On Bisimilarity and Substitution in Presence of Replication

Daniel Hirschkoff, Damien Pous

On Bisimilarity and Substitution in Presence of Replication*

(extended version)**

Daniel Hirschkoff[1] and Damien Pous[2]

[1] ENS Lyon, Université de Lyon, CNRS, INRIA
[2] CNRS, Laboratoire d'Informatique de Grenoble

Abstract. We prove a new congruence result for the π-calculus: bisimilarity is a congruence in the sub-calculus that does not include restriction nor sum, and features top-level replications. Our proof relies on algebraic properties of replication, and on a new syntactic characterisation of bisimilarity. We obtain this characterisation using a rewriting system rather than a purely equational axiomatisation. We then deduce substitution closure, and hence, congruence. Whether bisimilarity is a congruence when replications are unrestricted remains open.

1 Introduction

We study algebraic properties of behavioural equivalences, and more precisely, of strong bisimilarity ($\sim$). This has long been an important question in concurrency theory, with a particular focus on the search for axiomatisations of bisimilarity (see [1] for a survey). Our primary goal is to establish congruence results for the π-calculus [17]. At the heart of the π-calculus is the mechanism of *name-passing*, which is the source of considerable expressive power. Name-passing however introduces substitutions in the formalism, and these in turn lead to irregularities in the behavioural theory of processes: due to the input prefix, we need bisimilarity to be closed under substitutions for it to be a congruence.

To establish substitution closure, we exploit a new axiomatisation of bisimilarity. Several axiomatisation results for process calculi that feature an operator of parallel composition ($|$) have been derived by decomposing this operator using sum, and possibly left merge [6,5,1]. We, on the contrary, are interested in treating parallel composition as a primitive operator. One reason for this is that the sum operator is often absent from the π-calculus since it can be encoded [13],

* Work partially funded by the French ANR projects "Curry-Howard pour la Concurrence" CHOCO ANR-07-BLAN-0324 and COMPLICE ANR-08-BLANC-0211-01.

** Short version to appear in Proc. ICALP, vol. 6199 of LNCS, July, 2010. This version contains additional proofs, an appendix with the complete proofs about the π-calculus, and an appendix devoted to the extension of our results to the calculus with τ-prefixes.

under certain conditions. More importantly, this operator makes substitution closure fail [17,3], so that existing axiomatisations of bisimilarity in calculi featuring sum do not help when it comes to reason about congruence in the π-calculus.

In the present paper, we focus on properties of the replication operator [11], denoted by '!'. As [17,3] shows, bisimilarity is not substitution closed when both replication and name restriction are present in the calculus, and we have established in [8] that it is when we renounce to replication. To our knowledge, congruence of bisimilarity in the restriction-free π-calculus with replication is an open problem [17]; we provide here a partial answer.

Behavioural properties of replication. Replication is an "infinitary version" of parallel composition. Structural congruence traditionally contains the following *structural laws*: $!a.P \,|\, a.P \equiv !a.P$ and $!a.P \,|\, !a.P \equiv !a.P$ (given here for CCS), so that a replicated process acts as an unbounded number of parallel copies of that process. A contribution of this work is an analysis of *behavioural laws* capturing other properties of replication. For example, for any context C, we have

$$!a.P \,|\, C[a.P] \sim !a.P \,|\, C[\mathbf{0}] \qquad \text{and} \qquad !a.C[a.C[\mathbf{0}]] \sim !a.C[\mathbf{0}] \ .$$

($C[Q]$ stands for the process obtained by replacing the hole with Q in C, and $\mathbf{0}$ is the inactive process.) The left-hand side law is a generalisation the first structural congruence law: a replicated process can erase one of its copies arbitrarily deep in a term. The right-hand side law is more involved: read from right to left, it shows that a replicated process is able to replicate itself. Its most trivial instance is $!a.a \sim !a$.

Although the above laws are the basic ingredients we need in order to characterise bisimilarity in our setting, they do not form a complete axiomatisation of bisimilarity, as the following example shows:

$$P_1 = !a.(b|a.c) \mid !a.(c|a.b) \ \sim \ !a.b \mid !a.c = P_2 \ .$$

P_1 can be obtained from P_2 by inserting a copy of $a.b$ inside $!a.c$, and, symmetrically, a copy of $a.c$ inside $!a.b$. It seems reasonable to consider P_2 as a kind of normal form of P_1; however, P_1 and P_2 cannot be related using the above laws. Describing this phenomenon of "mutual replication" in all its generality leads to complicated equational schemata, so that we take another approach.

Overview. Our first contribution is a syntactic characterisation of bisimilarity on a fragment of CCS with top-level replications. This characterisation relies on a rewriting system for processes (such that P_1 above rewrites into P_2). An important technical notion we need to introduce is that of *seed*: a seed of P is a process bisimilar to P of minimal size; for example, P_2 is a seed of P_1. Our proof goes by characterising bisimilarity on seeds, and establishing that any process P can be rewritten into a seed of P.

Our second contribution is congruence of bisimilarity in the corresponding fragment of the π-calculus. Concretely, we prove that bisimilarity is substitution closed by considering *visible* bisimilarity (sometimes called *io*-bisimilarity [10]),

the equivalence obtained by disallowing challenges along internal communications. Visible bisimilarity is inherently substitution closed, and our characterisation allows us to show that it coincides with bisimilarity.

Since the technical developments that lead to congruence in the π-calculus follow to a large extent the path of our proofs for CCS, we moved them to the appendix. On the contrary, we provide detailed proofs and present most intermediate steps for CCS. We indeed view the reasonings we use in our proofs as an important contribution of this work. In particular, we make use of both algebraic and coinductive reasoning, notably using "up-to techniques" for bisimulation [16,14,15].

Outline. We describe the subset of CCS we work with and we prove general properties of replication in Sect. 2. In Sect. 3, we introduce the notion of seed, and give a characterisation of bisimilarity on seeds. The rewriting system is defined in Sect. 4, where we show that any process can be rewritten into a seed, and where we characterise strong bisimilarity. We present our new congruence result for the π-calculus in Sect. 5. Section 6 suggests directions for future work.

2 General Setting, and Properties of Replication

We let a, b range over a countable set of *names*; we work in a fragment of CCS which we call mCCS (pronounced 'miniCCS'), defined by the following grammar:

$$\begin{array}{rcl@{\quad}rcl@{\quad}r} \alpha,\beta & ::= & a \mid \overline{a} & \mu & ::= & \alpha \mid \tau & \text{(actions and labels)} \\ E,F & ::= & \mathbf{0} \mid \alpha.F \mid F|F & P,Q & ::= & F \mid !\alpha.F \mid P|P & \text{(processes)} \\ D & ::= & [] \mid \alpha.D \mid D|F & C & ::= & D \mid !\alpha.D \mid C|P & \text{(contexts)} \end{array}$$

This calculus features no restriction, no sum, and allows only top-level replicated prefixes. Note that the τ prefix is not included in the syntax, and only appears in *labels* (μ); we return to this point in Rmk. 24. We use P, Q to range over processes; according to the syntax, a *finite* process (F) is a process which does not contain an occurrence of the replication operator ($!\alpha.$). We omit trailing occurrences of $\mathbf{0}$, and write, e.g., $\alpha.\beta$ for $\alpha.\beta.\mathbf{0}$. We shall sometimes write $\prod_{i\in[1..k]} \alpha_i.F_i$ for $\alpha_1.F_1 \mid \ldots \mid \alpha_k.F_k$. We extend the syntactic operator of replication to a function defined over processes by letting

$$!\mathbf{0} \triangleq \mathbf{0} \qquad !(P|Q) \triangleq !P|!Q \qquad !!\alpha.F \triangleq !\alpha.F \ .$$

In particular, $!F$ will always denote a process having only replicated (parallel) components. We let C range over single-hole *contexts*, mapping finite processes to processes, and similarly for *finite contexts*, ranged over using D. Note that the hole cannot occur immediately under a replication in C.

The following labelled transition system (LTS) for mCCS is standard (we omit symmetric rules for parallel composition).

$$\frac{}{\alpha.F \xrightarrow{\alpha} F} \qquad \frac{}{!\alpha.F \xrightarrow{\alpha} !\alpha.F \mid F} \qquad \frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \qquad \frac{P \xrightarrow{\overline{a}} P' \qquad Q \xrightarrow{a} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

This LTS yields the following standard notion of *bisimilarity*, ($\sim$). We also define *visible bisimilarity* ($\dot{\sim}$), where silent transitions are not taken into account.

Definition 1 Strong bisimilarity ($\sim$) *is the largest symmetric binary relation over processes such that whenever* $P \sim Q$ *and* $P \xrightarrow{\mu} P'$*, there exists* Q' *such that* $P' \sim Q'$ *and* $Q \xrightarrow{\mu} Q'$*.* Visible bisimilarity ($\dot{\sim}$) *is defined similarly, by restricting challenges to the cases where* $\mu \neq \tau$*.*

Both bisimilarities are congruences. They are moreover preserved by the extended replication function, and we have $\sim \subseteq \dot{\sim}$. On finite processes, bisimilarity and visible bisimilarity coincide and can be characterised using the following *distribution law*, where there are as many occurrences of F on both sides [8]:

$$\alpha.(F|\alpha.F|\ldots|\alpha.F) \;\sim\; \alpha.F|\alpha.F|\ldots|\alpha.F \;. \tag{D}$$

We now present some important properties of replicated processes w.r.t. bisimilarity. The following proposition allows us to obtain the two laws from the introduction, that involve copying replicated sub-terms.

Proposition 2 *If* $C[\mathbf{0}] \sim !\alpha.F|P$*, then* $C[\mathbf{0}] \sim C[\alpha.F]$*.*

Proof. We show that $\mathcal{R}$= $\{(C[\mathbf{0}], C[\alpha.F])$ / $\forall C$ s.t. $C[\mathbf{0}] \sim !\alpha.F|P$ for some $P\}$ is a bisimulation up to transitivity [14,15]. There are several cases to consider:

- the hole occurs at top-level in the context ($C = []|Q$) and the right-hand side process does the following transition: $C[\alpha.F] \xrightarrow{\alpha} F|Q$. By hypothesis, $Q \sim !\alpha.F|P$ so that we find Q' such that $Q \xrightarrow{\alpha} Q'$ and $Q' \sim !\alpha.F|F|P$. Injecting the latter equality gives $Q' \sim Q|F$, so that Q' closes the diagram.
- the hole occurs under a replicated prefix of the context ($C = !\beta.D|Q$) that is fired: we have $C[\mathbf{0}] \xrightarrow{\beta} P_l = C[\mathbf{0}]|D[\mathbf{0}]$ and $C[\alpha.F] \xrightarrow{\beta} P_r = C[\alpha.F]|D[\alpha.F]$. This is where we reason up to transitivity: these processes are not related by $\mathcal{R}$ (we work with single-hole contexts), but we have $P_l \;\mathcal{R}\; P_c \;\mathcal{R}\; P_r$, for $P_c = C[\mathbf{0}]|D[\alpha.F]$, using contexts $C_{lc} = C[\mathbf{0}]|D$ and $C_{cr} = C|D[\alpha.F]$.
- the hole occurs under a non-replicated prefix in the context ($C = \beta.D|Q$), or the context triggers a transition that does not involve or duplicate the hole; it suffices to play the bisimulation game.
- we are left with the cases where a synchronisation is played; they can be handled similarly (in particular because contexts have a single hole). □

As a consequence, we obtain the validity of the following laws. We shall see in the sequel that together with the distribution law (D), they capture the essence of bisimilarity in our calculus.

$$!\alpha.F \mid C[\alpha.F] \;\sim\; !\alpha.F \mid C[\mathbf{0}] \tag{A}$$

$$!\alpha.D[\alpha.D[\mathbf{0}]] \;\sim\; !\alpha.D[\mathbf{0}] \tag{A$'$}$$

(Note that Prop. 2 and the above laws hold for full CCS and for the π-calculus, as long as the hole does not occur as argument of a sum in C and D, and C and D do not bind names occurring in $\alpha.F$.)

