On Bisimilarity and Substitution in Presence of Replication

Daniel Hirschkoff, Damien Pous

On Bisimilarity and Substitution in Presence of Replication

Daniel Hirschkoff[1] and Damien Pous[2]

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

Abstract. We provide a characterisation of strong bisimilarity in a fragment of CCS that contains only prefix, parallel composition, synchronisation, and a limited form of replication. The characterisation is not an axiomatisation, but is instead presented as a rewriting system.
Our approach is extended to derive a new congruence result in the π-calculus: congruence holds in the sub-calculus that does not include restriction nor sum, and features a limited form of replication. Whether this property holds with unrestricted replication remains open.

1 Introduction

In this paper, we study algebraic properties of behavioural equivalences, and, more precisely, of strong bisimilarity (written $\sim$). 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). Several axiomatisation results for process calculi that feature an operator of parallel composition ($|$) have been derived by decomposing this operator using other constructs, such as sum or left merge. We, on the contrary, are rather interested in treating parallel composition as a primitive operator, looking for algebraic laws that allow us to describe the properties of parallel composition w.r.t. bisimilarity. An important motivation for this choice is that we want to analyse bisimilarity in (fragments of) the π-calculus, where the sum is usually absent. Indeed, sum can be encoded under certain conditions in π [6]. Moreover, sum is the source of some problems in the behavioural theory – we return to this point below. By essence, existing axiomatisations of bisimilarity in calculi featuring sum do not help when it comes to reason about the π-calculus in absence of sum.

The present work continues [3], which describes an analysis of the algebraic properties of bisimilarity on finite fragments of CCS and the π-calculus. In this paper, we move to process calculi where it is possible to express infinite behaviours: we enrich our setting with a simple form of recursion, which is provided by a restricted form of the operator of *replication* ($!$). More precisely, we work in the sum-free and restriction-free fragments of CCS and π, where replications are allowed only at top-level (no replication can occur under a prefix).

Let us now briefly review some properties of replication w.r.t. bisimilarity in CCS, to illustrate what are the difficulties in analysing bisimilarity in our setting.

Replication can be seen as an 'infinitary version' of parallel composition. Indeed, we have $!a.P \,|\, a.P \sim !a.P$ and $!a.P \,|\, !a.P \sim !a.P$, which intuitively means that a replicated process acts as an unbounded number of copies of that process in parallel. It appears that we can generalise the first equality by allowing a replicated process to erase one of its copies not only at top-level, but arbitrarily deep in a term: we have

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

where C is an arbitrary context ($C[Q]$ stands for the process obtained by replacing the hole with Q in C, and $\mathbf{0}$ is the inactive process). A related phenomenon is expressed by the following law, whose most trivial instance is $!a.a \sim !a$:

$$!a.C[a.C[\mathbf{0}]] \sim !a.C[\mathbf{0}] \ ,$$

Although we show in this paper that the above laws are the basic ingredients we need in order to characterise bisimilarity, they are not sufficient to obtain an axiomatisation of bisimilarity, as the following example shows:

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

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, and we have not been able to come up with a simple, readable, presentation of bisimilarity based on equational laws.

The first contribution of this paper is a syntactic characterisation of (strong) bisimilarity on a fragment of CCS that features top-level replications. This characterisation exploits a rewriting relation on processes (such that P_1 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 consists in establishing congruence of bisimilarity in the corresponding fragment of the π-calculus. At the heart of the π-calculus is the mechanism of *name-passing*, which is the source of considerable expressive power. Name-passing 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 be preserved by all syntactic operators). Congruence of bisimilarity fails to hold even in rather minimal fragments of the π-calculus: in presence of sum, this property fails; as [10,2] shows, this is also the case as soon as replication and restriction are present in the calculus. We have shown in [3] that congruence holds when we renounce to replication; to our knowledge congruence of bisimilarity in the restriction- and sum-free π-calculus with full replication is still an open problem [10].

To derive substitution closure of bisimilarity in π, we concentrate on an analysis of *visible* bisimilarity (sometimes called *io*-bisimilarity [5]), the equivalence obtained from bisimilarity by disallowing challenges along internal communications. Visible bisimilarity is inherently substitution closed, and we show that it coincides with bisimilarity. The technical developments that lead to this result follow to a large extent the path of our proofs in the CCS case (actually, the coincidence of visible and standard bisimilarities shows up already in these proofs). However, the reasoning in the π-calculus cannot be straightforwardly deduced from the CCS case, for two main reasons. First, name binding introduces difficulties related to α-conversion and substitutions. Second, while our proofs on CCS exploit an axiomatisation of bisimilarity in the finite case, established in [3], we lack the counterpart of this result in π, so that we must adapt our proof strategy.

Along our presentation, we provide detailed proofs and present all intermediate steps, so that the contents of this paper are quite technical. We believe this to be necessary, since this kind of congruence properties is quite sensitive to the setting we work with. We moreover view the reasonings we use in our proofs as an important contribution of the present work. In particular, we make a rather heavy use of "up-to techniques" for bisimulation [9,7,8].

Outline. We describe the subset of CCS we work with, and 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, in order to characterise strong bisimilarity. We present our new congruence result for the π-calculus in Sect. 5, and Sect. 6 suggests directions for future work.

2 The Setting

We let a, b range over a countable set of *names*; we work in a subset of CCS, called mCCS, defined by the following grammar:

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

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 comment on this in Remark 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 $\alpha.\beta.\mathbf{0}$. For $F = \alpha_1.F_1 \mid \ldots \mid \alpha_k.F_k$, we shall sometimes write F as $\prod_{i\in[1..k]} \alpha_i.F_i$. We extend the syntactic operator of replication to a function defined over processes by letting

$$!\mathbf{0} \stackrel{\text{def}}{=} \mathbf{0} \qquad !(P|Q) \stackrel{\text{def}}{=} !P|!Q \qquad !!\alpha.F \stackrel{\text{def}}{=} !\alpha.F \ .$$

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

Fig. 1. Labelled Transition System for mCCS

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. Accordingly, we use D to range over (single-hole) *finite contexts*, mapping finite processes to finite processes. Note that the hole cannot occur immediately under a replication in C.

