On the Axiomatisation of Branching Bisimulation Congruence over CCS

Luca Aceto
Reykjavik University, Iceland
Gran Sasso Science Institute, L'Aquila, Italy

Valentina Castiglioni
Reykjavik University, Iceland

Anna Ingólfsdóttir
Reykjavik University, Iceland

Bas Luttik
Eindhoven University of Technology, The Netherlands

Abstract

In this paper we investigate the equational theory of (the restriction, relabelling, and recursion free fragment of) CCS modulo rooted branching bisimilarity, which is a classic, bisimulation-based notion of equivalence that abstracts from internal computational steps in process behaviour. Firstly, we show that CCS is not finitely based modulo the considered congruence. As a key step of independent interest in the proof of that negative result, we prove that each CCS process has a unique parallel decomposition into indecomposable processes modulo branching bisimilarity. As a second main contribution, we show that, when the set of actions is finite, rooted branching bisimilarity has a finite equational basis over CCS enriched with the left merge and communication merge operators from ACP.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting

Keywords and phrases Equational basis, Weak semantics, CCS, Parallel composition

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.6

Related Version *Technical report version with full proofs*: `http://arxiv.org/abs/2206.13927` [4]

Funding 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). 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).

Acknowledgements We thank the reviewers for their valuable comments that helped us to improve our contribution.

1 Introduction

This paper is a new chapter in the saga of the axiomatisation of the *parallel composition operator* $\|$ (also known as "*full*" *merge* [12, 13]) of the Calculus of Communicating Systems (CCS) [27]. The saga has its roots in the works [22, 23], in which Hennessy and Milner studied the *equational theory* of (recursion free) CCS and proposed a *ground-complete axiomatisation* for it modulo *strong bisimilarity* and *observational congruence*, two classic notions of behavioural *congruence* (i.e., an equivalence relation that is compositional with respect to the language operators) that allow one to establish whether two processes have the same *observable behaviour* [34]. That axiomatisation included infinitely many axioms, which were instances of the *expansion law* used to "simulate equationally" the operational semantics of $\|$. Then, Bergstra and Klop showed, in [12], that a *finite* ground-complete axiomatisation

33rd International Conference on Concurrency Theory (CONCUR 2022).
Editors: Bartek Klin, Sławomir Lasota, and Anca Muscholl; Article No. 6; pp. 6:1–6:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

modulo bisimilarity can be obtained by enriching CCS with two auxiliary operators, i.e., the *left merge* $\mathbin{\|\!\_}$ and the *communication merge* $|$, expressing one step in the pure interleaving and the synchronous behaviour of $\|$, respectively. Their result was strengthened by Aceto et al. in [7], 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 $\mathbin{\|\!\_}$ and $|$ are *sufficient* to finitely axiomatise $\|$ over CCS modulo bisimilarity. (*Henceforth, we only consider the recursion, restriction and relabelling free fragment of CCS.*) Moller showed, in [30, 31], that they are also *necessary.* He considered a minimal fragment of CCS, including only the inactive process, action prefixing, nondeterministic choice and interleaving, and proved that, even in the presence of a single action, bisimilarity does not afford a finite ground-complete axiomatisation over that language. Moller's proof technique was then used to show that the same negative result holds if we replace $\mathbin{\|\!\_}$ and $|$ with the so called *Hennessy's merge* [21], which denotes an asymmetric interleaving with communication, or, more generally, with a single binary auxiliary operator satisfying three assumptions given in [3].

The aforementioned works considered equational characterisations of $\|$ modulo strong bisimilarity. However, a plethora of behavioural congruences have been proposed in the literature, corresponding to different levels of abstraction from the information on process execution. Hence, another chapter in the saga consisted in extending the studies recalled above to the behavioural congruences in van Glabbeek's linear time-branching time spectrum [15]. The work [5] delineated the *boundary* between finite and non-finite axiomatisability of $\|$ modulo all the congruences in the spectrum.

Our contribution: branching bisimulation congruence. Some information on process behaviour can either be considered irrelevant or be unavailable to an external observer. *Weak behavioural semantics* have been introduced to study the effects of these unobservable (or *silent*) actions, usually denoted by τ, on the observable behaviour of processes, each semantics considering a different level of abstraction. A taxonomy of weak semantics is given in [17], and studies on the equational theories of various of these semantics have been carried out over the algebra BCCSP, which consists of the basic operators from CCS and CSP [24] but does not include $\|$ (see, among others, [6, 14, 20, 23, 33]). A finite, ground-complete axiomatisation of parallel composition modulo *rooted weak bisimilarity* (also known as *observational congruence* [23]) is provided by Bergstra and Klop in [13] over the algebra ACP$_\tau$ that includes the auxiliary operators $\mathbin{\|\!\_}$ and $|$. To the best of our knowledge, the only study on the axiomatisability of CCS's $\|$ over open terms modulo weak congruences is the negative result from [2], which shows that a class of weak congruences (including rooted weak bisimilarity) does not afford a finite, complete axiomatisation over the open terms of the minimal fragment of CCS with interleaving.

In this paper we focus on *branching bisimilarity* [19], which generalises strong bisimilarity to abstract away from τ-steps of terms while preserving their *branching structure* [19, 20], and its *rooted* version, which is a congruence with respect to CCS operators.

As a first main contribution, we show that *rooted branching bisimilarity affords no finite ground-complete axiomatisation over CCS.* To this end, we adapt the proof-theoretic technique used by Moller to prove the corresponding negative result for strong bisimilarity. We remark that, even though the general proof strategy is a natural extension of Moller's, our proof requires a number of original, non-trivial technical results on (rooted) branching bisimilarity.

In particular, we observe that equational proofs of τ-free equations might involve terms having occurrences of τ in some intermediate steps (see, e.g., page 175 of Moller's thesis [30]), and our proof of the negative result for rooted branching bisimilarity will account for those uses of τ, thus making our results special for the considered weak congruence. Moreover, as an intermediate step in our proof, we establish a result of independent interest: we show that *each CCS process has a unique decomposition into indecomposable processes modulo branching bisimilarity.* A similar result was proven in [26], but only for interleaving parallel composition. Here, we extend this result to the full merge operator, including thus the possibility of communication between the parallel components.

Having established the negative result, a natural question is whether the use of the auxiliary operators from [12] can help us to obtain an equational basis for rooted branching bisimilarity. Hence, as our second main contribution, we consider the language $\mathrm{CCS}_{\mathrm{LC}}$, namely CCS enriched with $\mathbin{\|\!\_}$ and $|$, and *we provide a complete axiomatisation for rooted branching bisimilarity over CCS_{LC} that is finite when so is the set of actions over which terms are defined.* This axiomatisation is obtained by extending the complete axiom system for strong bisimilarity over $\mathrm{CCS}_{\mathrm{LC}}$ from [7] with axioms expressing the behaviour of $\mathbin{\|\!\_}$ and $|$ in the presence of τ-actions (from [13]), and with the suitable τ-laws (from [20, 23]) necessary to deal with rooted branching bisimilarity. Specifically, we will see that we can express equationally the fact that left merge and communication merge distribute over choice (left merge in one argument, communication merge in both), thus allowing us to expand the behaviour of the parallel components using only a finite number of axioms, regardless of their size. A key step in the proof of the completeness result consists in another intermediate original contribution of this work: the definition of the semantics of *open* $\mathrm{CCS}_{\mathrm{LC}}$ terms.

Our contribution can then be summarised as follows:

1. We show that every branching equivalence class of CCS processes has a unique parallel decomposition into indecomposables.
2. We prove that rooted branching bisimilarity admits no finite equational axiomatisation over CCS.
3. We define the semantics of open $\mathrm{CCS}_{\mathrm{LC}}$ terms.
4. We provide a (finite) complete axiomatisation for $\sim_{\mathtt{RBB}}$ over $\mathrm{CCS}_{\mathrm{LC}}$.

2 Background

Labelled transition systems. As semantic model we consider classic *labelled transition systems* [25]. We assume a non-empty set of action names $\mathcal{A}$, and we let $\overline{\mathcal{A}}$ denote the set of action co-names, i.e., $\overline{\mathcal{A}} = \{\overline{a} \mid a \in \mathcal{A}\}$. As usual, we postulate that $\overline{\overline{a}} = a$ and $a \neq \overline{a}$ for all $a \in \mathcal{A}$. Then, we define $\mathcal{A}_\tau = \mathcal{A} \cup \overline{\mathcal{A}} \cup \{\tau\}$, where $\tau \notin \mathcal{A} \cup \overline{\mathcal{A}}$. Henceforth, we let $\mu, \nu, \dots$ range over actions in $\mathcal{A}_\tau$, and $\alpha, \beta, \dots$ range over actions in $\mathcal{A} \cup \overline{\mathcal{A}}$.

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

As usual, we use $p \xrightarrow{\mu} p'$ in lieu of $(p, \mu, p') \in \rightarrow$. For each $p \in \mathbf{P}$ and $\mu \in \mathcal{A}$, we write $p \xrightarrow{\mu}$ if $p \xrightarrow{\mu} p'$ holds for some p', and $p \not\xrightarrow{\mu}$ otherwise. The *initials* of p are the actions that label the outgoing transitions of p, that is, $\mathrm{init}(p) = \{\mu \in \mathcal{A}_\tau \mid p \xrightarrow{\mu}\}$.

