RAIRO-Theor. Inf. Appl. 60 (2026) 1
https://doi.org/10.1051/ita/2025021

RAIRO - Theoretical Informatics and Applications
www.rairo-ita.org

ON BISIMULATION IN ABSENCE OF RESTRICTION

Xian Xu*

Abstract. We revisit the standard bisimulation equalities in process models free of the restriction operator. As is well-known, in general the weak bisimilarity is coarser than the strong bisimilarity because it abstracts from internal actions. In absence of restriction, those internal actions become somewhat visible, so one might wonder if the weak bisimilarity is still 'weak'. We show that in CCScore (*i.e.*, Milner's standard CCS without τ-prefix, summation and relabelling) the weak bisimilarity indeed remains weak, *i.e.*, still strictly coarser than the strong bisimilarity, even without the restriction operator. Essentially, this is due to the existence of the replication operation, which can keep a process retaining its state (*i.e.*, the capacity of interaction). By virtue of these observations, we examine a variant of the weak bisimilarity, called quasi-strong bisimilarity. This quasi-strong bisimilarity requires the matching of internal actions to be conducted in the strong manner, as for the strong bisimilarity, and the matching of visible actions to have no trailing internal actions. We exhibit that in CCScore without the restriction operator, the weak bisimilarity exactly collapses onto this quasi-strong bisimilarity, which is moreover shown to coincide with the branching bisimilarity. These results reveal that in absence of the restriction operation, some ingredient of the weak bisimilarity indeed turns into strong, particularly the matching of internal actions.

Mathematics Subject Classification. 68Q85.

Received October 3, 2023. Accepted December 3, 2025.

1. Introduction

Process models study the behaviour of concurrent systems, particularly their equivalence or degree of similarity [1, 2]. Bisimulation equality, called bisimilarity, is the most exploited notion of such equivalence [3, 4]. For two concurrent systems, a strong bisimulation requires each action, whether external (*i.e.*, visible) or internal (*i.e.*, invisible), of one system to be precisely matched by the other. In contrast, a weak bisimulation, as its name suggests, allows the bisimulation to be observational. Namely, an external action of one system can be matched by the same action of the other, possibly mingled with some internal actions. As is well-known, the weak bisimulation equality is usually coarser than the strong bisimulation equality, because the former can hide some computation inside the system (*e.g.*, implementation). A typical way to achieve such hiding is by the restriction operator, sometimes called the localization operator, which literally has the effect of concealing a port/channel name from being discovered. For example, in the language of pure CCS [3], the following process M has three concurrent components P, Q, R connected by the operation of parallel composition ($|$), and the

Keywords and phrases: Strong bisimulation, weak bisimulation, restriction, first-order, higher-order, processes.

East China University of Science and Technology, Shanghai 200237, China.

* Corresponding author: xuxian@ecust.edu.cn

components P, Q share a restricted (or local) name m, represented by the restriction operation (m), that can be used to keep R from knowing or sensing something happening between P and Q, virtually forming a subsystem.

$$M \stackrel{\text{def}}{=} (m)(P \,|\, Q) \,|\, R$$

Restriction is a frequent and powerful operator in process models. Intensionally, it can hide from outside world the critical information, and extensionally, it facilitates internal or silent movement (possibly forced synchronization) so that the intrinsic implementation details are transparent to observers. From the viewpoint of computation, it provides a recourse to Turing completeness as well as interactional completeness in the measurement of computability [2, 5]. Concomitantly, some undecidability issues come along with the high computability, *e.g.*, undecidability of bisimulations, making it sometimes succumb to efficient use in practice. Therefore, it is sometimes tempting to work without the restriction operator. Although this may bring about certain decrease in expressiveness, the resulting model can avoid being too powerful to be tractable, and in effect be advantageous for practical applications on top of the model. Moreover, fortuitously some models without the restriction operator still turn out to be computationally complete, *e.g.*, higher-order processes [2, 6]. In a notable work, Hirschkoff *et al.* [7] study a subcalculus of the pi-calculus without the restriction and the choice operations but featuring a special top-level replication. The focus of that work is to provide a new congruence result for that subcalculus, by means of a syntactic characterisation of the (strong) bisimilarity. In this work, by contrast, we are interested in the relationship between the strong and weak bisimilarities in absence of the restriction opertation.

It is natural to consider this: If the restriction operator is removed (so one loses the power of hiding), would the weak bisimilarity still be coarser than the strong bisimilarity? Or would the 'gap' between the weak and strong bisimilarities become less pronounced? In this work, we look into this question and provide some answers. For our purpose, we take CCScore and its higher-order variant (named HOCCScore) as the test bed. CCScore (respectively HOCCScore) represents the standard CCS by Milner [3] (respectively plain CHOCS by Thomsen [8]) with the basic first-order (respectively higher-order) concurrency formalism, and without the τ-prefix, summation and relabelling. In turn, CCS$^-$ (respectively HOCCS$^-$) denotes CCScore (respectively HOCCScore) free of the restriction operation. It is worth noting that CCS$^-$ and HOCCS$^-$ still admit the replication operation (in the latter, it is a derived operation weaker than the original primitive one), otherwise these models would be far less interesting. We choose these models because they contain the interesting minimal part of the original models suitable (and non-trivial) for our study. Though useful, summation and relabelling are not essential for our work here. In particular, having the τ-prefix would actually defeat the purpose of this work, since it can be deemed as an operation derivable from the restriction operation. To see this, think of the τ-prefix $\tau.P$ as $(c)(c.P \,|\, \bar{c}.0)$ where $(c)Q, c.Q, \bar{c}.Q$ denote the standard CCS restriction, input, and output, respectively.

The main goal of this paper is to examine the bisimulation equalities in calculi free of restriction. We are interested in the relationship between the weak bisimilarity and the strong bisimilarity, particularly in absence of the restriction operation. Importantly, the lack of the restriction operator makes the interaction completely exposed to the environment, so that two weakly bisimilar processes may be forced to behave in the manner of strong bisimilarity, from certain perspective. Moreover, the situation can further vary when it comes to the style of interaction, say first-order synchronization (*i.e.*, implicit name-passing) or higher-order process-passing.

Contribution. We demonstrate that in CCS$^-$, the weak bisimilarity does somehow collapse onto a bisimulation equality called quasi-strong bisimilarity. In particular, this quasi-strong bisimilarity requests strong bisimulation on internal moves, and almost the same as the weak bisimilarity does for external actions. By 'almost', we mean that in the matching of an external (*i.e.*, visible) action α, one can make a few internal (τ) actions before α but none after it, *i.e.*, $\Longrightarrow\xrightarrow{\alpha}$ in standard notation (also known as the 'delay' transition [2]). The quasi-strong bisimilarity, as it appears, strengthens the weak bisimilarity, and moves toward the strong bisimilarity. Moreover, as a corollary, we show that the quasi-strong bisimilarity actually coincides with the branching bisimilarity, which in turn implies the coincidence between the weak bisimilarity and the branching bisimilarity.

Specifically, we prove in detail that in CCS$^-$, the weak bisimilarity can indeed be tightened, to be coincident with the quasi-strong bisimilarity. That being said, there appears to be still some distance from the strong bisimilarity. This is essentially attributed to the replication operator, which can introduce infinity that in turn generates some kind of state-preserving behaviour (if a process maintains all its interactional capability after doing some action, we refer to that action as state-preserving; otherwise it is state-changing). Consequently, to some extent, replication plays a role in realizing immutability in a concurrent system even without the restriction operator.

In contrast, in HOCCS$^-$ the relationship between the weak bisimilarity and the strong bisimilarity becomes much more complicated with the removal of the restriction. The analytical method of CCS$^-$cannot be used for HOCCS$^-$, mainly because the encoded replication is not quite the same as the original (primitive) one (*i.e.*, it reveals some visible actions other than those from the process to be replicated). To this end, we conjecture that the weak context bisimilarity collapses onto the strong context bisimilarity in HOCCS$^-$ (here the context bisimilarity is the standard bisimulation equality in higher-order process models [2]), and leave the details to further study. Intuitively, this situation intrinsically results from the complexity of process-passing.

This work can potentially help to precisely identify the boundaries between different notions of bisimulations, while clarifying further the features of the considered process models and the properties of the processes that are classified (differently) by the different bisimulations. In the theoretical regard, the results of this work give evidence and lend confidence to related studies of concurrent models concerning the relationship between the weak and strong bisimilarities in a setting free of the restriction operator. With regard to application, the results of this work can hopefully help in choosing a suitable bisimulation (for instance, a less demanding one with clauses easy to handle) when it comes to practical scenarios, such as the modeling and verification of complex concurrent systems (AI systems, blockchains, *etc.*). Also, the technical approach for these results might be of independent interest, since it provides some analytical tools for understanding restriction-free processes in the concurrent setting, and can potentially be applied in other similar process models.

Organization

The remainder of the paper is organized as follows. Section 2 tackles the bisimilarity in CCS$^-$. Section 3 discusses the situation for HOCCS$^-$. In both sections, we first define the syntax and semantics of the corresponding calculus, and then present the results about the bisimilarities. Section 4 concludes the paper and points to some future work.

2. On the bisimilation equality in CCS$^-$

In this section, we first define CCS$^-$, *i.e.*, CCScore without the restriction operation. Then we discuss the relationship between the strong and weak bisimilarities. In particular, we show that the strong bisimilarity is still strictly finer than the weak bisimilarity. However, the distance between the strong bisimilarity and the weak bisimilarity can be somewhat shortened. This is evidenced by the so-called quasi-strong bisimilarity, which requests more than the weak bisimilarity and moves closer to the strong bisimilarity, but still turns out to be coincident with the weak bisimilarity.