The labelled transition system (LTS) we associate to this process calculus is standard (Fig. 1 – we omit symmetric rules for parallel composition). This LTS yields a standard notion of *bisimilarity*, written $\sim$. We also define the notion of *visible bisimilarity* ($\dot{\sim}$) where silent transitions are not taken into account:

Definition 1 Strong bisimilarity ($\sim$) *is the largest binary relation over processes such that $P \sim Q$ entails:*

(i) if $P \xrightarrow{\mu} P'$ then there exists Q' such that $P' \sim Q'$ and $Q \xrightarrow{\mu} Q'$;
(ii) conversely, if $Q \xrightarrow{\mu} Q'$ then there exists P' such that $P' \sim Q'$ and $P \xrightarrow{\mu} P'$.

Visible bisimilarity ($\dot{\sim}$) *is defined similarly, by restricting (i) and (ii) to the cases where $\mu \neq \tau$.*

These bisimilarities are congruences, and are preserved by the extended replication function. On finite processes, we use the following characterisation [3]:

Definition 2 (Distribution law) *Let $\equiv$ be the smallest congruence generated by the laws of an abelian monoid for parallel composition (the neutral element being $\mathbf{0}$), and the following equation schema, called* distribution law*, where there are as many occurrences of F on both sides of the equation:*

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

It is easy to show that $\equiv$ is decidable. We moreover have:

Theorem 3 ([3]) *$\equiv$ coincides with strong bisimilarity ($\sim$) on finite processes.*

Fact 4 *We have: $\equiv \subseteq \sim \subseteq \dot{\sim}$.*

We shall prove that visible and strong bisimilarity coincide on mCCS (Thm. 23).

We now present some important properties of replicated processes w.r.t. bisimilarity. We first introduce some laws that involve copying replicated subterms, and then present a cancellation property about the the replicated subterms of two arbitrary bisimilar processes.

Proposition 5 *If* $C[\mathbf{0}] \sim !\alpha.F|P$, *then* $C[\mathbf{0}] \sim C[\alpha.F]$.

Proof. We show that $\mathcal{R}= \{(C[\mathbf{0}], C[\alpha.F])\ /\ \forall C \text{ s.t. } C[\mathbf{0}] \sim !\alpha.F|P \text{ for some } P\}$ is a strong bisimulation up to transitivity and parallel composition (cf. [7,8]).

There are several cases to consider in the bisimulation game:

- 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 in the context ($C = !\beta.D|Q$) and this prefix is fired: we have $C[\mathbf{0}] \xrightarrow{\beta} P_l = C[\mathbf{0}]|D[\mathbf{0}]$ and $C[\alpha.F] \xrightarrow{\beta} P_r = C[\alpha.F]|D[\alpha.F]$. Here the processes are not related by $\mathcal{R}$ (remember that we work with single-hole contexts), thus we start to reason by transitivity with $P_c = C[\mathbf{0}]|D[\alpha.F]$: we can deduce $P_l\ \mathcal{R}\ P_c$ by considering the context $C' = C[\mathbf{0}]|D[]$, and checking that $C'[\mathbf{0}] \sim !\alpha.F|P|D[\mathbf{0}]$. Finally, P_c and P_r are related by the closure of $\mathcal{R}$ under parallel contexts (by removing the $D[\alpha.F]$ component).
- 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.

$$!\alpha.F \mid C[\alpha.F] \ \sim\ !\alpha.F \mid C[0] \qquad \text{(A)}$$

$$!\alpha.D[\alpha.D[\mathbf{0}]] \ \sim\ !\alpha.D[\mathbf{0}] \qquad \text{(A')}$$

Note that these laws, as well as Prop. 5, hold for full CCS (and for the π-calculus as well), 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 $\alpha.F$.

From now on, we focus on visible bisimilarity ($\dot{\sim}$), so that we shall sometimes omit the adjective 'visible' in the discussion. We first establish that the finite parts of two bisimilar processes can be cancelled in bisimilarity games. We need for that the following property.

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 latter equivalence in the former, 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, this property is 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 [4]). Moreover, this property also holds for usual bisimilarity ($\sim$). The counterpart of this cancellation property does obviously 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 minimal size such that $P \dot{\sim} Q$.*

When P is a seed of P, we simply say that P is a seed.

In Sect. 4, we show how to rewrite an arbitrary process into a seed. We establish in this section that all seeds of a given process are equated by $\equiv$ (Prop. 15). Note that, because visible bisimilarity is a congruence in our calculus, all finite subterms of a seed are seeds. We also have that finite bisimilar processes necessarily have the same size. Thanks to Prop. 7, the replicated parts of bisimilar seeds are necessarily bisimilar. As a consequence, from this point until Prop. 15, 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 9 (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 $k > 0$, $\alpha_1, .., \alpha_k$, and $P_1, .., P_k$ such that $S \xrightarrow{\alpha_1} P_1 \ldots \xrightarrow{\alpha_k} P_k \equiv S|R$. We shall use R to range over such residual processes.*

Note that if $S \rightsquigarrow R$, then R is a parallel composition of sub-terms of the S_is. We can also remark that residuals and clean processes are stable under transitions: if $S\#F$ (resp. $S \rightsquigarrow F$) and $F \xrightarrow{\alpha} F'$, then $S\#F'$ (resp. $S \rightsquigarrow F'$). As shown by the following lemma, sub-terms of seeds are clean:

Lemma 10 *(i) The finite part of a seed is clean with respect to its replicated part: if $S|F$ is a seed, then $S\#F$;*

(ii) The residuals of a replicated seed are clean: if $S \rightsquigarrow R$, then $S\#R$.

[1] In the present case, 'emptying F_0' means playing all prefixes in 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.

Proof. (i) By contradiction: if $F \equiv D[\alpha_i.S_i]$, then $S|F \dot{\sim} S|D[\mathbf{0}]$ by law (A), which contradicts the minimality hypothesis about $S|F$.

