Are Two Binary Operators Necessary to Finitely Axiomatise Parallel Composition?

Anonymous Author(s)

Abstract

Bergstra and Klop have shown that *bisimilarity* has a *finite* equational axiomatisation over ACP/CCS extended with the binary *left* and *communication merge* operators. Moller proved that auxiliary operators are *necessary* to obtain a finite axiomatisation of bisimilarity over CCS, and Aceto et al. showed that this remains true when *Hennessy's merge* is added to that language. These results raise the question of whether there is *one* auxiliary *binary* operator whose addition to CCS leads to a finite axiomatisation of bisimilarity. This study provides a *negative answer* to that question based on a number of reasonable assumptions.

Keywords Equational logic, CCS, bisimulation, parallel composition, non-finitely based algebras.

1 Introduction

The purpose of this paper is to provide an answer to the following problem (see [1, Problem 8]): *Are the left merge and the communication merge operators* necessary *to obtain a finite equational axiomatisation of bisimilarity over the language CCS?* The interest in this problem is twofold, as an answer to it would: 1. clarify the status of the auxiliary operators *left merge* and *communication merge*, proposed in [11], in the finite axiomatisation of parallel composition, and 2. give further insight into properties that auxiliary operators used in the finite equational characterisation of parallel composition ought to afford.

We prove that, under some reasonable simplifying assumptions, there is no auxiliary binary operator that can be added to CCS to yield a finite equational axiomatisation of bisimilarity. Despite falling short of solving the above-mentioned problem in full generality, our negative result is a substantial generalisation of previous non-finite-axiomatisability theorems by Moller [23, 24] and Aceto et al. [4].

In order to put our contribution in context, we first describe the history of the problem we tackle and then give a bird's eye view of our results.

1.1 The story so far

In the late 1970s, Milner developed the *Calculus of Communicating Systems* (CCS) [20], a formal language based on a message-passing paradigm and aimed at describing communicating processes from an operational point of view. In detail, a *labelled transition system* (LTS) [18] was used to equip language expressions with an *operational semantics* [27]; the LTS giving the operational semantics of each CCS expression was defined using a collection of syntax-driven rules. The analysis of process behaviour was carried out via an observational *bisimulation*-based theory [26] that defines when two states in an LTS describe the same behaviour.

In particular, CCS included a *parallel composition operator* $\|$ to model the interactions among processes. Such an operator, also known as *merge* [11, 12], allows one both to *interleave* the behaviours of its argument processes (modelling concurrent computations) and to enable some form of *synchronisation* between them (modelling interactions). Later on, in collaboration with Hennessy, Milner studied the *equational theory* of (recursion free) CCS and proposed a *ground-complete axiomatisation* for it modulo bisimilarity [17]. More precisely, Hennessy and Milner presented a set $\mathcal{E}$ of *equational axioms* from which all equations over closed CCS terms (namely those with no occurrences of variables) that are *valid modulo bisimilarity* can be derived using the rules of *equational logic* [28]. Notably, the set $\mathcal{E}$ included infinitely many axioms, which were instances of the *expansion law* that was used to 'simulate equationally' the operational semantics of the parallel composition operator.

The ground-completeness result by Hennessy and Milner started the quest for a finite axiomatisation of CCS's parallel composition operator modulo bisimilarity.

Thanks to the work by Bergstra and Klop on the *Algebra of Communicating Processes* (ACP) in [11], it was possible to prove that if we enrich CCS with two auxiliary operators, namely the *left merge* $\mathbin{\|\!\!\!\_}$ and the *communication merge* $|$, expressing respectively one step in the asymmetric pure interleaving and the synchronous behaviour of $\|$, then a finite ground-complete axiomatisation modulo bisimilarity exists. Their result was then strengthened by Aceto et al. in [6], where it is proved that, over the fragment of CCS without recursion, restriction and relabelling, the auxiliary operators $\mathbin{\|\!\!\!\_}$ and $|$ allow for finitely axiomatising $\|$ modulo bisimilarity also when CCS terms with variables are considered. Moreover, in [9] that result is extended to the fragment of CCS with relabelling and restriction, but without communication. From those studies, we can infer that the left merge and communication merge operators are *sufficient* to finitely axiomatise parallel composition modulo bisimilarity. But is the addition of auxiliary operators *necessary* to obtain a finite equational axiomatisation, or can the use of the expansion law in the original axiomatisation of bisimilarity by Hennessy and Milner be replaced by a finite set of sound CCS equations?

LICS 2020, July 8–12, 2020, Beijing, China
2020.

To address that question, in [23, 24] Moller considered a minimal fragment of CCS, namely the one with action prefixing, nondeterministic choice and purely interleaving parallel composition, and proved that, even in the presence of a single action, bisimilarity does not afford a finite ground-complete axiomatisation over the closed terms in that language. This showed that auxiliary operators are indeed necessary to obtain a finite equational axiomatisation of bisimilarity.

Adapting Moller's proof technique, Aceto et al. proved, in [4], that if we replace $\mathbin{|\!\!\lfloor}$ and $\mid$ with the so called *Hennessy's merge* $\mathbin{|\!\!/}$ [16], which is a sort of combination of $\mathbin{|\!\!\lfloor}$ and $\mid$ as it denotes an asymmetric interleaving with communication, then the collection of equations that hold modulo bisimilarity over the recursion, restriction and relabelling free fragment of CCS enriched with $\mathbin{|\!\!/}$ is not finitely based (in the presence of at least two distinct complementary actions).

A natural question that arises from those *negative* results is the following, which formalises [1, Problem 8]:

> *Can one obtain a finite axiomatisation of the parallel composition operator in bisimulation semantics by adding* only one binary operator *to the signature of (recursion, restriction, relabelling free) CCS?* (P)

In this paper, we provide a partial *negative answer* to that question.

1.2 Our contribution

We study the problem of whether it is possible to obtain a finite ground-complete axiomatisation of bisimilarity by adding only one auxiliary binary operator to CCS. In detail, we analyse the axiomatisability of parallel composition over the language CCS_f, namely CCS enriched with a binary operator f that we use to express $\|$ as a derived operator. We prove that, under reasonable assumptions, the auxiliary operator f alone does not allow us to obtain a finite ground-complete axiomatisation of CCS_f modulo bisimilarity.

To this end, we assume that the operational semantics of f is given by rules in the de Simone format [14] (Assumption 1) and that the behaviour of the parallel composition operator is expressed equationally by a law that is akin to the one used by Bergstra and Klop to define $\|$ in terms of $\mathbin{|\!\!\lfloor}$ and $\mid$ (Assumption 2). We then argue that the latter assumption yields that the equation

$$x \| y \approx f(x,y) + f(y,x) \tag{A}$$

is valid modulo bisimilarity. Next we proceed by a case analysis over the possible sets of de Simone rules defining the behaviour of f, in such a way that the validity of Equation (A) modulo bisimilarity is guaranteed, and, for each case, we show the desired negative result using proof-theoretic techniques that have their roots in Moller's classic results in [23, 24]. This means that we identify a (case-specific) property of terms, called the *witness property* and denoted by W, associated with each finite set $\mathcal{E}$ of sound axioms and a natural number n. The idea is that, when n is *large enough*, whenever an equation $p \approx q$ is derivable from $\mathcal{E}$, then either both terms p and q satisfy $W_n(\mathcal{E})$, or none of them does. The negative result is then obtained by exhibiting a (case-specific) infinite family of valid equations $\{e_n \mid n \geq 0\}$ in which the witness property is not preserved, that is, for each $n \geq 0$, $W_n(\mathcal{E})$ is satisfied only by one side of e_n. Due to the choice of W, this means that the equations in the family cannot all be derived from a finite set of valid axioms and therefore no finite, sound axiom system can be complete.

To the best of our knowledge, in this paper we propose the first non-finite axiomatisability result for a process algebra in which one of the operators, namely the auxiliary operator f, does not have a fixed semantics. However, for our technical developments, it has been necessary to restrict the search space for f by means of the aforementioned simplifying assumptions and of a third, mild one (Assumption 3) that plays a role in the proof of one of the claims in our combinatorial analysis of the rules that f may have (see Lemma 4.3 and Remark 1). There are three main reasons behind Assumption 1:

- The de Simone format is the simplest congruence format for bisimilarity. Hence we must be able to deal with this case before proceeding to any generalisation.
- The specification of parallel composition, left merge and communication merge operators (and of the vast majority of process algebraic operators) is in de Simone format. Hence, that format was a natural choice also for operator f.
- The simplicity of the de Simone rules allows us to reduce considerably the complexity of our case analysis over the sets of available rules for the operator f. However, as witnessed by the developments in this article, even with this simplification, the proof of the desired negative result requires a large amount of delicate, technical work.

Assumptions 2 and 3 still allow us to obtain a significant generalisation of related works, such as [4], as we can see them as an attempt to identify the requirements needed to apply Moller's proof technique to Hennessy's merge like operators.

Even though the vast literature on process algebras offers a plethora of non-finite axiomatisability results for a variety of languages and semantics (see, for instance, the survey [5] from 2005), we are not aware of any previous attempt at proving a result akin to the one we present here. We have already addressed at length how our contribution fits within the study of the equational logic of processes and how it generalises previous results in that field. The proof-theoretic tools and the approach we adopt in proving our main theorem, which links equational logic with structural operational semantics and builds on a number of previous achievements (such as those in [2]), may have independent interest for researchers in logic in computer science. To our mind, achieving an answer to question (P) in full generality would be very pleasing for the concurrency-theory community, as it would finally clarify the canonical role of Bergstra and Klop's auxiliary operators in the finite axiomatisation of parallel composition modulo bisimilarity.

1.3 Organisation of contents

After a brief review, in Section 2, of basic notions on process semantics, CCS and equational logic, in Section 3 we present the two main simplifying assumptions under which we tackle the problem (P). In Section 4 we study the operational semantics of auxiliary operators f meeting our assumptions. In Section 5 we give a detailed presentation of the proof strategy we will follow to address (P). Sections 6–11 are then devoted to the technical development of our negative results. Due to space limitations, all proofs have been moved to the Appendix.