2.1. Calculus CCS$^-$

The syntax of CCS$^-$ is given as follows. We use capital letters to stand for processes.

$$P, Q := 0 \;\Big|\; a.P \;\Big|\; \overline{a}.P \;\Big|\; P \,|\, Q \;\Big|\; !P$$

The operational semantics is also standard and presented below (we skip the symmetric rules).

$$\frac{}{a.P \xrightarrow{a} P} \qquad \frac{}{\overline{a}.P \xrightarrow{\overline{a}} P}$$

$$\frac{P \xrightarrow{\overline{a}} P' \qquad Q \xrightarrow{a} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'} \qquad \frac{P \xrightarrow{\gamma} P'}{P \,|\, Q \xrightarrow{\gamma} P' \,|\, Q}$$

$$\frac{P \xrightarrow{\gamma} P'}{!P \xrightarrow{\gamma} P' \,|\, !P} \qquad \frac{P \xrightarrow{a} P' \qquad P \xrightarrow{\overline{a}} P''}{!P \xrightarrow{\tau} P' \,|\, P'' \,|\, !P}$$

There are three kinds of actions (ranged over by α, β, γ): input (a), output ($\overline{a}$), and internal (τ). The τ action is often referred to as silent or invisible, and the others as visible. We write $\overline{\gamma}$ for the complement of γ, *i.e.*, $\overline{\gamma}$ is $\overline{a}$ if γ is a and a if γ is $\overline{a}$. Sometimes we will omit the trailing 0 in a process, *e.g.*, $a, \overline{a}$ are shortcuts for $a.0, \overline{a}.0$ respectively. A context $C[\cdot]$ is a process with some subprocess replaced by a hole $[\cdot]$, and $C[Q]$ means substituting the hole in C with process Q. We use $\equiv$ to stand for the standard structural congruence for CCS$^-$ [2, 3]. It is the smallest congruence relation satisfying the monoid laws and commutative laws for parallel composition. That is,

$$P \,|\, 0 \equiv P \qquad P \,|\, (Q \,|\, R) \equiv (P \,|\, Q) \,|\, R \qquad P \,|\, Q \equiv Q \,|\, P$$

A name is said to be fresh if it does not appear in the current processes. We use $\mathrm{n}(\cdot)$ to denote the names in a set of processes or actions. We denote by $\Longrightarrow$ the reflexive transitive closure of internal actions, and $\stackrel{\lambda}{\Longrightarrow}$ is $\Longrightarrow \xrightarrow{\lambda} \Longrightarrow$. In addition, $\stackrel{\hat{\lambda}}{\Longrightarrow}$ is $\Longrightarrow$ when λ is τ and $\stackrel{\lambda}{\Longrightarrow}$ otherwise. We use $\xrightarrow{\tau}_k$ to mean k consecutive τ's. For a binary relation $\mathscr{R}$, we use $P \mathscr{R} Q$ as a shortcut for $(P, Q) \in \mathscr{R}$. Sometimes we write $P \xrightarrow{\lambda} \mathscr{R} P'$ to mean that there exists P'' such that $P \xrightarrow{\lambda} P''$ and $P'' \mathscr{R} P'$.

A process is divergent if it can fire an infinite sequence of τ actions, *e.g.*, $!(\overline{a} \,|\, a)$ and $!\overline{a} \,|\, !a$. Said otherwise, if a process is not divergent, then it can only engage finitely many internal actions.

Bisimulation

We now define the strong and weak bisimulations. It is noted that the bisimulations here explicitly consider the divergence property, so to distinguish between divergent and non-divergent processes [1, 2, 4]. We say that a relation $\mathscr{R}$ is divergence-sensitive, if for every $P \mathscr{R} Q$, P diverges if and only if Q does. Technically, this simplifies the discussion of the results in this section, because divergence sensitivity requires a pair of processes to simultaneously diverge or not, unconditionally.

Definition 2.1. A symmetric binary relation $\mathscr{R}$ on CCS$^-$ processes is a strong (respectively weak) bisimulation if it is divergence-sensitive and whenever $P\mathscr{R}Q$, it holds that

- if $P \xrightarrow{\alpha} P'$, then $Q \xrightarrow{\alpha} Q'$ (respectively $Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$) and $P'\mathscr{R}Q'$;

Two processes P and Q are strongly (respectively weakly) bisimilar, notation $P \sim_{ccs^-} Q$ (respectively $P \approx_{ccs^-} Q$), if there exists some strong (respectively weak) bisimulation $\mathscr{R}$ such that $P \mathscr{R} Q$.

We call $\approx_{ccs^-}$ and $\sim_{ccs^-}$ the weak bisimilarity and strong bisimilarity respectively. As is well-known, they are equivalence relations and congruences. We notice that, in Definition 2.1, the requirement of 'divergence-sensitive' is redundant for the strong bisimilarity, since a strong bisimulation is always divergence-sensitive. In the discussion of the bisimilarities, we may use the up-to techniques to build bisimulations, *e.g.*, bisimulation up-to context. These are well-established proof methods for process models; see [2] for a comprehensive introduction. For the sake of convenience, we give the definition of the (weak) bisimulation up-to context. We shall see how it is used in the passing.

Definition 2.2. A symmetric binary relation $\mathscr{R}$ on CCS$^-$ processes is a (weak) bisimulation up-to context if it is divergence-sensitive and whenever $P\mathscr{R}Q$, it holds that

- if $P \xrightarrow{\alpha} P'$, then $Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$ and there exist some context C, P_1 and Q_1 such that

$$P' \equiv C[P_1], \quad Q' \equiv C[Q_1], \quad \text{and} \quad P_1 \mathscr{R} Q_1$$

As the following lemma states, if a relation is a bisimulation up-to context, then it is subsumed by the weak bisimilarity. The proof of this lemma amounts to showing that the weak bisimilarity is contextual (*i.e.*, preserved by contexts), and divergence sensitivity would not raise obstacles because it does not involve explicit action matching; see [2] for a reference of proof and more discussion.

Lemma 2.3. *If $\mathscr{R}$ is a bisimulation up-to context, then it holds that $\mathscr{R} \subseteq \approx_{ccs^-}$.*

In terms of the bisimilarity, actions can be divided into two classes: state-changing and state-preserving. We say that an action α as occurring in $P \xrightarrow{\alpha} P'$ is state-changing if $P \not\approx_{ccs^-} P'$; otherwise it is state-preserving. It should be clear that the following implications are true [2, 3].

Lemma 2.4. *It holds that $\equiv \subsetneq \sim_{ccs^-} \subseteq \approx_{ccs^-}$.*

Both of the implications of Lemma 2.4 are immediate from the definitions. For the first implication of the lemma to be strict, we notice that $!a.0$ and $!a.0 \,|\, !a.0$ are strongly bisimilar but not structurally congruent.

Now, we define a bisimulation that stands in between the strong and weak bisimilarities. As will be shown, it coincides with the weak bisimilarity and is slightly weaker than the strong bisimilarity (thus so is the weak bisimilarity).

Definition 2.5. A symmetric binary relation $\mathscr{R}$ on CCS$^-$ processes is a quasi-strong bisimulation if it is divergence-sensitive, and whenever $P\mathscr{R}Q$, the following properties hold.

- if $P \xrightarrow{\alpha} P'$ and α is not τ, then $Q \Longrightarrow \xrightarrow{\alpha} Q'$ and $P'\mathscr{R}Q'$.
- if $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} Q'$ and $P'\mathscr{R}Q'$.

Two processes P and Q are quasi-strongly bisimilar, notation $P \sim^q_{ccs^-} Q$, if there exists some quasi-strong bisimulation $\mathscr{R}$ such that $P \mathscr{R} Q$.

For a quasi-strong bisimulation, since internal actions are bisimulated in a strong manner, divergence sensitivity somehow becomes a derived condition, and we include it in the definition for convenience. By definition, it is straightforward to see that $\sim_{ccs^-} \subseteq \sim^q_{ccs^-} \subseteq \approx_{ccs^-}$.

2.2. The relationship between the weak and strong bisimilarities in CCS$^-$

We first notice that the missing of the restriction operation might lend confidence to the coincidence between $\approx_{ccs^-}$ and $\sim_{ccs^-}$. But this turns out to be wrong. We will elaborate this in the current section.

Particularly, in CCS$^-$, the weak bisimilarity is still strictly coarser than the strong bisimilarity. The intrinsic reason is that we still have the replication operator, from which infinite behaviour arises. This renders false the matching of visible actions in a strong manner (the best we can have is as Proposition 2.19 demonstrates).

To see a counterexample, take

$$P_1 \stackrel{\text{def}}{=} !c.d \,|\, !\overline{c} \,|\, d$$
$$P_2 \stackrel{\text{def}}{=} !c.d \,|\, !\overline{c} \,|\, !c$$

Obviously $P_1 \not\sim_{ccs^-} P_2$. However, $P_1 \approx_{ccs^-} P_2$, because the action

$$P_1 \xrightarrow{d} !c.d \,|\, !\overline{c} \,|\, 0$$

can be simulated by

$$P_2 \xrightarrow{\tau} !c.d \,|\, d \,|\, 0 \,|\, !\overline{c} \,|\, !c \xrightarrow{d} !c.d \,|\, 0 \,|\, 0 \,|\, !\overline{c} \,|\, !c$$

Henceforth, every d produced by P_1 in simulating the subprocess $!c$ in P_2 can be simulated in a similar way. So we have the following proper inclusion.

