On characterising strong bisimilarity in a fragment of CCS with replication

Daniel Hirschkoff, Damien Pous

On characterising strong bisimilarity in a fragment of CCS with replication – note –

Daniel Hirschkoff[1] and Damien Pous[2]

[1] ENS Lyon, Université de Lyon, CNRS, INRIA
[2] SARDES, LIG, Grenoble, CNRS, INRIA

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.
We discuss how our method allows us 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. We have not formalised the latter result in all details.

1 Introduction

We study algebraic properties of strong bisimilarity in a sub-calculus of CCS. Like in previous work [1], of which the present study is a continuation, an important aspect of the setting we analyse is the absence of the sum construct, and, more generally, of any operator that would allow us to decompose parallel composition.

We present a rewriting system that allows us to characterise strong bisimilarity ($\sim$) in a very basic calculus that only features prefixes, parallel composition, and replicated prefixes, with the additional constraint that these can occur only at top-level. The restriction and choice (or sum) operators are not included. Handling replication is the novel aspect w.r.t. [1], and raises several difficulties when trying to analyse the algebraic properties of $\sim$.

Let us focus on the properties of replication w.r.t. strong bisimilarity. In our setting, the most important bisimilarity law for replication is written

$$!a.P \,|\, a.P \;=\; !a.P \;,$$

and expresses that a replicated process acts as an unbounded number of copies of that process in parallel.

It appears that we can generalise the above equality, by allowing a replicated process to erase one of its copies (we are reading the equality from left to right here) not only at top-level, but arbitrarily deep in a term. In other words, if C is a context (a process with a hole), the law

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

should hold for strong bisimilarity (the previous equality is obtained by taking $C = [\,]$).

This equality, together with the law $!a.P|!a.P = !a.P$, are the basic ingredients we need in order to characterise strong bisimilarity between replicated terms. However, these equations are not enough, as the following example shows: process $P_1 = !a.(b|a.c)|!a.(c|a.b)$ is bisimilar to $P_2 = !a.b|!a.c$. It seems reasonable to consider P_2 as the normal form of P_1. Intuitively, 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 $!a.b$. A related difficulty appears with equalities like $!a.(b|a.b) = !a.b$, where the copy is inserted in the replicated component itself.

Describing this phenomenon of "mutual replication" in all its generality would lead to complicated equational schemata, and we have not been able to come up with a simple, readable, presentation of strong bisimilarity based on equational laws. Instead, we introduce a rewriting relation on processes that allows us to compute normal forms w.r.t. strong bisimilarity (in particular, we are able to rewrite P_1 into P_2). This

$$\frac{}{\alpha.F \xrightarrow{\alpha} F} \qquad \frac{F_1 \xrightarrow{\alpha} F_1'}{F_1 \mid F_2 \xrightarrow{\alpha} F_1' \mid F_2} \qquad \frac{F_2 \xrightarrow{\alpha} F_2'}{F_1 \mid F_2 \xrightarrow{\alpha} F_1 \mid F_2'}$$

$$\frac{}{!\alpha.F \xrightarrow{\alpha} !\alpha.F \mid F} \qquad \frac{P_1 \xrightarrow{\alpha} P_1'}{P_1 \mid P_2 \xrightarrow{\alpha} P_1' \mid P_2} \qquad \frac{P_2 \xrightarrow{\alpha} P_2'}{P_1 \mid P_2 \xrightarrow{\alpha} P_1 \mid P_2'}$$

Fig. 1. Labelled Transition System for our Subset of CCS

has the advantage of exposing the basic laws that are at work when normalising a process. We show that our characterisation of strong bisimilarity still holds when we enrich the calculus with synchronisation. In turn, the method we describe can be applied to derive a new congruence result on a subset of the π-calculus (we must say we have not checked all details of this result yet).

Outline. We describe the subset of CCS we work with in Sect. 2; in Sect. 3, we introduce a notion of normal forms and prove useful some technical results. The rewriting system is defined in Sect. 4, where we show that it allows us to reach normal forms. Section 5 is devoted to the extension of our results to a calculus with synchronisations, closer to the standard CCS. In Sect. 6, we give concluding remarks, discussing in particular how these results lead to a new congruence property in the π-calculus.

