseteq q$ means that p and q have the same behaviour, but p is at least as fast as q . For instance, $\underline{\epsilon(1)}.a.0 \not\sqsupseteq a.0$, but $a.0 \sqsupseteq \underline{\epsilon(1)}.a.0$. In general, $p \sqsupseteq \underline{\epsilon(1)}.p$ holds for each $TACS^{UT}$ process term p , but, as highlighted by the previous example, the converse may fail.

As shown in [24], the faster-than preorder is a precongruence over $TACS^{UT}$.

Theorem 9. *$TACS^{UT}$ affords no finite ground-complete axiomatization modulo \sqsupseteq .*

In the remainder of this subsection, we prove the above result using Theorem 1. To this end, we begin by defining the following mapping from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$, which provides us with the basis for applying Theorem 1 in the proof of Theorem 9.

$$\begin{aligned} \widehat{\mathbf{0}} &= \mathbf{0} & \widehat{x} &= x & \widehat{\underline{\mu}.t} &= \begin{cases} a.\widehat{t} & \text{if } \mu = a, \\ \mathbf{0} & \text{if } \mu \neq a, \end{cases} \\ \widehat{\underline{\epsilon(1)}.t} &= \widehat{t} & \widehat{t + u} &= \widehat{t} + \widehat{u} & \widehat{t \parallel u} &= \widehat{t} \parallel \widehat{u} \end{aligned}$$

Lemma 7. *The mapping $\widehat{}$ defined above is structural.*

Consider now the set of Moller's equations \mathcal{M} , which are sound over CCS modulo bisimilarity. In the light of Theorem 1, in order to prove that the faster-than preorder is not finitely based over $TACS^{UT}$, it suffices only to show the following statements:

1. $t \sqsupseteq u \Rightarrow \widehat{t} \leftrightarrow_b \widehat{u}$, for each $t, u \in \mathcal{T}(\Sigma_e)$, and
2. $\widehat{\cdot}$ is \mathcal{M} -reflecting.

Note that, for each axiomatization E over the signature of $TACS^{UT}$,

$$E \vdash t = u \Rightarrow \widehat{E} \vdash \widehat{t} = \widehat{u}$$

holds by Theorem 2 since $\widehat{\cdot}$ is structural (Lemma 7). Therefore, once we prove the two statements above, Theorem 9 indeed follows as a corollary of Theorem 1.

Next, we give the proofs of the above two statements, which are very similar to those given in Section 4.2.

1. Proof of $t \sqsupseteq u \Rightarrow \widehat{t} \leftrightarrow_b \widehat{u}$.

In order to prove this statement, it suffices to show that the symmetric closure of the relation

$$R = \{(\sigma(\widehat{t}), \sigma(\widehat{u})) \mid t \sqsupseteq u \wedge \sigma : V \rightarrow \mathcal{C}(\Sigma_o)\}$$

is a bisimulation. To show that the symmetric closure of R satisfies the transfer property in Definition 4, we shall make use of the following two claims, whose proof will be given later.

- (a) For all $p \in \mathcal{C}(\Sigma_e)$ and $p' \in \mathcal{C}(\Sigma_o)$, if $\widehat{p} \xrightarrow{a} p'$ with respect to the operational semantics of CCS, then there exists some $p'' \in \mathcal{C}(\Sigma_e)$ such that $p \xrightarrow{a} p''$ with respect to the operational semantics of $TACS^{UT}$ and $\widehat{p''} \equiv p'$.
- (b) For all $p, p' \in \mathcal{C}(\Sigma_e)$, if $p \xrightarrow{a} p'$ with respect to the operational semantics of $TACS^{UT}$, then $\widehat{p} \xrightarrow{a} \widehat{p'}$ with respect to the operational semantics of CCS.

Assume now that $\sigma(\widehat{t}) R \sigma(\widehat{u})$ because $t \sqsupseteq u$, and $\sigma(\widehat{t}) \xrightarrow{a} p'_0$. Note that $\sigma(\widehat{t}) \equiv \widehat{\sigma(t)}$ by Lemmas 1 and 7. It follows from item 1a above that $\sigma(t) \xrightarrow{a} p_0$, for some p_0 such that $\widehat{p_0} \equiv p'_0$. Furthermore, $\sigma(u) \xrightarrow{a} p_1$, for some p_1 such that $p_0 \sqsupseteq p_1$, since $\sigma(t) \sqsupseteq \sigma(u)$ because $t \sqsupseteq u$. From item 2, Lemma 1 and Lemma 7, we have that $\widehat{\sigma(u)} \equiv \sigma(\widehat{u}) \xrightarrow{a} \widehat{p_1}$ and, by the definition of R , we may conclude that $\widehat{p_0} R \widehat{p_1}$. A similar argument applies when $\sigma(\widehat{t}) R \sigma(\widehat{u})$ because $u \sqsupseteq t$.

In order to complete the proof of this statement, we are therefore left to show items 1a and 1b. This we now proceed to do.

- (a) Proof of item 1a.

We prove this claim by an induction on the structure of p .

