A Distribution Law for CCS and a New Congruence Result for the pi-calculus

Daniel Hirschkoff, Damien Pous

A Distribution Law for CCS and a New Congruence Result for the π-calculus [⋆]

Daniel Hirschkoff and Damien Pous

LIP – ENS Lyon, CNRS, INRIA, UCBL, France

Abstract. We give an axiomatisation of strong bisimilarity on a small fragment of CCS that does not feature the sum operator. This axiomatisation is then used to derive congruence of strong bisimilarity in the finite π-calculus in absence of sum. To our knowledge, this is the only nontrivial subcalculus of the π-calculus that includes the full output prefix and for which strong bisimilarity is a congruence.

Introduction

In this paper, we study strong bisimilarity on two process calculi. We first focus on *microCCS* (μCCS), the very restricted fragment of CCS that only features prefix and parallel composition. Our main result on μCCS is that adding the following *distribution law*

$$\eta.(P \,|\, \eta.P \,|\, \ldots \,|\, \eta.P) \quad = \quad \eta.P \,|\, \eta.P \,|\, \ldots \,|\, \eta.P$$

to the laws of an abelian monoid for parallel composition yields a complete axiomatisation of strong bisimilarity (in the law above, η is a CCS prefix, of the form a or $\overline{a}$, and P is any CCS process – the same number of copies of P appear on both sides of the equation).

The distribution law is not new: it is mentioned – among other *'mixed equations'* relating prefixed terms and parallel compositions – in a study of bisimilarity on normed PA processes [8]. In our setting, this equality can be oriented from left to right to rewrite processes into normal forms, which intuitively exhibit as much concurrency as possible. Strong bisimilarity ($\sim$) between processes is then equivalent to equality of their normal forms. This rewriting phase allows us to actually compute *unique decompositions* of processes into *prime processes*, in the sense of [10]: a process P is prime if P is not bisimilar to the inactive process $\mathbf{0}$ and if $P \sim Q \,|\, R$ implies $Q \sim \mathbf{0}$ or $R \sim \mathbf{0}$.

The distribution law is an equational schema, corresponding to an infinite family of axioms, of the form $\eta.(P \,|\, (\eta.P)^k) = (\eta.P)^{k+1}$, for $k \geq 1$ (where Q^k denotes the k-fold parallel composition of process Q). We show that although our setting is rather simple, there exists no finite axiomatisation of $\sim$ on μCCS.

[⋆] Author's version of the paper published by Springer in Proc. FoSSaCS'07, available at `http://dx.doi.org/10.1007/978-3-540-71389-0_17`.

We then move to the study of strong bisimilarity in the π-calculus. Because of the presence of the input prefix, and of the related phenomenon of name-passing, bisimilarity is more complex in the π-calculus than in CCS. In particular, both early and late bisimilarity, that differ in their treatment of name substitution, fail to be congruences in the full π-calculus.

There exist subcalculi of the π-calculus for which strong bisimilarity is a congruence (we discuss these in Section 5). When this is the case, this equivalence coincides with *ground bisimilarity* ($\sim_{\mathrm{g}}$), which allows one to consider only one fresh name when inspecting an input transition, instead of the usual quantification involving all free names of the process. Congruence of strong bisimilarity is hence an important property: not only is it necessary in order to reason in a compositional way, but it also helps making bisimulation proofs simpler, by reducing the size of case analyses.

In the full π-calculus, in order to get congruence, one has to work with Sangiorgi's open bisimilarity [12], which has a more involved definition than the early and late variants. Tools like the Mobility Workbench [14], for instance, have adopted this equivalence on processes.

It is known [13] that bisimilarity in the π-calculus fails to be a congruence as soon as we have prefix, parallel composition, restriction and replication. In this work, we focus on the finite, sum-free π-calculus, that we call π_0. We rely on the axiomatisation of strong bisimilarity on μCCS to prove that ground bisimilarity ($\sim_{\mathrm{g}}$) is closed under substitutions in π_0, i.e., that whenever $P \sim_{\mathrm{g}} Q$, then $P\sigma \sim_{\mathrm{g}} Q\sigma$ for any substitution σ. Closure under substitution of ground bisimilarity entails that on π_0, ground, early, late and open bisimilarities coincide, and are congruences. The problem of congruence of $\sim_{\mathrm{g}}$ on π_0 is mentioned as an open question in [13, Chapter 5], and is known since at least 1998 [2]. To our knowledge, this is the first congruence result for a subcalculus of the π-calculus that includes the full output prefix (see Section 5 for a discussion on this).

At the heart of our proof of congruence is a notion that we call *mutual desynchronisation*, and that corresponds to the existence of processes T, T_{12}, T_{21} such that $T \xrightarrow{\eta_1}\xrightarrow{\eta_2} T_{12}$ and $T \xrightarrow{\eta_2}\xrightarrow{\eta_1} T_{21}$, for two distinct actions η_1 and η_2, and with $T_{12} \sim T_{21}$. We additionally require in the two sequences of transitions from T to T_{12} and T_{21} respectively that the second prefix being fired should occur under the first prefix in T. In other words, in such a situation, the process T behaves as if the two actions η_1, η_2 were offered concurrently, but the simultaneous firing of these actions can only be emulated by triggering consecutive prefixes.

Using our analysis of strong bisimilarity on μCCS, we show that mutual desynchronisations do not exist in μCCS. This is essentially due to the fact that our axiomatisation of $\sim$ on μCCS does not allow one to relate two *distinct* prefixes when performed concurrently and sequentially. When moving to the π-calculus, it turns out that substitution closure of $\sim_{\mathrm{g}}$ amounts to observing absence of mutual desynchronisations in π_0. We exploit a transfer property, that extracts a bisimilarity proof in μCCS from a bisimilarity proof in π_0, to relate the two calculi and to show that mutual desynchronisations do not exist in π_0, yielding congruence of $\sim_{\mathrm{g}}$.

Paper outline. We introduce μCCS and the distribution law in Section 1. Section 2 is devoted to the characterisation of $\sim$ on μCCS using normal forms. In Section 3, we prove that no finite axiomatisation of $\sim$ on μCCS exists. Section 4 presents the proof of our congruence result in the π-calculus, and we give concluding remarks in Section 5.

1 MicroCCS Processes and Normal Forms

We consider an infinite set $\mathcal{N}$ of names, ranged over with $a, b \ldots$. We define on top of $\mathcal{N}$ the set of processes of μCCS, the finite, public, sum-free CCS calculus, ranged over using $P, Q, R \ldots$, as follows:

$$\eta ::= a \;\big|\; \overline{a} \;, \qquad\qquad P ::= \mathbf{0} \;\big|\; \eta.P \;\big|\; P_1 \,|\, P_2 \;.$$

$\mathbf{0}$ is the nil process. η ranges over visible actions and co-actions, called *interactions*, and we let $\overline{\eta}$ stand for the co-action associated to η (we have $\overline{\overline{\eta}} = \eta$). For $k > 0$, we write P^k for the parallel composition of k copies of P, and we write $\prod_{i \in I} P_i$ for the parallel composition of all processes P_i for $i \in I$. It can be noted that our syntax does not include a construction of the form $\tau.P$ — see Remark 2.3 below.