We now recall a result from [12]:

Definition 3 (Prime process) *A process P is* prime *if $P \not\sim \mathbf{0}$ and $P \sim P_1|P_2$ entails $P_1 \sim \mathbf{0}$ or $P_2 \sim \mathbf{0}$ for all P_1, P_2. When $P \sim P_1 \mid \ldots \mid P_n$ where the P_is are prime, we shall call $P_1 \mid \ldots \mid P_n$ a* prime decomposition *of P.*

Theorem 4 (Decomposition of finite processes [12]) *Any finite process admits a prime decomposition, which is unique up to strong bisimilarity.*

Proof. See [12, Theorem 4.3.1]. □

Corollary 5 (Cancellation) *If $E \mid F \sim E' \mid F$ then $E \sim E'$.*

Note that $!a \sim !a \mid !a$ so that the the restriction to finite processes is essential in the above results. For processes with replications, we have the following rather different cancellation properties; we state them for visible bisimilarity, note that they are actually also valid for bisimilarity.

Lemma 6 *If $!F \dot{\sim} P|Q$, then $!F \dot{\sim} !F|P$.*

Proof. We reason purely algebraically: we replicate both sides of $!F \dot{\sim} P|Q$, and add P in parallel (since $!P \dot{\sim} !P|P$): this gives $!F \dot{\sim} !P|!Q \dot{\sim} !P|!Q|P$. We deduce $!F \dot{\sim} !F|P$ by injecting the first equivalence into the second one. □

Proposition 7 *If $!F|F_0 \dot{\sim} !E|E_0$ with F_0, E_0 finite, then $!F \dot{\sim} !E$.*

Proof. By emptying F_0 on the left-hand side[1], we find a finite process E_1 such that $!F \dot{\sim} !E|E_1$ $(*)$. Similarly, by emptying E_0 on the right-hand side we find F_1 such that $!F|F_1 \dot{\sim} !E$ $(**)$. By injecting the former equivalence in the latter, we have $!E|E_1|F_1 \dot{\sim} !E$ $(\dagger)$. By Lemma 6, $(**)$ gives $!E \dot{\sim} !E|F_1$, that we can inject into $(*)$ to obtain $!E|E_1|F_1 \dot{\sim} !F$. We finally deduce $!E \dot{\sim} !F$ from $(\dagger)$. □

Again, these properties are not specific to the subset of CCS we focus on: Prop. 7 holds provided that both F_0 and E_0 can be reduced to the empty process using transitions (this is the case, e.g., for the *normed* processes of [9]). The counterpart of this cancellation property does not hold; the replicated parts of bisimilar processes cannot always be cancelled: we cannot deduce $a \dot{\sim} \mathbf{0}$ from $!a|a \dot{\sim} !a|\mathbf{0}$.

3 Seeds

Definition 8 (Size, seed) *The* size of *P, noted $\sharp P$, is the number of prefixes in P. A* seed of *P is a process Q of least size such that $P \dot{\sim} Q$, whose number of replicated components is largest among the processes of least size. When P is a seed of P, we simply say that P is a seed.*

We show how to rewrite an arbitrary process into a seed in Sect. 4; in this section, we give a characterisation of bisimilarity on seeds (Prop. 16).

[1] In the present case, "emptying F_0" means playing all prefixes of F_0 in the bisimulation game between $!F|F_0$ and $!E|E_0$. We shall reuse this terminology in some proofs below; note that this is possible only with finite processes.

Definition 9 (Distribution congruence) *We call* distribution congruence *the smallest congruence relation $\equiv$ that satisfies the laws of an abelian monoid for $(|, \mathbf{0})$ and the distribution law (D).*

Fact 10 *We have $\equiv \subseteq \sim \subseteq \dot{\sim}$; the latter equivalence is substitution closed; on finite processes, the three relations coincide.*

Proof. The inclusions and the substitution closure of $\dot{\sim}$ are straightforward. On finite processes, $\equiv = \sim$ was proved in [8], and one can deduce from other results therein that $\dot{\sim} \subseteq \sim$ (a proof is given for π in appendix—Thm. A4). □

It is easy to show that distribution congruence is decidable, and only relates processes having the same size. In the sequel, we always work modulo distribution congruence[2]. We shall prove that on seeds, bisimilarity actually coincides with distribution congruence. Thanks to Prop. 7, the replicated parts of bisimilar seeds are necessarily bisimilar. As a consequence, in the remainder of this section, we fix a seed S having only replicated components: $S = \prod_i !\alpha_i.S_i$, and we study processes obtained by composing S with finite processes.

Definition 11 (Clean process, residual) *A finite process F is* clean w.r.t. *S, written $S\#F$, if F does not contain a sub-term of the form $\alpha_i.S_i$: for all i and finite context D, $F \not\equiv D[\alpha_i.S_i]$.*

A finite process R is a residual of *S, written $S \rightsquigarrow R$, when there exist $k > 0$, $\beta_1, .., \beta_k$, and $P_1, .., P_k$ such that $S \xrightarrow{\beta_1} P_1 \ldots \xrightarrow{\beta_k} P_k \equiv S|R$. We shall use R to range over such residual processes.*

Residuals and clean processes are stable under transitions (we shall use this property implicitely in the sequel), the finite part of a seed is necessarily clean:

Lemma 12 *(i) If $F \xrightarrow{\alpha} F'$ and $S\#F$ then $S\#F'$;*
(ii) If $F \xrightarrow{\alpha} F'$ $S \rightsquigarrow F$ then $S \rightsquigarrow F'$;
(iii) If $S|F$ is a seed, then $S\#F$.

Proof. We first remark that the $\alpha_i.S_i$ are prime: otherwise, we could construct a seed of S with the same size, but strictly more parallel components.

(i) Write $F \equiv \alpha.F_0|F_1 \xrightarrow{\alpha} F_0|F_1 \equiv F'$. By contrapositive, suppose that F' is not clean, i.e., $F' \equiv D[\alpha_i S_i]$ for some i, D. Since $\alpha_i.S_i$ is prime, it necessarily appears either in F_0 or in F_1, so that F cannot be clean.
(ii) Straigthforward.
(iii) By contradiction: if $F \equiv D[\alpha_i.S_i]$, then $S|F \sim S|D[\mathbf{0}]$ by law (A), which contradicts the minimality hypothesis about $S|F$. □

Also note that any residual is a parallel composition of sub-terms of the S_is:

Lemma 13 *If $S \rightsquigarrow \alpha.R$ with $\alpha.R$ prime, then $S_i \equiv D[\alpha.R]$ for some i, D.*

[2] This requires us to use the notion of prime process and prime decomposition at several places, to handle the distribution law (D) properly.

Proof. We prove that $S \xrightarrow{\beta_1} P_1 \ldots \xrightarrow{\beta_k} P_k \equiv S|R_0|R_1$ with R_0 prime entails that $S_i \equiv D[R_0]$ for some i, D, by induction on k. If $k = 0$ then $R_0 \equiv R_1 \equiv \mathbf{0}$, R_0 is not prime. For $k > 0$, assume that $S \xrightarrow{\beta_1} P_1 \ldots \xrightarrow{\beta_k} P_k \equiv S|R_0|R_1$ with R_0 prime. Necessarily, $P_{k-1} \equiv S|R$ with $S|R \xrightarrow{\beta_k} S|R_0|R_1$ for some R which we assume decomposed into primes. Since R_0 is prime, there are three cases to consider.

- Either R_0 is a parallel component of R, and we conclude by induction.
- Or $R \equiv \beta_k.(R_0|R_0')|R_1'$ for some R_0', R_1', with $\beta_k.(R_0|R_0')$ prime. By induction, there are i, D such that $S_i \equiv D[\beta_k.(R_0|R_0')]$; we take i and $D[\beta_k.[]|R_0']$.
- Or R_0 is a parallel component of a S_i (with $\alpha_i = \beta_k$), and we are done by taking i and the appropriate parallel context. □

(We need to suppose that $\alpha.R$ is prime in the above lemma because we work modulo distribution congruence—and in particular the distribution law (D): for example, if $S = !a.c|!b.c$, we have $S \rightsquigarrow c|c$ and hence $S \rightsquigarrow c.c$, and while c is a subterm of one of the S_i, $c.c$ is not.)

The following lemma summarises other properties about clean processes and residuals: a seed cannot absorb its non-trivial residuals (i), sub-terms of seeds are clean (ii):

Lemma 14 *(i) If $S \rightsquigarrow R$ and $S \dot{\sim} S|R$, then $R \equiv \mathbf{0}$.*
(ii) If $S \rightsquigarrow R$, then $S \# R$.

Proof. (i) Suppose by contradiction $R \equiv \alpha.R_0|R_1$, and chose $\alpha.R_0$ prime. Lemma 6 gives $S \dot{\sim} S|\alpha.R_0$, hence $S \dot{\sim} S|!\alpha.R_0$ $(*)$ by replicating all processes. Moreover, we have $S \rightsquigarrow \alpha.R_0$, by emptying R_1, so that Lemma 13 gives some i, D such that $S_i \equiv D[\alpha.R_0]$. Therefore, by $(*)$ and law (A), we obtain $S \dot{\sim} \prod_{j \neq i} !\alpha_j.S_j|!\alpha_i.D[\mathbf{0}]|!\alpha.R_0$. The latter process has the same size as S, but it has strictly more replicated components, which contradicts the fact that S is a seed (Def. 8).

(ii) By contradiction, suppose that $R \equiv D[\alpha_i.S_i]$. By emptying the prefixes of D, we have $S \rightsquigarrow \alpha_i.S_i$. Since $S \dot{\sim} S|\alpha_i.S_i$ for all i, this contradicts (i). □

The first point above leads to the following cancellation result:

Lemma 15 *If $S|F \dot{\sim} S|E$, $S \# F$, and $S \# E$, then $F \equiv E$.*

Proof. We prove the following stronger property, by induction on n: for all n, F, E such that $\sharp F, \sharp E \leq n$, $S \# F$, and $S \# E$, we have:

$$\begin{cases} (i) & \forall P, S|F \dot{\sim} P|E \text{ entails } \sharp E \leq \sharp F\,; \\ (ii) & S|F \dot{\sim} S|E \text{ entails } F \equiv E\,. \end{cases}$$

The case $n = 0$ is trivial; assume $n > 0$.

(i) Suppose $\sharp F < \sharp E$ by contradiction. By emptying F, we get P', E' such that $S \mathrel{\dot\sim} P'|E'$, with $0 < \sharp E' \leq \sharp E$. Write $E' = \alpha.E_0|E_1$, then $S \mathrel{\dot\sim} S|\alpha.E_0$ by Lemma 6, and $S|S_i \mathrel{\dot\sim} S|E_0$ for some i with $\alpha_i = \alpha$. Necessarily, $\sharp S_i \leq \sharp E_0$: otherwise, by emptying E_0, we would obtain a non empty residual R such $S|R \mathrel{\dot\sim} S$, which would contradict Lemma 14(i). Since $\sharp E_0 < \sharp E' \leq \sharp E \leq n$, we can use the induction hypothesis, so that $S_i \equiv E_0$, and hence $\alpha_i.S_i \equiv \alpha.E_0$, which contradicts $S\#E$.