Table 1 The SOS rules for CCS operators ($\mu \in \mathcal{A}_\tau$, $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$).

$$\frac{}{\mu.t \xrightarrow{\mu} t} \qquad \frac{t \xrightarrow{\mu} t'}{t + u \xrightarrow{\mu} t'} \qquad \frac{t \xrightarrow{\mu} t'}{t \parallel u \xrightarrow{\mu} t' \parallel u} \qquad \frac{t \xrightarrow{\alpha} t' \quad u \xrightarrow{\overline{\alpha}} u'}{t \parallel u \xrightarrow{\tau} t' \parallel u'}$$

The language CCS. We consider the recursion, relabelling and restriction free fragment of Milner's CCS [28], which for simplicity we still call CCS, given by the following grammar:

$$t ::= \ \mathbf{0} \mid x \mid \mu.t \mid t + t \mid t \parallel t \ ,$$

where x is a variable drawn from a countably infinite set $\mathcal{V}$ disjoint from $\mathcal{A}_\tau$, and $\mu \in \mathcal{A}_\tau$. We use the *Structural Operational Semantics* (SOS) framework [35, 36] to equip processes with an operational semantics. The SOS rules (or inference rules) for the CCS operators given above are reported in Table 1 (symmetric rules for + and $\parallel$ are omitted).

We shall use the meta-variables t, u, v, w to range over process terms, and write $\mathrm{var}(t)$ for the collection of variables occurring in the term t. We use a *summation* $\sum_{i \in \{1,\dots,k\}} t_i$ to abbreviate $t_1 + \dots + t_k$, where the empty sum represents $\mathbf{0}$. We call the term t_j ($j \in \{1, \dots, k\}$) a *summand* of $t = \sum_{i \in \{1,\dots k\}} t_i$ if it does not have + as head operator. 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, r. Moreover, we omit trailing $\mathbf{0}$'s from terms. A *(closed) substitution* 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. We let $\sigma[x \mapsto p]$ denote the substitution that maps the variable x into process p and behaves like σ on all other variables. In particular, $[x \mapsto p]$ denotes the substitution that maps the variable x into process p and behaves like the identity on all other variables.

The inference rules in Table 1 allow us to derive valid transitions between closed terms. The operational semantics for our language is then modelled by the LTS whose processes are the closed terms, and whose labelled transitions are those that are provable from the SOS rules. Henceforth, we let $\mathbf{P}$ denote the set of CCS processes. We remark that whenever $p \xrightarrow{\mu} p'$, then $\mathrm{size}(p) > \mathrm{size}(p')$.

Branching bisimilarity. *Branching bisimilarity* is a bisimulation-based behavioural equivalence that abstracts away from computation steps in processes that are deemed unobservable, while preserving their *branching structure*. The abstraction is achieved by labelling these computation steps with τ, and giving τ-labelled transitions a special treatment in the definition of the behavioural equivalence. Preservation of the branching structure is mainly due to the *stuttering* nature of branching bisimulation, which guarantees that the behaviour of a term is preserved in the execution of a sequence of silent steps [19, 20].

Let $\xrightarrow{\varepsilon}$ denote the reflexive and transitive closure of the transition $\xrightarrow{\tau}$.

▶ **Definition 2** (Branching bisimilarity). *Let* $(\mathbf{P}, \mathcal{A}_\tau, \rightarrow)$ *be a LTS.* Branching bisimilarity, *denoted by* $\sim_{\mathtt{BB}}$*, is the largest symmetric relation over* $\mathbf{P}$ *such that, whenever* $p \sim_{\mathtt{BB}} q$*, if* $p \xrightarrow{\mu} p'$*, then either:*

- $\mu = \tau$ *and* $p' \sim_{\mathtt{BB}} q$*, or*
- *there are processes* q', q'' *such that* $q \xrightarrow{\varepsilon} q'' \xrightarrow{\mu} q'$*,* $p \sim_{\mathtt{BB}} q''$*, and* $p' \sim_{\mathtt{BB}} q'$*.*

Table 2 Some axioms for rooted branching bisimilarity.

Some axioms for bisimilarity over CCS:

A0	$x + \mathbf{0} \approx x$	P0	$x \parallel \mathbf{0} \approx x$
A1	$x + y \approx y + x$	P1	$x \parallel y \approx y \parallel x$
A2	$(x + y) + z \approx x + (y + z)$	P2	$(x \parallel y) \parallel z \approx x \parallel (y \parallel z)$
A3	$x + x \approx x$		

Additional axioms for rooted branching bisimilarity over CCS:

TB	$\mu(\tau(x + y) + y) \approx \mu(x + y)$	T1	$\mu\tau x \approx \mu x$

Branching bisimilarity satisfies the *stuttering property* [20, Lemma 2.5]: *Assume that* $p \sim_{\mathtt{BB}} q$*. Whenever* $p \xrightarrow{\tau} p_1 \xrightarrow{\tau} \dots \xrightarrow{\tau} p_n$ *and* $p_n \sim_{\mathtt{BB}} q$*, for some* $n \geq 1$*, then* $p_i \sim_{\mathtt{BB}} q$ *for all* $i = 1, \dots, n - 1$*.*

To guarantee compositional reasoning over a process language, we require a behavioural equivalence $\sim$ to be a *congruence* with respect to all language operators. This consists in verifying whether, for all n-ary operators f

$$\text{if } t_i \sim t'_i \text{ for all } i = 1, \dots, n, \text{ then } f(t_1, \dots, t_n) \sim f(t'_1, \dots, t'_n).$$

It is well known that branching bisimilarity is an equivalence relation [11, 20]. Moreover, action prefixing and parallel composition satisfy the *simple BB cool rule format* [18] and hence $\sim_{\mathtt{BB}}$ is compositional with respect to those operators. However, $\sim_{\mathtt{BB}}$ is not a congruence with respect to nondeterministic choice. To remedy this inconvenience, the *root condition* is introduced: *rooted branching bisimilarity* behaves like strong bisimilarity on the initial transitions, and like branching bisimilarity on subsequent transitions.

▶ **Definition 3** (Rooted branching bisimilarity)**.** Rooted branching bisimilarity*, denoted by* $\sim_{\mathtt{RBB}}$*, is the symmetric relation over* $\mathbf{P}$ *such that, whenever* $p \sim_{\mathtt{RBB}} q$*, if* $p \xrightarrow{\mu} p'$*, then there is a process* q' *such that* $q \xrightarrow{\mu} q'$ *and* $p' \sim_{\mathtt{BB}} q'$*.*

It is well known that rooted branching bisimilarity is an equivalence relation [11, 20], and that $\sim_{\mathtt{RBB}}$ is a congruence over CCS (see, e.g., [18]).

Equational Logic. An *axiom system* $\mathcal{E}$ is a collection of (*process*) *equations* $t \approx u$ over the considered language, thus CCS in this paper. 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*.

We assume, without loss of generality, that the substitution rule is only applied on equations $(t \approx u) \in \mathcal{E}$. In this case, $\sigma(t) \approx \sigma(u)$ is called a *substitution instance* of an axiom in $\mathcal{E}$. Moreover, by postulating that for each axiom in $\mathcal{E}$ also its symmetric counterpart is present in $\mathcal{E}$, one may assume that the symmetry rule is never used in equational proofs.

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

Henceforth, we exploit the associativity and commutativity of $+$ and $\|$ modulo the relevant behavioural equivalences. The symbol $=$ will then denote equality modulo A1–A2 and P1–P2 in Table 2.

3 The main results

Our aim is to study the axiomatisability of rooted branching bisimilarity over CCS. Our investigations produced, as main outcomes, a negative result (Theorem 4) and a positive one (Theorem 5). In detail, in the first part of the paper we prove the following theorem:

▶ **Theorem 4.** *Rooted branching bisimilarity has no finite equational ground-complete axiomatisation over CCS.*

Given the negative result, it is natural to wonder whether an equational basis for rooted branching bisimilarity can be obtained if we enrich CCS with some auxiliary operators. Considering the similarities between $\sim_{\mathtt{RBB}}$ and strong bisimilarity, the principal candidates for this role are the left merge $⫽$ and the communication merge $|$ from [12]. Indeed, we show that if we add those two operators to the syntax of CCS, then we can obtain a complete axiomatisation of rooted branching bisimilarity over the new language, denoted by $\mathrm{CCS}_{\mathrm{LC}}$. The desired equational basis is given by the axiom system $\mathcal{E}_{\mathtt{RBB}}$, which is presented fully in Table 7 in Section 10. $\mathcal{E}_{\mathtt{RBB}}$ is an extension of the complete axiom system for strong bisimilarity over $\mathrm{CCS}_{\mathrm{LC}}$ from [7] with axioms expressing the behaviour of left merge and communication merge in the presence of τ-actions (taken from [13]), and with the suitable τ-laws necessary to deal with rooted branching bisimilarity (taken from [20, 23]).

Formally, our second main contribution consists in a proof of the following theorem:

▶ **Theorem 5** (Completeness)**.** *Let t, u be CCS_{LC} terms. If $t \sim_{\mathtt{RBB}} u$, then $\mathcal{E}_{\mathtt{RBB}} \vdash t \approx u$.*

We will also argue that this axiomatisation is finite when so is the set of actions. Hence, when $\mathcal{A}$ is finite, $\mathrm{CCS}_{\mathrm{LC}}$ modulo $\sim_{\mathtt{RBB}}$ is finitely based, unlike CCS.

Considering the amount of technical results that we will need to fulfil our objectives, we devote Section 4 to a presentation of the proof strategy that we will apply to obtain Theorem 4. Sections 5–7 then present the formalisation of the ideas discussed in that section. Similarly, in Section 8 we give a high-level description of the approach that we will follow to prove Theorem 5. The technical development of the proof is then reported in Sections 9–10.