Lemma 2.6. *In* CCS$^-$, $\sim_{ccs^-} \subsetneq \approx_{ccs^-}$.

In the remainder of this section, we prove the follow-up theorem (Thm. 2.7), which says that the weak bisimilarity can somehow be approximated by a partially strong bisimilarity, *i.e.*, the quasi-strong bisimilarity. It reveals that in CCS$^-$, there is truly some stronger characterization of the weak bisimilarity, or in other words, the gap between the strong and weak bisimilarities is diminished to some (arguably) noticeable extent. In a broader sense, this exhibits that the weak bisimilarity can indeed be strengthened in a process model without the restriction operator.

Theorem 2.7. *Assume that P and Q are* CCS$^-$*processes. Then $P \approx_{ccs^-} Q$ implies $P \sim^q_{ccs^-} Q$.*

This theorem immediately leads to the corollary below.

Corollary 2.8. *In* CCS$^-$, *it holds that* $\approx_{ccs^-} = \sim^q_{ccs^-}$.

To establish the implication claimed by Theorem 2.7, we exploit the structure of CCS$^-$ processes, and go through multiple analyses toward our goal, *i.e.*, the weak bisimilarity is indeed a quasi-strong bisimulation.

The following lemma claims that every τ action changes the interactional capability, *i.e.*, the 'state' of a process, except for those with infinite actions. We recall that a process P diverges if it has an infinite sequence of τ actions, *i.e.*, $P \xrightarrow{\tau} \cdots \xrightarrow{\tau} \cdots$, and we say that a process P has an infinite number of visible action α (α is not τ) if $P \Longrightarrow \xrightarrow{\alpha} \Longrightarrow \xrightarrow{\alpha} \cdots \Longrightarrow \xrightarrow{\alpha} \cdots$, among which a noticeable special case is $P \xrightarrow{\alpha} \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} \cdots$.

Lemma 2.9. *If P is not divergent and $P \xrightarrow{\tau} P'$, then $P \not\approx_{ccs^-} P'$.*

Proof. First, we note that the only possibility of having infinite actions is through replication, since the replication operator is the only way of producing infinity, by a simple induction. We have the following arguments that lead to the result of this lemma.

1. If P is not divergent, then for any a, P cannot have both an infinite number of $\overline{a}$ actions and an infinite number of a actions (notice that only τ actions occur between each two neighbouring visible actions). Assume for a contradiction that P has both an infinite number of $\overline{a}$ actions and an infinite number of a actions. Then there are two possibilities.
   (a) These two threads, *i.e.*, infinite numbers of $\overline{a}$ actions and a actions respectively, are in parallel composition. That is, $P \equiv P_1 \,|\, P_2$ in which P_1 is capable of an infinite number of $\overline{a}$ actions and P_2 is capable of an infinite number of a actions;

(b) The two thread are intertwined, *i.e.*, $P \Longrightarrow \xrightarrow{\gamma_1} \Longrightarrow \xrightarrow{\gamma_2} \cdots \Longrightarrow \xrightarrow{\gamma_j} \Longrightarrow \cdots$ in which γ_i is either $\overline{a}$ or a.

We claim that both case (a) and case (b) would lead to the divergence of P, a contradiction. Case (a) is obvious. For case (b), to yield the infinite visible action sequence (separated by τ actions only), every action γ_i must be consumable through interactions, *i.e.*, each γ_i should go away by taking part in some interaction.

More specifically, in order for P to have an infinite number of both a and $\overline{a}$, all the γ_i in the sequence should be expendable by means of interactions, otherwise such behavior of P (say, having an infinite number of a actions) would not happen since it would be separated by non-τ actions. As such, the procedure of the consumption of the γ_i actions would lead to a divergence path of P, which is a contradiction.

2. Due to 1, if P is not divergent and can make a τ that comes from an interaction, say over a, between its parallel components (*i.e.*, two processes on parallel position), then $P \xrightarrow{\overline{a}}$ and also $P \xrightarrow{a}$, but not both of these two actions occur infinitely. This, in turn, implies that P can make either a finite number of $\overline{a}$ actions or a finite number of a actions. Suppose it is the latter (*i.e.*, $P \Longrightarrow \xrightarrow{a}$ occurs a finite number of times), and the former is similar. Then we conclude that P cannot be bisimilar to P' because P' is short of one action a for such a name a.

More specifically, the central point here is that the result of 1 ensures that if P makes a τ action through an interaction over a, then it cannot have an infinite number of both a and $\overline{a}$ actions, provided it is not divergent. This, in turn, indicates that there can be only a finite number of a or $\overline{a}$ that P can make. Consequently, a τ action cannot consume actions a or $\overline{a}$ while preserving its state, since the resulting process would have strictly fewer actions a or $\overline{a}$. □

We have a straightforward corollary of Lemma 2.9, since $\sim_{ccs^-}$ is included in $\approx_{ccs^-}$.

Corollary 2.10. *If P is not divergent and $P \xrightarrow{\tau} P'$, then $P \not\approx_{ccs^-} P'$.*

Another corollary also follows straight away.

Corollary 2.11. *If $P \xrightarrow{\tau} P'$ and $P \sim_{ccs^-} P'$ (or $P \approx_{ccs^-} P'$), then P is divergent.*

The following lemma represents that τ actions resulting from a replication in a process do not change the state of that process.

Lemma 2.12. *Whenever $!P \Longrightarrow P'$, it holds that $!P \approx_{ccs^-} P'$.*

Proof. In order to prove this lemma, we first tackle a base case where $!P \Longrightarrow P'$ contains one τ action, and then show the result of thus lemma on the basis of it.

- We show the result for the case when the length of the τ actions by $!P$ is 1, *i.e.*, $!P \xrightarrow{\tau} P'$.
  From the semantics, the τ action by $!P$ must be one of the following cases. Notice that in all of the cases, $P' \equiv Q \mid !P$ for some Q.
  1. $!P \xrightarrow{\tau} P_1 \mid !P$ and $P_1' \stackrel{\text{def}}{=} P_1 \mid !P \equiv P'$ (and $Q \equiv P_1$) due to $P \xrightarrow{\tau} P_1$.
     We show that the following relation $\mathscr{R}$ is a weak bisimulation up-to context (Def. 2.2; see also [2]), so that $!P \approx_{ccs^-} P_1'$.

$$\mathscr{R} \stackrel{\text{def}}{=} \{(!P, P_1')\} \cup \approx_{ccs^-}$$

     Assume $!P \,\mathscr{R}\, P_1'$. We have two simulation scenarios.
     (a) Suppose $P_1' \xrightarrow{\alpha} P_1''$. Then $!P$ simulates by $!P \xrightarrow{\tau} P_1' \xrightarrow{\alpha} P_1''$, and $P_1'' \,\mathscr{R}\, P_1''$ because $P_1'' \approx_{ccs^-} P_1''$. In the spirit of "up-to context", one simply sets the context to be $[\cdot]$.

(b) Conversely, suppose $!P \xrightarrow{\alpha} P_2 \,|\, !P \stackrel{\text{def}}{=} P_2'$ (when α is τ, we may assume P_1 is not P_2 because otherwise the simulation is trivial). Then P_1' simulates by

$$\begin{array}{rcl} P_1' & \equiv & P_1 \,|\, !P \xrightarrow{\alpha} P_1 \,|\, P_2 \,|\, !P \\ & \equiv & P_2 \,|\, P_1 \,|\, !P \\ & \equiv & P_2 \,|\, P_1' \end{array}$$

The simulation now continues by taking advantage of the "up-to context". By setting $C \stackrel{\text{def}}{=} P_2 \,|\, [\cdot]$, we have

$$P_2' \equiv P_2 \,|\, !P \equiv C[!P]$$

$$P_1 \,|\, P_2 \,|\, !P \equiv P_2 \,|\, P_1' \equiv C[P_1']$$

in which $!P \,\mathscr{R}\, P_1'$. So we are done.

2. $!P \xrightarrow{\tau} P_1 \,|\, P_2 \,|\, !P \equiv P' \stackrel{\text{def}}{=} P_1''$ (and $Q \equiv P_1 \,|\, P_2$) due to $P \xrightarrow{a} P_1$ and $P \xrightarrow{\overline{a}} P_2$.
We show that the following relation $\mathscr{R}'$ is a weak bisimulation up-to context, which implies that $!P \approx_{ccs^-} P_1''$.

$$\mathscr{R}' \stackrel{\text{def}}{=} \{(!P, P_1'')\} \cup \approx_{ccs^-}$$

Assume $!P \,\mathscr{R}'\, P_1''$. We have two simulation scenarios.

(a) Suppose $P_1'' \xrightarrow{\alpha} P_1'''$. Then $!P$ simulates by $!P \xrightarrow{\tau} P_1'' \xrightarrow{\alpha} P_1'''$, and $P_1''' \,\mathscr{R}'\, P_1'''$ follows due to $P_1''' \approx_{ccs^-} P_1'''$. One can set the context to be $[\cdot]$ so to comply with the requirement of "up-to context".