2 Background

We begin by introducing the basic definitions and results on which the technical developments to follow are based.

2.1 Labelled Transition Systems and Bisimilarity

As semantic model we consider classic *labelled transition systems* [18].

Definition 2.1. A *labelled transition system* (LTS) is a triple $(S, A, \longrightarrow)$, where S is a set of *states*, A is a set of *actions*, and $\longrightarrow \subseteq S \times A \times S$ is a (*labelled*) *transition relation*.

As usual, we use $t \xrightarrow{\mu} t'$ in lieu of (t, μ, t'). For each $t \in S$ and $\mu \in A$, we write $t \xrightarrow{\mu}$ if $t \xrightarrow{\mu} t'$ holds for some t', and $t \not\xrightarrow{\mu}$ otherwise. The *initials* of t are the actions that label the outgoing transitions of t, that is, $\mathrm{init}(t) = \{\mu \mid t \xrightarrow{\mu}\}$. For a sequence of actions $s = \mu_1 \cdots \mu_k$ ($k \geq 0$), and states t, t', we write $t \xrightarrow{s} t'$ iff there exists a sequence of transitions $t = t_0 \xrightarrow{\mu_1} t_1 \xrightarrow{\mu_2} \cdots \xrightarrow{\mu_k} t_k = t'$. If $t \xrightarrow{s} t'$ holds for some state t', then s is a *trace* of t. Moreover, we say that s is a maximal trace of t if $\mathrm{init}(t') = \emptyset$. By means of traces, we associate two classic notions with a state t: its *depth*, denoted by $depth(t)$, and its *norm*, denoted by $norm(t)$. For a state t whose set of traces is finite, they express, respectively, the length of a *longest* trace of t and that of a *shortest* maximal trace. Formally, $depth(t) = \sup\{k \mid t \text{ has a trace of length } k\}$ and $norm(t) = \inf\{k \mid t \text{ has a maximal trace of length } k\}$.

In this paper, we shall consider the states in a labelled transition system modulo bisimilarity [21, 26], allowing us to establish whether two processes have the same behaviour.

Definition 2.2. Let $(S, A, \longrightarrow)$ be a labelled transition system. *Bisimilarity*, denoted by $\underline{\leftrightarrow}$, is the largest binary symmetric relation over S such that whenever $t \,\underline{\leftrightarrow}\, u$ and $t \xrightarrow{\mu} t'$, then there is a transition $u \xrightarrow{\mu} u'$ with $t' \,\underline{\leftrightarrow}\, u'$. If $t \,\underline{\leftrightarrow}\, u$, then we say that t and u are *bisimilar*.

It is well-known that bisimilarity is an equivalence relation (see, e.g., [21, 26]). Moreover, two bisimilar states have the same depth and norm.

2.2 The Language CCS$_f$

The language we consider in this paper is obtained by adding a single binary operator f to the recursion, restriction and relabelling free subset of Milner's CCS [21], henceforth referred to as CCS$_f$, and is given by the following grammar:

$$t ::= \mathbf{0} \mid x \mid a.t \mid \bar{a}.t \mid \tau.t \mid t + t \mid t \parallel t \mid f(t, t) \ ,$$

where x is a variable drawn from a countably infinite set $\mathcal{V}$, a is an action, and $\bar{a}$ is its complement. We assume that the actions a and $\bar{a}$ are distinct. Following Milner [21], the action symbol τ will result from the synchronised occurrence of the complementary actions a and $\bar{a}$.

In order to obtain the desired negative results, it will be sufficient to consider the above language with three unary prefixing operators; so there is only one action a with its corresponding complementary action $\bar{a}$. Our results carry over unchanged to a setting with an arbitrary number of actions, and corresponding unary prefixing operators. Henceforth, we let $\mu \in \{a, \bar{a}, \tau\}$ and $\alpha \in \{a, \bar{a}\}$. As usual, we postulate that $\bar{\bar{a}} = a$. We shall use the meta-variables t, u, v, w to range over process terms, and write $var(t)$ for the collection of variables occurring in the term t. The *size* of a term is the number of operator symbols in it. A process term is *closed* if it does not contain any variables. Closed terms, or *processes*, will be typically denoted by p, q, r. Moreover, trailing $\mathbf{0}$'s will often be omitted from terms.

A *(closed) substitution* is a mapping from process variables to (closed) CCS$_f$ terms. For every term t and substitution σ, the term obtained by replacing every occurrence of a variable x in t with the term $\sigma(x)$ will be written $\sigma(t)$. Note that $\sigma(t)$ is closed, if so is σ. We shall sometimes write $\sigma[x \mapsto p]$ to denote the substitution that maps the variable x into process p and behaves like σ on all other variables.

In the remainder of this paper, we exploit the associativity and commutativity of $+$ with respect to bisimilarity and we consider process terms modulo them, namely we do not distinguish $t + u$ and $u + t$, nor $(t + u) + v$ and $t + (u + v)$. In what follows, the symbol $=$ will denote equality modulo the above identifications. We use a *summation* $\sum_{i \in \{1,\ldots,k\}} t_i$ to denote the term $t = t_1 + \cdots + t_k$, where the empty sum represents $\mathbf{0}$. We can also assume that the terms t_i, for $i \in \{1, \ldots, k\}$, do not have $+$ as head operator, and refer to them as the *summands* of t.

Henceforth, for each action μ and $m \geq 0$, we let μ^0 denote $\mathbf{0}$ and μ^{m+1} denote $\mu(\mu^m)$. For each action μ and positive integer $i \geq 0$, we also define

$$\mu^{\leq i} = \mu + \mu^2 + \cdots + \mu^i \ .$$

2.3 Equational Logic

An *axiom system* $\mathcal{E}$ is a collection of (*process*) *equations* $t \approx u$ over CCS_f. An equation $t \approx u$ is *derivable* from an axiom system $\mathcal{E}$, notation $\mathcal{E} \vdash t \approx u$, if there is an *equational proof* for it from $\mathcal{E}$, namely if $t \approx u$ can be inferred from the axioms in $\mathcal{E}$ using the *rules* of *equational logic*, which are reflexivity, symmetry, transitivity, substitution and closure under CCS_f contexts. We refer the interested reader to Appendix A for a complete presentation of such rules.

We are interested in equations that are valid modulo some congruence relation $\mathcal{R}$ over closed terms. The equation $t \approx u$ is said to be *sound* modulo $\mathcal{R}$ if $\sigma(t)\,\mathcal{R}\,\sigma(u)$ for all closed substitutions σ. For simplicity, if $t \approx u$ is sound, then we write $t\,\mathcal{R}\,u$. An axiom system is *sound* modulo $\mathcal{R}$ if, and only if, all of its equations are sound modulo $\mathcal{R}$. Conversely, we say that $\mathcal{E}$ is *ground-complete* modulo $\mathcal{R}$ if $p\,\mathcal{R}\,q$ implies $\mathcal{E} \vdash p \approx q$ for all closed terms p, q. We say that $\mathcal{R}$ has a *finite*, ground-complete, axiomatisation, if there is a *finite* axiom system $\mathcal{E}$ that is sound and ground-complete for $\mathcal{R}$.

3 The simplifying assumptions

The aim of this paper is to investigate whether bisimilarity admits a finite equational axiomatisation over CCS_f, for some binary operator f. Of course, this question only makes sense if f is an operator that preserves bisimilarity. In this section we discuss two assumptions we shall make on the auxiliary operator f in order to meet such requirement and to tackle problem (P) in a simplified technical setting.

3.1 The de Simone format

One way to guarantee that f preserves bisimilarity is to postulate that the behaviour of f is described using Plotkin-style rules that fit a rule format that is known to preserve bisimilarity, see, e.g., [8] for a survey of such rule formats. The simplest format satisfying this criterion is the format proposed by de Simone in [14]. We believe that if we can't deal with operations specified in that format, then there is little hope to generalise our results. Therefore, we make the following

Assumption 1. The behaviour of f is described by rules in de Simone format.

Definition 3.1. An SOS rule ρ for f is in *de Simone format* if it has the form

$$\rho = \frac{\{x_i \xrightarrow{\mu_i} y_i \mid i \in I\}}{f(x_1, x_2) \xrightarrow{\mu} t} \qquad (1)$$

where

- $I \subseteq \{1, 2\}$,
- the variables x_1, x_2 and y_i $(i \in I)$ are all different (and are called the *variables of the rule*),
- μ and μ_i $(i \in I)$ are contained in $\{a, \bar{a}, \tau\}$, and
- t (called the *target of the rule*) is a CCS_f term over variables $\{x_1, x_2, y_i \mid i \in I\}$ such that
  – each variable occurs at most once in t, and
  – if $i \in I$, then x_i does not occur in t.

Henceforth, we shall assume, without loss of generality, that the variables x_1, x_2, y_1 and y_2 are the only ones used in operational rules. Moreover, if μ is the label of the transition in the conclusion of a de Simone rule ρ, we shall say that ρ has μ as *label*.

The SOS rules for all of the classic CCS operators, reported below, are in de Simone format, and so are those for Hennessy's $/\!\!/$ operator from [16] and for Bergstra and Klop's left and communication merge operators [10], at least if we disregard issues related to the treatment of successful termination. Thus restricting ourselves to operators whose operational behaviour is described by de Simone rules leaves us with a good degree of generality.

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

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

The transition rules for the classic CCS operators above and those for the operator f give rise to transitions between CCS_f terms. The operational semantics for CCS_f is thus given by the LTS whose states are CCS_f terms, and whose transitions are those that are provable using the rules.