(ii) Again, by contradiction, suppose that $R \equiv D[\alpha_i.S_i]$. By emptying the prefixes of D, we find R' such that $S \rightsquigarrow R'$ and $R' \equiv \alpha_i.S_i$ $(*)$. Now, by definition of $\equiv$, there are two cases to consider:

- $R' = \alpha_i.R''$, with $S_i \equiv R''$. In this case, we deduce from $S \rightsquigarrow R'$ that $\alpha_i.R''$ is a sub-term of one of the S_js: there exist j, D' such that $S_j \equiv D'[\alpha_i.R'']$. From $(*)$, we deduce $S_j \equiv D'[\alpha_i.S_i]$, which yields $S \dot{\sim} \prod_{k\neq j} !\alpha_k.S_k \mid !\alpha_j.D'[\mathbf{0}]$ by (A) (we necessarily have $i \neq j$ by a reasoning over the size of terms). This is contradictory with the fact that S is a seed.
- $R' \equiv \alpha_i.R''|\alpha_i.R''|\ldots|\alpha_i.R''$ and $S_i \equiv R''|\alpha_i.R''|\ldots|\alpha_i.R''$ (by definition of $\equiv$, the distribution law (D) is the only way to relate a prefixed process to a non-trivial parallel composition). We have $!\alpha_i.S_i \equiv !\alpha_i.(R''|\alpha_i.R''|\ldots|\alpha_i.R'') \dot{\sim} !\alpha_i.R''$, so that S cannot be a seed. □

Note that the above proof relies on the syntactical properties of $\equiv$. We shall need to adapt it when moving to the π-calculus (Sect. 5).

The following technical lemma states that any replicated component of a process bisimilar to S can be used to replace one of the components of S (this is actually not specific to seeds – S just needs to be replicated).

Lemma 11 *If $S \dot{\sim} !\alpha.F|P$, then there exists j such that $S \dot{\sim} !\alpha.F \mid \prod_{i\neq j} !\alpha_i.S_i$ and $\alpha_j = \alpha$.*

Proof. By firing $\alpha.F$ on the right-hand side, we find j such that $\alpha_j = \alpha$ and $S|S_j \dot{\sim} !\alpha.F|F|P$, from which we deduce $S|S_j \dot{\sim} S|F$, by re-injecting equivalence $S \dot{\sim} !\alpha.F|P$. Then we show that the singleton relation $\{(S, !\alpha.F|\prod_{i\neq j} !\alpha_i.S_i)\}$ is a bisimulation up to bisimilarity and parallel contexts [9,10].

- when a transition on α_i is triggered, with $i \neq j$, we reason up to parallel composition in order to remove the S_i component on both sides;
- when a transition on α_j (or α) is triggered, we have to relate processes $S|S_j$ and $!\alpha.F|F|\prod_{i\neq j} !\alpha_i.S_i$; we reason up to bisimilarity in order to rewrite $S|S_j$ into $S|F$ and then up to parallel context in order to remove F. □

We now have enough material to prove that the replicated part of bisimilar seeds can be cancelled in bisimilarity games: for any seeds $S|E$ and $S|F$,

$$\text{if } S|F \dot{\sim} S|E \text{ then } F \dot{\sim} E \ .$$

We first consider the case where F is empty, and E is a residual:

Lemma 12 *$S \dot{\sim} S|R$ and $S \rightsquigarrow R$ entail $R = \mathbf{0}$.*

Proof. Suppose by contradiction $R \equiv \alpha.R'|R$". Lemma 6 gives $S \dot{\sim} S|\alpha.R'$, hence $S \dot{\sim} S|!\alpha.R'$ by replicating all processes. By Lemma 11 there exists i such that $S \dot{\sim} !\alpha.R'|\prod_{k\neq i} !\alpha_k.S_k$. Now, since $S \rightsquigarrow R$, there exist j, D such that $S_j \equiv D[\alpha.R']$. If $i = j$, we have obtained a smaller seed ($\sharp\alpha.R' < \sharp\alpha_i.S_i$); otherwise, $!\alpha.R'|!\alpha_j.D[\mathbf{0}]|\prod_{k\neq i,j} \alpha_k.S_k$ is a smaller seed by (A). □

We then solve the case where F is clean and E is a residual. We need to study this configuration in order to be able to reason by induction:

Lemma 13 *If $S|F \dot{\sim} S|R$, $S\#F$, and $S \rightsquigarrow R$, then $F \equiv R$.*

Proof. First notice that $\sharp F \geq \sharp R$: otherwise, by emptying F on the left-hand side, we find $R' \neq \mathbf{0}$ such that $S \dot{\sim} S|R'$ with $S \rightsquigarrow R'$, which is contradictory with Lemma 12. Then we proceed by induction on the size of F, and we show at each step of the induction that the relation $\{(F, R)\} \cup \equiv$ is a visible bisimulation:

- when $F \xrightarrow{\alpha} F'$, we find R' such that $S|R \xrightarrow{\alpha} S|R'$ and $S|F' \dot{\sim} S|R'$. By induction, $F' \equiv R'$, and we deduce a posteriori that R' comes from R: otherwise, we would have $\sharp R' \geq \sharp R = 1 + \sharp F'$ which contradicts $F' \equiv R'$.
- when $R \xrightarrow{\alpha} R'$, either we find F' such that $F \xrightarrow{\alpha} F'$ and $S|F' \dot{\sim} S|R'$, which allows us to close the diagram, by induction; or we find i such that $S|S_i|F \dot{\sim} S|R'$. We show that the latter case is impossible. By emptying R' on the right-hand side, we get R'', F' such that $S|R''|F' \dot{\sim} S$. By the above remark about sizes, $\sharp F > \sharp R'$, whence $F' \neq \mathbf{0}$, so that we can write $F' \equiv \beta.F_0|F_1$ and deduce $S|\beta.F_0 \dot{\sim} S$ by Lemma 6. Then, by firing the β prefix, we find i such that $\beta = \alpha_i$ and $S|F_0 \dot{\sim} S|S_i$. Since $\sharp F_0 < \sharp F' \leq \sharp F$, we can apply the induction hypothesis and deduce that $F_0 \equiv S_i$, whence $\beta.F_0 \equiv \alpha_i.S_i$. This is contradictory with $S\#F$: $\beta.F_0$ is a sub-term of F. □

This finally leads to the announced cancellation result:

Lemma 14 *If $S|F \dot{\sim} S|E$ $S\#F$, and $S\#E$, then $F \equiv E$.*

Proof. First observe that if $\sharp F < \sharp E$, then we can empty F by playing challenges on the left hand side, and we obtain $S \dot{\sim} S|E'$ with $E' \neq \mathbf{0}$, which is impossible by Lemma 13. Hence $\sharp F = \sharp E$, by symmetry.