(b) Conversely, suppose $!P \xrightarrow{\alpha} P_3 \,|\, !P \stackrel{\text{def}}{=} P_3'$ (we may assume P_3 is not $P_1 \,|\, P_2$ when α is τ because otherwise the simulation is trivial). Then P_1'' simulates by

$$\begin{array}{rcl} P_1'' & \equiv & P_1 \,|\, P_2 \,|\, !P \xrightarrow{\alpha} P_1 \,|\, P_2 \,|\, P_3 \,|\, !P \\ & \equiv & P_3 \,|\, P_1 \,|\, P_2 \,|\, !P \\ & \equiv & P_3 \,|\, P_1'' \end{array}$$

Taking advantage of the "up-to context" and setting $C' \stackrel{\text{def}}{=} P_3 \,|\, [\cdot]$, we have

$$P_3' \equiv P_3 \,|\, !P \equiv C'[!P]$$

$$P_1 \,|\, P_2 \,|\, P_3 \,|\, !P \equiv P_3 \,|\, P_1'' \equiv C'[P_1'']$$

in which $!P \,\mathscr{R}\, P_1''$. So we are done.

- Now we show the lemma in its general case, *i.e.*, $!P \Longrightarrow P'$. Through a routine (transition) induction, it can be shown that P' must be of the form $P'' \,|\, !P$ for some P'' (up-to $\equiv$). We can obtain $!P \approx_{ccs^-} P'$ by showing the following relation $\mathscr{R}_1$ to be a weak bisimulation up-to context.

$$\mathscr{R}_1 \stackrel{\text{def}}{=} \{(!P, P')\} \cup \approx_{ccs^-}$$

To achieve this and complete the proof, we use the same arguments as case 1 above in showing $\mathscr{R}$ and $\mathscr{R}'$ to be weak bisimulations up-to context. The only difference here is that there may be more than one τ action in the weak transition by $!P$. To this point, the form of the derived process P' remains the same after the weak transition $!P\Longrightarrow P' \equiv P'' \,|\, !P$. That is, the resulting process P' from $!P\Longrightarrow P'$ has the same form as from $!P \xrightarrow{\tau} P'$, so the following arguments for establishing the weak bisimilarity would be the same. So we conclude that the result of this lemma holds. □

It is noteworthy that Lemma 2.12 is also true in the standard CCS (with restriction), because the proof does not rely on the absence of the restriction operation. However, if $!P$ has some visible actions, then its state is not preserved any longer. That is, for P' such that $!P\Longrightarrow\xrightarrow{\alpha_1}\Longrightarrow\xrightarrow{\alpha_2}\Longrightarrow\cdots\Longrightarrow\xrightarrow{\alpha_k}\Longrightarrow P'$ in which α_i (i=1, ..., k) is visible (*i.e.*, not τ), it does not necessarily hold that $!P \approx_{ccs^-} P'$. The reason is that the action by $!P$ may reveal some action $!P$ cannot match without first doing another visible action. It is not hard to contrive a counterexample, in which P' can do some action that $!P$ cannot do. For instance, let $P \stackrel{\text{def}}{=} !a.c$. Then $!P \xrightarrow{a} c \,|\, !P \equiv P'$, and $P' \xrightarrow{c}$ whereas $!P$ cannot.

We now continue to examine the state-preserving/state-changing τ actions in the upcoming lemmas. Again by state-changing, we mean that if $P \xrightarrow{\alpha} P'$, then $P \not\approx_{ccs^-} P'$ (more often than not, α here is τ, though it appears to still make sense if it is visible); otherwise it is state-preserving.

Lemma 2.13. *Assume P is a* CCS$^-$ *process. Then there exists $k \geqslant 0$ and P' such that $P \xrightarrow{\tau}_k P'$, and $P' \approx_{ccs^-} P''$ for any P'' such that $P'\Longrightarrow P''$.*

Proof. Before going ahead, we provide some observation. If P is not divergent, then the result obviously holds, because we can consume all the finite τ actions P can fire to reach a state that can make no more τ. That state satisfies the claim of the lemma (in a void way). Now assume that P is divergent. Intuitively, the number k is referring to those τ's that are not introduced by the replication. Consuming these τ actions leads P to a state ready for starting the divergent path. By "the divergent path", we mean that the process can make the same τ over and again, without changing the state.

Now we give the proof by induction on the structure of P.

- The cases when P is 0, $a.P$, or $\overline{a}.P$ are trivial.
- P is $P_1 \,|\, P_2$. There are several subcases.
  1. P_1 is divergent while P_2 is not. (For example, $P_1 \stackrel{\text{def}}{=} !a \,|\, !\overline{a} \,|\, \overline{c}$, $P_2 \stackrel{\text{def}}{=} c$. In here and what follows, we may use examples simply for more of an illustration, and they are not meant to be part of the proof anyhow.)
     In this case, the claim of the lemma holds for P_1 by induction hypothesis. That is, there exists $k_1 \geqslant 0$ and P_1' such that $P_1 \xrightarrow{\tau}_{k_1} P_1'$, and $P_1' \approx_{ccs^-} P_1''$ for any P_1'' such that $P_1' \Longrightarrow P_1''$. Now since P_2 is not divergent, we can expend all the possible τ actions concerning P_2, including those by P_2 alone or from finite interactions between P_2 and P_1. By "finite interactions" we mean that the interactions are composed of visible actions that cannot be repeated for infinitely many times. In addition, notice that by means of interaction, we only consume those finite τ actions from inside P_2 or between P_1 and P_2. Suppose this expending operation constitutes k_1' ($k_1' \geqslant 0$) τ actions, and takes P_2 to P_2'. After this P_2 may still have infinite visible actions and have infinite τ's with P_1, but this does not matter because we now can use induction hypothesis to obtain that there exists $k_2 \geqslant 0$ and P_2'' such that $P_2' \xrightarrow{\tau}_{k_2} P_2''$, and $P_2'' \approx_{ccs^-} P_2'''$ for any P_2''' such that $P_2'' \Longrightarrow P_2'''$. Consequently, we have that $P \xrightarrow{\tau}_{k_1+k_1'+k_2} P'$ for some P', and $P' \approx_{ccs^-} P''$ for any P'' such that $P'\Longrightarrow P''$.
  2. P_2 is divergent while P_1 is not. Similar to the previous case.
  3. Both P_1 and P_2 are divergent. (For example, $P_1 \stackrel{\text{def}}{=} !a \,|\, !\overline{a} \,|\, \overline{c}$, $P_2 \stackrel{\text{def}}{=} c \,|\, !b \,|\, !\overline{b}$.)
     By induction hypothesis, we know that there exists $k_i \geqslant 0$ ($i = 1, 2$) and P_i' such that $P_i \xrightarrow{\tau}_{k_i} P_i'$, and $P_i' \approx_{ccs^-} P_i''$ for any P_i'' such that $P_i' \Longrightarrow P_i''$. Similar to case 1, we can remove those finite, say $k_3 \geqslant 0$, τ actions between P_1 and P_2. Then we know that $P \xrightarrow{\tau}_{k_1+k_2+k_3} P'$, and $P' \approx_{ccs^-} P''$ for any P'' such that $P'\Longrightarrow P''$.
  4. Neither of P_1 and P_2 is divergent, but P is divergent. We break into two subcases.

(a) We first separate a subcase that P_1 is indirectly divergent (the case for P_2 is similar).
(For example, $P_1 \stackrel{\text{def}}{=} \overline{c}.(!a \,|\, !\overline{a})$, $P_2 \stackrel{\text{def}}{=} c$)
That is, P_1 becomes divergent after having some finite interaction with P_2. Suppose after some finite interaction with P_2, P_1 and P_2 become P_1' and P_2' respectively, where P_1' is divergent. Then $P \Longrightarrow P_1' \,|\, P_2'$, and the result holds by proceeding as in case 1.

(b) Excluding the indirectly divergent cases as in (a), we may suppose $P \equiv P_1 \,|\, P_2 \xrightarrow{\tau}_k P_1' \,|\, P_2'$ in which $k \geqslant 0$ and neither P_1' nor P_2' can do τ actions.
(For example, $P_1 \stackrel{\text{def}}{=} !a \,|\, \overline{c}$, $P_2 \stackrel{\text{def}}{=} c \,|\, !\overline{a}$)
That is, we can deplete all the finite interactions in/between P_1 and P_2 as in case 1. Now since P is divergent, we know that: first, there must be some interaction between P_1' and P_2'; second, the visible actions comprising the interaction, and thus the interaction itself, can recur forever. By "recur forever", we formally mean that for any $k \geq 0$ such that $P_1' \underbrace{\xrightarrow{\alpha}\Longrightarrow}_{k\ times} Q$ for some Q, it holds that $Q \stackrel{\alpha}{\Longrightarrow}$; similar for P_2'. Otherwise, one would not have divergence in this case. Thus suppose, in particular, that $P_1' \stackrel{\alpha}{\Longrightarrow}$ and $P_2' \stackrel{\bar{\alpha}}{\Longrightarrow}$, and these actions run infinitely. Then $P_1' \,|\, P_2'$ is exactly the P' as needed, because it holds that $P_1' \,|\, P_2' \xrightarrow{\tau}_{k'} P_1'' \,|\, P_2'' \approx_{ccs^-} P_1' \,|\, P_2'$ for any $k' \geqslant 0$, due to the infinite repository of the same actions. By "infinite repository", we mean formally that the action is available all the time, *i.e.*, $P \xrightarrow{\alpha} P' \sim P$, due to the infinite actions coming from the replication operation.

- P is $!P_1$. Suppose $!P_1$ can do some τ (the result trivially holds otherwise). Then taking k as zero would be sufficient, by Lemma 2.12. That is, all the incoming τ actions do not change the state of $!P_1$.