Structural congruence, written $\equiv$, is defined as the smallest congruence satisfying the following laws:

$$(C_1) \quad P \,|\, Q \equiv Q \,|\, P \qquad (C_2) \quad P \,|\, (Q \,|\, R) \equiv (P \,|\, Q) \,|\, R \qquad (C_3) \quad P \,|\, \mathbf{0} \equiv P$$

We introduce a labelled transition system (LTS) for μCCS. Actions labelling transitions, ranged over with μ, are either interactions, or a special silent action, written τ.

Definition 1.1 (Operational semantics and behavioural equivalence). *The LTS for μCCS is given by the following rules:*

$$\eta.P \xrightarrow{\eta} P \qquad \frac{P \xrightarrow{\eta} P' \qquad Q \xrightarrow{\overline{\eta}} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'} \qquad \frac{P \xrightarrow{\mu} P'}{P \,|\, Q \xrightarrow{\mu} P' \,|\, Q} \qquad \frac{P \xrightarrow{\mu} P'}{Q \,|\, P \xrightarrow{\mu} Q \,|\, P'}$$

A bisimulation *is a symmetrical relation $\mathcal{R}$ between processes such that whenever $P \mathrel{\mathcal{R}} Q$ and $P \xrightarrow{\mu} P'$, there exists Q' such that $Q \xrightarrow{\mu} Q'$ and $P' \mathrel{\mathcal{R}} Q'$.*
Bisimilarity*, written $\sim$, is the union of all bisimulations.*

Definition 1.2 (Size). *Given P, $\#(P)$ (called the* size *of P) is defined by:*

$$\#(\mathbf{0}) \overset{\text{def}}{=} 0 \qquad \#(P_1 \,|\, P_2) \overset{\text{def}}{=} \#(P_1) + \#(P_2) \qquad \#(\eta.P) \overset{\text{def}}{=} 1 + \#(P) \;.$$

Lemma 1.3. *$P \equiv Q$ implies $P \sim Q$ which in turn implies $\#(P) = \#(Q)$.*

Proof. The first implication follows by proving that $\equiv$ is a bisimulation.

Suppose then by contradiction that there exist P, Q such that $P \sim Q$ and $\#(P) < \#(Q)$; and choose such P with minimal size. Q has at least one prefix: $Q \xrightarrow{\eta} Q'$ and we get $P \xrightarrow{\eta} P'$ with $P' \sim Q'$. Necessarily, we must have $\#(P') < \#(Q')$ and $\#(P') < \#(P)$ which contradicts the minimality hypothesis. □

Definition 1.4 (Distribution law). *The* distribution law *is given by the following equation, where the same number of copies of P appears on both sides:*

$$\eta.(P \,|\, \eta.P \,|\, \ldots \,|\, \eta.P) \quad = \quad \eta.P \,|\, \eta.P \,|\, \ldots \,|\, \eta.P \ .$$

We shall use this equality, oriented from left to right, to rewrite processes. We write $P \rightsquigarrow P'$ when there exist P_1, P_2 such that $P \equiv P_1$, $P_2 \equiv P'$ and P_2 is obtained from P_1 by replacing a sub-term of the form of the left-hand side process with the right-hand side process.

Remark 1.1 (On the distribution law and PA). Among the studies about properties of $\sim$ in process algebras that include parallel composition (see [1] for a recent survey on axiomatisations), some works focus on calculi where parallel composition is treated as a primitive operator (as opposed to being expressible using sum or other constructs like the left merge operator). As mentioned above, particularly relevant to this work is [8], where Hirshfeld and Jerrum *"develop a structure theory for PA that completely classifies the situations in which a sequential composition of two processes can be bisimilar to a parallel composition"*. [8] establishes decidability of $\sim$ for normed PA processes: in that setting, the formal analogue of the distribution law (Def. 1.4) holds with η and P being two processes — the 'dot' operator is a general form of sequential composition. This equality is valid in [8] whenever η is a 'monomorphic process', meaning that η can only reduce to $\mathbf{0}$ (which corresponds to μCCS), or to η itself. [6] presents a finite axiomatisation of PA that exploits the operators of sum and left merge.

Lemma 1.5. *The relation $\rightsquigarrow$ is strongly normalising and confluent.*

Proof. If $P \rightsquigarrow P'$ then the weight of P' (defined as sum of the depths of all prefixes occuring in P') is strictly smaller than the weight of P, whence the strong normalisation. We then remark that $\rightsquigarrow$ is locally confluent, and conclude with Newman's Lemma. □

Thus, for any process P, $\rightsquigarrow$ defines a normal form unique up to $\equiv$, that will be denoted by $\mathsf{n}(P)$. We let $A, B, \ldots$ range over normal forms.

The following lemma states that $\rightsquigarrow$ preserves bisimilarity:

Lemma 1.6. *If $P \rightsquigarrow P'$, then $P \sim P'$. For any P, $P \sim \mathsf{n}(P)$.*

Proof. The relation $(\rightsquigarrow \cup (\rightsquigarrow)^{-1} \cup \equiv)$ is a bisimulation. □

2 Characterisation of Bisimilarity in MicroCCS

Our characterisation of $\sim$ on μCCS makes use of the notion of decomposition into *prime processes*, defined as follows:

Definition 2.1. *A process P is* prime *if $P \not\sim \mathbf{0}$ and $P \sim P_1 \,|\, P_2$ implies $P_1 \sim \mathbf{0}$ or $P_2 \sim \mathbf{0}$.*

When $P \sim P_1 \,|\, \dots \,|\, P_n$ where the P_is are prime, we shall call $P_1 \,|\, \dots \,|\, P_n$ a prime decomposition *of P.*

Proposition 2.2 (Unique decomposition). *Any process admits a prime decomposition which is unique up to bisimilarity: if $P_1 \,|\, \dots \,|\, P_n$ and $Q_1 \,|\, \dots \,|\, Q_m$ are two prime decompositions of the same process, then $n = m$ and $P_i \sim Q_i$ for all $i \in [1..n]$, up to a permutation of the indices.*

Proof. Similar to the proof of [11, Theorem 4.3.1]: the case of μCCS is not explicitly treated in that work, but the proof can be adapted rather easily. □

An immediate consequence of the above result is the following property:

Corollary 2.3 (Cancellation). *For all P, Q, R, $P \,|\, R \sim Q \,|\, R$ implies $P \sim Q$.*

Note that this is not true in presence of replication: $a \,|\, !a \sim \mathbf{0} \,|\, !a$, but $a \not\sim \mathbf{0}$.