We then show that $\mathcal{R} = \{(F, E) \ / \ S|F \dot{\sim} S|E\}$ is a bisimulation. If $F \xrightarrow{\alpha} F'$, then $S|F \xrightarrow{\alpha} S|F'$, which by hypothesis entails that $S|E$ can answer this challenge. By the remark above, $S|E$ necessarily answers by firing E, since otherwise we would get equivalent processes with finite parts having different sizes. This allows us to show that E can answer the challenge, and that $\mathcal{R}$ is a bisimulation by symmetry. We conclude with Thm. 3. □

We can now characterise bisimilarity on seeds:

Proposition 15 *Let P, P' be two seeds; if $P \dot{\sim} P'$, then $P \equiv P'$.*

Proof. Write $P \equiv S|F$ and $P' \equiv S'|F'$ where S, S' are replicated processes. By Prop. 7, $S \dot{\sim} S'$. Moreover, as remarked above, 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 13 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 10(ii)). This map is bijective: we could otherwise construct a smaller seed; whence $S \equiv S'$.

By Lemma 10(i), $S\#F$ and $S'\#F'$, which allows us to deduce $F \equiv F'$, using Lemma 14. 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 CCS.

Proposition 16 *If P is a seed and $P \xrightarrow{\mu} P'$, then P' is a seed.*

4 Rewriting Processes to Normal Forms

By Prop. 15, the seed of a process P is unique up to $\equiv$; 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 that preserves strong bisimilarity. This has several consequences about mCCS: visible and strong bisimilarity coincide; bisimilarity is decidable using a rather simple algorithm; bisimilarity is closed under substitutions. The last point will be of particular relevance in the study of the π-calculus (Sect. 5): the corresponding result will lead to congruence of bisimilarity.

Definition 17 (Rewriting) *Any process T induces a relation between processes, written $\xrightarrow{T}$, defined by the following rules, modulo $\equiv$:*

$$\frac{T \equiv !\alpha.F|Q}{C[\alpha.F] \xrightarrow{T} C[\mathbf{0}]}\,(B1) \qquad \frac{}{!\alpha.F|!\alpha.F|P \xrightarrow{T} !\alpha.F|P}\,(B2)$$

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

Before studying how this rewriting system can be related with the notion of seed, we give some intuitions about how it works. 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 $\xrightarrow{T}$ whenever 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 somehow has to be guessed). It is rather a procedure to check that some process T is a 'simplification' of P – Lemma 18 below validates this intuition. Axiom (B2) is rather self-explanatory. We explain how (B1) relates to laws (A) and (A') by considering the following examples:

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

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

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

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

$$!a|!a.b \xrightarrow{!a|!b} !a|!a \xrightarrow{!a|!b} !a \qquad (5)$$

– The first example shows how (B1) can be used to erase redundant sub-terms in the finite part of processes. At each rewrite step, a copy of a component of the seed (here, $!a.b|!c$) is erased: $a.b.c$ is not clean, according to Def. 9. The second example works similarly, except that simplification occurs in a replicated component. These two examples correspond to law (A) above.

– The rewriting step in (3) corresponds to an application of law (A'): a replicated component can be 'simplified with itself'.
– The fourth 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 these two simplifications in sequence.
– Finally, in example (5), 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.

According to the examples above, there is a natural sufficient condition for rewriting to be correct w.r.t. strong bisimilarity:

Lemma 18 (Soundness) *If $P \xrightarrow{T}{}^{*} T$, then $P \sim T$.*

Proof. By induction over the number of rewriting steps. If this number is zero, then this is obvious; suppose now $P \xrightarrow{T} P' \xrightarrow{T}{}^{*} T$. The induction hypothesis gives $P' \sim T$. We reason by cases over the axiom used to rewrite P into P':

– (B1): 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. 5, hence $P \sim T$.
– (B2): we easily have $P \sim P'$, hence $P \sim T$. □

(Note that we implicitly rely on Thm. 3 to reason modulo $\equiv$ in the above proof.) This criterion yields to the following definition:

Definition 19 (Joinability) *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 $Q \xrightarrow{T}{}^{*} T$.*

Remark 20 *This definition of joinability is a priori not effective: for each T, the relation $\xrightarrow{T}$ is decidable (it is finitely branching and strongly normalising), but we are not told how to find T. The proof of Thm. 23 below solves this point: in order to test whether $P \curlyvee Q$, it suffices to search for T among the processes whose size is smaller than both P and Q.*

The other property which is required in order to characterise strong bisimilarity is *completeness* of the rewriting system, i.e., strong 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:

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

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

Any transition at α_i by $!F$ is answered by S at some $\alpha_{\sigma i}$, yielding $!F|F_i \mathrel{\dot\sim} S|S_{\sigma i}$, which in turn gives $S|F_i \mathrel{\dot\sim} S|S_{\sigma i}$, by injecting $(*)$. By Lemma 13, 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 axiom (B1), the resulting process being bisimilar to P.

Suppose now that we are in case (a) for all transitions from $!F$, that is, for all i, there exists σi such that $\alpha_i.F_i \equiv \alpha_{\sigma i}.S_{\sigma i}$. 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 axiom (B2). 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 $S \mid F^P \mathrel{\dot\sim} S \mid F^S$ from $(*)$, and since $S \# F^S$ by Lemma 10(i), there are two cases according to Lemma 14:

- 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 (B1), getting a process bisimilar to P. □

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: we are done; or $P \xrightarrow{\mathsf{s}(P)} P'$ with $P \mathrel{\dot\sim} P'$. We easily check that $\sharp P' < \sharp P$ so that by induction, we have $P' \xrightarrow{\mathsf{s}(P')}{}^{*} \mathsf{s}(P')$. Since $P \mathrel{\dot\sim} P'$, we have $\mathsf{s}(P) \equiv \mathsf{s}(P')$ by Prop. 15, which allows us to conclude that $P \xrightarrow{\mathsf{s}(P)} P' \xrightarrow{\mathsf{s}(P)}{}^{*} \mathsf{s}(P)$, as the rewriting system is defined modulo $\equiv$. □