(ii) By the above point, $\sharp F = \sharp E$. We show that $\mathcal{R} \triangleq \{(F, E)\} \cup \equiv$ is a visible bisimulation. If $F \xrightarrow{\alpha} F'$, then $S|F \xrightarrow{\alpha} S|F'$, and $S|E$ can answer this challenge: $S|E \xrightarrow{\alpha} S|E'$ with $S|F' \mathrel{\dot\sim} S|E'$. If the answer comes from E, we are done by induction: $\sharp E' = \sharp F' = \sharp F - 1 \leq n - 1$. Otherwise, i.e., if $S|F' \mathrel{\dot\sim} S|S_i|E$ for some i, we get a contradiction with (i): we would have $\sharp E \leq \sharp F' = \sharp E - 1$. Challenges of E are handled symmetrically. □

We can now characterise bisimilarity on seeds:

Proposition 16 *For all seeds P, P', $P \mathrel{\dot\sim} P'$ iff $P \sim P'$ iff $P \equiv P'$.*

Proof. By Fact 10, it suffices to show that $P \mathrel{\dot\sim} P'$ entails $P \equiv P'$. Write P and P' as $S|F$ and $S'|F'$, where S, S' are replicated processes. By Prop. 7, $S \mathrel{\dot\sim} S'$. Moreover, S and S' are necessarily seeds because P and P' are (hence the notation). Write $S \equiv \prod_{i \leq m} !\alpha_i.S_i$ and $S' \equiv \prod_{j \leq n} !\alpha'_j.S'_j$, play each prefix on the left-hand side and apply Lemma 15 to show that there exists a map $\sigma : [1..m] \to [1..n]$, such that $\alpha_i.S_i \equiv \alpha'_{\sigma i}.S'_{\sigma i}$ (recall that $S\#S_j$ by Lemma 14(ii)). This map is bijective: we could otherwise construct a smaller seed. Therefore, $S \equiv S'$. By Lemma 12(iii), $S\#F$ and $S'\#F'$, which allows us to deduce $F \equiv F'$, using Lemma 15. Finally, $P \equiv P'$. □

We conclude this section by the following remark: seeds are stable under transitions, so that they actually form a proper sub-calculus of mCCS.

Proposition 17 *If P is a seed and $P \xrightarrow{\mu} P'$, then P' is a seed.*

Proof. Write $P \equiv S|F$, where S is replicated. S is a seed since P is and we easily check that $P' \equiv S|F'$, with $S\#F'$. Now, let $S'|F''$ be a seed of P'. By Prop. 7, $S \mathrel{\dot\sim} S'$, so that $S \equiv S'$ by Prop. 16. We conclude with Lemma 15: $F' \equiv F''$, so that P' is indeed also a seed. □

4 Rewriting Processes to Normal Forms

By Prop. 16, the seed of a process P is unique up to distribution congruence ($\equiv$); in the sequel, we denote it by $\mathsf{s}(P)$. In this section, we show that the seed of a process can be obtained using a rewriting system. This entails two important properties of mCCS: visible and strong bisimilarity coincide and bisimilarity is closed under substitutions (i.e., bisimilar processes remain bisimilar when applying an arbitrary name substitution).

Definition 18 (Rewriting) *Any process T induces a relation between processes, written $\xrightarrow{T}$, defined by the following rules, modulo $\equiv$:*

$$\frac{T \equiv !\alpha.F \mid Q}{C[\alpha.F] \xrightarrow{T} C[\mathbf{0}]}\ (\text{R1}) \qquad \frac{}{!\alpha.F \mid !\alpha.F \mid P \xrightarrow{T} !\alpha.F \mid P}\ (\text{R2})$$

The reflexive transitive closure of $\xrightarrow{T}$ is written $\xrightarrow{T}{}^*$.

We give some intuitions about how the rewriting rules work. First, only the replicated part of T matters when rewriting with $\xrightarrow{T}$. Relation $\xrightarrow{T}$ is nevertheless defined for an arbitrary process T in order to facilitate the presentation of some results below.

Then, we observe that it is only sensible to rewrite P using $\xrightarrow{T}$ when T is a seed of P. This means in particular that the rewriting system does not provide a direct way to compute the seed of a process (since the seed has to be guessed). It is rather a procedure to check that some process T is a "simplification" of P— Lemma 19 below validates this intuition. Rule (R2) is rather self-explanatory. The rewriting rule (R1) is related to laws (A) and (A′); we illustrate its use by considering the following examples:

$$!a.b \mid !b \mid b.a \xrightarrow{!a|!b} !a.b \mid !b \mid b \xrightarrow{!a|!b} !a.b \mid !b \xrightarrow{!a|!b} !a \mid !b \qquad (1)$$

$$!a.(b \mid a.b) \xrightarrow{!a.b} !a.b \qquad (2)$$

$$!a.b \mid !b.a \xrightarrow{!a|!b} !a.b \mid !b \xrightarrow{!a|!b} !a \mid !b \qquad (3)$$

$$!a|!a.b \xrightarrow{!a|!b} !a|!a \xrightarrow{!a|!b} !a \qquad (4)$$

(1) The first example shows how (R1) can be used to "implement" law (A) and erase redundant sub-terms. At each rewrite step, a copy of a component of the seed (here, $!a|!b$) is erased. In the third rewriting step, simplification occurs in a replicated component.
(2) Law (A′) is applied: a replicated component can be "simplified with itself".
(3) This example illustrates how the rewriting system solves the problem we exposed in the introduction (processes P_1 and P_2), where two replicated components have to simplify each other: by guessing the seed ($!a|!b$), we are able to apply the two simplifications in sequence.
(4) Here, we make a wrong guess about the seed: when assuming that $!b$ is part of the seed, we can erase the prefix b in $!a.b$. However, at the end, we are not able to validate this assumption: $!b$ does not appear in the normal form.

Accordingly, we obtain the following correctness criterion:

Lemma 19 (Soundness) *If $P \xrightarrow{T}{}^* T$, then $P \sim T$.*

Proof. By induction over the number of rewriting steps. This is obvious if this number is zero; suppose now $P \xrightarrow{T} P' \xrightarrow{T}{}^* T$. The induction hypothesis gives $P' \sim T$; we reason by cases on the rule used to rewrite P:

– (R1): this means that $P \equiv C[\alpha.F]$, $P' \equiv C[\mathbf{0}]$ and $T \equiv !\alpha.F|Q$. From $!\alpha.F|Q \sim C[\mathbf{0}]$, we deduce $!\alpha.F|Q \sim C[\alpha.F]$ by Prop. 2, hence $P \sim T$.
– (R2): we easily have $P \sim P'$, hence $P \sim T$. □

Definition 20 (Joinability) *We say that processes P and Q are* joinable*, written $P \curlyvee Q$, if there exists a process T such that $P \xrightarrow{T}{}^* T$ and $Q \xrightarrow{T}{}^* T$.*

By Lemma 19, $\curlyvee \subseteq \sim$; the other property which is required in order to characterise bisimilarity is *completeness* of the rewriting system, i.e., that all bisimilar processes can be joined. For this, we show that any process can be rewritten into a seed. The proof necessitates the following technical lemma (recall that $\mathsf{s}(P)$ denotes the seed of P):

Lemma 21 *For all P, either P is a seed, or $P \xrightarrow{\mathsf{s}(P)} P'$ for some P' s.t. $P \sim P'$.*

Proof. Write $P \equiv !F|F^P$ and $\mathsf{s}(P) \equiv S|F^S$, with $F \equiv \prod_i \beta_i.F_i$ and $S \equiv \prod_j !\alpha_j.S_j$. By Prop. 7, and since $P \dot{\sim} \mathsf{s}(P)$, $!F \dot{\sim} S$ $(*)$.

Any transition at β_i by $!F$ is answered by S at some $\alpha_{\sigma i}$, yielding equivalence $!F|F_i \;\dot{\sim}\; S|S_{\sigma i}$, which in turn gives $S|F_i \;\dot{\sim}\; S|S_{\sigma i}$, by injecting $(*)$. By Lemma 15, either (a) $F_i \equiv S_{\sigma i}$, or (b) $\neg(S\#F_i)$. In the latter case, (b), this means that P admits some $\alpha_j.S_j$ as a sub-term, and can be rewritten using rule (R1), the resulting process being bisimilar to P, by Prop. 2.

Suppose now that we are in case (a) for all transitions from $!F$, that is, for all i, there exists σi such that $\beta_i.F_i \equiv \alpha_{\sigma i}.S_{\sigma i}$. We observe that the converse (associating a ηj to all js) also holds, and that the number of parallel components in $!F$ is necessarily greater than the number of components in S (otherwise, we would obtain a smaller seed). In the case where this number is strictly greater, this means a replicated component appears twice in $!F$, so that P can be rewritten using rule (R2). We are left with the case where the two processes have the same number of components, which entails that they are equated by $\equiv$.

To sum up, either P can be rewritten, or $!F \equiv S$. In the latter case, we deduce $S \mid F^P \dot{\sim} S \mid F^S$ from $(*)$, and since $S\#F^S$ by Lemma 12(iii), there are two cases according to Lemma 15: either $F^P \equiv F^S$, in which case $P \equiv \mathsf{s}(P)$: P is a seed; or $\neg(S\#F^P)$, i.e., F^P admits some $\alpha_j.S_j$ as a sub-term, and we can rewrite P using (R1), getting a process bisimilar to P by Prop. 2. □

Proposition 22 (Completeness) *For all P, $P \xrightarrow{\mathsf{s}(P)}{}^* \mathsf{s}(P)$.*

Proof. By induction on the size of P. By Lemma 21, either P is a seed and we are done; or $P \xrightarrow{\mathsf{s}(P)} P'$ with $P \sim P'$. We easily check that $\sharp P' < \sharp P$ so that by induction, we have $P' \xrightarrow{\mathsf{s}(P')}{}^* \mathsf{s}(P')$. From $P \sim P'$, we deduce $\mathsf{s}(P) \sim \mathsf{s}(P')$, so that $\mathsf{s}(P) \equiv \mathsf{s}(P')$ by Prop. 16. This allows us to obtain $P \xrightarrow{\mathsf{s}(P)} P' \xrightarrow{\mathsf{s}(P)}{}^* \mathsf{s}(P)$, as the rewriting system is defined modulo $\equiv$. □

Thanks to our characterisation of bisimilarity on seeds (Prop. 16), we obtain:

Theorem 23 (Characterisation) *In mCCS, visible and strong bisimilarity coincide with joinability: $P \dot{\sim} Q$ iff $P \sim Q$ iff $P \curlyvee Q$.*

Proof. We have $\curlyvee \subseteq \sim \subseteq \dot{\sim}$ by Lemma 19. Then, $P \dot{\sim} Q$ entails $\mathsf{s}(P) \equiv \mathsf{s}(Q)$ by Prop. 16. Since $P \xrightarrow{\mathsf{s}(P)}{}^{*} \mathsf{s}(P)$ and $Q \xrightarrow{\mathsf{s}(Q)}{}^{*} \mathsf{s}(Q)$ by Prop. 22, we get $P \curlyvee Q$. □

This result has several consequences. First, we do not need to play silent transitions in bisimulation games (recall the absence of τ prefix). Second, bisimilarity is substitution closed in mCCS. Third, bisimilarity is decidable in mCCS: the definition of joinability is a priori not effective (we are not told how to find T); however, according to the proof of Thm. 23, it suffices to search for T among the processes whose size is smaller than both P and Q to test whether $P \curlyvee Q$. (we can show that for all T the relation $\xrightarrow{T}$ is finitely branching and strongly normalising, so that the predicate $\xrightarrow{T}{}^{*} T$ is decidable).