2 The Setting

We work in the subset of CCS defined by the following grammar, where we rely on a countable set of *actions* $\alpha, \beta, \ldots$:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}r} F & ::= & \mathbf{0} \mid \alpha.F \mid F|F & P,Q & ::= & F \mid !\alpha.F \mid P|P & \text{(processes)} \\ D & ::= & [] \mid \alpha.D \mid D|F & C & ::= & D \mid !\alpha.D \mid C|P & \text{(contexts)} \end{array}$$

Our calculus features no communication, no restriction, no sum, and allows replication only on prefixes, at top-level. We use P, Q to range over processes. A *finite* process (F) is a process which does not contain an occurrence of the replication operator. For $F = \alpha_1.F_1 \mid \ldots \mid \alpha_k.F_k$, we shall sometime write F as $\prod_{i\in[1..k]} \alpha_i.F_i$, and denote by $!F$ or $\prod_{i\in[1..k]} !\alpha_i.F_i$ the process $!\alpha_1.F_1 \mid \ldots \mid !\alpha_k.F_k$. Note that $!F$ will always denote a process having replicated components only.

We use C to 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 directly under a replication in C.

The labelled transition system associated to this process calculus is standard (Fig. 1 – note that there is no synchronisation rule, this will be addressed in Sect. 5), and yields a notion of *strong bisimilarity*, written $\sim$, which is a congruence.

We shall rely on the following characterisation of strong bismilarity for finite processes, which is established in [1]:

Definition 1 (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.*

$$\alpha.(F|\alpha.F|\ldots|\alpha.F) \;=\; \alpha.F|\alpha.F|\ldots|\alpha.F \;,$$

It is easy to show that this congruence is decidable, and we have

Theorem 2 *$\equiv$ coincides with strong bisimilarity ($\sim$) on finite processes.*

3 Preliminary Technical Results

We present some technical results about strong bisimilarity. Most of these help us isolating the replicated part from the finite part in processes being compared. Indeed, when characterising strong bisimilarity, we shall prove that $P \sim Q$ implies that the replicated parts of P and Q are bisimilar, and we also need somehow to reason about the finite parts of P and Q.

The following property is necessary to derive correction of the rewrite system we define below.

Proposition 3 *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. [3, 2]).

There are three 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 $Q' \sim\ !\alpha.F|F|P$. By injecting the latter equality, we obtain $Q' \sim Q|F$ 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 $C[\alpha.F] \xrightarrow{\beta} P_r = C[\alpha.F]|D[\alpha.F]$. This is where we need the up-to technique: these processes are not related by $\mathcal{R}$ (recall that we work with single-hole contexts). However, we can deduce $P_1\ \mathcal{R}\ P_c = C[\mathbf{0}]|D[\alpha.F]$, by considering the context $C' = C[\mathbf{0}]|D[]$, and checking that $C'[\mathbf{0}] \sim\ !\alpha.F|P|D[\mathbf{0}]$. We finally check that P_c and P_r are related by the closure of $\mathcal{R}$ under parallel contexts (by removing the $D[\alpha.F]$ component).
- in the last case, either the hole occurs under a non-replicated prefix in the contexts ($C = \beta.D|Q$), or the contexts triggers a transition that does not involve or duplicate the hole; this case is treated by a simple reasoning – just play the bisimulation game. □

As a consequence, we obtain the validity of the following laws:

$$!\alpha.F \mid C[\alpha.F] \ \sim\ !\alpha.F \mid C[0] \quad (A) \qquad\qquad !\alpha.D[\alpha.D[\mathbf{0}]] \ \sim\ !\alpha.D[\mathbf{0}] \quad (A')$$

Lemma 4 *If $!F \sim \alpha.F'|Q$, then $!F \sim\ !F|\alpha.F'$.*

Proof. Purely algebraically: replicate everything and add $!\alpha.F'$ in parallel, this yields $!F|!\alpha.F' \sim\ !\alpha.F'|!Q|!\alpha.F'$, from which we deduce $!F|!\alpha.F' \sim\ !\alpha.F'|!Q \sim\ !\alpha.F'|!Q|\alpha.F' \sim\ !F|\alpha.F'$. (Note that, when writing $!Q$, we actually refer to the process obtained by adding replication at top-level on the finite components of Q; we easily show that this operation preserves bisimilarity.) □

Lemma 5 *If $F = \prod_i \alpha_i.F_i$ and $!F \sim\ !\alpha.F'|Q$, then there exists j s.t. $!F \sim\ !\alpha.F' \mid \prod_{i\neq j} !\alpha_i.F_i$ and $\alpha_j = \alpha$.*

Proof. By firing $\alpha.F'$ on the right-hand side, we find j such that $\alpha_j = \alpha$ and $!F|F_j \sim\ !\alpha.F'|F'|Q$, from which we deduce $!F|F_j \sim\ !F|F'$. Then we show that the singleton relation $\{(!F, !\alpha.F' | \prod_{i\neq j} !\alpha_i.F_i)\}$ is a bisimulation up to bisimilarity and parallel contexts.