The characterisation of $\sim$ using the distribution law follows from the observation that if a normal form is a prefixed process, then it is prime. This idea is used in the proof of Lemma 2.5. We first establish a technical result, that essentially exploits the same argument as the proof of Theorem 4.2 in [7].

Lemma 2.4. *If $\eta.P \sim Q \,|\, Q'$, with $Q, Q' \not\sim \mathbf{0}$, then there exist A and $k > 1$ such that $\eta.P \sim (\eta.A)^k$ and $\eta.A$ is a normal form.*

Proof. By Lemma 1.6, we have $\eta.P \sim \mathsf{n}(Q \,|\, Q')$. Furthermore, we have that $\mathsf{n}(Q \,|\, Q') \equiv \prod_{i \leq k} \eta_i.A_i$, where $k > 1$ and the processes $\eta_i.A_i$ are in normal form.

Since the η prefix must be triggered to answer any challenge from the right hand side, we have $\eta_i = \eta$ and $P \sim A_i \,|\, \prod_{l \neq i} \eta.A_l$ for all $i \leq k$. In particular, when $i \neq j$, we have $P \sim A_i \,|\, \eta.A_j \,|\, \prod_{l \notin \{i,j\}} \eta.A_l \sim \eta.A_i \,|\, A_j \,|\, \prod_{l \notin \{i,j\}} \eta.A_l$ and hence, by Corollary 2.3, $A_i \,|\, \eta.A_j \sim \eta.A_i \,|\, A_j$. By reasoning on the sizes of the parallel components in the prime decompositions of these two terms, we conclude that $\eta.A_i \sim \eta.A_j$ for all $i, j \leq k$.

Hence, we have $\eta.P \sim (\eta.A_1)^k$ with $k > 1$ and $\eta.A_1$ is a normal form. □

Lemma 2.5. *Let A, B be two normal forms, $A \sim B$ implies $A \equiv B$.*

Proof. We show by induction on n that for all A with $\#(A) = n$, we have

(i) if A is a prefixed process, then A is prime;
(ii) for any B, $A \sim B$ implies $A \equiv B$.

The case $n = 0$ is immediate. Suppose that the property holds for all $i < n$, with $n \geq 1$.

(i) We write $A = \eta.A'$, and suppose by contradiction $A \sim P_1 \,|\, P_2$ with $P_1, P_2 \not\sim \mathbf{0}$. By Lemma 2.4, we have $A \sim (\eta.B)^k$ with $k > 1$ and $\eta.B$ in normal form. By triggering the prefix on the left hand side, we have $A' \sim B \,|\, (\eta.B)^{k-1}$. It follows by induction that $A' \equiv B \,|\, (\eta.B)^{k-1}$ (using property (ii)), and hence $A \equiv \eta.(B \,|\, (\eta.B)^{k-1}$, which is in contradiction with the fact that A is in normal form.

(ii) Suppose now $A \sim B$.

- If A is a prefixed process, B is prime by the previous point ($\#(B) = \#(A)$ by Lemma 1.3). Necessarily, $A \equiv \eta.A'$ and $B \equiv \eta.B'$ with $A' \sim B'$. By induction, this entails $A' \equiv B'$, and $A \equiv B$.
- Otherwise, $A = \eta_1.A_1 \,|\, \ldots \,|\, \eta_k.A_k$ with $k > 1$, and we know by induction (property (i)) that $\eta_i.A_i$ is prime for all $i \leq k$. Similarly, we have $B = \eta'_1.B_1 \,|\, \ldots \,|\, \eta'_l.B_l$ with $\eta'_i.B_i$ prime for all $i \leq l$.
  By Proposition 2.2, $k = m$ and $\eta_i.A_i \sim \eta'_i.B_i$ (up to a permutation of the indices), which gives $\eta'_i = \eta_i$ and $A_i \sim B_i$ for all $i \leq k$. By induction, we deduce $A_i \equiv B_i$ for all i, which finally implies $A \equiv B$. □

Lemmas 1.6 and 2.5 allow us to deduce the following result.

Theorem 2.6. *Let P, Q be two μCCS processes. Then $P \sim Q$ iff $\mathsf{n}(P) \equiv \mathsf{n}(Q)$.*

Remark 2.1 (Unique decomposition of processes). Our proof relies on unique decomposition of processes (Prop. 2.2), that first appeared in [10]. Unique decomposition has been established for a variety of process algebras, and used as a way to prove decidability of behavioural equivalence and to give complexity bounds for the associated decision procedure ([9, 3] cite relevant references).

In the present study, beyond the existence of a unique decomposition, we are interested in a syntactic characterisation of $\sim$ (which will in particular allow us to derive Lemma 4.6 below). In this sense, our work is close to [5], where the notion of *maximally parallel process* in CCS (with choice) is studied. [5] defines a rewriting process through which maximally parallel normal forms can be computed, and shows that in the case of μCCS, such normal forms are unique. However, no syntactical characterisation of the set of normal forms is presented, and such a characterisation cannot be directly deduced from the (rather involved) definition of the rewriting process for full CCS.

We instead restrict ourselves to μCCS from the start, and rely explicitly on the distribution law in order to 'extract' prime components of processes.

Remark 2.2 (Closure under substitutions). In (full) CCS, two strongly bisimilar processes need not remain bisimilar whenever we apply a substitution that replaces names with names. The standard counterexample is given by $a.\overline{b} + \overline{b}.a \sim a \,|\, \overline{b}$: when we replace b with a, we obtain two processes that are distinguished by $\sim$, since the latter can perform a τ transition that cannot be matched by the former. This irregularity is the basis of the standard counterexample showing that strong bisimilarity is not a congruence in the π-calculus.

In μCCS, on the other hand, $\sim$ is closed under substitutions: the intuitive reason is that two processes related by an instance of the distribution law remain equivalent when a substitution is applied (we can show in particular that for any substitution σ, $\mathsf{n}(P\sigma) \equiv \mathsf{n}(\mathsf{n}(P)\sigma)$). This is not the case for the expansion law, of which the counterexample above is an instance.

Remark 2.3 (τ prefix and weak bisimilarity). We do not address weak bisimilarity in the present work. In μCCS, strong and weak bisimilarity coincide, i.e., the internal transitions of processes are completely determined by the visible actions (interactions). When including τ prefixes in the syntax, it can be proved that adding the law $\tau.P = P$ is enough to characterise weak bisimilarity. The τ prefix is usually absent in the π-calculus, to which we shall move in Section 4. Since some results on CCS will be transferred to the π-calculus, we did not include this construct in μCCS.