In what follows, we shall consider the collection of *closed CCS_f terms* modulo bisimilarity. Since the SOS rules defining the operational semantics of CCS_f are in de Simone's format, we have that bisimilarity is a congruence with respect to CCS_f operators, that is, $\mu p \leftrightarrow \mu q$, $p + p' \leftrightarrow q + q'$, $p \| p' \leftrightarrow q \| q'$ and $f(p, p') \leftrightarrow f(q, q')$ hold whenever $p \leftrightarrow q$, $p' \leftrightarrow q'$ and p, p', q, q' are closed CCS_f terms.

Bisimilarity is extended to arbitrary CCS_f terms thus:

Definition 3.2. Let t, u be CCS_f terms. Then $t \leftrightarrow u$ if and only if $\sigma(t) \leftrightarrow \sigma(u)$ for every closed substitution σ.

3.2 Axiomatising $\|$ with f

Our second simplifying assumption concerns how the operator f can be used to axiomatise parallel composition. To this end, a fairly natural assumption on an axiom system over CCS_f is that it includes an equation of the form

$$x \| y \approx t(x, y) \qquad (2)$$

where t is a CCS_f term that does not contain occurrences of $\|$ with $var(t) \subseteq \{x, y\}$. More precisely, the term will be in the general form $t(x, y) = \sum_{i \in I} t_i(x, y)$, where I is a finite index set and, for each $i \in I$, $t_i(x, y)$ does not have $+$ as head operator.

We now proceed to refine the form of the term $t(x, y)$, in order to guarantee the soundness, modulo bisimilarity, of Equation (2). Intuitively, no term $t_i(x, y)$ can have prefixing as head operator. In fact, if $t(x, y)$ had a summand

$\mu.t'(x,y)$, for some $\mu \in \{a, \bar{a}, \tau\}$, then one could easily show that $\mathbf{0}\|\mathbf{0} \not\!\underline{\leftrightarrow}\, t(\mathbf{0},\mathbf{0})$, since $t(\mathbf{0},\mathbf{0})$ could perform a μ-transition, unlike $\mathbf{0}\|\mathbf{0}$. Similarly, $t(x,y)$ cannot have a variable as a summand, for otherwise we would have $a\|\tau \not\!\underline{\leftrightarrow}\, t(a,\tau)$. Indeed, assume, without loss of generality, that $t(x,y)$ has a summand x. Then, $t(a,\tau) \xrightarrow{a} \mathbf{0}$, yielding that $t(a,\tau)$ has norm 1, whereas $a\|\tau$ has norm 2. We can therefore assume that, for each $i \in I$,

$$t_i(x,y) = f(t_i^1(x,y), t_i^2(x,y))$$

for some CCS$_f$ terms $t_i^j(x,y)$, with $j \in \{1,2\}$. To further narrow down the options on the form that the subterms $t_i^j(x,y)$ might have, we would need to make some assumptions on the behaviour of the operator f. For the sake of generality, we assume that the terms $t_i^j(x,y)$ are in the simplest form, namely they are variables in $\{x,y\}$. Such an assumption is reasonable because to allow prefixing and/or nested occurrences of f-terms in the scope of the terms $t_i(x,y)$ we would need to define (at least partially) the operational semantics of f, thus making our results less general as, roughly speaking, we would need to study one possible auxiliary operator at a time (the one identified by the considered set of de Simone rules). Moreover, if we look at how parallel composition is expressed equationally as a derived operator in terms of Hennessy's merge or Bergstra and Klop's left and communication merge or as in reference [2], viz. via the equations

$$\begin{aligned} x \parallel y &\approx (x \,|\!\!/\, y) + (y \,|\!\!/\, x) \\ x \parallel y &\approx (x \lfloor\!\lfloor y) + (y \lfloor\!\lfloor x) + (x \mid y) \\ x \parallel y &\approx (x \lfloor\!\lfloor y) + (x \rfloor\!\rfloor y) + (x \mid y) \ , \end{aligned}$$

we see the emergence of a pattern: the parallel composition operator is always expressed in terms of sums of terms built from the auxiliary operators and variables.

Therefore, from now on we'll make the following:

Assumption 2. For some $J \subseteq \{x,y\}^2$, the equation

$$x \parallel y \approx \sum \{ f(z_1,z_2) \mid (z_1,z_2) \in J \} \qquad (3)$$

holds modulo bisimilarity. We shall sometimes use t_J to denote the right-hand side of the above equation and use $t_J(p,q)$ to stand for the process $\sigma[x \mapsto p, y \mapsto q](t_J)$.

Using our assumptions, we now proceed to further investigate the relation between operator f and parallel composition, obtaining a refined form for Equation (3) (Proposition 3.5 below).

Lemma 3.3. *Assume that Assumptions 1 and 2 hold. Then:*

1. *The index set J on the right-hand side of (3) is non-empty.*
2. *The set of transition rules for f is non-empty.*
3. *Each transition rule for f has some premise.*
4. *The terms $f(x,x)$ and $f(y,y)$ are not summands of t_J, the right-hand side of Equation (3).*

Proof: The proof can be found in Appendix B.1. □

As consequence, we may infer that the index set J in the term on the right-hand side of Equation (3) is either one of the singletons $\{(x,y)\}$ or $\{(y,x)\}$, or it is the set $\{(x,y),(y,x)\}$. Due to Moller's results to the effect that bisimilarity has no finite ground-complete axiomatisation over CCS [23, 25], the former option can be discarded, as shown in the following:

Proposition 3.4. *If J is a singleton, then CCS$_f$ admits no finite equational axiomatisation modulo bisimilarity.*

Proof: The proof can be found in Appendix B.2. □

As a consequence, we can restate our Assumption 2 in the following simplified form:

Proposition 3.5. *Equation* (3) *can be refined to the form:*

$$x \parallel y \approx f(x,y) + f(y,x) \ . \qquad (4)$$

Moreover, in the light of Moller's results in [23, 25], we can restrict ourselves to considering only operators f such that $x \parallel y \approx f(x,y)$ does not hold modulo bisimilarity.

For later use, we note a useful consequence of the soundness of Equation (4) modulo bisimilarity.

Lemma 3.6. *Assume that Equation (4) holds modulo $\underline{\leftrightarrow}$. Then depth$(p)$ is finite for each closed CCS$_f$ term p.*

Proof: The proof can be found in Appendix B.3. □

We can therefore refine the Assumption 2 to its version given in Equation (4).

4 The operational semantics of f

In order to prove Theorem 4.7, we shall, first of all, understand what rules f may and must have in order for Equation (4) to hold modulo bisimilarity (Proposition 4.4 below).

We begin this analysis by restricting the possible forms the SOS rules for the operator f *may* take.

Lemma 4.1. *Suppose that f meets Assumption 1, and that Equation* (4) *is sound modulo bisimilarity. Let ρ be a de Simone rule for f with μ as label. Then the following statements hold.*

1. *If $\mu = \tau$ then the set of premises $\{x_i \xrightarrow{\mu_i} y_i \mid i \in I\}$ of ρ can only have one of the following possible forms:*
   - *$\{x_i \xrightarrow{\tau} y_i\}$ for some $i \in \{1,2\}$, or*
   - *$\{x_1 \xrightarrow{\alpha} y_1, x_2 \xrightarrow{\bar{\alpha}} y_2\}$ for some $\alpha \in \{a, \bar{a}\}$.*
2. *If $\mu = \alpha$ for some $\alpha \in \{a, \bar{a}\}$, then the set of premises $\{x_i \xrightarrow{\mu_i} y_i \mid i \in I\}$ can only have the form $\{x_i \xrightarrow{\alpha} y_i\}$ for some $i \in \{1,2\}$.*

Proof: The proof can be found in Appendix C.1. □

The previous lemma limits the form of the premises that rules for f *may* have in order for Equation (4) to hold modulo bisimilarity. We now characterise the rules that f *must* have in order for it to satisfy that equation.

Firstly, we deal with *synchronisation*.

Lemma 4.2. *Assume that Equation (4) holds modulo bisimilarity. Then the operator f must have a rule of the form*

$$\frac{x_1 \xrightarrow{\alpha} y_1 \quad x_2 \xrightarrow{\bar{\alpha}} y_2}{f(x_1,x_2) \xrightarrow{\tau} t(y_1,y_2)} \tag{5}$$

for some $\alpha \in \{a, \bar{a}\}$ and term t. Moreover, for each rule for f of the above form the term $t(x,y)$ is bisimilar to $x \parallel y$.

Proof: The proof can be found in Appendix C.2. □

Henceforth we assume, without loss of generality that the target of a rule of the form (5) is $y_1 \| y_2$.

We introduce the unary predicates $S^f_{a,\bar{a}}$ and $S^f_{\bar{a},a}$ to identify which rules of type (5) are available for f. In detail, $S^f_{a,\bar{a}}$ holds if f has a rule of type (5) with premises $x_1 \xrightarrow{a} y_1$ and $x_2 \xrightarrow{\bar{a}} y_2$. $S^f_{\bar{a},a}$ holds in the symmetric case.

We consider now the *interleaving* behaviour in the rules for f. In order to properly characterise the rules for f as done in the previous Lemma 4.2, we consider an additional simplifying assumption on the form that the targets of the rules for f might have.

Assumption 3. If t is the target of a rule for f, then t is either a variable or a term obtained by applying a single CCS_f operator to the variables of the rule, according to the constraints of the de Simone format.

Remark 1. *Notice that Assumption 3 does not contradict Lemma 4.2. Assumption 3 is only used in the proof of the second claim in the following Lemma 4.3. We have been unable to prove the claim without it. We do not know whether that assumption is needed, but we conjecture that it is not.*

Lemma 4.3. *Let $\mu \in \{a, \bar{a}, \tau\}$. Then the operator f must have a rule of the form*

$$\frac{x_1 \xrightarrow{\mu} y_1}{f(x_1,x_2) \xrightarrow{\mu} t(y_1,x_2)} \tag{6}$$

or a rule of the form

$$\frac{x_2 \xrightarrow{\mu} y_2}{f(x_1,x_2) \xrightarrow{\mu} t(x_1,y_2)} \tag{7}$$

for some term t. Moreover, under Assumption 3, for each rule for f of the above forms the term $t(x,y)$ is bisimilar to $x \parallel y$.