It should be noted that Christensen et al. already proved decidability of bisimilarity in a larger subset of CCS [5], so that the latter consequence is not surprising. However, their axiomatisation exploits the expansion law, so that it cannot be used to establish substitution closure in our setting.

Remark 24 *The τ prefix is not included in our presentation of mCCS. We extend our results in Appendix B to handle this prefix. The overall strategy is the same, but the proof involves some non-trivial additional technicalities. Basically, difficulties arise when a process answers to a transition emanating from a τ-prefix using a synchronisation (consider, e.g., $!a|!\overline{a}|\tau \sim !a|!\overline{a}$). From the point of view of the axiomatisation, it suffices to extend the rewriting system using the following law:*

$$!a.E \mid !\overline{a}.F \mid !\tau.(E \,|\, F) \;\sim\; !a.E \mid !\overline{a}.F$$

5 Congruence of Strong Bisimilarity in the π-calculus

In this section, we adapt the previous results from CCS to the π-calculus in order to obtain closure of bisimilarity under substitutions, and deduce congruence in the restriction-free π-calculus with only top-level replications.

In moving from CCS to π, some care has to be taken. The first reason for that is that "being a sub-term of" is more subtle in π than in CCS, because of issues related to binding and α-conversion. The second reason is that the LTS for the π-calculus involves substitutions, and we must choose how to handle these in the definition of behavioural equivalence. Among the various notions of bisimilarity that exist for π, we shall actually adopt the simplest and coarsest one, namely ground bisimilarity: when ground bisimilarity is closed under substitutions, the ground, early and late versions of the equivalence coincide [17].

We let x, y, a, b range over a countable set of *names*. We work in the subset of the π-calculus, called $m\pi$, defined by replacing actions from the syntax of mCCS (Sect. 2) with the following productions: $\alpha, \beta \;::=\; a(x) \mid \overline{a}\langle b \rangle$. As usual,

$$\frac{}{\overline{a}\langle b\rangle.F \xrightarrow{\overline{a}\langle b\rangle} F} \qquad \frac{}{!\overline{a}\langle b\rangle.F \xrightarrow{\overline{a}\langle b\rangle} !\overline{a}\langle b\rangle.F \mid F} \qquad \frac{P \xrightarrow{\mu} P' \qquad \mathrm{bn}(\mu) \cap \mathrm{fn}(Q) = \emptyset}{P|Q \xrightarrow{\mu} P'|Q}$$

$$\frac{y \notin \mathrm{fn}(a(x).F)}{a(x).F \xrightarrow{a(y)} F\{y/x\}} \qquad \frac{y \notin \mathrm{fn}(a(x).F)}{!a(x).F \xrightarrow{a(y)} !a(x).F \mid F\{y/x\}} \qquad \frac{P \xrightarrow{\overline{a}\langle b\rangle} P' \qquad Q \xrightarrow{a(x)} Q'}{P|Q \xrightarrow{\tau} P'|Q'\{b/x\}}$$

Fig. 1. Labelled Transition System for $m\pi$.

the operator of input prefix is binding, we write $\mathrm{fn}(P)$ for the set of free names of P, $\mathrm{bn}(\alpha)$ for the set of names bound by α, and we let $P\{y/x\}$ stand for the capture-avoiding substitution of x with y in P. Note that contexts (C) can bind names (e.g., $a(x).[]$). The LTS for $m\pi$ is presented on Fig. 1, where symmetric rules for parallel composition are omitted. The conditions involving freshness of names ensure that $P \xrightarrow{a(x)} P'$ entails $x \notin \mathrm{fn}(P)$; this allows us to give a simple definition of ground bisimilarity:

Definition 25 Ground bisimilarity*, denoted by* $\sim$*, is the largest symmetric binary relation such that* $P \sim Q$ *entails that* $\mathrm{fn}(P) = \mathrm{fn}(Q)$*, and that whenever* $P \xrightarrow{\mu} P'$*, there exists* Q' *s.t.* $Q \xrightarrow{\mu} Q'$ *and* $P' \sim Q'$*.* Visible ground bisimilarity ($\dot{\sim}$) *is defined similarly, by restricting challenges to the cases where* $\mu \neq \tau$*.*

Since we lack the restriction operator, the condition on free names is actually enforced by standard notions of bisimilarity. In particular, this definition coincides with the standard definition of ground bisimilarity on $m\pi$ [17]: input prefixes are tested with fresh names. On finite $m\pi$-processes, ground bisimilarity is a substitution closed congruence [8], so that it coincides with early and late bisimilarities. We need to show that it also coincides with visible bisimilarity (the proof, given in appendix, exploits some technical results from [8]):

Theorem 26 *On finite* $m\pi$ *processes,* $\dot{\sim}$ *and* $\sim$ *coincide.*

As for CCS, we then establish that visible and ground bisimilarities coincide on all $m\pi$ processes. Since visible bisimilarity is easily shown to be substitution closed (Prop. 27 below, proved in the appendix), this allows us to deduce congruence and coincidence with the other notions of bisimilarity (Thm. 28).

Proposition 27 *Visible bisimilarity is a substitution closed congruence.*

The reasoning goes along the same lines as for CCS, so that we only review the main differences, referring to appendix A for detailed proofs. We need to adapt the definition of distribution congruence, and we rely on Thm. 26 to prove that distribution congruence is contained in ground bisimilarity. As expected, we need to impose conditions on names when stating results involving contexts; for example, in Prop. 2, C should not bind free names of $\alpha.F$. Note moreover that we need to go against a Barendregt convention to perform some rewriting

steps. For example, we want to allow $!a(x).a(x) \xrightarrow{!a(x)} !a(x)$. We finally obtain coincidence of visible and ground bisimilarities, which yields congruence:

Theorem 28 (Characterisation and congruence) *In $m\pi$, early, late, visible and ground bisimilarity coincide and define a substitution closed congruence.*

6 Conclusions and future work

We have presented a characterisation of strong bisimilarity in the restriction- and sum-free fragment of CCS, where replications are only allowed at top-level (Thm. 23). This has allowed us to put forward important algebraic properties of replication w.r.t. bisimilarity. By extending this result to the π-calculus, we have established congruence of bisimilarity in the corresponding fragment (Thm. 28).

6.1 Expressiveness

As established in [5], strong bisimilarity is decidable in mCCS. Moreover, given a mCCS process P, both questions of whether P exhibits a diverging computation and of whether P has a finite computation leading to a stuck process are decidable. The former result follows from [4, Cor. 3], because mCCS is a subset of CCS[3]. The latter result follows from [2, Thm. 2], where the same property is proved for CCS without name restriction. These results suggest that mCCS has a limited expressive power; in fact, mCCS is not Turing expressive, and there is no encoding from CCS into mCCS up to failures equivalence [2].

Similar properties are likely to hold for $m\pi$: name-passing should not add that much expressive power, as long as we keep name restriction outside the calculus (intuitively because all names are known from the beginning). Nevertheless, our point is not to claim that mCCS and $m\pi$ are interesting calculi in terms of expressive power, but rather that they are in terms of building axiomatisations of behavioural equivalences. As we explain below, we plan to move to richer calculi, which, rather naturally, will end up having greater expressive power.

6.2 Future work

We would like to generalize our results further, by finding extensions of the calculi we have studied for which bisimilarity is substitution closed. A counterexample involving the operators of restriction and replication is presented in [3] to establish non-congruence of bisimilarity. Therefore, in light of [8, Corollary 5.9] and Thm. 28, we can think of two paths to explore: either add a limited usage of restriction to the language, or study the full replication operator (note that adding the full replication operator would not drastically increase the expressive power: CCS without name restriction remains strictly less expressive than CCS [2]).

[3] We use here 'CCS' to refer to CCS with replication and without recursion.

Adding the restriction operator. The counterexample of [3] suggests that restrictions occurring immediately under replications are problematic. A natural extension of mCCS would therefore consist in adding restriction only to the grammar for finite processes—we indeed know from [8] that restriction does not break substitution closure on finite processes. Adding the τ prefix is a first step (cf. [7] and Rmk. 24) in this direction: this prefix can be encoded as $(\nu c)\,(\overline{c}|c.P)$, for a fresh c. However, an important difficulty in adapting our proofs is the definition and analysis of a counterpart of visible bisimilarity in presence of restriction.

Beyond top-level replications. Handling arbitrary replications seems really challenging. We have started investigating the case where replication is not at top-level, but where nested replications (i.e., replications that occur under replications) are forbidden. The law

$$\alpha.C[!\alpha.C[\mathbf{0}]] \;\sim\; !\alpha.C[\mathbf{0}]$$

seems important to capture bisimilarity in this setting: it somehow generalises the distribution law (D) to replicated processes, and it allows one to equate processes like $!a$ and $a.!a$. We do not know at the moment whether this law, together with the laws presented above, is sufficient to characterise bisimilarity. One of the difficulties in studying this richer language is that seeds are no longer stable under reduction (Prop. 17): for example, $!a.b|c.!b$ is a seed while its reduct along c, $!a.b|!b$, is not, being bisimilar to $!a|!b$.

Related to this question is the work on HOcore [10], a restriction-free higher-order π-calculus where strong bisimilarity is characterised by the distribution law. In this calculus, replication can be encoded using the higher-order features. The encoding is not fully abstract, however, so that it does not entail substitution closure in presence of "standard" replication.

Weak bisimilarity. Rather complex laws appear when moving from the strong to the weak case. For example, the following laws are valid for weak bisimilarity:

$$!\overline{a}.a \,|\, a.b \;\approx\; !\overline{a}.a \,|\, a \,|\, b \;, \qquad\qquad !\overline{a} \,|\, !a.b \;\approx\; !\overline{a} \,|\, !a \,|\, !b \;.$$

In both cases, although the related processes have the same size, the right-hand side process could be considered as a seed. We do not know how to generalise the first equivalence. For the second one, the following law, where $\langle P\rangle_a$ is defined homomorphically, except for $\langle a.P\rangle_a = \langle \overline{a}.P\rangle_a = \langle P\rangle_a$, is an interesting candidate:

$$!\overline{a}.P \,|\, !a.Q \;\approx\; !\overline{a} \,|\, !a \,|\, !\langle P\rangle_a \,|\, !\langle Q\rangle_a \;.$$

Acknowledgements. We are grateful to the anonymous referees for their numerous and valuable comments.

