Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 194 (2008) 59–84
www.elsevier.com/locate/entcs

Linearity, Persistence and Testing Semantics in the Asynchronous Pi-Calculus

Diletta Cacciagrano,[2] Flavio Corradini[3]

Dipartimento di Matematica e Informatica
Università degli Studi di Camerino, Italy

Jesús Aranda [1,4]

INRIA Futurs, LIX École Polytechnique, France
Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Colombia

Frank D. Valencia[5]

CNRS and LIX École Polytechnique, France

Abstract

In [24] the authors studied the expressiveness of persistence in the asynchronous π-calculus (Aπ) wrt weak barbed congruence. The study is incomplete because it ignores the issue of divergence. In this paper, we present an expressiveness study of persistence in the asynchronous π-calculus (Aπ) wrt De Nicola and Hennessy's testing scenario which is sensitive to divergence. Following [24], we consider Aπ and three sub-languages of it, each capturing one source of persistence: the *persistent-input* calculus (PIAπ), the *persistent-output* calculus (POAπ) and *persistent* calculus (PAπ). In [24] the authors showed encodings from Aπ into the semi-persistent calculi (i.e., POAπ and PIAπ) correct wrt weak barbed congruence. In this paper we prove that, under some general conditions, there cannot be an encoding from Aπ into a (semi)-persistent calculus preserving the *must* testing semantics.

Keywords: Asynchronous Pi-Calculus, Linearity, Persistence, Testing Semantics.

[1] The work of Jesús Aranda has been supported by COLCIENCIAS (Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología "Francisco José de Caldas") and INRIA Futurs.

[2] Email:diletta.cacciagrano@unicam.it

[3] Email:flavio.corradini@unicam.it

[4] Email:jesus.aranda@lix.polytechnique.fr

[5] Email:frank.valencia@lix.polytechnique.fr

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2007.11.006

1 Introduction

In [24] the authors present an expressiveness study of linearity and persistence of processes. Since several calculi presuppose persistence on their processes, the authors address the expressiveness issue of whether such persistence restricts the systems that we can specify, model or reason about in the framework. Their work is conducted using the standard notion of weak barbed congruence and hence it ignores divergence issues. Since divergence plays an important role in expressiveness studies, particularly in those studies involving persistence, in this work we aim at extending and strengthening their study by using the standard notion of testing equivalences. As elaborated below, our technical results contrast and complement those in [24]. More importantly, our results also clarify and support informal expressiveness claims in the literature.

Motivation: *Linearity* is present in process calculi such as CCS, CSP, the π-calculus [20] and Linear CCP [31,14], where messages are consumed upon being received. In the π-calculus the system $\bar{x}z \mid x(y).P \mid x(y).Q$ represents a message with a datum z, tagged with x, that can be *consumed* by either $x(y).P$ or $x(y).Q$. *Persistence of messages* is present in several process calculi. Perhaps the most prominent representative of such calculi is Concurrent Constraint Programming (CCP) [32]. Here the messages (or items of information) can be read but, unlike in Linear CCP, they cannot be consumed. Other prominent examples can be found in the context of calculi for analyzing and describing security protocols: Crazzolara and Winskel's SPL [12], the Spi Calculus variants by Fiore and Abadi [15] and by Amadio et all [2], and the calculus of Boreale and Buscemi [5] are operationally defined in terms of configurations containing messages which cannot be consumed. *Persistent receivers* arise, e.g. in the notion of *omega receptiveness* [29], where the input of a name is always available—but always with the same continuation. In the π-calculus persistent receivers are used, for instance, to model functions, objects, higher-order communications, or procedure definitions. Furthermore, persistence of *both* messages and receivers arise in the context of CCP with universally-quantified persistent ask operations. In the context of calculi for security, persistent receivers can be used to specify protocols where principals are willing to run an unbounded number of times (and persistent messages to model the fact that every message can be remembered by the spy). In fact, the approach of specifying protocols in a persistent setting, with an unbounded number of sessions, has been explored in [4] by using a classic logic Horn clause representation of protocols (rather than a linear logic one).

Expressiveness of Persistence - Drawbacks and Conjectures: The study in [24] is conducted in the *asynchronous* π-calculus (Aπ), which naturally captures the persistent features mentioned above. Persistent messages (and receivers) can simply be specified using the *replication* operator of the calculus which creates an unbounded number of copies of a given process. In particular, the authors in [24] investigate the existence of encodings from Aπ into three sub-languages of it, each capturing one source of persistence: the *persistent-input* calculus (PIAπ), defined as

Aπ where inputs are replicated; *persistent-output* calculus (POAπ), defined dually, i.e. outputs rather than inputs are replicated; *persistent* calculus (PAπ), defined as Aπ but with all inputs and outputs are replicated. The main result basically states that we need one source of linearity, i.e. either on inputs (PIAπ) or outputs (POAπ) to encode the behavior of arbitrary Aπ processes via weak barbed congruence. Nevertheless, the main drawback of the work [24] is that the notion of correct encoding is based on weak barbed bisimulation (congruence), which is not sensitive to *divergence*. In particular, the encoding provided in [24] from Aπ into PIAπ is weak barbed congruent preserving but not divergence preserving. Although in some situations divergence may be ignored, in general it is an important issue to consider in the correctness of encodings [8,17,16,18,7].

In fact, the informal claims of extra expressivity of Linear CCP over CCP in [3,14] are based on discrimination introduced by divergence that is clearly ignored by the standard notion of weak bisimulation. Furthermore, the author of [11] suggests as future work to extend SPL, which uses only persistent messages and replication, with recursive definitions to be able to program and model recursive protocols such as those in [1,25]. Nevertheless, one can give an encoding of recursion in SPL from an easy adaptation of the composition between the Aπ encoding of recursion [30] (where recursive calls are translated into *linear* Aπ outputs and recursive definitions into persistent inputs) and the encoding of Aπ into POAπ in [24]. The resulting encoding is correct up-to weak bisimulation. The encoding of Aπ into POAπ, however, introduces divergence and hence the composite encoding does not seem to invalidate the justification for extending SPL with recursive definitions. The above works suggest that the expressiveness study of persistence is relevant but incomplete if divergence is not taken into account.

This work: In this paper we shall therefore study the existence of encodings from Aπ into the persistent sub-languages mentioned above using testing semantics [13].

Our main contribution is to provide a uniform and general result stating that, under some reasonable conditions, Aπ cannot be encoded into any of the above (semi-) persistent calculi while preserving the *must* testing semantics. The general conditions involve compositionality on the encoding of constructors such as parallel composition, prefix, and replication. The main result contrasts and completes the ones in [24]. It also supports the informal claims of extra expressivity mentioned above. We shall also state other more specialized impossibility results for must preserving encodings from Aπ into the semi-persistent calculi, focusing on specific properties of each target calculus. This helps clarifying some previous assumptions on the interplay between syntax and semantics in encodings of process calculi. We believe that, since the study is conducted in Aπ with well-established notions of equivalence, we can easily adapt our results to other asynchronous frameworks such as CCP languages and the above-mentioned calculi for security.

2 The calculi

Here we define the calculi we study. We first recall the (monadic) *asynchronous π-calculus* (Aπ). The other calculi are defined as syntactic restrictions of Aπ.

2.1 The asynchronous pi-calculus

Let $\mathcal{N}$ (ranged over by $x, y, z, \ldots$) be a set of names. The set of the asynchronous π-calculus processes (ranged over by P, Q, $R \ldots$) is generated by the following grammar:

$$P, Q, \ldots ::= \quad 0 \;\Big|\; \bar{x}z \;\Big|\; x(y).P \;\Big|\; P \mid Q \;\Big|\; (\nu x)P \;\Big|\; !\,P$$

Intuitively, an *output* $\bar{x}z$ represents a message z tagged with a name x indicating that it can be received (or *consumed*) by an *input process* $x(y).P$ which behaves, upon receiving z, as $P\{z/y\}$. Furthermore, $x(y).P$ binds the names y in P. The other binder is the *restriction* $(\nu x)P$ which declares a name x private to P. The *parallel composition* $P \mid Q$ means P and Q running in parallel. The *replication* $!P$ means $P \mid P \mid \ldots$, i.e., $!P$ represents a *persistent resource*.

We use the standard notations $bn(Q)$ for the *bound names* in Q, and $fn(Q)$ for the *free names* in Q. The set of names of P is defined as $n(P) = fn(P) \cup bn(P)$. We let $\sigma, \vartheta \ldots$ range over (non-capturing) substitutions of names on processes.

The *reduction* relation $\longrightarrow$ is the least binary relation on processes satisfying the rules in Table 1. $\stackrel{*}{\longrightarrow}$ denotes the reflexive, transitive closure of $\longrightarrow$. The reductions are quotiented by the *structural congruence* relation $\equiv$.

Definition 2.1 [Structural congruence] Let $\equiv$ be the smallest congruence over processes satisfying α-equivalence, the commutative monoid laws for composition with 0 as identity, the replication law $!P \equiv P \mid !P$, the restriction laws $(\nu x)0 \equiv 0$, $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$ and the extrusion law: $(\nu x)(P \mid Q) \equiv P \mid (\nu x)Q$ if $x \notin fn(P)$.

$$\text{Com} \quad \bar{x}z \mid x(y).P \longrightarrow P\{z/y\}$$

$$\text{Par} \quad \frac{P \longrightarrow P'}{P \mid Q \longrightarrow P' \mid Q} \qquad \text{Res} \quad \frac{P \longrightarrow P'}{(\nu x)P \longrightarrow (\nu x)P'}$$

$$\text{Cong} \quad \frac{P \equiv P' \quad P' \longrightarrow Q' \quad Q' \equiv Q}{P \longrightarrow Q}$$

Table 1
Reduction Rules.

2.2 The (semi-)persistent calculi

The *persistent-input* calculus PIAπ results from Aπ by requiring all input processes to be replicated. Processes in PIAπ are generated by the following grammar:

$$P, Q, \ldots ::= \quad 0 \;\Big|\; !\,x(y).P \;\Big|\; \bar{x}y \;\Big|\; P \mid Q \;\Big|\; (\nu x)P \;\Big|\; !\,P$$

The *persistent-output* calculus POAπ arises as from Aπ by requiring all outputs to be replicated. Processes in POAπ are generated by the following grammar:

$$P, Q, \ldots ::= \quad 0 \;\Big|\; x(y).P \;\Big|\; !\,\bar{x}y \;\Big|\; P \mid Q \;\Big|\; (\nu x)P \;\Big|\; !\,P$$

Finally, we have the *persistent* calculus PAπ, a subset of Aπ where output and input processes must be replicated. Processes in PAπ are generated by the following grammar:

$$P, Q, \ldots ::= \quad 0 \;\Big|\; !\,x(y).P \;\Big|\; !\,\bar{x}y \;\Big|\; P \mid Q \;\Big|\; (\nu x)P \;\Big|\; !\,P$$

The relation $\longrightarrow$ for PIAπ, POAπ and PAπ can be equivalently defined as in Table 1, with Com replaced respectively with Com(PIAπ), Com(POAπ) and Com(PAπ) rules (Table 2). The new rules reflect the *persistent-input* and *linear-output* nature of PIAπ (Rule Com(PIAπ)), the *linear-input* and *persistent-output* nature of POAπ (Rule Com(POAπ)), and the *persistent* nature of PAπ (Rule Com(PAπ)).

Com(PIAπ)	$\bar{x}z \mid !\,x(y).P \longrightarrow P\{z/y\} \mid !\,x(y).P$
Com(POAπ)	$!\,\bar{x}z \mid x(y).P \longrightarrow !\,\bar{x}z \mid P\{z/y\}$
Com(PAπ)	$!\,\bar{x}z \mid !\,x(y).P \longrightarrow P\{z/y\} \mid !\,\bar{x}z \mid !\,x(y).P$

Table 2
Reduction Rules.