- Assume that $p \equiv \mathbf{0}$. This is impossible since then $\widehat{p} \equiv \mathbf{0}$ cannot make an a -transition.
- Assume that $p \equiv \underline{a}.p_0$. Then p must be of the form $\underline{a}.p_0$ (in order for \widehat{p} to make an a -transition) and thus, $\widehat{p} = \underline{a}.\widehat{p_0} \xrightarrow{a} \widehat{p_0}$. The claim thus follows by taking $p'' \equiv p_0$.
- Assume that $p \equiv \epsilon(1).p_0$. Then $\widehat{p} \equiv \widehat{p_0} \xrightarrow{a} p'$. It follows from the induction hypothesis that $p_0 \xrightarrow{a} p''$ and $\widehat{p''} \equiv p'$, for some p'' . Using deduction rule **(dd1)**, we infer that $\epsilon(1).p_0 \xrightarrow{a} p''$ and we already have that $\widehat{p''} \equiv p'$.
- Assume that $p \equiv p_0 + p_1$. Then $\widehat{p} \equiv \widehat{p_0} + \widehat{p_1}$. Without loss of generality, assume that transition $\widehat{p_0} + \widehat{p_1} \xrightarrow{a} p'$ is due to **(c0)**; thus, $\widehat{p_0} \xrightarrow{a} p'$. It then follows from the induction hypothesis that $p_0 \xrightarrow{a} p''$ for some p'' such that $\widehat{p''} \equiv p'$. By applying deduction rule **(c0)**, we obtain $p \equiv p_0 + p_1 \xrightarrow{a} p''$, and we are done.
- The case $p \equiv p_0 \parallel p_1$ is similar to one above.

- (b) Proof of item 1b.

By an induction on the depth of the proof for $p \xrightarrow{a} p'$. We distinguish the following cases based on the last deduction rule applied to obtain $p \xrightarrow{a} p'$.

- (a) Then p is of the form $\underline{a}.p_0$ and $p' \equiv p_0$. Thus, according to the same deduction rule in the semantics of CCS, we have $\widehat{p} \equiv \underline{a}.\widehat{p_0} \xrightarrow{a} \widehat{p_0}$.

- (**dd1**) Then $p \equiv \epsilon(1).p_0$ and $p_0 \xrightarrow{a} p'$. It follows from the induction hypothesis that $\widehat{p}_0 \xrightarrow{a} \widehat{p}'$ and by the definition of $\widehat{\cdot}$, we have that $\widehat{p} \equiv \widehat{p}_0$.
- (**c0**) Then $p \equiv p_0 + p_1$ and $p_0 \xrightarrow{a} p'$. It follows from the induction hypothesis that $\widehat{p}_0 \xrightarrow{a} \widehat{p}'$ and, using rule (**c0**) in the semantics of CCS, we infer that $\widehat{p}_0 + \widehat{p}_1 \xrightarrow{a} \widehat{p}'$. Furthermore, by the definition of $\widehat{\cdot}$, we have that $\widehat{p} \equiv \widehat{p}_0 + \widehat{p}_1$, and we are done.

The cases for deduction rules (**c1**), (**p0**) and (**p1**) are similar to the case of (**c0**).

The proof of the first statement is now complete.

2. Proof of the fact that $\widehat{\cdot}$ is \mathcal{M} -reflecting.

We show that all axioms in \mathcal{M} are sound modulo \sqsubseteq once we underline all the occurrences of the a -prefixing operator in CCS terms. (The statement then follows immediately since $\widehat{t} = t$ holds for each CCS term t , where \underline{t} denotes the term resulting by underlining all the a -prefixes in t .) To this end, we prove the following two claims.

- (a) For each $p \in \mathcal{C}(\Sigma_o)$, $\underline{p} \xrightarrow{\epsilon(1)} \underline{p}'$ iff $\underline{p} \equiv \underline{p}'$. By an induction on the structure of p . The cases for **0** and $a.p_0$ follow from deduction rules (**tn**) and (**ta**), respectively. The cases for $p_0 + p_1$ and $p_0 \parallel p_1$ follow from (**tc**) and (**tp**) and the induction hypothesis, respectively.
- (b) For each $p, q \in \mathcal{C}(\Sigma_o)$, $p \xleftrightarrow{b} q \Rightarrow \underline{p} \sqsubseteq \underline{q}$.

We show that the relation

$$R = \{(\underline{p}, \underline{q}) \mid p \xleftrightarrow{b} q \text{ and } p, q \in \mathcal{C}(\Sigma_o)\}$$

satisfies the defining transfer properties for \sqsubseteq (see Definition 7). To this end, assume that $\underline{p} R \underline{q}$ and $\underline{p} \xrightarrow{a} r$ for some r (with respect to the operational semantics of $TACS^{UT}$). It is easy to see that, with respect to the operational semantics of CCS, $p \xrightarrow{a} p'$ for some p' such that $r = \underline{p}'$. It follows from $p \xleftrightarrow{b} q$ that $q \xrightarrow{a} q'$ (with respect to the operational semantics of CCS) for some q' such that $p' \xleftrightarrow{b} q'$. It is now a simple matter to see that $q \xrightarrow{a} q'$ with respect to the operational semantics of $TACS^{UT}$. Finally, $r = \underline{p}' R \underline{q}'$ holds by the definition of R .

Furthermore, if $\underline{p} \xrightarrow{\epsilon(1)} \underline{p}'$, it follows from the above item that $\underline{p} \equiv \underline{p}'$. Again using the above item, we have that $\underline{q} \xrightarrow{\epsilon(1)} \underline{q}$ and, by assumption, $\underline{p} R \underline{q}$. It also follows immediately from $p \xleftrightarrow{b} q$ that $\mathcal{U}(\underline{p}) = \mathcal{U}(\underline{q})$, and we are done since the relation R is symmetric.

Since all the provisos of Theorem 1 are met, Theorem 9 follows.

4.6 Other Timed Calculi, Equivalences and Preorders

There are many other timed extensions of CCS in the literature, and each of these languages comes equipped with notions of behavioural equivalence and/or preorder. In this section, we introduce a couple of the resulting process algebras studied in the research literature, and give the appropriate reductions to prove their non-finite axiomatizability using Theorem 1. Since the proofs of the provisos of Theorem 1 are almost identical to those in the previous two subsections, we dispense with them in this subsection.