Thanks to our characterisation of bisimilarity on seeds (Prop. 15), we obtain:

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

Proof. By Lemma 18 and Fact 4, we have $\curlyvee \subseteq \sim \subseteq \mathrel{\dot\sim}$.
Conversely, suppose that $P \mathrel{\dot\sim} Q$. By Prop. 22, $P \xrightarrow{\mathsf{s}(P)}{}^{*} \mathsf{s}(P)$ and $Q \xrightarrow{\mathsf{s}(Q)}{}^{*} \mathsf{s}(Q)$. Moreover, $\mathsf{s}(P) \equiv \mathsf{s}(Q)$ by Prop. 15, so that $P \curlyvee Q$. □

Not only this result states that we do not need to play silent transitions in bisimulation games: according to Remark 20, it actually gives a way to decide whether $P \sim Q$. Another consequence is substitution closure of bisimilarity in mCCS: since silent transitions are not played in visible bisimilarity, this property holds easily for this equivalence.

Remark 24 *The τ prefix is not included in our presentation of CCS. Adding τ to the syntax of actions (α) breaks the inclusion $\sim \subseteq \mathrel{\dot\sim}$: tests performed by $\mathrel{\dot\sim}$ on silent transitions are too restrictive. For example, for $P = !a.\overline{a}|\overline{a}$, we have $P \sim P|\tau$ but $P \not\mathrel{\dot\sim} P|\tau$ since the τ challenge on the right can only be answered with a synchronisation. We did not find a satisfactory way of including the τ prefix in the present work.*

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

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

Fig. 2. Labelled Transition System for $m\pi$

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 [10]. The third reason is that unlike in the CCS case, [3] does not provide an axiomatisation of $\sim$ on finite π-calculus processes: only closure of $\sim$ under substitutions is established. As a consequence, we need to reason modulo bisimilarity on finite processes rather than modulo the (syntactical) congruence $\equiv$.

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 grammar:

$$\alpha, \beta \ ::= \ a(x) \mid \overline{a}\langle b\rangle \ .$$

As usual, the operator of input prefix is binding, we write $\mathrm{fn}(P)$ for the set of free names of P, and we let $P\{y/x\}$ stand for the capture-avoiding substitution of x with y in P. We say that a relation is *substitution closed* if two related processes remain related when applying an arbitrary substitution. Note that contexts (C) can bind names (e.g., $a(x).[]$). The LTS for $m\pi$ is presented on Fig. 2, where symmetric rules for parallel composition are omitted. Note that 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 binary relation on closed processes such that* $P \sim Q$ *entails* $\mathrm{fn}(P) = \mathrm{fn}(Q)$ *and:*

(i) if $P \xrightarrow{\mu} P'$ *then there exists* Q' *s.t.* $Q \xrightarrow{\mu} Q'$ *and* $P' \sim Q'$*;*
(ii) if $Q \xrightarrow{\mu} Q'$ *then there exists* P' *s.t.* $P \xrightarrow{\mu} P'$ *and* $P' \sim Q'$*.*

Visible ground bisimilarity*, denoted by* $\dot{\sim}$*, is defined similarly, by restricting* (i) *and* (ii) *to the cases where* $\mu \neq \tau$*.*

Since we lack the restriction operator, it is easy to check that the condition on free names is actually enforced by standard notions of bisimilarity. Therefore, this definition coincides with the standard definition of ground bisimilarity on $m\pi$: input prefixes are tested with fresh names (whether we test with a single fresh name or with all fresh names does not matter). As for CCS, we show that visible and ground bisimilarities coincide. Since visible bisimilarity is easily shown to be substitution closed (Prop. 26 below, whose proof is given in appendix, like for most results from this section), this allows us to deduce congruence and coincidence with the other notions of bisimilarity (Corollary 36).

Proposition 26 $\dot{\sim}$ *is a substitution closed congruence.*

The following theorem is a consequence of the results from [3]. Although this is not a syntactical characterisation of bisimilarity on finite π processes, it will be used as a replacement for Thm. 3 to reason in the π-calculus.

Theorem 27 *On finite* $m\pi$ *processes,* $\dot{\sim}$ *and* $\sim$ *coincide and define a substitution closed congruence.*

The reasoning then goes basically along the same path as for CCS, except that we rely on the following congruence instead of $\equiv$. In doing so, we work with less 'syntactical' definitions; in particular, the rewriting system is no longer effective.

Definition 28 *We define* $\cong$ *as the smallest congruence that satisfies the laws of an abelian monoid for* $(|, \mathbf{0})$ *and contains the restriction of* $\dot{\sim}$ *to finite processes.*

Lemma 29 *We have* $\quad \cong \subseteq \sim \subseteq \dot{\sim}$.

Intuitively, by Thm. 27 and Lemma 29, $\cong$ satisfies all the properties of $\equiv$ that we needed in the CCS case. As expected, we need to impose conditions on name binding when stating results involving contexts. For example, Prop. 5 becomes:

Proposition 30 *If* $C[\mathbf{0}] \sim\ !\alpha.F|P$*, where* C *does not bind any free name of* $\alpha.F$*, then* $C[\mathbf{0}] \sim C[\alpha.F]$*.*

Since we lack the counterpart of Thm. 3, we define seeds more carefully:

Definition 31 ($m\pi$ seeds) *A* seed of P *is a process* Q *of minimal size such that* $P \dot{\sim} Q$*, whose number of replicated components is maximal among the processes of minimal size.*

This refinement is required in order to prove the counterpart of Lemma 10(ii) – Lemma A8 in the appendix. All proofs go through, and we obtain:

Proposition 32 *Suppose $P \dot{\sim} P'$, where P and P' are seeds. Then $P \cong P'$.*

The rewriting relation is defined modulo $\cong$ instead of $\equiv$ (Def. A13). Since axioms (B1) and (B2) are defined up to $\cong$, they are in particular defined modulo α-conversion. Note moreover that we need to go against a Barendregt convention in order to be able to apply them to input prefixes, as illustrated on the following valid rewriting step: $!a(x).(\overline{x}\langle b\rangle | a(x).\overline{x}\langle b\rangle) \xrightarrow{!a(x).\overline{x}\langle b\rangle} !a(x).\overline{x}\langle b\rangle$.

