Non-finite Axiomatisability Results via Reductions: CSP Parallel Composition and CCS Restriction

Luca Aceto[1,2](✉), Elli Anastasiadi[1], Valentina Castiglioni[1], and Anna Ingólfsdóttir[1]

[1] ICE-TCS, Department of Computer Science, Reykjavik University, Reykjavik, Iceland
luca@ru.is

[2] Gran Sasso Science Institute, L'Aquila, Italy

Abstract. This paper studies the existence of finite, ground-complete axiomatisations of CSP-like parallel composition operators, and the restriction operator from CCS, modulo bisimilarity. More specifically, we build on Moller's result to the effect that bisimilarity does not have a finite, ground-complete equational axiomatisation over a minimal fragment of CCS, and we use a reduction technique by Aceto et al. to lift it to various extensions of BCCSP with CSP-like parallel operators, and to the recursion and relabelling free fragment of CCS.

1 Introduction

Some of Frits Vaandrager's early seminal contributions were firmly rooted in the theory and applications of process algebras and their semantics. Having been brought up in the tradition of Bergstra and Klop's Algebra of Communicating Processes (ACP) [14,16,17], Frits Vaandrager studied semantic models of algebraic process description languages [25,26], equational axiomatisations of process equivalences [15] and their application in verification (see, for instance, [24,39]). Moreover, together with Aceto and Bloom, in [3] he initiated the study of methods for generating finite, ground-complete, equational axiomatisations of bisimilarity [32,36] from operational specifications given in the GSOS format [18]. The techniques proposed in [3] can be used to synthesise auxiliary operators, such as Bergstra and Klop's left- and communication-merge operators, that make finite axiomatisations possible and paved the way to several further studies in the literature—see, for instance, the developments presented in [10,21,27,31].

The use of auxiliary operators to obtain finite, equational, ground-complete axiomatisations of bisimilarity, even for very inexpressive process algebras, was justified by Moller in [33–35], where he showed that bisimilarity has no finite axiomatisation

This work has been partially supported by the project "*Open Problems in the Equational Logic of Processes*" (OPEL) of the Icelandic Research Fund (grant No. 196050-051). E. Anastasiadi has been supported by the project "*Runtime and Equational Verification Of Concurrent Programs*" (REVOCOP) of the Reykjavik University Research Fund (grant No. 222021-051). V. Castiglioni has been supported by the project "*Programs in the wild: Uncertainties, adaptabiLiTy and veRificatiON*" (ULTRON) of the Icelandic Research Fund (grant No. 228376-051).

N. Jansen et al. (Eds.): Vaandrager Festschrift 2022, LNCS 13560, pp. 1–26, 2022.
https://doi.org/10.1007/978-3-031-15629-8_1

over minimal fragments of Milner's Calculus of Communicating Systems (CCS) [32] and Bergstra and Klop's ACP. (Henceforth, we will consider the recursion, relabelling and restriction free fragment of CCS, which, for simplicity, we still denote as CCS.) Moller's above-mentioned, path-breaking, negative results have been followed by a wealth of research on non-finitely-based fragments of process algebras—see, for instance, [1,2,4–8,12,20].

Our Contribution. In this paper, we celebrate Frits Vaandrager's early contributions to the study of algebraic process description languages by answering the two questions that Rob van Glabbeek[1] asked the first author after his invited talk at LICS 2021[2]:

> Would Moller's non-finite axiomatisability result for CCS remain true if we replaced CCS parallel composition with the parallel operators from Hoare's Communicating Sequential Processes (CSP) [29]? And what if we added the restriction operator to CCS instead?

Our first contributions concern the existence of finite, ground-complete axiomatisations of bisimilarity over process algebras that extend the language BCCSP [22,28,32] with parallel operators from CSP. (BCCSP is a common fragment of Milner's CCS and Hoare's CSP suitable for describing finite process behaviour.)

For each set of actions A, the CSP parallel operator $|_A$ behaves like interleaving parallel composition for all actions that are not contained in A, but requires transitions of its operands that are labelled with some action $a \in A$ to synchronise. The result of such a synchronisation is an a-labelled transition of the composite parallel process, which can itself synchronise further with a-labelled steps from its environment. Therefore, unlike CCS parallel composition that is based on hand-shaking communication, the parallel operators from CSP support multi-way synchronisation and span the whole spectrum from pure interleaving parallel composition (the operator $|_\emptyset$) to synchronous composition (the operator $|_{\mathrm{Act}}$, where Act is the whole collection of actions that processes may perform).

We start our investigations by considering the languages $\mathrm{BCCSP}^{\mathrm{p}}_A$, which extend BCCSP with the parallel operator $|_A$ for some subset A of the whole set of actions Act, and $\mathrm{BCCSP}^{\mathrm{p}}_\tau$, which contains the parallel operator $|_A$ for each $A \subseteq \mathrm{Act}$, and the τ-prefixing operator for a distinguished action $\tau \notin \mathrm{Act}$. We show that Moller's non-finite axiomatisability result for bisimilarity still holds over $\mathrm{BCCSP}^{\mathrm{p}}_A$, when A is a strict subset of Act, and $\mathrm{BCCSP}^{\mathrm{p}}_\tau$. On the other hand, bisimilarity affords a finite, ground-complete axiomatisation over $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$.

The proofs of the above-mentioned negative results for $\mathrm{BCCSP}^{\mathrm{p}}_A$, when A is a strict subset of Act, and $\mathrm{BCCSP}^{\mathrm{p}}_\tau$ employ a reduction-based technique proposed in [11] for showing new, non-finite axiomatisability results over process algebras from already-established ones. In our setting, such reductions are translations from terms in the languages $\mathrm{BCCSP}^{\mathrm{p}}_A$ ($A \subset Act$) and $\mathrm{BCCSP}^{\mathrm{p}}_\tau$ to those in the fragment of CCS studied by Moller that

[1] Rob van Glabbeek was one of Frits Vaandrager's early collaborators and fellow doctoral student at CWI.

[2] See https://www.youtube.com/watch?v=2PxM3f0QWDM for a recording of that talk.

- preserve sound equations and equational provability over the source language, and
- reflect an infinite family of equations responsible for the non-finite axiomatisability of the target language.

No reduction from $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$ to CCS satisfying the former property modulo bisimilarity reflects Moller's family of equations witnessing his negative result over CCS. Therefore, the reduction technique cannot be applied to $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$. Indeed, we present a finite, ground-complete axiomatisation of bisimilarity over $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$.

We also show that, if we consider the language $\mathrm{BCCSP}^{\mathrm{p}}$, namely BCCSP with a parallel operator $|_A$ for each $A \subseteq \mathrm{Act}$, then no reduction that is structural, i.e. that does not introduce new variables and it is defined compositionally over terms, can reflect Moller's family of equations. However, we conjecture that bisimilarity does not admit a finite, ground-complete axiomatisation over $\mathrm{BCCSP}^{\mathrm{p}}$.

For our final contribution, we consider the language $\mathrm{CCS}^{\mathrm{r}}$, namely CCS enriched with *restriction operators* of the form $\cdot\backslash R$. Informally, $R \subseteq \mathrm{Act}$ is a set of actions that are restricted, meaning that the execution of a-labelled transitions (and of their "complementary actions") is prevented in $t\backslash R$ for all $a \in R$. By exploiting the reduction technique described above, we show that Moller's negative result can be lifted to $\mathrm{CCS}^{\mathrm{r}}$, giving thus that bisimilarity admits no finite, ground-complete axiomatisation over CCS with restriction.

Our contributions can then be summarised as follows:

1. We consider $\mathrm{BCCSP}^{\mathrm{p}}_{A}$, i.e., BCCSP enriched with *one* CSP-style parallel composition operator $|_A$, with $A \subset \mathrm{Act}$, and we show that, over that language, bisimilarity admits no finite, ground-complete axiomatisation (Theorem 4).
2. We consider $\mathrm{BCCSP}^{\mathrm{p}}$, i.e., BCCSP enriched with *all* CSP-style parallel composition operators $|_A$, and we show that there is no structural reduction from $\mathrm{BCCSP}^{\mathrm{p}}$ to CCS that can reflect the family of equations used by Moller to prove the negative result for bisimilarity over CCS (Theorem 5).
3. We consider $\mathrm{BCCSP}^{\mathrm{p}}_{\tau}$, i.e., $\mathrm{BCCSP}^{\mathrm{p}}$ enriched with the τ-prefixing, and we show that this algebra admits no finite, ground-complete axiomatisation modulo bisimilarity (Theorem 6).
4. We consider $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$, i.e., BCCSP enriched with the CSP-style parallel composition operator $|_{\mathrm{Act}}$, and we present a finite, ground-complete axiomatisation for it, modulo bisimilarity (Theorem 7).
5. We consider $\mathrm{CCS}^{\mathrm{r}}$, i.e., CCS with the restriction operator, and we show that bisimilarity has no finite, ground-complete axiomatisation over it (Theorem 8).

Organisation of Contents. In Sect. 2 we review basic notions on process semantics, behavioural equivalences, and equational logic. We also briefly recap Moller's negative result for bisimilarity over CCS. In Sect. 3 we give a bird's-eye view of the reduction technique from [11]. In Sect. 4, we present the lifting of Moller's negative result to $\mathrm{BCCSP}^{\mathrm{p}}_{A}$ (for $A \subset \mathrm{Act}$) and $\mathrm{BCCSP}^{\mathrm{p}}_{\tau}$, we study the case of $\mathrm{BCCSP}^{\mathrm{p}}$, and then we discuss the collapse of the negative result in the case of $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$. In Sect. 5, we use the reduction technique to prove the non-finite axiomatisability result for $\mathrm{CCS}^{\mathrm{r}}$. We conclude by discussing some directions for future work in Sect. 6.

2 Preliminaries

In this section we present some background notions on process algebras and equational logic. To make our contribution self-contained, we also briefly recap Moller's work on the nonexistence of finite axiomatisations modulo bisimilarity over the recursion, relabelling, and restriction free fragment of CCS (henceforth simply referred to as CCS).

Labelled Transition Systems and Bisimilarity. As semantic model for the algebraic process description languages that we will study, we consider classic *Labelled Transition Systems* [30].

Definition 1 (Labelled Transition System). *A* labelled transition system *(*LTS*) is a triple* $(\mathcal{P}, \mathrm{Act}, \rightarrow)$*, where* $\mathcal{P}$ *is a set of* processes *(or* states*),* Act *is a set of* actions *(or* labels*), and* $\rightarrow\ \subseteq \mathcal{P} \times \mathrm{Act} \times \mathcal{P}$ *is a (*labelled*)* transition relation.

In what follows, we assume that the set of actions Act is finite and non-empty. We let $p, q, \ldots$ range over $\mathcal{P}$, and $a, b, \ldots$ over Act. Moreover, as usual, we use $p \xrightarrow{a} p'$ in lieu of $(p, a, p') \in \rightarrow$. For each $p \in \mathcal{P}$ and $a \in \mathrm{Act}$, we write $p \xrightarrow{a}$ if $p \xrightarrow{a} p'$ holds for some p', and $p \not\xrightarrow{a}$ otherwise. For a sequence of actions $\rho = a_1 \cdots a_n$ $(n \geq 0)$, and processes p, p', we write $p \xrightarrow{\rho} p'$ if and only if there exists a sequence of transitions $p = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} p_n = p'$. If $p \xrightarrow{\rho} p'$ holds for some process p', then ρ is a *trace* of p. All the LTSs we will consider in this paper are finite and loop-free. The *depth* of a process p in such an LTS, denoted by $\mathrm{depth}(p)$, is then defined as the length of a longest trace of p.

Behavioural equivalences have been introduced as a tool to establish whether the behaviours of two processes are *indistinguishable for their observers*. In the literature we can find several notions of behavioural equivalence based on the observations that an external observer can make on a process. In this paper we consider the classic notion of *bisimilarity* [32, Chapter 4, Definition 1].

Definition 2 (Bisimilarity). *Let* $(\mathcal{P}, \mathrm{Act}, \rightarrow)$ *be a LTS. A binary symmetric relation* $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is a* bisimulation *if, and only if, whenever* $(p, q) \in \mathcal{R}$ *and* $p \xrightarrow{a} p'$ *then there exists a process* q' *such that* $q \xrightarrow{a} q'$ *and* $(p', q') \in \mathcal{R}$*. We say that* p *and* q *are* bisimilar *if there is a bisimulation relation* $\mathcal{R}$ *such that* $(p, q) \in \mathcal{R}$*.*

The union of all the bisimulation relations is called bisimilarity*, and denoted by* $\sim$*.*

It is well known that $\sim$ is an equivalence relation over $\mathcal{P}$, and it is the largest bisimulation relation [32, Chapter 4, Proposition 2].