References

1. L. Aceto, W.J. Fokkink, A. Ingolfsdottir, and B. Luttik. Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles: steps on the road to infinity*, volume 3838 of *LNCS*, pages 338–367. Springer, 2005.
2. J. Aranda, F. D. Valencia, and C. Versari. On the expressive power of restriction and priorities in CCS with replication. In *Proc. FOSSACS*, volume 5504 of *LNCS*, pages 242–256. Springer, 2009.
3. M. Boreale and D. Sangiorgi. Some congruence properties for π-calculus bisimilarities. *Theoretical Computer Science*, 198:159–176, 1998.
4. N. Busi, M. Gabbrielli, and G. Zavattaro. On the expressive power of recursion, replication and iteration in process calculi. *J. of Mathematical Structures in Computer Science*, 19(6):1191–1222, 2009.
5. S. Christensen, Y. Hirshfeld, and F. Moller. Decidable subsets of CCS. *Computer Journal*, 37(4):233–242, 1994.
6. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of ACM*, 32(1):137–161, 1985.
7. D. Hirschkoff and D. Pous. Extended version of this abstract. Available from `http://hal.archives-ouvertes.fr/hal-00375604/`.
8. D. Hirschkoff and D. Pous. A distribution law for CCS and a new congruence result for the pi-calculus. *Logial Methods in Computer Science*, 4(2), 2008.
9. Y. Hirshfeld and M. Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *ICALP*, volume 1644 of *LNCS*, pages 412–421. Springer, 1999.
10. I. Lanese, J. A. Pérez, D. Sangiorgi, and A. Schmitt. On the expressiveness and decidability of higher-order process calculi. In *LICS*, pages 145–155. IEEE, 2008.
11. R. Milner. Functions as Processes. *J. of Mathematical Structures in Computer Science*, 2(2):119–141, 1992.
12. F. Moller. *Axioms for Concurrency*. PhD thesis, University of Edinburgh, 1988.
13. U. Nestmann and B.C. Pierce. Decoding choice encodings. *Information and Computation*, 163:1–59, 2000.
14. D. Pous. Complete lattices and up-to techniques. In *APLAS*, volume 4807 of *LNCS*, pages 351–366. Springer, 2007.
15. D. Pous. *Techniques modulo pour les bisimulations*. PhD thesis, ENS Lyon, 2008.
16. D. Sangiorgi. On the bisimulation proof method. *J. of Mathematical Structures in Computer Science*, 8:447–479, 1998.
17. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

A Extension to the $m\pi$-calculus

In this appendix, we adapt the proofs from mCCS to $m\pi$, and establish the results announced in Sect. 5.

A.1 Setting and algebraic laws about replication

The results from Sect. 2 extend without difficulties to the π-calculus:

Proposition A1 (Prop. 2) *If $C[\mathbf{0}] \sim\, !\alpha.F|P$, where C does not bind any free name of $\alpha.F$, then $C[\mathbf{0}] \sim C[\alpha.F]$.*

Proof. Similar to the proof of Prop. 2, the fact that C should not bind any free name of $\alpha.F$ is used when the fired prefix is an input and guards the hole: this ensures that $\alpha.F$ is not affected by the induced substitution. □

As a consequence, we obtain the validity of laws (A) and (A′), with the extra proviso that C (resp. D) does not bind names occurring free in $\alpha.F$ (resp. $\alpha.D$):

Lemma A2 (Lemma. 6) *If* $!F \mathrel{\dot\sim} P|Q$*, then* $!F \mathrel{\dot\sim} !F|P$.

Proof. We rely on the same purely algebraic reasoning as for Lemma 6, since the relevant laws are valid in $m\pi$ (i.e., $!P \mathrel{\dot\sim} !P|P$ and the fact that $\mathrel{\dot\sim}$ is preserved by extended replication). □

Proposition A3 (Prop. 7) *If* $!F|F_0 \mathrel{\dot\sim} !E|E_0$*, then* $!F \mathrel{\dot\sim} !E$.

Proof. Working in $m\pi$ does not prevent us from emptying E_0 and F_0. The rest of the CCS proof uses algebraic arguments, and can be replayed. □

A.2 Seeds

Seeds in $m\pi$ are defined exactly like in mCCS (Def. 8). We then prove the counterpart of Fact. 10. We start with coincidence of visible and ground bisimilarity on finite processes: while this can be derived from the results in [8], visible bisimilarity is not taken into account in that paper.

Theorem A4 (Thm. 26) *On finite* $m\pi$ *processes,* $\mathrel{\dot\sim}$ *and* $\sim$ *coincide.*

Proof. It suffices to prove that $\mathrel{\dot\sim} \subseteq \sim$. We exploit a technical result from [8], the absence of 'mutual desynchronisation' (Lemma 4.4), i.e.,

$$\text{if } \alpha \neq \beta,\ E \xrightarrow{\alpha} E',\ F \xrightarrow{\beta} F', \text{ then } \forall F_0,\ \beta.E \,|\, F' \,|\, F_0 \not\mathrel{\dot\sim} E' \,|\, \alpha.F \,|\, F_0.$$

In [8], this result is proved for the finite fragment of mCCS, and then extended to the finite sum-free fragment of the π-calculus, by considering an 'erasing' translation from π into CCS (cf. Def. 5.3, Lemma 5.4 and Prop. 5.5 in [8]—the translation transforms visibly bisimilar π-calculus processes into bisimilar CCS processes, so that the absence of mutual desynchronisation can be established w.r.t. visible bisimilarity in π).

Using this property, we show that the restriction of $\mathrel{\dot\sim}$ to finite processes is a ground bisimulation, i.e., that challenges along silent transitions can be answered: suppose that $E \mathrel{\dot\sim} F$, and $E \xrightarrow{\tau} E'$. W.l.o.g., we can write $E = a(b).E_2|\overline{a}\langle b\rangle.E_1|E_0$. By playing the input prefix, and then the output prefix, $E \mathrel{\dot\sim} F$ gives $F \xrightarrow{a(b)} \xrightarrow{\overline{a}\langle b\rangle} F_1'$ with $E' \sim F_1'$. By playing theses prefixes in reverse order, we obtain $F \xrightarrow{\overline{a}\langle b\rangle} \xrightarrow{a(b)} F_2'$ with $E' \sim F_2'$. There are two cases to consider:

- if one of these sequences of transitions emanating from F involves the firing of concurrent prefixes, then we can deduce $F \xrightarrow{\tau} F_i'$, and close the diagram;

– if both correspond to the firing of sequential prefixes, i.e., $F_1 \equiv a(b).U|\overline{a}\langle b\rangle.V|F_0$ with $U \xrightarrow{\overline{a}\langle b\rangle} U'$, $V \xrightarrow{a(b)} V'$, $F_1' \equiv U'|\overline{a}\langle b\rangle.V|F_0$, and $F_2' \equiv a(b).U|V'|F_0$, we check that F_1' and F_2' determine a mutual desynchronisation ($F_1' \dot{\sim} E' \dot{\sim} F_2'$), which is contradictory. □

We then show that visible bisimilarity is a substitution closed congruence, on all $m\pi$ processes. We let σ range over capture-avoiding name substitutions; we rely on the following lemma to reason about reducts along input transitions.

Lemma A5 *$P\sigma \xrightarrow{a_0(x)} P_0$ iff there exists z, a, P' such that $P \xrightarrow{a(z)} P'$, $a_0 = a\sigma$, and $P_0 = P'\{\sigma, z \to x\}$ (where $\{\sigma, z \to x\}$ is the parallel substitution that extends σ with the replacement of x for z).*

Proposition A6 (Prop. 27) *In $m\pi$, $\dot{\sim}$ is a substitution closed congruence.*

Proof. Using Lemma A5, we show that $\{(P\sigma, Q\sigma) \ / \ \sigma, \ P \dot{\sim} Q\}$ is a visible ground bisimulation. This is possible because $\dot{\sim}$ does not test challenges along silent transitions (however, unlike for CCS, we cannot fix the substitution). Congruence then follows: we use substitution closure in order to handle the input prefix. □

Another difficulty is that [8] does not provide an algebraic characterisation of bisimilarity on finite π-processes—while it does for finite mCCS processes, using the distribution law (D). Therefore, we can no longer work with a "structural" definition of distribution congruence: we have to use the following definition.

Definition A7 (Distribution congruence for $m\pi$—Def. 9) *We call* distribution congruence *the smallest congruence relation $\equiv$ that satisfies the laws of an abelian monoid for $(|, \mathbf{0})$ and contains the restriction of $\sim$ to finite processes.*

This definition and the above results allow us to deduce the remaining inclusions corresponding to Fact. 10, about mCCS:

Lemma A8 *In $m\pi$, we have $\equiv \subseteq \sim \subseteq \dot{\sim}$.*

Proof. The second inclusion is immediate from the definitions. For the first inclusion, we show that $\equiv$ is a ground bisimulation. We exploit the fact that $\equiv$ is substitution closed on finite processes (Prop. A6 and Thm. A4) in order to handle replicated input prefixes: if $!a(x).F \equiv !a(x).E$ because $F \equiv E$, and, if $!a(x).F \xrightarrow{a(y)} !a(x).F \,|\, F\{y/x\}$, then $!a(x).E$ answers with the obvious transition, and we check that $!a(x).F \,|\, F\{y/x\} \equiv !a(x).E \,|\, E\{y/x\}$: thanks to substitution closure on finite processes, we have $F\{y/x\} \equiv E\{y/x\}$. □

The notion of residual process remains unchanged; we have to adapt the notion of clean process (w.r.t. a fixed seed having only replicated components only: $S = \prod_i !\alpha_i.S_i$):

Definition A9 (Clean process, residual—counterpart of Def. 11)
F is clean w.r.t. *S, written $S\#F$, if it is not the case that for some i and finite context D, $F \equiv D[\alpha_i.S_i]$, where D does not bind any free name of $\alpha_i.S_i$.*

R is a residual of *S, written $S \rightsquigarrow R$ when there exist $k > 0$, $\beta_1, .., \beta_k$, and $P_1, .., P_k$ such that $S \xrightarrow{\beta_1} P_1 \ldots \xrightarrow{\beta_k} P_k \equiv S|R$. We shall use R to range over such residual processes.*

Lemma 12 still holds: clean processes and residuals are preserved by labelled transitions, and the finite part of a seed is clean (note that the assumption about bound names in the definition of a clean process is required for the latter point to hold). Lemma 13 needs to be strenghtened:

Lemma A10 (Lemma 13) *If $S \rightsquigarrow \alpha.R$ with $\alpha.R$ prime, then $S_i \equiv D[\alpha.R]$ for some i, D such that D does not bind free names of S.*

Proof. We proceed like in the proof of Lemma 13. Some care is required in the second case of the alternative, where $R \equiv \beta_k.(R_0|R'_0)|R'_1$ for some R'_0, R'_1, with $\beta_k.(R_0|R'_0)$ prime. By induction, there are i, D such that $S_i \equiv D[\beta_k.(R_0|R'_0)]$ and D does not bind names of S; we can take i and $D[\beta_k.[]|R'_0]$: since the β_k-transition was performed in parallel with S, β_k cannot bind names of S. □

Lemma A11 (Lemma 14) *(i) If $S \rightsquigarrow R$ and $S \dot{\sim} S|R$, then $R \equiv \mathbf{0}$.*
(ii) If $S \rightsquigarrow R$, then $S\#R$.