Proof: The proof can be found in Appendix C.3. □

Henceforth we assume, without loss of generality, that the target of a rule of the form (6) is $y_1 \| x_2$ and the target of a rule of the form (7) is $x_1 \| y_2$.

For each $\mu \in \{a, \bar{a}, \tau\}$, we introduce two unary predicates, L^f_μ and R^f_μ, that allow us to identify which rules with label μ are available for f. In detail,

- L^f_μ holds if f has a rule of the form (6) with label μ;
- R^f_μ holds if f has a rule of the form (7) with label μ.

We write $L^f_\mu \wedge R^f_\mu$ to denote that f has both a rule of the form (6) and one of the form (7) with label μ. We stress that, for each action μ, the validity of predicate L^f_μ does not prevent R^f_μ from holding, and vice versa. Throughout the paper, in case *only one* of the two predicates holds, we will clearly state it.

Summing up, we have obtained that:

Proposition 4.4. *If f meets Assumptions 1 and 3 and Equation (4) is sound modulo bisimilarity, then f must satisfy $S^f_{\alpha,\bar{\alpha}}$ for at least one $\alpha \in \{a, \bar{a}\}$, and, for each $\mu \in \{a, \bar{a}, \mu\}$, at least one of L^f_μ and R^f_μ.*

Proof: Immediate from Lemmas 4.2 and 4.3. □

The following proposition states that this is enough to obtain the soundness of Equation (4).

Proposition 4.5. *Assume that all of the rules for f have the form (5), (6), or (7). If $S^f_{\alpha,\bar{\alpha}}$ holds for at least one $\alpha \in \{a, \bar{a}\}$, and, for each $\mu \in \{a, \bar{a}, \tau\}$, at least one of L^f_μ and R^f_μ holds, then Equation (4) is sound modulo bisimilarity.*

Proof: The proof can be found in Appendix C.4. □

As an immediate consequence of the form of the rules for f given in Proposition 4.5, we have the following lemma:

Lemma 4.6. *Assume that all of the rules for f have the form (5), (6), or (7). Then each closed term p in CCS_f is finitely branching, that is, the set $\{(\mu, q) \mid p \xrightarrow{\mu} q\}$ is finite.*

Remark 2. *A standard consequence of the finiteness of the depth (Lemma 3.6) and the finite branching of closed terms in CCS_f is that each closed CCS_f term is bisimilar to a synchronisation tree [20], that is, a closed term built only using the constant* **0**, *the unary prefixing operations and the binary + operation. Since bisimilarity is a congruence over CCS_f, this means, in particular, that an equation $t \approx u$ over CCS_f is sound modulo bisimilarity if, and only if, the closed terms $\sigma(t)$ and $\sigma(u)$ are bisimilar for each substitution mapping variables to synchronisation trees. Moreover, we can use the sub-language of synchronisation trees, which is common to all of the languages CCS_f, to compare terms from these languages for different choices of binary operation f with respect to bisimilarity.*

We devote the remainder of this paper to prove the following result:

Theorem 4.7. *Assume that f satisfies Assumptions 1 and 3, and that Equation* (4) *holds modulo bisimilarity. Then bisimilarity admits no finite equational axiomatisation over CCS_f.*

5 The proof strategy

In this section, we discuss the general reasoning behind the proof of Theorem 4.7 and present the proof method we use to obtain the desired negative result (Theorem 5.1 below).

In light of Propositions 4.4 and 4.5, to prove Theorem 4.7 we will proceed by a case analysis over the possible sets of allowed SOS rules for operator f. In each case, our proof method will follow the same general schema, which has its roots in Moller's arguments to the effect that bisimilarity is not finitely based over CCS (see, e.g., [4, 23–25]), and that we present here at an informal level.

The main idea is to identify a *witness property of the negative result*. This is a specific property of terms associated with each finite, sound axiom system $\mathcal{E}$ over CCS_f. Typically, it only depends on some feature of the axiom system $\mathcal{E}$ that can characterised by a natural number $n \geq 0$, such as for instance the maximum size of the equations in $\mathcal{E}$. Hence, we denote the witness property by $W_n(\mathcal{E})$, for $n \geq 0$. Notably, when n is *large enough*, we require $W_n(\mathcal{E})$ to be an invariant that is preserved by $\mathcal{E}$-derivations, that is, whenever an equation $p \approx q$ can be derived from $\mathcal{E}$ then either both p and q satisfy $W_n(\mathcal{E})$, or none of them does. Then, we exhibit an infinite family of valid equations, say e_n, called accordingly *witness family of equations for the negative result*, in which $W_n(\mathcal{E})$ is not preserved, namely it is satisfied only by one side of each equation. Thus, our order of business will be to prove the following result:

Theorem 5.1. *Suppose that Assumptions 1–3 are met. Let $\mathcal{E}$ be a finite axiom system over CCS_f that is sound with respect to bisimilarity. Then there is an infinite family e_n, $n \geq 0$, of sound equations such that $\mathcal{E}$ does not prove the equation e_n, for each n that is larger than the size of each term in the equations in $\mathcal{E}$.*

In this paper, the property $W_n(\mathcal{E})$ corresponds to having a summand that is bisimilar to a specific process. In detail:

1. We identify, for each case, a family of processes $f(\mu, p_n)$, where $n \in \mathbb{N}$, and the choice of μ and p_n is tailored to the particular set of SOS rules allowed for f. Moreover, process p_n will have size at least n, for each $n \geq 0$. Sometimes, we shall refer to the processes $f(\mu, p_n)$ as the *witness processes*.
2. We prove that by choosing *n large enough*, given a finite set of valid equations $\mathcal{E}$ and processes $p, q \underline{\leftrightarrow} f(\mu, p_n)$, if $\mathcal{E} \vdash p \approx q$ and p has a summand bisimilar to $f(\mu, p_n)$, then also q has a summand bisimilar to $f(\mu, p_n)$. Informally, we will choose n greater than the size of all the equations in $\mathcal{E}$, so that we are guaranteed that the behaviour of the summand bisimilar to $f(\mu, p_n)$ is due to a closed substitution instance of a variable.
3. We provide an infinite family of valid equations e_n in which one side has a summand bisimilar to $f(\mu, p_n)$, but the other side does not. In light of item 2, this implies that such a family of equations cannot be derived from any finite collection of valid equations over CCS_f, modulo bisimilarity, thus proving Theorem 5.1.

To narrow down the combinatorial analysis over the allowed sets of SOS rules for f we examine first the *distributivity properties*, modulo bisimilarity, of the operator f with respect to summation.

First of all, we notice that f cannot be distributive with respect to summation in both arguments. This is a consequence of our previous analysis of the operational rules that such an operator f may and must have in order for Equation (4) to hold. However, it can also be shown in a purely algebraic manner as we do in Appendix D.1.

Lemma 5.2. *A binary operator satisfying Equation (4) cannot be distributive with respect to + in both arguments.*

Hence, we can limit ourselves to considering binary operators satisfying our constraints that, modulo bisimilarity, are distributive with respect to + in one argument or in none.

We consider these two possibilities in turn.

5.1 Distributivity in one argument

Due to our Assumptions 1–3, we can exploit a result from [2] to characterise the rules for an operator f that distributes over summation in one of its arguments. More specifically, [2, Lemma 4.3] gives a condition on the rules for a *smooth operator* g in a GSOS system that includes the + operator in its signature, which guarantees that g distributes over summation in one of its arguments. (The rules defining the semantics of smooth operators are a generalisation of those in de Simone format.) Here we show that, for operator f, the condition in [2, Lemma 4.3] is both necessary and sufficient for distributivity of f in one of its two arguments.

Lemma 5.3. *Let $i \in \{1, 2\}$. Modulo bisimilarity, operator f distributes over summation in its i-th argument if and only if each rule for f has a premise $x_i \xrightarrow{\mu_i} y_i$, for some μ_i.*

Proof: The proof can be found in Appendix D.2. □

By Proposition 4.4, Lemma 5.3 implies that, when f is distributive in one argument, either L^f_μ holds for all $\mu \in \{a, \bar{a}, \tau\}$ or R^f_μ holds for all $\mu \in \{a, \bar{a}, \tau\}$, and $S_{\alpha,\bar{\alpha}}$ holds for at least one $\alpha \in \{a, \bar{a}\}$. Notice that if L^f_μ holds for each action μ and both $S^f_{a,\bar{a}}$ and $S^f_{\bar{a},a}$ hold, then f behaves as Hennessy's merge $|\!\!/$ [16], and our Theorem 4.7 specializes to [4, Theorem 18]. Hence we assume, without loss of generality, that $S^f_{\alpha,\bar{\alpha}}$ holds for only one $\alpha \in \{a, \bar{a}\}$. A similar reasoning applies if R^f_μ holds for each action μ.

In Section 8 we will present the proof of Theorem 4.7 in the case of an operator f that distributes over summation in its first argument (see Theorem 8.1).

5.2 Distributivity in neither argument

We now consider the case in which f does not distribute with respect to summation in either argument.

Also in this case, we can exploit Lemma 5.3 to obtain a characterisation of the set of rules allowed for an operator f

satisfying the desired constraints. In detail, we infer that there must be $\mu, \nu \in \{a, \bar{a}, \tau\}$, not necessarily distinct, such that L^f_μ and R^f_ν hold. Otherwise, as f must have at least one rule for each action (see Proposition 4.4), at least one argument would be involved in the premises of each rule, and this would entail distributivity with respect to summation in that argument.