Notation 2.1 *We shall use $\mathcal{P}$ to range over the set of the calculi so-far defined $\{A\pi, PIA\pi, POA\pi, PA\pi\}$.*

3 Testing semantics

In [13] De Nicola and Hennessy propose a framework for defining pre-orders that is widely acknowledged as a realistic scenario for system testing. It means to define formally when one process is a correct implementation of another considering specially unsafe contexts, in which is particularly important what is the revealed information of the process in any context or test. In this section we summarize the basic definitions behind the testing machinery for the π-calculi.

Definition 3.1 [Observers]

- The set of names $\mathcal{N}$ is extended as $\mathcal{N}' = \mathcal{N} \cup \{\omega\}$ with $\omega \notin \mathcal{N}$. By convention we let $fn(\omega) = \{\omega\}$ and $bn(\omega) = \emptyset$ (ω is used to report success).

- The set $\mathcal{O}$ (ranged over by $o, o', o'', E, E', \ldots$) of observers (tests) is defined like $\mathcal{P}$, where the grammar is extended with the production $P ::= \omega.P$.
- $\xrightarrow{\omega}$ is the least predicate over $\mathcal{O}$ satisfying the inference rules in Table 3.

$$\text{Omega}\quad \omega.E \xrightarrow{\omega} \qquad \text{Res}\quad \frac{E \xrightarrow{\omega}}{(\nu y)E \xrightarrow{\omega}}$$

$$\text{Par}\quad \frac{E_1 \xrightarrow{\omega}}{E_1 \mid E_2 \xrightarrow{\omega}} \qquad \text{Cong}\quad \frac{E' \xrightarrow{\omega} \quad E' \equiv E}{E \xrightarrow{\omega}}$$

Table 3
Predicate $\xrightarrow{\omega}$.

Definition 3.2 [Maximal computations] Given $P \in \mathcal{P}$ and $o \in \mathcal{O}$, a maximal computation from $P \mid o$ is either an infinite sequence of the form

$$P \mid o = E_0 \longrightarrow E_1 \longrightarrow E_2 \longrightarrow \ldots$$

or a finite sequence of the form

$$P \mid o = E_0 \longrightarrow E_1 \longrightarrow \ldots \longrightarrow E_n \not\longrightarrow .$$

Definition 3.3 [May, must and fair relations [6]] Given $P \in \mathcal{P}$ and $o \in \mathcal{O}$, define:

- P *may* o if and only if *there is* a maximal computation (as in Def. 3.2) such that $E_i \xrightarrow{\omega}$, for some $i \geq 0$;
- P *must* o if and only if *for every* maximal computation (as in Def. 3.2) there exists $i \geq 0$ such that $E_i \xrightarrow{\omega}$;
- P *fair* o [6] if and only if *for every* maximal computation (as in Def. 3.2) and $\forall i \geq 0$, $\exists\, E_i'$ such that $E_i \xrightarrow{*} E_i'$ and $E_i' \xrightarrow{\omega}$.

4 Encoding linearity into persistence

First, we recall some notions about encodings. An encoding is a mapping from the terms of a calculus into the terms of another. In general a "good" encoding satisfies some additional requirements, but there is no agreement on a general notion of "good" encoding. Perhaps indeed there should not be a unique notion, but several, depending on the purpose. In this paper we shall study the existence of encodings $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P}$ from π into $\mathcal{P} \in \{\mathrm{PA}\pi, \mathrm{PIA}\pi, \mathrm{POA}\pi\}$ and focus on typical requirements such as compositionality w.r.t. certain operators, and the correctness w.r.t. a given semantics.

[6] It may be possible to give other equivalent definitions not based on maximal computations by using properties of the calculi under consideration such as: if $P \xrightarrow{\omega}$ and $P \longrightarrow P'$ then $P' \xrightarrow{\omega}$. For uniformity, however, we have used a well-known testing semantics definition based on the notion of *maximal* computations.

Compositionality and multi-hole contexts: We shall use notion of (multi-hole) process *contexts* [30] to describe compositionality. Recall that a $\mathcal{P}$ context C with k holes is a term with occurrences of k distinct *holes* $[\,]_1, \ldots, [\,]_k$ such that a $\mathcal{P}$ process must result from C if we replace all the occurrences of each $[\,]_i$ with a $\mathcal{P}$ process. The context C is *singularly-structured* if each hole occurs exactly once. For example, $[\,]_1 \mid x(y).([\,]_2 \mid [\,]_1)$ is an Aπ non singularly-structured context with two holes. Given $P_1, \ldots, P_k \in \mathcal{P}$ and a context C with k holes, $C[P_1, \ldots, P_k]$ is the process that results from replacing the occurrences of each $[\,]_i$ with P_i. The names of a context C with k holes, $n(C)$, are those of $C[Q_1, \ldots, Q_k]$ where each Q_i is 0. The free and bound names of a context are defined analogously. We can regard the input prefix $x(y)$, $\mid$ and ! as the operators of arity 1, 2 and 1 respectively in Aπ in the obvious sense.

Definition 4.1 [Compositionality w.r.t. an operator] Let *op* be an n-ary operator of Aπ. An encoding $[\![\cdot]\!] : \text{A}\pi \to \mathcal{P}$ is *compositional* w.r.t. *op* iff there is a $\mathcal{P}$ context C_{op} with n holes such that $[\![op(P_1, .., P_n)]\!] = C_{op}[[\![P_1]\!], .., [\![P_n]\!]]$.

In the following, $C[\cdot]$ denotes contexts with one hole and $C[\cdot, \cdot]$ contexts with two holes. Furthermore, given an encoding $[\![\cdot]\!] : \text{A}\pi \to \mathcal{P}$, we define ${C_{op}}^{[\![\cdot]\!]}$ as the context C such that $[\![op(P_1, \ldots, P_n)]\!] = C[[\![P_1]\!], \ldots, [\![P_n]\!]]$. We shall often omit the "$[\![\cdot]\!]$" in ${C_{op}}^{[\![\cdot]\!]}$ since it is easy to infer from the context.

Remark 4.2 [Homomorphism wrt parallel composition] An interesting case of compositionality is *homomorphism* w.r.t a given operator *op*: The operator is mapped into the same operator of the target language, i.e. $[\![op(P_1, .., P_n)]\!] = op([\![P_1]\!], \ldots, [\![P_n]\!])$. Homomorphism w.r.t parallelism, also called distribution-preserving [33,26,27], can arguably be considered as a reasonable requirement for an encoding. In particular, the works [33,26,27,23,9,16,17] support the distribution-preserving hypothesis by arguing that it corresponds to requiring that the degree of distribution of the processes is maintained by the translation, i.e. no coordinator is added. Some of these works are in the context of solving electoral problems and some others in more general scenarios [16,17]. Other works [22,28], however, argue that the requirement can be quite demanding as it rules out practical implementation of distributed systems. Some of our impossibility results will appeal to the distribution-preserving hypothesis.

Remark 4.3 Typically, the C_{op} mentioned in Definition 4.1 is a singularly-structured multi-hole context in encodings of operators such as input prefix, parallel composition and replication. Note that, if the encoding is homomorphic wrt *op*, then C_{op} is a singularly-structured multi-hole context.

Correctness wrt testing: Concerning semantic correctness, we consider preservation of *sat* testing, where *sat* can be respectively *may*, *must* and *fair*. Given an encoding $e = [\![\cdot]\!] : \text{A}\pi \to \mathcal{P}$, we assume that its lifted version e' from the set of observers of π to the ones of $\mathcal{P}$ is an encoding satisfying the following: $e'(o) = e(o)$, in the case o has no occurrences of ω.

Definition 4.4 [Soundness, completeness and *sat*-preservation] Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P}$. We say that $[\![\cdot]\!]$ is:

- *sound w.r.t. sat* iff $\forall\, P \in \mathrm{A}\pi$, $\forall\, o \in \mathcal{O}$, $[\![P]\!]$ *sat* $[\![o]\!]$ implies P *sat* o;
- *complete w.r.t. sat* iff $\forall\, P \in \mathrm{A}\pi$, $\forall\, o \in \mathcal{O}$, P *sat* o implies $[\![P]\!]$ *sat* $[\![o]\!]$;
- *sat-preserving* iff $[\![\cdot]\!]$ is *sound* and *complete* w.r.t. *sat*.

4.1 Some encodings from asynchronous pi-calculus into its semi-persistent subsets

We consider the following encoding from Aπ to PIAπ, defined in [24].

Definition 4.5 The encoding $[\![\cdot]\!] : \mathrm{A}\pi \to \mathrm{PIA}\pi$ is a homomorphism for 0, parallel composition, restriction and replication, otherwise is defined as

- $[\![\bar{x}z]\!] = \bar{x}z$, and
- $[\![x(y).P]\!] = (\nu tf)(\bar{t} \mid !x(y).(\nu l)(\bar{l} \mid !t.!l.([\![P]\!] \mid !\bar{f}) \mid !f.!l.\bar{x}y))$
  where $t, f, l \notin fn(P) \cup \{x, y\}$. (The lifted version is given adding $[\![\omega.P]\!] = \omega.[\![P]\!]$.)

This encoding enjoys a strong property: namely, for any P, $[\![P]\!] \approx P$, where $\approx$ denotes weak barbed congruence [30]. This implies, in the testing scenario, a property stronger than *sat*-preservation.

Proposition 4.6 *Let* $[\![\cdot]\!] : A\pi \to PIA\pi$ *as in Definition 4.5.* $\forall\, P \in \mathrm{A}\pi$, $\forall\, o \in \mathcal{O} \subseteq \mathrm{PIA}\pi$ P *sat* o iff $[\![P]\!]$ *sat* o, where *sat* can be respectively *may* and *fair*.

To prove that the statement does not hold in the case of *must* semantics, consider $P = (a.0 \,|\, !\bar{a})$ and $o = a.\omega.0$: then P *must* o but $[\![P]\!]$ *must* o.

Extending the notion of barb to ω, clearly $P \,|\, o \approx [\![P \,|\, o]\!]$ as $P \,|\, o \in \mathrm{A}\pi$ and, by homomorphism w.r.t parallel composition, we obtain that $P \,|\, o \approx [\![P]\!] \,|\, [\![o]\!]$. This is enough to hold fair- and may-preserving.

In [24] the encoding in Definition 4.5 is used to get an encoding of Aπ into POAπ, by composing it with the following mapping from PIAπ into POAπ.

Definition 4.7 The encoding $f = [\![\cdot]\!] : \mathrm{PIA}\pi \to \mathrm{POA}\pi$ is a homomorphism for 0, parallel composition, restriction, and replication, otherwise is defined as

- $[\![\bar{x}z]\!] = (\nu s)(!\bar{x}s \mid s(r).!\bar{r}z)$, and
- $[\![!x(y).P]\!] = !x(s).(\nu r)(!\bar{s}r \mid r(y).[\![P]\!])$
  where $s, r \notin fn(P) \cup \{x, z\}$. (The lifted version is given adding $[\![\omega.P]\!] = \omega.[\![P]\!]$.)

Let g be $[\![\cdot]\!] : \mathrm{A}\pi \to \mathrm{PIA}\pi$ in Definition 4.5. The encoding $h = [\![\cdot]\!] : \mathrm{A}\pi \to \mathrm{POA}\pi$ is the composite function $f \circ g$.