4 Proof strategy for Theorem 4

In this section we present the proof strategy we will apply to obtain Theorem 4.

Our proof follows the so-called *proof-theoretic approach* to non-finite-axiomatisability results, whose use in the field of process algebra stems from [30–32], where Moller proved that CCS modulo strong bisimilarity is not finitely based. In the proof-theoretic approach, the idea is to identify a specific property of terms parametric in $n \geq 0$, say $\mathbb{P}_n$, and show that if $\mathcal{E}$ is an arbitrary finite axiom system that is sound with respect to $\sim_{\mathtt{RBB}}$, then $\mathbb{P}_n$ is preserved by provability from $\mathcal{E}$ when n is "*large enough*". Next, we exhibit an infinite family of equations $\{\mathfrak{e}_n \mid n \geq 0\}$ over closed terms that are all sound modulo $\sim_{\mathtt{RBB}}$, but are such that only one side of $\mathfrak{e}_n$ satisfies $\mathbb{P}_n$, for each $n \geq 0$. In particular, this implies that whenever n is "large enough" then the sound equation $\mathfrak{e}_n$ cannot be proved from $\mathcal{E}$. Since $\mathcal{E}$ is an arbitrary finite sound axiom system, it follows that no finite sound axiomatisation can prove all the equations in the family $\{\mathfrak{e}_n \mid n \geq 0\}$ and therefore no finite sound axiomatisation is ground complete for CCS modulo modulo $\sim_{\mathtt{RBB}}$.

The choice of $\mathbb{P}_n$ and the family of equations. In [30–32] Moller applied the proof method sketched above to prove that strong bisimilarity has no finite, complete axiomatisation over CCS. The key idea underlying this result is that, since $\|$ does not distribute over $+$ in either of its arguments modulo strong bisimilarity, then no finite, sound axiom system can "*expand*" the initial behaviour of process $a \parallel \sum_{i=1}^{n} a^i$ (where $a^i = aa^{i-1}$ for each $i = 1, \ldots, n$, with $a^0 = \mathbf{0}$) when n is large.

Since, by definition, rooted branching bisimilarity behaves exactly like strong bisimilarity on the first step, and parallel composition does not distribute over choice in either of its arguments, modulo $\sim_{\mathtt{RBB}}$, it is natural to exploit a similar strategy to prove Theorem 4. In detail, we will consider, for each $n \geq 2$, the process $p_n = \sum_{i=2}^{n} aa^{\leq i}$, where $a^{\leq i} = \sum_{j=1}^{i} a^j$ for each $i = 2, \ldots, n$. Then, for each $n \geq 2$, the property $\mathbb{P}_n$ will consist in having a summand rooted branching bisimilar to the process $a \parallel p_n$, and we will show that, when n is large enough, $\mathbb{P}_n$ is an invariant under provability from an arbitrary finite, sound axiom system (Theorem 18). Hence, the sound equation $\mathfrak{e}_n \,:\, a \parallel p_n \approx ap_n + \sum_{i=2}^{n} a(a \parallel a^{\leq i})$ cannot be derived from $\mathcal{E}$ because its right-hand side has no summand that is rooted branching bisimilar to $a \parallel p_n$, unlike its left-hand side. Therefore no finite sound axiom system can prove the infinite family of equations $\{\mathfrak{e}_n \mid n \geq 2\}$, yielding the desired negative result.

In proving that $\mathbb{P}_n$ is invariant under provability, one pivotal ingredient will be the fact that processes p_n and $a^{\leq i}$, for $n \geq 2$ and $i \in \{2, \ldots, n\}$, are *indecomposable*. The existence of a unique parallel decomposition into indecomposable processes modulo *branching bisimilarity* over CCS with *interleaving parallel composition* was studied in [26]. In Section 6, we extend the result from [26] to the full merge operator, thus including communication (Proposition 16).

The choice of n. The choice of a sufficiently large n plays a crucial role in proving that $\mathbb{P}_n$ is an invariant under provability from a finite, sound axiom system $\mathcal{E}$ (Theorem 18). The key step in that proof deals with the case in which $p \approx q$ is a substitution instance of an equation in $\mathcal{E}$ (Proposition 20), i.e., $p = \sigma(t)$, $q = \sigma(u)$, and $t \approx u \in \mathcal{E}$ for some terms t, u and closed substitution σ. In this case, assuming that $n > \mathrm{size}(t)$, we can prove that if $p = \sigma(t)$ satisfies $\mathbb{P}_n$ then this is due to the behaviour of $\sigma(x)$ for some variable x. In order to reach this conclusion, in Section 5, we study how the behaviour of closed instances of terms may depend on the behaviour of the closed instances of variables occurring in them. Moreover, we one can show that if $t \approx u$ is sound modulo rooted branching bisimilarity and x occurs in t, then it occurs also in u. Hence, we can infer that $\sigma(x)$ triggers in $\sigma(u)$ the same behaviour that it induced in $\sigma(t)$, and thus that $q = \sigma(u)$ satisfies $\mathbb{P}_n$.

5 Decomposing the semantics of terms

In the proofs to follow, we shall sometimes need to establish a correspondence between the behaviour of open terms and that of their closed instances. In detail, we are interested in the correspondence between a transition $\sigma(t) \xrightarrow{\mu} p$, for some term t, closed substitution σ, action μ, and process p, and the behaviour of t and that of $\sigma(x)$, for each variable x occurring in t. The simplest case is a direct application of the operational semantics in Table 1.

▶ **Lemma 6.** *For all terms t, t', substitution σ, and $\mu \in \mathcal{A}_\tau$, if $t \xrightarrow{\mu} t'$ then $\sigma(t) \xrightarrow{\mu} \sigma(t')$.*

Let us focus now on the role of variables. 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

Table 3 Inference rules for the transition relation $\xrightarrow{\ell}_{\rho}$ ($\mu \in \mathcal{A}_\tau$, $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$).

$$(a_1)\ \frac{}{x \xrightarrow{(x)}_{\mu} x_\mu} \qquad (a_2)\ \frac{t \xrightarrow{\ell}_{\rho} c}{t + u \xrightarrow{\ell}_{\rho} c} \qquad (a_3)\ \frac{t \xrightarrow{\ell}_{\rho} c}{t \parallel u \xrightarrow{\ell}_{\rho} c \parallel u}$$

$$(a_4)\ \frac{t \xrightarrow{(x)}_{\alpha} c \quad u \xrightarrow{(y)}_{\overline{\alpha}} c'}{t \parallel u \xrightarrow{(x,y)}_{\tau} c \parallel c'} \qquad (a_5)\ \frac{t \xrightarrow{(x)}_{\alpha} c \quad u \xrightarrow{\overline{\alpha}} u'}{t \parallel u \xrightarrow{(x)}_{\alpha,\tau} c \parallel u'} \qquad (a_6)\ \frac{t \xrightarrow{\alpha} t' \quad u \xrightarrow{(x)}_{\overline{\alpha}} c}{t \parallel u \xrightarrow{(x)}_{\overline{\alpha},\tau} t' \parallel c}$$

x. In this case, we say that x *triggers the behaviour* of t. To fully describe this situation, we introduce an auxiliary transition relation over open terms. The notion of *configuration* over terms, which stems from [8], will play an important role in their definition.

The presence of communication in CCS entails a complex definition of the semantics of configurations. In particular, it is necessary to introduce a fresh set of variables $\mathcal{V}_{\mathcal{A}_\tau} = \{x_\mu \mid x \in \mathcal{V}, \mu \in \mathcal{A}_\tau\}$, disjoint from $\mathcal{V}$, and terms. Intuitively, the symbol x_μ denotes that the closed term substituted for an occurrence of variable x has begun its execution (expressed in terms of a μ-action), and it contributes thus to triggering the behaviour of the term in which x occurs (see Example 8 below). Moreover, we also need to introduce special labels and subscripts for the auxiliary transitions over configurations, which will be of the form $c \xrightarrow{\ell}_{\rho} c'$. Briefly, the label ℓ is used to keep track of the variables that trigger the transition $c \xrightarrow{\ell}_{\rho} c'$. The subscript ρ, instead, will allow us to correctly define the semantics of communication: it will allow us to distinguish a τ-action directly performed by (the term substituted for) a variable x (transition $c \xrightarrow{(x)}_{\tau} c'$, with $\rho = \tau$), from a τ-action resulting from the communication of x with a subterm of the configuration (transition $c \xrightarrow{(x)}_{\alpha,\tau} c'$, with $\rho = \alpha, \tau$, where α is the action performed by the term substituted for x).

CCS configurations are defined over the set of variables $\mathcal{V}_{\mathcal{A}_\tau}$ and CCS terms.

▶ **Definition 7.** *The collection of CCS configurations, denoted by $\mathcal{C}$, is given by:*

$$c ::= x_\mu \quad | \quad t \quad | \quad c \parallel c \ , \qquad \textit{where } t \textit{ is a term, and } x_\mu \in \mathcal{V}_{\mathcal{A}_\tau}.$$

The auxiliary transitions of the form $\xrightarrow{\ell}_{\rho}$ are then formally defined via the inference rules in Table 3, where we omitted the symmetric rules to (a_2), (a_4), (a_5) and (a_6). We have that $\rho \in \mathcal{A}_\tau \cup ((\mathcal{A} \cup \overline{\mathcal{A}}) \times \{\tau\})$, whereas the label ℓ can be either of the form (x) or (x, y), for some variables $x, y \in \mathcal{V}$. Given a variable x and a label ℓ, we write $x \in \ell$ if x occurs in ℓ.