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

Now we define our notion of normal forms (*seeds*).

Definition 6 (Size, seed) *The size of P, noted $\sharp P$, is the number of prefixes in P.*
A seed *of P, noted $\mathsf{seed}(P)$ is a process of minimal size such that $P \sim \mathsf{seed}(P)$.*

The seed of a process is defined modulo bisimilarity. We establish in this section that all seeds of a process are actually equated by $\equiv$ (Prop. 14). Note that, because $\sim$ is a congruence in our calculus, if $P_1|P_2$ is a seed, then P_1 is a seed. Indeed, if $\sharp P_1' < \sharp P_1$ and $P_1' \sim P_1$, then $P_1|P_2 \sim P_1'|P_2$, which contradicts the fact that $P_1|P_2$ is a seed.

Notations.

We shall use S, S' to range over seeds having only replicated components. We write $P \rightarrow^k Q$ whenever there exist $\alpha_1, .., \alpha_k$ and $P_0, .., P_k$ such that $P = P_0 \xrightarrow{\alpha_1} P_1 \ldots \xrightarrow{\alpha_k} P_k \equiv Q$. Note that $P \rightarrow^k \alpha.F$ for some k if and only if $P \equiv D[\alpha.F]$ for some finite context D. For $S = \prod_i !\alpha_i.S_i$, we write $S\#F$ to denote the fact that $\neg(\exists i, k,\ F \rightarrow^k \alpha_i.S_i)$, i.e., that F does not contain a sub-term of the form $\alpha_i.S_i$. On the contrary, we write $S \rightsquigarrow F$ when there exists $k > 0$ such that $S \rightarrow^k S|F$, that is, when F is a parallel composition of sub-terms of the S_is. In the sequel, we shall use R to range over finite processes satisfying the latter property.

We can remark that if $S\#F$ (resp. $S \rightsquigarrow F$) and $F \xrightarrow{\alpha} F'$, then $S\#F'$ (resp. $S \rightsquigarrow F'$).

Lemma 7 *(i) If $S|F$ is a seed, then $S\#F$; (ii) if $S \rightsquigarrow R$, then $S\#R$.*

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

(ii) Again, by contradiction, suppose that $R \equiv D[\alpha_i.S_i]$. Since, $S \rightsquigarrow R$, there exist j, D' such that $S_j \equiv D'[D[\alpha_i.S_i]]$, from which we deduce $S \sim \prod_{k\neq j} !\alpha_k.S_k \mid !a_j.D'[D[\mathbf{0}]]$ by (A) (we necessarily have $i \neq j$). This is contradictory with the fact that S is a seed. □

Lemma 8 *$S \sim S|R$ and $S \rightsquigarrow R$ entail $R = \mathbf{0}$.*

Proof. Suppose by contradiction $R = \alpha.R'|R"$. By Lemma 4, we have $S \sim S|\alpha.R'$ and $S \sim S|!\alpha.R'$ by replicating all processes. By Lemma 5 there exists i such that $S \sim !\alpha.R'|\prod_{k\neq i} \alpha_k.S_k$. Now, since $S \rightsquigarrow R$, there exist some j, D such that $S_j \equiv D[\alpha.R']$; 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. □

Lemma 9 *$!F_1|F_1' \sim !F_2|F_2'$ entails $!F_1 \sim !F_2$.*

Proof. Write S_i for the seed of $!F_i$, $i = 1, 2$. We have $S_1|F_1' \sim S_2|F_2'$. By emptying F_1' on the left[1], we obtain $S_1 \sim S_2|F"_2|R_2$ for some $F"_2, R_2$. Now, by emptying on the right, we get $S_1|R_1 \sim !S_2'$. Injecting the latter equivalence in the one we have previously obtained gives

$$S_1 \sim S_1|R_1|F"_2|R_2 \ .$$

If $R_1 \not\sim \mathbf{0}$, we can apply Lemma 4 to deduce $S_1 \sim S_1|R_1$. But this gives a contradiction by Lemma 8. Hence $R_1 \sim \mathbf{0}$, which gives us, since we have established $S_1|R_1 \sim S_2$, that $S_1 \sim S_2$. Finally, $!F_1 \sim !F_2$. □