Remark 1. Bisimilarity preserves the depth of processes, i.e., whenever $p \sim q$, then $\mathrm{depth}(p) = \mathrm{depth}(q)$.

The Language BCCSP. In this paper we will consider several algebraic process description languages, each characterised by the presence of a particular operator, or sets of operators. As all those languages are extensions of BCCSP [28], consisting of the basic operators from CCS [32] and CSP [29], in this section we use that language

Table 1. The SOS rules for BCCSP operators ($a \in \mathrm{Act}$).

$$\text{(act)}\ \frac{}{a.t \xrightarrow{a} t} \qquad \text{(lSum)}\ \frac{t \xrightarrow{a} t'}{t + u \xrightarrow{a} t'} \qquad \text{(rSum)}\ \frac{u \xrightarrow{a} u'}{t + u \xrightarrow{a} u'}$$

to introduce some general notions and notations on term algebras that will be useful throughout the remainder of the paper.

BCCSP terms are defined by the following grammar:

$$t ::= \quad \mathbf{0} \quad | \quad x \quad | \quad a.t \quad | \quad t + t, \tag{BCCSP}$$

where x is drawn from a countably infinite set of variables Var, a is an action from Act, $a.(\cdot)$ is the prefixing operator, defined for each $a \in \mathrm{Act}$, and $\cdot + \cdot$ is the nondeterministic choice operator. We shall use the meta-variables $t, u, \ldots$ to range over process terms, and write $\mathrm{var}(t)$ for the collection of variables occurring in the term t. The *size* of a term t, denoted by $\mathrm{size}(t)$, is the number of operator symbols in t. A term is *closed* if it does not contain any variables. Closed terms, or *processes*, will be denoted by $p, q, \ldots$. In particular, we denote the set of all BCCSP terms by $\mathbb{T}(\mathrm{BCCSP})$, and the set of closed BCCSP terms (or BCCSP processes) by $\mathcal{P}(\mathrm{BCCSP})$. This notation can be directly extended to all the languages that we will consider. Moreover, we omit trailing $\mathbf{0}$'s from terms and we use a *summation* $\sum_{i=1}^{k} t_i$ to denote the term $t = t_1 + \cdots + t_k$, where the empty sum represents $\mathbf{0}$. Henceforth, for each action $a \in \mathrm{Act}$ and natural number $m \geq 0$, we let a^0 denote $\mathbf{0}$ and a^{m+1} denote $a.(a^m)$.

We use the *Structural Operational Semantics* (SOS) framework [37] to equip processes with an operational semantics. The SOS rules (also called inference rules, or deduction rules) for the BCCSP operators given above are reported in Table 1. A (*closed*) *substitution* σ is a mapping from process variables to (closed) terms. Substitutions are extended from variables to terms, transitions, and rules in the usual way. Note that $\sigma(t)$ is closed, if so is σ. The inference rules in Table 1 allow us to derive valid transitions between closed BCCSP terms. The operational semantics for BCCSP is then modelled by the LTS whose processes are the closed terms in $\mathcal{P}(\mathrm{BCCSP})$, and whose labelled transitions are those that are provable from the SOS rules. The same approach will be applied to all the extensions of BCCSP that we will consider. The SOS rules of each language will be presented in the respective sections.

We call an equivalence relation a *congruence* over a language if it is compositional with respect to the operators of the language, i.e., the replacement of a component with an equivalent one does not affect the overall behaviour. Formally, the congruence property for bisimilarity over BCCSP, and its extensions, consists in verifying whether, given any n-ary operator f,

$$f(p_1, \ldots, p_n) \sim f(q_1, \ldots, q_n) \text{ whenever } p_i \sim q_i \text{ for all } i = 1, \ldots, n.$$

Since all the operators considered in this paper are defined by inference rules in the de Simone format [38], by [23, Theorem 4] we have that bisimilarity is a *congruence* over BCCSP and over all the languages that we will study.

Table 2. Rules of equational logic (f is any n-ary operator in L).

$$\frac{}{t \approx t} \qquad \frac{t_1 \approx t_2,\ t_2 \approx t_3}{t_1 \approx t_3} \qquad \frac{t \approx t'}{t' \approx t}$$

$$\frac{}{\sigma(t) \approx \sigma(t')}\ t \approx t' \in E \qquad \frac{t_1 \approx t_1',\ \ldots,\ t_n \approx t_n'}{f(t_1, \ldots, t_n) \approx f(t_1', \ldots, t_n')}$$

Table 3. Finite equational basis for BCCSP modulo bisimilarity.

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

Equational Logic. An *equational axiomatisation* (or *axiom system*) over a language L is a collection E of equations $t \approx u$, which are referred to as *axioms*, over the terms in L. We write $E \vdash t \approx u$ if the equation $t \approx u$ is derivable from the axioms in E using the rules of equational logic, presented in Table 2, that express, respectively, reflexivity, symmetry, transitivity, substitution, and closure under L contexts. Without loss of generality, we assume that substitution can be used *only* when $(t \approx u) \in E$. In this case, $\sigma(t) \approx \sigma(u)$ is called a *substitution instance* of an axiom in E.

We are interested in equations that are valid modulo some congruence relation $\mathcal{R}$ over the closed terms in the language L. An equation $t \approx u$ is *sound* modulo $\mathcal{R}$, written $t\,\mathcal{R}\,u$, when $\sigma(t)\,\mathcal{R}\,\sigma(u)$ for all closed substitutions σ. An axiomatisation E is *sound* modulo $\mathcal{R}$ over L if for all terms t, u in $\mathbb{T}(L)$, we have that whenever $E \vdash t \approx u$, then $t\,\mathcal{R}\,u$. E is (*ground-*)*complete* modulo $\mathcal{R}$ if $t\,\mathcal{R}\,u$ implies $E \vdash t \approx u$, for all (closed) L terms t and u. A congruence $\mathcal{R}$ is said to be *finitely based* if there exists a finite axiomatisation E that is sound and complete modulo $\mathcal{R}$.

A classic question is whether an algebra modulo the chosen notion of behavioural congruence (in this work, bisimilarity) affords a finite equational axiomatisation. For example, as shown by Hennessy and Milner in [28], the equations in Table 3 are a finite axiomatisation of bisimilarity over BCCSP. We denote by $\mathcal{E}_0$ the axiom system consisting of the equations in Table 3. Later on, we will extend this set of axioms to present our positive result for $\text{BCCSP}^{\text{p}}_{\text{Act}}$.

Moller's Result over CCS. In his thesis [33], Moller gave a celebrated non-finite axiomatisability result in the field of process algebra, namely:

Theorem 1. *Bisimilarity admits no finite, ground-complete axiomatisation over CCS.*

Specifically, Moller considered the language CCS_a with interleaving parallel composition, defined over $\text{Act} = \{a\}$ by the following syntax:

$$t ::= \quad \mathbf{0} \quad | \quad x \quad | \quad a.t \quad | \quad t + t \quad | \quad t \parallel t \qquad (\text{CCS}_a)$$

Table 4. The SOS rules for CCS_a interleaving parallel composition.

$$\text{(lPar)}\ \frac{t \xrightarrow{a} t'}{t \parallel u \xrightarrow{a} t' \parallel u} \qquad \text{(rPar)}\ \frac{u \xrightarrow{a} u'}{t \parallel u \xrightarrow{a} t \parallel u'}$$

where $x \in \text{Var}$, and $\parallel$ denotes the interleaving parallel composition operator. The SOS rules for CCS_a operators are given by the rules in Table 1, plus the rules for the interleaving parallel operator presented in Table 4.

In detail, for his result, Moller applied the following proof strategy, later referred to as the proof-theoretic approach to negative results [9]. He considered the infinite family of equations Φ with

$$\Phi = \{\varphi_n \mid n \geq 0\}$$

$$\varphi_n \ : \ a \parallel (\sum_{i=1}^{n} a^i) \ \approx \ a.(\sum_{i=1}^{n} a^i) + (\sum_{i=2}^{n+1} a^i) \qquad (n \geq 0)$$

and he proved that whenever n is larger than the size of any term occurring in the equations in a finite, sound axiom system E, then equation φ_n cannot be derived from E. Hence, Theorem 1 specialised to the following result, which will play a fundamental role in the technical development of our contributions:

Theorem 2 (Moller's negative result [33, **Theorem 5.2.12]).** *No finite axiom system that is sound modulo bisimilarity over CCS_a can prove the whole family of equations Φ. Thus no finite, ground-complete axiom system can exist for CCS_a modulo bisimilarity.*

3 The Proof Strategy: Reduction Mappings

The non-finite axiomatisability results that we will present in this paper are all obtained by means of a proof technique, proposed in [11], that allows for transferring this kind of negative results across process languages. Even though we only apply that technique out-of-the-box towards establishing new results, we decided to give, in this section, an overview of the terminology and results presented in [11], to improve the readability of our paper. As our studies are focused on the axiomatisability of bisimilarity, we consider only this behavioural congruence in the presentation below.

We consider two processes description languages defined over the same set of variables: L_{neg} and L_{new}. L_{neg} is known to be non-finitely axiomatisable modulo bisimilarity, whereas L_{new} is the language for which we want to prove this negative result. The aim of the proof technique proposed in [11] is to establish whether it is possible to *lift* the known result for L_{neg} to L_{new}. This approach is based on a variation of the classic idea of *reduction mappings* that, in this setting, are *translations* from $\mathbb{T}(L_{\text{new}})$ to $\mathbb{T}(L_{\text{neg}})$ that preserve soundness and provability.

Given a translation mapping $\widehat{\cdot} : \mathbb{T}(L_{\text{new}}) \to \mathbb{T}(L_{\text{neg}})$ and a collection E of equations over L_{new} terms, we let $\widehat{E} = \{\widehat{t} \approx \widehat{u} \mid t \approx u \in E\}$. The notion of *reduction* is then formalised as follows:

Definition 3 (Reduction). *A mapping* $\widehat{\cdot}\colon \mathbb{T}(L_{\mathrm{new}}) \to \mathbb{T}(L_{\mathrm{neg}})$ *is a reduction from* $\mathbb{T}(L_{\mathrm{new}})$ *to* $\mathbb{T}(L_{\mathrm{neg}})$*, when for all* $t, u \in \mathbb{T}(L_{\mathrm{new}})$*:*

1. $t \sim u \implies \widehat{t} \sim \widehat{u}$*, i.e.,* $\widehat{\cdot}$ *preserves sound equations, and*
2. $E \vdash t \approx u \implies \widehat{E} \vdash \widehat{t} \approx \widehat{u}$*, for each axiom system* E *over* L_{new}*, i.e.,* $\widehat{\cdot}$ *preserves provability.*

Interestingly, in [11, Theorem 2] it is proved that if a mapping is *structural*, then it automatically satisfies Definition 3.2. Hence, the notion of structural mapping will be crucial in the development of our results, as it allows for a significant simplification of the technical proofs.

Definition 4 (Structural mapping). *A mapping* $\widehat{\cdot}\colon \mathbb{T}(L_{\mathrm{new}}) \to \mathbb{T}(L_{\mathrm{neg}})$ *is structural if:*

- *It is the identity function over variables, i.e.,* $\widehat{x} = x$ *for each variable* x*.*
- *It does not introduce new variables, i.e., the set of variables occurring in the term* $\widehat{f(x_1, \dots, x_n)}$ *is included in* $\{x_1, \dots, x_n\}$*, for each operator* f *in* L_{new} *and sequence of distinct variables* $x_1, \dots, x_n$*.*
- *It is defined compositionally, i.e.* $\widehat{f(t_1, \dots, t_n)} = \widehat{f(x_1, \dots, x_n)}[\widehat{t_1}/x_1, \dots, \widehat{t_n}/x_n]$ *for each operator* f *in* L_{new}*, sequence of distinct variables* $x_1, \dots, x_n$ *and sequence of terms* $t_1, \dots, t_n$*. (Here* $[\widehat{t_1}/x_1, \dots, \widehat{t_n}/x_n]$ *stands for the substitution mapping each variable* x_i *to* $\widehat{t_i}$ $(1 \le i \le n)$*, and acting like the identity function on all the other variables.)*

Given a substitution $\sigma\colon \mathrm{Var} \to \mathbb{T}(L_{\mathrm{new}})$, we let $\widehat{\sigma}\colon \mathrm{Var} \to \mathbb{T}(L_{\mathrm{neg}})$ denote the substitution that maps each variable x to $\widehat{\sigma(x)}$.