Proof. (i) Suppose that R is non-empty, and write $R \equiv \alpha.R_0|R_1$ with $\alpha.R_0$ prime. Lemma A2 gives $S \dot{\sim} S|\alpha.R_0$, hence $S \dot{\sim} S|!\alpha.R_0$ by replicating all processes. Moreover, $S \rightsquigarrow \alpha.R_0$ by emptying R_1, so Lemma A10 gives i, D such that $S_i \equiv D[\alpha.R_0]$ and D does not bind free names of S. Since $S \dot{\sim} S|\alpha.R_0$, the free names of $\alpha.R_0$ are contained in those of S, so that they cannot be captured by D. This allows us to use law (A) to obtain a contradiction, like in the CCS case. □

(ii) By contradiction, suppose that $R \equiv D[\alpha_i.S_i]$, where D does not bind free names of $\alpha.S_i$. By emptying the prefixes of D, we get $S \rightsquigarrow \alpha_i.S_i$ (since D does not capture names of $\alpha_i.S_i$, $\alpha_i.S_i$ appears unchanged after the sequence of transitions). Since $S \dot{\sim} S|\alpha_i.S_i$, this contradicts (i). □

Lemma A12 (Lemma 15) *If $S|F \dot{\sim} S|E$ $S\#F$, and $S\#E$, then $F \equiv E$.*

Proof. Same proof as for Lemma 15. □

Proposition A13 (Prop. 16) *For all seeds P, P', we have $P \dot{\sim} P'$ iff $P \sim P'$ iff $P \equiv P'$.*

Proof. Same proof as for Prop. 16. □

A.3 Rewriting system

The rewriting system is extended to $m\pi$ by avoiding name captures when erasing components of the seed using rule (R1); joinability is defined as previously:

Definition A14 (counterpart of Defs. 18 and 20) *Any process T induces a relation between processes, written $\xrightarrow{T}$, defined by the following rules, modulo distribution congruence ($\equiv$):*

$$\frac{T \equiv !\alpha.F|Q \qquad \mathrm{cn}(C) \cap \mathrm{fn}(\alpha.F) = \emptyset}{C[\alpha.F] \xrightarrow{T} C[\mathbf{0}]}\ (\mathrm{R1}) \qquad \frac{}{!\alpha.F|!\alpha.F|P \xrightarrow{T} !\alpha.F|P}\ (\mathrm{R2})$$

(Where $\mathrm{cn}(C)$ denotes the set of names captured by C.) The reflexive transitive closure of $\xrightarrow{T}$ is written $\xrightarrow{T}{}^*$. We say that processes P and Q are joinable*, written $P \curlyvee Q$, whenever there exists a process T such that $P \xrightarrow{T}{}^* T$ and $Q \xrightarrow{T}{}^* T$.*

With these definitions, the proofs of Lemmas 19, 21, and Prop. 22 can be replayed without additional difficulties, so that we obtain:

Theorem A15 (Characterisation—Thm. 23) *In $m\pi$, visible and strong bisimilarity coincide with joinability: $P \dot{\sim} Q$ iff $P \sim Q$ iff $P \curlyvee Q$.*

Since visible bisimilarity is a substitution closed congruence on $m\pi$ (Prop. A6), we deduce that the same holds for ground bisimilarity. This in turn entails coincidence with early and late bisimilarities, as stated in Thm. 28.

B Adding τ prefixes

B.1 The Setting

In this section, we study $m\mathrm{CCS}_\tau$, the enrichment of mCCS with τ prefixes: for this, we merge the entry for prefixes and labels in the grammar given at the beginning of Section 2:

$$\alpha, \beta, \mu \ ::= \ a \mid \overline{a} \mid \tau$$

The definition of bisimilarity ($\sim$) remains unchanged (Def. 1), but visible bisimilarity ($\dot{\sim}$) has to be adapted. Indeed, visible bisimilarity, as defined previously, is no longer contained in bisimilarity, since, e.g., $\tau.P \dot{\sim} \mathbf{0}$.

Indeed, we shall work with *prefix* bisimilarity, defined by forbidding the use of the synchronisation rule from the LTS:

Definition B1 *We say that a process P does a* prefix transition *along μ to P', denoted by $P \overset{\mu}{\longmapsto} P'$, if $P \xrightarrow{\mu} P'$ without using the synchronisation rule, i.e., by firing a single μ prefix.* Prefix bisimilarity ($\dot{\sim}$) *is the largest symmetric binary relation over processes such that whenever $P \dot{\sim} Q$ and $P \overset{\mu}{\longmapsto} P'$, there exists Q' such that $P' \dot{\sim} Q'$ and $Q \overset{\mu}{\longmapsto} Q'$.*

(Note that $\overset{\alpha}{\mapsto} = \overset{\alpha}{\to}$ for $\alpha = a, \overline{a}$, and $\overset{\tau}{\mapsto} \subsetneq \overset{\tau}{\to}$.) Visible bisimilarity and prefix bisimilarity coincide on mCCS processes—without τ prefixes—whence the notation. Using the results from [8], we obtain:

Theorem B2 *Prefix bisimilarity* ($\dot{\sim}$) *coincides with strong bisimilarity* ($\sim$) *and distribution congruence* ($\equiv$) *on finite $mCCS_\tau$ processes.*

Proof. That prefix bisimilarity coincides with distribution congruence on finite $m\mathrm{CCS}_\tau$ processes is a consequence of the results from [8] (although this paper only deals with strong bisimilarity, in the calculus without τ-prefixes): let θ be a bijection from $m\mathrm{CCS}_\tau$ prefixes to an arbitrary set, and let $F\theta$ denote the process obtained from F by replacing all its prefixes by their image under θ, seen as input prefixes. For all E, F, we have

$$E \dot{\sim} F \quad \Leftrightarrow \quad E\theta \dot{\sim} F\theta \quad \Leftrightarrow \quad E\theta \sim F\theta \quad \Leftrightarrow \quad E\theta \equiv F\theta \quad \Leftrightarrow \quad E \equiv F$$

The first and last equivalences follow from the fact that prefix bisimilarity and distribution congruence are insensitive to the interpretation of prefixes. The second equivalence comes from the fact that a process of the form $E\theta$ cannot perform any τ transition, since it contains only input prefixes. The third one is a direct consequence of [8, Thm. 2.6].

Distribution congruence is always contained in strong bisimilarity (since all its axioms are valid, in particular the distribution law (D)). It remains to show that strong bisimilarity is contained is prefix bisimilarity. We first remark that $E \sim F$ entails $\sharp E = \sharp F$: by contradiction, pick the smallest E such that there exists F with $E \sim F$ and $\sharp E < \sharp F$. E cannot be empty since F is not; if $E \overset{\alpha}{\to} E'$ for some $\alpha \neq \tau$, F can answer and we get a smaller counter-example; if E is a parallel composition of τ prefixes, then firing one of these prefixes also yield a smaller counter-example, since F cannot answer with a synchronisation (E would not be able to answer to the underlying visible transitions). We can finally show that $\sim$ is a prefix bisimulation: if $P \sim Q$ and $P \overset{\mu}{\mapsto} P'$ then $Q \overset{\mu}{\to} Q'$ with $P' \sim Q'$; we necessarily have $Q \overset{\mu}{\mapsto} Q'$: we must have $\sharp P = \sharp Q$ and $\sharp P' = \sharp Q'$ by the above remark, and $\sharp P = \sharp P' + 1$. □

This characterisation no longer holds in presence of replication: we have $!a|!\overline{a}|\tau \sim !a|!\overline{a}$ while these processes are not prefix bisimilar: the left-hand side challenge on τ cannot be answered with a prefix, it requires a synchronisation on a. As a consequence, we will work with prefix bisimilarity only with finite processes, as a coinductive tool to prove distribution congruence results.

The above phenomenon is what we call a *"1/2 move"*: the firing of a τ prefix is answered by firing two visible prefixes. Accordingly, a *"1/1 move"* is a step in the bisimulation game where the firing of a prefix, visible or not, is answered by a prefix transition—where only one prefix is fired. When reasoning about a 1/2 move, we shall use the notation $P \overset{a|\overline{a}}{\longrightarrow} P'$ to represent the fact that we derive $P \overset{\tau}{\to} P'$ by firing two parallel prefixes, a and $\overline{a}$ (so that, a priori, we do not have but $P \overset{\tau}{\mapsto} P'$).

The main difficulty in handling τ prefixes comes from 1/2 moves; we have to prove that these cannot arise in bisimilarity games between seeds. As far as the axiomatisation is concerned, it suffices to integrate the following law:

$$!a.E \mid !\overline{a}.F \;\sim\; !a.E \mid !\overline{a}.F \mid !\tau.(E|F) \tag{B}$$

The remaining of the section is organised like previously: we prove that strong bisimilarity ($\sim$) coincides with distribution congruence ($\equiv$) on seeds, and we define a rewriting system to rewrite any process into its seed. Since we cannot use use prefix bisimilarity with infinite processes, we obtain substitution closure of bisimilarity by showing that rewriting steps are preserved by substitutions.

Prop. 2, Lemma 6, and Prop. 7 scale smoothly to $m\mathrm{CCS}_\tau$; in particular, working with standard bisimilarity rather than visible bisimilarity is not problematic. The above law (B) is easily shown to hold, using a bisimulation up to context.

B.2 Seeds

The seeds are defined as previously (Def. 8). Like in Sect. 3, we first focus on a replicated seed $S = \prod_i !\alpha_i.S_i$. Intuitively, the *spores* of S (Def. B3 below) are the $\alpha_i.S_i$ components, plus the τ-prefixed finite processes corresponding to possible synchronisations of these components. While the notion of residual remains unchanged, we need to take these additional components into account when we extend the notion of clean process. Typically, the process $b.\tau.P$ cannot be clean with respect to $!a|!\overline{a}.P$, since the subterm $\tau.P$ can be erased. We are therefore led to introduce the following shortcut:

Definition B3 (Spore) *A finite prefixed process $\alpha.E$ is a* spore *of a process $P \equiv \prod_i \alpha_i.P_i|F$, written $E \dot{\subset} P$, if one of the following conditions holds:*

- *either $\alpha.E \equiv \alpha_i.P_i$ for some i,*
- *or $\alpha.E \equiv \tau.(P_i|P_j)$ for some i, j, a such that $\alpha_i = a$ and $\alpha_j = \overline{a}$.*

Definition B4 (Clean process) *A finite process F is* clean w.r.t. *S, written $S\#F$, if F does not contain spores of S: for all spores $E \dot{\subset} S$ and all finite contexts D, $F \not\equiv D[E]$.*

Lemma 13 scales (the notion of residual did not change):

Lemma 29 *If $S \rightsquigarrow \alpha.R$ with $\alpha.R$ prime, then $S_i \equiv D[\alpha.R]$ for some i, D.*

Fact B5 *If a finite process $\alpha.E$ is not prime then there exist $F, n \geq 1$ such that $E \equiv F|(\alpha.F)^n$ and $\alpha.F$ is prime.*

Lemma B6 *For all spores $\alpha.E \dot{\subset} S$,*

(i) $S|\alpha.E \sim S$,
(ii) $\alpha.E$ is prime.

Proof. (i) follows from law (B). For (ii), we reason by contradiction: assume that $\alpha.E$ is not prime, i.e., that $E \equiv F|(\alpha.F)^n$ for some F, $n \geq 1$, with $\alpha.F$ prime. We distinguish two cases, according to the definition of spores:

- if $\alpha.E \equiv \alpha_i.S_i$, then $S \sim \prod_{j\neq i} !\alpha_j.S_j|!\alpha.F$, which is a smaller seed;
- if $\alpha.E \equiv \tau.(S_i|S_j)$ for some i, j, a such that $\alpha_i = a$ and $\alpha_j = \overline{a}$, then $S_i|S_j \equiv F|(\tau.F)^n$. Since $\alpha.F$ is prime it appears at top-level in S_i or S_j, say S_j. Therefore $S_j \equiv \tau.F|S'_j$ and $S_i|S'_j \equiv F|(\tau.F)^{n-1}$ by Corollary 5; we get a contradiction with $S' \triangleq \prod_{k\neq j} !\alpha_k.S_k|!\overline{a}.S'_j$ which is a smaller seed: we have $S' \sim S'|!\tau.(S_i|S'_j)$ by law (B), whence $S' \sim S'|!\tau.(F|(\tau.F)^{n-1})$, and $S' \sim S'|!(\tau.F)^n \sim S'|!\tau.F$, so that $S' \sim S$ by Prop. 2. □

This allows us to extend Lemma 14:

Lemma B7 *(i) If $S \rightsquigarrow R$ and $S \sim S|R$, then $R \equiv \mathbf{0}$.*
(ii) If $S \rightsquigarrow R$, then $S\#R$.
(iii) If $S|F$ is a seed, then $S\#F$.