3 Nonexistence of a Finite Axiomatisation

We let $\mathcal{D}$ stand for the set of equations consisting of the three axioms of structural congruence (C_1, C_2, C_3), and the infinite family of *distribution axioms*

$$(D_k): \ \eta.(P \,|\, (\eta.P)^k) \ = \ (\eta.P)^{k+1}, \ k \geq 1 \ .$$

We let $\mathcal{D}_k$ stand for the finite restriction of $\mathcal{D}$ where only the first k distribution axioms are included ($(D_i)_{1\leq i\leq k}$). We shall write $\mathcal{E} \vdash P = Q$ whenever $P = Q$ can be derived in equational logic using a given set $\mathcal{E}$ of axioms, and $\mathcal{E} \not\vdash P = Q$ when this is not the case.

$(D_k)_{k\geq 1}$ forms an equational schema for the distribution law, and Theorem 2.6 states that $\mathcal{D}$ is a complete axiomatisation of strong bisimilarity on μCCS. Using a rather classical approach (i.e., establishing ω-completeness and proving compactness, see [1]), this leads to the nonexistence of a finite axiomatisation of $\sim$ on μCCS. The lemma below provides the central technical property satisfied by the $(\mathcal{D}_k)_{k\geq 1}$ which is necessary to derive Theorem 3.2, that says that $\mathcal{D}$ is *intrinsically* infinite.

Lemma 3.1. *Let a be a name. For any k, there exists n s.t. $\mathcal{D}_k \not\vdash a.a^n = a^{n+1}$.*

Remember that a^n stands for the n-fold parallel composition of $a.\mathbf{0}$, so that the above equality is an instance of axiom (D_n).

Proof. Let n be a number strictly greater than k such that $n+1$ is prime, and let $\theta(P, Q)$ denote the predicate: "$P \sim Q \sim a^{n+1}$, $P \equiv a.P'$, and $Q \equiv Q_1 \,|\, Q_2$ with $Q_1, Q_2 \not\equiv \mathbf{0}$". Suppose now that $\mathcal{D}_k \vdash a.a^n = a^{n+1}$, and consider the shortest proof of $\mathcal{D}_k \vdash P = Q$ for some processes P, Q such that either $\theta(P, Q)$ or $\theta(Q, P)$. Since $\theta(a.a^n, a^{n+1})$ holds, such a minimal proof does exist. We reason about the last rule used in the derivation of this proof in equational logic. For syntactic reasons, this cannot be reflexivity, a contextual rule, nor one of the structural congruence axioms. It can be neither symmetry nor transitivity, since otherwise this would give a shorter proof satisfying θ. The only possibility is thus the use of one of the distribution axioms, say D_i with $1 \leq i \leq k$ and $a^{n+1} \sim Q \equiv (a.Q')^{i+1}$. By Lemma 1.3, since $\#(a^{n+1}) = n+1$, $i+1$ has to divide $n+1$. This is contradictory, because we have $2 \leq i+1 \leq k+1 < n+1$, and $n+1$ is prime. □

Theorem 3.2 (No finite axiomatisation of $\sim$). *For any finite set of axioms $\mathcal{E}$, there exist processes P and Q such that $P \sim Q$ but $\mathcal{E} \not\vdash P = Q$.*

Proof. Standard, by proving that $\mathcal{D}$ is ω-complete and then using the Compactness Theorem (see [1]). □

4 A New Congruence Result for the π-calculus

4.1 The Finite, Sum-free π-calculus

π-calculus processes are built from an infinite set $\mathcal{N}_\pi$ of names, ranged over using $a, b \ldots, m, n \ldots, p, q \ldots, x, y \ldots$, according to the following grammar:

$$\phi ::= m(x) \;\big|\; \overline{m}n \ , \qquad\qquad P ::= \mathbf{0} \;\big|\; \phi.P \;\big|\; P_1 \,|\, P_2 \;\big|\; (\boldsymbol{\nu}p)P \ .$$

The input prefix $m(x)$ binds name x in the continuation process, and so does name restriction $(\boldsymbol{\nu}n)$ in the restricted process. A name that is not bound is said to be free, and we let $\mathrm{fn}(P)$ stand for the free names of P. We assume that any process that we manipulate satisfies a *Barendregt convention*: every bound name is distinct from the other bound and free names of the process. We shall use a, b, c to range over free names of processes, p, q, r (resp. x, y) to range over names bound by restriction (resp. by input), and m, n to range over any name, free or bound (note that these naming conventions are used in the above grammar). Structural congruence on π_0, written $\equiv$, is the smallest congruence that is an equivalence relation, contains α-equivalence, and satisfies the following laws:

$$P \,|\, \mathbf{0} \equiv P \qquad P \,|\, (Q \,|\, R) \equiv (P \,|\, Q) \,|\, R \qquad P \,|\, Q \equiv Q \,|\, P \qquad (\boldsymbol{\nu}p)\mathbf{0} \equiv \mathbf{0}$$

$$(\boldsymbol{\nu}p)(\boldsymbol{\nu}q)P \equiv (\boldsymbol{\nu}q)(\boldsymbol{\nu}p)P \qquad P \,|\, (\boldsymbol{\nu}p)Q \equiv (\boldsymbol{\nu}p)(P \,|\, Q) \quad \text{if } p \notin \mathrm{fn}(P)$$

We let $P[n/x]$ stand for the capture avoiding substitution of name x with name n in P. We use σ to range over substitutions in π_0 (that simultaneously replace several names).

Definition 4.1 (Late operational semantics and ground bisimilarity). *The late operational semantics of π_0 is given by a transition relation whose set of labels is defined by:*

$$\mu ::= a(x) \;\big|\; \overline{a}b \;\big|\; \overline{a}(p) \;\big|\; \tau \ .$$

Names x and p are said to be bound in actions $a(x)$ and $\overline{a}(p)$ respectively, and other names are free. We use $\mathrm{bn}(\mu)$ *(resp.* $\mathrm{fn}(\mu)$*) to denote the set of bound (resp. free) names of action μ.*

The late transition relation, written $\rightarrow_\pi$, is given by the following rules (symmetrical versions of the rules involving parallel composition are omitted):

$$\frac{}{\phi.P \xrightarrow{\phi}_\pi P} \qquad \frac{P \xrightarrow{a(x)}_\pi P' \qquad Q \xrightarrow{\overline{a}b}_\pi Q'}{P\,|\,Q \xrightarrow{\tau}_\pi P'[b/x]\,|\,Q'}$$

$$\frac{P \xrightarrow{\overline{a}b}_\pi P'}{(\boldsymbol{\nu}b)P \xrightarrow{\overline{a}(b)}_\pi P'}\ a \neq b \qquad \frac{P \xrightarrow{a(x)}_\pi P' \qquad Q \xrightarrow{\overline{a}(p)}_\pi Q'}{P\,|\,Q \xrightarrow{\tau}_\pi (\boldsymbol{\nu}p)(P'[p/x]\,|\,Q')}$$

$$\frac{P \xrightarrow{\mu}_\pi P'}{P\,|\,Q \xrightarrow{\mu}_\pi P'\,|\,Q}\ \mathrm{bn}(\mu)\cap\mathrm{fn}(Q)=\emptyset \qquad \frac{P \xrightarrow{\mu}_\pi P'}{(\boldsymbol{\nu}p)P \xrightarrow{\mu}_\pi (\boldsymbol{\nu}p)P'}\ p \notin \mathrm{fn}(\mu)$$