We will split the proof of Theorem 4.7 for an operator f that, modulo bisimilarity, does not distribute over summation in either argument into three main cases:

1. In Section 9, we consider the case of $L^f_\alpha \wedge R^f_\alpha$ holding, for some $\alpha \in \{a, \bar{a}\}$ (Theorem 9.1).
2. In Section 10, we deal with the case of f having only one rule for α, only one rule for $\bar{\alpha}$, and such rules are of different forms. As we will see, we will need to distinguish two subcases, according to which predicate $S^f_{\alpha,\bar{\alpha}}$ holds (Theorem 10.1 and Theorem 10.2).
3. Finally, in Section 11, we study the case of f having only one rule with label α, only one rule with label $\bar{\alpha}$, and such rules are of the same type (Theorem 11.1).

Before proceeding to the proofs of the various cases, we dedicate Section 6 and Section 7 to the introduction of some general preliminary results and observations, concerning, respectively, the *equational theory of* CCS_f and the analysis of the *initial behaviour* of closed instantiations of terms, which will play a key role in our proofs.

6 The equational theory of CCS_f

In this section we study some aspects of the equational theory of CCS_f modulo bisimilarity that are useful in the proofs of our negative results. In particular, we show that, due to Equation (4), proving the negative result over CCS_f is equivalent to proving it over its reduct CCS^-_f, whose signature does not contain occurrences of $\|$ (Proposition 6.2 below).

Furthermore, we discuss the relation between the available rules for f and the bisimilarity of terms of the form $f(p, q)$ with $\mathbf{0}$. As we will see, in the case of an operator f that distributes with respect to summation in one argument, it is possible to *saturate* the axiom systems [23] yielding a simplification in the proofs (Proposition 6.5 below). On the other hand, we cannot rely on saturation for an operator f that distributes with respect to $+$ in neither of its arguments.

6.1 Simplifying equational proofs

We show that it is sufficient to prove that bisimilarity admits no finite equational axiomatisation over CCS^-_f, consisting of the CCS_f terms that do not contain occurrences of $\|$.

Definition 6.1. For each CCS_f term t, we define $\hat{t}$ as follows:

$$\begin{array}{ll} \hat{\mathbf{0}} = \mathbf{0} & \widehat{t+u} = \hat{t} + \hat{u} \\ \hat{x} = x & \widehat{f(t,u)} = f(\hat{t}, \hat{u}) \\ \widehat{\mu t} = \mu \hat{t} & \widehat{t \| u} = f(\hat{t}, \hat{u}) + f(\hat{u}, \hat{t}) \ . \end{array}$$

Then, for any axiom system $\mathcal{E}$ over CCS_f, we let $\widehat{\mathcal{E}} = \{\hat{t} \approx \hat{u} \mid (t \approx u) \in \mathcal{E}\}$.

We notice that, for each CCS_f term t, the term $\hat{t}$ is in CCS^-_f. Moreover, if t contains no occurrences of the parallel composition operator, then $\hat{t} = t$. Since Equation (4) is sound with respect to bisimilarity, which is a congruence relation, it is not hard to show that each term t in CCS_f is bisimilar to $\hat{t}$. Therefore if $\mathcal{E}$ is an axiom system over CCS_f that is sound with respect to bisimilarity, then $\widehat{\mathcal{E}}$ is an axiom system over CCS^-_f that is sound with respect to bisimilarity.

The following result states the reduction of the non-finite axiomatisability of $\leftrightarroweq$ over CCS_f to that of $\leftrightarroweq$ over CCS^-_f.

Proposition 6.2. *Let $\mathcal{E}$ be an axiom system over CCS_f. Then:*

1. *If $\mathcal{E} \vdash t \approx u$, then $\widehat{\mathcal{E}} \vdash \hat{t} \approx \hat{u}$.*
2. *If $\mathcal{E}$ is a complete axiomatisation of $\leftrightarroweq$ over CCS_f, then $\widehat{\mathcal{E}}$ completely axiomatises $\leftrightarroweq$ over CCS^-_f.*
3. *If bisimilarity is not finitely axiomatisable over CCS^-_f, then it is not finitely axiomatisable over CCS_f either.*

Proof: The proof can be found in Appendix E.1. □

In light of this result, henceforth we shall focus on proving that $\leftrightarroweq$ affords no finite equational axiomatisation over CCS^-_f.

6.2 Bisimilarity with 0

As a further simplification, we can focus on the $\mathbf{0}$ *absorption properties* of CCS^-_f operators. Informally, we can restrict the axiom system to a collection of equations that do not introduce unnecessary terms that are bisimilar to $\mathbf{0}$ in the equational proofs, namely $\mathbf{0}$ summands and $\mathbf{0}$ *factors*.

Definition 6.3. We say that a CCS^-_f term t has a $\mathbf{0}$ *factor* if it contains a subterm of the form $f(t', t'')$, where either t' or t'' is bisimilar to $\mathbf{0}$.

The $\mathbf{0}$ absorption properties of f depend crucially on the allowed set of SOS rules for f. Notably, we have different results, according to the distributivity properties of f.

We examine first the case of an operator f that, modulo bisimilarity, distributes over summation in its first argument.

In this case, an example of a collection of equations over CCS^-_f that are sound with respect to $\leftrightarroweq$ is given by axioms A0–A3, F0–F1:

$$\begin{array}{llll} \mathrm{A0} & x + \mathbf{0} \approx x & \mathrm{F0} & f(\mathbf{0}, x) \approx \mathbf{0} \\ \mathrm{A1} & x + y \approx y + x & \mathrm{F1} & f(x, \mathbf{0}) \approx x \ . \\ \mathrm{A2} & (x + y) + z \approx x + (y + z) & & \\ \mathrm{A3} & x + x \approx x & & \end{array}$$

Axioms A0 and F0 are enough to establish that each CCS^-_f term that is bisimilar to $\mathbf{0}$ is also provably equal to $\mathbf{0}$.

Lemma 6.4. *Let t be a CCS^-_f term. Then $t \leftrightarroweq \mathbf{0}$ if, and only if, the equation $t \approx \mathbf{0}$ is provable using axioms A0 and F0 from left to right.*

Proof: The proof is sketched in Appendix E.2. □

In light of the above result, in the technical developments to follow, when dealing with an operator f that distributes over $+$ in its first argument we shall assume, without loss of generality, that each axiom system we consider includes the equations A0–A3, F0–F1. This assumption means, in particular, that our axiom systems will allow us to identify each term that is bisimilar to $\mathbf{0}$ with $\mathbf{0}$.

It is well-known (see, e.g., Sect. 2 in [15]) that if an equation relating two closed terms can be proved from an axiom system $\mathcal{E}$, then there is a closed proof for it. In addition, if $\mathcal{E}$ satisfies a further closure property, called *saturation*[1], in addition to those mentioned earlier, and that closed equation relates two terms containing no occurrences of $\mathbf{0}$ as a summand or factor, then there is a closed proof for it in which all of the terms have no occurrences of $\mathbf{0}$ as a summand or factor, as formalised in the following proposition.

Proposition 6.5. *Assume that $\mathcal{E}$ is a saturated axiom system. Then, the proof from $\mathcal{E}$ of an equation $p \approx q$, where p and q are terms not containing occurrences of $\mathbf{0}$ as a summand or factor, need not use terms containing occurrences of $\mathbf{0}$ as a summand or factor.*

Since the proof of this result follows the same lines of that of [23, Proposition 5.1.5], we omit it from the main text. We refer the interested reader to Appendix E.3 for the details.

In light of Proposition 6.5, henceforth, when dealing with an operator f that distributes with respect to $+$ in one of its arguments, we shall limit ourselves to considering saturated axiom systems.

For what concerns an operator f that does not distribute over $+$ in either of its arguments, we conjecture that *the valid equations, modulo bisimilarity, of the form $t \approx \mathbf{0}$ cannot be proved by means of any finite, sound set of axioms*. This would imply that, in the case at hand, we cannot assume that we can use saturation to simplify finite axiom systems.

Nevertheless, this gives us the opportunity to prove an even stronger result: we can, in fact, show that *even if we were able to provide a finite axiom system $\mathcal{E}$ allowing us to infer that a term that is bisimilar to $\mathbf{0}$ is also provably equal to $\mathbf{0}$, then $\mathcal{E}$ could not be finitely extended in order to obtain a finite axiomatisation of bisimilarity over CCS^-_f*. (We refer the interested reader to Appendix E.4 for more details).

7 Decomposing the semantics of terms

As outlined in Section 5, to obtain the desired negative results we will proceed by a case analysis on the operational rules for operator f. However, there are a few preliminary results that hold for all cases and that will be useful in the upcoming proofs. We dedicate this section to presenting these results and some auxiliary notions.

In the proofs to follow, we shall sometimes need to establish a correspondence between the behaviour of open terms and the semantics of their closed instances, with a special focus on the role of variables. In detail, we need to consider the possible origins of a transition of the form $\sigma(t) \xrightarrow{\alpha} p$, for some action $\alpha \in \{a, \bar{a}\}$, closed substitution σ, CCS^-_f term t and closed term p. In fact, the equational theory is defined over process terms, whereas the semantic properties can be verified only on their closed instances.

Lemma 7.1. *Let $\mu \in \{a, \bar{a}, \tau\}$. Then for all t, t' and substitutions σ it holds that if $t \xrightarrow{\mu} t'$ then $\sigma(t) \xrightarrow{\mu} \sigma(t')$.*

However, a transition $\sigma(t) \xrightarrow{\mu} p$ may also derive from the initial behaviour of some closed term $\sigma(x)$, provided that the collection of initial moves of $\sigma(t)$ depends, in some formal sense, on that of the closed term substituted for the variable x. Roughly speaking, our aim is now to provide the conditions under which $\sigma(t) \xrightarrow{\mu} p$ can be inferred from $\sigma(x) \xrightarrow{\nu} q$, for some $\mu, \nu \in \{a, \bar{a}, \tau\}$ and processes p, q. As one might expect, in our setting the provability of transitions needs to be parametric with respect to the rules for f.

Example 7.2. Consider the CCS^-_f term $t = f(x, \tau)$. Firstly, we notice that if R^f_τ holds then we can infer that $\sigma(t) \xrightarrow{\tau} \sigma(x) \| \mathbf{0}$ for all closed substitutions σ. Assume now that $\sigma(x) = a$. Clearly, we can derive $\sigma(t) \xrightarrow{a} \mathbf{0} \| \tau$ only if L^f_a holds.