Proof. (i) Exactly like for Lemma 14(i).
(ii) By contradiction, suppose that $R \equiv D[\alpha.E]$ with $\alpha.E \Subset S$. By emptying the prefixes of D, we have $S \rightsquigarrow \alpha.E$. Since $S \sim S|\alpha.E$ by Lemma B6(i), this contradicts (i).
(iii) By contradiction: if $F \equiv D[\alpha.E]$ with $\alpha.E \Subset S$, then $S \sim S|!\alpha.E$, and $S|F \sim S|D[\mathbf{0}]$ by Prop. 2, which violates the minimality hypothesis about $S|F$. □

Lemma B8 *If $S|R_1 \sim S|R_2$, $S \rightsquigarrow R_1$, and $S \rightsquigarrow R_2$, then $R_1 \equiv R_2$.*

Proof. Call $k_i = \sharp R_i$, for $i = 1, 2$; we reason by induction on $\min(k_1, k_2)$. If $R_1 \equiv \mathbf{0}$ or $R_2 \equiv \mathbf{0}$, then we are done by Lemma B7(i). Otherwise, we prove that $\{(R_1, R_2)\} \cup \equiv$ is a prefix bisimulation; assume w.l.o.g. that $0 < k_1 \leq k_2$.

Suppose that $R_1 \overset{\mu}{\mapsto} R'_1$; it suffices to find some R'_2 such that $R_2 \overset{\mu}{\mapsto} R'_2$ and $R'_1 \equiv R'_2$. By hypothesis, we obtain $S|R_2 \overset{\mu}{\to} S|R'_2$ for some R'_2 with $S|R'_1 \sim S|R'_2$, and by induction, $R'_1 \equiv R'_2$. It remains to prove that $R_2 \overset{\mu}{\mapsto} R'_2$. We reason by cases, depending on whether R'_2 was obtained using a $1/1$ move or a $1/2$ move.

- if $S|R_2 \overset{\mu}{\mapsto} S|R'_2$. i.e., the transition was obtained by firing a single prefix, then necessarily $R_2 \overset{\mu}{\mapsto} R'_2$: otherwise, if the transition comes from S, i.e., $R'_2 \equiv R_2|S_i$ for some i, we have $k_2 = \sharp R_2 \leq \sharp R'_2 = \sharp R'_1 = k_1 - 1 < k_1$, a contradiction.
- if $S|R_2 \xrightarrow{a|\overline{a}} S|R'_2$, then we find a contradiction. We have $\sharp R'_2 = \sharp R'_1 = k_1 - 1$, we distinguish several cases according to how the above transition was triggered:
  - S does $\xrightarrow{a|\overline{a}}$. In this case, R_2 is a subterm of R'_2, which entails $k_2 \leq k_1 - 1 = \sharp R'_1$, a contradiction.

- S does $\xrightarrow{a}$, R_2 does $\xrightarrow{\overline{a}}$. In this case, R_1 and R_2 have the same size, $S \xrightarrow{a} S$, and $R_2 \xrightarrow{\overline{a}} R_2'$. To the challenge $S|R_2 \xrightarrow{\overline{a}} S|R_2'$, by reasoning as above (since $k_1 = k_2$), $S|R_1$ necessarily answers by firing $R_1 \xrightarrow{\overline{a}} R_1''$ with $R_1'' \equiv R_2'$. We finally have $R_1' \equiv R_1''$, whence $R_1 \overset{\tau}{\mapsto} R_1'$ and $R_1 \xrightarrow{\overline{a}} \equiv R_1''$ which is impossible with finite $m\mathrm{CCS}_\tau$ processes.
  The case where a and $\overline{a}$ are swapped is treated the same way.
- $R_2 \xrightarrow{a|\overline{a}} R_2'$. Since $\sharp R_1' = \sharp R_2' = k_1 - 1$, we have $k_2 = k_1 + 1$. We can play the challenges $R_2 \xrightarrow{a}\xrightarrow{\overline{a}} R_2'$, to which $S|R_1$ replies by $S|R_1 \xrightarrow{a}\xrightarrow{\overline{a}} S|R_1''$. Since $\sharp R_2' < k_1$, we can apply induction to deduce $R_1'' \equiv R_2' \equiv R_1'$. We thus have $R_1 \overset{\tau}{\mapsto} R_1'$, and either $R_1 \equiv R_1'$, or $R_1 \xrightarrow{a} \equiv R_1'$, or $R_1 \xrightarrow{\overline{a}} \equiv R_1'$, or $R_1 \xrightarrow{a}\xrightarrow{\overline{a}} \equiv R_1'$ (according to how the sequence of transitions $S|R_1 \xrightarrow{a}\xrightarrow{\overline{a}} S|R_1''$ is derived): all these possibilities are contradictory, hence a contradiction alltogether.

Finally, all prefix transitions $R_1 \xrightarrow{\mu} R_1'$, are answered by prefix transitions $R_2 \xrightarrow{\mu} R_2'$ with $R_1' \equiv R_2'$. In particular, this entails $k_1 = k_2$, and we can answer the challenges of R_2 symmetrically (we had only supposed $k_1 \leq k_2$). We deduce that $R_1 \sim R_2$, i.e., $R_1 \equiv R_2$. □

We now show that one can always empty a residual using a prefix bisimulation game, where only 1/1 moves are played:

Lemma B9 *Suppose $S|R \sim P$ with $S \rightsquigarrow R$; P can do $\sharp R$ prefix transitions to some P' such that $S \sim P'$.*

Proof. We reason by induction over the size of R:

- If $R \equiv 0$, we are done.
- If R contains a top-level prefix $\alpha \neq \tau$: $R \xrightarrow{\alpha} R'$. Then, P necessarily answers with a 1/1 move, and we are done by induction.
- If R contains only τ prefixes at top-level, write $R \equiv \tau.R_0|R_1$ and play this prefix: we find P_0 such that $P \xrightarrow{\tau} P_0$ and $S|R_0|R_1 \sim P_0$. If $P \overset{\tau}{\mapsto} P_0$, then we conclude by induction; otherwise, i.e., if $P \xrightarrow{a|\overline{a}} P'$, we find a contradiction: by playing $P \xrightarrow{a}\xrightarrow{\overline{a}} P_0$, we deduce $S|R \xrightarrow{a}\xrightarrow{\overline{a}} S|R|R'$ with $S|R|R' \sim P_0$ (since R contains only τ prefixes at top-level it cannot be used to answer these challenges). Therefore, $S|R|R' \sim S|R_0|R_1$, whence $R|R' \equiv R_0|R_1$ by Lemma B8. This is impossible since $\sharp(R_0|R_1) = \sharp R - 1 < \sharp(R|R')$. □

Lemma B10 *If $S|F \sim S|R$, with $S\#F$ and $S \rightsquigarrow R$, then $F \equiv R$.*

Proof. We proceed by induction on $\sharp F$. At each step, we first prove the following property:

$$\forall P, R,\ S|F|P \sim S|R \text{ and } S \rightsquigarrow R \text{ entail } \sharp F \leq \sharp R \qquad (\dagger)$$

Indeed, by emptying R using Lemma B9, we get F', P' such that $S|F'|P' \sim S$, where F' is the process that is obtained from F. In particular, $S \# F'$, and $\sharp F' \leq \sharp F \leq \sharp F' + \sharp R$ since a $1/1$ game was played. Therefore, it suffices to show that $F' \equiv \mathbf{0}$. Assume by contradiction that $F' \equiv \alpha F_0 | F_1$; by Lemma 6, $S|\alpha.F_0 \sim S$. By firing the α prefix, we find i such that $S|F_0 \sim S|S_i$ and $\alpha_i = \alpha$. By induction, since $\sharp F_0 < \sharp F' \leq \sharp F$, we obtain $F_0 \equiv S_i$, and hence $\alpha.F_0 \equiv \alpha_i.S_i$, which is contradictory with $S \# F'$.

Suppose now that $S|F \sim S|R$, and let us prove that $F \equiv R$ by showing that $\{(F, R)\} \cup \equiv$ is a prefix bisimulation. We consider the left-to-right part of the game first: suppose that $F \overset{\alpha}{\mapsto} F'$; we find R' such that $S|R \overset{\alpha}{\to} S|R'$ and $S|F' \sim S|R'$, whence $F' \equiv R'$ by induction, so that it suffices to show that $R \overset{\alpha}{\mapsto} R'$. If $S|R \overset{\alpha}{\mapsto} S|R'$, then either $R \overset{\alpha}{\mapsto} R'$ and we are done, or $R' \equiv S_i|R$ for some i, which is impossible by (†): we would have $\sharp F = \sharp F' + 1 = \sharp R' + 1 \geq \sharp R + 1 > \sharp R$. Otherwise, if $\alpha = \tau$ and $S|R \xrightarrow{a|\overline{a}} S|R'$, then we find a contradiction by a case analysis on how this transition was derived:

- if R did not participate, i.e., $R' \equiv R_0|R$, the above reasoning about sizes yields a contradiction.
- if R did $\overset{a}{\to}$ and S did $\overset{\overline{a}}{\to}$, i.e., $R' \equiv S_i|R''$ with $R \overset{a}{\to} R''$. We have $\sharp F = \sharp F' + 1 = \sharp R' + 1 = \sharp S_i + \sharp R'' + 1 = \sharp S_i + \sharp R$, so that $S_i \equiv \mathbf{0}$, $R' \equiv R''$, and $\sharp F = \sharp R$. We now play the a transition from $S|R$, so that we get F'' such that $S|F \overset{a}{\to} S|F''$ and $S|F'' \sim S|R''$. If $F \overset{a}{\to} F''$ then $F'' \equiv R''$ by induction, whence $F'' \equiv F'$, which is impossible since we also have $F \overset{\tau}{\mapsto} F'$. Otherwise, if $F'' \equiv S_j|F$, then $\sharp F \leq \sharp R''$ by (†), which is contradictory with $\sharp F = \sharp R$.
- the case where a and $\overline{a}$ are swapped is handled symmetrically.
- if F did not participate, i.e., $R \xrightarrow{a|\overline{a}} R'$, then $\sharp F = \sharp R - 1$, and by firing the two prefixes in sequence, from $S|R$, we find F'' such that $S|F \overset{a}{\to}\overset{\overline{a}}{\to} S|F''$ and $S|F'' \sim S|R''$.
  - If $F \overset{a}{\to}\overset{\overline{a}}{\to} F''$ then $F'' \equiv R''$ by induction, which is impossible since $\sharp F'' = \sharp F - 2$ and $\sharp R'' = \sharp R - 2 = \sharp F - 1$;
  - if F does not participate, i.e. $F'' \equiv R_0|F$, then $\sharp F \leq \sharp R''$ by (†), which contradict $\sharp R'' = \sharp F - 1$;
  - if both S and F did participate, i.e., $F'' \equiv S_i|F_0$ with $F \overset{a}{\to} F_0$ (and $a = \overline{\alpha_i}$), then we play this transition from $S|F$ to find R_0 such that $S|R \overset{a}{\to} S|R_0$ and $S|F_0 \sim S|R_0$. By induction $F_0 \equiv R_0$, which is impossible since $\sharp F_0 = \sharp F - 1$ and $\sharp R_0 \geq \sharp R - 1 = \sharp F$.