Proposition 1. *Assume that* $\widehat{\cdot}\colon \mathbb{T}(L_{\mathrm{new}}) \to \mathbb{T}(L_{\mathrm{neg}})$ *is a structural mapping. Then*

- $\widehat{\sigma(t)} = \widehat{\sigma}(\widehat{t})$*, for each term* $t \in \mathbb{T}(L_{\mathrm{new}})$*, and for each substitution* $\sigma\colon \mathrm{Var} \to \mathbb{T}(L_{\mathrm{new}})$*.*
- *The mapping satisfies Definition 3.2.*

Assume now that we have an infinite collection E of equations that are sound modulo bisimilarity, but that are not derivable from any finite, sound axiom system over L_{neg}. The idea in [11] is then that if a structural mapping $\widehat{\cdot}$ is a reduction from $\mathbb{T}(L_{\mathrm{new}})$ to $\mathbb{T}(L_{\mathrm{neg}})$ that contains all the equations in E in its range, then the "malicious" collection of equations that map to those in E cannot be derivable from any finite, sound axiom system over L_{new}. In fact, if those derivations were possible, then the equational properties of $\widehat{\cdot}$ would allow us to write derivations (obtained via the translations of the equational proofs) of the equations in E from a finite, sound axiom system over L_{neg}. As this contradicts the established negative result over L_{neg}, the non-finite axiomatisability result over L_{new} follows.

The intuitions above are formalised in the following definition and theorem.

Definition 5 (E-reflection). *Let* E *be a collection of equations over* L_{neg}*. A reduction* $\widehat{\cdot}$ *is* E-reflecting*, when for each* $t \approx u \in E$*, there are terms* $t', u' \in \mathbb{T}(L_{\mathrm{new}})$ *such that the equation* $t' \approx u'$ *is sound modulo* $\sim$*,* $\widehat{t'} = t$ *and* $\widehat{u'} = u$*. A reduction is* ground E*-reflecting, if the conditions above are satisfied over closed equations.*

Theorem 3 (The lifting theorem). *Assume that there is a collection of (closed) equations E over L_{neg} that is sound modulo $\sim$ and that is not derivable from any finite sound axiom system over L_{neg}. If there exists a (ground) E-reflecting reduction from L_{new} to L_{neg}, then there is no sound and (ground-)complete finite axiom system for $\sim$ over L_{new}.*

We remark that the notion of (ground) E-reflecting reduction requires that only the equations in E are reflected. This means that to establish the negative result over L_{neg} it is enough to identify a particular family of equations that is reflected, disregarding the effects of the reduction on other sound equations. For our purposes, it will be enough to consider the family of equations Φ used by Moller to prove Theorem 2. Hence, our target language will always be CCS$_a$, for some action a, and we will use the lifting technique presented in this section to prove negative results for the languages BCCSP$^{\mathrm{p}}_A$ (Sect. 4.2), BCCSP$^{\mathrm{p}}_\tau$ (Sect. 4.3), and CCS$^{\mathrm{r}}$ (Sect. 5).

4 Axiomatisability Results for CSP Parallel Composition

In this section we investigate the existence of finite, ground-complete axiomatisations of bisimilarity over the process description languages BCCSP$^{\mathrm{p}}_A$ (for all $A \subset \mathrm{Act}$), BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$, BCCSP$^{\mathrm{p}}$ and BCCSP$^{\mathrm{p}}_\tau$. In detail, we apply the reduction technique presented in Sect. 3 to lift Moller's negative result to BCCSP$^{\mathrm{p}}_A$, for each $A \subset \mathrm{Act}$, and to BCCSP$^{\mathrm{p}}_\tau$ (Theorem 4 and Theorem 6, respectively). In between, we show that the reduction technique cannot be applied to BCCSP$^{\mathrm{p}}$ (Theorem 5). Conversely, we establish a positive result for BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$, providing a finite, ground-complete axiomatisation for bisimilarity over this language (Theorem 7).

4.1 The Languages BCCSP$^{\mathrm{p}}_A$, BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$, BCCSP$^{\mathrm{p}}$ and BCCSP$^{\mathrm{p}}_\tau$

The languages that we consider in this section are obtained by extending BCCSP with instances of the CSP-like parallel composition operator $|_A$, where $A \subseteq \mathrm{Act}$ is the set of actions that must be performed synchronously by the parallel components. For this reason, we shall henceforth refer to A in $|_A$ as to the *synchronisation* set. The operator then behaves like interleaving parallel composition on the complement of A.

In detail, the languages are defined by the following grammar

$$t ::= \quad \mathbf{0} \quad | \quad x \quad | \quad a.t \quad | \quad t+t \quad | \quad t \,|_A\, t,$$

with $x \in \mathrm{Var}$ and $a \in \mathrm{Act}$, and they differ in the choice of the synchronisation set(s) $A \subseteq \mathrm{Act}$ as follows:

BCCSP$^{\mathrm{p}}_A$ The parallel operator $|_A$ is defined only over the fixed set $A \subset \mathrm{Act}$ (notice that the inclusion is strict).

BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$ The only synchronisation set is the entire set of actions Act.

BCCSP$^{\mathrm{p}}$ There are no restrictions on the choice of synchronisation sets, i.e. the signature of the language contains the operator $|_A$ for all $A \subseteq \mathrm{Act}$.

Table 5. SOS rules for the parallel operator $|_A$, $A \subseteq \mathrm{Act}$.

$$\text{(lParA)}\ \frac{t \xrightarrow{a} t'}{t \mid_A u \xrightarrow{a} t' \mid_A u}\ a \notin A \qquad \text{(rParA)}\ \frac{u \xrightarrow{a} u'}{t \mid_A u \xrightarrow{a} t \mid_A u'}\ a \notin A$$

$$\text{(syncA)}\ \frac{t \xrightarrow{a} t', \quad u \xrightarrow{a} u'}{t \mid_A u \xrightarrow{a} t' \mid_A u'}\ a \in A$$

$\mathbf{BCCSP}_\tau^\mathrm{p}$ This is like $\mathrm{BCCSP}^\mathrm{p}$ with the additional property that the prefixing operator is of the form $\mu.t$, with $\mu \in \mathrm{Act} \cup \{\tau\}$ for a special action label $\tau \notin \mathrm{Act}$ (see Sect. 4.3 for further details).

The SOS rules for the CSP-like parallel composition operator $|_A$ are given in Table 5. The operational semantics of each of the above-mentioned languages is then given by the rules in Table 1 and those in Table 5, in which A is instantiated according to the considered language.

Let $L \in \{\mathrm{BCCSP}_A^\mathrm{p}, \mathrm{BCCSP}_{\mathrm{Act}}^\mathrm{p}, \mathrm{BCCSP}^\mathrm{p}, \mathrm{BCCSP}_\tau^\mathrm{p}\}$. Since in the technical results to follow we will need to distinguish between transitions over L processes and transitions over CCS_a processes, to avoid possible confusion we will denote the transition relation over $\mathcal{P}(L)$ induced by the rules in Tables 1 and 5 by $\rightarrow_\mathrm{p}$. Similarly, we can properly instantiate the definition of bisimilarity over L processes:

Definition 6 (Bisimilarity over $\mathbf{BCCSP}_A^\mathrm{p}$, $\mathbf{BCCSP}_{\mathrm{Act}}^\mathrm{p}$, $\mathbf{BCCSP}^\mathrm{p}$ and $\mathbf{BCCSP}_\tau^\mathrm{p}$). *Let L be any of $BCCSP_A^\mathrm{p}$, $BCCSP_{\mathrm{Act}}^\mathrm{p}$, $BCCSP^\mathrm{p}$, $BCCSP_\tau^\mathrm{p}$. Bisimulation relations over L processes are defined by applying Definition 2 to the LTS $(\mathcal{P}(L), \mathrm{Act}, \rightarrow_\mathrm{p})$ induced by the SOS rules in Tables 1 and 5. We use the symbol $\sim_\mathrm{p}$ to denote bisimilarity over L processes.*

It is worth noticing that, as briefly outlined above, when the parallel components t, u in $t \mid_A u$ contain only actions that are not in A, then the semantics of $|_A$ coincides with the semantics of CCS interleaving parallel composition. On the other hand, when t and u contain only actions in A, then $|_A$ behaves like “synchronous” parallel composition. The following example highlights these observations.

Example 1. Let $A \subseteq \mathrm{Act}$ and $b \in A$. It is not difficult to see that

$$b \mid_A \sum_{i=1}^{n} b^i \sim_\mathrm{p} b \qquad\qquad (n \geq 1)$$

and therefore

$$b \mid_A \sum_{i=1}^{n} b^i \sim_\mathrm{p} b \mid_A \sum_{j=1}^{m} b^j \qquad\qquad (n, m \geq 1).$$

In particular, we have that the axiom

$$b.x \mid_A (b.y + z) \approx (b.x \mid_A b.y) + (b.x \mid_A z) \qquad \text{if } b \in A$$

is sound modulo $\sim_\mathrm{p}$ over the languages considered in this section.

Conversely, if we pick an action $a \notin A$, then we have

$$a \mid_A \sum_{i=1}^{n} a^i \sim_{\mathrm{p}} a.\sum_{i=1}^{n} a^i + \sum_{j=1}^{n} a^{j+1} \qquad (n \geq 0)$$

and thus

$$a \mid_A \sum_{i=1}^{n} a^i \not\sim_{\mathrm{p}} a \mid_A \sum_{j=1}^{m} a^j \qquad (n \neq m).$$

◀

Notice that, for $a \notin A$, if we let

$$\varphi_A^n : a \mid_A \sum_{i=1}^{n} a^i \approx a.\sum_{i=1}^{n} a^i + \sum_{j=1}^{n} a^{j+1} \qquad (n \geq 0), \tag{1}$$

then the family of equations $\Phi_A = \{\varphi_A^n \mid n \in \mathbb{N}\}$ can be thought of as the counterpart in BCCSP$^{\mathrm{p}}_A$ of the family Φ used by Moller to prove Theorem 2. As we will see, this correspondence will be instrumental in applying the reduction technique to those languages.

4.2 The Negative Result for BCCSP$^{\mathrm{p}}_A$

We start our investigations with BCCSP$^{\mathrm{p}}_A$, for a given set $A \subset \mathrm{Act}$. In particular, by applying the proof methodology discussed in Sect. 3, we prove that:

Theorem 4. *BCCSP$^{\mathrm{p}}_A$ does not have a finite, ground-complete axiomatisation modulo bisimilarity.*

Our first step consists in defining a mapping allowing us to rewrite BCCSP$^{\mathrm{p}}_A$ terms into CCS$_a$ terms. As the target language is built over a specific action, it is natural to have a definition of our mapping that is parametric in that action. Hence, choose an action $a \in \mathrm{Act} \setminus A$. Notice that the requirement that the inclusion $A \subset \mathrm{Act}$ be strict guarantees that such an action a exists.

Definition 7 (The mapping p_a^A). *The mapping $\mathrm{p}_a^A \colon \mathbb{T}(BCCSP^{\mathrm{p}}_A) \to \mathbb{T}(CCS_a)$ is defined inductively over the structure of terms as follows:*

$$\mathrm{p}_a^A(\mathbf{0}) = \mathbf{0} \qquad \mathrm{p}_a^A(x) = x \qquad \mathrm{p}_a^A(t+u) = \mathrm{p}_a^A(t) + \mathrm{p}_a^A(u)$$

$$\mathrm{p}_a^A(b.t) = \begin{cases} a.\mathrm{p}_a^A(t) & \textit{if } b = a, \\ \mathbf{0} & \textit{otherwise.} \end{cases} \qquad \mathrm{p}_a^A(t \mid_A u) = \mathrm{p}_a^A(t) \parallel \mathrm{p}_a^A(u).$$

By Definition 7, for each $t \in \mathbb{T}(\mathrm{BCCSP}^{\mathrm{p}}_A)$, the only action occurring in $\mathrm{p}_a^A(t)$ is a.

In order to lift the negative result in Theorem 2 to BCCSP$^{\mathrm{p}}_A$, we need to prove that the proposed mapping p_a^A is a ground Φ-reflecting reduction. Let us first focus on showing that p_a^A is a reduction, i.e., we need to show that it satisfies the two constraints in Definition 3.

Remark 2. For simplicity, we shall sometimes extend the mapping notation from terms to equations. For instance, if $e : t \approx u$ is an equation over $\text{BCCSP}_A^{\mathrm{p}}$ terms, we shall write $\mathrm{p}_a^A(e)$ to denote the equation over CCS_a terms $\mathrm{p}_a^A(t) \approx \mathrm{p}_a^A(u)$.

The following lemma is immediate from Definition 7.

Lemma 1. *The mapping* p_a^A *is structural.*

Hence, in light of Proposition 1, the mapping p_a^A satisfies Definition 3.2. Our order of business will now be to show that p_a^A preserves sound equations.