To fully describe this situation, for each $\mu \in \{a, \bar{a}, \tau\}$, we introduce the auxiliary transition relation $\rightarrow_\mu$ over open terms. To this end, we present the notion of *configuration* over CCS^-_f terms, which stems from [7]. Configurations are terms defined over a set of variables $\mathcal{V}_\mathrm{d} = \{x_d \mid x \in \mathcal{V}\}$, disjoint from $\mathcal{V}$, and CCS^-_f terms. Intuitively, the symbol x_d (read "during x") will be used to denote that the closed term substituted for an occurrence of variable x has begun its execution.

Definition 7.3. The collection of *CCS^-_f configurations* is given by the following grammar:

$$c ::= t \mid x_d \mid c \| t \mid t \| c \ ,$$

where t is a CCS^-_f term, and $x_d \in \mathcal{V}_d$.

For example, the configuration $x_d \| f(a, x)$ is meant to describe a state of the computation of some term in which the (closed term substituted for the) occurrence of variable x on the left-hand side of the $\|$ operator has begun its execution, but the one on the right-hand side has not.

We introduce also special labels for the auxiliary transitions $\rightarrow_\mu$, to keep track of which rules for f are available, and thus which one triggered the move by the closed instance of x. In detail, we let x_l denote that the closed instance of x is responsible for the transition when L^f_μ holds. In case R^f_μ holds, we use x_r. Finally, x_b is used when $L^f_\mu \wedge R^f_\mu$ holds.

[1] The formal definition of the notion of saturated axiom system is not necessary to follow the upcoming technical developments and may be found in the appendices as Definition E.4.

The auxiliary transitions of the form $\longrightarrow_\mu$ are then formally defined via the inference rules below

$$(a_1)\ \frac{L^f_\mu}{x \xrightarrow{x_\mathrm{l}}_\mu x_\mathrm{d}} \qquad (a_2)\ \frac{R^f_\mu}{x \xrightarrow{x_\mathrm{r}}_\mu x_\mathrm{d}} \qquad (a_3)\ \frac{L^f_\mu \wedge R^f_\mu}{x \xrightarrow{x_\mathrm{b}}_\mu x_\mathrm{d}}$$

$$(a_4)\ \frac{t_1 \xrightarrow{x_\mathrm{w}}_\mu c}{t_1 + t_2 \xrightarrow{x_\mathrm{w}}_\mu c}\ \mathrm{w} \in \{\mathrm{l},\mathrm{r},\mathrm{b}\} \qquad (a_5)\ \frac{t_2 \xrightarrow{x_\mathrm{w}}_\mu c}{t_1 + t_2 \xrightarrow{x_\mathrm{w}}_\mu c}\ \mathrm{w} \in \{\mathrm{l},\mathrm{r},\mathrm{b}\}$$

$$(a_6)\ \frac{t_1 \xrightarrow{x_\mathrm{l}}_\mu c}{f(t_1,t_2) \xrightarrow{x_\mathrm{l}}_\mu c\|t_2} \qquad (a_7)\ \frac{t_2 \xrightarrow{x_\mathrm{r}}_\mu c}{f(t_1,t_2) \xrightarrow{x_\mathrm{r}}_\mu t_1\|c}$$

$$(a_8)\ \frac{t_1 \xrightarrow{x_\mathrm{b}}_\mu c}{f(t_1,t_2) \xrightarrow{x_\mathrm{b}}_\mu c\|t_2} \qquad (a_9)\ \frac{t_2 \xrightarrow{x_\mathrm{b}}_\mu c}{f(t_1,t_2) \xrightarrow{x_\mathrm{b}}_\mu t_1\|c}$$

Example 7.4. Consider the term $t = f(x,\tau)$ from Example 7.2. Assume, for instance, that L^f_a holds, yielding the transition $x \xrightarrow{x_\mathrm{l}}_a x_\mathrm{d}$, due to rule (a_1). Then, an application of rule (a_6) would give $f(x,\tau) \xrightarrow{x_\mathrm{l}}_a x_\mathrm{d}\|\tau$ with the following meaning: since the rules for f allow a-moves of the first argument to yield a-moves of terms of the form $f(p,q)$, then an a-transition by (an instance of) variable x occurring in the first argument of f will induce an a-move of $f(x,\tau)$.

Conversely, assume that only R^f_a holds. Then, by applying rule (a_2) we obtain that $x \xrightarrow{x_\mathrm{r}}_a x_\mathrm{d}$ and, from the rules, it is not possible to derive any $\longrightarrow_a$ transition of $f(x,\tau)$ from that of x, modelling the fact that the rules for f prevent the execution of a-moves from the first argument.

Lemmas 7.5 and 7.6 formalise the decomposition of the semantics of CCS^-_f terms. We remark that, due to Lemma 4.3, at least one between L^f_μ and R^f_μ holds for each μ.

Lemma 7.5. *Let $\mu \in \{a,\bar{a},\tau\}$, t be a CCS^-_f term, x be a variable, $\mathrm{w} \in \{\mathrm{l},\mathrm{r},\mathrm{b}\}$ and σ be a closed substitution. If $\sigma(x) \xrightarrow{\mu} p$ for some process p, and $t \xrightarrow{x_\mathrm{w}}_\mu c$ for some configuration c, then $\sigma(t) \xrightarrow{\mu} \sigma[x_\mathrm{d} \mapsto p](c)$.*

Proof: The proof follows by induction on the structure of t and the derivation of the auxiliary transition $t \xrightarrow{x_\mathrm{w}}_\mu c$. □

Lemma 7.6. *Let $\alpha \in \{a,\bar{a}\}$, t be a CCS^-_f term, σ be a closed substitution and p be a closed term. Whenever $\sigma(t) \xrightarrow{\alpha} p$, then one of the following holds:*

1. *There is term t' such that $t \xrightarrow{\alpha} t'$ and $\sigma(t') = p$.*
2. *There are a variable x, a process q and a configuration c such that:*
   a. *only L^f_α holds, $\sigma(x) \xrightarrow{\alpha} q$, $t \xrightarrow{x_\mathrm{l}}_\alpha c$ and $\sigma[x_\mathrm{d} \mapsto q](c) = p$;*
   b. *only R^f_α holds, $\sigma(x) \xrightarrow{\alpha} q$, $t \xrightarrow{x_\mathrm{r}}_\alpha c$ and $\sigma[x_\mathrm{d} \mapsto q](c) = p$; or*
   c. *$L^f_\alpha \wedge R^f_\alpha$ holds, $\sigma(x) \xrightarrow{\alpha} q$, $t \xrightarrow{x_\mathrm{b}}_\alpha c$ and $\sigma[x_\mathrm{d} \mapsto q](c) = p$.*

Proof: The proof can be found in Appendix F.1. □

Next, we proceed to a more detailed analysis of the contribution of variables to the behaviour of closed instantiations of terms in which they occur.

Lemma 7.7. *Let t be a term in CCS^-_f, σ be a closed substitution and $\alpha \in \{a,\bar{a}\}$. Assume that $\sigma(t) \underline{\leftrightarrow} \sum_{i=1}^n \alpha.p_i + q$ for some n greater than the size of t and closed terms p_i, q with $p_i \underline{\not\leftrightarrow} p_j$ whenever $i \neq j$. Then t has a summand x, for some variable x, such that $\sigma(x) \underline{\leftrightarrow} \sum_{j\in J} \alpha.p_j + q'$ for some $J \subseteq \{1,\dots,n\}$, with $|J| \geq 2$, and some closed term q'.*

Proof: The proof can be found in Appendix F.2. □

The next result shows a particular case of Lemma 7.7, in which we can infer that, provided the term t has only one summand and has neither $\mathbf{0}$ summands nor factors, not only is a variable x responsible for the additional behaviour of t, but that t coincides with x.

Lemma 7.8. *Let t be a term in CCS^-_f that does not have $+$ as head operator, and let σ be a closed substitution. Let $\alpha \in \{a,\bar{a}\}$ and $\mu \in \{a,\bar{a},\tau\}$ with $\alpha \neq \mu$. Assume that $\sigma(t)$ has neither $\mathbf{0}$ summands nor factors, and that $\sigma(t) \underline{\leftrightarrow} \alpha.\mu^{\leq i_1} + \cdots + \alpha.\mu^{\leq i_m}$, for some $m > 1$ and $1 \leq i_1 < \ldots < i_m$. Then $t = x$, for some variable x.*

Proof: The proof can be found in Appendix F.4. □

We can now establish whether some of the initial behaviour of two bisimilar terms is determined by the same variable.

Proposition 7.9. *Let x be a variable and t, u be CCS^-_f terms with $t \underline{\leftrightarrow} u$ and such that neither t nor u have $\mathbf{0}$ summands or factors. If t has a summand x, then so does u.*

Proof: The proof can be found in Appendix F.5. □

8 Negative result: the case $L^f_a, L^f_{\bar{a}}, L^f_\tau$

In this section we discuss the nonexistence of a finite axiomatisation of CCS^-_f in the case of an operator f that, modulo bisimilarity, distributes over summation in one of its arguments. We will expand only the case of f distributing in the first argument. (The case of distributivity in the second argument follows by a straightforward adaptation of the arguments we will use in this section.) Hence, in the current setting, we can assume that the set of SOS rules for f is the following:

$$\frac{x_1 \xrightarrow{\mu} y_1}{f(x_1,x_2) \xrightarrow{\mu} y_1\|x_2}\ \forall\mu \in \{a,\bar{a},\tau\} \qquad \frac{x_1 \xrightarrow{\alpha} y_1 \quad x_2 \xrightarrow{\bar{\alpha}} y_2}{f(x_1,x_2) \xrightarrow{\tau} y_1\|y_2}$$

namely, only L^f_μ holds for each action μ, and only $S_{\alpha,\bar{\alpha}}$ holds for some $\alpha \in \{a,\bar{a}\}$.