TACS^{LT} and the MT-preorder. In [25], Lüttgen and Vogler introduced the language $TACS^{LT}$, which is syntactically the same as DiTCCS, but its only delay prefixing operator is $\epsilon(1).\dots$ Semantically, unlike DiTCCS, $TACS^{LT}$ does not implement the so-called *maximal progress* and allows for time delays for τ -prefixing, just like any other action prefixing. To obtain the SOS rules for $TACS^{UT}$, one must take the SOS rules of DiTCCS presented in Table 1, fix $d = 1$ in the rules for delay transitions, remove rules (**td1**) and (**td2**), which do not apply when one only considers unit-delay transitions, and replace symbol α in deduction rule (**ta**) by μ . This means that $\epsilon(1).P$ indicates a delay of *at least one time unit* before the execution of P . Hence, like in DiTCCS and contrary to the situation in $TACS^{UT}$, $\epsilon(1).a \xrightarrow{a}$. As argued by Lüttgen and Vogler, $TACS^{LT}$ is a calculus that is suitable for the study of lower bounds on the execution speed of processes.

In this part of the paper, we refer to the signature of $TACS^{LT}$ as Σ_e since we use this language as our source language in applying Theorem 1.

The notion of preorder that is considered over $TACS^{LT}$ in [25] is the MT-preorder due to Moller and Tofts [33].

Definition 8. *The relation $\overset{\rhd}{\sim}_{MT}$, called MT-preorder, is the largest relation such that, for all $p, q \in \mathcal{C}(\Sigma_e)$, $p \overset{\rhd}{\sim}_{MT} q$ iff for all $p' \in \mathcal{C}(\Sigma_e)$, and action μ ,*

1. *if $p \xrightarrow{\mu} p'$, then there exist a $k \geq 0$ and $q', p'' \in \mathcal{C}(\Sigma_e)$ such that $q \xrightarrow{\epsilon(1)k} \mu \xrightarrow{\mu} q'$, $p' \xrightarrow{\epsilon(1)k} p''$ and $p'' \overset{\rhd}{\sim}_{MT} q'$,*
2. *if $p \xrightarrow{\epsilon(1)} p'$, then there exists a $q' \in \mathcal{C}(\Sigma_e)$ such that $q \xrightarrow{\epsilon(1)} q'$ and $p' \overset{\rhd}{\sim}_{MT} q'$,*
3. *if $q \xrightarrow{\mu} q'$, then there exists a $p' \in \mathcal{C}(\Sigma_e)$ such that $p \xrightarrow{\mu} p'$ and $p' \overset{\rhd}{\sim}_{MT} q'$, and*
4. *if $q \xrightarrow{\epsilon(1)} q'$, then there exists a $p' \in \mathcal{C}(\Sigma_e)$ such that $p \xrightarrow{\epsilon(1)} p'$ and $p' \overset{\rhd}{\sim}_{MT} q'$.*

As shown in [24, Theorem 2], the MT-preorder is a precongruence over $TACS^{LT}$. Moreover, $\overset{\rhd}{\sim}_{MT}$ coincides with strong bisimilarity over CCS terms. It follows that the family of equations \mathcal{M} is sound modulo $\overset{\rhd}{\sim}_{MT}$. These observations pave the way to the following result.

Theorem 10. *$TACS^{LT}$ affords no finite ground-complete axiomatization modulo $\overset{\rhd}{\sim}_{MT}$.*

The above result can, once more, be proved by instantiating Theorem 1. The function $\hat{\cdot}$ from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$ is identical to the same function for $TACS^{UT}$, defined in Section 4.5, if one removes the underlinings under the action and delay prefixes. It is not hard to show that $\hat{\cdot}$ is an \mathcal{M} -reflecting structural reduction, from which Theorem 10 follows.

In proving that $\hat{\cdot}$ is a reduction, the following lemma is used.

Lemma 8.

1. *For all $p \in \mathcal{C}(\Sigma_e)$ and $r \in \mathcal{C}(\Sigma_o)$, if $\hat{p} \xrightarrow{\alpha} r$ with respect to the operational semantics of CCS, then there exist some $k \geq 0$ and $p', p'' \in \mathcal{C}(\Sigma_e)$ such that $p \xrightarrow{\epsilon(1)k} p' \xrightarrow{\alpha} p''$ with respect to the operational semantics of $TACS^{LT}$ and $\widehat{p'} \equiv r$.*
2. *For all $p, p', p'' \in \mathcal{C}(\Sigma_e)$ and $k \geq 0$, if $p \xrightarrow{\epsilon(1)k} p' \xrightarrow{\alpha} p''$ with respect to the operational semantics of $TACS^{LT}$, then $\widehat{p} \xrightarrow{\alpha} \widehat{p''}$ with respect to the operational semantics of CCS.*

TACS and Urgent Timed Bisimulation. In [26], $TACS^{UT}$ and $TACS^{LT}$ were combined to obtain $TACS$. In this calculus, the underlined prefixing operators, inherited from $TACS^{UT}$, are used to model potentially urgent actions and upper time bounds on action occurrences. The non-underlined prefixing operators, inherited from $TACS^{LT}$, are used to model lazy actions and lower time bounds on action occurrences.

In this part of the paper, we refer to the signature of $TACS$ as Σ_e since we use this language as our source language in applying Theorem 1.

The rules for the operational semantics of $TACS$ are just a combination of those for $TACS^{UT}$ and $TACS^{LT}$. Finally, the set $\mathcal{U}(p)$ of *urgent actions* of a $TACS$ process p is defined by structural induction on processes in [26, Table 2, page 212]. The key clauses in such a definition are as follows:

$$\begin{aligned} \mathcal{U}(\underline{\mu}.p) &= \{\mu\} \\ \mathcal{U}(\underline{\epsilon(1)}.p) &= \mathcal{U}(\epsilon(1).p) = \mathcal{U}(\mu.p) = \emptyset . \end{aligned}$$

This is in agreement with the intuition that $\underline{\mu}.p$ indicates the potential urgency of initial action μ , whereas that action is lazy in any of the other prefixing contexts in $TACS$.