Lemma 2. *For all* $p \in \mathcal{P}(BCCSP_A^{\mathrm{p}})$ *and* $q \in \mathcal{P}(CCS_a)$*, if* $\mathrm{p}_a^A(p) \xrightarrow{a} q$*, then there exists a* $BCCSP_A^{\mathrm{p}}$ *process* p' *such that* $p \xrightarrow{a}_{\mathrm{p}} p'$ *and* $\mathrm{p}_a^A(p') = q$.

Proof. We proceed by structural induction over p.

- Case $p = \mathbf{0}$. This is vacuous, since $\mathrm{p}_a^A(p)$ has no outgoing transition.
- Case $p = b.p_0$. By Definition 7 and the assumption that $\mathrm{p}_a^A(p) \xrightarrow{a} q$, we have that $b = a \notin A$ and $\mathrm{p}_a^A(p_0) = q$. As $p \xrightarrow{a}_{\mathrm{p}} p_0$, the claim follows.
- Case $p = p_1 \mid_A p_2$. By Definition 7, we have that $\mathrm{p}_a^A(p) = \mathrm{p}_a^A(p_1) \parallel \mathrm{p}_a^A(p_2)$. Moreover, by the proviso of the lemma, $\mathrm{p}_a^A(p_1) \parallel \mathrm{p}_a^A(p_2) \xrightarrow{a} q$, for some CCS_a process q. This follows by an application of either rule (lPar) or rule (rPar) from Table 4. We can assume, without loss of generality, that rule (lPar) was applied. (The case of an application of rule (rPar) follows from a similar reasoning.) Hence $\mathrm{p}_a^A(p_1) \xrightarrow{a} q'$ for some CCS_a process q' such that $q' \parallel \mathrm{p}_a^A(p_2) = q$. By the induction hypothesis, we obtain that $p_1 \xrightarrow{a}_{\mathrm{p}} p_1'$ for some $p_1' \in \mathcal{P}(\text{BCCSP}_A^{\mathrm{p}})$ such that $\mathrm{p}_a^A(p_1') = q'$. Hence, as $p_1 \xrightarrow{a}_{\mathrm{p}} p_1'$ and $a \notin A$, we can apply rule (lParA) from Table 5 and obtain that $p = p_1 \mid_A p_2 \xrightarrow{a}_{\mathrm{p}} p_1' \mid_A p_2$. Since $\mathrm{p}_a^A(p_1' \mid_A p_2) = \mathrm{p}_a^A(p_1') \parallel \mathrm{p}_a^A(p_2) = q' \parallel \mathrm{p}_a^A(p_2) = q$, the claim follows.
- Case $p = p_1 + p_2$. This case is similar to the case of parallel composition discussed above. The only difference is that rules (lSum) and (rSum) from Table 1 are applied in place of rules (lParA) and (rParA), respectively. □

Lemma 3. *For all* $p, p' \in \mathcal{P}(BCCSP_A^{\mathrm{p}})$*, if* $p \xrightarrow{a}_{\mathrm{p}} p'$ *then* $\mathrm{p}_a^A(p) \xrightarrow{a} \mathrm{p}_a^A(p')$.

Proof. We proceed by induction on the size of the proof for the transition $p \xrightarrow{a}_{\mathrm{p}} p'$. We distinguish three cases, according to the last inference rule from Tables 1 and 5 that is applied in the proof. (Notice that the analysis of symmetric rules is omitted.) We remark that since $a \notin A$, rule (syncA) cannot be applied as the last rule in the proof for $p \xrightarrow{a}_{\mathrm{p}} p'$.

- Rule (act). In this case, we have that $p = a.p'$ and $p \xrightarrow{a}_{\mathrm{p}} p'$. By Definition 7, we have that $\mathrm{p}_a^A(a.p') = a.\mathrm{p}_a^A(p')$, and, thus, we can apply rule (act) and obtain that $\mathrm{p}_a^A(p) = \mathrm{p}_a^A(a.p') = a.\mathrm{p}_a^A(p') \xrightarrow{a} \mathrm{p}_a^A(p')$. Hence the claim follows in this case.
- Rule (lSum). In this case, we have that $p = p_0 + p_1$, $p_0 \xrightarrow{a}_{\mathrm{p}} p'$, and $\mathrm{p}_a^A(p) = \mathrm{p}_a^A(p_0) + \mathrm{p}_a^A(p_1)$. By the inductive hypothesis we get that $\mathrm{p}_a^A(p_0) \xrightarrow{a} \mathrm{p}_a^A(p')$. By applying now rule (lSum), we conclude that $\mathrm{p}_a^A(p) = \mathrm{p}_a^A(p_0) + \mathrm{p}_a^A(p_1) \xrightarrow{a} \mathrm{p}_a^A(p')$.

- Rule (lParA). In this case, as $a \notin A$, we have that $p = p_0 \mid_A p_1$, $p_0 \xrightarrow{a}_{\mathrm{p}} p_0'$ for some $p_0' \in \mathcal{P}(\mathrm{BCCSP}_A^{\mathrm{p}})$, and $p' = p_0' \mid_A p_1$. By induction, we obtain that $\mathrm{p}_a^A(p_0) \xrightarrow{a} \mathrm{p}_a^A(p_0')$. Hence, by applying rule (lPar) from Table 4 to $\mathrm{p}_a^A(p)$, we get that $\mathrm{p}_a^A(p) = \mathrm{p}_a^A(p_0) \parallel \mathrm{p}_a^A(p_1) \xrightarrow{a} \mathrm{p}_a^A(p_0') \parallel \mathrm{p}_a^A(p_1) = \mathrm{p}_a^A(p_0' \mid_A p_1) = \mathrm{p}_a^A(p')$. □

We can now proceed to prove that p_a^A satisfies Definition 3.1 as well. Moreover, we show that it is also ground Φ-reflecting.

Proposition 2. *The mapping p_a^A satisfies the following properties:*

1. *For all $t, u \in \mathbb{T}(BCCSP_A^{\mathrm{p}})$, if $t \sim_{\mathrm{p}} u$ then $\mathrm{p}_a^A(t) \sim \mathrm{p}_a^A(u)$.*
2. *The mapping p_a^A is ground Φ-reflecting.*

Proof. We prove the two items separately.

1. First, observe that for every (closed) term t in CCS_a there is a (closed) term $t_a^{\mathrm{p},A}$ in $\mathrm{BCCSP}_A^{\mathrm{p}}$ such that $\mathrm{p}_a^A(t_a^{\mathrm{p},A}) = t$. The term $t_a^{\mathrm{p},A}$ is defined as follows:

$$\mathbf{0}_a^{\mathrm{p},A} = \mathbf{0} \qquad x_a^{\mathrm{p},A} = x \qquad (a.t)_a^{\mathrm{p},A} = a.t_a^{\mathrm{p},A}$$
$$(t+u)_a^{\mathrm{p},A} = t_a^{\mathrm{p},A} + u_a^{\mathrm{p},A} \qquad (t \parallel u)_a^{\mathrm{p},A} = t_a^{\mathrm{p},A} \mid_A u_a^{\mathrm{p},A}.$$

   Given a CCS_a substitution σ, we define $\sigma_a^{\mathrm{p},A}$ to be the $\mathrm{BCCSP}_A^{\mathrm{p}}$ substitution given by $\sigma_a^{\mathrm{p},A}(x) = (\sigma(x))_a^{\mathrm{p},A}$. By Proposition 1 and since the mapping p_a^A is structural (Lemma 1), we have that

$$\mathrm{p}_a^A(\sigma_a^{\mathrm{p},A}(t)) = \mathrm{p}_a^A(\sigma_a^{\mathrm{p},A})(\mathrm{p}_a^A(t)) = \sigma(\mathrm{p}_a^A(t)),$$

   for all $t \in \mathbb{T}(\mathrm{BCCSP}_A^{\mathrm{p}})$.
   To prove the claim, it is then enough to show that the following relation

$$\mathcal{R} = \{(\sigma(\mathrm{p}_a^A(t)), \sigma(\mathrm{p}_a^A(u))) \mid t \sim_{\mathrm{p}} u \text{ and } \sigma\colon \mathrm{Var} \to \mathcal{P}(\mathrm{CCS}_a)\}$$

   is a bisimulation relation over CCS_a processes.
   Notice, first of all, that since $\sim_{\mathrm{p}}$ is symmetric, then so is $\mathcal{R}$. Assume now that $\sigma(\mathrm{p}_a^A(t)) \mathrel{\mathcal{R}} \sigma(\mathrm{p}_a^A(u))$, where $t, u \in \mathbb{T}(\mathrm{BCCSP}_A^{\mathrm{p}})$ and σ is a closed CCS_a substitution. By the definition of $\mathcal{R}$, we have that $t \sim_{\mathrm{p}} u$. Assume now that $\sigma(\mathrm{p}_a^A(t)) \xrightarrow{a} q$ for some $q \in \mathcal{P}(\mathrm{CCS}_a)$. By the observation above, this means that $\mathrm{p}_a^A(\sigma_a^{\mathrm{p},A}(t)) \xrightarrow{a} q$. By Lemma 2, we get that $\sigma_a^{\mathrm{p},A}(t) \xrightarrow{a}_{\mathrm{p}} p'$ for some $p' \in \mathcal{P}(\mathrm{BCCSP}_A^{\mathrm{p}})$ such that $\mathrm{p}_a^A(p') = q$. As $t \sim_{\mathrm{p}} u$ implies that $\sigma_a^{\mathrm{p},A}(t) \sim_{\mathrm{p}} \sigma_a^{\mathrm{p},A}(u)$, we have that $\sigma_a^{\mathrm{p},A}(u) \xrightarrow{a}_{\mathrm{p}} p''$, for some $p'' \in \mathcal{P}(\mathrm{BCCSP}_A^{\mathrm{p}})$ such that $p' \sim_{\mathrm{p}} p''$. Additionally, by Lemma 3 we have that $\sigma(\mathrm{p}_a^A(u)) = \mathrm{p}_a^A(\sigma_a^{\mathrm{p},A}(u)) \xrightarrow{a} \mathrm{p}_a^A(p'')$. We can then conclude by noticing that, since $p' \sim_{\mathrm{p}} p''$, by definition of $\mathcal{R}$ it holds that $q = \mathrm{p}_a^A(p') \mathrel{\mathcal{R}} \mathrm{p}_a^A(p'')$, i.e., $\mathcal{R}$ is a bisimulation relation over CCS_a processes.
2. In order to show that p_a^A is ground Φ-reflecting, it is enough to argue that the family Φ_A consisting of the closed equations φ_A^n defined in Eq. 1 is mapped exactly onto Φ. Since $a \notin A$ we have that p_a^A simply replaces all the occurrences of $\mid_A$ in each equation φ_A^n with $\parallel$. Hence, we have that $\mathrm{p}_a^A(\varphi_A^n) = \varphi_n$, for each $n \geq 0$. □

From Lemma 1 and Proposition 2, we can infer that p_a^A is a well-defined reduction as in Definition 3, and it is also ground Φ-reflecting. Theorem 4 then follows by Theorem 2 and Theorem 3.

4.3 The Case of BCCSP$^{\mathrm{p}}$ and the Negative Result for BCCSP$^{\mathrm{p}}_{\tau}$

Given the negative result over BCCSP$^{\mathrm{p}}_{A}$, it is natural to wonder what happens when we extend that language to BCCSP$^{\mathrm{p}}$, namely BCCSP enriched with an operator $|_A$, for each $A \subseteq \mathrm{Act}$.

One might expect that bisimilarity does not have a finite, ground-complete axiomatisation over BCCSP$^{\mathrm{p}}$ and indeed we conjecture that such a results holds. However, the reduction method cannot be applied to prove such a claim. Specifically, consider the language BCCSP$^{\mathrm{p}}$ over $\mathrm{Act} = \{a\}$. We can prove the following result:

Theorem 5. *There is no structural reduction from BCCSP$^{\mathrm{p}}$ to CCS$_a$ that is ground Φ-reflecting.*

Proof. To simplify notation, let us use $|_a$ in place of $|_{\{a\}}$.

Assume that $\widehat{\cdot}$ is a structural reduction from BCCSP$^{\mathrm{p}}$ to CCS$_a$. Our aim is to prove that $\widehat{\cdot}$ is not ground Φ-reflecting.