Lemma 10 *If $S|F \sim S|R$, $S\#F$, and $S \rightsquigarrow R$, then $F \sim R$.*

Proof. We proceed by induction on the size of F. If $F = \mathbf{0}$, we have $R = \mathbf{0}$ by Lemma 8; otherwise, we first prove that F and R have the same size:

- if $\sharp F < \sharp R$, by emptying F on the left-hand side, we find $R' \neq \mathbf{0}$ such that $S|R \rightarrow^{\sharp F} S|R'$, $S \rightsquigarrow R'$ and $S \sim S|R'$; this is contradictory with Lemma 8;
- if $\sharp F > \sharp R$, by emptying R on the right-hand side, we find R', F' with $0 < \sharp F' \leq \sharp F$ such that $S|F \rightarrow^{\sharp R} S|R'|F'$, $S \rightsquigarrow R'$, $S\#F'$ and $S|R'|F' \sim S$. Then we write $F' = \alpha.F_0|F_1$ and deduce $S|\alpha.F_0 \sim S$ by Lemma 4; then, by firing the α prefix, we find i such that $\alpha = \alpha_i$ and $S|F_0 \sim S|S_i$. We check that $\sharp F_0 < \sharp F$ so that we can apply the induction hypothesis and deduce that $F_0 \sim S_i$, whence $\alpha.F_0 \sim \alpha_i.S_i$, and $\alpha.F_0 \equiv \alpha.S_i$ by Thm. 2. This is contradictory with $S\#F$ ($\alpha.F_0$ is a sub-term of F).

[1] In the present case, 'emptying F_1'' means playing all prefixes in F_1' in the bisimulation game between $S_1|F_1'$ and $S_2|F_2'$ – we shall reuse this terminology in some proofs below.

This concludes the proof that F and R have the same size. We then show that the relation $\{(F,R)\}\cup\sim$ is a bisimulation:

- when $F \xrightarrow{\alpha} F'$, we find R' such that $S|R \xrightarrow{\alpha} S|R'$ and $S|F' \sim S|R'$; by induction, $F' \sim R'$, and we deduce that R' is a derivative of R, since otherwise, we would have $\sharp R' \geq \sharp R = 1 + \sharp F'$ which is impossible.
- when $R \xrightarrow{\alpha} R'$, either we find F' such that $F \xrightarrow{\alpha} F'$ and $S|F' \sim S|R'$, which allows us to close the diagram, by induction; or we find i such that $S|S_i|F \sim S|R'$. In this case, we empty R' on the right-hand side, yielding R'' and $F' \neq \mathbf{0}$ such that $S|R''|F' \sim S$; by Lemma 4, $S|F' \sim S$, and $F' \sim \mathbf{0}$ by induction, which is contradictory. □

Lemma 11 *If $S|F_1 \sim S|F_2$ and $S\#F_i$ $(i = 1, 2)$, then $F_1 \sim F_2$.*

Proof. First observe that if $\sharp F_1 < \sharp F_2$, then we can empty F_1 by playing challenges on the left hand side, and we obtain $S \sim S|F_2'$ with $F_2' \not\sim \mathbf{0}$, which is impossible by Lemma 10. Hence $\sharp F_1 = \sharp F_2$.

We then show that $\mathcal{R}= \{(F_1, F_2) \ / \ S|F_1 \sim S|F_2\}$ is a bisimulation. If $F_1 \xrightarrow{\mu} F_1'$, then $S|F_1 \xrightarrow{\mu} S|F_1'$, which by hypothesis entails that $S|F_2$ can answer this challenge. By the remark above, $S|F_2$ necessarily answers by firing F_2, since otherwise we would get equivalent processes with finite parts having different sizes. This allows us to show that F_2 can answer the challenge, and that $\mathcal{R}$ is a bisimulation. □