The distinguished variables x_μ allow us to keep track of which variable and action trigger the behaviour of the term, and they also allow us to present substitutions in an intuitive fashion. As explained in the following example, it is precisely because of substitutions (and communication) that we need to make the action μ explicit in x_μ.

▶ **Example 8.** Let $x \in \mathcal{V}$ and consider the term $x \parallel x$. By rules (a_1) and (a_4) in Table 3, we have that $x \parallel x \xrightarrow{(x,x)}_{\tau} x_\alpha \parallel x_{\overline{\alpha}}$ because $x \xrightarrow{(x)}_{\alpha} x_\alpha$ and $x \xrightarrow{(x)}_{\overline{\alpha}} x_{\overline{\alpha}}$. Hence, given any substitution σ such that $\sigma(x) \xrightarrow{\alpha} p_1$ and $\sigma(x) \xrightarrow{\overline{\alpha}} p_2$, for some terms p_1, p_2, we want to be able to correctly infer that $\sigma(x) \parallel \sigma(x) \xrightarrow{\tau} p_1 \parallel p_2$. Since the two occurrences of x, x_α and $x_{\overline{\alpha}}$, can be distinguished by the subscripts, the substitution $\sigma[x_\alpha \mapsto p_1, x_{\overline{\alpha}} \mapsto p_2](x_\alpha \| x_{\overline{\alpha}}) = p_1 \| p_2$ is well-defined. Without the subscripts, it would not have been possible to correctly define the substitution σ on the configuration c that is the target of $x \parallel x \xrightarrow{(x,x)}_{\tau} c$.

▶ **Lemma 9.** *Let t be term and σ be a closed substitution. Let $x, y \in \mathcal{V}$.*

1. *For any $\mu \in \mathcal{A}_\tau$, if $\sigma(x) \xrightarrow{\mu} p$, for some process p, and $t \xrightarrow{(x)}_{\mu} c$, for some configuration $c \in \mathcal{C}$, then $\sigma(t) \xrightarrow{\mu} \sigma[x_\mu \mapsto p](c)$.*
2. *For any $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$, if $\sigma(x) \xrightarrow{\alpha} p$, for some process p, and $t \xrightarrow{(x)}_{\alpha,\tau} c$, for some configuration $c \in \mathcal{C}$, then $\sigma(t) \xrightarrow{\tau} \sigma[x_\alpha \mapsto p](c)$.*
3. *For any $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$, if $\sigma(x) \xrightarrow{\alpha} p_x$, $\sigma(y) \xrightarrow{\overline{\alpha}} p_y$, for some processes p_x, p_y, and $t \xrightarrow{(x,y)}_{\tau} c \in \mathcal{C}$, for some configuration c, then $\sigma(t) \xrightarrow{\tau} \sigma[x_\alpha \mapsto p_x, y_{\overline{\alpha}} \mapsto p_y](c)$.*

Lemma 9 shows how the auxiliary transitions can be used to derive the behaviour of $\sigma(t)$ from those of the variables in t. We are now interested in analysing the converse situation: we show how a transition $\sigma(t) \xrightarrow{\mu} p$ can stem from transitions of the term t and of the process $\sigma(x)$, for $x \in \mathrm{var}(t)$. We limit ourselves to present the case of silent actions $\sigma(t) \xrightarrow{\tau} p$ as it requires a detailed analysis. The case of transitions labelled with observable actions is simpler and therefore omitted.

▶ **Lemma 10.** *Let t be a term, σ be a closed substitution, and p be a process. If $\sigma(t) \xrightarrow{\tau} p$, then one of the following holds:*

1. *There is a term t' s.t. $t \xrightarrow{\tau} t'$ and $\sigma(t') = p$.*
2. *There are a variable x, a process q, and a configuration c s.t. $\sigma(x) \xrightarrow{\tau} q$, $t \xrightarrow{(x)}_{\tau} c$, and $\sigma[x_\tau \mapsto q](c) = p$.*
3. *There are a variable x, a process q, and a configuration c s.t., for some $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$, $\sigma(x) \xrightarrow{\alpha} q$, $t \xrightarrow{(x)}_{\alpha,\tau} c$, and $\sigma[x_\alpha \mapsto q](c) = p$.*
4. *There are variables x, y, processes q_x, q_y and a configuration c s.t., for some $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$, $\sigma(x) \xrightarrow{\alpha} q_x$, $\sigma(y) \xrightarrow{\overline{\alpha}} q_y$, $t \xrightarrow{(x,y)}_{\tau} c$, and $\sigma[x_\alpha \mapsto q_x, y_{\overline{\alpha}} \mapsto q_y](c) = p$.*

6 Unique parallel decomposition

As explained in Section 4, our approach for establishing that $\mathbb{P}_n$ is invariant under equational proofs relies on processes having a unique parallel decomposition modulo $\sim_{\mathrm{BB}}$.

▶ **Definition 11** (Parallel decomposition modulo $\sim_{\mathrm{BB}}$). *A process p is* indecomposable *if $p \not\sim_{\mathrm{BB}} \mathbf{0}$ and $p \sim_{\mathrm{BB}} p_1 \| p_2$ implies $p_1 \sim_{\mathrm{BB}} \mathbf{0}$ or $p_2 \sim_{\mathrm{BB}} \mathbf{0}$, for all processes p_1 and p_2. A* parallel decomposition *of a process p is a finite multiset $\lbag p_1, \dots, p_k \rbag$ of indecomposable processes $p_1, \dots, p_k$ such that $p \sim_{\mathrm{BB}} p_1 \| \cdots \| p_k$. We say that p has a* unique parallel decomposition *if p has a parallel decomposition $\lbag p_1, \dots, p_k \rbag$ and for every other parallel decomposition $\lbag p'_1, \dots, p'_\ell \rbag$ of p there exists a bijection $f : \{1, \dots, k\} \to \{1, \dots, \ell\}$ such that $p_i \sim_{\mathrm{BB}} p'_{f(i)}$ for all $1 \leq i \leq k$.*

To prove that processes have a unique parallel decomposition we shall exploit a general result stating that a partial commutative monoid has unique decomposition if it can be endowed with a *weak decomposition order* that satisfies *power cancellation* [26]; we shall define and explain the notions below. Note that, in view of axioms P0–P2, which are (also) sound modulo $\sim_{\mathrm{BB}}$, the set of processes $\mathbf{P}$ modulo $\sim_{\mathrm{BB}}$ is a commutative monoid with respect to the binary operation naturally induced by $\|$ on $\sim_{\mathrm{BB}}$-equivalence classes and the $\sim_{\mathrm{BB}}$-equivalence class of $\mathbf{0}$ as identity element. We permit ourselves a minor abuse in notation and use $\to$ to (also) denote the binary relation $\{(p, q) \mid \exists \mu.\ p \xrightarrow{\mu} q\}$, and proceed to argue that $\to$ induces a weak decomposition order satisfying power cancellation on the commutative monoid of processes modulo $\sim_{\mathrm{BB}}$.

Given any process p and $n \geq 1$, let p^n denote the n-fold parallel composition $p \parallel p^{n-1}$, with $p^0 = \mathbf{0}$. We first state some properties of the reflexive-transitive closure $\rightarrow^*$ of $\rightarrow$:

▶ **Proposition 12.** *The relation $\rightarrow^*$ is an inversely well-founded partial order on processes satisfying the following properties:*

1. *For every process p there exists a process p' such that $p \rightarrow^* p' \sim_{\mathrm{BB}} \mathbf{0}$.*
2. *For all processes p, p' and q, if $p \rightarrow^* p'$, then $p \parallel q \rightarrow^* p' \parallel q$ and $q \parallel p \rightarrow^* q \parallel p'$.*
3. *For all processes p, q and r, if $p \parallel q \rightarrow^* r$, then there exist p' and q' such that $p \rightarrow^* p'$, $q \rightarrow^* q'$ and $r = p' \parallel q'$.*
4. *For all processes p and q, if $p \rightarrow^* q^n$ for all $n \in \mathbb{N}$, then $q \sim_{\mathrm{BB}} \mathbf{0}$.*

The following lemma is a direct consequence of the definition of branching bisimilarity.

▶ **Lemma 13.** *For all processes p, p' and q, if $p \sim_{\mathrm{BB}} q$ and $p \rightarrow^* p'$, then there exists q' such that $q \rightarrow^* q'$ and $p' \sim_{\mathrm{BB}} q'$.*

By this lemma we can define a binary relation $\preceq$ on $\mathbf{P}/{\sim_{\mathrm{BB}}}$, the set of $\sim_{\mathrm{BB}}$-equivalence classes of processes, by stating that $[p]_{\sim_{\mathrm{BB}}} \preceq [q]_{\sim_{\mathrm{BB}}}$ if, and only if, there exists $p' \in [p]_{\sim_{\mathrm{BB}}}$ such that $q \rightarrow^* p'$ (here $[p]_{\sim_{\mathrm{BB}}}$ and $[q]_{\sim_{\mathrm{BB}}}$ denote the $\sim_{\mathrm{BB}}$-equivalence classes of p and q, respectively). The following result is then a straightforward corollary of Proposition 12.