If F has no prefix ($F \equiv \mathbf{0}$), then $R \equiv \mathbf{0}$ by Lemma B7(i); otherwise, the above study shows that necessarily $\sharp F = \sharp R$. This allows us to close the right-to-left part of the game: suppose that $R \overset{\alpha}{\mapsto} R'$, then we find F' such that $S|F \overset{\alpha}{\to} S|F'$ and $S|F' \sim S|R'$. If $S|F \overset{\alpha}{\mapsto} S|F'$ then either $F \overset{\alpha}{\mapsto} F'$ and we are done by induction, or $F' \equiv S_i|F$, which is impossible since $S|S_i|F \sim S|R'$ entails $\sharp F \leq \sharp R'$ by (†), while $\sharp R' = \sharp R - 1 = \sharp F - 1$. Otherwise, if $\alpha = \tau$ and $S|F \xrightarrow{a|\overline{a}} S|F'$,

then F necessarily moves, by using (†) as previsouly, and we are left with the following cases:

- $F \xrightarrow{a|\overline{a}} F'$, which is impossible since by induction, $F' \equiv R'$, while $\sharp F' = \sharp F - 2 < \sharp F - 1 = \sharp R - 1 = \sharp R'$;
- $F' \equiv S_i|F''$, with $F \xrightarrow{a} F''$ (and $a = \overline{\alpha_i}$). We play this transition from $S|F$ to find R'' such that $S|R \xrightarrow{a} S|R''$ and $S|F'' \sim S|R''$. By induction $F'' \equiv R''$, so that we necessarily have $R \xrightarrow{a} R''$, by size considerations. Since $S|S_i|F'' \sim S|R'$, we deduce $S|S_i|R'' \sim S|R'$ from $F'' \equiv R''$, whence $S_i|R'' \equiv R'$ by Lemma B8. Since $\sharp R'' = \sharp R' = \sharp R - 1$, we have $S_i \equiv \mathbf{0}$, and $R'' \equiv R'$. This is impossible since $R \overset{\tau}{\mapsto} R'$ and $R \xrightarrow{a} R''$. □

Corollary B11 *If $S|F \sim S$ and $S\#F$, then $F \equiv \mathbf{0}$.*

Lemma B12 *If $S|F_1 \sim S|F_2$ with $S\#F_1$ and $S\#F_2$, then $F_1 \equiv F_2$.*

Proof. We reason by induction over the minimum of the sizes of F_1 and F_2. The case where one of these processes is empty is given by Corollary B11. Suppose w.l.o.g. that $\sharp F_1 \leq \sharp F_2$, we show that $\{(F_1, F_2)\} \cup \equiv$ is a prefix bisimulation. We consider the left-to-right part of the game first: suppose that $F_1 \overset{\alpha}{\mapsto} F_1'$, we deduce $S|F_2 \xrightarrow{\alpha} S|F_2'$ with $S|F_1' \sim S|F_2'$. By induction, $F_1' \equiv F_2'$, and F_2 necessarily moved (otherwise, $\sharp F_2' \geq \sharp F_2 \geq \sharp F_1 > \sharp F_1'$ which is contradictory). If $F_2 \overset{\alpha}{\mapsto} F_2'$, we are done; we are left with the following cases where $\alpha = \tau$.

- Either $F_2 \xrightarrow{a|\overline{a}} F_2'$, in which case, by playing $F_2 \xrightarrow{a} \xrightarrow{\overline{a}} F_2'$, we deduce $S|F_1 \xrightarrow{a} \xrightarrow{\overline{a}} S|F_1''$, and, since $\sharp F_2' = \sharp F_1' < \sharp F_1$, $F_1'' \equiv F_2'$ by induction. Therefore, $F_1' \equiv F_1''$, and in particular $\sharp F_1' = \sharp F_1''$, so that either $F_1 \xrightarrow{a} F_1''$, or $F_1 \xrightarrow{\overline{a}} F_1''$. In both cases, we obtain a contradiction with $F_1 \overset{\tau}{\mapsto} F_1' \equiv F_1''$.
- Or $F_2 \xrightarrow{a} F_2'$ and $S \xrightarrow{\overline{a}} S$. By playing $S|F_2 \xrightarrow{a} S|F_2'$, we obtain $S|F_1 \xrightarrow{a} S|F_1'' \sim S|F_2'$, which gives $F_1'' \equiv F_2'$ by induction, since $\sharp F_2' < \sharp F_1$. Therefore, $F_1'' \equiv F_1'$ and $F_1 \xrightarrow{a} F_1''$, which is contradictory with , $F_1 \xrightarrow{\tau} F_1'$.

Finally, there exists F_2' such that $F_2 \overset{\alpha}{\mapsto} F_2'$, and, in passing, we observe that $\sharp F_1 = \sharp F_2$. The latter equality allows us to replay the same proof by symmetry to get the right-to-left part of the game. We deduce $F_1 \dot{\sim} F_2$, i.e., $F_1 \equiv F_2$. □

We can now characterise bisimilarity on $m\mathrm{CCS}_\tau$ seeds:

Proposition B13 *For all seeds P, P', $P \sim P'$ iff $P \equiv P'$.*

Proof. It suffices to show that $P \sim P'$ entails $P \equiv P'$. Write P and P' as $S|F$ and $S'|F'$, where S, S' are replicated processes. By Prop. 7, $S \sim S'$. Moreover, S and S' are necessarily seeds because P and P' are (hence the notation). Write $S \equiv \prod_{i \leq m} !\alpha_i.S_i | \prod_{i \leq m^\tau} !\tau.S_i^\tau$ and $S' \equiv \prod_{j \leq n} !\alpha_j'.S_j' | \prod_{i \leq n^\tau} !\tau.S_i'^\tau$, where the α_i and α_j' are visible prefixes. Play each visible prefix on the left-hand side and apply Lemma 15 to show that there exists a map $\sigma : [1..m] \to [1..n]$, such that

$\alpha_i.S_i \equiv \alpha'_{\sigma i}.S'_{\sigma i}$ (recall that $S\#S_j$ by Lemma 14(ii)). This map is bijective: we could otherwise construct a smaller seed. Then play the τ prefixes to obtain a second bijection $\sigma^\tau : [1..m^\tau] \to [1..n^\tau]$, such that $S_i^\tau \equiv S'^\tau_{\sigma^\tau i}$; the situation where one process answers with a synchronisation is ruled out using the first bijection (if $S_i^\tau \equiv S'^\tau_{j0}|S'^\tau_{j1}$ then the $!\tau.S_i$ component of S can be removed using law (B) since S already contains $!a.S_{\sigma^{-1}j0}$ and $!a.S_{\sigma^{-1}j1}$.

Therefore, $S \equiv S'$. By Lemma 12(iii), $S\#F$ and $S'\#F'$, which allows us to deduce $F \equiv F'$, using Lemma 15. Finally, $P \equiv P'$. □

B.3 Rewriting Processes to Normal Forms

We need to adapt the rewriting system, so that law (B) is taken into account:

Definition B14 (Rewriting) *Any process T induces a relation between processes, written $\xrightarrow{T}$, defined by the following rules, modulo $\equiv$:*

$$\frac{\alpha.F \Subset T}{C[\alpha.F] \xrightarrow{T} C[\mathbf{0}]}\ (\mathrm{R1}) \qquad \frac{\alpha.F \Subset P}{!\alpha.F \,|\, P \xrightarrow{T} P}\ (\mathrm{R2})$$

For example, these modifications respectively allow one to obtain:

$$!a.b \,|\, !\overline{a}.\tau.b \xrightarrow{!a.b \,|\, !\overline{a}} !a.b \,|\, !\overline{a} \qquad\qquad !a.b \,|\, !\overline{a} \,|\, !\tau.b \xrightarrow{T} !a.b \,|\, !\overline{a}$$

Soundness is established as previously:

Lemma B15 (Soundness) *If $P \xrightarrow{T}{}^* T$, then $P \sim T$.*

Then we proceed with completeness:

Lemma B16 *For all P, either P is a seed, or $P \xrightarrow{\mathsf{s}(P)} P'$ for some P' s.t. $P \sim P'$.*

Proof. Write $P \equiv !F|F^P$ and $\mathsf{s}(P) \equiv S|F^S$, with $F \equiv \prod_i \beta_i.F_i$ and $S \equiv \prod_j !\alpha_j.S_j$. By Prop. 7, and since $P \sim \mathsf{s}(P)$, $!F \sim S$ $(*)$.

Any transition at β_i by $!F$ is answered by S with some splore $\beta_i.E \Subset S$ yielding equivalence $!F|F_i \sim S|E$, which in turn gives $S|F_i \sim S|E$, by injecting $(*)$. By Lemma 15, either (a) $F_i \equiv E$, or (b) $\neg(S\#F_i)$. In the latter case, (b), this means that P admits some spore of S as a sub-term, and can be rewritten using rule (R1), the resulting process being bisimilar to P, by Prop. 2.

Like in the proof of Lemma 21, this remark allows us either to do one rewriting step, or to establish that the $\beta_i.F_i$s with $\beta_i \neq \tau$ and the $\alpha_j.F_j$s with $\alpha_j \neq \tau$ are in one-to-one correspondence (since "visible" spores are just the $\alpha_j.S_j$s). We then proceed with the $\beta_i.F_i$ for which $\beta_i = \tau$: in the case (b), we are done, in the case (a), either $(a.1)$: $E \equiv S_{j0}|S_{j1}$ for some $j0, j1, b$ with $\alpha_{j0} = b$ and $\alpha_{j1} = \overline{b}$, in this case, P can be rewritten by using rule (R2) to remove the $!\tau.F_i$ (since $\alpha_{j0}.S_{j0}$ and $\alpha_{j1}.S_{j1}$ correspond to some $\beta_{i0}.F_{i0}$ and $\beta_{i1}.F_{i1}$ thanks to the bijection between "visible" replicated components); or $(a.2)$: $E = S_{\sigma i}$ for some σi with $\alpha_{\sigma i} = \tau$. If

we are always in case $(a.2)$, this allows us to extend our correspondence to all replicated components (if the correspondence is not injective, then we can use rule (R2) to perform a rewriting step).

To sum up, either P can be rewritten, or $!F \equiv S$. In the latter case, we deduce $S \mid F^P \sim S \mid F^S$ from $(*)$, and since $S \# F^S$ by Lemma 12(iii), there are two cases according to Lemma 15: either $F^P \equiv F^S$, in which case $P \equiv \mathsf{s}(P)$: P is a seed; or $\neg(S \# F^P)$, i.e., F^P admits some spore of S as a sub-term, and we can rewrite P using (R1), getting a process bisimilar to P by Prop. 2. □

Like previously, this allows us to conclude thanks to the characterisation of bisimilarity on seeds (Prop. B13):

Proposition B17 (Completeness) *For all P, $P \xrightarrow{\mathsf{s}(P)}{}^{*} \mathsf{s}(P)$.*

Theorem B18 (Characterisation) *In $mCCS_\tau$, strong bisimilarity coincides with joinability: $P \sim Q$ iff $P \curlyvee Q$.*

Corollary B19 *In $mCCS_\tau$, strong bisimilarity is closed under substitutions.*

Proof. We check that $P \xrightarrow{T} P'$ entails $P\sigma \xrightarrow{T\sigma} P'\sigma$ for any name substitution σ (using the fact that distribution congruence is closed under substitutions). □