A ground bisimulation *is a symmetric relation $\mathcal{R}$ between processes such that whenever $P \mathrel{\mathcal{R}} Q$ and $P \xrightarrow{\mu}_\pi P'$, there exists Q' s.t. $Q \xrightarrow{\mu}_\pi Q'$ and $P' \mathrel{\mathcal{R}} Q'$.*

Ground bisimilarity*, written $\sim_{\mathrm{g}}$, is the union of all ground bisimulations.*

Note that we do not respect the convention on names in the rule to infer a bound output, precisely because we are transforming a free name (b) into a bound name.

Lemma 4.2. *Suppose that $P\sigma \xrightarrow{\mu}_\pi P'$.*

1. *If μ is $\overline{a}b$, $\overline{a}(p)$ or $a(x)$, then $P \xrightarrow{\mu'}_\pi P''$ with $\mu'\sigma = \mu$ and $P''\sigma = P'$.*
2. *If $\mu = \tau$ then one of the three following properties hold, where the input and output actions are offered concurrently by P in the last two cases.*
   - *(a) $P \xrightarrow{\tau}_\pi P''$ and $P''\sigma = P'$,*
   - *(b) $P \xrightarrow{\overline{b}c}_\pi \xrightarrow{a(x)}_\pi P''$ where $\sigma(a) = \sigma(b)$ and $P''[c/x]\sigma \sim P'$,*
   - *(c) $P \xrightarrow{\overline{b}(p)}_\pi \xrightarrow{a(x)}_\pi P''$ where $\sigma(a) = \sigma(b)$ and $((\boldsymbol{\nu}p)P''[p/x])\sigma \sim P'$.*

Proof. Similar to the proof of Lemma 1.4.13 in [13], where the early transition semantics is treated. □

4.2 Mutual Desynchronisations

We now introduce the notion of mutual desynchronisation in μCCS, which is defined as the existence of processes obeying certain conditions in the calculus. We shall see that because of τ synchronisations, the absence of mutual desynchronisations is related to substitution closure of $\sim$.

Definition 4.3 (Mutual desynchronisation in μCCS). *We say that there exists a* mutual desynchronisation in μCCS *whenever there are two prefixes η_1, η_2, and five μCCS processes P, P', Q, Q', R such that $\eta_1 \neq \eta_2$, $P \xrightarrow{\eta_1} P'$, $Q \xrightarrow{\eta_2} Q'$ and $\eta_2.P \,|\, Q' \,|\, R \sim P' \,|\, \eta_1.Q \,|\, R$.*

The notion of mutual desynchronisation is not specific to μCCS. As explained in the introduction, it corresponds to a situation where three processes T, T_{12}, T_{21} satisfy:

- $T \xrightarrow{\eta_1}\xrightarrow{\eta_2} T_{12}$ and $T \xrightarrow{\eta_2}\xrightarrow{\eta_1} T_{21}$, where the second prefix being triggered occurs under the first one in both sequences of transitions.
- $\eta_1 \neq \eta_2$ and $T_{12} \sim T_{21}$.

The proofs of Lemmas 4.9 and 4.10 will expose analogous situations in π_0.

Definition 4.4. *We define, for any μCCS process P and prefix η, the* contribution of P at η, *written $s_\eta(P)$, by*

$$s_\eta(\mathbf{0}) \overset{\text{def}}{=} 0 \qquad\qquad s_\eta(\eta'.P) \overset{\text{def}}{=} 0 \quad \textit{if } \eta \neq \eta'$$

$$s_\eta(P_1 \,|\, P_2) \overset{\text{def}}{=} s_\eta(P_1) + s_\eta(P_2) \qquad\qquad s_\eta(\eta.P) \overset{\text{def}}{=} \#(\eta.P)$$

Intuitively, $s_\eta(P)$ is the total size of the parallel components of P that start with the prefix η.

Lemma 4.5. *$P \sim Q$ implies $s_\eta(P) = s_\eta(Q)$ for all η.*

Proof. Follows from Theorem 2.6 and the observation that the distribution law preserves the contribution of a process at a given interaction prefix. □

Lemma 4.6 (No mutual desynchronisation). *There exists no mutual desynchronisation in μCCS.*

Proof. Suppose by contradiction that there are processes such that $P \xrightarrow{\eta_1} P'$, $Q \xrightarrow{\eta_2} Q'$ and $\eta_2.P \,|\, Q' \,|\, R \sim P' \,|\, \eta_1.Q \,|\, R$.

By the cancellation property (Corollary 2.3), we have $\eta_2.P \,|\, Q' \sim P' \,|\, \eta_1.Q$, hence for all η, $s_\eta(\eta_2.P \,|\, Q') = s_\eta(P' \,|\, \eta_1.Q)$ (Lemma 4.5).

Since $s_{\eta_1}(\eta_2.P \,|\, Q') = s_{\eta_1}(Q') \leq \#(Q')$ and $s_{\eta_1}(P' \,|\, \eta_1.Q)) \geq s_{\eta_1}(\eta_1.Q) = \#(Q') + 2$, by taking $\eta = \eta_1$ we finally get $\#(Q') \geq \#(Q') + 2$. □

This result will be used to show that a situation corresponding to a mutual desynchronisation cannot arise in π_0. Notice that the proof depends in an essential way on Lemma 4.5, which in turn relies on the axiomatisation of $\sim$ in μCCS (Theorem 2.6).

In what follows, we fix two distinct names a and b, that will occur free in the processes we shall consider. The definitions and results below will depend on a and b, but we avoid making this dependency explicit, in order to ease readability. Names a and b will be fixed in the proof of Lemma 4.11.

Definition 4.7 (Erasing a π_0 process). *Given a π_0 process P, we define the* erasing of P, *written $\mathcal{E}(P)$, as follows:*

$$\begin{array}{ccc} \mathcal{E}(P_1 \,|\, P_2) \overset{\text{def}}{=} \mathcal{E}(P_1) \,|\, \mathcal{E}(P_2) & \mathcal{E}((\boldsymbol{\nu} p)P) \overset{\text{def}}{=} \mathcal{E}(P) & \mathcal{E}(\mathbf{0}) \overset{\text{def}}{=} \mathbf{0} \\ \mathcal{E}(a(x).P) \overset{\text{def}}{=} a.\mathcal{E}(P) & \mathcal{E}(m(x).P) \overset{\text{def}}{=} \mathbf{0} \textit{ if } m \neq a & \\ \mathcal{E}(\overline{b}n.P) \overset{\text{def}}{=} \overline{b}.\mathcal{E}(P) & \mathcal{E}(\overline{m}n.P) \overset{\text{def}}{=} \mathbf{0} \textit{ if } m \neq b & \end{array}$$

Note that a and b play different roles in the definition of $\mathcal{E}(\cdot)$.