▶ **Corollary 14.** *The relation $\preceq$ is a weak decomposition order on $\mathbf{P}/{\sim_{\mathrm{BB}}}$, namely:*

1. *it is well-founded, i.e., every non-empty subset of $\mathbf{P}/{\sim_{\mathrm{BB}}}$ has a $\preceq$-minimal element;*
2. *the identity element $[\mathbf{0}]_{\sim_{\mathrm{BB}}}$ of $\mathbf{P}/{\sim_{\mathrm{BB}}}$ is the least element of $\mathbf{P}/{\sim_{\mathrm{BB}}}$ with respect to $\preceq$, i.e., $[\mathbf{0}]_{\sim_{\mathrm{BB}}} \preceq [p]_{\sim_{\mathrm{BB}}}$ for all $p \in \mathbf{P}$;*
3. *it is compatible, i.e., for all $p, q, r \in \mathbf{P}$ if $[p]_{\sim_{\mathrm{BB}}} \preceq [q]_{\sim_{\mathrm{BB}}}$, then $[p \parallel r]_{\sim_{\mathrm{BB}}} \preceq [q \parallel r]_{\sim_{\mathrm{BB}}}$;*
4. *it is precompositional, i.e., for all $p, q, r \in \mathbf{P}$ we have that $[p]_{\sim_{\mathrm{BB}}} \preceq [q \parallel r]_{\sim_{\mathrm{BB}}}$ implies $[p]_{\sim_{\mathrm{BB}}} = [q' \parallel r']_{\sim_{\mathrm{BB}}}$ for some $[q']_{\sim_{\mathrm{BB}}} \preceq [q]_{\sim_{\mathrm{BB}}}$ and $[r']_{\sim_{\mathrm{BB}}} \preceq [r]_{\sim_{\mathrm{BB}}}$; and*
5. *it is Archimedean, i.e., for all $p, q \in \mathbf{P}$ we have that $[p^n]_{\sim_{\mathrm{BB}}} \preceq [q]_{\sim_{\mathrm{BB}}}$ for all $n \in \mathbb{N}$ implies that $[p]_{\sim_{\mathrm{BB}}} = [\mathbf{0}]_{\sim_{\mathrm{BB}}}$.*

According to [26, Theorem 34] it now remains to prove that $\preceq$ satisfies power cancellation. The weak decomposition order $\preceq$ on the commutative monoid of processes modulo $\sim_{\mathrm{BB}}$ satisfies *power cancellation* if for every indecomposable process p and for all processes q and r such that $[p]_{\sim_{\mathrm{BB}}} \not\preceq [q]_{\sim_{\mathrm{BB}}}, [r]_{\sim_{\mathrm{BB}}}$, for all $k \in \mathbb{N}$, we have that $[p^k \parallel q]_{\sim_{\mathrm{BB}}} = [p^k \parallel r]_{\sim_{\mathrm{BB}}}$ implies $[q]_{\sim_{\mathrm{BB}}} = [r]_{\sim_{\mathrm{BB}}}$.

▶ **Proposition 15.** *The weak decomposition order $\preceq$ on the commutative monoid of processes modulo $\sim_{\mathrm{BB}}$ satisfies power cancellation.*

We have now established that $\preceq$ is a weak decomposition order on the commutative monoid of processes modulo $\sim_{\mathrm{BB}}$ that satisfies power cancellation. Thus, with an application of [26, Theorem 34] we get the following unique parallel decomposition result.

▶ **Proposition 16.** *Every process in $\mathbf{P}$ has a unique parallel decomposition.*

In what follows, we shall make use of the following direct consequence of Proposition 16.

▶ **Corollary 17.** *If $p \parallel r \sim_{\mathrm{BB}} q \parallel r$, then $p \sim_{\mathrm{BB}} q$.*

7 Nonexistence of a finite axiomatisation

We devote this section to proving Theorem 4. Following the strategy sketched in Section 4, we introduce a particular family of equations on which we will build our negative result:

$$p_n = \sum_{i=2}^{n} aa^{\leq i} \qquad (n \geq 2)$$

$$\mathfrak{e}_n: \quad a \parallel p_n \approx ap_n + \sum_{i=2}^{n} a(a \parallel a^{\leq i}) \qquad (n \geq 2).$$

It is easy to check that each equation $\mathfrak{e}_n$, for $n \geq 2$, is sound modulo rooted branching bisimilarity (as, in particular, it is sound modulo strong bisimilarity).

In order to prove Theorem 4, we proceed to show that no finite collection of equations over CCS that are sound modulo rooted branching bisimilarity can prove all of the equations $\mathfrak{e}_n$ $(n \geq 2)$ from the family given above. Formally, for each $n \geq 2$, we consider the property $\mathbb{P}_n$: *having a summand rooted branching bisimilar to* $a \parallel p_n$. Then, we prove the following:

▶ **Theorem 18.** *Let $\mathcal{E}$ be a finite axiom system over CCS that is sound modulo $\sim_{\mathtt{RBB}}$, let n be larger than the size of each term in the equations in $\mathcal{E}$, and let p, q be closed terms such that $p, q \sim_{\mathtt{RBB}} a \parallel p_n$. If $\mathcal{E} \vdash p \approx q$ and p satisfies $\mathbb{P}_n$ then so does q.*

The crucial step in the proof of Theorem 18 is delivered by the proposition below, which ensures that the property $\mathbb{P}_n$ $(n \geq 2)$ is preserved by the closure under substitutions of equations in a finite, sound axiom system. Proposition 20 is proved by means of the technical results provided so far, and the notion of **0**-*factor* of a term:

▶ **Definition 19.** *We say that a term t has a* **0** factor *if it contains a subterm of the form $t' \parallel t''$, and either $t' \sim_{\mathtt{RBB}} \mathbf{0}$ or $t'' \sim_{\mathtt{RBB}} \mathbf{0}$.*

▶ **Proposition 20.** *Let $t \approx u$ be an equation over CCS terms that is sound modulo $\sim_{\mathtt{RBB}}$. Let σ be a closed substitution with $p = \sigma(t)$ and $q = \sigma(u)$. Suppose that p and q have neither* **0** *summands nor* **0** *factors, and $p, q \sim_{\mathtt{RBB}} a \parallel p_n$ for some n larger than the sizes of t and u. If p satisfies $\mathbb{P}_n$, then so does q.*

Theorem 18 shows the property $\mathbb{P}_n$ to be an invariant under provability from finite sound axiom systems. As the left-hand side of equation $\mathfrak{e}_n$, i.e., the term $a \parallel p_n$, satisfies $\mathbb{P}_n$, whilst the right-hand side, i.e., the term $ap_n + \sum_{i=2}^{n} a(a \parallel a^{\leq i})$, does not, we can conclude that the infinite collection of equations $\mathfrak{e}_n$ $(n \geq 2)$ cannot be derived from any finite, sound axiom system. Hence, Theorem 4 follows.

8 Towards a positive result

We now proceed to study the role of the auxiliary operators *left merge* ($\llfloor$) and *communication merge* ($\mid$) from [12] in the axiomatisation of parallel composition modulo $\sim_{\mathtt{RBB}}$. We will show that by adding them to CCS we can obtain a complete axiomatisation of rooted branching bisimilarity over the new language. This axiomatisation is finite if so is $\mathcal{A}_\tau$.

We denote the language obtained by enriching CCS with $\llfloor$ and $\mid$ by $\mathrm{CCS}_{\mathrm{LC}}$:

$$t ::= \mathbf{0} \mid x \mid \mu.t \mid t + t \mid t \parallel t \mid t \llfloor t \mid t \mid t \enspace , \qquad (\mathrm{CCS}_{\mathrm{LC}})$$

where $x \in \mathcal{V}$, and $\mu \in \mathcal{A}_\tau$. The SOS rules for the $\mathrm{CCS}_{\mathrm{LC}}$ operators are given by the rules in Table 1 plus those reported in Table 4.

Table 4 Additional SOS rules for $\mathrm{CCS}_{\mathrm{LC}}$ operators ($\mu \in \mathcal{A}_\tau$, $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$).

$$\frac{t \xrightarrow{\mu} t'}{t \mathbin{\|\!\lfloor} u \xrightarrow{\mu} t' \parallel u} \qquad \frac{t \xrightarrow{\alpha} t' \quad u \xrightarrow{\overline{\alpha}} u'}{t \mid u \xrightarrow{\tau} t' \parallel u'}$$

To obtain the desired completeness result, we consider the axiom system $\mathcal{E}_{\mathsf{RBB}}$ (see Table 7 in Section 10), obtained by extending the complete axiom system for strong bisimilarity over $\mathrm{CCS}_{\mathrm{LC}}$ from [7] with axioms expressing the behaviour of $\mathbin{\|\!\lfloor}$ and $\mid$ in the presence of τ-actions (from [13]), and with the suitable τ-laws (from [20, 23]) necessary to deal with rooted branching bisimilarity. Then, we adjust the semantics of configurations given in Section 5 to the $\mathrm{CCS}_{\mathrm{LC}}$ setting, and we use it to extend the definition of rooted branching bisimilarity to open $\mathrm{CCS}_{\mathrm{LC}}$ terms (Definition 24). Usually, a behavioural equivalence $\sim$ is defined over processes and is then possibly extended to open terms by saying that $t \sim u$ iff $\sigma(t) \sim \sigma(u)$ for all closed substitutions σ. However, we adopt the same approach of, e.g., [10, 16, 29], and present the definition of $\sim_{\mathsf{RBB}}$ directly over configurations. We will show in Section 9 that the two approaches yield the same equivalence relation over terms (Theorem 25). Finally, we apply the strategy used in [10] to obtain the completeness of the axiomatisation of prefix iteration with silent moves modulo rooted branching bisimilarity:

1. We identify *normal forms* for $\mathrm{CCS}_{\mathrm{LC}}$ terms (Definition 27) and show that each term can be proven equal to a normal form using $\mathcal{E}_{\mathsf{RBB}}$ (Proposition 28).
2. We establish a relationship between $\sim_{\mathsf{BB}}$ and derivability in $\mathcal{E}_{\mathsf{RBB}}$ (Proposition 29).
3. We show that for all terms t, u, if $t \sim_{\mathsf{RBB}} u$, then $\mathcal{E}_{\mathsf{RBB}} \vdash t \approx u$ (Theorem 5).