Because of this encoding maps a linear output into a replicated one with the same barb, the composite encoding $h = [\![\cdot]\!] : \mathrm{A}\pi \to \mathrm{POA}\pi$ in Definition 4.7 does not satisfy $[\![P]\!] \approx P$. It has a weaker property: namely, $P \approx Q$ iff $[\![P]\!] \approx \overset{\mathrm{POA}\pi}{[\cdot]} [\![Q]\!]$, where $[\![P]\!] \approx \overset{\mathrm{POA}\pi}{[\cdot]} [\![Q]\!]$ means that $\forall C$ context in Aπ, $[\![C]\!][\![P]\!]$ and $[\![C]\!][\![Q]\!]$ (assuming $[\![\,[\,]\,]\!] = [\,]$) are weak barbed bisimilar [30]. Similarly, the results for the

composite encoding from Aπ into POAπ in a testing scenario are weaker than these ones for the encoding from Aπ into PIAπ. Obviously, the following proposition would not hold if *sat* were *must*. Consider $P = !\bar{a}$ and $o = a.\omega.0$: then P *must* o but $[\![P]\!]$ $\not$*must* $[\![o]\!]$.

Proposition 4.8 Let $h = [\![\cdot]\!] : \mathrm{A}\pi \to \mathrm{POA}\pi$ as in Definition 4.7. $\forall P \in \mathrm{A}\pi, \forall o \in \mathcal{O}$, P *sat* o if and only if $[\![P]\!]$ *sat* $[\![o]\!]$, where *sat* can be respectively *may* and *fair*.

5 Uniform impossibility results for persistence

This section is the core of the paper and it focuses on general and uniform negative results for encodings of Aπ into PIAπ, POAπ and PAπ, respectively. We identify some reasonable conditions which will guarantee that none of these encodings can be must-preserving. In particular, we show that there does not exist a must-preserving compositional encoding, homomorphic wrt replication, from π-calculus into any semi-persistent calculus. The proofs mainly rely on the following statement: if $[\![\cdot]\!]$ is an encoding from Aπ into $\mathcal{P}$ satisfying (1) compositionality w.r.t. input prefix, (2) must-preservation and (3) $[\![\omega.0]\!] \xrightarrow{\omega}$ then $\forall x, y \in \mathcal{N}$, any hole is prefixed in $C^{[\![\cdot]\!]}_{x(y)}$.

We believe that the hypothesis $[\![\omega.0]\!] \xrightarrow{\omega}$ is reasonable for an encoding. It can follow from the existence of a divergent process in the range of the encoding, which is necessary if the encoding preserves divergence—recall that P *diverges*, $P\uparrow$, if there is an infinite sequence of reductions from P. However, the hypothesis $[\![\omega.0]\!] \xrightarrow{\omega}$ can be also obtained in a purely syntactic way, i.e without divergence assumption, defining $[\![\omega.P]\!] = \omega.[\![P]\!]$.

Theorem 5.1 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P}$, with $\mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$, be an encoding satisfying:

1. compositionality w.r.t. input prefix, parallelism and replication,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$,
3. $\exists x, y, z :\ n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_{x(y)}) = n(C^{[\![\cdot]\!]}_!) \cap n([\![\bar{x}z]\!]) = n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_|) = \emptyset$,
4. $C^{[\![\cdot]\!]}_!$ is a singularly-structured context.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. (Sketch of:) Suppose that $[\![\cdot]\!]$ in $C^{[\![\cdot]\!]}_!$ is not in the scope of a replication. Then it is possible to prove that the hole is prefixed in $C^{[\![\cdot]\!]}_!$. Now it suffices to consider that $x(y).0$ *must* $!\omega.0$ but $C_{x(y)}[[\![0]\!]]$ $\not$*must* $C_!{[}[\![\omega.0]\!]]$, since every hole is prefixed in $C^{[\![\cdot]\!]}_{x(y)}$, the hole is prefixed in $C^{[\![\cdot]\!]}_!$ and $C_{x(y)}[[\![0]\!]] \mid C_![[\![\omega.0]\!]] \not\xrightarrow{}$ by (3).

Now suppose that $[\![\cdot]\!]$ in $C^{[\![\cdot]\!]}_!$ is in the scope of a replication. Then it is possible to prove that $\forall x', z' \in \mathcal{N}$, either $C_![C_{x(y)}[[\![\omega.0]\!]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$ or $C_{x(y)}[[\![\omega.0]\!]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$ has at least one infinite computation such that $[\![\omega.0]\!]$ does not interact or participate in the computation. Now it suffices to consider both $P \mid o$ (with $[\![P]\!] \mid [\![o]\!]$) and $P' \mid o'$ (with $[\![P']\!] \mid [\![o']\!]$), where $P = !x(y).x'(y').\omega.0$, $o = \bar{x}z \mid \bar{x}'z'$ $(x \neq x')$, $P' =$

$x(y).x'(y').\omega.0$ and $o' = !(\bar{x}z \mid \bar{x}'z')$, obtaining that $[\![\cdot]\!]$ cannot be must-preserving. □

Let us discuss the premises in the above theorem. Compositionality is in general a reasonable condition for an encoding. As argued above, the second condition is validated if the encoding is to preserve divergence. The third condition is validated if in the encoding of each operator *op* the context where the encodings of the operands are placed, i.e. C_{op}, uses unique names only. Replication represents an infinite parallel composition, so it is arguably reasonable to require homomorphism for replication since homomorphism for the parallel operator is arguably a reasonable requirement—see Remark 4.2. Regarding (4), we already pointed out in Remark 4.3 that in compositional encodings the contexts C_{op} are typically singularly-structured [7] .

We conclude this section with a theorem stating a general and uniform impossibility result for the existence of encodings from Aπ into any (semi-)persistent calculus. The statement results as an immediate consequence of Theorem 5.1 in the case of homomorphism w.r.t replication, as it implies $n(C_!^{[\![\cdot]\!]}) = \emptyset$.

Theorem 5.2 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P}$, with $\mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$, be an encoding satisfying:

1. compositionality w.r.t. input prefix and parallelism,
2. homomorphism w.r.t replication,
3. $[\![\omega.0]\!] \xrightarrow{\omega}$.

Then $[\![\cdot]\!]$ is not must-preserving.

6 Specialized impossibility results for persistence

In the previous section we gave a uniform impossibility result for the existence of encodings of Aπ into the (semi-)persistent calculi. In this section, we give further impossibility results, under different hypotheses, taking into account particular features of some of the (semi-)persistent calculi, namely PAπ and PIAπ [8] .

For technical reasons we introduce a particular kind of contexts in $\mathcal{P}$ that differ from those we have introduced in Section 4, in that brackets do not disappear once we "fill the holes" with process terms.

Definition 6.1 [Focusing contexts] A *focusing context* $C\{\,\}$ for $\mathcal{P}$ is generated by the following grammar:

$$C\{\,\} := \ \{\,\}\sigma \ \Big|\ 0 \ \Big|\ out \ \Big|\ in.C\{\,\} \ \Big|\ (\nu x)C\{\,\} \ \Big|\ C\{\,\} \mid C\{\,\} \ \Big|\ !C\{\,\}$$

where σ is a (name) substitution, and *in* and *out* are resp. input and output, according to $\mathcal{P}$ syntax. (e.g. $in = !x(y)$, and either $out = \bar{x}z$ if $\mathcal{P} = \mathrm{PIA}\pi$ or $out = !\bar{x}z$ if $\mathcal{P} = \mathrm{PA}\pi$)

[7] Notice that the case $[\![!P]\!] = [\![P]\!] \mid ![\![P]\!]$, where $C_! = [.] \mid ![.]$ is not singularly-structured, can be rewritten via $\equiv$ as $[\![!P]\!] = ![\![P]\!]$, where the corresponding $C_! = ![\cdot]$ is singularly-structured.

[8] We also stated this kind of specialized result for POAπ but for reasons of space and its restricted nature it has been moved in the appendix

Notation 6.1 *Given a focusing context $C\{\}$ and $P \in \mathcal{P}$, $C\{P\}$ is the term obtained by replacing each occurrence $\{\ \}\sigma$ in $C\{\ \}$ by $\{P\}\sigma$. We denote by $\mathcal{L}(P)$ (ranged over by $B, B', ..$) the set $\{C\{P\} \mid P \in \mathcal{P}, C\{\ \}$ is a focusing context$\}$.*

An occurrence of $\{P\}\sigma$ is prefixed in $B \in \mathcal{L}(P)$ if it is in the scope of an input prefix. We write $Pref(B)$ when every occurrence of $\{P\}\sigma$ is prefixed in B.

The structural congruence and the reduction semantics for the language $\mathcal{L}(P)$ are both defined on the basis of the ones for $\mathcal{P}$, the only difference being that terms are in $\mathcal{L}(P)$ instead than in $\mathcal{P}$ and that unguarded braces (i.e. terms out of the scope of an input prefix like $\{P\}\sigma$) are assumed as deadlocked terms. This is not a concern, because for the proof of our main results, for every σ each occurrence of $\{P\}\sigma$ is prefixed, i.e. in the scope of an input prefix.

It is possible to prove that $\mathcal{L}(P)$ is closed under substitution and, as a consequence, under reduction. Denoting by $Unbrace(B)$ the $\mathcal{P}$ process obtained by removing all the braces from B and by applying the substitutions, it is also possible to prove that: (i) $B \in \mathcal{L}(P)$, then $B \longrightarrow B'$ implies $B' \in \mathcal{L}(P)$ and $Unbrace(B) \longrightarrow Unbrace(B')$, and (ii) $Pref(B)$ and $Unbrace(B) \longrightarrow R$ implies that $\exists B' \in \mathcal{L}(P)$ such that $B \longrightarrow B'$ and $R \equiv Unbrace(B')$.

Focusing contexts are extended for the testing machinery, adding rule $\{\omega.E\}\sigma \xrightarrow{\omega}$ in Table 3. Notice that, since every σ is defined over $\mathcal{N}$ and $\omega \notin \mathcal{N}$, then $\forall E \in \mathcal{P}$ and $B \in \mathcal{L}(P)$, (i) $\{\omega.E\}\sigma \xrightarrow{\omega}$; (ii) $B \xrightarrow{\omega}$ implies $B\sigma \xrightarrow{\omega}$; (iii) $B \xrightarrow{\omega}$ if and only if $Unbrace(B) \xrightarrow{\omega}$, where $B\sigma$ represents the result of the application of σ to B (assuming to use α-equivalence to avoid collision of names).

Persistent calculus: To prove our main results, we define a function over $\mathcal{L}(P)$, $min(B)$ (Table 4), and a predicate, Pr (Table 5).

$min(B) = +\infty$ if $B \in \mathcal{P}$;	$min((\nu x)B) = min(B)$;
$min(\{P\}) = 0$;	$min(B \mid B') = \mathbf{min}\{min(B), min(B')\}$;
$min(x(y).B) = 1 + min(B)$;	$min(!B) = min(B)$.

Table 4
Function *min*.

$$\text{Red}\ \frac{min(!x(y).B) \geq 2}{Pr(!\bar{x}z \mid !x(y).B)} \qquad \text{Res}\ \frac{Pr(B)}{Pr((\nu y)B)}$$

$$\text{Par}\ \frac{Pr(B_1)}{Pr(B_1 \mid B_2)} \qquad \text{Cong}\ \frac{Pr(B')\ ,\ B' \equiv B}{Pr(B)}$$

Table 5
Predicate *Pr*.

We can prove that Pr is closed under reduction and it implies *Pref*. As a consequence, for every $B \in \mathcal{L}(P)$ such that $Pr(B)$, it is possible to build a non-empty

maximal computation from B where any term of the computation verifies the predicate Pr. We can now state a rather strong negative result for PAπ.