The notion of equivalence that is used for the full $TACS$ calculus in [26] is *urgent timed bisimilarity*.

Definition 9. *A symmetric relation $R \subseteq \mathcal{C}(\Sigma_e) \times \mathcal{C}(\Sigma_e)$ is an urgent timed bisimulation when for all $(p, q) \in R$ and $p' \in \mathcal{C}(\Sigma_e)$,*

1. for each $\mu \in A \cup \bar{A} \cup \{\tau\}$, if $p \xrightarrow{\mu} p'$ then there exists a $q' \in \mathcal{C}(\Sigma_e)$ such that $q \xrightarrow{\mu} q'$ and $(p', q') \in R$,
2. if $p \xrightarrow{\epsilon(1)} p'$ then $\mathcal{U}(q) \subseteq \mathcal{U}(p)$ and there exists a $q' \in \mathcal{C}(\Sigma_e)$ such that $q \xrightarrow{\epsilon(1)} q'$ and $(p', q') \in R$.

Urgent timed bisimilarity is the largest urgent timed bisimulation.

As shown in [26], urgent timed bisimilarity is a congruence over *TACS*. Moreover, urgent timed bisimilarity coincides with strong bisimilarity over CCS terms. It follows that the family of equations \mathcal{M} is sound modulo urgent timed bisimilarity.

Theorem 11. *TACS affords no finite ground-complete axiomatization modulo urgent timed bisimilarity.*

The above result can, again, be proved by instantiating Theorem 1. The function $\hat{\cdot}$ from $\mathcal{T}(\Sigma_e)$ to $\mathcal{T}(\Sigma_o)$ is just a combination of the reductions for *TACS*^{UT} and *TACS*^{LT}, and is given below for the sake of completeness.

$$\begin{array}{l} \hat{\mathbf{0}} = \mathbf{0} \\ \widehat{a.t} = \widehat{a}.t = a.\widehat{t} \quad \widehat{\mu.t} = \widehat{\mu}.t = \mathbf{0} \text{ for } \mu \neq a \\ \widehat{\epsilon(1).t} = \epsilon(1).\widehat{t} = \widehat{t} \quad \widehat{t+u} = \widehat{t} + \widehat{u} \quad \widehat{t \parallel u} = \widehat{t} \parallel \widehat{u} \end{array}$$

It is not hard to show that $\hat{\cdot}$ is an \mathcal{M} -reflecting structural reduction, from which Theorem 10 follows. In proving that $\hat{\cdot}$ is a reduction, we use the extension of Lemma 8 to *TACS*.

4.7 Interactive Markov Chains and Markovian Bisimilarity

In [22], Hermanns presented the calculus of Interactive Markov Chains (IMC) to model and reason about stochastic processes. The syntax of IMC (modulo minor notational changes) is given below:

$$P ::= \mathbf{0} \mid \mu.P \mid \lambda.P \mid P + P \mid P \parallel_S P \text{ ,}$$

where $\mu \in A \cup \{\tau\}$, $\lambda \in \mathbb{R}^{\geq 0}$ (we use $\mathbb{R}^{\geq 0}$ to denote the set of non-negative real numbers) and $S \subseteq A$. Here $\mathbf{0}$ stands, as usual, for an inactive process. For each $\mu \in A \cup \{\tau\}$, following Milner, $\mu.P$ represents action prefixing. On the other hand, $\lambda.P$, with $\lambda \in \mathbb{R}^{\geq 0}$, is rate prefixing, meaning that before proceeding with P there is a delay drawn from a negative exponentially-distributed random variable with rate λ . Nondeterministic choice has its usual interpretation. IMC uses a CSP-like scheme for parallel composition, i.e., in $P \parallel_S Q$, the two processes run in parallel, but must synchronize on actions in S . In this subsection, we denote the signature of IMC by Σ_e .

The operational semantics of IMC is given by the following deduction rules, where β ranges over $A \cup \{\tau\} \cup \mathbb{R}^{\geq 0}$.

$$\begin{array}{l} (\beta) \frac{}{\beta.x \xrightarrow{\beta} x} \quad (\mathbf{ic0}) \frac{x_0 \xrightarrow{\beta} y}{x_0 + x_1 \xrightarrow{\beta} y} \quad (\mathbf{ic1}) \frac{x_1 \xrightarrow{\beta} y}{x_0 + x_1 \xrightarrow{\beta} y} \\ (\mathbf{ip0}) \frac{x_0 \xrightarrow{\beta} y_0}{x_0 \parallel_S x_1 \xrightarrow{\beta} y_0 \parallel_S x_1} \beta \notin S \quad (\mathbf{ip1}) \frac{x_1 \xrightarrow{\beta} y_1}{x_0 \parallel_S x_1 \xrightarrow{\beta} x_0 \parallel_S y_1} \beta \notin S \quad (\mathbf{ip2}) \frac{x_0 \xrightarrow{\beta} y_0 \quad x_1 \xrightarrow{\beta} y_1}{x_0 \parallel_S x_1 \xrightarrow{\beta} y_0 \parallel_S y_1} \beta \in S \end{array}$$

As discussed in, e.g., [22, Page 91], the above rules define a transition relation $\xrightarrow{\mu}$, for each action μ . On the other hand, the rules should be read as defining a *multi-relation* when $\beta \in \mathbb{R}^{\geq 0}$. This ensures, for instance, that *two* λ -labeled transitions are generated for any process of the form $\lambda.p + \lambda.p$, which should be equivalent to $2\lambda.p$. We refer the reader to [22] for a thorough discussion of this issue, which, however, will be immaterial in the technical developments to follow.