Lemma 33 (Soundness) *If $P \xrightarrow{T}{}^{*} T$, then $P \sim T$.*

Proposition 34 (Completeness) *For all P, $P \xrightarrow{\mathsf{s}(P)}{}^{*} \mathsf{s}(P)$.*

Theorem 35 (Characterisation) *$P \dot{\sim} Q$ iff $P \sim Q$.*

Corollary 36 *In $m\pi$, early, late 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 several important algebraic properties of replication w.r.t. strong bisimilarity. By extending this result to the π-calculus, we have established congruence of strong bisimilarity in the corresponding fragment (Corollary 36). In [2], a counterexample that exploits the operators of restriction and replication is presented to establish non-congruence of bisimilarity. In light of [3, Corollary 5.9] and Corollary 36, it seems that failure of congruence originates from an interaction between these operators.

We conclude with directions for future work.

Weak bisimilarity. Unlike in the finite calculus of [3], where weak bisimilarity ($\approx$) coincides with strong bisimilarity (up to the removal of τ-prefixes, if they are included in the syntax), more complex laws appear when moving to the weak case, in presence of replication. For example, the following equations (relating seeds in the sense of strong bisimilarity) are valid for weak bisimilarity:

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

These equalities show that the notion of seed has to be defined with more care: in both cases, it is not clear which process should be viewed as the seed. The first equivalence above can be generalised to the following law:

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

where $\langle P\rangle_a$ is defined homomorphically, except for $\langle a.P\rangle_a = \langle \overline{a}.P\rangle_a = \langle P\rangle_a$. We do not know how to generalise the second equivalence.

Guarded replications. We have started investigating the case where replication is not at top-level, but where nested replications – that is, 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 $!\alpha$ and $\alpha.!\alpha$. 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 the fact that seeds are no longer stable under reduction (Prop. 16): for example, $!a.b|c.!b$ is a seed, while its reduct along c, $!a.b|!b$ is not.

Related to this question is the work on HOcore [5], a restriction-free higher order π-calculus: strong bisimilarity is characterised by the distribution law, and (potentially guarded) replications can be encoded using the higher order features. This encoding is not fully abstract, however, so that these results do not entail substitution closure in presence of 'real' replication.

Handling nested replications seems even more challenging. More generally, we hope that understanding bisimilarity in CCS would help in establishing new (non-)congruence results for the π-calculus, as we have done in the present work.

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*. Springer Verlag, 2005.
2. M. Boreale and D. Sangiorgi. Some Congruence Properties for π-calculus Bisimilarities. *TCS*, 198:159–176, 1998.
3. D. Hirschkoff and D. Pous. A Distribution Law for CCS and a New Congruence Result for the Pi-calculus. *LMCS*, 4(2), 2008.
4. Y. Hirshfeld and M. Jerrum. Bisimulation Equivalence is Decidable for Normed Process Algebra. Technical Report ECS-LFCS-98-386, LFCS, 1998. an abstract has appeared in the proceedings of ICALP'99, LNCS 1644, Springer Verlag.
5. I. Lanese, J. A. Pérez, D. Sangiorgi, and A. Schmitt. On the expressiveness and decidability of higher-order process calculi. In *Proc. LICS '08*, pages 145–155. IEEE Computer Society, 2008.
6. U. Nestmann and B.C. Pierce. Decoding Choice Encodings. In *Proc. of CONCUR'96*, volume 1119 of *LNCS*, pages 179–194. Springer Verlag, 1996.
7. D. Pous. Complete Lattices and Up-to Techniques. In *Proc. APLAS '07*, volume 4807 of *LNCS*, pages 351–366. Springer Verlag, 2007.
8. D. Pous. *Techniques modulo pour les bisimulations*. PhD thesis, ENS Lyon, 2008.
9. D. Sangiorgi. On the Bisimulation Proof Method. *Journal of Mathematical Structures in Computer Science*, 8:447–479, 1998.
10. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

A Proofs about the $m\pi$-calculus

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

We let σ range over capture-avoiding name substitutions.

Lemma A1 *$P\sigma \xrightarrow{a_0(x)} P_0$ 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 (26) *$\dot{\sim}$ is a substitution closed congruence.*

Proof. Using Lemma A1, we show that $\{(P\sigma, Q\sigma) \ / \ \sigma, \ P \dot{\sim} Q\}$ is a visible ground bisimulation (this is due to the fact that do not test challenges along silent transitions – however, unlike for CCS, we cannot fix the substitution). Then, congruence is almost immediate: we use substitution closure in order to handle the input prefix. □

In [3], ground bisimilarity is shown to be a congruence on the finite π-calculus without going through visible bisimilarity. The results obtained in [3] nevertheless allow to conclude that visible and ground bisimilarities coincide:

Theorem (27) *On finite $m\pi$ processes, $\dot{\sim}$ and $\sim$ coincide and form a substitution closed congruence.*

Proof. By Prop. 26, it suffices to prove that $\dot{\sim} \subseteq \sim$. We exploit a technical result from [3], the absence of 'mutual desynchronisation' (Lemma 4.4), i.e.,

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

In [3], 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 [3] – the translation transforms visible ground bisimilar π-calculus processes into bisimilar CCS processes, so that the absence of mutual desynchronisation can be established w.r.t. $\dot{\sim}$ in π).

Using this property, we show that the restriction of $\dot{\sim}$ to finite processes is a ground bisimulation, i.e., that challenges along silent transitions can be answered: suppose that $E \dot{\sim} F$, 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, $E \dot{\sim} F$ 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 corresponds to the firing of concurrent prefixes, then we can deduce $F \xrightarrow{\tau} F_i'$ so that the diagram can be closed;

– if both of these correspond to the firing of sequential prefixes, that is, $F_1 \equiv a(b).U|\overline{a}\langle b\rangle.V|F_0$ 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. □

Lemma (29) *The following inclusions hold:* $\cong \subseteq \sim \subseteq \dot{\sim}$.