Lemma 12 *$S|R_1 \sim S|R_2$ and $S \rightsquigarrow R_i$ $(i = 1, 2)$ entail $R_1 \equiv R_2$.*

Proof. By Lemma 11, we have $R_1 \sim R_2$ ($S\#R_i$ by Lemma 7(ii)). We conclude with Thm. 2: R_1 and R_2 are finite processes. □

Lemma 13 *If $S \sim S'$, then $S \equiv S'$.*

Proof. Write $S = \prod_{i\leq m} !\alpha_i.S_i$ and $S' = \prod_{j\leq n} !\alpha_j'.S_j'$, play each prefix on the left-hand side and apply Lemma 12 to show that there exists a map $\sigma : [1..n] \to [1..m]$, such that $\alpha_i.S_i \equiv \alpha'_{\sigma(i)}.S'_{\sigma(i)}$. This map is bijective: otherwise we could construct a smaller seed. □

Proposition 14 (Uniqueness of seeds) *Suppose $P \sim P'$, where P and P' are seeds. Then $P \equiv P'$.*

Proof. Write $P \equiv S|F$ and $P' \equiv S'|F'$. As remarked above, S and S' are necessarily seeds because P and P' are (hence the notation). By Lemma 9, $S \sim S'$, whence $S \equiv S'$ by Lemma 13. Necessarily, $S\#F$ and $S'\#F'$, which allows us to deduce, using Lemma 11, that $F \sim F'$. Finally, $P \equiv P'$, by Thm. 2. □

4 Rewriting Processes to Normal Forms

Definition 15 (Rewriting, convertibility) *Any process P induces a relation between processes, written $\xrightarrow{P}$, defined by the following axioms, modulo $\equiv$:*

$$C[\alpha.F] \xrightarrow{!\alpha.F|F'} C[\mathbf{0}] \quad (B1) \qquad\qquad !\alpha.F|!\alpha.F|P \xrightarrow{Q} !\alpha.F|P \quad (B2)$$

The reflexive transitive closure of $\xrightarrow{P}$ is written $\xrightarrow{P}{}^*$; we say that P and Q are convertible*, written $P \leftrightarrows Q$, whenever there exists a process T such that $P \xrightarrow{T}{}^* T$ and $Q \xrightarrow{T}{}^* T$.*

Example: we can check that process $!\alpha.(\beta|\alpha.\beta)$ is normalised into $!\alpha.\beta$ via the sequence $!\alpha.(\beta|\alpha.\beta) \xrightarrow{!\alpha.\beta|\mathbf{0}} !\alpha.\beta$ using axiom (B1). This is the way our rewriting relation proceeds to compute normal forms. In this case, an equational reasoning would be possible, as follows: $!\alpha.(\beta|\alpha.\beta) = !(\alpha.\beta|\alpha.\beta) = !\alpha.\beta|!\alpha.\beta = !\alpha.\beta$ (we use the law (A') for the first step).

Lemma 16 *If $Q \xrightarrow{T}{}^* T$, then $Q \sim T$.*

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

- (B1): this means that $Q = C[\alpha.P]$, $Q' = C[\mathbf{0}]$ and $T = !\alpha.P|P'$. From $!\alpha.P|P' \sim C[\mathbf{0}]$, we deduce $!\alpha.P|P' \sim C[\alpha.P]$ by Prop. 3, hence $Q \sim T$.
- (B2): we easily have $Q \sim Q'$, hence $Q \sim T$. □

Lemma 17 *Given P, the relation $\xrightarrow{P}{}^{*}$ terminates.*

Proof. The size of processes strictly decreases along reductions. □

Lemma 18 *For all P, either $P \equiv \mathsf{seed}(P)$, or $P \xrightarrow{\mathsf{seed}(P)} P'$ for some P' s.t. $P \sim P'$.*

Proof. Write

$$P = (\prod_i !\alpha_i.F_i) \mid F^P \qquad \text{and} \qquad \mathsf{seed}(P) = (\prod_j !\alpha_j.S_j) \mid F^S \ ,$$

and set $S = \prod_j !\alpha_j.S_j$. By definition, $P \sim \mathsf{seed}(P)$, which gives, by Lemma 9,

$$\prod_i !\alpha_i.F_i \sim \prod_j !\alpha_j.S_j \ . \tag{1}$$

A transition by the left hand side process is answered by the right hand side process, yielding process $\prod_i !\alpha_i.F_i \mid F_n \ \sim \ !\prod_j !\alpha_j.S_j \mid S_m$, which gives, by injecting equivalence (1), $S \mid F_n \sim S \mid S_m$.