Now the proof is completed. □

By virtue of Lemma 2.13, the number of such τ actions as state-changing is finite in a sense. That is to say, a CCS$^-$ process can only make finite state-changing τ actions in a sequence, without any intertwining visible actions.

Corollary 2.14. *Suppose that P can make a (possibly infinite) sequence of τ actions, then only a finite initial segment of it is state-changing. This τ sequence can be written as $P \xrightarrow{\tau} P_1 \xrightarrow{\tau} P_2 \cdots \xrightarrow{\tau} P_k \xrightarrow{\tau} \cdots$, where P and P_1 through P_k are state-changing and those afterwards (if any) are state-preserving.*

Proof. This is a consequence of what we have proven in Lemma 2.13. One simply chooses the minimal k as designated by that lemma. □

In general, it is not true that a process can only have finite state-changing τ actions (not necessarily from the start and within a sequence). For example, the process $!a.(b \,|\, \overline{b}.c)$ can have infinite state-changing τ actions. However, the τ actions in a row without any in-between visible actions can only have finite state-changing (consecutive) τ actions as stated in the foregoing corollary.

Moreover, a τ action that changes the state of a process, which may not come from a replication, should be simulated also by a state-changing τ. We capture these ideas in the following two lemmas.

Lemma 2.15. *Assume $P \approx_{ccs^-} Q$. Then in the τ sequences that P and Q can make respectively (as described in Cor. 2.14), the numbers of state-changing τ actions are equal.*

Proof. By Corollary 2.14, we can assume that the τ sequences by P and Q are respectively

$$
\begin{aligned}
&P \xrightarrow{\tau} P_1 \xrightarrow{\tau} P_2 \cdots \xrightarrow{\tau} P_k \xrightarrow{\tau} \cdots \\
&Q \xrightarrow{\tau} Q_1 \xrightarrow{\tau} Q_2 \cdots \xrightarrow{\tau} Q_{k'} \xrightarrow{\tau} \cdots
\end{aligned}
$$

where P (respectively Q) and P_1 (respectively Q_1) through P_k (respectively $Q_{k'}$) are state-changing and those afterwards (if any) are state-preserving. We need to show $k = k'$. Assume to the contrary that $k \neq k'$, say $k < k'$ (the case $k > k'$ is similar).

We then argue that this would break the bisimulation between P and Q. We observe that every τ action is composed of two complementary visible actions (since there is no restriction operation, every interaction is based on a visible name), so one can make these two visible actions in a sequential manner to reach the same state. Say $P \xrightarrow{\tau} P'$ through interaction over a, then from the operational semantics we must have $P \xrightarrow{a} \xrightarrow{\overline{a}} P'$, which we call a pair-action. Moreover, because the τ considered here is state-changing, these complementary visible actions must not be repeated forever (otherwise, they would lead to repeated τ actions, rendering it state-preserving). So P can reach P_k in a finite number of such pair-actions. Since $P \approx_{ccs^-} Q$, it must be the case that Q consumes exactly the same number of pair actions (of visible actions), and this corresponds to k state-changing τ actions. That is, Q must evolve to Q_k in order to keep pace with P so that the bisimulation can be maintained. However, as assumed, Q can make more state-changing τ actions, *i.e.*, more pair-actions correspondingly. These (non-repeating) pair actions cannot be bisimulated by P since P at that stage only has state-preserving τ actions (and consequently repeating pair-actions). Hence, we have a contradiction. □

Lemma 2.15 helps to develop the following intuition: in order to simulate a state-changing τ, one must do at least one τ so as to change its state; moreover, it cannot do more than one state-changing τ because it has only finite such τ actions in the (bisimulating) sequence of internal actions, even though it can do a couple of state-preserving τ actions; making such a state-changing τ is sufficient for the bisimulation.

Therefore, we have the follow-up lemma on how to match internal actions that are state-changing.

Lemma 2.16. *Suppose $P \approx_{ccs^-} Q$. If $P \xrightarrow{\tau} P'$ in which the τ is state-changing, then $Q \xrightarrow{\tau} Q' \approx_{ccs^-} P'$ in which the τ is state-changing as well.*

Proof. Suppose $P \xrightarrow{\tau} P'$ in which the τ is state-changing, since $P \approx_{ccs^-} Q$, we know that $Q \Longrightarrow Q_1$ for some Q_1 and $P' \approx_{ccs^-} Q_1$. Because the τ by P is state-changing, it must originate from two complementary visible actions, say a and $\overline{a}$, that cannot be simulated by P' anyhow. Thus Q must not do nothing, so we can rewrite the simulation by Q as $Q \xrightarrow{\tau} Q_2 \Longrightarrow Q_1 \approx_{ccs^-} P'$. We claim that in this simulation, the τ in $Q \xrightarrow{\tau} Q_2$ is state-changing, and the τ actions in $Q_2 \Longrightarrow Q_1$ are state-preserving. Then the result of the lemma follows.

To see why Q is forced to simulate with exactly one state-changing τ, we note that Q can only make finite state-changing τ action sequence (by Cor. 2.14) whose length is the same as that of P's (by Lem. 2.15). If the simulation consumes any more or fewer state-changing τ actions in the sequence, Q will eventually lose the pace with P and fail to bisimulate it. Thus P and Q are obliged to engage a step-wise state-changing bisimulation, as they only hold a finite number of such actions.

Therefore, it must be the case that the τ in $Q \xrightarrow{\tau} Q_2$ is state-changing, and the τ actions from Q_2 to Q_1 are state-preserving, *i.e.*, $Q_2 \approx_{ccs^-} Q_1 \approx_{ccs^-} P'$. To conclude, now we have $Q \xrightarrow{\tau} Q_2 \approx_{ccs^-} P'$, and Q_2 is the Q' we seek in the statement of the lemma. □

A corollary out of Lemma 2.15 and Lemma 2.16 is that a state-preserving τ must be simulated by a state-preserving one, since it cannot be bisimulated by a state-changing τ action.

Corollary 2.17. *Suppose $P \approx_{ccs^-} Q$. If $P \xrightarrow{\tau} P'$ in which the τ is state-preserving, then $Q \xrightarrow{\tau} Q' \approx_{ccs^-} P'$ in which the τ is state-preserving as well.*

The upcoming lemma somewhat generalizes the results in the foregoing lemmas.

Lemma 2.18. *Assume P and Q are* CCS$^-$ *processes, and k is as described in Lemma 2.13 for P. If $P \approx_{ccs^-} Q$, then*

1. *$Q \xrightarrow{\tau}_k Q'$, and $Q' \approx_{ccs^-} Q''$ for any Q'' such that $Q' \Longrightarrow Q''$;*

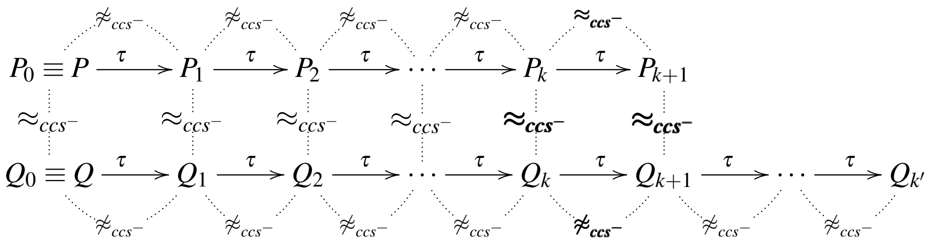

FIGURE 1. Figure for the proof of Lemma 2.18.

2. *there is a minimal k satisfying both (1) for Q and the property as described in Lemma 2.13 for P; i.e., P and Q make the same number of state-changing τ action sequence before entering (state-preserving) divergence (if any). In other words, formally, both of the following hold.*
   - $P \xrightarrow{\tau}_k P'$, *and* $P' \approx_{ccs^-} P''$ *for any* P'' *such that* $P' \Longrightarrow P''$.
   - $Q \xrightarrow{\tau}_k Q'$, *and* $Q' \approx_{ccs^-} Q''$ *for any* Q'' *such that* $Q' \Longrightarrow Q''$.

Proof. It should be clear that (2) follows from (1) and Lemma 2.13.

For (1), one has to make sure that Q has the same behavioural pattern. To see this, we remember that the bisimulation here requires synergistic divergence, so Q is divergent too whenever P is. We also notice that Q has the property due to Lemma 2.13, *i.e.*, there exists $k' \geqslant 0$ and Q' such that $Q \xrightarrow{\tau}_{k'} Q'$, and $Q' \approx_{ccs^-} Q''$ for any Q'' such that $Q' \Longrightarrow Q''$.

For the sake of convenience and w.l.o.g., we assume that k and k' are both the minimal such integers respectively (the result for 'not minimal' follows from this case). We need to show that $k = k'$. Intuitively, if $k < k'$ or $k > k'$, then we can derive that P and Q are not bisimilar. Assume to the contrary that $k < k'$ (the case $k > k'$ is similar). This implies that Q becomes state-preserving later than P. In other words, this means that Q becomes state-preserving at a time later than P due to taking more τ actions prior to turning state-preserving. In light of Corollary 2.10, starting from $P \approx_{ccs^-} Q$ and before entering the state-preserving stage (diverging if any), both P and Q change their states stepwise in each 'bisimulating' action. Specifically, if $k < k'$, consider the following two sequences.

$$P \equiv P_0 \xrightarrow{\tau} P_1 \xrightarrow{\tau} P_2 \cdots \xrightarrow{\tau} P_k \xrightarrow{\tau} P_{k+1}$$

$$Q \equiv Q_0 \xrightarrow{\tau} Q_1 \xrightarrow{\tau} Q_2 \cdots \xrightarrow{\tau} Q_k \xrightarrow{\tau} Q_{k+1} \xrightarrow{\tau} \cdots \xrightarrow{\tau} Q_{k'}$$

We have

(1) $P_i \not\approx_{ccs^-} P_{i+1}$ $(i = 0, ..., k-1)$. Before turning into state-preserving stage (*i.e.*, divergence), every τ changes the state of the process evolving from P (recall that we choose the minimal k).
(2) $P_k \approx_{ccs^-} P_{k+1}$. Once entering the phase of state-preserving, no τ actions change the state.
(3) $Q_i \not\approx_{ccs^-} Q_{i+1}$ $(i = 0, ..., k'-1)$. For a reason similar to (1).
(4) $P_i \approx_{ccs^-} Q_i$ $(i = 0, ..., k)$. This is ensured by Lemma 2.16 and Lemma 2.15, since the τ actions (until P_k and Q_k) change the states.