Proof. The second inclusion is immediate from the definitions. For the first inclusion, we show that $\cong$ is a ground bisimulation. We exploit the fact that $\sim$ is substitution closed on finite processes (Thm. 27) in order to handle replicated input prefixes: if $!a(x).F \cong !a(x).E$ because $F \cong E$, then $F \sim E$ ($\cong$ and $\sim$ coincide on finite processes, according to Thm. 27), so that when $!a(x).F \xrightarrow{a(y)} !a(x).F \,|\, F\{y/x\}$, $!a(x).E$ answers with the obvious transition, and we check that $!a(x).F \,|\, F\{y/x\} \cong !a(x).E \,|\, E\{y/x\}$: thanks to substitution closure, we deduce $F\{y/x\} \sim E\{y/x\}$. □

Proposition (30) *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. 5, 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 the following laws, with the extra proviso that C (resp. D) does not bind any name occurring free in $\alpha.F$ (resp. $\alpha.D$):

$$!\alpha.F \mid C[\alpha.F] \;\sim\; !\alpha.F \mid C[0] \qquad \text{(A)}$$

$$!\alpha.D[\alpha.D[\mathbf{0}]] \;\sim\; !\alpha.D[\mathbf{0}] \qquad \text{(A')}$$

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

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

Proposition A3 *If* $!F|F_0 \dot{\sim} !E|E_0$ *with* F_0, E_0 *finite, then* $!F \dot{\sim} !E$*.*

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. □

From this point until the proof of Prop. 32, we consider a seed S having only replicated components: $S = \prod_i !\alpha_i.S_i$.

Definition A4 (Clean process, residual – counterpart of Def. 9)

– F *is* clean w.r.t. S*, written* $S\#F$*, if it is not the case that for some* i *and finite context* D*,* $F \cong D[\alpha_i.S_i]$*, 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 $k > 0$, $\alpha_1, .., \alpha_k$, and $P_1, .., P_k$ such that $S \xrightarrow{\alpha_1} P_1 \ldots \xrightarrow{\alpha_k} P_k \cong S|R$. We shall use R to range over such residual processes.*

Residuals are stable under reduction: if $S \rightsquigarrow R$ and $R \xrightarrow{\alpha} R'$, then $S \rightsquigarrow R'$. For the stability of clean processes, we need an additional condition:

Fact A5 *If F is clean, $F \xrightarrow{\alpha} F'$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(S) = \emptyset$, then F' is clean.*

Fact A6 *If $S \rightsquigarrow \alpha.R|R'$, then there exists j, D such that D does not capture any free name of S and $S_j \cong D[\alpha.R]$.*

Fact A7 *If $E \dot{\sim} F$ then $\sharp E = \sharp F$.*

Lemma A8

(i) The finite part of a seed is clean with respect to its replicated part: if $S|F$ is a seed, then $S\#F$;
(ii) The residuals of a replicated seed are clean: if $S \rightsquigarrow R$, then $S\#R$.

Proof. (i) By contradiction: if $F \cong D[\alpha_i.S_i]$, then $S|F \cong S|D[\alpha_i.S_i] \dot{\sim} S|D[\mathbf{0}]$ by law (A), which contradicts the minimality hypothesis about $S|F$.

(ii) Again, by contradiction, suppose that $R \cong D[\alpha_i.S_i]$. By emptying the prefixes of D, we find R' such that $S \rightsquigarrow R'$ and

$$R' \cong \alpha_i.S_i \ . \qquad (*)$$

Necessarily, R' can be written as $\prod_{j \leq k} \alpha_i.R_j$; we distinguish two cases – the second one corresponds to the case where we needed to reason about the syntactical structure of $\equiv$, in the proof of Lemma 10.

– If $k = 0$, i.e., R' is a prefixed process (of the form $\alpha_0.R_0$), we deduce from $S \rightsquigarrow R'$ that $S_j \cong D'[R']$ for some j, D' (Fact A6). From $(*)$, we deduce $S_j \cong D'[\alpha_i.S_i]$, which yields $S \dot{\sim} \prod_{k \neq j} !\alpha_k.S_k \mid !\alpha_j.D'[\mathbf{0}]$ by (A) (we necessarily have $i \neq j$ by a size consideration). This is contradictory with the fact that S is a seed.
– Suppose now $k > 0$, i.e., R' contains at least two non-empty parallel components. From $(*)$, we deduce $S \dot{\sim} \prod_{j \neq i} !\alpha_j.S_j \mid \prod_{j \leq k} !\alpha_i.R_j$. We check that the latter process has the same size as S (the replacement of $\alpha_i.S_i$ with R' preserves the size, by Fact. A7). It has strictly more replicated components, however. Because of our refinement of the notion of seed, this is a contradiction. □

Lemma A9 *If $S \dot{\sim} !\alpha.F|P$, then there exists j such that $S \dot{\sim} !\alpha.F \mid \prod_{i \neq j} !\alpha_i.S_i$ and $\alpha_j = \alpha$.*

Proof. We analyse the case where $\alpha = a(x)$: we can suppose w.l.o.g. that $x \notin \mathrm{fn}(P)$ (hence $x \notin \mathrm{fn}(S)$). By firing $a(x).F$ on the right-hand side, we find j such that $\alpha_j = a(x)$ and $S|S_j \dot{\sim} !a(x).F|F|P$ (this might involve α-converting the jth component of S so that $\alpha_j = a(x)$ — this is possible because $x \notin \mathrm{fn}(S)$). This allows us to deduce $S|S_j \dot{\sim} S|F$, and we finish the proof like in the CCS case. □

Lemma A10 *$S \dot{\sim} S|R$ and $S \rightsquigarrow R$ entail $R = \mathbf{0}$.*

Proof. Suppose by contradiction $R = \alpha.R'|R$". By Lemma A2, we have $S \dot{\sim} S|\alpha.R'$ hence $S \dot{\sim} S|!\alpha.R'$ by replicating all processes. By Lemma A9 there exists i such that $S \dot{\sim} !\alpha.R'|\prod_{k\neq i}!\alpha_k.S_k$. Now, since $S \rightsquigarrow R$, there exist some j, D such that $S_j \cong D[\alpha.R']$; if $i = j$, we have obtained a smaller seed; otherwise, we use (A) to show that $!\alpha.R'|!\alpha_j.D[\mathbf{0}]|\prod_{k\neq i,j}\alpha_k.S_k$ is a smaller seed (since $S \dot{\sim} S|\alpha.R'$, $\mathrm{fn}(\alpha.R') \subseteq \mathrm{fn}(S)$, so that D does not bind inappropriate names according to Fact A6). □