It is immediate from the definition that $\mathcal{E}(P)$ is a μCCS process whose only prefixes are a and $\overline{b}$. Intuitively, $\mathcal{E}(P)$ only exhibits the interactions of P at a (in input) and b (in output) that are not guarded by interactions on other names.

Lemma 4.8 (Transitions of $\mathcal{E}(P)$). *Consider a π_0 process P. We have:*

- *If $P \xrightarrow{a(x)}_{\pi} P'$, then $\mathcal{E}(P) \xrightarrow{a} \mathcal{E}(P')$.*
- *If $P \xrightarrow{\overline{b}c}_{\pi} P'$ or $P \xrightarrow{\overline{b}(p)}_{\pi} P'$, then $\mathcal{E}(P) \xrightarrow{\overline{b}} \mathcal{E}(P')$.*
- *Conversely, if $\mathcal{E}(P) \xrightarrow{a} P_0$, then there exist x and P' such that $P_0 = \mathcal{E}(P')$ and $P \xrightarrow{a(x)}_{\pi} P'$. Similarly, if $\mathcal{E}(P) \xrightarrow{\overline{b}} P_0$, there exist c, p, P' such that $P_0 = \mathcal{E}(P')$ and either $P \xrightarrow{\overline{b}c}_{\pi} P'$ or $P \xrightarrow{\overline{b}(p)}_{\pi} P'$.*

Proof. Simple reasoning on the LTSs of μCCS and π_0. □

Proposition 4.1 (Transfer). *If $P \sim_{\text{g}} Q$ in π_0, then $\mathcal{E}(P) \sim \mathcal{E}(Q)$ in μCCS.*

Proof. We reason by induction on the size of P (defined as the number of prefixes in P). Consider a transition of $\mathcal{E}(P)$; as observed above, it can only be a transition along a or a transition along $\overline{b}$.

Suppose $\mathcal{E}(P) \xrightarrow{a} P_0$. By Lemma 4.8, $P \xrightarrow{a(x)}_{\pi} P'$ and $P_0 = \mathcal{E}(P')$. Since $P \sim_{\text{g}} Q$, $Q \xrightarrow{a(x)}_{\pi} Q'$ for some Q' such that $P' \sim_{\text{g}} Q'$. By induction, the latter relation gives $\mathcal{E}(P') \sim \mathcal{E}(Q')$, and $Q \xrightarrow{a(x)}_{\pi} Q'$ gives by Lemma 4.8 $\mathcal{E}(Q) \xrightarrow{a} \mathcal{E}(Q')$.

The case $\mathcal{E}(P) \xrightarrow{\overline{b}} P_0$ is treated similarly: by Lemma 4.8, there are two cases, according to whether P does a free output or a bound output. Reasoning like above allows us to conclude in both cases. □

We can now present our central technical result about π_0, which comes in two lemmas.

Lemma 4.9. *If $Q \sim_{\text{g}} (\boldsymbol{\nu}\widetilde{p})(a(x).P_1 \,|\, \overline{b}c.P_2 \,|\, P_3)$, then there exist some Q_1, Q_2, Q_3, $\widetilde{q}$, such that $Q \equiv (\boldsymbol{\nu}\widetilde{q})(a(x).Q_1 \,|\, \overline{b}c.Q_2 \,|\, Q_3)$ and*

$$(\boldsymbol{\nu}\widetilde{p})(P_1 \,|\, P_2 \,|\, P_3) \;\sim_{\text{g}}\; (\boldsymbol{\nu}\widetilde{q})(Q_1 \,|\, Q_2 \,|\, Q_3).$$

Proof. Let $P = (\boldsymbol{\nu}\widetilde{p})(a(x).P_1 \mid \overline{b}c.P_2 \mid P_3)$ and $P' = (\boldsymbol{\nu}\widetilde{p})(P_1 \mid P_2 \mid P_3)$.

Note that by our conventions on notations, $c \notin \widetilde{p}$.

Since $Q \sim_{\mathrm{g}} P$ and P can perform two transitions along $a(x)$ and $\overline{b}c$ respectively, Q can also perform these transitions, which gives

$Q \equiv (\boldsymbol{\nu}\widetilde{q})(a(x).Q_1 \mid \overline{b}c.Q_2 \mid Q_3)$ for some $\widetilde{q}, Q_1, Q_2, Q_3$,

the first (resp. second) component exhibiting the prefix that is triggered to answer the challenge on $a(x)$ (resp. $\overline{b}c$).

Consider now the challenge $P \xrightarrow{\overline{b}c}_{\pi}\xrightarrow{a(x)}_{\pi} P'$, to which Q answers by performing $Q \xrightarrow{\overline{b}c}_{\pi}\xrightarrow{a(x)}_{\pi} Q_{ba}$, with $P' \sim_{\mathrm{g}} Q_{ba}$. If $Q_{ba} = (\boldsymbol{\nu}\widetilde{q})(Q_1 \mid Q_2 \mid Q_3)$, that is, if Q triggers the prefixes on top of its first and second components, then we are done. Similarly, if Q triggers a prefix in Q_3 to answer the second challenge, say $Q_3 = a(x).Q_4 \mid Q_5$, we can set $Q_1' = a(x).Q_4$ and $Q_3' = Q_1 \mid Q_5$, and the lemma is proved.

The case that remains to be analysed is when $Q_2 \xrightarrow{a(x)}_{\pi} Q_2'$ and $Q_{ba} = (\boldsymbol{\nu}\widetilde{q})(a(x).Q_1 \mid Q_2' \mid Q_3) \sim_{\mathrm{g}} (\boldsymbol{\nu}\widetilde{p})(P_1 \mid P_2 \mid P_3)$.

We then consider the challenge where P fires its two topmost prefixes $a(x)$ and $\overline{b}c$ in the other sequence, namely $P \xrightarrow{a(x)}_{\pi}\xrightarrow{\overline{b}c}_{\pi} P'$. By hypothesis, Q triggers the prefix of its first component for the first transition. To perform the second transition, Q can fire the prefix $\overline{b}c$ either in its second or third component, in which case, as above, we are done, or, and this is the last possibility, the prefix $\overline{b}c$ occurs in Q_1. This means $Q_{ab} = (\boldsymbol{\nu}\widetilde{q})(Q_1' \mid \overline{b}c.Q_2 \mid Q_3) \sim_{\mathrm{g}} (\boldsymbol{\nu}\widetilde{p})(P_1 \mid P_2 \mid P_3)$, with $Q_1 \xrightarrow{\overline{b}c}_{\pi} Q_1'$.