Now, these observations lead to a contradiction, *i.e.*, we have $P_k \approx_{ccs^-} P_{k+1} \approx_{ccs^-} Q_{k+1}$ since P_k must simulate Q_k's τ, but then $Q_k \approx_{ccs^-} Q_{k+1}$, which is contradictory. This scenario is depicted as in Figure 1, in which the place where the contradiction emerges is highlighted with the bold font. That is, the transitive path $Q_k \approx_{ccs^-} P_k \approx_{ccs^-} P_{k+1} \approx_{ccs^-} Q_{k+1}$ yields a contradiction with the edge connecting Q_k and Q_{k+1}, *i.e.*, $Q_k \not\approx_{ccs^-} Q_{k+1}$.

□

The next lemma gives a crucial property for analyzing the matching of τ actions in terms of the weak bisimilarity.

Proposition 2.19. *Assume $P \approx_{ccs^-} Q$ and $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} \approx_{ccs^-} P'$.*

Proof. By assumption, we know that $P \xrightarrow{\tau} P'$ implies $Q \Longrightarrow \approx_{ccs^-} P'$. We need to more precisely analyze the matching weak transition by Q. The analysis separates the cases whether P is divergent or not. We first tackle the case when P is not divergent. By Lemma 2.9, we know that $P \not\approx_{ccs^-} P'$, *i.e.*, that τ is state-changing. Then By Lemma 2.16, we have $Q \xrightarrow{\tau} Q' \approx_{ccs^-} P'$ in which τ is state-changing.

Now we consider the case when P is divergent. Since $P \approx_{ccs^-} Q$, we know that Q diverges too. Assume k is as decided by Lemma 2.18. That is, P and Q make the same number of state-changing τ action sequence before entering divergence (in which τ is state-preserving). There are two subcases.

(1) If k is 0, then the τ in $P \xrightarrow{\tau} P'$ must be state-preserving. By Corollary 2.17, $Q \xrightarrow{\tau} Q' \approx_{ccs^-} P'$ in which τ is state-preserving.
(2) If k is not 0, then the τ in $P \xrightarrow{\tau} P'$ can be state-changing or state-preserving (*i.e.*, from the divergence after k). In the former, the result follows as above in the case P is not divergent. In the latter, we conclude as in (1). □

We have now finished analyzing the τ action in a bisimulation. Finally, we deal with the situation for visible actions.

Proposition 2.20. *Assume $P \approx_{ccs^-} Q$ and $P \xrightarrow{\alpha} P'$ in which α is not τ, then $Q \Longrightarrow Q_1 \xrightarrow{\alpha} Q_2 \approx_{ccs^-} P'$ for some Q_1, Q_2, in which $Q \approx_{ccs^-} Q_1$ and the τ actions in $Q \Longrightarrow Q_1$ are state-preserving.*

Proof. By the premise, Q must simulate by $Q \Longrightarrow Q_1 \xrightarrow{\alpha} Q_2 \Longrightarrow Q' \approx_{ccs^-} P'$ for some Q_1, Q_2, and Q'. In this simulation, by Lemmas 2.15, 2.16 and Corollary 2.17, the τ sequence between Q and Q_1, and the τ sequence between Q_2 and Q', contain no state-changing τ actions. If assumed otherwise, Q' would be unable to match P' due to being short of enough state-changing τ actions as compared with those of P'. So the internal action sequences in the bisimulation must be state-preserving. Consequently, we have $Q \Longrightarrow \xrightarrow{\alpha} Q_2 \approx_{ccs^-} P'$, as needed. □

In general, the result stated in Proposition 2.20 turns out to be the best we can do to strengthen the simulation of a visible action, as opposed to the counterexample that distinguishes the strong and weak bisimilarities (see Lem. 2.6). Nonetheless, Proposition 2.20 can be refined further if one simply focuses on non-divergent processes. In that case, we end up with the strong bisimilarity, as Corollary 2.21 reveals. With this corollary in position, it makes sense to speculate that, if the replication operator were to be eliminated from CCS$^-$ (though this makes the calculus less interesting), then the weak bisimilarity would flatten onto the strong bisimilarity.

Corollary 2.21. *Assume P is not divergent. If $P \approx_{ccs^-} Q$ and $P \xrightarrow{\alpha} P'$ where α is not τ, then $Q \xrightarrow{\alpha} \approx_{ccs^-} P'$.*

Proof. Since P is not divergent, neither is Q. By Proposition 2.20, we know that $Q \Longrightarrow Q_1 \xrightarrow{\alpha} Q_2 \approx_{ccs^-} P'$, where $Q \Longrightarrow Q_1$ has only state-preserving τ actions. However, Q is not divergent. This means that the τ actions in $Q \Longrightarrow Q_1$ can only be state-changing. This is a contradiction, which leads to the only possibility that $Q \Longrightarrow Q_1$ contains zero τ actions, *i.e.*, Q_1 is Q. Thus we are done. □

Now, Proposition 2.19 and Proposition 2.20 amount to Theorem 2.7, the main result.

Proof of Theorem 2.7. We define

$$\mathscr{R} \stackrel{\text{def}}{=} \{(P, Q) \mid P \approx_{ccs^-} Q\}.$$

We show that $\mathscr{R}$ is a quasi-strong bisimulation. Assume $P \mathscr{R} Q$. We have two cases.

- $P \xrightarrow{\alpha} P'$ in which α is not τ. By Proposition 2.20, $Q \Longrightarrow \xrightarrow{\alpha} Q' \approx_{ccs^-} P'$. So we have $P' \mathscr{R} Q'$.
- $P \xrightarrow{\tau} P'$. By Proposition 2.19, $Q \xrightarrow{\tau} Q' \approx_{ccs^-} P'$. So we have $P' \mathscr{R} Q'$. □

2.3. Further results and discussion

In this section, we make some further discussion concerning the obtained results. We argue that Corollary 2.8 also holds for the branching bisimilarity [9, 10] (in place of the weak bisimilarity). This, in turn, would lead to the coincidence between the weak bisimilarity and the branching bisimilarity.

Indeed, it is interesting to exploit further Theorem 2.7, particularly the relationship with the branching bisimilarity [9, 10], a well-known equivalence relation on processes that preserves the branching structure of processes. The definition of branching bisimulation is as follows.

Definition 2.22. A symmetric binary relation $\mathscr{R}$ on CCS$^-$ processes is a branching bisimulation if it is divergence-sensitive, and whenever $P \mathscr{R} Q$, the following properties hold.

- if $P \xrightarrow{\alpha} P'$, then either
  - α is τ and $P' \mathscr{R} Q$; Or
  - $Q \Longrightarrow Q'' \xrightarrow{\alpha} Q'$, $P \mathscr{R} Q''$ and $P' \mathscr{R} Q'$.

Two processes P and Q are branching bisimilar, notation $P \approx^{br}_{ccs^-} Q$, if there exists some branching bisimulation $\mathscr{R}$ such that $P \mathscr{R} Q$.

We call $\approx^{br}_{ccs^-}$ the branching bisimilarity. To be consistent with the current setting, we impose divergence-sensitiveness on the branching bisimulation. It is not hard to see that the branching bisimilarity implies the weak bisimilarity, *i.e.*, $\approx^{br}_{ccs^-} \subseteq \approx_{ccs^-}$.

Here comes an important observation of the proof of Theorem 2.7. It uses Proposition 2.20, which states that the internal actions in $Q \Longrightarrow \xrightarrow{\alpha} Q' \approx_{ccs^-} P'$ in the first clause of the proof of Theorem 2.7 are actually state-preserving. Following this observation, we can strengthen the definition of quasi-strong bisimulation without changing any distinguishing power.

Definition 2.23. A symmetric binary relation $\mathscr{R}$ on CCS$^-$ processes is a quasi-strong branching bisimulation if it is divergence-sensitive, and whenever $P \mathscr{R} Q$, the following properties hold.

- if $P \xrightarrow{\alpha} P'$ and α is not τ, then $Q \Longrightarrow Q'' \xrightarrow{\alpha} Q'$, $P \mathscr{R} Q''$ and $P' \mathscr{R} Q'$.
- if $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} Q'$ and $P' \mathscr{R} Q'$.

Two processes P and Q are quasi-strongly branching bisimilar, notation $P \sim^{qb}_{ccs^-} Q$, if there exists some quasi-strong branching bisimulation $\mathscr{R}$ such that $P \mathscr{R} Q$.

Through the same proof routine as that of Theorem 2.7, we can infer that $\sim^{qb}_{ccs^-}$ also coincides with the weak bisimilarity.