To this end, we start by recalling that, since $\widehat{\cdot}$ is structural (Definition 4), then:

$$\widehat{at} = \widehat{ax}[\widehat{t}/x], \text{ for each } t \in \mathbb{T}(\mathrm{BCCSP^p}) \tag{2}$$

$$\widehat{t_1 \odot t_2} = \widehat{x_1 \odot x_2}[\widehat{t_1}/x_1, \widehat{t_2}/x_2], \text{ for each } t_1, t_2 \in \mathbb{T}(\mathrm{BCCSP^p}) \text{ and binary operator } \odot \in \{+, |_\emptyset, |_a\}. \tag{3}$$

Moreover, as $\widehat{\cdot}$ preserves sound equations (Definition 3), we have that:

$$\widehat{ax\,|_a\,\mathbf{0}} \sim \widehat{\mathbf{0}} \sim \widehat{\mathbf{0}\,|_a\,ax}; \tag{4}$$

$$\widehat{a^n\,|_a\,a^n} \sim \widehat{a^n}, \text{ for all } n \geq 0; \tag{5}$$

$$\widehat{\mathbf{0} + \mathbf{0}} \sim \widehat{\mathbf{0}}; \tag{6}$$

$$\widehat{\mathbf{0}\,|_\emptyset\,\mathbf{0}} \sim \widehat{\mathbf{0}}. \tag{7}$$

Assume now that

$$\widehat{x_1 |_a x_2} = t \tag{8}$$

for some $t \in \mathbb{T}(\mathrm{CCS}_a)$ with $\mathrm{var}(t) \subseteq \{x_1, x_2\}$ (as $\widehat{\cdot}$ is structural).

We can distinguish two cases, according to whether t is a closed term or not. In both cases, we shall show that $\widehat{\cdot}$ is not ground Φ-reflecting.

- CASE 1: t IS A CLOSED CCS$_a$ TERM. In this case, for each $n \geq 0$, we have that

  $$t \sim \widehat{a^n} \sim \mathbf{0}. \tag{9}$$

  Indeed,

  $$\begin{aligned} \widehat{a^n} &\sim \widehat{a^n\,|_a\,a^n} && \text{(by 5)} \\ &\sim t[\widehat{a^n}/x_1, \widehat{a^n}/x_2] && \text{(by 3 and 8)} \\ &\sim t[\widehat{\mathbf{0}}/x_1, \widehat{\mathbf{0}}/x_2] && \text{(since } t \text{ is closed)} \\ &\sim \widehat{\mathbf{0}} && \text{(by 3 and 5 with } n = 0). \end{aligned}$$

  We now claim that

Claim 1: *For each $p \in \mathcal{P}(BCCSP^{\mathrm{p}})$, it holds that $\widehat{p} \sim \widehat{\mathbf{0}}$.*

Before proving Claim 1 above, we observe that by using it we can immediately show that the mapping $\widehat{\cdot}$ is not ground Φ-reflecting. Indeed, since

$$a \parallel a \not\sim a \parallel (a + a^2),$$

by Claim 1 there cannot be two processes $p, q \in \mathcal{P}(\mathrm{BCCSP}^{\mathrm{p}})$ such that $\widehat{p} = a \parallel a$ and $\widehat{q} = a \parallel (a + a^2)$. Let us now prove Claim 1.

Proof of Claim 1: We proceed by induction on the structure of process p.

- The case $p = \mathbf{0}$ is trivial.
- Case $p = aq$. We have

$$\begin{aligned} \widehat{p} &= \widehat{ax}[\widehat{q}/x] && \text{(by 2)} \\ &\sim \widehat{ax}[\widehat{\mathbf{0}}/x] && \text{(by induction and } \sim \text{ is a congruence)} \\ &\sim \widehat{a\mathbf{0}} && \text{(by 2)} \\ &\sim \hat{\mathbf{0}} && \text{(by 9).} \end{aligned}$$

- Case $p = p_1 \odot p_2$ for some binary operator $\odot \in \{+, |_{\emptyset}, |_a\}$. In this case,

$$\begin{aligned} \widehat{p} &= \widehat{x_1 \odot x_2}[\widehat{p_1}/x_1, \widehat{p_2}/x_2] && \text{(by 3)} \\ &\sim \widehat{x_1 \odot x_2}[\widehat{\mathbf{0}}/x_1, \widehat{\mathbf{0}}/x_2] && \text{(by induction and } \sim \text{ is a congruence)} \\ &\sim \widehat{\mathbf{0} \odot \mathbf{0}} && \text{(by 3)} \\ &\sim \widehat{\mathbf{0}} && \text{(by 5–7 according to the form of } \odot\text{).} \end{aligned}$$

  This concludes the Proof of Claim 1.

The proof of Case 1 is now complete.

– CASE 2: t IS AN OPEN CCS$_a$ TERM. Assume, without loss of generality, that t contains at least an occurrence of x_1. (The cases of $x_2 \in \mathrm{var}(t)$ and $x_1, x_2 \in \mathrm{var}(t)$ can be treated in a similar fashion and are therefore omitted.) Firstly, we observe that for each $p \in \mathcal{P}(\mathrm{BCCSP}^{\mathrm{p}})$

$$\begin{aligned} \widehat{\mathbf{0}} &\sim \widehat{ap \,|_a\, \mathbf{0}} && \text{(by 4)} \\ &= t[\widehat{ap}/x_1, \widehat{\mathbf{0}}/x_2] && \text{(by 3 and 8).} \end{aligned}$$

Moreover, we recall that for every $u \in \mathrm{CCS}_a$ and $y \in \mathrm{Var}$, it holds that whenever $y \in \mathrm{var}(u)$ then $\mathrm{depth}(\sigma(y)) \leq \mathrm{depth}(\sigma(u))$ for every closed substitution σ. Hence, since $t \in \mathbb{T}(\mathrm{CCS}_a)$ and $x_1 \in \mathrm{var}(t)$, we have that

$$\mathrm{depth}(\widehat{ap}) \leq \mathrm{depth}(\widehat{ap \,|_a\, \mathbf{0}}) = \mathrm{depth}(\widehat{\mathbf{0}}). \tag{10}$$

We claim that

Claim 2: *For each* $n \geq 0$ *and processes* $p_1, \dots, p_n \in \mathcal{P}(BCCSP^{\mathrm{p}})$*, it holds that* $\mathrm{depth}(\widehat{\sum_{i=1}^{n} ap_i}) \leq \mathrm{depth}(\widehat{\mathbf{0}})$.

Proof of Claim 2: We proceed by induction on $n \geq 0$.

- The case $n = 0$ is trivial.
- For the inductive step, we have that:

$$\begin{aligned}
&\mathrm{depth}(\widehat{\sum_{i=1}^{n+1} ap_i}) \\
&= \mathrm{depth}(\widehat{\sum_{i=1}^{n} ap_i + ap_{n+1}}) \\
&= \mathrm{depth}(\widehat{x_1 + x_2}[\widehat{\sum_{i=1}^{n} ap_i}/x_1, \widehat{ap_{n+1}}/x_2]) && \text{(by 3)} \\
&\leq \mathrm{depth}(\widehat{x_1 + x_2}[\widehat{\mathbf{0}}/x_1, \widehat{\mathbf{0}}/x_2]) && \text{(by induction and 10)} \\
&= \mathrm{depth}(\widehat{\mathbf{0} + \mathbf{0}}) && \text{(by 3)} \\
&= \mathrm{depth}(\widehat{\mathbf{0}}) && \text{(by 6 and Remark 1).}
\end{aligned}$$

This concludes the Proof of Claim 2.

Claim 3: *For each* $p \in \mathcal{P}(BCCSP^{\mathrm{p}})$ *it holds that* $\mathrm{depth}(\widehat{p}) \leq \widehat{(}\mathbf{0})$.

Proof of Claim 3: First of all, we notice that each BCCSP$^{\mathrm{p}}$ process can be rewritten into *head normal form* up to bisimilarity. This means that, given any $p \in \mathcal{P}(\mathrm{BCCSP}^{\mathrm{p}})$, we have that $p \sim \sum_{i=1}^{n} ap_i$ for some $n \geq 0$ and $p_1, \dots, p_n \in \mathcal{P}(\mathrm{BCCSP}^{\mathrm{p}})$.

Since $\widehat{\cdot}$ preserves sound equations, we have

$$\widehat{p} \sim \widehat{\sum_{i=1}^{n} ap_i}.$$

Hence, by Claim 2 above, it follows that

$$\mathrm{depth}(\widehat{p}) = \mathrm{depth}(\widehat{\sum_{i=1}^{n} ap_i}) \leq \mathrm{depth}(\widehat{\mathbf{0}}). \tag{11}$$

This concludes the Proof of Claim 3.

We can now proceed to show that $\widehat{\cdot}$ is not ground Φ-reflecting. Let $k = \mathrm{depth}(\widehat{\mathbf{0}})$. We have that equation $\varphi_k \in \Phi$ is of the form:

$$a \parallel (\sum_{i=1}^{k} a^i) \approx a.(\sum_{i=1}^{k} a^i) + \sum_{i=2}^{k+1} a^i.$$

In particular, the depth of $a \parallel (\sum_{i=1}^{k} a^i)$ is $k + 1$. Therefore, by 11, there is no $p \in \mathcal{P}(\mathrm{BCCSP}^{\mathrm{p}})$ such that $\widehat{p} = a \parallel (\sum_{i=1}^{k} a^i)$.

The proof of Case 2 is now concluded.

This completes the proof of the Theorem 5. □

Although we proved Theorem 5 in the simplified setting of $\mathrm{Act} = \{a\}$, it is not difficult to see that the proof can be extended to the general case $\{a\} \subset \mathrm{Act}$ in a straightforward manner.

Since the reduction method cannot be applied, one might show the non-existence of a finite, ground-complete axiomatisation of bisimilarity over BCCSP$^{\mathrm{p}}$ by adapting the strategy employed by Moller in his proof of Theorem 2. However, since that proof would require several pages of technical results, we leave it as an avenue for future research, and we deal with the presence of all the operators $|_A$ in a simplified setting.

The basic idea behind the reduction defined for BCCSP$^{\mathrm{p}}_A$ is that we can always identify an action $a \in \mathrm{Act} \setminus A$ such that the parallel operator $|_A$ always allows for interleaving of a-moves of its arguments. Clearly, if we add an operator $|_A$ for each $A \subseteq \mathrm{Act}$ to the language, it is no longer possible to identify such an action. There is, however, a special action that is not used to build syntactically CSP terms, but it is however necessary to express their semantics: the silent action $\tau \notin \mathrm{Act}$. CSP terms are defined over Act, which means that the language does not offer a τ-prefixing operator; however, in order to properly define the operational semantics of the internal choice operator, the set of action labels in the LTS is $\mathrm{Act} \cup \{\tau\}$. In particular, as explained in [19], the operational semantics of the parallel operators always allow for the interleaving of τ-moves of their arguments.

Hence, we now consider BCCSP$^{\mathrm{p}}_\tau$, i.e., the extension of BCCSP$^{\mathrm{p}}$ that includes the τ-prefixing operator, and we prove the following result:

Theorem 6. *BCCSP$^{\mathrm{p}}_\tau$ modulo bisimilarity does not afford a finite, ground-complete axiomatisation.*

To this end, we apply the same proof technique that we used in Sect. 4.2 for BCCSP$^{\mathrm{p}}_A$. The reduction mapping for BCCSP$^{\mathrm{p}}_\tau$ is almost identical to the mapping p_a^A defined for BCCSP$^{\mathrm{p}}_A$, the only difference being that now we consider the language CCS$_\tau$ as target language, i.e., CCS$_a$ with $a = \tau$.

Remark 3. Theorem 2 remains true over CCS$_\tau$. In fact, as we are considering strong bisimilarity, there is no difference between τ and any other observable action $a \in \mathrm{Act}$. Specifically, if we let Φ_τ be the family of equations in Φ in which each occurrence of a is replaced by τ, then we can repeat Moller's arguments in a step-by-step fashion to obtain that no finite axiom system, that is sound modulo bisimilarity, can prove the whole family of equations Φ_τ.

Definition 8 (The mapping p_τ). *The mapping $\mathrm{p}_\tau \colon \mathbb{T}(BCCSP^{\mathrm{p}}_\tau) \to \mathbb{T}(CCS_\tau)$ is defined inductively over the structure of BCCSP$^{\mathrm{p}}_\tau$ terms as follows:*

$$\mathrm{p}_\tau(\mathbf{0}) = \mathbf{0} \qquad \mathrm{p}_\tau(x) = x \qquad \mathrm{p}_\tau(t + u) = \mathrm{p}_\tau(t) + \mathrm{p}_\tau(u)$$

$$\mathrm{p}_\tau(\mu.t) = \begin{cases} \tau.\mathrm{p}_\tau(t) & \text{if } \mu = \tau, \\ \mathbf{0} & \text{otherwise;} \end{cases} \qquad \mathrm{p}_\tau(t \mid_A u) = \mathrm{p}_\tau(t) \parallel \mathrm{p}_\tau(u).$$

Intuitively, we use the mapping p_τ to eliminate any action $b \neq \tau$ from terms, so that a process $\mathrm{p}_\tau(p \mid_A q)$ can perform a transition $\mathrm{p}_\tau(p \mid_A q) \xrightarrow{\tau} p'$, for some CCS$_a$ process p', if and only if $b = \tau$. (Recall that, by construction $\tau \notin A$ for each $A \subseteq \mathrm{Act}$.)