To sum up, we have $Q_{ab} = (\boldsymbol{\nu}\widetilde{q})(Q_1' \mid \overline{b}c.Q_2 \mid Q_3) \sim_{\mathrm{g}} (\boldsymbol{\nu}\widetilde{q})(a(x).Q_1 \mid Q_2' \mid Q_3) = Q_{ba}$, with $Q_1 \xrightarrow{\overline{b}c}_{\pi} Q_1'$ and $Q_2 \xrightarrow{a(x)}_{\pi} Q_2'$: this resembles the mutual desynchronisation of Definition 4.3, translated into the π-calculus.

Indeed, we can construct a mutual desynchronisation in μCCS: $Q_{ab} \sim_{\mathrm{g}} Q_{ba}$ implies $\mathcal{E}(Q_{ab}) \sim \mathcal{E}(Q_{ba})$ by Prop. 4.1, and $Q_1 \xrightarrow{\overline{b}c}_{\pi} Q_1'$ (resp. $Q_2 \xrightarrow{a(x)}_{\pi} Q_2'$) implies by Lemma 4.8 $\mathcal{E}(Q_1) \xrightarrow{\overline{b}} \mathcal{E}(Q_1')$ (resp. $\mathcal{E}(Q_2) \xrightarrow{a} \mathcal{E}(Q_2')$). Finally, using Lemma 4.6, we obtain a contradiction, which concludes our proof. □

Lemma 4.10. *If $Q \sim_{\mathrm{g}} (\boldsymbol{\nu}p, \widetilde{p})(a(x).P_1 \mid \overline{b}p.P_2 \mid P_3)$, then there exist some Q_1, Q_2, Q_3, such that $Q \equiv (\boldsymbol{\nu}p, \widetilde{q})(a(x).Q_1 \mid \overline{b}p.Q_2 \mid Q_3)$ and*

$$(\boldsymbol{\nu}\widetilde{p})(P_1 \mid P_2 \mid P_3) \ \sim_{\mathrm{g}} \ (\boldsymbol{\nu}\widetilde{q})(Q_1 \mid Q_2 \mid Q_3).$$

Proof (Hint). The proof follows the same lines as for the previous lemma. The only difference is when analysing the transitions that lead to Q_{ab}: to perform the second transition, Q can either extrude the name called p in the equality $Q \equiv (\boldsymbol{\nu}p, \widetilde{q})(a(x).Q_1 \mid \overline{b}p.Q_2 \mid Q_3)$, or otherwise Q can be α-converted in order to extrude another name. In the case where Q chooses to extrude a different name, we can suppose without loss of generality that the necessary α-conversion is a swapping between name p and a name $q_1 \in \widetilde{q}$, which brings us back to the case where name p is the one being extruded.

The presence of a bound output introduces some notational complications when expressing Q_{ab}, but basically it does not affect the proof w.r.t. the proof of Lemma 4.9, because the function $\mathcal{E}(\cdot)$ is not sensitive to name permutations that do not involve a or b. □

4.3 Congruence

Theorem 4.11 (Closure of $\sim_{\mathrm{g}}$ under substitution). *If $P \sim_{\mathrm{g}} Q$ then for any substitution σ, $P\sigma \sim_{\mathrm{g}} Q\sigma$.*

Proof. We prove that the relation $\mathcal{R} \overset{\text{def}}{=} \{(P\sigma, Q\sigma) \mid P \sim_{\mathrm{g}} Q\}$ is a ground bisimulation. We consider P, Q such that $P \sim_{\mathrm{g}} Q$ and suppose $P\sigma \xrightarrow{\mu}_{\pi} P_0$. We examine the transitions of P that make it possible for $P\sigma$ to do a μ-transition to P_0.

According to Lemma 4.2, there are two possibilities. The first possibility corresponds to the situation where μ comes from an action that P can perform, i.e., $P \xrightarrow{\mu'}_{\pi} P'$ for some μ', with $P'\sigma = P_0$ and $\mu'\sigma = \mu$ (cases 1 and 2a in Lemma 4.2). Since $P \sim_{\mathrm{g}} Q$, $Q \xrightarrow{\mu'}_{\pi} Q'$ and $P' \sim_{\mathrm{g}} Q'$ for some Q'. We can prove that $Q\sigma \xrightarrow{\mu} Q'\sigma$, and since $P' \sim_{\mathrm{g}} Q'$ we have $(P'\sigma, Q'\sigma) \in \mathcal{R}$.

The second possibility (which corresponds to the difficult case) is given by $\mu = \tau$, where the synchronisation in P' has been made possible by the application of σ. There are in turn two cases, corresponding to whether the synchronisation involves a free or a bound name. In the former case, $P \xrightarrow{a(x)}_{\pi} P'$ and $P \xrightarrow{\bar{b}c}_{\pi} P''$ for some a, x, b, c, P', P''. This entails $P \equiv (\boldsymbol{\nu}\widetilde{p})(a(x).P_1 \,|\, \bar{b}c.P_2 \,|\, P_3)$ for some $\widetilde{p}, P_1, P_2, P_3$, and, since $P \sim_{\mathrm{g}} Q$, we conclude by Lemma 4.9 that $Q \equiv (\boldsymbol{\nu}\widetilde{q})(a(x).Q_1 \,|\, \bar{b}c.Q_2 \,|\, Q_3)$ and

$$(\boldsymbol{\nu}\widetilde{p})(P_1 \,|\, P_2 \,|\, P_3) \;\sim_{\mathrm{g}}\; (\boldsymbol{\nu}\widetilde{q})(Q_1 \,|\, Q_2 \,|\, Q_3) \;\; .$$

By definition of $\mathcal{R}$, this equivalence implies that we can apply any substitution to these two processes to yield processes related by $\mathcal{R}$, and in particular $[c/x]\sigma$, which gives:

$$((\boldsymbol{\nu}\widetilde{p})(P_1 \,|\, P_2 \,|\, P_3))[c/x]\sigma \;\; \mathcal{R} \;\; ((\boldsymbol{\nu}\widetilde{q})(Q_1 \,|\, Q_2 \,|\, Q_3))[c/x]\sigma \;\; .$$

Using the Barendregt convention hypothesis, this amounts to

$$P_0 \equiv ((\boldsymbol{\nu}\widetilde{p})(P_1[c/x] \,|\, P_2 \,|\, P_3))\sigma \;\; \mathcal{R} \;\; ((\boldsymbol{\nu}\widetilde{q})(Q_1[c/x] \,|\, Q_2 \,|\, Q_3))\sigma \overset{\text{def}}{=} Q_0 \;\; .$$

We can then conclude by checking that $Q\sigma \xrightarrow{\tau}_{\pi} Q_0$.