Lemma 2.24. *In* CCS$^-$*, it holds that* $\approx_{ccs^-} = \sim^{qb}_{ccs^-}$.

Now examining the difference between the quasi-strongly branching bisimulation and the branching bisimulation, it is straightforward to see that both of the clauses of the quasi-strongly branching bisimulation imply that of the branching bisimulation. Hence the following lemma.

Lemma 2.25. *In* CCS$^-$*, it holds that* $\sim^{qb}_{ccs^-} \subseteq \approx^{br}_{ccs^-}$.

Proof. To show that $\sim^{qb}_{ccs^-}$ is a branching bisimulation, we focus on the τ-clause, since it is the only distinct part. In particular, the second clause of the quasi-strongly branching bisimulation implies that of the branching bisimulation because we can rewrite $Q \xrightarrow{\tau} Q'$ as $Q \Longrightarrow Q \xrightarrow{\tau} Q'$. □

The lemmas above lead to the follow-up corollary.

Corollary 2.26. *In* CCS$^-$*, it holds that* $\approx_{ccs^-} = \approx^{br}_{ccs^-}$.

Proof. By Lemma 2.25, we have $\sim^{qb}_{ccs^-} \subseteq \approx^{br}_{ccs^-} \subseteq \approx_{ccs^-}$. Then the equality follows by Lemma 2.24. □

To conclude, all the discussion so far boils down to the next theorem.

Theorem 2.27. *In* CCS$^-$*, it holds that* $\approx_{ccs^-} = \sim^{q}_{ccs^-} = \sim^{qb}_{ccs^-} = \approx^{br}_{ccs^-}$.

We make some remarks before ending this section. As mentioned, in spirit of Corollary 2.21 and Proposition 2.20, if CCS$^-$ is further deprived of the replication, we believe that the weak bisimilarity would fall onto the strong bisimilarity. This is virtually not hard to verify, by means of going through all the analysis above but ignoring those parts concerning the replication. Indeed, in a CCS variant with neither replication nor restriction, it appears obvious that the strong and weak bisimilarities would coincide. The intuition is that in such a subcalculus, the number of visible actions (and thus the τ actions since it is not among the prefixes) that the two compared processes can perform respectively must be finite. That being said, we notice that such coincidence would fail again if one assumes that τ is a primitive prefix like input and output. We do not extend the discussion into details here, as a calculus with neither restriction nor replication appears not very interesting.

Moreover, it might be intriguing to see if one can find a subcalculus of CCS in which the weak and strong bisimilarities coincide, *e.g.*, with a very special form of replication. That is, as a potential variant of CCS, consider a subcalculus in which all replications are of the form $!a.P$, and moreover the channel name a cannot occur elsewhere in input (but can occur in output). Such a subcalculus would probably need a type system to impose the constraints. Indeed, it would be likely that the proof approach here could be adapted to show the coincidence since, intuitively, now the number of output and the corresponding interaction it can trigger are finite and can be matched in a strong fashion. In addition, this subcalculus would potentially be useful in certain cases that only require infinite input behavior (*e.g.*, online service for logging that needs to run 7×24 hours with only input). Also to this point, the analysis and the corresponding result on quasi-strong bisimilarity may offer some potential tool for analyzing finite-state processes in a broader field.

3. On the bisimulation equality in HOCCS$^-$

In this section, we define HOCCS$^-$, *i.e.*, HOCCScore without the restriction operation, and then discuss the relationship between the strong and weak bisimilarities.

3.1. Calculus HOCCS$^-$

A HOCCS$^-$ process is given by the following grammar. We denote names by lowercase letters, processes by uppercase letters, and process variables by X, Y, Z.

$$P, P' ::= 0 \;\Big|\; X \;\Big|\; a(X).P \;\Big|\; \overline{a}P'.P \;\Big|\; P \,|\, P'$$

The operators have their standard meaning: input prefix $a(X).P$, output prefix $\overline{a}P'.P$, and parallel composition $P \,|\, P'$. We stipulate that parallel composition has the least precedence.

A process variable X occurring in P is bound by input-prefix $a(X).P$ and free otherwise. We use fpv$(\cdot)$, bpv$(\cdot)$, pv$(\cdot)$ respectively to denote free process variables, bound process variables, and process variables in a set of processes. In addition, we use n$(\cdot)$ to denote the names in a set of processes. A name or process variable is fresh if it does not appear in the processes under consideration. Closed processes are those having no free variables and are considered by default. As usual, we use $a.0$ as a shortcut for $a(X).0$, and $\overline{a}.0$ for $\overline{a}0.0$; moreover, the trailing 0 is often omitted. A tilde $\widetilde{\ }$ represents a tuple. A higher-order substitution $P\{A/X\}$ replaces free occurrences of variable X with A, and can be extended to tuples in the expected entry-wise way.

A context C, or $C[\cdot]$ to emphasize the hole in it, is a process with some subprocess replaced by the hole $[\cdot]$, and $C[A]$ is the process obtained by substituting A for the hole. We denote by $E[\widetilde{X}]$ the process expression E (possibly) with free occurrence of the variables $\widetilde{X}$, and $E[\widetilde{A}]$ stands for $E\{\widetilde{A}/\widetilde{X}\}$. Essentially, $E[X]$ can be treated as a multihole context [2], and sometimes we also write $E[\cdot]$ in the follow-up discussion.

The semantics of HOCCS$^-$ (on closed processes) is as follows. The symmetric rules are omitted.

$$\frac{}{a(X).P \xrightarrow{a(A)} P\{A/X\}} \qquad \frac{}{\overline{a}A.P \xrightarrow{\overline{a}A} P}$$

$$\frac{P \xrightarrow{\lambda} P'}{P\,|\,Q \xrightarrow{\lambda} P'\,|\,Q} \qquad \frac{P \xrightarrow{a(A)} P' \quad Q \xrightarrow{\overline{a}A} Q'}{P\,|\,Q \xrightarrow{\tau} P'\,|\,Q'}$$

We denote by α, λ the actions: internal move (τ), higher-order input ($a(A)$), and higher-order output ($\overline{a}A$). Sometimes for clarity, we may write $\overline{a}[A]$ for higher-order output. Operations fpv($\cdot$), bpv($\cdot$), pv($\cdot$), n($\cdot$) can be similarly defined on actions. As usual, $\Longrightarrow$ is the reflexive transitive closure of internal actions, and $\stackrel{\lambda}{\Longrightarrow}$ is $\Longrightarrow\xrightarrow{\lambda}\Longrightarrow$. Also, $\stackrel{\hat{\lambda}}{\Longrightarrow}$ is $\Longrightarrow$ when λ is τ and $\stackrel{\lambda}{\Longrightarrow}$ otherwise. We use $\xrightarrow{\tau}_k$ to mean k consecutive τ's. We denote by $\equiv$ the standard structural congruence [2, 11]. It is the smallest equivalence relation satisfying α-convertibility over (bound) process variables, the monoid laws, and the commutative laws for parallel composition. That is,

$$a(X).P \equiv a(Y).P\{Y/X\} \quad (Y \text{ fresh}) \qquad P\,|\,0 \equiv P$$

$$P\,|\,(Q\,|\,R) \equiv (P\,|\,Q)\,|\,R \qquad P\,|\,Q \equiv Q\,|\,P$$

As is well known, replication can be somehow encoded in HOCCS$^-$[2, 8, 12]. That is, we can define

$$!P \stackrel{\mathrm{def}}{=} \overline{c}Q_{c,P}\,|\,Q_{c,P}, \qquad Q_{c,P} \stackrel{\mathrm{def}}{=} c(X).(\overline{c}X\,|\,X\,|\,P) \qquad (c \text{ fresh})$$

Sometimes, the name c used to achieve such a replication is referred to as a replicator name. In addition, we can also define the so-called guarded replication for a prefix ϕ (input or output).

$$!^g\,\phi.P \stackrel{\mathrm{def}}{=} \overline{c}Q_{c,\phi,P}\,|\,Q_{c,\phi,P},$$

$$Q_{c,\phi,P} \stackrel{\mathrm{def}}{=} c(X).(\phi.(\overline{c}X\,|\,X\,|\,P))$$

We remark that the definition of the replication in HOCCS$^-$does not give the real equivalent for the (original) primitive replication, due to the absence of the restriction operation. What is distinct is that it may have visible actions over the auxiliary fresh name c. In fact, it is a weak version of replication that can be defined in HOCCS$^-$ at best. In particular, it produces infinite behavior in HOCCS$^-$with neither primitive replication nor restriction, leveraging the expressiveness of higher-order concurrency.

Context bisimulation

For HOCCS$^-$, we have the following standard notion of context bisimulation [13, 14].

Definition 3.1 (Context bisimulation)**.** A symmetric relation $\mathscr{R}$ on (closed) HOCCS$^-$ processes is a (weak) context bisimulation (respectively strong context bisimulation), if $P \mathscr{R} Q$ implies the following properties:

1. if $P \xrightarrow{\alpha} P'$ in which α is $a(A)$ or τ, then $Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$ (respectively $Q \xrightarrow{\alpha} Q'$) for some Q' and $P' \mathscr{R} Q'$.

2. if $P \xrightarrow{\overline{a}A} P'$, then $Q \overset{\overline{a}B}{\Longrightarrow} Q'$ (respectively $Q \xrightarrow{\overline{a}B} Q'$) for some B and Q', and for every $E[X]$ it holds that

$$E[A] \,|\, P' \,\mathscr{R}\, E[B] \,|\, Q' \qquad (*)$$

The (weak) context bisimilarity (respectively strong context bisimilarity), denoted by $\approx_{hoccs^-}$ (respectively $\sim_{hoccs^-}$), is the largest context bisimulation (respectively strong context bisimulation).