Theorem 6.2 Let $[\![\cdot]\!]$ be an encoding from Aπ into PAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. By contradiction, it suffices to suppose $[\![\cdot]\!]$ being must-preserving, consider $P = \bar{x}z|\bar{x}z$ and $o = x(y).x(y).\omega.0.$ and observe that $Pr([\![\bar{x}z|\bar{x}z]\!]|C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}])$ holds. Hence, it is possible to prove that there is a non-empty maximal computation from $[\![\bar{x}z \mid \bar{x}z]\!] \,|[\![x(y).x(y).\omega.0]\!]$ where any term of the computation verifies the predicate Pr, i.e. every term does not perform ω (since every occurrence of $[\![\omega.0]\!]$ is prefixed). □

The above theorem resembles the impossibility result in [24] about the existence of an encoding from Aπ into PAπ wrt weak bisimulation (and output equivalence). However, the hypothesis of the result in [24] is different. Namely, it is restricted to encodings homomorphic wrt parallelism.

Persistent-input calculus: Regarding PIAπ (and POAπ), a Pr-like predicate does not preserve *Pref* (it suffices to consider $B_1 = \bar{b} \mid \bar{c} \mid !b.!c.\{P\}\sigma$, where $P \in$ PIAπ, and $B_2 = !\bar{b} \mid !\bar{c} \mid b.c.\{P\}\sigma$, where $P \in$ POAπ). In the case of PIAπ, an ad-hoc predicate, Pr_{in}, is defined. The predicate selects those processes $B \in \mathcal{L}(P)$ such that - every $\{P\}\sigma$ occurrence is in the scope of an input prefix $x(y)$, for some $x \in fn(B)$ and $y \in \mathcal{N}$, - there exists an input component $!x(y).B$ (prefixing $\{P\}\sigma$) such that $min(!x(y).B) \geq 2$, - every parallel component $!x_i(y).B$ is such that $min(!x_i(y).B) \geq 1$ if $x_i = x$ and $min(!x_i(y).B) \geq 2$ if $x_i \neq x$. The results for Pr can be proven in a similar way for Pr_{in}. In particular, whenever $\exists x \in fn(B)$ such that $Pr_{in}(B, x)$, it is possible to build a maximal computation from B where any term of the computation verifies the predicate Pr_{in}. Hence, it leads us to the negative result below.

Theorem 6.3 Let $[\![\cdot]\!]$ be an encoding from Aπ into PIAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$,
3. if $fn(P) \cap bn(x(y)) = \emptyset$ then $fn([\![P]\!]) \cap bn(C^{[\![\cdot]\!]}_{x(y)}) = \emptyset$,
4. $[\![x(y).P]\!] \equiv (\nu x_1)..(\nu x_n)(!u(v).C[[\![P]\!]] \mid T)$, for some $x_1, .., x_n, C, T$ with $u \neq x_i$ for any i.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. It is possible to prove that $\exists h \in fn(C^{[\![\cdot]\!]}_{x(y)})$: $Pr_{in}(C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}], h)$. Now, it suffices to assume, by contradiction, $[\![\cdot]\!]$ being must-preserving and proving that $Pr_{in}([\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}], h)$ holds. Hence, it is possible to prove that there is a non-empty maximal computation from $[\![\bar{x}z \mid \bar{x}z]\!] \mid [\![x(y).x(y).\omega.0]\!]$ where

any term of the computation verifies the predicate Pr_{in}, i.e. every term does not perform ω (since every occurrence of $[\![\omega.0]\!]$ is prefixed). □

Notice that the encoding in Definition 4.5 satisfies every condition of the following theorem and, more important, that Pr_{in} does not rely on any divergence assumption, differently from Pr. We have already argued for the first two conditions as being reasonable. Intuitively, the third condition expresses that a *non-binding* property wrt input prefix: if in a source term $x(y).P$ none of the free names of P is bound by the input prefix, then the free names of $[\![P]\!]$ must not be bound either (by a binder in the context where $[\![P]\!]$ is placed) in the encoding of $[\![x(y).P]\!]$. Finally, the fourth condition basically expresses that Aπ inputs should be mapped into PIAπ inputs possibly allowing some other material around it. This is validated, e.g., by encodings that preserve input/output polarities—i.e. Aπ inputs/outputs must be mapped into PIAπ input/outputs[9].

7 Related work and concluding remarks

Most of the related work was discussed in the introduction. In a different context, in [22] it is shown that the separate choice encoding of the π-calculus into the asynchronous π-calculus is faithful with respect to weak bisimulation, while in [8] the authors prove that no must-preserving encoding of the (choiceless) synchronous pi-calculus into the asynchronous one exists. Hence must semantics is a good candidate to study the expressiveness of persistence when divergence is taken into account. Nevertheless, differently from [8], this work does not consider any synchronous language, i.e. the must semantics is studied in a uniform and purely asynchronous framework. As previously mentioned the study of persistence in [24] is incomplete as ignores the crucial issue of divergence. In this paper, we used the divergence-sensitive framework of testing semantics and adapted and exploited the techniques of [8] to give a more complete account of the expressiveness of persistence in asynchronous calculi. In particular, as discussed in the introduction, this work supports informal expressiveness loss claims in persistent asynchronous languages [3,14,11].

References

[1] J. Alves-Foss. An Efficient Secure Authenticated Group Key Exchange Algorithm for Large and Dynamic Groups. In *Proceedings of the 23rd National Information Systems Security Conference*, 2000.

[2] R. Amadio and D. Lugiez and V. Vanackere. On the Symbolic Reduction of Processes with Cryptographic Functions. *TCS: Theoretical Computer Science* 290, 2003.

[3] E. Best, F. de Boer, and C. Palamidessi. Partial order and sos semantics for linear constraint programs. *In Proc. of Coordination'97*, volume 1282 of LNCS, 1997.

[4] B. Blanchet. From linear to classical logic by abstract interpretation. *Information Processing Letters* 95(5), 2005.

[5] M. Boreale and M. Buscemi. A Framework for the Analysis of Security Protocols, *Lecture Notes in Computer Science* 2421, 2002.

[9] E.g., the encoding in Definition 4.5 satisfies all conditions of Theorem 6.3.

[6] E. Brinksma, A. Rensink, W. Vogler. Fair Testing, *Proc. of CONCUR'95, LNCS* 962, pp. 313-327, 1995.

[7] D. Cacciagrano, F. Corradini. On Synchronous and Asynchronous Commu- nication Paradigms, *Proc. of ICTCS '01, LNCS* 2202, pp. 256-268, 2001.

[8] D. Cacciagrano, F. Corradini, C. Palamidessi. Separation of Synchronous and Asynchronous Communication Via Testing. *Proc. of EXPRESS'05.* Electr. Notes Theor. Comput. Sci. 154(3): 95-108, 2006. An extended version will appear in *Theoretical Computer Science.*

[9] M. Carbone, S. Maffeis. On the Expressive Power of Polyadic Synchronisation in pi-calculus. *Nord. J. Comput.* 10(2): 70-98, 2003.

[10] I. Castellani, M. Hennessy. Testing Theories for Asynchronous Languages, *Proc. of FSTTCS '98, LNCS* 1530, pp. 90-101, 1998.

[11] F. Crazzolara. Language, Semantics, and Methods for Security Protocols. *PhD Dissertation*, University of Aarhus, Denmark, 2003.

[12] F. Crazzolara and G. Winskel. Events in security protocols, *Proceedings of the 8th ACM Conference on Computer and Communications Security*, ACM Press, 2001.

[13] R. De Nicola, M. Hennessy. Testing Equivalence for Processes, *Theoretical Computer Science* 34, pp. 83-133, 1984.

[14] F. Fages, P. Ruet, and S. Soliman. Linear concurrent constraint programming: operational and phase semantics. *Information and Computation*, 2001.

[15] M. Fiore and M. Abadi. Computing symbolic models for verifying cryptographic protocols. *Proc. CSFW-14. IEEE*, 2001.

[16] D. Gorla: On the Relative Expressive Power of Asynchronous Communication Primitives. *FoSSaCS 2006*, 47-62, 2006.

[17] D. Gorla: Synchrony vs Asynchrony in Communication Primitives *Proc. of EXPRESS'06*, 47-62, 2006.

[18] S. Maffeis and I. Phillips. On the computational strength of pure ambient calculi. *Proc. of EXPRESS '03*, 2003.

[19] R. Milner. *Communication and Concurrency*, Prentice-Hall International, 1989.

[20] R. Milner, J. Parrow, D. Walker. A Calculus of Mobile Processes, Part I and II, *Information and Computation* 100, pp. 1-78, 1992.

[21] M. Merro, D. Sangiorgi. On asynchrony in name-passing calculi, *Proc. of ICALP '98, LNCS* 1443, 1998.

[22] U. Nestmann. What is a 'Good' Encoding of Guarded Choice?, *Information and Computation* 156, pp. 287-319, 2000.

[23] C. Palamidessi. Comparing the Expressive Power of the Synchronous and Asynchronous π-calculus, *Mathematical Structures in Computer Science* 13(5), pp. 685-719, 2003. A preliminary version appeared in the proceedings of POPL '97.

[24] C. Palamidessi, V. Saraswat, F. Valencia and B. Victor. On the Expressiveness of Linearity vs Persistence in the Asynchronous Pi Calculus. *LICS 2006:59-68*, 2006.

[25] L. C. Paulson. Mechanized proofs for a recursive authentication protocol. In *10th Computer Security Foundations Workshop*, 1997.

[26] I. Phillips and M. Vigliotti Electoral Systems in Ambient Calculi. *FoSSaCS'04.* 2004.

[27] I. Phillips, M. Vigliotti. Leader Election in Rings of Ambient Processes. *Electr. Notes Theor. Comput. Sci.* 128(2): 185-199, 2005.

[28] K.V.S. Prasad. Broadcast Calculus Interpreted in CCS up to Bisimulation. *In Proceedings of Express'01*, volume 52 of Electronic Notes in Theoretical Computer Science, pages 83-100. Elsevier, 2002.

[29] D. Sangiorgi. The name discipline of uniform receptiveness. *Theoretical Computer Science*, 221(12):457493, 1999.

[30] D. Sangiorgi and D. Walker. *The π-calculus: A Theory of Mobile Processes.* Cambridge University Press, 2001.

[31] V. Saraswat and P. Lincoln. Higher-order linear concurrent constraint programming. *Technical report, Xerox PARC*, 1992.

[32] V. Saraswat. Concurrent Constraint Programming. The MIT Press, 1993.

[33] M. Vigliotti, I. Phillips, C. Palamidessi. Separation Results Via Leader Election Problems. *FMCO 2005*, 172-194, 2005.

A Appendix

In this section, we give the definitions and the proofs omitted in Section 4.

Definition A.1 Define $P \downarrow_{\bar{x}}$ iff $\exists z_1, \ldots, z_n, y, R : P \equiv (\nu z_1)..(\nu z_n)(\bar{x}y \mid R)$ and $\forall i \in [1..n], x \neq z_i$. Furthermore, $P \Downarrow_{\bar{x}}$ iff $\exists Q : P \stackrel{*}{\longrightarrow} Q \downarrow_{\bar{x}}$.

Definition A.2 (*Barbed Bisimilarity, Barbed Congruence*) A *weak barbed bisimulation* is a symmetric relation $\mathcal{R}$ satisfying the following: $(P, Q) \in \mathcal{R}$ implies that:

(i) $P \longrightarrow P'$ then $\exists Q' : Q \stackrel{*}{\longrightarrow} Q' \wedge (P', Q') \in \mathcal{R}$.

(ii) $P \downarrow_{\bar{x}}$ then $Q \Downarrow_{\bar{x}}$.

We say that P and Q are *weak barbed bisimilar*, written $P \dot{\approx} Q$, iff $(P, Q) \in \mathcal{R}$ for some weak barbed bisimulation $\mathcal{R}$. Furthermore, *weak barbed congruence* $\approx$ is defined as: $P \approx Q$ iff for every process context $C[\cdot]$, $C[P] \dot{\approx} C[Q]$.

Proposition 4.6 $\forall\, P \in \mathrm{A}\pi$, $\forall\, o \in \mathcal{O} \subseteq \mathrm{PIA}\pi$, P *sat* o if and only if $[\![P]\!]$ *sat* o, where *sat* can be respectively *may* and *fair*.