For each closed term p and set of closed terms C , we define

$$\gamma(p, C) = \sum \{ \lambda \mid \lambda \in \mathbb{R}^{\geq 0}, p \xrightarrow{\lambda} p' \text{ and } p' \in C \} \text{ .}$$

Note that

$$\gamma(\lambda.p + \lambda.p, \{p\}) = 2\lambda = \gamma(2\lambda.p, \{p\}) ,$$

since $\xrightarrow{\lambda}$ is a multi-relation.

A notion of behavioural equality for IMC is *strong Markovian bisimilarity* [22, Definition 5.2.1], which is defined below for the sake of completeness.

Definition 10. *An equivalence relation $R \subseteq \mathcal{C}(\Sigma_e) \times \mathcal{C}(\Sigma_e)$ is a strong Markovian bisimulation when for all $(p, q) \in R$, $p' \in \mathcal{C}(\Sigma_e)$, and $\mu \in A \cup \{\tau\}$,*

1. *if $p \xrightarrow{\mu} p'$, then there exists a $q' \in \mathcal{C}(\Sigma_e)$ such that $q \xrightarrow{\mu} q'$ and $(p', q') \in R$, and*
2. *if $p \xrightarrow{\tau}$, then $\gamma(p, C) = \gamma(q, C)$ for each equivalence class $C \in \mathcal{C}(\Sigma_e)/R$.*

Processes $p, q \in \mathcal{C}(\Sigma_e)$ are strongly Markovian bisimilar, denoted by $p \xleftrightarrow{m} q$, when there exists a strong Markovian bisimulation R such that $(p, q) \in R$.

Note that, in the light of condition 2 in the above definition, delay rates are irrelevant for processes that afford τ -labelled transitions. For instance, $\tau.p + \lambda.q \xleftrightarrow{m} \tau.p$ for all closed IMC terms p and q . This condition in the definition strong Markovian bisimilarity implements a form of *maximal progress*.

As shown in [22], strong Markovian bisimilarity is a congruence over IMC. Moreover, in [22, Page 110] Hermanns gave a complete axiomatization of strong Markovian bisimilarity over a language that includes the fragment of IMC we consider here. That axiomatization, however, involves the use of an expansion-like law—axiom (X) on Table 5.6 in [22]. It is therefore natural to ask oneself whether the use of an axiom schema like (X) can be “simulated” by means of a finite collection of equations over the signature of IMC. We now show that it cannot, as stated in the following theorem.

Theorem 12. *Strong Markovian bisimilarity has no finite ground-complete axiomatization over the language IMC.*

The above theorem can be proved by applying Theorem 1. Since IMC uses CSP-style synchronization [23], we have to slightly adapt our basic theory of CCS in order to reduce IMC to it. Namely, we take a subset of CCS in which the only action is the internal action τ . As stated in Remark 4, Theorem 5 still holds true when τ is taken to be the only action in the signature of CCS. *In the remainder of this subsection, we use \mathcal{M} to denote the collection of Moller’s equations over action τ .*

In order to apply Theorem 1, we define the following translation from IMC terms, i.e., $\mathcal{T}(\Sigma_e)$, to CCS terms (with τ as the only action), i.e., $\mathcal{T}(\Sigma_o)$.

$$\begin{array}{ll} \widehat{\mathbf{0}} = \mathbf{0} & \widehat{x} = x \\ \widehat{\tau.p} = \tau.\widehat{p} & \widehat{\beta.p} = \mathbf{0} \text{ for } \beta \neq \tau, \beta \in A \cup \{\tau\} \cup \mathbb{R}^{\geq 0} \\ \widehat{p + q} = \widehat{p} + \widehat{q} & \widehat{p \parallel_S q} = \widehat{p} \parallel \widehat{q} \end{array}$$

Lemma 9. *The mapping $\widehat{\cdot}$ defined above is structural.*

Consider now the set of Moller’s equations \mathcal{M} , which are sound over CCS modulo bisimilarity. In the light of Theorem 1, in order to prove that strong Markovian bisimilarity is not finitely based over IMC, it suffices only to show the following statements:

1. $t \xleftrightarrow{m} u \Rightarrow \widehat{t} \xleftrightarrow{b} \widehat{u}$, for each $t, u \in \mathcal{T}(\Sigma_e)$, and
2. $\widehat{\cdot}$ is \mathcal{M} -reflecting.

Note that $E \vdash t = u \Rightarrow \widehat{E} \vdash \widehat{t} = \widehat{u}$ holds by Theorem 2 since $\widehat{\cdot}$ is structural (Lemma 9).

Once we prove the above claims, Theorem 12 follows as a corollary of Theorem 1. The proof of the former claim is very similar to the proof of the same statement in the case of DiTCCS, and uses the following lemma.

Lemma 10.

1. Assume that $\widehat{p} \xrightarrow{\tau} r$ holds with respect to the operational semantics of CCS for some closed IMC term p and CCS term r . Then $p \xrightarrow{\tau} p'$ holds with respect to the operational semantics of IMC for some closed IMC term p' such that $\widehat{p}' = r$.
2. If $p \xrightarrow{\tau} p'$ holds with respect to the operational semantics of IMC for some closed IMC terms p, p' then $\widehat{p} \xrightarrow{\tau} \widehat{p}'$ holds with respect to the operational semantics of CCS.

Remark 6. Statement 2 in the above lemma would *not* hold if we considered CCS terms and transitions over an action $a \neq \tau$, and we defined our reduction mapping for action prefixes as we did in the previous subsections. For example, $a.\mathbf{0} \parallel_S a\mathbf{0} \xrightarrow{a} \mathbf{0} \parallel_S \mathbf{0}$ when $a \in S$. However, $a.\mathbf{0} \parallel a\mathbf{0} \xrightarrow{a} \mathbf{0} \parallel \mathbf{0}$. This is the reason why we find it convenient to work with CCS over action τ in the proof of Theorem 12.