9 Rooted branching bisimilarity over terms

In this section we discuss the decomposition of the semantics of $\mathrm{CCS}_{\mathrm{LC}}$ terms, and the extension of the definition of (rooted) branching bisimilarity to open $\mathrm{CCS}_{\mathrm{LC}}$ terms.

The first step towards our completeness result consists in providing a semantics for open $\mathrm{CCS}_{\mathrm{LC}}$ terms. To this end, we need to extend the semantics of configurations given in Section 5. For the sake of readability, we present the syntax of $\mathrm{CCS}_{\mathrm{LC}}$ configurations and the inference rules for variables and summations, even though they are identical to the corresponding ones presented in Section 5 for CCS. However, we omit the explanations on the roles of labels ℓ, ρ, and variables x_μ, as those can be found in Section 5. In particular, the use of variables $x_\mu \in \mathcal{V}_{\mathcal{A}_\tau}$ (as explained in Example 8) remains unchanged.

▶ **Definition 21** ($\mathrm{CCS}_{\mathrm{LC}}$ configuration). *The collection of CCS_{LC} configurations, denoted by $\mathcal{C}_{LC}$, is given by:*

$$c ::= x_\mu \quad \mid \quad t \quad \mid \quad c \parallel c \ , \qquad \textit{where } t \textit{ is a } \mathit{CCS}_{LC} \textit{ term, and } x_\mu \in \mathcal{V}_{\mathcal{A}_\tau}.$$

The auxiliary transitions of the form $\xrightarrow{\ell}_\rho$ are formally defined via the inference rules in Table 5, where we omitted the rules (a_1') and (a_2') for prefixing and choice (which are identical to, respectively, rules (a_1) and (a_2) in Table 3) the symmetric rules to (a_2'), (a_4'), (a_5') and (a_6'), as well as the rules for $\parallel$. We remark that Lemma 10 can be easily extended to $\mathrm{CCS}_{\mathrm{LC}}$ to show how a transition $\sigma(t) \xrightarrow{\mu} p$ can stem from transitions of the $\mathrm{CCS}_{\mathrm{LC}}$ term t and of the process $\sigma(x)$, for $x \in \mathrm{var}(t)$.

Since $\mathcal{V}_{\mathcal{A}_\tau}$ is disjoint from $\mathcal{V}$, we also need to introduce auxiliary rules for the special configuration $x_\mu \in \mathcal{V}_{\mathcal{A}_\tau}$. These are identified by a proper label x_μ on the transition and reported in Table 6 as rules (c_1) and (c_2). To conclude our analysis of the decomposition

Table 5 Inference rules for the transition relation $\xrightarrow{\ell}_{\rho}$ ($\mu \in \mathcal{A}_{\tau}$, $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$).

$$(a_3')\ \frac{t \xrightarrow{\ell}_{\rho} c}{t \mathbin{\lfloor\!\lfloor} u \xrightarrow{\ell}_{\rho} c \parallel u}$$

$$(a_4')\ \frac{t \xrightarrow{(x)}_{\alpha} c \quad u \xrightarrow{(y)}_{\overline{\alpha}} c'}{t \mid u \xrightarrow{(x,y)}_{\tau} c \parallel c'} \qquad (a_5')\ \frac{t \xrightarrow{(x)}_{\alpha} c \quad u \xrightarrow{\overline{\alpha}} u'}{t \mid u \xrightarrow{(x)}_{\alpha,\tau} c \parallel u'} \qquad (a_6')\ \frac{t \xrightarrow{\alpha} t' \quad u \xrightarrow{(x)}_{\overline{\alpha}} c}{t \mid u \xrightarrow{(x)}_{\overline{\alpha},\tau} t' \parallel c}$$

Table 6 Inference rules completing the operational semantics of $\mathrm{CCS_{LC}}$ configurations ($\mu \in \mathcal{A}_{\tau}$).

$$(c_1)\ \frac{}{x_{\mu} \xrightarrow{x_{\mu}} x_{\mu}} \qquad (c_2)\ \frac{c_1 \xrightarrow{x_{\mu}} c_1'}{c_1 \parallel c_2 \xrightarrow{x_{\mu}} c_1' \parallel c_2} \qquad (c_3)\ \frac{c_1 \xrightarrow{\mu} c_1'}{c_1 \parallel c_2 \xrightarrow{\mu} c_1' \parallel c_2} \qquad (c_4)\ \frac{c_1 \xrightarrow{\ell}_{\rho} c_1'}{c_1 \parallel c_2 \xrightarrow{\ell}_{\rho} c_1' \parallel c_2}$$

of the semantics of terms, we then need to extend the transition relations $\xrightarrow{\mu}$ and $\xrightarrow{\ell}_{\rho}$ to configurations. This is done by rules (c_3) and (c_4) in Table 6, where their symmetric counterparts have been omitted. Let $\overset{\xi}{\twoheadrightarrow}$ range over the possible transitions over configurations, i.e., $\overset{\xi}{\twoheadrightarrow}$ can be either $\xrightarrow{\mu}$, $\xrightarrow{\ell}_{\rho}$, or $\xrightarrow{x_{\mu}}$. The operational semantics of $\mathrm{CCS_{LC}}$ configurations is then given by the LTS whose states are configurations in $\mathcal{C}_{\mathrm{LC}}$, whose actions are in $\mathcal{A}_{\tau} \cup \mathcal{V} \cup \mathcal{V}_{\mathcal{A}_{\tau}}$, and whose transitions are those that are provable from the rules in Tables 1, 4, 5, and 6.

Following the same approach of, e.g. [10,16,29], we now present the definitions of branching and rooted branching bisimulation equivalences directly over configurations.

▶ **Definition 22** (Branching bisimulation over configurations). *A symmetric relation $\mathcal{R}$ over $\mathcal{C}_{LC}$ is a* branching bisimulation *iff whenever $c_1 \,\mathcal{R}\, c_2$, if $c_1 \overset{\xi}{\twoheadrightarrow} c_1'$ then:*

- *either $\overset{\xi}{\twoheadrightarrow} = \xrightarrow{\tau}$ and $c_1' \,\mathcal{R}\, c_2$,*
- *or $c_2 \xrightarrow{\varepsilon} c_2'' \overset{\xi}{\twoheadrightarrow} c_2'$ for some c_2'', c_2' such that $c_1 \,\mathcal{R}\, c_2''$ and $c_1' \,\mathcal{R}\, c_2'$.*

Two configurations c_1, c_2 are branching bisimilar*, denoted by $c_1 \sim_{\mathtt{BB}} c_2$, iff there exists a branching bisimulation $\mathcal{R}$ such that $c_1 \,\mathcal{R}\, c_2$.*

The definition of $\sim_{\mathtt{BB}}$ given in Definition 22 yields the same equivalence relation over configurations that we would have obtained with the standard approach, i.e., by defining $c_1 \sim_{\mathtt{BB}} c_2$ iff $\sigma(c_1) \sim_{\mathtt{BB}} \sigma(c_2)$ for all closed substitutions σ.

▶ **Theorem 23.** *For all configurations $c_1, c_2 \in \mathcal{C}_{LC}$ it holds that $c_1 \sim_{\mathtt{BB}} c_2$ iff $\sigma(c_1) \sim_{\mathtt{BB}} \sigma(c_2)$ for all closed substitutions σ.*

The approach for $\sim_{\mathtt{BB}}$ can be extended in a straightforward manner to $\sim_{\mathtt{RBB}}$.

▶ **Definition 24** (Rooted branching bisimilarity over configurations). *Let $c_1, c_2 \in \mathcal{C}_{LC}$. We say that c_1 and c_2 are* rooted branching bisimilar*, denoted by $c_1 \sim_{\mathtt{RBB}} c_2$, iff:*

- *if $c_1 \overset{\xi}{\twoheadrightarrow} c_1'$ then $c_2 \overset{\xi}{\twoheadrightarrow} c_2'$ for some c_2' such that $c_1' \sim_{\mathtt{BB}} c_2'$;*
- *if $c_2 \overset{\xi}{\twoheadrightarrow} c_2'$ then $c_1 \overset{\xi}{\twoheadrightarrow} c_1'$ for some c_1' such that $c_1' \sim_{\mathtt{BB}} c_2'$.*

▶ **Theorem 25.** *For all $c_1, c_2 \in \mathcal{C}_{LC}$ it holds that $c_1 \sim_{\mathtt{RBB}} c_2$ iff $\sigma(c_1) \sim_{\mathtt{RBB}} \sigma(c_2)$ for all closed substitutions σ.*

Table 7 Equational basis modulo rooted branching bisimilarity.