It is well-known that both $\approx_{hoccs^-}$ and $\sim_{hoccs^-}$ are congruences [2, 8, 13, 14]. Relation $\approx_{hoccs^-}$ (similar for $\sim_{hoccs^-}$) can be extended to open processes in the usual way: suppose $\widetilde{X} = \mathrm{fpv}(P, P')$, then $P \approx_{hoccs^-} P'$ if and only if $P\{\widetilde{A}/\widetilde{X}\} \approx_{hoccs^-} P'\{\widetilde{A}/\widetilde{X}\}$ for all closed $\widetilde{A}$.

It should be clear that by the definitions, the following implications are true (see [2, 13]).

Lemma 3.2. *It holds that* $\equiv \subseteq \sim_{hoccs^-} \subseteq \approx_{hoccs^-}$.

3.2. The relationship between the weak and strong context bisimilarities in HOCCS$^-$

The following conjecture states that the weak context bisimilarity is no coarser than the strong context bisimilarity.

Conjecture 3.3. *On (closed)* HOCCS$^-$ *process, we have* $\approx_{hoccs^-} \subseteq \sim_{hoccs^-}$.

Informal account. We have Conjecture 3.3 because, unlike CCS$^-$, it does not appear possible to unveil a difference between the weak and strong bisimilarities in HOCCS$^-$, whose (derived) replication is not perfect without the restriction operation. To see this, we argue that $!P$ is not strongly context bisimilar to $!P \mid P$. For convenience, we reproduce the definition of replication as follows.

$$!P \overset{\text{def}}{=} \overline{c}Q_{c,P} \,|\, Q_{c,P}, \qquad\qquad Q_{c,P} \overset{\text{def}}{=} c(X).(\overline{c}X \,|\, X \,|\, P)$$

Now, suppose that P can make an action $P \xrightarrow{\lambda'} P'$ other than those over c, *e.g.*, $P \overset{\text{def}}{=} \overline{f}0.0$. Then the action $!P \,|\, P \xrightarrow{\lambda'} !P \,|\, P'$ cannot be matched by $!P$ because $!P$ can only fire immediate visible actions over c. So we conclude that $!P \not\sim_{hoccs^-} !P \,|\, P$.

As a matter of fact, $!P$ is even not weakly context bisimilar to $!P \mid P$, as exhibited in the following. First, it should be clear that an action by $!P \,|\, P$, say $!P \,|\, P \xrightarrow{\lambda} T$ where λ can be τ or visible, can be matched by $!P \xrightarrow{\tau} !P \,|\, P \xrightarrow{\lambda} T$. However, the simulation in the opposite direction does not hold. Specifically, taking P as $\overline{f}0.0$ and A as $\overline{g}0.0$, we have the only possible simulation as follows, if $!P$ makes an input over c.

$$\begin{array}{lcl} !P & \xrightarrow{c(A)} & \overline{c}Q_{c,P} \mid \overline{c}A \mid A \mid P \overset{\text{def}}{=} R_1 \\ & \text{must be simulated by} & \\ !P \mid P & \xrightarrow{c(A)} & \overline{c}Q_{c,P} \mid \overline{c}A \mid A \mid P \mid P \overset{\text{def}}{=} R_2 \end{array}$$

Then $R_2 \xrightarrow{\overline{f}0} \xrightarrow{\overline{f}0}$ which R_1 cannot simulate because R_1 only has one action over f. So we conclude that $!P \not\approx_{hoccs^-} !P \,|\, P$.

Essentially, the crux here is that the 'replication' may fire actions over the auxiliary name c, which is visible due to the absence of the restriction operator. Currently, we believe it quite likely that $\approx_{hoccs^-}$ collapses onto $\sim_{hoccs^-}$, but this would be conceivably hard to prove and calls for potentially new techniques. A primary reason is that higher-order communication and, accordingly, the context bisimulation is intrinsically different from first-order ones. We thus leave this to be settled in the future. We also remark that the first inclusion of

Lemma 3.2 is strict. That is, $\equiv \subsetneq \sim_{hoccs^-}$, where the *difference* between $\equiv$ and $\sim_{hoccs^-}$ results from some distributive law [15]; see [6] for more details.

4. Conclusion

This paper has been focusing on the relationship between the strong and the weak bisimilarities, in process models from which the restriction operator is removed. We have presented a few observations about such relationship in both a first-order model (CCS^-) and a higher-order model (HOCCS^-). Basically, it is shown invariant in CCS^- that the weak bisimilarity remains strictly weaker than the strong bisimilarity, even without the restriction's capacity of hiding information. Essentially, this is a consequence of the replication operation. The situation is different in HOCCS^-, mainly because replication is primitive in the former but weakly encodable in the latter.

Our work demonstrates that, compared with the restriction operation, the replication operation can also 'conceal' information, but in a distinct way: by generating an abundance of identical processes. In other words, such an abundance of identical processes renders it possible to maintain the state of a process. This insight allows us to narrow the gap between the strong and weak bisimilarities in CCS^- by defining a "quasi-strong" bisimilarity. Formally, this relation brings the weak bisimilarity closer to the strong one by means of strengthening the weak bisimilarity in two ways: (1) It enforces strong bisimulation for silent actions (*i.e.*, τ actions). (2) It disregards trailing silent actions when matching a visible action. Crucially, we prove that the quasi-strong bisimilarity is equivalent to the weak bisimilarity. This coincidence indicates that the original weak bisimilarity can be reinforced with stricter simulation requirements without altering its expressive power. Moreover, it confirms that the absence of the restriction operator is not incidental; we can leverage this to make weak bisimilarity more tractable. As a significant corollary, we also establish the coincidence of the branching bisimilarity with the weak bisimilarity.

There are some questions worthy of further investigation. A first one is to prove or disprove the conjecture, aforementioned in Section 3, that in HOCCS^- the weak bisimilarity collapses onto the strong bisimilarity, and/or that the strong context bisimilarity collapses onto the structural congruence. One more direction is to exploit more (process) models without the restriction operator, *e.g.*, value-passing models or (higher-order) ambient models, for properties that stem from the absence of the restriction operator.

Acknowledgments

This work has been supported by project NSF of China (62572319,62072299,61872142) and Shanghai "Science and Technology Innovation Action Plan" Special project for key technologies of blockchain (24BC3200500,24BC3200300). We thank the anonymous referees for their insightful comments and constructive suggestions. We also acknowledge the support from Shirong Ma and other colleagues at ECUST and SJTU.

Funding

This research was funded by National Natural Science Foundation of China (62572319,62072299,61872142) and Shanghai "Science and Technology Innovation Action Plan" Special project for key technologies of blockchain (24BC3200500,24BC3200300).

Conflicts of interest

The authors have nothing to disclose.

Data availability statement

This article has no associated data generated and/or analyzed.

Author contribution statement

The author was solely responsible for the conception, design, analysis, and writing of this study in its entirety.

References

[1] D. Sangiorgi, Concurrency theory: timed automata, testing, program synthesis. *Distrib. Comput.* **25** (2012) 3–4.

[2] D. Sangiorgi and D. Walker, The Pi-calculus: A Theory of Mobile Processes. Cambridge University Press (2001).

[3] R. Milner, Communication and Concurrency. Prentice Hall (1989).

[4] D. Sangiorgi, Introduction to Bisimulation and Coinduction. Cambridge University Press (2011).

[5] Y. Fu, Theory of interaction. *Theor. Comput. Sci.* **611** (2015) 1–49.

[6] I. Lanese, J.A. Pérez, D. Sangiorgi and A. Schmitt, On the expressiveness and decidability of higher-order process calculi. *Inf. Computat.* **209** (2011) 198–226.

[7] D. Hirschkoff and D. Pous, On bisimilarity and substitution in presence of replication, in *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming (ICALP 2010)*, *LNCS* 6199 (2010) 454–465.

[8] B. Thomsen, Plain CHOCS, a second generation calculus for higher-order processes. *Acta Inf.* **30** (1993) 1–59.

[9] R.J. van Glabbeek and W.P. Weijland, Branching time and abstraction in bisimulation semantics (extended abstract), in: *Proceedings of the 11th IFIP World Computer Congress, Information Processing '89* (1989) 613–618.

[10] R.J. van Glabbeek and W.P. Weijland, Branching time and abstraction in bisimulation semantics. *J. ACM* **43** (1996) 555–600.

[11] R. Milner, J. Parrow and D. Walker, A calculus of mobile processes (parts I and II). *Inf. Computat.* **100** (1992) 1–77.

[12] I. Lanese, J.A. Pérez, D. Sangiorgi and A. Schmitt, *On the expressiveness and decidability of higher-order process calculi*, in *Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS 2008)*. IEEE Computer Society (2008) 145–155.

[13] D. Sangiorgi, *Expressing Mobility in Process Algebras: First-order and Higher-order Paradigms.* PhD thesis, University of Edinburgh (1992).

[14] D. Sangiorgi, Bisimulation for higher-order process calculi. *Inf. Computat.* **131** (1996) 141–178.

[15] D. Hirschkoff and D. Pous, A distribution law for CCS and a new congruence result for the pi-calculus. *Logical Methods Comput. Sci.* **4** (2008) https://doi.org/10.2168/LMCS-4(2:4)2008.