In order to establish the latter claim, it suffices only to show that all axioms $t = u$ in \mathcal{M} are sound modulo \leftrightarrow_m , once we replace all occurrences of \parallel with \parallel_\emptyset in CCS terms. Indeed, the statement then follows by taking the IMC terms t' and u' that are obtained from t and u , respectively, by replacing each occurrence of \parallel with \parallel_\emptyset .

Again, the proof is similar to that of similar claims in the previous sections, but a few details are slightly different. We prove the following two statements, where, for each $p \in \mathcal{C}(\Sigma_o)$, p_\emptyset is the IMC term obtained from p by replacing each occurrence of \parallel with \parallel_\emptyset .

1. For each $p \in \mathcal{C}(\Sigma_o)$ and non-negative real number λ , $p_\emptyset \xrightarrow{\lambda}$. This means that $\gamma(p_\emptyset, C) = 0$ for each $C \in \mathcal{C}(\Sigma_e)/\leftrightarrow_m$. (This claim is easily established by structural induction on p .)
2. For all $p, q \in \mathcal{C}(\Sigma_o)$, if $p \leftrightarrow_b q$ then $p_\emptyset \leftrightarrow_m q_\emptyset$. (This follows because the relation

$$R = \{(p_\emptyset, q_\emptyset) \mid p \leftrightarrow_b q \text{ and } p, q \in \mathcal{C}(\Sigma_o)\} \cup \{(r, r) \mid r \in \mathcal{C}(\Sigma_e)\}$$

is easily seen to be a strong Markovian bisimulation. Note that R is an equivalence relation over $\mathcal{C}(\Sigma_e)$.)

Since all the provisos of Theorem 1 are met, Theorem 12 follows.

5 Limitations of Our Approach

As witnessed by the applications described in the previous section, our reduction-based method for proving non-finite axiomatizability results, based on Theorem 1, is widely applicable. Moreover, in all the applications of Theorem 1 we presented in Section 4, we used CCS modulo bisimilarity as our target language for an \mathcal{M} -reflecting reduction. In this section, we give an example of an equational theory within the realm of classic process algebra, whose non-finite axiomatizability cannot be shown in that fashion.

5.1 CCS_Ω and Prebisimilarity

Consider the following syntax for the language CCS_Ω (a variant of the calculus presented in [7]).

$$P ::= \mathbf{0} \mid \Omega \mid a.P \mid P + P \mid P \parallel P$$

The operational semantics of CCS_Ω is given by two ingredients: \xrightarrow{a} transitions, which are defined by the same deduction rules as in CCS (thus, Ω has no outgoing transitions), and a *convergence predicate* \downarrow , which is the least predicate over closed CCS_Ω satisfying the rules given below.

$$\frac{}{\mathbf{0} \downarrow} \quad \frac{}{a.p \downarrow} \quad \frac{p \downarrow \quad q \downarrow}{p + q \downarrow} \quad \frac{p \downarrow \quad q \downarrow}{p \parallel q \downarrow}$$

We write $p \uparrow$ if it does not hold that $p \downarrow$. So, for instance, $a.\Omega \downarrow$, but $\Omega \uparrow$ and $a \parallel \Omega \uparrow$.

The following notion of prebisimilarity is a relevant notion of behavioural preorder in the presence of divergence as adopted in, e.g., [1, 7, 8, 20, 27]. We refer the interested reader to those papers and the references therein for a wealth of results on the semantic theory of CCS_Ω modulo prebisimilarity.

Definition 11. *The relation $\overset{\sqsubseteq}{\sim}_{pre}$ is the largest relation over the closed terms of CCS_Ω satisfying the following clauses, whenever $p \overset{\sqsubseteq}{\sim}_{pre} q$,*

1. for each p' , if $p \xrightarrow{a} p'$ then there exists a q' such that $q \xrightarrow{a} q'$ and $p' \overset{\sqsubseteq}{\sim}_{pre} q'$;
2. if $p \downarrow$, then
 - (a) $q \downarrow$ and
 - (b) for each q' , if $q \xrightarrow{a} q'$, then there exists a p' such that $p \xrightarrow{a} p'$ and $p' \overset{\sqsubseteq}{\sim}_{pre} q'$.

The relation $\overset{\sqsubseteq}{\sim}_{pre}$ is a preorder and a precongruence over closed CCS_Ω terms.

Before, we proceed with our results in this section, we introduce an auxiliary definition capturing the concept of “convergence along all traces” of a process execution, which we call strong convergence.

Definition 12. *The set of strongly convergent processes is the largest set of closed CCS_Ω terms such that, for each strongly convergent p , it holds that $p \downarrow$, and p' is also strongly convergent whenever $p \xrightarrow{a} p'$.*

The following lemma will be useful in the technical developments in the rest of this section.

Lemma 11. *The following statements hold.*

1. For each CCS_Ω term t , it holds that $\Omega \overset{\sqsubseteq}{\sim}_{pre} t$.
2. For each two closed CCS_Ω terms p and q , if $p \overset{\sqsubseteq}{\sim}_{pre} q$ and p is strongly convergent, then q is also strongly convergent.
3. Each closed CCS_Ω term p is strongly convergent if and only if it is a CCS term—that is, if it does not contain occurrences of Ω .
4. For each two CCS terms t and u , $t \overset{\sqsubseteq}{\sim}_{pre} u$ if and only if $t \leftrightarrow_b u$.

5.2 Non-finite Axiomatizability of $\overset{\sqsubseteq}{\sim}_{pre}$ and Limitations of our Proof Methodology

Our order of business in this section will be to show that the inequational theory of CCS_Ω modulo prebisimilarity is not finitely axiomatizable. Moreover, we prove that CCS_Ω modulo prebisimilarity affords no finite ground-complete axiomatization. Afterwards, we prove that our meta-theorem comes short of providing a proof for that result if we use CCS modulo bisimilarity as the target language of our reduction.