Equational basis modulo strong bisimilarity: $\mathcal{E}_{\mathrm{B}}$			
A0	$x + \mathbf{0} \approx x$	C0	$\mathbf{0} \mid x \approx \mathbf{0}$
A1	$x + y \approx y + x$	C1	$x \mid y \approx y \mid x$
A2	$(x + y) + z \approx x + (y + z)$	C2	$(x \mid y) \mid z \approx x \mid (y \mid z)$
A3	$x + x \approx x$	C3	$(x + y) \mid z \approx x \mid z + y \mid z$
		C4	$\alpha x \mid \beta y \approx \tau(x \parallel y) \quad \text{if } \alpha = \overline{\beta}$
L0	$\mathbf{0} \mathbin{\|\!\!\_} x \approx \mathbf{0}$	C5	$\alpha x \mid \beta y \approx \mathbf{0} \quad \text{if } \alpha \neq \overline{\beta}$
L1	$\mu x \mathbin{\|\!\!\_} y \approx \mu(x \parallel y)$	C6	$(x \mathbin{\|\!\!\_} y) \mid z \approx (x \mid z) \mathbin{\|\!\!\_} y$
L2	$(x \mathbin{\|\!\!\_} y) \mathbin{\|\!\!\_} z \approx x \mathbin{\|\!\!\_} (y \parallel z)$	C7	$x \mid y \mid z \approx \mathbf{0}$
L3	$x \mathbin{\|\!\!\_} \mathbf{0} \approx x$		
L4	$(x + y) \mathbin{\|\!\!\_} z \approx x \mathbin{\|\!\!\_} z + y \mathbin{\|\!\!\_} z$	P	$x \parallel y \approx x \mathbin{\|\!\!\_} y + y \mathbin{\|\!\!\_} x + x \mid y$
Additional axioms for $\sim_{\mathrm{RBB}}$: $\mathcal{E}_{\mathrm{RBB}} = \mathcal{E}_{\mathrm{B}} \cup \{TB, TL\}$			
TB	$\mu(\tau(x + y) + y) \approx \mu(x + y)$	TL	$x \mathbin{\|\!\!\_} \tau y \approx x \mathbin{\|\!\!\_} y$
Derivable axioms			
D1	$x \parallel y \approx y \parallel x$	DT1	$\mu \tau x \approx \mu x$
D2	$(x \parallel y) \parallel z \approx x \parallel (y \parallel z)$	DT2	$x \mathbin{\|\!\!\_} (\tau(y + z) + y) \approx x \mathbin{\|\!\!\_} (y + z)$
D3	$(x \mathbin{\|\!\!\_} y) \mid (z \mathbin{\|\!\!\_} w) \approx (x \mid z) \mathbin{\|\!\!\_} (y \parallel w)$	DT3	$\tau x \mid y \approx \mathbf{0}$
D4	$x \parallel \mathbf{0} \approx x$		

10 The equational basis

We now present the complete axiomatisation for rooted branching bisimilarity over $\mathrm{CCS}_{\mathrm{LC}}$.

In [20] it was proved that if we consider the fragment BCCS of CCS (i.e., the fragment consisting only of $\mathbf{0}$, variables, prefixing, and choice), then a ground-complete axiomatisation of rooted branching bisimilarity over BCCS is given by $\mathcal{E}_0 \cup \{\mathrm{TB}\}$, where $\mathcal{E}_0 = \{\mathrm{A0,A1,A2,A3}\}$ from Table 2 (also reported in Table 7), and axiom TB is in Table 7. Informally, TB reflects that if executing a τ-step does not discard any observable behaviour, then it is redundant. In [7] it was proved that the axiom system $\mathcal{E}_{\mathrm{B}}$ given in Table 7, is a complete axiomatisation of bisimilarity over $\mathrm{CCS}_{\mathrm{LC}}$. Starting from these works, we now study a complete axiomatisation for $\sim_{\mathrm{RBB}}$. Our aim is to show that the axiom system $\mathcal{E}_{\mathrm{RBB}} = \mathcal{E}_{\mathrm{B}} \cup \{\mathrm{TB,TL}\}$ presented in Table 7 is a *complete axiomatisation of rooted branching bisimilarity* over $\mathrm{CCS}_{\mathrm{LC}}$.

If executing a τ-move does not resolve a choice within a parallel component, then it will also not resolve a choice of the parallel composition; axiom TL expresses a similar property of rooted branching bisimilarity for left merge. Interestingly, by combining TL and TB, it is possible to derive, as shown below, equation DT2 in Table 7, which is the equation for the left merge corresponding to TB.

$$x \mathbin{\|\!\!\_} (\tau(y + z) + y) \overset{(\mathrm{TL})}{\approx} x \mathbin{\|\!\!\_} \tau(\tau(y + z) + y) \overset{(\mathrm{TB})}{\approx} x \mathbin{\|\!\!\_} \tau(y + z) \overset{(\mathrm{TL})}{\approx} x \mathbin{\|\!\!\_} (y + z).$$

In Table 7 we report also some other equations that can be derived from $\mathcal{E}_{\mathrm{RBB}}$, and that are useful in the technical development of our results. We refer the reader interested in the derivation proofs of D1–D3 and DT3 to [7]. Notice that DT1 corresponds essentially to the substitution instance of TB in which y is mapped to $\mathbf{0}$.

First of all, it is immediate to prove the soundness of $\mathcal{E}_{\mathtt{RBB}}$ modulo $\sim_{\mathtt{RBB}}$.

▶ **Theorem 26** (Soundness). *The axiom system $\mathcal{E}_{\mathtt{RBB}}$ is sound modulo $\sim_{\mathtt{RBB}}$ over CCS_{LC}.*

To obtain the desired completeness result, we apply the same strategy used in [10] that consists in the three steps discussed in Section 8.

Let us proceed to the first step: identifying normal forms for $\mathrm{CCS}_{\mathrm{LC}}$ terms.

▶ **Definition 27** (Normal forms). *The set of normal forms over CCS_{LC} is generated by the following grammar:*

$$\begin{aligned} S &::= \mu.N \quad | \quad x \mathbin{\|\!\_} N \quad | \quad (x \mid \alpha) \mathbin{\|\!\_} N \quad | \quad (x \mid y) \mathbin{\|\!\_} N \\ N &::= \mathbf{0} \quad | \quad S \quad | \quad N + N \end{aligned}$$

where $x, y \in \mathcal{V}$, $\mu \in \mathcal{A}_\tau$ and $\alpha \in \mathcal{A} \cup \overline{\mathcal{A}}$. Normal forms generated by S are also called simple normal forms *and are characterised by the fact that they do not have $+$ as head operator.*

▶ **Proposition 28.** *For every term t there is a normal form N such that $\mathcal{E}_{\mathtt{RBB}} \vdash t \approx N$.*

We can then proceed to prove that branching bisimilar terms can be proven equal to rooted branching bisimilar terms using the axiom system $\mathcal{E}_{\mathtt{RBB}}$.

▶ **Proposition 29.** *For CCS_{LC} terms t, u, if $t \sim_{\mathtt{BB}} u$ then $\mathcal{E}_{\mathtt{RBB}} \vdash \mu.t \approx \mu.u$, for any $\mu \in \mathcal{A}_\tau$.*

The completeness of the axiom system $\mathcal{E}_{\mathtt{RBB}}$ then follows from Proposition 28 and Proposition 29. Notice that axioms L1 and TB are actually axiom schemata that both generate $|\mathcal{A}_\tau|$ axioms. Similarly, the schema C4 generates $2|\mathcal{A}|$ axioms, and C5 generates $2|\mathcal{A}| \times (2|\mathcal{A}| - 1)$ axioms. Hence, $\mathcal{E}_{\mathtt{RBB}}$ is *finite* when so is the set of actions.

▶ **Theorem 5** (Completeness). *Let t, u be CCS_{LC} terms. If $t \sim_{\mathtt{RBB}} u$, then $\mathcal{E}_{\mathtt{RBB}} \vdash t \approx u$.*

11 Concluding remarks

In this paper we have shown that the use of auxiliary operators, such as the left merge and communication merge, is crucial to obtain a finite, complete axiomatisation of the CCS parallel composition operator modulo rooted branching bisimilarity. Indeed, rooted branching bisimilarity does not afford a finite, complete axiomatisation over CCS without the auxiliary operators (our negative result), whereas CCS with the auxiliary operators added does have such a finite complete axiomatisation modulo rooted branching bisimilarity (our positive result).

A natural direction for future research is the extension of our results to other weak congruences from the spectrum [17]. The infinite family of equations used in the proof of our negative result (Theorem 4) is the same as that used by Moller to prove that CCS does not afford a finite complete axiomatisation of strong bisimilarity [32]. Our proof that the parametric property $\mathbb{P}_n$ is preserved by provability from every collection of equations that are bounded in size by n and that are sound with respect to rooted branching bisimilarity refines Moller's proof that $\mathbb{P}_n$ is preserved by provability if the equations are required to be sound with respect to strong bisimilarity. Our next goal will be to identify the weakest congruence $\sim$ in the spectrum that includes strong bisimilarity and for which provability from a collection of sound equations that are sound with respect to $\sim$ preserves $\mathbb{P}_n$. It will then follow that CCS does not afford a finite complete axiomatisation for all congruences including strong bisimilarity and included in $\sim$.

Regarding extensions of the positive result, we will focus on three weak congruences, namely *rooted η-bisimilarity* ($\sim_{\mathtt{R\eta B}}$), *rooted delay bisimilarity* ($\sim_{\mathtt{RDB}}$), and *rooted weak bisimilarity* ($\sim_{\mathtt{RWB}}$), and provide complete axiomatisations for them. We are confident that the axiomatisation for $\sim_{\mathtt{R\eta B}}$ can be obtained by exploiting a proof technique from [10] based on the notion of *saturation*. It should then be established that $\sim_{\mathtt{R\eta B}}$ coincides with $\sim_{\mathtt{RBB}}$ on the class of η-saturated terms. Hence, if we can show that each term is provably equal to an η-saturated term using the axiom system for $\sim_{\mathtt{R\eta B}}$, the completeness of the considered axiom system then directly follows from that for $\sim_{\mathtt{RBB}}$ we provided in this paper.