Proof. $P \approx [\![P]\!]$ implies that $\forall\, o \in \mathcal{O} \subseteq \mathrm{PIA}\pi$, $P \mid o \approx [\![P]\!] \mid o$. Extending the notion of barb to ω, we have $T \Downarrow_\omega$ iff $T \stackrel{\omega}{\Longrightarrow}$. Suppose P *fair* o. Then for every maximal computation $P \mid o = E_0 \longrightarrow E_1 \longrightarrow .. \longrightarrow E_i \,[\longrightarrow \ldots]$ we have $E_i \stackrel{*}{\longrightarrow} E_i' \downarrow_\omega$, for every $i \geq 0$. Since $P \mid o \approx [\![P]\!] \mid o$, then for every maximal computation $[\![P]\!] \mid o = A_0 \longrightarrow A_1 \longrightarrow .. \longrightarrow A_i \,[\longrightarrow \ldots]\; A_i \Downarrow_\omega$, for every $i \geq 0$. I.e. $[\![P]\!]$ *fair* o. Notice that *may* is a special case of *fair*: P *may* o implies $P \mid o \stackrel{*}{\longrightarrow} E_0' \downarrow_\omega$ and, since $P \mid o \approx [\![P]\!] \mid o$, it implies that $[\![P]\!] \mid o \Downarrow_\omega$, i.e. $[\![P]\!]$ *may* o.□

B Appendix

In this section, we give the proofs omitted in Section 5. We will use $\langle P \rangle$ to denote some restricted version of P, i.e. any process of the form $(\nu x_1)..(\nu x_n)P$, for some $x_1, \ldots, x_n \in fn(P)$.

Proposition B.1 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P}$, with $P \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$, be an encoding satisfying:

1. must-preservation,
2. $\exists P \in \mathrm{A}\pi$ such that $[\![P]\!] \uparrow$.

Then $[\![\omega.0]\!] \stackrel{\omega}{\longrightarrow}$.

Proof. Let $P \in \mathrm{A}\pi$ such that $[\![P]\!] \uparrow$. Since P *must* $\omega.0$ and the encoding $[\![\cdot]\!]$ is *must*-preserving, then $[\![P]\!]$ *must* $[\![\omega.0]\!]$. Since $[\![P]\!] \uparrow$, we have $[\![\omega.0]\!] \stackrel{\omega}{\longrightarrow}$. □

Lemma B.2 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$ be an encoding satisfying:

1. compositionality w.r.t. input prefix,
2. must-preservation,

3. $[\![\omega.0]\!] \xrightarrow{\omega}$.

Then $\forall x, y \in \mathcal{N}$, every hole is prefixed in $C^{[\![\cdot]\!]}_{x(y)}$.

Proof. By definition we have $0 \not\ must\ x(y).\omega.0$, and since $[\![\cdot]\!]$ is must-preserving, we have $[\![0]\!] \not\ must\ [\![x(y).\omega.0]\!]$. Hence, $[\![0]\!] \not\ must\ C_{x(y)}[[\![\omega.0]\!]]$. Since $[\![\omega.0]\!] \xrightarrow{\omega}$ by hypothesis, every occurrence of $[\![\omega.0]\!]$ has to be prefixed in $C^{[\![\cdot]\!]}_{x(y)}$. □

The following two technical lemmas are used for proving our main results.

Lemma B.3 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$ be an encoding satisfying:

1. compositionality w.r.t. input prefix and replication,
2. must-preservation,
3. $[\![\omega.0]\!] \xrightarrow{\omega}$,
4. $\exists x, y, z : n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_{x(y)}) = n(C^{[\![\cdot]\!]}_!) \cap n([\![\bar{x}z]\!]) = \emptyset$,
5. $C^{[\![\cdot]\!]}_!$ is a singularly-structured context,
6. the hole in the context $C^{[\![\cdot]\!]}_!$ is not in the scope of a replication.

Then the hole is prefixed in $C^{[\![\cdot]\!]}_!$.

Proof. Since $\bar{x}z \not\ must\ x(y).x(y).\omega.0$, $!\bar{x}z\ must\ x(y).x(y).\omega.0$ and $[\![\cdot]\!]$ is must-preserving, we have $[\![\bar{x}z]\!] \not\ must\ [\![x(y).x(y).\omega.0]\!]$ and $[\![!\bar{x}z]\!]\ must\ [\![x(y).x(y).\omega.0]\!]$. Since there is an unsuccessful maximal computation from $[\![\bar{x}z]\!] \mid [\![x(y).x(y).\omega.0]\!]$, then there is an unsuccessful maximal computation from $([\![\bar{x}z]\!] \mid [\![x(y).x(y).\omega.0]\!])\alpha$, where α denotes α-equivalence.

By contradiction, suppose that the hole is not prefixed in $C^{[\![\cdot]\!]}_!$. Then $[\![\bar{x}z]\!]$ is not prefixed in $C_![[\![\bar{x}z]\!]] \mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]$. Since every hole is prefixed in $C^{[\![\cdot]\!]}_{x(y)}$ and the hole is not in the scope of a replication in $C^{[\![\cdot]\!]}_!$, we can prove, by induction on the structure of $C^{[\![\cdot]\!]}_!$, that $\exists B \in \mathcal{P}$ such that $C_![[\![\bar{x}z]\!]] \mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]$ is congruent to $B = \langle T | [\![\bar{x}z]\!] | C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]\rangle$, where $bn(C_![[\![\bar{x}z]\!]|C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]) = bn(B)$ and $T \in \mathcal{P}$ (without loss of generality, we use the same notation for $C_![[\![\bar{x}z]\!]] \mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]$ before and after applying α-equivalence). Recall that $[\![\bar{x}z]\!]$, $C^{[\![\cdot]\!]}_{x(y)}$ and T do not contain ω. Now, consider the following (unsuccessful) maximal computation from $[\![\bar{x}z]\!] \mid C_{x(y)}[C_x(y)[[\![\omega.0]\!]]]$ (there exists at least one):

$$[\![\bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]] = A_0 \longrightarrow A_1 \longrightarrow \ldots \longrightarrow A_i[\longrightarrow \ldots]$$

where $\forall i \geq 0, A_i \not\xrightarrow{\omega}$.

- If this computation is infinite: then there exists an unsuccessful maximal computation from B, i.e. from $[\![!\bar{x}z]\!] \mid [\![x(y).x(y).\omega.0]\!]$, contradicting the hypothesis.
- If this computation is finite: then $B \xrightarrow{*} \langle T \mid A_i\rangle$, where $A_i \not\longrightarrow$ and $A_i \not\xrightarrow{\omega}$.
  If $T \uparrow$, again there exists an unsuccessful maximal computation from B, i.e. from $[\![!\bar{x}z]\!] \mid [\![x(y).x(y).\omega.0]\!]$, contradicting the hypothesis.

Otherwise, $T \xrightarrow{*} D$, i.e $B \xrightarrow{*} \langle D|\, A_i\rangle$, where $D \not\longrightarrow$, $A_i \not\longrightarrow$, $D \not\xrightarrow{\omega}$ and $A_i \not\xrightarrow{\omega}$.

Since $\equiv$ does not change free names, $fn(C_!{[}[\![\bar{x}z]\!]\mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]) = fn(B)$. Since B is such that $bn(C_!{[}[\![\bar{x}z]\!]\mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]) = bn(B)$, we have $n(C_!{[}[\![\bar{x}z]\!]\mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]]) = n(B)$. Furthermore, $fn(A_i) \subseteq (n([\![\bar{x}z]\!]\,|\,C_{x(y)}[C_{x(y)}[\cdot]])\cup n([\![\omega.0]\!])) \subseteq (n([\![\bar{x}z]\!])\cup n(C^{[\![\cdot]\!]}_{x(y)})\cup n([\![\omega.0]\!]))$ and $fn(D) \subseteq n(T) \subseteq n(C^{[\![\cdot]\!]}_!)$.

By initial hypothesis, every occurrence of $[\![\omega.0]\!]$ is prefixed in A_i and $n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_{x(y)}) = n(C^{[\![\cdot]\!]}_!) \cap n([\![\bar{x}z]\!]) = \emptyset$. It follows that $\langle D|\, A_i\rangle \not\longrightarrow$. Since $\langle D|\, A_i\rangle \not\xrightarrow{\omega}$, we contradict that $[\![!\bar{x}z]\!]$ *must* $[\![x(y).x(y).\omega.0]\!]$.

□

Lemma B.4 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$ be an encoding satisfying:

1. compositionality w.r.t. input prefix and replication,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$,
3. $\exists x, y, z : n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_{x(y)}) = n(C^{[\![\cdot]\!]}_!) \cap n([\![\bar{x}z]\!]) = \emptyset$,
4. $C^{[\![\cdot]\!]}_!$ is a singularly-structured context,
5. the hole in the context $C^{[\![\cdot]\!]}_!$ is not in the scope of a replication.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. By contradiction, suppose $[\![\cdot]\!]$ is *must*-preserving. Then $x(y).0$ *must* $!\omega.0$. Consider $C_{x(y)}[[\![0]\!]]\,|\,C_!{[}[\![\omega.0]\!]$: since every hole is prefixed in $C_{x(y)}$, the hole is prefixed in $C^{[\![\cdot]\!]}_!$ and $C_{x(y)}[[\![0]\!]] \mid C_!{[}[\![\omega.0]\!]] \not\longrightarrow$ by (3), we have $C_{x(y)}[[\![0]\!]]$ *must* $C_!{[}[\![\omega.0]\!]$. □

Lemma B.5 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$ be an encoding satisfying:

1. compositionality w.r.t. input prefix, parallelism and replication,
2. must-preservation,
3. $[\![\omega.0]\!] \xrightarrow{\omega}$,
4. $\exists x, y, z : n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_{x(y)}) = n(C^{[\![\cdot]\!]}_!) \cap n([\![\bar{x}z]\!]) = n(C^{[\![\cdot]\!]}_!) \cap n(C^{[\![\cdot]\!]}_|) = \emptyset$,
5. $C^{[\![\cdot]\!]}_!$ is a singularly-structured context,
6. the hole in the context $C^{[\![\cdot]\!]}_!$ is in the scope of a replication.

Then $\forall x', z' \in \mathcal{N}$, either $C_!{[}C_{x(y)}[[\![\omega.0]\!]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$ or $C_{x(y)}[[\![\omega.0]\!]] \mid C_!{[}[\![\bar{x}z \mid \bar{x}'z']\!]]$ has at least one infinite computation such that $[\![\omega.0]\!]$ does not interact or participate in the computation.

Proof. Let's assume, by contradiction, that both $C_!{[}C_{x(y)}[[\![\omega.0]\!]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$ and $C_{x(y)}[[\![\omega.0]\!]] \mid C_!{[}[\![\bar{x}z \mid \bar{x}'z']\!]]$ do not have infinite computations where $[\![\omega.0]\!]$ interacts. Then $C_!{[}C_{x(y)}[\omega.0]]$, $[\![\bar{x}z \mid \bar{x}'z']\!]$, $C_{x(y)}[[\![\omega.0]\!]]$ and $C_!{[}[\![\bar{x}z \mid \bar{x}'z']\!]]$ do not have infinite computations where $[\![\omega.0]\!]$ don't interact.

By must-preservation, $[\![!x(y).\omega.0]\!]$ *must* $[\![\bar{x}z \mid \bar{x}'z']\!]$, where $[\![\omega.0]\!]$ is prefixed in

$C_{x(y)}[[\omega.0]]]$. From (2) and (5) we can show that $[\![!\omega.0]\!] \longrightarrow .. \longrightarrow \langle !\langle [\![\omega.0]\!]|P\rangle|Q\rangle$.