By Lemma 11, this gives: either (i) $F_n \sim S_m$, which means by Theorem 2 $F_n \equiv S_m$, or (ii) $\neg(S\#F_n)$ (indeed, $\neg(S\#S_m)$ is impossible, since this would allow us to compute a seed having a smaller size than $\mathsf{seed}(P)$). In the latter case, (ii), this means that P can be rewritten using axiom (B1), and the resulting process is bisimilar to P.

Suppose now that we are in case (i) for all possible transitions from the $\alpha_i.F_i$s, that is, for all i, there exists j such that $\alpha_i.F_i \equiv \alpha_j.S_j$. We observe that the converse (associating a i to all js) also holds, and that the number of parallel components in $\prod_i !\alpha_i.F_i$ is necessarily greater than the number of components in S. In the case where this number is strictly greater, this means that $\xrightarrow{\mathsf{seed}(P)}$ can be used to rewrite the left hand side process in (1), using axiom (B2). In this case, the resulting process is bisimilar to P.

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, we have shown that either $\prod_i !\alpha_i.F_i$ can be rewritten, or $\prod_i !\alpha_i.F_i \equiv S$. In the latter case, we can inject equivalence (1) in $P \sim \mathsf{seed}(P)$, which gives $S \mid F^P \sim S \mid F^S$. We can apply Lemma 11 again, which gives two possibilities. The first possibility is that $F^P \sim F^S$, in which case $F^P \equiv F^S$, and finally $P \equiv \mathsf{seed}(P)$. The second possibility is that $\neg(S\#F^P)$ (as above, $\neg(S\#F^S)$ is not possible since this would allow us to compute a seed of smaller size). In that case, we can rewrite P using (B1), and getting a process bisimilar to P.

Finally, either $P \equiv \mathsf{seed}(P)$, or P can be rewritten using $\xrightarrow{\mathsf{seed}(P)}$. □

Proposition 19 *For all P, $P \xrightarrow{\mathsf{seed}(P)}{}^{*} \mathsf{seed}(P)$.*

Proof. Follows by Lemmas 18 and 17. □

Theorem 20 (Characterisation) *$P \leftrightarrows Q$ iff $P \sim Q$.*

Proof. Suppose $P \leftrightarrows Q$. By definition, this gives the existence of T s.t. $P \xrightarrow{T}{}^{*} T$ and $Q \xrightarrow{T}{}^{*} T$. We deduce $P \sim Q$ by applying Lemma 16 twice and transitivity of $\sim$. Hence $\leftrightarrows \subseteq \sim$.

To establish the converse, suppose $P \sim Q$. Write, using Proposition 19, $P \xrightarrow{\mathsf{seed}(P)}{}^{*} \mathsf{seed}(P)$ and $Q \xrightarrow{\mathsf{seed}(Q)}{}^{*} \mathsf{seed}(Q)$. By definition, $P \sim \mathsf{seed}(P)$ and $Q \sim \mathsf{seed}(Q)$, which entails $\mathsf{seed}(P) \sim \mathsf{seed}(Q)$. This gives by Proposition 14 $\mathsf{seed}(P) \equiv \mathsf{seed}(Q)$, which finally gives $P \leftrightarrows Q$. □

This result gives a way to decide whether $P \sim Q$, via $\leftrightarrows$. Indeed, although Definition 15 does not tell how to find process T, that allows one to derive $P \leftrightarrows Q$, Thm. 20 allows us to reduce this problem to checking whether $Q \xrightarrow{\mathsf{seed}(P)} \mathsf{seed}(P)$. For this, it suffices to look for $\mathsf{seed}(P)$ among all processes of size smaller than $\sharp P$.

5 Adding Synchronisations

We can now move to a calculus closer to standard CCS, called mCCS, by instantiating actions with the following grammar, where a range over a countable set of *names*: actions are either input or output prefixes.

$$\alpha \ ::= \ a \mid \overline{a}$$

The LTS we obtain with this definition is not that of CCS: we need to add the following rules for synchronisations, where τ is the label for internal moves.