Theorem 13. *CCS_Ω affords no finite sound and ground-complete axiomatization modulo $\overset{\sqsubseteq}{\sim}_{pre}$.*

Proof. Assume, towards a contradiction, that E is a finite collection of sound axioms over CCS_Ω modulo $\overset{\sqsubseteq}{\sim}_{pre}$, and that E is ground-complete. It follows from Lemma 11(4) that Moller’s set of equations \mathcal{M} is sound modulo $\overset{\sqsubseteq}{\sim}_{pre}$. Thus, we have that $E \vdash \mathcal{M}$. Moreover, we may assume that each of the proofs of equations in \mathcal{M} from E only mentions closed terms.

Let E' be the set of axioms contained in E that are used in the proofs of equations in \mathcal{M} from E .

Claim. Each axiom in E' is an equation over the signature of CCS that is sound modulo \leftrightarrow_b .

Using this claim, a contradiction to Theorem 5 follows, since we have found a finite set of sound CCS axioms, namely E' , which can prove all the equations in \mathcal{M} .

To prove the above claim, observe that all the process terms involved in a proof of an equation in \mathcal{M} are strongly convergent (Lemma 11(2)). It follows that if $t \leq u$ is contained in E' , then Ω does not occur in t and u (Lemma 11(3)). Therefore t and u are CCS terms, and the equation $t = u$ is sound over CCS modulo strong bisimilarity, since $t \leq u$ is sound over CCS_Ω modulo $\stackrel{\Xi}{\sim}_{pre}$ and CCS is a sub-language of CCS_Ω (Lemma 11(4)).

We may therefore conclude that CCS_Ω affords no finite sound and ground-complete axiomatization modulo $\stackrel{\Xi}{\sim}_{pre}$. \square

It is natural to wonder whether the above result can be established, like all those we presented in Section 4, by using CCS modulo bisimilarity as our target language for an \mathcal{M} -reflecting reduction. The following theorem shows that this is not possible.

Theorem 14. *There is no \mathcal{M} -reflecting reduction from CCS_Ω modulo $\stackrel{\Xi}{\sim}_{pre}$ to CCS modulo strong bisimilarity.*

Proof. Consider a reduction $\hat{\cdot}$ from CCS_Ω modulo $\stackrel{\Xi}{\sim}_{pre}$ to CCS modulo strong bisimilarity. This reduction is a constant function modulo strong bisimilarity over CCS_Ω terms. To see this, recall that, for each CCS_Ω term t , it holds that $\Omega \stackrel{\Xi}{\sim}_{pre} t$ (Lemma 11(1)). Since reductions preserve the validity of inequations (see Definition 1), we have that $\hat{\Omega}$ is strongly bisimilar to \hat{t} , for each CCS_Ω term t . This means that $\hat{\cdot}$ is indeed a constant function modulo strong bisimilarity, as claimed. A constant function modulo strong bisimilarity is *not* \mathcal{M} -reflecting since $a^1 \parallel (a^1 + a^2 + \dots + a^n)$ and $a^1 \parallel (a^1 + a^2 + \dots + a^m)$ are not strongly bisimilar when $n \neq m$. \square

So far, we have been unable to find a proof of Theorem 13 using our reduction technique.

Remark 7. The attentive reader may have already observed that a stronger result than Theorem 14 does in fact hold. Indeed, let E be a set of CCS equations that are sound modulo strong bisimilarity, and contain two equations $p_1 = q_1$ and $p_2 = q_2$, where p_1 and p_2 are not strongly bisimilar. Then there is no E -reflecting reduction from CCS_Ω modulo $\stackrel{\Xi}{\sim}_{pre}$ to CCS modulo strong bisimilarity. Indeed, as indicated in the proof of Theorem 14, any reduction $\hat{\cdot}$ from CCS_Ω modulo $\stackrel{\Xi}{\sim}_{pre}$ to CCS modulo strong bisimilarity is a constant function modulo strong bisimilarity over CCS_Ω terms. If $\hat{\cdot}$ were E -reflecting, then

$$p_1 = \hat{t}_1 \leftrightarrow_b \hat{t}_2 = p_2 \quad ,$$

for some CCS_Ω terms t_1 and t_2 . This contradicts our assumption that p_1 and p_2 are not strongly bisimilar.

6 Conclusions

In this paper, we have proposed a meta-theorem for proving non-finite axiomatizability results. This theorem can be used to show such results when there exists a reduction from the calculus under consideration to a calculus for which non-finite axiomatizability is known. If the reduction is defined structurally (in the sense of Definition 3), then one only needs to prove that the reduction preserves sound (in)equalities and that it reflects a set of “difficult” (in)equations that form the core of the non-finite axiomatizability result over the target calculus. We have shown seven new non-finite axiomatizability results in process algebra by applying our meta-theorem and reducing different calculi (modulo their respective notion of equivalence or preorder) to a subset of CCS. We intend to apply our reduction technique to obtain several other new non-finite axiomatizability results in process algebra.

The above-mentioned conditions on the reductions can be established following similar lines for the different calculi and different notions of (pre)congruence studied in this paper. The resulting proofs are substantially more concise and simpler than typical proofs of non-finite axiomatizability.

We believe that the proofs of the aforementioned two conditions can be further simplified if one commits to particular models such as those given by Plotkin-style SOS rules. A promising future research direction is to study whether one can apply our meta-theorem in conservative and orthogonal language extensions. Using the SOS meta-theory, one can seek sufficient syntactic conditions on the reduction function that would automatically provide us with the properties required by our meta-theorem.