We reason similarly for the case where the synchronisation involves the transmission of a bound name, using Lemma 4.10 instead of Lemma 4.9. We remark that Lemma 4.10 gives $(\boldsymbol{\nu}\widetilde{p})(P_1 \,|\, P_2 \,|\, P_3) \sim_{\mathrm{g}} (\boldsymbol{\nu}\widetilde{q})(Q_1 \,|\, Q_2 \,|\, Q_3)$, and in this case $P\sigma \xrightarrow{\tau}_{\pi} (\boldsymbol{\nu}p, \widetilde{p})(P_1[p/x] \,|\, P_2 \,|\, P_3)\sigma$ (resp. $Q\sigma \xrightarrow{\tau}_{\pi} (\boldsymbol{\nu}p, \widetilde{q})(Q_1[p/x] \,|\, Q_2 \,|\, Q_3)\sigma$). In order to be able to add the restriction on p to the terms given by Lemma 4.10, we rely on the fact that $\sim_{\mathrm{g}}$ is preserved by restriction: $P \sim_{\mathrm{g}} Q$ implies $(\boldsymbol{\nu} p)P \sim_{\mathrm{g}} (\boldsymbol{\nu} p)Q$ for any P, Q, p. We can then reason as above to conclude. □

Corollary 4.12 (Congruence of bisimilarity in π_0). *In π_0, ground, early and late bisimilarity coincide and are congruences.*

Proof. By a standard argument (see [13]): since $\sim_{\mathrm{g}}$ is closed under substitution, $\sim_{\mathrm{g}}$ is an open bisimulation. □

It is known (see [13]) that adding either replication or sum to π_0 yields a calculus where strong bisimilarity fails to be a congruence.

5 Conclusion

We have presented an axiomatisation of strong bisimilarity on a small subcalculus of CCS, and a new congruence result for the π-calculus.

Technically, the notion of mutual desynchronisation is related to substitution closure of strong bisimilarity, as soon as substitutions can create new interactions by identifying two names.

We have shown in Section 4 that there exists no mutual desynchronisation in π_0, and that $\sim_{\mathrm{g}}$ is a congruence. In (full) CCS, mutual desynchronisations exist, a simple example being given by $a.\overline{b} + \overline{b}.a$. The latter process is bisimilar to $a \,|\, \overline{b}$, but the equality fails to hold when b is replaced with a. The same reasoning holds for the π-calculus with choice. It hence appears that in finite calculi, mutual desynchronisations give rise to counterexamples to substitution closure of strong bisimilarity. The situation is less clear when infinite behaviours can be expressed. For instance, in the extension of μCCS with replication, the process $!a \,|\, !\overline{b}$ is bisimilar to $P \overset{\mathrm{def}}{=} !a.\overline{b} \,|\, !\overline{b}.a$. Process P leads to a mutual desynchronisation: we have $P \xrightarrow{a}\xrightarrow{\overline{b}} \equiv P \xrightarrow{\overline{b}}\xrightarrow{a} \equiv P$. We do not know at present whether $\sim$ is substitution-closed in this extension of μCCS (we may remark that the two aforementioned processes remain bisimilar when b is replaced with a).

Some subcalculi of the π-calculus where strong bisimilarity is a congruence are obtained by restricting the output prefix [13]. In the *asynchronous π-calculus* ($A\pi$), mutual desynchronisations do not appear, basically because the output action is not a prefix. Strong bisimilarity is a congruence on $A\pi$. In the *private π-calculus* ($P\pi$), since only private names are emitted, no substitution generated by a synchronisation can identify two previously distinct names. Hence, although mutual desynchronisations exist in $P\pi$ (due to the presence of the sum operator), strong bisimilarity is not substitution closed, but is a congruence. Indeed, to obtain the latter property, we only need to consider the particular substitutions at work in $P\pi$, which cannot identify two names.

The question of substitution closure can also be raised in the framework of *location sensitive behavioural equivalences* such as distributed bisimilarity (see [4]). Without having a formal proof for this claim, we expect this equivalence to be substitution closed on restriction-free CCS. We believe this should be the case because in absence of restriction, distributed bisimilarity is discriminating enough to analyse the maximum degree of parallelism in processes (in particular, the expansion law is not valid for location sensitive equivalences).

Regarding future extensions of this work, we would like to study whether our approach can be adapted to analyse weak bisimilarity in π_0 (as mentioned in Remark 2.3, strong and weak bisimilarity coincide in μCCS). Another interesting direction, as hinted above, would be to study strong bisimilarity on infinite, restriction-free calculi (in CCS and the π-calculus).

Acknowledgements. We are grateful to Arnaud Carayol for interesting discussions at early stages of this work. An anonymous referee provided numerous helpful suggestions, which helped us in particular to improve the proof of Theorem 2.6. We benefited from support by the french initiative “ACI GEOCAL” and from the ANR project “MoDyFiable”.

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. O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification over Infinite States. In *Handbook of Process Algebra*, pages 545–623. Elsevier, 2001.
4. I. Castellani. *Handbook of Process Algebra*, chapter Process Algebras with Localities, pages 945–1045. North-Holland, 2001.
5. F. Corradini, R. Gorrieri, and D. Marchignoli. Towards parallelization of concurrent systems. *Informatique Théorique et Applications*, 32(4-6):99–125, 1998.
6. W. Fokkink and B. Luttik. An ω-complete Equational Specification of Interleaving. In *Proc. of ICALP'00*, volume 1853 of *LNCS*, pages 729–743. Springer Verlag, 2000.
7. Y. Hirshfeld and M. Jerrum. Bisimulation Equivalence is Decidable for Normed Process Algebra. Technical Report ECS-LFCS-98-386, LFCS, 1998.
8. Y. Hirshfeld and M. Jerrum. Bisimulation Equivalence is Decidable for Normed Process Algebra. In *Proc. of ICALP'99*, volume 1644 of *LNCS*, pages 412–421. Springer Verlag, 1999.
9. B. Luttik. What is Algebraic in Process Theory? *Concurrency Column, Bulletin of the EATCS*, 88, 2006.
10. R. Milner and F. Moller. Unique Decomposition of Processes. *TCS*, 107(2):357–363, 1993.
11. F. Moller. *Axioms for Concurrency.* PhD thesis, University of Edinburgh, 1988.
12. D. Sangiorgi. A Theory of Bisimulation for the π-Calculus. *Acta Informatica*, 33(1):69–97, 1996.
13. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge University Press, 2001.
14. B. Victor, F. Moller, M. Dam, and L.-H. Eriksson. The Mobility Workbench. available from `http://www.it.uu.se/research/group/mobility/mwb`, 2006.