First, we note that this mapping is a structural mapping.

Lemma 4. *The mapping* p_τ *is structural.*

We now state the corresponding results over BCCSP$^{\mathrm{p}}_\tau$ to Lemma 2 and Lemma 3.

Lemma 5. *For all* $p \in \mathcal{P}(BCCSP^{\mathrm{p}}_\tau)$ *and* $q \in \mathcal{P}(CCS_\tau)$*, if* $\mathrm{p}_\tau(p) \xrightarrow{\tau} q$*, then there exists a* BCCSP$^{\mathrm{p}}_\tau$ *process* p'*, such that* $p \xrightarrow{\tau}_{\mathrm{p}} p'$ *and* $\mathrm{p}_\tau(p') = q$*.*

Proof. The proof is by structural induction over p. We omit it since it is similar to that of Lemma 2. □

Lemma 6. *For all* $p, p' \in \mathcal{P}(BCCSP^{\mathrm{p}}_\tau)$*, if* $p \xrightarrow{\tau}_{\mathrm{p}} p'$ *then* $\mathrm{p}_\tau(p) \xrightarrow{\tau} \mathrm{p}_\tau(p')$*.*

Proof. The proof proceeds by induction over the size of the proof for $p \xrightarrow{\tau}_{\mathrm{p}} p'$. It is analogous to the proof of Lemma 3, and it is therefore omitted. □

The following result, which extends Proposition 2 to BCCSP$^{\mathrm{p}}_\tau$, allows us to prove that p_τ is a well-defined reduction mapping that is also ground Φ_τ-reflecting.

Proposition 3. *The following properties hold for the mapping* p_τ*:*

1. For all $t, u \in \mathbb{T}(BCCSP^{\mathrm{p}}_\tau)$*, if* $t \sim_{\mathrm{p}} u$*, then* $\mathrm{p}_\tau(t) \sim \mathrm{p}_\tau(u)$*.*
2. The mapping p_τ *is ground* Φ_τ*-reflecting.*

Proof. 1. We start by observing that for every (closed) term t in CCS$_\tau$ there is a (closed) term t^{p}_τ in BCCSP$^{\mathrm{p}}_\tau$ such that $\mathrm{p}_\tau(t^{\mathrm{p}}_\tau) = t$. The term t^{p}_τ is defined as follows:

$$\mathbf{0}^{\mathrm{p}}_\tau = \mathbf{0} \qquad x^{\mathrm{p}}_\tau = x \qquad (\tau.t)^{\mathrm{p}}_\tau = \tau.t^{\mathrm{p}}_\tau$$
$$(t+u)^{\mathrm{p}}_\tau = t^{\mathrm{p}}_\tau + u^{\mathrm{p}}_\tau \qquad (t \parallel u)^{\mathrm{p}}_\tau = t^{\mathrm{p}}_\tau \mid_\emptyset u^{\mathrm{p}}_\tau .$$

Then, given a CCS$_\tau$ substitution σ, we define the BCCSP$^{\mathrm{p}}_\tau$ substitution σ^{p}_τ by $\sigma^{\mathrm{p}}_\tau(x) = (\sigma(x))^{\mathrm{p}}_\tau$. By Lemma 4 and Proposition 1, we have that $\mathrm{p}_\tau(\sigma^{\mathrm{p}}_\tau(t)) = \mathrm{p}_\tau(\sigma^{\mathrm{p}}_\tau)(\mathrm{p}_\tau(t)) = \sigma(\mathrm{p}_\tau(t))$ for all $t \in \mathbb{T}(\mathrm{BCCSP}^{\mathrm{p}}_\tau)$.

The proof of this statement then proceeds as that of the corresponding statement in Proposition 2, and it is therefore omitted.

2. Consider the family of equations $\Phi_{\tau,\emptyset} = \{\varphi^n_{\tau,\emptyset} \mid n \in \mathbb{N}\}$, where the closed equations $\varphi^n_{\tau,\emptyset}$ are defined as in Eq. 1, using the set $\emptyset$ as synchronisation set, and replacing each occurrence of a with τ. It is straightforward to prove that $\mathrm{p}_\tau(\varphi^n_{\tau,\emptyset}) = \varphi_{\tau,n}$ for each $n \in \mathbb{N}$. Hence, p_τ is ground Φ_τ-reflecting. □

Theorem 6 is then obtained as a direct consequence of Lemma 4, Proposition 3, Theorem 3, and Theorem 2.

Table 6. Additional axioms for $\text{BCCSP}^{\text{p}}_{\text{Act}}$.

(P1)	$x \mid_{\text{Act}} y \;\approx\; y \mid_{\text{Act}} x$
(P2)	$(x + y) \mid_{\text{Act}} z \;\approx\; (x \mid_{\text{Act}} z) + (y \mid_{\text{Act}} z)$
(P3)	$(x \mid_{\text{Act}} \mathbf{0}) \;\approx\; \mathbf{0}$
(P4)	$(a.x \mid_{\text{Act}} a.y) \;\approx\; a.(x \mid_{\text{Act}} y)$, for each $a \in \text{Act}$
(P5)	$(a.x \mid_{\text{Act}} b.y) \;\approx\; \mathbf{0}$, for $b \neq a$, and $a, b \in \text{Act}$.

4.4 The Case of $\text{BCCSP}^{\text{p}}_{\text{Act}}$

We now argue that the requirement that the inclusion $A \subset \text{Act}$ be strict, used in Sect. 4.2, is indeed necessary for Theorem 4 to hold. We also notice that a similar requirement is not explicitly expressed for the validity of Theorem 6, proved in Sect. 4.3, because having $\mid_A$ defined for all $A \subseteq \text{Act}$ automatically guaranteed the existence of at least one synchronisation set A such that $a \notin A$ for some action $a \in \text{Act}$, namely the synchronisation set $A = \emptyset$. Moreover, as discussed in Example 1, given a synchronisation set A, the requirement $a \notin A$ is crucial to guarantee the soundness modulo bisimilarity of equation φ^n_A, for any $n \in \mathbb{N}$ (see Eq. 1).

In this section, we handle the border case of the language $\text{BCCSP}^{\text{p}}_{\text{Act}}$, which includes only the parallel operator $\mid_{\text{Act}}$, and we show that for this special case a positive result holds: we provide a *finite, ground-complete axiomatisation* of bisimilarity over this language. Let us consider the axiom system $\mathcal{E}_{\text{p}} = \mathcal{E}_0 \cup \{\text{P1}, \text{P2}, \text{P3}, \text{P4}, \text{P5}\}$, where $\mathcal{E}_0$ consists of the axioms in Table 3, and axioms P1–P5 are reported in Table 6. Notice that the axiom schemata P4 and P5 generate only finitely many axioms. More precisely, P4 generates $|\text{Act}|$ axioms, and P5 generates $|\text{Act}| \times (|\text{Act}| - 1)$ axioms. We will now prove the following result:

Theorem 7. *$\mathcal{E}_{\text{p}}$ is a finite, ground-complete axiomatisation of $\text{BCCSP}^{\text{p}}_{\text{Act}}$ modulo bisimilarity.*

The idea behind the proof of Theorem 7 is that the axioms in Table 6 allow us to eliminate all occurrences of the parallel operator $\mid_{\text{Act}}$ from $\text{BCCSP}^{\text{p}}_{\text{Act}}$ processes. Hence, every $\text{BCCSP}^{\text{p}}_{\text{Act}}$ process can be proven equal to a BCCSP process using $\mathcal{E}_{\text{p}}$. The ground-completeness of $\mathcal{E}_{\text{p}}$ then follows from that of $\mathcal{E}_0$ proven in [28]. To that end, we first show:

Lemma 7. *For all closed BCCSP terms p and q, there exists a closed BCCSP term r such that $\mathcal{E}_{\text{p}} \vdash p \mid_{\text{Act}} q \approx r$.*

Proof. The proof is by induction on $\text{size}(p \mid_{\text{Act}} q)$. First of all we notice that, given any closed BCCSP term p, we can assume, without loss of generality, that $p = \sum_{i \in I} a_i p_i$ for some finite index set I, actions $a_i \in \text{Act}$, and closed BCCSP terms p_i, for $i \in I$. In fact, in case p is not already in this shape, then by applying axioms A2 and A4 in Table 3 we can remove superfluous occurrences of $\mathbf{0}$ summands. In particular, we remark that this transformation does not increase the number of operator symbols occurring in p.

Thus we proceed under the assumption that

$$p = \sum_{i \in I} a_i.p_i \quad \text{and} \quad q = \sum_{j \in J} b_i.q_j.$$

We proceed by a case analysis on the cardinality of the sets of indexes I and J.

- If either $I = \emptyset$ or $J = \emptyset$, then $p = \mathbf{0}$ or $q = \mathbf{0}$. In light of P1, without loss of generality, we can assume that $q = \mathbf{0}$ and we have that $p \mid_{\mathrm{Act}} q = p \mid_{\mathrm{Act}} \mathbf{0}$. Thus by applying axiom P3, we get $\mathcal{E}_{\mathrm{p}} \vdash p \mid_{\mathrm{Act}} q \approx \mathbf{0}$ and we are done.
- If both I and J are singletons, then we have that $p = a.p'$ and $q = b.q'$, for some $a, b \in \mathrm{Act}$ and BCCSP processes p' and q'.
  If $a = b$, then we use axiom P4 to get $\mathcal{E}_{\mathrm{p}} \vdash a.p' \mid_{\mathrm{Act}} a.q' \approx a.(p' \mid_{\mathrm{Act}} q')$. Since the size of $p' \mid_{\mathrm{Act}} q'$ is smaller than that of $p \mid_{\mathrm{Act}} q$, by the induction hypothesis, there exists a BCCSP process r' such that $\mathcal{E}_{\mathrm{p}} \vdash p' \mid_{\mathrm{Act}} q' \approx r'$. Thus we have $\mathcal{E}_{\mathrm{p}} \vdash a.p' \mid_{\mathrm{Act}} a.q' \approx a.r'$, which is a BCCSP process.
  In the case that $a \neq b$, then we can use axiom P5, to infer $\mathcal{E}_{\mathrm{p}} \vdash a.p' \mid_{\mathrm{Act}} b.q' \approx \mathbf{0}$ and we are done.
- We can now assume, without loss of generality, that $|I| > 1$ and $|J| \geq 1$. This means that we can express p as the summation of two summands of smaller size that are different from $\mathbf{0}$, i.e. $p = p_1 + p_2$, for some BCCSP processes p_1 and p_2. Then, we use axiom P2 to get $\mathcal{E}_{\mathrm{p}} \vdash (p_1 + p_2) \mid_{\mathrm{Act}} q \approx (p_1 \mid_{\mathrm{Act}} q) + (p_2 \mid_{\mathrm{Act}} q)$. Since both $p_1 \mid_{\mathrm{Act}} q$ and $p_2 \mid_{\mathrm{Act}} q$ have size less than that of $p \mid_{\mathrm{Act}} q$, by the induction hypothesis, we have that there exist BCCSP processes r' and r'' such that $\mathcal{E}_{\mathrm{p}} \vdash p_1 \mid_{\mathrm{Act}} q \approx r'$ and $\mathcal{E}_{\mathrm{p}} \vdash p_2 \mid_{\mathrm{Act}} q \approx r''$. We thus have that $\mathcal{E}_{\mathrm{p}} \vdash p \mid_{\mathrm{Act}} q \approx r' + r''$, which is a BCCSP process, and we are done. □

The above lemma is the key step in the elimination of $\mid_{\mathrm{Act}}$ from closed terms. Namely:

Proposition 4. *For every closed BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$ process p there exists a closed BCCSP process q such that $\mathcal{E}_{\mathrm{p}} \vdash p \approx q$.*

Proof. The proof is straightforward by structural induction on p and using Lemma 7 in the case that p is of the form $p_1 \mid_{\mathrm{Act}} p_2$, for some BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$ processes p_1, p_2. □

The ground-completeness of $\mathcal{E}_{\mathrm{p}}$ over BCCSP$^{\mathrm{p}}_{\mathrm{Act}}$ follows from Proposition 4 and the ground-completeness of $\mathcal{E}_0$ over BCCSP [28].

5 The Case of Restriction

In this section we apply the reduction technique described in Sect. 3 to show that bisimilarity does not have a finite, ground-complete equational axiomatisation over the recursion and relabelling free fragment of CCS. In detail, we assume a finite set of action names Act, and we let $\overline{\mathrm{Act}}$ denote the set of action co-names, i.e., $\overline{\mathrm{Act}} = \{\overline{a} \mid a \in \mathrm{Act}\}$. As usual, we postulate that $\overline{\overline{a}} = a$ and $a \neq \overline{a}$ for all $a \in \mathrm{Act}$.