According to the proof strategy sketched in Section 5, we now introduce a particular family of equations on which we will build our negative result. We define

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

$$e_n : \quad f(\alpha, p_n) \approx \alpha p_n + \sum_{i=0}^{n} \tau \alpha^{\leq i} \quad (n \geq 0) \ .$$

It is not difficult to check that the infinite family of equations e_n is sound with respect to bisimilarity.

Our order of business is now to prove the instance of Theorem 5.1 considering the family of equations e_n above, showing that no finite collection of equations over CCS_f^- that are sound modulo bisimilarity can prove all of the equations e_n $(n \geq 0)$.

Formally, we prove the following theorem:

Theorem 8.1. *Assume an operator f such that only L^f_μ holds for each action μ and only $S^f_{\alpha,\bar{\alpha}}$ holds. Let $\mathcal{E}$ be a finite axiom system over CCS_f^- that is sound modulo $\leftrightarrow$. Let n be larger than the size of each term in the equations in $\mathcal{E}$. Assume that p and q are closed terms that contain no occurrences of $\mathbf{0}$ as a summand or factor, and that $p, q \leftrightarrow f(\alpha, p_n)$. If $\mathcal{E} \vdash p \approx q$ and p has a summand bisimilar to $f(\alpha, p_n)$, then so does q.*

Proof: The proof can be found in Appendix G. □

Then, since the left-hand side of equation e_n, viz. the term $f(\alpha, p_n)$, has a summand bisimilar to $f(\alpha, p_n)$, whilst the right-hand side, viz. the term $\alpha p_n + \sum_{i=0}^{n} \tau \alpha^{\leq i}$, does not, we can conclude that the infinite collection of equations $\{e_n \mid n \geq 0\}$ is the desired witness family. Theorem 5.1 is then proved for the considered class of auxiliary binary operators.

9 Negative result: the case $L^f_\alpha \wedge R^f_\alpha$

In this section we investigate the first case, out of three, related to an operator f that does not distribute, modulo bisimilarity, over summation in either of its arguments.

We choose $\alpha \in \{a, \bar{a}\}$ and we assume that the set of rules for f includes

$$\frac{x_1 \xrightarrow{\alpha} y_1}{f(x_1, x_2) \xrightarrow{\alpha} y_1 \| x_2} \qquad \frac{x_2 \xrightarrow{\alpha} y_2}{f(x_1, x_2) \xrightarrow{\alpha} x_1 \| y_2} \ ,$$

namely, predicate $L^f_\alpha \wedge R^f_\alpha$ holds for f.

We stress that the validity of the negative result we prove in this section does not depend on which types of rules with labels $\bar{\alpha}$ and τ are available for f. Moreover, the case of an operator for which $L^f_{\bar{\alpha}} \wedge R^f_{\bar{\alpha}}$ holds can be easily obtained from the one we are considering, and it is therefore omitted.

We now introduce the infinite family of valid equations, modulo bisimilarity, that will allow us to obtain the negative result in the case at hand. We define

$$q_n = \sum_{i=0}^{n} \alpha \bar{\alpha}^{\leq i} \qquad\qquad (n \geq 0)$$

$$e_n : f(\alpha, q_n) \approx \alpha q_n + \sum_{i=0}^{n} \alpha(\alpha \| \bar{\alpha}^{\leq i}) \qquad (n \geq 0).$$

Following the proof strategy from Section 5, we aim to show that, when n is *large enough*, the witness property of having a summand bisimilar to $f(\alpha, q_n)$ is preserved by derivations from a finite, sound axiom system $\mathcal{E}$, as stated in the following theorem:

Theorem 9.1. *Assume an operator f such that $L^f_\alpha \wedge R^f_\alpha$ holds. Let $\mathcal{E}$ be a finite axiom system over CCS_f^- that is sound modulo $\leftrightarrow$. Let n be larger than the size of each term in the equations in $\mathcal{E}$. Assume p and q are closed terms that contain no $\mathbf{0}$ summands or factors, and $p, q \leftrightarrow f(\alpha, q_n)$. If $\mathcal{E} \vdash p \approx q$ and p has a summand bisimilar to $f(\alpha, q_n)$, then so does q.*

Proof: The proof can be found in Appendix H. □

Then, we can conclude that the infinite collection of equations $\{e_n \mid n \geq 0\}$ is the desired witness family. In fact, the left-hand side of equation e_n, viz. the term $f(\alpha, q_n)$, has a summand bisimilar to $f(\alpha, q_n)$, whilst the right-hand side, viz. the term $\alpha q_n + \sum_{i=0}^{n} \alpha(\alpha \| \bar{\alpha}^{\leq i})$, does not. This concludes the proof of Theorem 5.1 in this case.

10 Negative result: the case L^f_α, $R^f_{\bar{\alpha}}$

In this section we deal with the second case related to an operator f that does not distribute over summation in either argument. This time, given $\alpha \in \{a, \bar{a}\}$, we assume that operator f has only one rule with label α and only one rule with label $\bar{\alpha}$, and moreover we assume such rules to be of different types. In detail, we expand the case in which for action α only the predicate L^f_α holds, and for action $\bar{\alpha}$ only $R^f_{\bar{\alpha}}$ holds, namely f has rules:

$$\frac{x_1 \xrightarrow{\alpha} y_1}{f(x_1, x_2) \xrightarrow{\alpha} y_1 \| x_2} \qquad \frac{x_2 \xrightarrow{\bar{\alpha}} y_2}{f(x_1, x_2) \xrightarrow{\bar{\alpha}} x_1 \| y_2} \ .$$

Once again, the proof for the symmetric case with $L^f_{\bar{\alpha}}$ and R^f_α holding is omitted.

To obtain the proof of the negative result, we consider the same family of witness processes $f(\alpha, p_n)$ from Section 8. However, differently from the previous case, the definition of the witness family of equations depends on which rules of type (5) are available for f. More precisely, we need to split the proof of the negative result into two cases, according to whether the rules for f allow α and p_n to synchronise or not.

10.1 Case 1: Possibility of synchronisation

Assume first that $S^f_{\alpha,\bar{\alpha}}$ holds, so that the rule

$$\frac{x_1 \xrightarrow{\alpha} y_1 \quad x_2 \xrightarrow{\bar{\alpha}} y_2}{f(x_1, x_2) \xrightarrow{\tau} y_1 \| y_2}$$

allows for synchronisation between α and p_n. In this setting, the infinite family of equations

$$e_n : f(\alpha, p_n) \approx \alpha p_n + \sum_{i=0}^{n} \bar{\alpha}(\alpha \| \alpha^{\leq i}) + \sum_{i=0}^{n} \tau \alpha^{\leq i} \quad (n \geq 0)$$

is sound modulo bisimilarity and it constitutes a family of witness equations, as stated in the following:

Theorem 10.1. *Assume an operator f such that only L^f_α holds for α, only $R^f_{\bar\alpha}$ holds for $\bar\alpha$, and $S^f_{\alpha,\bar\alpha}$ holds.*

Let $\mathcal{E}$ be a finite axiom system over CCS_f^- that is sound modulo $\leftrightarrow$. Let n be larger than the size of each term in the equations in $\mathcal{E}$. Assume that p and q contain no $\mathbf{0}$ summands or factors, and $p, q \leftrightarrow f(\alpha, p_n)$. If $\mathcal{E} \vdash p \approx q$ and p has a summand bisimilar to $f(\alpha, p_n)$, then so does q.

Proof: The proof can be found in Appendix I. □

This proves Theorem 5.1 in the considered setting, as the left-hand side of equation e_n, viz. the term $f(\alpha, p_n)$, has a summand bisimilar to $f(\alpha, p_n)$, whilst the right-hand side, viz. the term $\alpha p_n + \sum_{i=0}^n \bar\alpha(\alpha \| \bar\alpha^{\le i}) + \sum_{i=0}^n \tau\alpha^{\le i}$, does not.

10.2 Case 2: No synchronisation

Assume now that the synchronisation between α and p_n is prevented, namely only $S^f_{\bar\alpha,\alpha}$ holds. Then, the witness family of equations changes as follows:

$$e_n : f(\alpha, p_n) \approx \alpha p_n + \sum_{i=0}^{n} \bar\alpha(\alpha \| \alpha^{\le i}) \quad (n \ge 0) \ .$$

Our order of business is then to prove the following:

Theorem 10.2. *Assume an operator f such that only L^f_α holds for α, only $R^f_{\bar\alpha}$ holds for $\bar\alpha$, and only $S^f_{\bar\alpha,\alpha}$ holds.*

Let $\mathcal{E}$ be a finite axiom system over CCS_f^- that is sound modulo $\leftrightarrow$. Let n be larger than the size of each term in the equations in $\mathcal{E}$. Assume p and q contain no $\mathbf{0}$ summands or factors, and $p, q \leftrightarrow f(\alpha, p_n)$. If $\mathcal{E} \vdash p \approx q$ and p has a summand bisimilar to $f(\alpha, p_n)$, then so does q.

Proof: The proof can be found in Appendix J. □

Once again, the validity of Theorem 5.1 follows by noticing that the left-hand side of equation e_n, viz. the term $f(\alpha, p_n)$, has a summand bisimilar to $f(\alpha, p_n)$, whilst the right-hand side, viz. the term $\alpha p_n + \sum_{i=0}^n \bar\alpha(\alpha \| \bar\alpha^{\le i})$, does not.