Lemma A11 *If $S|F \dot{\sim} S|R$, $S\#F$, and $S \rightsquigarrow R$, then $F \cong R$.*

Proof. First notice that $\sharp F \geq \sharp R$: otherwise, by emptying F on the left-hand side we find $R' \neq \mathbf{0}$ such that $S \dot{\sim} S|R'$ with $S \rightsquigarrow R'$, which is contradictory with Lemma A10. Then we proceed by induction on the size of F, and we show that the relation $\{(F, R)\} \cup \cong$ is a visible ground bisimulation:

- when $F \xrightarrow{\alpha} F'$, we find R' such that $S|R \xrightarrow{\alpha} S|R'$ and $S|F' \dot{\sim} S|R'$. By induction, $F' \cong R'$ ($\mathrm{bn}(\alpha)$ cannot intersect $\mathrm{fn}(S)$, so that $S\#F'$ by Fact A5). We deduce that R' is a derivative of R: otherwise, we would have $\sharp R' \geq \sharp R = 1 + \sharp F'$ which contradicts $F' \dot{\sim} R'$ by Fact A7.
- when $R \xrightarrow{\alpha} R'$, either we find F' such that $F \xrightarrow{\alpha} F'$ and $S|F' \dot{\sim} S|R'$, which allows us to close the diagram, by induction; or we find i such that $S|S_i|F \dot{\sim} S|R'$. We show that the latter case is impossible. By emptying R' on the right-hand side, we get R'', F' such that $S|R''|F' \dot{\sim} S$. By the above remark $F' \neq \mathbf{0}$, so that we can write $F' = \alpha.F_0|F_1$ and deduce $S|\alpha.F_0 \dot{\sim} S$ by Lemma A2. Then, by firing the α prefix, we find i such that $\alpha = \alpha_i$ and $S|F_0 \dot{\sim} S|S_i$. We check that $\sharp F_0 < \sharp F' \leq \sharp F$ so that we can apply the induction hypothesis and deduce that $F_0 \cong S_i$, whence $\alpha.F_0 \cong \alpha_i.S_i$ ($\cong$ is a congruence by definition). We show that this is contradictory with $S\#F$. Indeed, by construction, there exists a sequence of transitions leading from $S|F$ to $S|F'$. Therefore, F can be written as $D[F']$ up to α-conversion, where D does not bind any free name of S. By considering $D' = D|F_1$, we have $F \cong D'[\alpha.F_0]$; Moreover, since $\alpha.F_0 \cong \alpha_i.S_i$, $\mathrm{fn}(\alpha.F_0) \subseteq \mathrm{fn}(S)$, and D' cannot bind free names of $\alpha.F_0$. Finally, $F \cong D'[\alpha_i.S_i]$, which contradicts $S\#F$. □

Lemma A12 *If $S|F \dot{\sim} S|E$ $S\#F$, and $S\#E$, then $F \cong E$.*

Proof. Same proof as for Lemma 14, except that we do not conclude using Thm. 3. □

Proposition (32) *Suppose $P \dot{\sim} P'$, where P and P' are seeds. Then $P \cong P'$.*

Proof. Same proof as for Prop. 15. □

Definition A13 ($m\pi$ rewriting – counterpart to Def. 17) *Any process T induces a relation between processes, written $\xrightarrow{T}$, defined by the following axioms, modulo $\cong$:*

$$\frac{T \cong !\alpha.F|Q \qquad \mathrm{cn}(C) \cap \mathrm{fn}(\alpha.F) = \emptyset}{C[\alpha.F] \xrightarrow{T} C[\mathbf{0}]}(B1) \qquad \frac{}{!\alpha.F|!\alpha.F|P \xrightarrow{T} !\alpha.F|P}(B2)$$

(Where $\mathrm{cn}(C)$ denotes the set names captured by C.) The reflexive transitive closure of $\xrightarrow{T}$ is written $\xrightarrow{T}{}^*$.

Lemma A14 *For all P, either P is a seed, or $P \xrightarrow{\mathsf{s}(P)} P'$ for some P' such that $P \dot{\sim} P'$.*

Proof. Same proof as for Lemma 21. □

Lemma (33 – Soundness) *If $P \xrightarrow{T}{}^* T$, then $P \sim T$.*

Proof. Same proof as for Lemma 18, except that we use Thm. 27 instead of Thm. 3 to reason modulo $\cong$. □

Proposition (34 – 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: we are done; or $P \xrightarrow{\mathsf{s}(P)} P'$ with $P \dot{\sim} P'$. We easily check that $\sharp P' < \sharp P$ so that by induction, we have $P' \xrightarrow{\mathsf{s}(P')}{}^* \mathsf{s}(P')$. Since $P \dot{\sim} P'$, we have $\mathsf{s}(P) \cong \mathsf{s}(P')$ by Prop. 15, which allows us to conclude that $P \xrightarrow{\mathsf{s}(P)} P' \xrightarrow{\mathsf{s}(P)}{}^* \mathsf{s}(P)$, the rewriting system being defined modulo $\cong$. □

Theorem (35 – Characterisation) *$P \dot{\sim} Q$ iff $P \sim Q$.*

Proof. Suppose that $P \dot{\sim} Q$. By Prop. 34, $P \xrightarrow{\mathsf{s}(P)}{}^* \mathsf{s}(P)$ and $Q \xrightarrow{\mathsf{s}(Q)}{}^* \mathsf{s}(Q)$, so that $P \sim \mathsf{s}(P)$ and $Q \sim \mathsf{s}(Q)$ by Lemma 33. We conclude with Prop. 32: $\mathsf{s}(P) \cong \mathsf{s}(Q)$. The reverse implication was proved by Lemma 29. □

We finally prove stability of seeds under transitions (this results holds both for $m\pi$ and mCCS).

Proposition (16) *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, $S \dot{\sim} S'$, so that $S \equiv S'$ by Prop. 15. We conclude with Lemma 14: $F' \equiv F''$, so that P' is indeed also a seed. □