Table 7. The SOS rules for $\mathrm{CCS}^{\mathtt{r}}$ operators ($\mu \in \mathrm{Act}_\tau$, $\alpha \in \mathrm{Act} \cup \overline{\mathrm{Act}}$).

$$\text{(r1)}\ \frac{t \xrightarrow{\mu} t'}{t \mid u \xrightarrow{\mu} t' \mid u} \qquad \text{(r2)}\ \frac{u \xrightarrow{\mu} u'}{t \mid u \xrightarrow{\mu} t \mid u'} \qquad \text{(r3)}\ \frac{t \xrightarrow{\alpha} t' \quad u \xrightarrow{\overline{\alpha}} u'}{t \mid u \xrightarrow{\tau} t' \mid u'}$$

$$\text{(r4)}\ \frac{t \xrightarrow{\alpha} t'}{t\backslash L \xrightarrow{\alpha} t'\backslash R}\ \alpha, \overline{\alpha} \notin R \qquad \text{(r5)}\ \frac{t \xrightarrow{\tau} t'}{t\backslash L \xrightarrow{\tau} t'\backslash R}$$

Then, we let $\mathrm{Act}_\tau = \mathrm{Act} \cup \overline{\mathrm{Act}} \cup \{\tau\}$, where $\tau \notin \mathrm{Act} \cup \overline{\mathrm{Act}}$. Henceforth, we let $\mu, \nu, \ldots$ range over actions in Act_τ, $\alpha, \beta, \ldots$ range over actions in $\mathrm{Act} \cup \overline{\mathrm{Act}}$, and $a, b, \ldots$ range over actions in Act.

We denote by $\mathrm{CCS}^{\mathtt{r}}$ the recursion and relabelling free fragment of CCS with the full merge operator (denoted by $\mid$) generated by the following grammar:

$$t ::= \quad \mathbf{0} \quad \mid \quad x \quad \mid \quad \mu.t \quad \mid \quad t + t \quad \mid \quad t \mid t \quad \mid \quad t\backslash R \qquad\qquad (\mathrm{CCS}^{\mathtt{r}})$$

where $x \in \mathrm{Var}$, $\mu \in \mathrm{Act}_\tau$ and $R \subseteq \mathrm{Act} \cup \overline{\mathrm{Act}}$.

Following [32], the action symbol τ will result from the synchronised occurrence of the complementary actions α and $\overline{\alpha}$, as described by the inference rules in Table 7.

We recall that the *restriction operator* $t\backslash R$ prevents t (and its derivatives) from performing any α-transition, for all $\alpha \in R$.

The operational semantics of $\mathrm{CCS}^{\mathtt{r}}$ is obtained by adding the inference rules for the full merge and the restriction operator given in Table 7 to the rules for BCCSP operators given in Table 1. In the technical results that follow, we will need to distinguish between transitions over $\mathrm{CCS}^{\mathtt{r}}$ processes, and transitions over CCS_a processes. Hence, to avoid possible confusion and favour thus readability, we adopt the same strategy we used in Sect. 4, and use special symbols to distinguish them: we denote the transition relation induced by the rules in Tables 1 and 7 by $\rightarrow_{\mathtt{r}}$, and bisimilarity over $\mathcal{P}(\mathrm{CCS}^{\mathtt{r}})$ by $\sim_{\mathtt{r}}$.

Definition 9 (Bisimulation over $\mathrm{CCS}^{\mathtt{r}}$). *Bisimulation relations over $\mathit{CCS}^{\mathtt{r}}$ processes are defined by applying Definition 2 to the LTS $(\mathcal{P}(\mathit{CCS}^{\mathtt{r}}), \mathrm{Act}_\tau, \rightarrow_{\mathtt{r}})$ induced by the SOS rules in Table 7. We use the symbol $\sim_{\mathtt{r}}$ to denote bisimilarity over $\mathit{CCS}^{\mathtt{r}}$ processes.*

5.1 The Negative Result

Our main goal in this section is to prove the following theorem:

Theorem 8. *Bisimilarity has no finite, ground-complete axiomatisation over $\mathit{CCS}^{\mathtt{r}}$.*

To this end, as already done in Sects. 4.2 and 4.3, we exploit the reduction technique from [11] and Moller's non-finite axiomatisability result from CCS_a (Theorem 2). In detail:

- We select a particular action $a \in \mathrm{Act}$.
- We consider the language CCS_a and the instantiation of the equations φ_n in the family $\varPhi$ over processes defined using only that action.

- We provide a translation mapping from CCS$^{\mathtt{r}}$ to CCS$_a$, denoted by $\mathtt{r}_a$, whose definition will be parametric in the chosen action a, that will allow us to eliminate all CCS$^{\mathtt{r}}$ terms in which the execution of a is restricted, while ensuring the possibility to perform any a-transition that is unrestricted.

It will be then enough to show that the mapping $\mathtt{r}_a$ is structural, it preserves the soundness of equations from CCS$^{\mathtt{r}}$ to CCS$_a$, and it is ground Φ-reflecting, to obtain the validity of the lifting of the negative result in Theorem 2 to CCS$^{\mathtt{r}}$, proving thus Theorem 8.

5.2 The Reduction

Choose an action a from the action set Act. Then we define a mapping $\mathtt{r}_a \colon \mathbb{T}(\mathrm{CCS}^{\mathtt{r}}) \to \mathbb{T}(\mathrm{CCS}_a)$ allowing us to rewrite any CCS$^{\mathtt{r}}$ term into a CCS$_a$ term.

Definition 10 (The mapping $\mathtt{r}_a$). *The mapping $\mathtt{r}_a \colon \mathbb{T}(CCS^{\mathtt{r}}) \to \mathbb{T}(CCS_a)$ is defined inductively as follows:*

$$\mathtt{r}_a(\mathbf{0}) = \mathbf{0} \qquad \mathtt{r}_a(t+u) = \mathtt{r}_a(t) + \mathtt{r}_a(u)$$

$$\mathtt{r}_a(x) = x \qquad \mathtt{r}_a(t \mid u) = \mathtt{r}_a(t) \parallel \mathtt{r}_a(u)$$

$$\mathtt{r}_a(\mu.t) = \begin{cases} a.\mathtt{r}_a(t) & \textit{if } \mu = a \\ \mathbf{0} & \textit{otherwise} \end{cases} \qquad \mathtt{r}_a(t \backslash R) = \begin{cases} \mathtt{r}_a(t) & \textit{if } a, \overline{a} \notin R \\ \mathbf{0} & \textit{otherwise.} \end{cases}$$

Notice that a is the only action that may possibly occur in $\mathtt{r}_a(t)$, for each $t \in \mathbb{T}(\mathrm{CCS}^{\mathtt{r}})$.

We now proceed to show that the mapping $\mathtt{r}_a$ is a well-defined reduction, according to Definition 3. As a first step, we notice that $\mathtt{r}_a$ is structural by definition.

Lemma 8. *The mapping $\mathtt{r}_a$ is structural.*

We now proceed to prove two technical lemmas, that will be useful to prove that $\mathtt{r}_a$ is a reduction.

Lemma 9. *For all $p \in \mathcal{P}(CCS^{\mathtt{r}})$, and $q \in \mathcal{P}(CCS_a)$, if $\mathtt{r}_a(p) \xrightarrow{a} q$, then there exists some $p' \in \mathcal{P}(CCS^{\mathtt{r}})$ such that $p \xrightarrow{a}_{\mathtt{r}} p'$ and $\mathtt{r}_a(p') = q$.*

Proof. The proof proceeds by structural induction over the $\mathcal{P}(\mathrm{CCS}^{\mathtt{r}})$ process p. As for prefixing, nondeterministic choice, and parallel composition the proof is analogous to that of the corresponding steps in Lemma 2, we limit ourselves to present only the inductive step related to the restriction operator.

Let $p = p_1 \backslash R$. We can distinguish two cases, according to whether $a \in R$ or $\overline{a} \in R$, or not (see Definition 10):

- Assume that $a \in R$ or $\overline{a} \in R$. Then $\mathtt{r}_a(p) = \mathbf{0}$, and this case becomes vacuous as $\mathtt{r}_a(p) \overset{a}{\not\to}$.
- Assume now that $a, \overline{a} \notin R$. Then $\mathtt{r}_a(p) = \mathtt{r}_a(p_1)$ and $\mathtt{r}_a(p_1) \xrightarrow{a} q$. By induction over p_1, there is some $p_1' \in \mathcal{P}(\mathrm{CCS}^{\mathtt{r}})$ such that $p_1 \xrightarrow{a}_{\mathtt{r}} p_1'$ and $\mathtt{r}_a(p_1') = q$. Since $a, \overline{a} \notin R$, by an application of rule (r4) from Table 7 we obtain that $p \xrightarrow{a}_{\mathtt{r}} p_1' \backslash R$. Finally, by Definition 10, since $a, \overline{a} \notin R$ it follows that $\mathtt{r}_a(p_1' \backslash R) = \mathtt{r}_a(p_1') = q$ as required. □

Lemma 10. *For all $p, p' \in \mathcal{P}(CCS^{\mathtt{r}})$, if $p \xrightarrow{a}_{\mathtt{r}} p'$, then $\mathtt{r}_a(p) \xrightarrow{a} \mathtt{r}_a(p')$.*

Proof. The proof proceeds by induction over the size of the proof for the transition $p \xrightarrow{a}_{\mathtt{r}} p'$. Also in this case, given the similarities with the proofs of the corresponding cases in Lemma 3, we limit ourselves to analyse only the case in which the last inference rule from Table 7 that is applied in the proof for $p \xrightarrow{a}_{\mathtt{r}} p'$ is rule (r4), i.e., the rule for restriction. (In particular, we remark that since $a \neq \tau$, rules (r3) and (r5) cannot be applied as the last rules in the proof for $p \xrightarrow{a}_{\mathtt{r}} p'$.)

Let (r4) be the last rule applied in the proof. In this case, $p = p_1 \backslash R$, $p_1 \xrightarrow{a}_{\mathtt{r}} p_1'$, and $p' = p_1' \backslash R$. In particular, the application of rule (r4) guarantees that $a, \overline{a} \notin R$, so that $\mathtt{r}_a(p) = \mathtt{r}_a(p_1)$, by Definition 10. By induction we obtain that $\mathtt{r}_a(p_1) \xrightarrow{a} \mathtt{r}_a(p_1')$. Clearly, this directly gives $\mathtt{r}_a(p) \xrightarrow{a} \mathtt{r}_a(p_1')$. Since, moreover, $a, \overline{a} \notin R$, by Definition 10 we also get that $\mathtt{r}_a(p') = \mathtt{r}_a(p_1' \backslash R) = \mathtt{r}_a(p_1')$. We can then conclude that $\mathtt{r}_a(p) \xrightarrow{a} \mathtt{r}_a(p')$. □

We now have all the ingredients necessary to prove that the mapping $\mathtt{r}_a$ is a well-defined ground Φ-reflecting reduction.

Proposition 5. *The mapping $\mathtt{r}_a$ satisfies the following properties:*

1. *For each $t, u \in \mathbb{T}(CCS^{\mathtt{r}})$, $t \sim_{\mathtt{r}} u$ implies $\mathtt{r}_a(t) \sim \mathtt{r}_a(u)$.*
2. *The mapping $\mathtt{r}_a$ is ground Φ-reflecting.*

Proof. We prove the two statements separately.

1. First of all, for each $t \in \mathbb{T}(\mathrm{CCS}_a)$ we define $t_a^{\mathtt{r}} \in \mathbb{T}(\mathrm{CCS}^{\mathtt{r}})$ as follows:

$$\mathbf{0}_a^{\mathtt{r}} = \mathbf{0} \qquad x_a^{\mathtt{r}} = a \qquad (a.t)_a^{\mathtt{r}} = a.t_a^{\mathtt{r}}$$
$$(t+u)_a^{\mathtt{r}} = t_a^{\mathtt{r}} + u_a^{\mathtt{r}} \qquad (t \parallel u)_a^{\mathtt{r}} = t_a^{\mathtt{r}} \mid u_a^{\mathtt{r}}.$$

   It is then immediate to check that for each $t \in \mathbb{T}(\mathrm{CCS}_a)$ we have that $\mathtt{r}_a(t_a^{\mathtt{r}}) = t$. Then, given any CCS_a substitution σ, we define $\sigma_a^{\mathtt{r}}$ as the $\mathrm{CCS}^{\mathtt{r}}$ substitution such that $\sigma_a^{\mathtt{r}}(x) = (\sigma(x))_a^{\mathtt{r}}$. The claim then follows by applying the same reasoning used in the proof of Proposition 2.
2. Consider the family of equations $\Phi_{\mathtt{r}}$ defined as follows:

$$\varphi_{\mathtt{r}}^n : a \mid \sum_{i=1}^{n} a^i \approx a.\sum_{i=1}^{n} a^i + \sum_{j=1}^{n} a^{j+1} \qquad (n \geq 0)$$
$$\Phi_{\mathtt{r}} = \{\varphi_{\mathtt{r}}^n \mid n \geq 0\}.$$

   It is straightforward to prove that $\mathtt{r}_a(\varphi_{\mathtt{r}}^n) = \varphi_n$ for each $n \in \mathbb{N}$, and thus that $\mathtt{r}_a$ is ground Φ-reflecting. □

Theorem 8 is then a immediate consequence of Lemma 8, Proposition 5, Theorem 3, and Theorem 2.