From (1) we know that $\forall U$, there is at least one computation such that $[\![!U]\!] \longrightarrow .. \longrightarrow \langle !\langle [\![U]\!]|P\rangle|Q\rangle$. From the above and considering $U = C_{x(y)}[[\omega.0]]$, there is at least one computation

$$[\![!x(y).\omega.0]\!] \;=\; C_!\big[C_{x(y)}[[\omega.0]]\big] \;\longrightarrow\; \ldots \;\longrightarrow\; \langle !\langle C_{x(y)}[[\omega.0]] \mid P\rangle \mid Q\rangle$$

where $\langle C_{x(y)}[[\omega.0]] \mid P\rangle \not\longrightarrow$ (otherwise, $!\langle C_{x(y)}[[\omega.0]] \mid P\rangle$ diverges without intervention from $[\![\omega.0]\!]$, as $[\![\omega.0]\!]$ is prefixed in $C^{[\![\cdot]\!]}_{x(y)}$).

By (4), we know that $\langle !\langle C_{x(y)}[[\omega.0]] \mid P\rangle \mid Q\rangle \equiv !C_{x(y)}[[\omega.0]] \mid \langle P \mid Q\rangle$. We also know that $[\![\bar{x}z \mid \bar{x}'z']\!] \not\longrightarrow$, (otherwise $C_![[\![\bar{x}z \mid \bar{x}'z']\!]$ would be divergent without the intervention from $[\![\omega.0]\!]$). From the above, we have at least one computation

$$C_!\big[C_{x(y)}[[\omega.0]]\big] \mid [\![\bar{x}z \mid \bar{x}'z']\!] \;\longrightarrow\; \ldots \;\longrightarrow\; !C_{x(y)}[[\omega.0]] \mid \langle P \mid Q\rangle \mid [\![\bar{x}z \mid \bar{x}'z']\!]$$

where $C_{x(y)}[[\omega.0]] \not\longrightarrow$, $[\![\bar{x}z \mid \bar{x}'z']\!] \not\longrightarrow$ and $S = \langle P \mid Q\rangle \not\xrightarrow{\omega}$ (as $[\![\omega.0]\!]$ is prefixed in any possible occurrence of $C^{[\![\cdot]\!]}_{x(y)}$ in $\langle P \mid Q\rangle$).

As $C_{x(y)}[[\omega.0]] \not\longrightarrow$, $[\![\bar{x}z \mid \bar{x}'z']\!] \not\longrightarrow$, $C_{x(y)}[[\omega.0]] \not\xrightarrow{\omega}$, $[\![\bar{x}z \mid \bar{x}'z']\!] \not\xrightarrow{\omega}$ and by must-preservation we know that $C_{x(y)}[[\omega.0]]$ *must* $[\![\bar{x}z \mid \bar{x}'z']\!]$, then there must be at least one interaction between $C_{x(y)}[[\omega.0]]$ and $[\![\bar{x}z \mid \bar{x}'z']\!]$. By compositionality w.r.t bang and input prefix, the structure of $C^{[\![\cdot]\!]}_{x(y)}$ and $[\![\bar{x}z \mid \bar{x}'z']\!]$ can be characterized in one of the following ones:

1. either $C_{x(y)}[\cdot] \equiv \langle h(k).P'|Q'\rangle$ or $\equiv \langle !h(k).P'|Q'\rangle$, and either $[\![\bar{x}z|\bar{x}'z']\!] \equiv \langle \bar{h}k|R'\rangle$ or $\equiv \langle !\bar{h}k|R'\rangle$.

   $\mathcal{P}$ = POAπ: then $!C_{x(y)}[[\omega.0]] \mid [\![\bar{x}z \mid \bar{x}'z']\!] \uparrow$, without intervention from $[\![\omega.0]\!]$. As $C_![C_{x(y)}[[\omega.0]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!] \longrightarrow \ldots \longrightarrow !C_{x(y)}[[\omega.0]] \mid S \mid [\![\bar{x}z \mid \bar{x}'z']\!]$, $C_![C_{x(y)}[[\omega.0]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$ diverges without intervention from $[\![\omega.0]\!]$.

   $\mathcal{P}$ = PIAπ: then $C_{x(y)}[[\omega.0]] \mid ![\![\bar{x}z \mid \bar{x}'z']\!] \uparrow$ without intervention from $[\![\omega.0]\!]$. As $C_{x(y)}[[\omega.0]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]] \longrightarrow \ldots \longrightarrow C_{x(y)}[[\omega.0]] \mid S \mid ![\![\bar{x}z \mid \bar{x}'z']\!]$, $C_{x(y)}[[\omega.0]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$ diverges without intervention from $[\![\omega.0]\!]$.
2. either $C_{x(y)}[\cdot] \equiv \langle \bar{h}k|R'\rangle$ or $\equiv \langle !\bar{h}k|R'\rangle$, and either $[\![\bar{x}z|\bar{x}'z']\!] \equiv \langle h(k).P'|Q'\rangle$ or $\equiv \langle !h(k).P'|Q'\rangle$.

   $\mathcal{P}$ = POAπ: then $C_{x(y)}[[\omega.0]]|![\![\bar{x}z|\bar{x}'z']\!] \uparrow$ without intervention from $[\![\omega.0]\!]$. As $C_{x(y)}[[\omega.0]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]] \longrightarrow \ldots \longrightarrow C_{x(y)}[[\omega.0]] \mid S \mid ![\![\bar{x}z \mid \bar{x}'z']\!]$, $C_{x(y)}[[\omega.0]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$ diverges without intervention from $[\![\omega.0]\!]$.

   $\mathcal{P}$ = PIAπ: then $!C_{x(y)}[[\omega.0]] \mid [\![\bar{x}z \mid \bar{x}'z']\!] \uparrow$, without intervention from $[\![\omega.0]\!]$. As $C_![C_{x(y)}[[\omega.0]]]| [\![\bar{x}z \mid \bar{x}'z']\!] \longrightarrow \ldots \longrightarrow !C_{x(y)}[[\omega.0]] \mid S| [\![\bar{x}z \mid \bar{x}'z']\!]$, $C_![C_{x(y)}[[\omega.0]]]|[\![\bar{x}z \mid \bar{x}'z']\!]$ diverges without intervention from $[\![\omega.0]\!]$.

From the above, we can conclude that for any encoding $[\![\cdot]\!]$ from Aπ into PIAπ or POAπ, either $C_{x(y)}[[\omega.0]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$ or $C_![C_{x(y)}[[\omega.0]]]|[\![\bar{x}z \mid \bar{x}'z']\!]$ diverges without intervention from $[\![\omega.0]\!]$ (or both in the case from Aπ into PAπ). □

Lemma B.6 Let $[\![\cdot]\!] : \mathrm{A}\pi \to \mathcal{P} \in \{\mathrm{PIA}\pi, \mathrm{POA}\pi, \mathrm{PA}\pi\}$ be an encoding satisfying:

1. compositionality w.r.t. input prefix, parallelism and replication,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$,
3. $\exists x, y, z : n(C_!^{[\![\cdot]\!]}) \cap n(C_{x(y)}^{[\![\cdot]\!]}) = n(C_!^{[\![\cdot]\!]}) \cap n([\![\bar{x}z]\!]) = n(C_!^{[\![\cdot]\!]}) \cap n(C_|^{[\![\cdot]\!]}) = \emptyset$,
4. $C_!^{[\![\cdot]\!]}$ is a singularly-structured context,
5. the hole in the context $C_!^{[\![\cdot]\!]}$ is in the scope of a replication.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. Suppose that $[\![\cdot]\!]$ is must-preserving. Consider $P = !x(y).x'(y').\omega.0$, $o = \bar{x}z \mid \bar{x}'z'$ $(x \neq x')$, $P' = x(y).x'(y').\omega.0$ and $o' = !(\bar{x}z \mid \bar{x}'z')$. It is possible to verify that P *must* o and P' *must* o'.

P *must* o implies $[\![P]\!]$ *must* $[\![o]\!]$ by must-preservation. It follows that $C_![C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]]]$ *must* $[\![\bar{x}z \mid \bar{x}'z']\!]$. As $[\![\omega.0]\!]$ is prefixed in $C_{x'(y')}^{[\![\cdot]\!]}$, then in every computation from $C_![C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$, $C_{x'(y')}[[\![\omega.0]\!]]$ must interact to unprefix one occurrence of $[\![\omega.0]\!]$.

P' *must* o' implies $[\![P']\!]$ *must* $[\![o']\!]$ by must-preservation. It follows that $C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]]$ *must* $C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$. As $[\![\omega.0]\!]$ is prefixed in $C_{x'(y')}^{[\![\cdot]\!]}$, then in every computation from $C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$, $C_{x'(y')}[[\![\omega.0]\!]]$ must interact to unprefix one occurrence of $[\![\omega.0]\!]$.

By lemma B.5, $C_![C_{x(y)}[[\![\omega.0]\!]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$ or $C_{x(y)}[[\![\omega.0]\!]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$ has at least one infinite computation such that $[\![\omega.0]\!]$ does not interact or participate in the computation. Applying (1), either $C_![C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]]] \mid [\![\bar{x}z \mid \bar{x}'z']\!]$ or $C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]] \mid C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$ has at least one infinite computation such that $C_{x'(y')}[[\![\omega.0]\!]]$ does not interact or participate in this computation, i.e. either $C_![C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]]]$ *must* $[\![\bar{x}z|\bar{x}'z']\!]$ or $C_{x(y)}[C_{x'(y')}[[\![\omega.0]\!]]]$ *must* $C_![[\![\bar{x}z \mid \bar{x}'z']\!]]$. It contradicts that $[\![\cdot]\!]$ is must-preserving. □

C Appendix

In this section, we give the proofs omitted in Section 6.

A class of calculi with focusing contexts

Proposition C.1 Let $B \in \mathcal{L}(P)$. Then:

i) $B \longrightarrow B'$ implies $B' \in \mathcal{L}(P)$ and $\mathit{Unbrace}(B) \longrightarrow \mathit{Unbrace}(B')$;

ii) $\mathit{Pref}(B)$ and $\mathit{Unbrace}(B) \longrightarrow R$ implies that $\exists B' \in \mathcal{L}(P)$ such that $B \longrightarrow B'$ and $R \equiv \mathit{Unbrace}(B')$.

Proof. First, note that $\mathcal{L}(P)$ is closed under substitution, and that the structural congruence is preserved by *Unbrace*. First we prove item (i). We proceed by induction on the depth of the reduction $B \longrightarrow B'$.

$$\text{Com}\ \ out \mid in.B \longrightarrow \begin{cases} B\{z/y\} \mid in.B & (\text{PIA}\pi)\ \ in = !x(y), out = \bar{x}z \\ out \mid B\{z/y\} & (\text{POA}\pi)\ in = x(y), out = !\bar{x}z \\ B\{z/y\} \mid out \mid in.B & (\text{PA}\pi)\ \ in = !x(y), out = !\bar{x}z \end{cases}$$

$$\text{Par}\ \ \frac{B_1 \longrightarrow B_1'}{B_1 \mid B_2 \longrightarrow B_1' \mid B_2} \qquad \text{Res}\ \ \frac{B \longrightarrow B'}{(\nu x)B \longrightarrow (\nu x)B'}$$

$$\text{Cong}\ \ \frac{B_1 \equiv B_1'\ ,\ B_1' \longrightarrow B_2'\ ,\ B_2' \equiv B_2}{B_1 \longrightarrow B_2}$$

Table C.1
Reduction Rules in $\mathcal{L}(P)$.