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

In doing so, we change the notion of strong bisimilarity: the standard CCS bisimilarity, that we shall denote using $\dot{\sim}$, tests internal moves while our notion of bisimilarity ($\sim$) plays visible challenges only. Therefore, we have $\dot{\sim} \subseteq \sim$.

The following result says that $\leftrightarrows$ is actually enough to capture strong bisimilarity on mCCS. As a consequence, we do not need to test τ transitions to obtain the discriminating power of $\dot{\sim}$.

Proposition 21 *Let P and Q be two processes. Then $P \dot{\sim} Q$ if and only if $P \leftrightarrows Q$.*

Proof. By Thm. 20 and the above remark, it suffices to show that $\leftrightarrows \subseteq \dot{\sim}$. This amounts to check that the distribution law and Prop. 3 are valid for $\dot{\sim}$: we just need to check that silent challenges can be answered in the corresponding bisimulation candidates. □

Note that the τ prefix is not included in this presentation of CCS; indeed, adding τ to the syntax of actions (α) would *a priori* break the inclusion $\dot{\sim} \subseteq \sim$: tests performed by $\sim$ on τ-transitions would be too restrictive, since the only way to answer would be to use a τ prefix – synchronisations would not be allowed. In the light of Prop. 21, we actually believe that τ prefixes could be added, that is, that they are played in one-to-one correspondence in bisimilarity games.

We conclude this section by proving that bisimilarity is closed under substitutions in mCCS. We use σ to range over *substitutions*, that are functions mapping names to names; we write $P\sigma$ for the process we obtain by applying σ on all names of P.

Proposition 22 ($\sim$ is closed under substitutions in mCCS) *If $P \sim Q$, then for any σ, $P\sigma \sim Q\sigma$.*

Proof. We show the property for $\leftrightarrows$. Suppose $P \leftrightarrows Q$, which gives the existence of T such that, in particular, $P \xrightarrow{T}{}^{*} T$. By inspecting the shape of axioms (B1) and (B2), and reasoning by induction over the number of rewrite steps, we can deduce that $P\sigma \xrightarrow{T\sigma}{}^{*} T\sigma$. Similarly, $Q\sigma \xrightarrow{T\sigma}{}^{*} T\sigma$. Hence $P\sigma \leftrightarrows Q\sigma$. □

6 Concluding Remarks

6.1 Extending our Characterisation

In absence of restriction in the calculus, it is easy to see that applying replication to prefixed processes only is of no harm in terms of expressiveness, because of the following rather standard structural congruence laws for $\equiv$ (which are of course valid strong bisimilarity laws):

$$!(P|Q) \equiv !P|!Q \qquad\qquad !!P \equiv !P \qquad\qquad !\mathbf{0} \equiv \mathbf{0}$$

We have started investigating the question of characterising $\sim$ in the case where replication is not at top-level (but where nested replications – that is, replications that occur under replications – are forbidden). The law

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

seems important to capture $\sim$ in this setting. We do not know at the moment whether it is sufficient to characterise $\sim$.

Handling nested replications seems even more challenging.

6.2 Congruence of Strong Bisimilarity in the π-calculus

Because of the input prefix, congruence of strong bisimilarity requires closure of this relation under substitutions. In presence of sum, this property fails; as [3] shows, this is also the case as soon as replication and restriction are present in the calculus (in absence of sum).

[1] shows that congruence holds when we renounce to replication, that is, in the sub-calculus that features input and output prefixes, parallel composition and restriction.

Our investigations have convinced us that the same holds if instead we renounce to restriction: we believe that the reasoning seen above can be ported to the following subset of the π-calculus:

$$F \;::=\; \mathbf{0} \;\big|\; F|F \;\big|\; a(x).F \;\big|\; \overline{a}\langle b\rangle.F \qquad\qquad P \;::=\; F \;\big|\; !a(x).F \;\big|\; P|P$$

The analogue of Prop. 22 gives us closure under substitutions of strong bisimilarity, which in turn yields congruence. Note that when bisimilarity is closed under substitutions, the ground, early and late versions of the equivalence coincide. To adapt our method from mCCS to the π-calculus, we work with ground bisimilarity.

References

1. D. Hirschkoff and D. Pous. A Distribution Law for CCS and a New Congruence Result for the Pi-calculus. *LMCS*, 4(2), 2008.
2. D. Pous. *Techniques modulo pour les bisimulations*. PhD thesis, ENS Lyon, 2008.
3. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.