The quest for complete axiomatisations for $\sim_{\mathtt{RDB}}$ and $\sim_{\mathtt{RWB}}$ will require a different approach, as these equivalences are not preserved by the communication merge operator. For instance, we have that $\tau.a \sim_{\mathtt{RWB}} \tau.a + a$, but $\tau.a \mid \overline{a}.b \not\sim_{\mathtt{RWB}} (\tau.a + a) \mid \overline{a}.b$. Regarding $\sim_{\mathtt{RDB}}$, similar observations can be made (see [18] for more details). The complete axiomatisation for observational congruence [23] (and thus rooted weak bisimilarity) over ACP$_\tau$ presented in [13] includes the axiom

$$\tau.x \mid y \approx x \mid y. \tag{TC}$$

Similarly, in [1, 21] it was argued that in order to reason compositionally, and obtain an equational theory of CCS modulo observational congruence, it is necessary to define the operational semantics of communication merge in terms of inference rules of the form

$$\frac{t \overset{\alpha}{\Longrightarrow} t' \quad u \overset{\overline{\alpha}}{\Longrightarrow} u'}{t \mid u \overset{\tau}{\Longrightarrow} t' \parallel u'}$$

where we use $\overset{\mu}{\Longrightarrow}$ as a short-hand for the sequence of transitions $\xrightarrow{\varepsilon}\xrightarrow{\mu}\xrightarrow{\varepsilon}$. This means that in order for $\mid$ to preserve $\sim_{\mathtt{RWB}}$ (and/or $\sim_{\mathtt{RDB}}$), we need to consider a sequence of weak transitions as a single step. Clearly, since $\mid$ is an auxiliary operator that we introduce specifically to obtain finite axiomatisations, its semantics can be defined in the most suitable way for our purposes, i.e., so that it is consistent with the considered congruence relation. However, it is also clear that if we modify the semantics of one operator in CCS$_{\mathrm{LC}}$, then we are working with a new language. In particular, some axioms that are sound modulo *strong* bisimilarity (and thus also modulo $\sim_{\mathtt{RBB}}$) over CCS$_{\mathrm{LC}}$ become unsound modulo rooted weak bisimilarity over the new language: this is the case of axioms C6 and C7 in Table 7. As a consequence, we cannot exploit the completeness of the axiomatisation for rooted branching bisimilarity to derive complete axiomatisations for rooted weak bisimilarity and rooted delay bisimilarity, but we must provide new axiomatisations for them and prove their completeness from scratch. Hence, we leave as future work the quest for complete axiomatisations for $\sim_{\mathtt{RWB}}$ and $\sim_{\mathtt{RDB}}$ over (recursion, relabelling, and restriction free) CCS with left merge and communication merge.

References

1 Luca Aceto. On “Axiomatising Finite Concurrent Processes”. *SIAM J. Comput.*, 23(4):852–863, 1994. doi:10.1137/S0097539793243600.

2 Luca Aceto, Elli Anastasiadi, Valentina Castiglioni, Anna Ingólfsdóttir, and Bas Luttik. In search of lost time: Axiomatising parallel composition in process algebras. In *Proceedings of LICS 2021*, pages 1–14. IEEE, 2021. doi:10.1109/LICS52264.2021.9470526.

3 Luca Aceto, Valentina Castiglioni, Wan Fokkink, Anna Ingólfsdóttir, and Bas Luttik. Are two binary operators necessary to finitely axiomatise parallel composition? In *Proceedings of CSL 2021*, volume 183 of *LIPIcs*, pages 8:1–8:17, 2021. doi:10.4230/LIPIcs.CSL.2021.8.

4 Luca Aceto, Valentina Castiglioni, Anna Ingolfsdottir, and Bas Luttik. On the axiomatisation of branching bisimulation congruence over ccs, 2022. doi:10.48550/ARXIV.2206.13927.

5 Luca Aceto, Valentina Castiglioni, Anna Ingólfsdóttir, Bas Luttik, and Mathias R. Pedersen. On the axiomatisability of parallel composition: A journey in the spectrum. In *Proceedings of CONCUR 2020*, volume 171 of *LIPIcs*, pages 18:1–18:22, 2020. doi:10.4230/LIPIcs.CONCUR.2020.18.

6 Luca Aceto, David de Frutos-Escrig, Carlos Gregorio-Rodríguez, and Anna Ingólfsdóttir. Axiomatizing weak simulation semantics over BCCSP. *Theor. Comput. Sci.*, 537:42–71, 2014. doi:10.1016/j.tcs.2013.03.013.

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

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

9 Luca Aceto, Anna Ingólfsdóttir, Bas Luttik, and Paul van Tilburg. Finite equational bases for fragments of CCS with restriction and relabelling. In *Proceedings of IFIP TCS 2008*, volume 273 of *IFIP*, pages 317–332, 2008. doi:10.1007/978-0-387-09680-3_22.

10 Luca Aceto, Rob J. van Glabbeek, Wan Fokkink, and Anna Ingólfsdóttir. Axiomatizing prefix iteration with silent steps. *Inf. Comput.*, 127(1):26–40, 1996. doi:10.1006/inco.1996.0047.

11 Twan Basten. Branching bisimilarity is an equivalence indeed! *Inf. Process. Lett.*, 58(3):141–147, 1996. doi:10.1016/0020-0190(96)00034-8.

12 Jan A. Bergstra and Jan Willem Klop. Process algebra for synchronous communication. *Information and Control*, 60(1-3):109–137, 1984. doi:10.1016/S0019-9958(84)80025-X.

13 Jan A. Bergstra and Jan Willem Klop. Algebra of communicating processes with abstraction. *Theor. Comput. Sci.*, 37:77–121, 1985. doi:10.1016/0304-3975(85)90088-X.

14 Taolue Chen, Wan Fokkink, and Rob J. van Glabbeek. Ready to preorder: The case of weak process semantics. *Inf. Process. Lett.*, 109(2):104–111, 2008. doi:10.1016/j.ipl.2008.09.003.

15 Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *Proceedings of CONCUR '90*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297, 1990. doi:10.1007/BFb0039066.

16 Rob J. van Glabbeek. A complete axiomatization for branching bisimulation congruence of finite-state behaviours. In *Proceedings of MFCS'93*, volume 711 of *Lecture Notes in Computer Science*, pages 473–484, 1993. doi:10.1007/3-540-57182-5_39.

17 Rob J. van Glabbeek. The linear time - branching time spectrum II. In *Proceedings of CONCUR'93*, volume 715 of *Lecture Notes in Computer Science*, pages 66–81, 1993. doi:10.1007/3-540-57208-2_6.

18 Rob J. van Glabbeek. On cool congruence formats for weak bisimulations. *Theor. Comput. Sci.*, 412(28):3283–3302, 2011. doi:10.1016/j.tcs.2011.02.036.

19 Rob J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In *IFIP Congress*, pages 613–618, 1989.

20 Rob J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics. *J. ACM*, 43(3):555–600, 1996. doi:10.1145/233551.233556.

21 Matthew Hennessy. Axiomatising finite concurrent processes. *SIAM J. Comput.*, 17(5):997–1017, 1988. doi:10.1137/0217063.

22 Matthew Hennessy and Robin Milner. On observing nondeterminism and concurrency. In *Proceedings of ICALP 1980*, volume 85 of *Lecture Notes in Computer Science*, pages 299–309, 1980. doi:10.1007/3-540-10003-2_79.

23 Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985. doi:10.1145/2455.2460.

24 Tony Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

25 Robert M. Keller. Formal verification of parallel programs. *Commun. ACM*, 19(7):371–384, 1976. doi:10.1145/360248.360251.

26 Bas Luttik. Unique parallel decomposition in branching and weak bisimulation semantics. *Theor. Comput. Sci.*, 612:29–44, 2016. doi:10.1016/j.tcs.2015.10.013.

27 Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980. doi:10.1007/3-540-10235-3.

28 Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.

29 Robin Milner. A complete axiomatisation for observational congruence of finite-state behaviors. *Inf. Comput.*, 81(2):227–247, 1989. doi:10.1016/0890-5401(89)90070-9.

30 Faron Moller. *Axioms for Concurrency*. PhD thesis, Department of Computer Science, University of Edinburgh, July 1989. Report CST-59-89. Also published as ECS-LFCS-89-84.

31 Faron Moller. The importance of the left merge operator in process algebras. In *Proceedings of ICALP ‘90*, volume 443 of *Lecture Notes in Computer Science*, pages 752–764, 1990. doi:10.1007/BFb0032072.

32 Faron Moller. The nonexistence of finite axiomatisations for CCS congruences. In *Proceedings of LICS ’90*, pages 142–153, 1990. doi:10.1109/LICS.1990.113741.

33 Rocco De Nicola and Matthew Hennessy. Testing equivalence for processes. In *Proceedings of ICALP 1983*, volume 154 of *Lecture Notes in Computer Science*, pages 548–560, 1983. doi:10.1007/BFb0036936.

34 David M. R. Park. Concurrency and automata on infinite sequences. In *Proceedings of GI-Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183, 1981. doi:10.1007/BFb0017309.

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

36 Gordon D. Plotkin. A structural approach to operational semantics. *J. Log. Algebraic Methods Program.*, 60-61:17–139, 2004.