- $B \in \mathcal{P}$: the proof is trivial, since $\mathcal{P}$ is closed under $\longrightarrow$ and $\forall P \in \mathcal{P}$, $P = \mathit{Unbrace}(P)$.
- $B = \{P\}\sigma$: this case is not possible, since $\{P\}\sigma \not\longrightarrow$.
- $B = out \mid in.B''$: we consider $in.B'' = x(y).B''$ and $out = \bar{x}z$, since the other combinations can be proven similarly. Then $B \longrightarrow B''\{z/y\}$. We also have $\mathit{Unbrace}(B) = out \mid in.\mathit{Unbrace}(B'') \longrightarrow \mathit{Unbrace}(B''\{z/y\})$.
- Cases $B = (\nu x)B' \longrightarrow (\nu x)B''$ and $B = B_1 \mid B_2 \longrightarrow B_1' \mid B_2$ can be proven by induction hypothesis on B' and on B_1, respectively.
- Case $B \equiv B_1 \longrightarrow B_2$ is trivial, since $\equiv$ is preserved by *Unbrace*.

Now we prove item (ii). We proceed by induction on the depth of the reduction $\mathit{Unbrace}(B) = T \longrightarrow R$, assuming $\mathit{Pref}(B)$.

- $T = out \mid in.T''$: we consider $in.T'' = x(y).T''$ and $out = \bar{x}z$, since the other combinations can be proven similarly. Then $T = out \mid in.T'' \longrightarrow T''\{z/y\}$. Define $B = out \mid in.B''$, where $\mathit{Unbrace}(B'') = T''$. Then $\mathit{Unbrace}(B''\{z/y\}) = T''\{z/y\}$ and $B \longrightarrow B''\{z/y\}$.
- Cases $T = (\nu x)T' \longrightarrow (\nu x)T''$ and $T = T_1 \mid T_2 \longrightarrow T_1' \mid T_2$ can be proven by induction hypothesis.
- Case $T \equiv T_1 \longrightarrow T_2$ is trivial, since $\equiv$ is preserved by *Unbrace*.

□

Lemma C.2 Let $B_1, B_2 \in \mathcal{L}(P)$ such that $B_1 \equiv B_2$. Then $\mathit{min}(B_1) = \mathit{min}(B_2)$.

Proof. Only axiom $!B \equiv B \mid !B$ can look difficult to prove. Other axioms are trivial. If $B \in \mathcal{P}$, $\mathit{min}(!B) = \mathit{min}(B \mid !B) = +\infty$. Suppose $B \notin \mathcal{P}$. Then we have $\mathit{min}(!B) = \mathit{min}(B)$ and $\mathit{min}(B \mid !B) = \mathbf{min}\{\mathit{min}(B), \mathit{min}(!B)\} = \mathit{min}(B)$. □

Negative results for PAπ

Proposition C.3 Let $P \in \text{PA}\pi$, $B \in \mathcal{L}(P)$, such that $\mathit{Pr}(B)$. Then $\exists B' \in \mathcal{L}(P)$ such that $B \longrightarrow B'$ and $\mathit{Pr}(B')$.

Proof. To prove the statement we proceed by induction on the depth of the derivation of $Pr(B)$. We recall that $\mathcal{L}(P)$ is closed under $\longrightarrow$ and that the cases $B \in \mathcal{P}$ and $B = \{P\}\sigma$ are not possible, since $Pr(B)$ implies $Pref(B)$, i.e. $min(B) \in [1.. + \infty)$.

- $B = !\bar{x}z \mid !x(y).B''$, where $min(!x(y).B'') \geq 2$: then $B \longrightarrow B' = B''\{z/y\} \mid B$. Since $Pr(B)$, it follows $Pr(B')$.
- Cases $B = (\nu x)B'$, $B = B_1|B_2$ and $B \equiv B_1$ can be proven by induction hypothesis on B' and on B_1, assuming $Pr(B')$, $B' \longrightarrow B''$ and $Pr(B')$, and $Pr(B_1)$, $B_1 \longrightarrow B_1'$ and $Pr(B_1')$, respectively.

□

Proposition C.4 Let $P \in \mathrm{PA}\pi$, $B \in \mathcal{L}(P)$ such that $Pr(B)$. Then there exists a non-empty maximal computation from B

$$B = B_0 \longrightarrow B_1 \longrightarrow B_2 \longrightarrow \ldots \longrightarrow B_i[\longrightarrow \ldots]$$

such that $\forall i \geq 0, Pref(B_i)$.

Proof. By Proposition C.3, $\exists B_1 \in \mathcal{L}(P)$ such that $B \longrightarrow B_1$ and $Pr(B_1)$. Now it suffices to iterate, noticing that $\forall i \geq 0$, $Pr(B_i)$ implies $Pref(B_i)$. □

Lemma C.5 Let $[\![\cdot]\!]$ be an encoding from Aπ into PAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. must-preservation,
3. $[\![\omega.0]\!] \xrightarrow{\omega}$.

Then $\forall x, y, z \in \mathcal{N}$, $Pr([\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}])$.

Proof. First, we prove that $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \longrightarrow$. By contradiction, suppose $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \not\longrightarrow$. By Lemma B.2, every hole is prefixed in $C_{x(y)}^{[\![\cdot]\!]}$. This implies that $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \not\xrightarrow{\omega}$, that is $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]] \not\longrightarrow$ and $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}[[\![\omega.0]\!]]] \not\xrightarrow{\omega}$. It means that $[\![\bar{x}z \mid \bar{x}z]\!]$ *must* $[\![x(y).x(y).\omega.0]\!]$, contradicting that $\bar{x}z \mid \bar{x}z$ *must* $x(y).x(y).\omega.0$.

Since every hole is prefixed in $C_{x(y)}^{[\![\cdot]\!]}$, we have that $C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \equiv \langle !h(k).C[!h(k).C'\{[\![\omega.0]\!]\} \mid T'] \mid T\rangle$, where $T \in \mathrm{PA}\pi$ and $h \notin \langle\cdot\rangle$ (otherwise, we could not unprefix one occurrence of $[\![\omega.0]\!]$). Since $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \longrightarrow$, it follows that $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \equiv \langle !\bar{h}k \mid !h(k).C[!h(k).C'\{[\![\omega.0]\!]\} \mid T'] \mid T''\rangle$. Since $min(!h(k).C[!h(k).C'\{[\![\omega.0]\!]\}]) \geq 2$, $Pr([\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}])$ holds.□

Theorem C.6 Let $[\![\cdot]\!]$ be an encoding from Aπ into PAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. By contradiction, suppose $[\![\cdot]\!]$ is must-preserving. Let $P = \bar{x}z \mid \bar{x}z$ and $o = x(y).x(y).\omega.0$. Consider $[\![P]\!] \mid [\![o]\!] = [\![\bar{x}z \mid \bar{x}z]\!] \mid [\![x(y).x(y).\omega.0]\!]$. By Lemma C.5, $Pr([\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}])$. Hence $Pref([\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}])$.

By Proposition C.4, there exists a non-empty maximal computation from $B = [\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}]$

$$B = B_0 \longrightarrow B_1 \longrightarrow B_2 \longrightarrow \ldots \longrightarrow B_i[\longrightarrow \ldots]$$

such that $\forall i \geq 0, Pref(B_i)$. As a consequence, $\forall i \geq 0, B_i \stackrel{\omega}{\not\longrightarrow}$. It follows that there exists a maximal computation from $Unbrace(B) = P \mid o$

$$Unbrace(B) = T_0 \longrightarrow T_1 \longrightarrow T_2 \longrightarrow \ldots \longrightarrow T_i[\longrightarrow \ldots]$$

such that $\forall i \geq 0, T_i \stackrel{\omega}{\not\longrightarrow}$. This means that P *must* o, contradicting the must-preservation hypothesis of $[\![\cdot]\!]$. □

Negative results for PIAπ

$$\text{P}\ \frac{B_1 \in \mathcal{L}(P)\ ,\ B\{z/y\} = C_1[!x(y).B_1],\quad x \notin bn(C_1)}{P(!x_i(y).B, x)}$$

$$\text{Base}\ \frac{P(!x(y).B, x)}{Pr_{in}(!x(y).B, x)} \qquad \text{Res}\ \frac{Pr_{in}(B, x)\ ,\ x \neq y}{Pr_{in}((\nu y)B, x)}$$

$$\text{Cong}\ \frac{Pr_{in}(B', x)\ ,\ B' \equiv B}{Pr_{in}(B, x)} \qquad \text{Par1}\ \frac{Pr_{in}(B_1, x)\ ,\ P(!x_i(y_i).B_i, x)}{Pr_{in}(B_1 \mid !x_i(y_i).B_i, x)}$$

$$\text{Par2}\ \frac{Pr_{in}(B_1, x)\ ,\ min(!x(y_i).B_i) = 1}{Pr_{in}(B_1 \mid !x(y_i).B_i, x)} \qquad \text{Par3}\ \frac{Pr_{in}(B_1, x)\ ,\ B_2 \in \text{PIA}\pi}{Pr_{in}(B_1 \mid B_2, x)}$$

Table C.2
Predicate Pr_{in}.

Lemma C.7 Let $P \in \text{PIA}\pi$, $B \in \mathcal{L}(P)$. $\exists x \in fn(B)$ such that $Pr_{in}(B, x)$ iff

$$B \equiv N(x) = (\nu y_1)..(\nu y_m)(\prod_{i=1}^{a} !x(y_i).B_i \mid \prod_{j=1}^{b} !x(y_j).B_j \mid \prod_{h=1}^{c} !x_h(y_h).B_h \mid T)$$

where $a \geq 1$, $m, b, c \geq 0$, $\forall k \in [1..m]\ x \neq y_k$, $\forall i \in [1..a]\ P(!x(y_i).B_i, x)$, $\forall j \in [1..b]\ min(!x(y_j).B_j) = 1$, $\forall h \in [1..c]\ x_h \neq x$ and $P(!x_h(y_h).B_h, x)$, and $T \in \text{PIA}\pi$.

Proof. Consider the *if* implication: given the term $N(x)$, $x \in fn(N(x))$ and $Pr_{in}(N(x), x)$ hold. For the *only if* implication it suffices to prove that for each rule in Table C.2 (unless Rule P) the postcondition can be written, via $\equiv$, as $N(x)$. □

Lemma C.8 Let $[\![\cdot]\!]$ be an encoding from Aπ into PIAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. must-preservation,
3. $[\![\omega.0]\!] \stackrel{\omega}{\longrightarrow}$.

Then $fn(C^{[\![\cdot]\!]}_{x(y)}) \neq \emptyset$.

Proof. Suppose $fn(C^{[\![\cdot]\!]}_{x(y)}) = \emptyset$. Since $[\![\bar{x}z]\!]$ *must* $C_{x(y)}[[\![\omega.0]\!]]$, it follows that $\forall A$ such that $C_{x(y)}[[\![\omega.0]\!]] \longrightarrow \ldots \longrightarrow A$, $fn(A)/\{\omega\} = \emptyset$ and $A \longrightarrow \ldots \longrightarrow A' \xrightarrow{\omega}$. Then $C_{x(y)}[[\![0]\!]]$ *must* $C_{x(y)}[[\![\omega.0]\!]]$, i.e. $x(y).0$ *must* $x(y).\omega.0$. It is a contradiction of the must-preservation hypothesis on $[\![\cdot]\!]$. □

Lemma C.9 Let $[\![\cdot]\!]$ be an encoding from Aπ into PIAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. must-preservation,
3. $[\![\omega.0]\!] \xrightarrow{\omega}$.

Then $C_{x(y)}[C_{x(y)}[\cdot]] \equiv \langle !u(v).C[!u'(v).C'[\cdot] \mid T'] \mid T\rangle$.

Proof. It follows immediately from Lemma B.2. □

Lemma C.10 and Proposition C.11 are useful to prove Proposition C.12.

Lemma C.10 Let $P \in$ PIAπ, $B \in \mathcal{L}(P)$. $\exists x \in fn(B) : Pr_{in}(B, x)$ implies $Pref(B)$.