6 Concluding Remarks

In this paper, we have exploited the reduction technique from [11], for the lifting of negative results across process algebras, to prove the non-finite axiomatisability of various extensions of BCCSP modulo bisimilarity. In detail, we have proved that bisimilarity does not admit a finite, ground-complete axiomatisation 1. over $\mathrm{BCCSP}^{\mathrm{p}}_{A}$, i.e., BCCSP enriched with a CSP-like parallel operator $|_A$, with $A \subset \mathrm{Act}$, 2. over $\mathrm{BCCSP}^{\mathrm{p}}_{\tau}$, i.e., BCCSP enriched with τ-prefixing, $\tau \notin \mathrm{Act}$, and CSP-like parallel operators with any possible synchronisation set, and 3. over $\mathrm{CCS}^{\mathrm{r}}$, i.e., the recursion and relabelling free fragment of CCS. Interestingly, among all these negative results, we found a positive one: if we consider only the CSP-like parallel operator $|_{\mathrm{Act}}$, forcing all the actions in the parallel components to be synchronised, then a finite, ground-complete axiomatisation of bisimilarity over $\mathrm{BCCSP}^{\mathrm{p}}_{\mathrm{Act}}$ exists. Moreover, we have proved that the reduction technique from [11] cannot be applied in the case of $\mathrm{BCCSP}^{\mathrm{p}}$, i.e., BCCSP enriched with all parallel operators $|_A$ for $A \subseteq \mathrm{Act}$.

As a natural step for future work, we will provide a direct proof of the fact that bisimilarity does not admit a finite, ground-complete axiomatisation over $\mathrm{BCCSP}^{\mathrm{p}}$.

Moreover, we plan to investigate how far the lifting technique of [11] can be pushed. In particular, we are interested in studying whether (some variations of) it can be used to lift known results for strong behavioural equivalences to their weak counterparts or to potentially extend results over weak behavioural congruences (such as Theorem 10 presented in [1]) to new settings.

Another possible direction for future work, would be to focus on full recursion free CCS. Aceto, Ingólfsdóttir, Luttik and van Tilburg gave an equational axiomatisation of bisimilarity over recursion-free CCS with interleaving parallel composition and the left-merge operator in [13]. That result crucially depends on the fact that restriction and relabelling distribute over interleaving parallel composition. On the other hand, neither restriction nor relabelling distribute over parallel composition in the presence of synchronisation. Obtaining a complete axiomatisation of full recursion free CCS modulo bisimilarity, with restriction, relabelling and parallel composition that allows for synchronisation is a natural, and very challenging, avenue for future research.

Acknowledgements. The first author thanks Frits Vaandrager for the joint work they did about thirty years ago and Rob van Glabbeek for asking the questions that led to the research presented in this article.

We thank the reviewers for their valuable comments on our paper. In particular, we are very grateful to the reviewer who spotted a technical error in the original manuscript.

References

1. Aceto, L., Anastasiadi, E., Castiglioni, V., Ingólfsdóttir, A., Luttik, B.: In search of lost time: axiomatising parallel composition in process algebras. In: Proceedings of LICS 2021, pp. 1–14. IEEE (2021). https://doi.org/10.1109/LICS52264.2021.9470526
2. Aceto, L., Anastasiadi, E., Castiglioni, V., Ingólfsdóttir, A., Luttik, B., Pedersen, M.R.: On the axiomatisability of priority III: priority strikes again. Theor. Comput. Sci. **837**, 223–246 (2020). https://doi.org/10.1016/j.tcs.2020.07.044

3. Aceto, L., Bloom, B., Vaandrager, F.W.: Turning SOS rules into equations. Inf. Comput. **111**(1), 1–52 (1994). https://doi.org/10.1006/inco.1994.1040
4. Aceto, L., Castiglioni, V., Fokkink, W., Ingólfsdóttir, A., Luttik, B.: Are two binary operators necessary to finitely axiomatise parallel composition? In: Proceedings of CSL 2021. LIPIcs, vol. 183, pp. 8:1–8:17 (2021). https://doi.org/10.4230/LIPIcs.CSL.2021.8
5. Aceto, L., Castiglioni, V., Ingólfsdóttir, A., Luttik, B., Pedersen, M.R.: On the axiomatisability of parallel composition: a journey in the spectrum. In: Proceedings of CONCUR 2020. LIPIcs, vol. 171, pp. 18:1–18:22 (2020). https://doi.org/10.4230/LIPIcs.CONCUR.2020.18
6. Aceto, L., Fokkink, W., van Glabbeek, R.J., Ingólfsdóttir, A.: Nested semantics over finite trees are equationally hard. Inf. Comput. **191**(2), 203–232 (2004). https://doi.org/10.1016/j.ic.2004.02.001
7. Aceto, L., Fokkink, W., Ingólfsdóttir, A.: A menagerie of non finitely based process semantics over BPA* - from ready simulation to completed traces. Math. Struct. Comput. Sci. **8**(3), 193–230 (1998). http://journals.cambridge.org/action/displayAbstract?aid=44743
8. Aceto, L., Fokkink, W., Ingólfsdóttir, A., Luttik, B.: CCS with Hennessy's merge has no finite-equational axiomatization. Theor. Comput. Sci. **330**(3), 377–405 (2005). https://doi.org/10.1016/j.tcs.2004.10.003
9. Aceto, L., Fokkink, W., Ingolfsdottir, A., Luttik, B.: Finite equational bases in process algebra: results and open questions. In: Middeldorp, A., van Oostrom, V., van Raamsdonk, F., de Vrijer, R. (eds.) Processes, Terms and Cycles: Steps on the Road to Infinity. LNCS, vol. 3838, pp. 338–367. Springer, Heidelberg (2005). https://doi.org/10.1007/11601548_18
10. Aceto, L., Fokkink, W., Ingólfsdóttir, A., Luttik, B.: A finite equational base for CCS with left merge and communication merge. ACM Trans. Comput. Log. **10**(1), 6:1–6:26 (2009). https://doi.org/10.1145/1459010.1459016
11. Aceto, L., Fokkink, W., Ingólfsdóttir, A., Mousavi, M.R.: Lifting non-finite axiomatizability results to extensions of process algebras. Acta Inf. **47**(3), 147–177 (2010). https://doi.org/10.1007/s00236-010-0114-7
12. Aceto, L., Fokkink, W., Ingólfsdóttir, A., Nain, S.: Bisimilarity is not finitely based over BPA with interrupt. Theor. Comput. Sci. **366**(1–2), 60–81 (2006). https://doi.org/10.1016/j.tcs.2006.07.003
13. Aceto, L., Ingólfsdóttir, A., Luttik, B., van Tilburg, P.: Finite equational bases for fragments of CCS with restriction and relabelling. In: Ausiello, G., Karhumäki, J., Mauri, G., Ong, L. (eds.) TCS 2008. IIFIP, vol. 273, pp. 317–332. Springer, Boston, MA (2008). https://doi.org/10.1007/978-0-387-09680-3_22
14. Baeten, J.C.M., Basten, T., Reniers, M.: Process Algebra: Equational Theories of Communicating Processes. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2009). https://doi.org/10.1017/CBO9781139195003
15. Baeten, J.C.M., Vaandrager, F.W.: An algebra for process creation. Acta Informatica **29**(4), 303–334 (1992). https://doi.org/10.1007/BF01178776
16. Bergstra, J.A., Klop, J.W.: Process algebra for synchronous communication. Inf. Control **60**(1–3), 109–137 (1984). https://doi.org/10.1016/S0019-9958(84)80025-X
17. Bergstra, J.A., Klop, J.W.: Algebra of communicating processes with abstraction. Theor. Comput. Sci. **37**, 77–121 (1985). https://doi.org/10.1016/0304-3975(85)90088-X
18. Bloom, B., Istrail, S., Meyer, A.R.: Bisimulation can't be traced. J. ACM **42**(1), 232–268 (1995). https://doi.org/10.1145/200836.200876
19. Brookes, S.D., Roscoe, A.W., Walker, D.J.: An operational semantics for CSP. Report, University of Oxford (1986)
20. Chen, T., Fokkink, W., van Glabbeek, R.J.: On the axiomatizability of impossible futures. Log. Methods Comput. Sci. **11**(3) (2015). https://doi.org/10.2168/LMCS-11(3:17)2015

21. Fokkink, W.J., Luttik, S.P.: An ω-complete equational specification of interleaving. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 729–743. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45022-X_61
22. Glabbeek, R.J.: The linear time - branching time spectrum. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 278–297. Springer, Heidelberg (1990). https://doi.org/10.1007/BFb0039066
23. van Glabbeek, R.J.: Full abstraction in structural operational semantics (extended abstract). In: Proceedings of AMAST 1993, pp. 75–82. Workshops in Computing (1993)
24. van Glabbeek, R., Vaandrager, F.: Modular specifications in process algebra. In: Wirsing, M., Bergstra, J.A. (eds.) Algebraic Methods 1987. LNCS, vol. 394, pp. 465–506. Springer, Heidelberg (1989). https://doi.org/10.1007/BFb0015049
25. van Glabbeek, R., Vaandrager, F.: Petri net models for algebraic theories of concurrency. In: de Bakker, J.W., Nijman, A.J., Treleaven, P.C. (eds.) PARLE 1987. LNCS, vol. 259, pp. 224–242. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-17945-3_13
26. Groote, J.F., Vaandrager, F.W.: Structured operational semantics and bisimulation as a congruence. Inf. Comput. **100**(2), 202–260 (1992). https://doi.org/10.1016/0890-5401(92)90013-6
27. Groote, J.F., de Vink, E.P.: An axiomatization of strong distribution bisimulation for a language with a parallel operator and probabilistic choice. In: ter Beek, M.H., Fantechi, A., Semini, L. (eds.) From Software Engineering to Formal Methods and Tools, and Back. LNCS, vol. 11865, pp. 449–463. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30985-5_26
28. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. J. ACM **32**(1), 137–161 (1985). https://doi.org/10.1145/2455.2460
29. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall (1985)
30. Keller, R.M.: Formal verification of parallel programs. Commun. ACM **19**(7), 371–384 (1976). https://doi.org/10.1145/360248.360251
31. Middelburg, C.A.: Probabilistic process algebra and strategic interleaving. Sci. Ann. Comput. Sci. **30**(2), 205–243 (2020). https://doi.org/10.7561/SACS.2020.2.205
32. Milner, R.: Communication and Concurrency. PHI Series in Computer Science. Prentice Hall (1989)
33. Moller, F.: Axioms for Concurrency. Ph.D. thesis, Department of Computer Science, University of Edinburgh, July 1989. https://era.ed.ac.uk/bitstream/handle/1842/11182/Moller1989.pdf, report CST-59-89. Also published as ECS-LFCS-89-84
34. Moller, F.: The importance of the left merge operator in process algebras. In: Paterson, M.S. (ed.) ICALP 1990. LNCS, vol. 443, pp. 752–764. Springer, Heidelberg (1990). https://doi.org/10.1007/BFb0032072
35. Moller, F.: The nonexistence of finite axiomatisations for CCS congruences. In: Proceedings of LICS 1990, pp. 142–153 (1990). https://doi.org/10.1109/LICS.1990.113741
36. Park, D.: Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) GI-TCS 1981. LNCS, vol. 104, pp. 167–183. Springer, Heidelberg (1981). https://doi.org/10.1007/BFb0017309
37. Plotkin, G.D.: A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University (1981)
38. de Simone, R.: Higher-level synchronising devices in Meije-SCCS. Theor. Comput. Sci. **37**, 245–267 (1985). https://doi.org/10.1016/0304-3975(85)90093-3
39. Vaandrager, F.W.: Algebraic techniques for concurrency and their application. Ph.D. thesis, University of Amsterdam, February 1990