11 Negative result: the case L^f_τ

This section considers the last case in our analysis, namely that of an operator f that does not distribute, modulo bisimilarity, over summation in either argument and that has the same rule type for actions $\alpha, \bar\alpha$. Here, we present solely the case in which L^f_τ holds, and only $R^f_\alpha, R^f_{\bar\alpha}$ hold for $\alpha, \bar\alpha$, namely f has rules:

$$\frac{x_1 \xrightarrow{\tau} y_1}{f(x_1, x_2) \xrightarrow{\tau} y_1 \| x_2} \qquad \frac{x_2 \xrightarrow{\alpha} y_2}{f(x_1, x_2) \xrightarrow{\alpha} x_1 \| y_2} \qquad \frac{x_2 \xrightarrow{\bar\alpha} y_2}{f(x_1, x_2) \xrightarrow{\bar\alpha} x_1 \| y_2} .$$

The symmetric case can be obtained from this one in a straightforward manner.

Interestingly, the validity of the negative result we consider in this section is independent of which rules of type (5) are available for f, and of the validity of the predicate R^f_τ.

Consider the family of equations defined by:

$$e_n : f(\tau, q_n) \approx \tau q_n + \sum_{i=0}^{n} \alpha(\tau \| \bar\alpha^{\le i}) \qquad (n \ge 0)$$

where the processes q_n are the same used in Section 9. Theorem 11.1 below proves that the collection of equations e_n, $n \ge 0$, is a witness family of equations for our negative result.

Theorem 11.1. *Assume an operator f such that L^f_τ holds and only R^f_α and $R^f_{\bar\alpha}$ hold for actions α and $\bar\alpha$.*

Let $\mathcal{E}$ be a finite axiom system over CCS_f^- that is sound modulo $\leftrightarrow$. Let n be larger than the size of each term in the equations in $\mathcal{E}$. Assume p and q contain no $\mathbf{0}$ summands or factors, and $p, q \leftrightarrow f(\tau, q_n)$. If $\mathcal{E} \vdash p \approx q$ and p has a summand bisimilar to $f(\tau, q_n)$, then so does q.

Proof: The proof can be found in Appendix K. □

As the left-hand side of equation e_n, viz. the term $f(\tau, q_n)$, has a summand bisimilar to $f(\tau, q_n)$, whilst the right-hand side, viz. the term $\tau q_n + \sum_{i=0}^n \alpha(\tau \| \bar\alpha^{\le i})$, does not, we can conclude that the collection of infinitely many equations e_n ($n \ge 0$) is the desired witness family. This concludes the proof of Theorem 5.1 for this case and our proof of Theorem 4.7.

12 Conclusions

In this paper, we have shown that, under a number of reasonable assumptions, we cannot use a single binary auxiliary operator f, whose semantics is defined via inference rules in the de Simone format, to obtain a finite axiomatisation of bisimilarity over the recursion, restriction and relabelling free fragment of CCS. Our result constitutes a first step towards a definitive justification of the canonical standing of the left and communication merge operators by Bergstra and Klop. We envisage the following ways in which we might generalise the contribution presented in this study. Firstly, we will try to get rid of Assumptions 2 and 3. Next, it is natural to relax Assumption 1 by considering the GSOS format [13] in place of the de Simone format. However, as shown by the heavy amount of technical results necessary to prove our main result even in our simplified setting, we believe that this generalisation cannot be obtained in a straightforward manner and that it will require the introduction of new techniques. It would also be very interesting to explore whether some version of problem (P) can be solved using existing results from equational logic and universal algebra. So far, our efforts in this direction have been unsuccessful.

References

[1] Luca Aceto. 2003. Some of My Favourite Results in Classic Process Algebra. *Bulletin of the EATCS* 81 (2003), 90–108.

[2] Luca Aceto, Bard Bloom, and Frits W. Vaandrager. 1994. Turning SOS Rules into Equations. *Inf. Comput.* 111, 1 (1994), 1–52. https://doi.org/10.1006/inco.1994.1040

[3] Luca Aceto, Taolue Chen, Anna Ingólfsdóttir, Bas Luttik, and Jaco van de Pol. 2011. On the axiomatizability of priority II. *Theor. Comput. Sci.* 412, 28 (2011), 3035–3044. https://doi.org/10.1016/j.tcs.2011.02.033

[4] Luca Aceto, Wan Fokkink, Anna Ingólfsdóttir, and Bas Luttik. 2005. CCS with Hennessy's merge has no finite-equational axiomatization. *Theor. Comput. Sci.* 330, 3 (2005), 377–405. https://doi.org/10.1016/j.tcs.2004.10.003

[5] Luca Aceto, Wan Fokkink, Anna Ingólfsdóttir, and Bas Luttik. 2005. Finite Equational Bases in Process Algebra: Results and Open Questions. In *Processes, Terms and Cycles: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday (Lecture Notes in Computer Science)*, Aart Middeldorp, Vincent van Oostrom, Femke van Raamsdonk, and Roel C. de Vrijer (Eds.), Vol. 3838. Springer, 338–367. https://doi.org/10.1007/11601548_18

[6] Luca Aceto, Wan Fokkink, Anna Ingólfsdóttir, and Bas Luttik. 2009. A finite equational base for CCS with left merge and communication merge. *ACM Trans. Comput. Log.* 10, 1 (2009), 6:1–6:26. https://doi.org/10.1145/1459010.1459016

[7] Luca Aceto, Wan Fokkink, Anna Ingólfsdóttir, and Sumit Nain. 2006. Bisimilarity is not finitely based over BPA with interrupt. *Theor. Comput. Sci.* 366, 1-2 (2006), 60–81. https://doi.org/10.1016/j.tcs.2006.07.003

[8] Luca Aceto, Wan Fokkink, and Chris Verhoef. 2001. Structural Operational Semantics. In *Handbook of Process Algebra*. North-Holland / Elsevier, 197–292. https://doi.org/10.1016/b978-044482830-9/50021-7

[9] Luca Aceto, Anna Ingólfsdóttir, Bas Luttik, and Paul van Tilburg. 2008. Finite Equational Bases for Fragments of CCS with Restriction and Relabelling. In *Proceedings of IFIP TCS 2008 (IFIP)*, Vol. 273. 317–332. https://doi.org/10.1007/978-0-387-09680-3_22

[10] Jan A. Bergstra and Jan Willem Klop. 1984. The Algebra of Recursively Defined Processes and the Algebra of Regular Processes. In *Proceedings of ICALP 2011 (Lecture Notes in Computer Science)*, Vol. 172. 82–94. https://doi.org/10.1007/3-540-13345-3_7

[11] Jan A. Bergstra and Jan Willem Klop. 1984. Process Algebra for Synchronous Communication. *Information and Control* 60, 1-3 (1984), 109–137. https://doi.org/10.1016/S0019-9958(84)80025-X

[12] Jan A. Bergstra and Jan Willem Klop. 1985. Algebra of Communicating Processes with Abstraction. *Theor. Comput. Sci.* 37 (1985), 77–121. https://doi.org/10.1016/0304-3975(85)90088-X

[13] Bard Bloom, Sorin Istrail, and Albert R. Meyer. 1995. Bisimulation Can't be Traced. *J. ACM* 42, 1 (1995), 232–268. https://doi.org/10.1145/200836.200876

[14] Robert de Simone. 1985. Higher-Level Synchronising Devices in Meije-SCCS. *Theor. Comput. Sci.* 37 (1985), 245–267. https://doi.org/10.1016/0304-3975(85)90093-3

[15] Jan Friso Groote. 1990. A New Strategy for Proving omega-Completeness applied to Process Algebra. In *Proceedings of CONCUR '90 (Lecture Notes in Computer Science)*, Vol. 458. 314–331. https://doi.org/10.1007/BFb0039068

[16] Matthew Hennessy. 1988. Axiomatising Finite Concurrent Processes. *SIAM J. Comput.* 17, 5 (1988), 997–1017. https://doi.org/10.1137/0217063

[17] Matthew Hennessy and Robin Milner. 1985. Algebraic Laws for Nondeterminism and Concurrency. *J. ACM* 32, 1 (1985), 137–161. https://doi.org/10.1145/2455.2460

[18] Robert M. Keller. 1976. Formal Verification of Parallel Programs. *Commun. ACM* 19, 7 (1976), 371–384. https://doi.org/10.1145/360248.360251

[19] Bas Luttik and Vincent van Oostrom. 2005. Decomposition orders another generalisation of the fundamental theorem of arithmetic. *Theor. Comput. Sci.* 335, 2-3 (2005), 147–186. https://doi.org/10.1016/j.tcs.2004.11.019

[20] Robin Milner. 1980. *A Calculus of Communicating Systems*. Lecture Notes in Computer Science, Vol. 92. Springer. https://doi.org/10.1007/3-540-10235-3

[21] Robin Milner. 1989. *Communication and concurrency*. Prentice Hall.

[22] Robin Milner and Faron Moller. 1993. Unique Decomposition of Processes. *Theor. Comput. Sci.* 107, 2 (1993), 357–363. https://doi.org/10.1016/0304-3975(93)90176-T

[23] Faron Moller. 1989. *Axioms for Concurrency*. Ph.D. Dissertation. Department of Computer Science, University of Edinburgh. Report CST-59-89. Also published as ECS-LFCS-89-84.

[24] Faron Moller. 1990. The Importance of the Left Merge Operator in Process Algebras. In *Proceedings of ICALP '90 (Lecture Notes in Computer Science)*, Vol. 443. 752–764. https://doi.org/10.1007/BFb0032072

[25] Faron Moller. 1990. The Nonexistence of Finite Axiomatisations for CCS Congruences. In *Proceedings of LICS '90*. 142–153. https://doi.org/10.1109/LICS.1990.113741

[26] David M. R. Park. 1981. Concurrency and Automata on Infinite Sequences. In *Proceedings of GI-Conference (Lecture Notes in Computer Science)*, Vol. 104. 167–183. https://doi.org/10.1007/BFb0017309

[27] Gordon D. Plotkin. 1981. *A structural approach to operational semantics*. Report DAIMI FN-19. Computer Science Department, Aarhus University.

[28] Walter Taylor. 1977. Equational Logic. In *Contributions to Universal Algebra*. North-Holland, 465 – 501. https://doi.org/10.1016/B978-0-7204-0725-9.50040-X