Furthermore, in this paper, we pointed out a limitation of our meta-theorem by presenting a non-finite axiomatizability result that cannot be proved using our general strategy of reducing calculi to CCS. Studying the roots of such limitations may lead to improvements upon the meta-theorem presented in this paper.

References

1. S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92(1):161–218, 1991.
2. L. Aceto, T. Chen, W. Fokkink, and A. Ingolfsdottir. On the axiomatizability of priority. Proceedings of Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Part II, volume 4052 of *LNCS*, pages 480–491, Springer, 2006.
3. L. Aceto, W. Fokkink and A. Ingolfsdottir. Ready to preorder: Get your BCCSP axiomatization for free! Proceedings of CALCO’07, volume 4624 of *LNCS*, pages 338–367. Springer, 2007.
4. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. CCS with Hennessy’s merge has no finite equational axiomatization. *Theoretical Computer Science*, 330(3):377–405, 2005.
5. 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 *LNCS*, pages 338–367. Springer, 2005.
6. L. Aceto, W. Fokkink, A. Ingolfsdottir, and S. Nain. Bisimilarity is not finitely based over BPA with interrupt. *Theoretical Computer Science*, 366(1–2):60–81, 2006.
7. L. Aceto and M. Hennessy. Termination, deadlock, and divergence. *Journal of the ACM*, 39(1):147–187, 1992.
8. L. Aceto, A. Ingolfsdottir. CPO models for compact GSOS languages. *Information and Computation*, 129(1):107–141, 1996.
9. L. Aceto, A. Ingolfsdottir and M. R. Mousavi. Impossibility results for the equational theory of Timed CCS. Proceedings of CALCO’07, volume 4624 of *LNCS*, pages 80–95, Springer, 2007.
10. L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, pages 197–292. North-Holland, 2001.
11. L. Aceto, W. Fokkink and C. Verhoef. Conservative extension in structural operational semantics. In G. Paun, G. Rozenberg and A. Salomaa, eds., *Current Trends in Theoretical Computer Science - Entering the 21st Century*, pages 504–524, World Scientific, 2001.
12. J.C.M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.
13. J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1–3):109–137, 1984.
14. S.C.C. Blom, W.J. Fokkink, J.F. Groote, I. van Langevelde, B. Lisser, and J.C. van de Pol. mCRL: A toolset for analysing algebraic specifications. Proceedings of CAV’01, volume 2102 of *LNCS*, pages 250–254, Springer, 2001.
15. S. L. Bloom and Z. Ésik. Nonfinite axiomatizability of shuffle inequalities. Proceedings of TAP-SOFT’95, volume 915 of *LNCS*, pages 318–333. Springer, 1995.
16. Z. Ésik and M. Bertol. Nonfinite axiomatizability of the equational theory of shuffle. *Acta Informatica*, 35(6):505–539, 1998.
17. W.J. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Information and Computation*, 146(1):24–54, 1998.
18. J. F. Groote, A. Mathijssen, M. Reniers, Y. Usenko, and M. van Weerdenburg. The formal specification language mCRL2. In *Proceedings of the Dagstuhl Seminar*, 2007. Available from <http://www.mcr12.org>.
19. C. Gunter. *Semantics of Programming Languages: Structures and Techniques*. Foundations of Computing Series. MIT Press, 1992.

20. M. Hennessy. A term model for synchronous processes. *Information and Control*, 51(1):58–75, 1981.
21. M. Hennessy. *Algebraic Theory of Processes*. Foundations of Computing Series. MIT Press, 1988.
22. H. Hermans. *Interactive Markov Chains and the Quest for Quantified Quality*. Volume 2428 of LNCS, Springer, 2002.
23. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
24. G. Lüttgen and W. Vogler. Bisimulation on speed: worst-case efficiency. *Information and Computation*, 191(2): 105–144, 2004.
25. G. Lüttgen and W. Vogler. Bisimulation on speed: Lower time bounds. *Proceedings of FOSSACS'04*, volume 2987 of LNCS, Springer, 2004.
26. G. Lüttgen and W. Vogler. Bisimulation on speed: A unified approach. *Theoretical Computer Science*, 360(1–3):209–227, 2006.
27. R. Milner. A modal characterisation of observable machine. *Proceedings of CAAP'81*, volume 112 of LNCS, pages 25–34, Springer, 1981.
28. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
29. F. Moller. *Axioms for Concurrency*. Ph.D. Thesis, University of Edinburgh, 1989.
30. F. Moller. The nonexistence of finite axiomatisations for CCS congruences. In *Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 142–153. IEEE Computer Society, 1990.
31. F. Moller. The importance of the left merge operator in process algebras. In *Proceedings of ICALP'90*, volume 443 of LNCS, pages 752–764. Springer, 1990.
32. F. Moller and C.M.N. Tofts. A temporal calculus of communicating systems. In *Proceedings of CONCUR'90*, volume 458 of LNCS, pages 401–415. Springer, 1990.
33. F. Moller and C. Tofts. Relating processes with respect to speed. In *Proceedings of CONCUR '91*, volume 527 of LNCS, pages 424–438. Springer, 1991.
34. M.R. Mousavi and M.A. Reniers. Orthogonal extensions in structural operational semantics. Proceedings of ICALP'05, volume 3580 of LNCS, pages 1214–1225, Springer, 2005.
35. X. Nicollin, J. Sifakis. The algebra of timed processes, ATP: Theory and application. *Information and Computation*, 114(1):131–178, 1994.
36. G. D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60–61:17–139, 2004.
37. M. Sipser. *Introduction to the Theory of Computation*. 2nd Ed., PWS Publishing, 2005.
38. W. Yi. Real-time behaviour of asynchronous agents. In *Proceedings of CONCUR'90*, volume 458 of LNCS, pages 502–520. Springer, 1990.