Proof. Trivial. □

Proposition C.11 Let $P \in$ PIAπ, $B \in \mathcal{L}(P)$, $\exists x \in fn(B)$ such that $Pr_{in}(B, x)$ and $B \longrightarrow B'$ for some $B' \in \mathcal{L}(P)$. Then $\exists B'' \in \mathcal{L}(P)$ such that $B \longrightarrow B''$, $x \in fn(B'')$ and $Pr_{in}(B'', x)$.

Proof. By Lemma C.7, B can be written in the normal form $N(x)$ as in Lemma C.7. By operational Rule Cong, we consider $N(x) \longrightarrow B'' \equiv B'$, for some $B'' \in \mathcal{L}(P)$. We can suppose to apply α- equivalence in such a way $\forall a, b \in bn(N(x)), a \neq b$ and $\forall a \in bn(N(x))$ and $\forall b \in fn(N(x))$, $a \neq b$. We distinguish four cases:

a. $T \longrightarrow T'$: trivial;

b. $T \equiv \bar{x}z \mid T'$ and $!x(y_i).B_i \mid T \longrightarrow B_i\{z/y_i\} \mid !x(y_i).B_i \mid T'$ for some $i \in [1..a]$ and $a \geq 1$: without loss of generality, suppose that $i = 1$ and there is only one hole in C_1 in Rule P of Table C.2. Since $B_1\{z/y_1\} = C_1[!x(y).B'_1]$ and $x \notin bn(C_1)$, the case $min(B'_1\{z/y_1\}) \geq 2$ implies that $B_1\{z/y_1\} \equiv \langle !\alpha(\beta).C'_1[!x(y).B'_1] \mid T''\rangle$, $x \notin \langle\cdot\rangle$, $x \notin bn(C'_1)$ and $x \notin bn(T'')$ (either $\alpha = x$ or $\alpha \neq x$), while the case $min(B_1\{z/y_1\}) = 1$ implies that $B_1\{z/y_1\} \equiv \langle !x(y).B'_1 \mid T''\rangle$, where $x \notin \langle\cdot\rangle$, $x \notin bn(T'')$ and $T'' \in$ PIAπ in both cases. It is possible to prove that in both cases B'' can be written in a normal form $N''(x)$, i.e. $x \in fn(B'')$ and $Pr_{in}(B'', x)$.

c. $T \equiv \bar{x}z \mid T'$ and $!x(y_j).B_j \mid T \longrightarrow B_j\{z/y_j\} \mid !x(y_j).B_j \mid T'$ for some $j \in [1..b]$ and $b \geq 0$: without loss of generality, suppose $j = 1$. Since there is at least one $!x(y_i).B_i$ for some $i \in [1..a]$ (being $a \geq 1$), we can replace this reduction with the reduction from $!x(y_i).B_i \mid T$, considered in item (b).

d. $T \equiv \bar{x}_h z | T'$ and $!x_h(y_h).B_h | T \longrightarrow B_h\{z/y_h\} | !x_h(y_h).B_h | T'$ for some $h \in [1..c]$ and $c \geq 0$: without loss of generality, suppose $h = 1$ and there is only one hole in C_1 in Rule P of Table C.2. We recall that $x_1 \neq x$. Since $B_1\{z/y_1\} = C_1[!x(y).B'_1]$ and

$x \notin bn(C_1)$, $min(B_1'\{z/y_1\}) \geq 2$ implies $B_1'\{z/y_1\} \equiv \langle !\alpha(\beta).C_1'[!x(y).B_1'] \,|\, T''\rangle$, $x \notin \langle\cdot\rangle$, $x \notin bn(C_1')$ and $x \notin bn(T'')$ (either $\alpha = x$ or $\alpha \neq x$), while $min(B_1\{z/y_1\}) = 1$ implies $B_1\{z/y_1\} \equiv \langle !x(y).B_1' \mid T''\rangle$, where $x \notin \langle\cdot\rangle$, $x \notin bn(T'')$ and $T'' \in \text{PIA}\pi$ in both cases. As in item (a), applying $\equiv$ it is possible to prove that in both cases B'' can be written in a normal form $N''(x)$, i.e. $x \in fn(B'')$ and $Pr_{in}(B'', x)$.

□

Proposition C.12 Let $P \in \text{PIA}\pi$, $B \in \mathcal{L}(P)$, $\exists x \in fn(B)$ such that $Pr_{in}(B, x)$. Then there exists a maximal computation from B (also empty)

$$B = B_0 \longrightarrow B_1 \longrightarrow B_2 \longrightarrow \ldots \longrightarrow B_i[\longrightarrow \ldots]$$

such that $\forall i \geq 0, Pref(B_i)$.

Below, we prove that, under reasonable conditions, there exists a term satisfying the predicate Pr_{in} (Lemma C.13), and finally the impossibility result for PIAπ (Theorem C.14).

Lemma C.13 Let $[\![\cdot]\!]$ be an encoding from Aπ into PIAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. must-preservation,
3. $[\![\omega.0]\!] \xrightarrow{\omega}$,
4. if $fn(P) \cap bn(x(y)) = \emptyset$ then $fn([\![P]\!]) \cap bn(C^{[\![\cdot]\!]}_{x(y)}) = \emptyset$,
   (Preservation of independence wrt input prefix)
5. $[\![x(y).P]\!] \equiv \langle !u(v).C[[\![P]\!]] \mid T\rangle$, where $u \notin \langle\cdot\rangle$.

Then $\exists h \in fn(C^{[\![\cdot]\!]}_{x(y)})$: $Pr_{in}(C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}], h)$.

Proof. From (1), we know that $C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \equiv \langle !u(v).C[[\![x(y).\omega.0]\!]] \mid T\rangle \equiv \langle !u(v).C[\langle !u(v).C[[\![\omega.0]\!]] \mid T'\rangle] \mid T\rangle$, where u is a free name in both cases (in the more external case by (5) and in the internal case by (4)). Then we can verify that for u in $fn(C^{[\![\cdot]\!]}_{x(y)})$, $Pr_{in}(C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}], h)$ holds. □

Theorem C.14 Let $[\![\cdot]\!]$ be an encoding from Aπ into PIAπ that satisfies:

1. compositionality w.r.t. input prefix,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$,
3. $\exists h \in fn(C^{[\![\cdot]\!]}_{x(y)})$: $Pr_{in}(C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}], h)$.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. By contradiction, suppose $[\![\cdot]\!]$ being must-preserving. We can apply α-equivalence to $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}]$ in such a way to avoid collision among bound/free names. By (3), we have that $h \in fn(C_{x(y)}[C_{x(y)}[\cdot]])$, and by Table C.2, $Pr_{in}([\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}], h)$ holds. Moreover, we can prove that $[\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \longrightarrow . \; [\![\bar{x}z \mid \bar{x}z]\!] \mid C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}] \not\longrightarrow$ would imply $[\![\bar{x}z \mid \bar{x}z]\!]$ *must* $C_{x(y)}[C_{x(y)}\{[\![\omega.0]\!]\}]$, i.e. $\bar{x}z \mid \bar{x}z$ *must* $x(y).x(y).\omega.0$, contradict-

ing the must-preservation hypothesis on $[\![\cdot]\!]$. Now, it suffices to apply Proposition C.12. □

Negative results for POAπ

The following theorem states a negative result for the 0-adic versions of Aπ and POAπ (denoted resp. by ACCS and POA$_{\text{ACCS}}$). It can be reformulated for Aπ and POAπ by imposing some syntactic restrictions to both source and target language. The hypotheses are quite strong, in particular (3) and (4). However, they are reasonable for acknowledgment-based encodings, where two partners $[\![a(y).P]\!]$ and $[\![\bar{a}z]\!]$ start a communication protocol on a well-known channel x and keep on communicating by means of private channels. Although an encoding could easily violate the above conditions, this result is important since, differently from the previous ones, no form of divergence is either introduced or hidden, i.e. the must-preserving property is violated without taking into any divergence hypothesis.

Theorem C.15 Let $[\![\cdot]\!]$ be an encoding from ACCS into POA$_{\text{ACCS}}$ that satisfies:

1. compositionality w.r.t. input prefix,
2. $[\![\omega.0]\!] \xrightarrow{\omega}$,
3. $\forall a \in \mathcal{N}$, $|fn(C_a^{[\![\cdot]\!]})| = |fn([\![\bar{a}]\!])| = 1$;
4. $x \in fn(K)$ implies $\#_{\text{fn}}(x, K) = 1$, where $K \in \{C_a^{[\![\cdot]\!]}, [\![\bar{a}]\!]\}$ and $\#_{\text{fn}}(x, K)$ denotes the number of free occurrences of x in K.

Then $[\![\cdot]\!]$ is not must-preserving.

Proof. In the following, $(!)P$ denotes both $!P$ and P. By contradiction, suppose $[\![\cdot]\!]$ is must-preserving. By Lemma B.2, every hole is prefixed in $C_a^{[\![\cdot]\!]}$. It follows that $\forall a \in \mathcal{N}$ and $\forall j \geq 1$, $[\![(\bar{a})^j]\!] \mid C_a[[\![\omega.0]\!]] \longrightarrow$, where $(\bar{a})^j$ denotes the parallel composition of j copies of $\bar{a}$: if $[\![(\bar{a})^j]\!] \mid C_a[[\![\omega.0]\!]] \not\longrightarrow$, $[\![(\bar{a})^j]\!]$ *must* $[\![a.\omega.0]\!]$, i.e. $(\bar{a})^j$ *must* $a.\omega.0$, contradicting the must-preservation hypothesis on $[\![\cdot]\!]$. It follows that both $fn(C_a[\cdot])$ and $fn([\![(\bar{a})^j]\!])$ are not empty sets, i.e. item (3) is well-defined. We can write $[\![(\bar{a})^j]\!]$ and $C_a^{[\![\cdot]\!]}$ as follows:

1. either $[\![(\bar{a})^j]\!] \equiv (\nu x_1)..(\nu x_m)(!\bar{x} \mid B_0)$, where $fn(B_0) = \emptyset$ and $\forall i, x \neq x_i$

and one of the following configurations

a $C_a[\cdot] \equiv (\nu x_1)..(\nu x_n)(x.G_1[\cdot] \mid B_1)$,
b $C_a[\cdot] \equiv (\nu x_1)..(\nu x_n)(!x.G_1[\cdot] \mid B_1)$,
c $C_a[\cdot] \equiv (\nu x_1)..(\nu x_n)(x.C_1[\cdot] \mid B_1)$,
d $C_a[\cdot] \equiv (\nu x_1)..(\nu x_n)(!x.C_1[\cdot] \mid B_1)$,
e $C_a[\cdot] \equiv (\nu x_1)..(\nu x_n)(x.B_1 \mid G_1[\cdot])$,
f $C_a[\cdot] \equiv (\nu x_1)..(\nu x_n)(!x.B_1 \mid G_1[\cdot])$,

where $\forall i, x \neq x_i$, $fn(G_1) = fn(C_1) = fn(B_1)/\{\omega\} = \emptyset$, every hole is prefixed in G_1 and not in C_1,

2. or $C_a[\cdot] \equiv (\nu x_1)..(\nu x_m)(!\bar{x} \mid G_2[\cdot])$, where $fn(G_2) = \emptyset$, $\forall i, x \neq x_i$ and every hole is prefixed in G_1

and one of the following configurations

a $[\![(\bar{a})^j]\!] \equiv (\nu x_1)..(\nu x_n)(x.B_1 \mid B_2)$,

b $[\![(\bar{a})^j]\!] \equiv (\nu x_1)..(\nu x_n)(!x.B_1 \mid B_2)$, where $\forall i, x \neq x_i$, $fn(B_1) = fn(B_2) = \emptyset$.

In the cases of (1-a), (1-b), (1-e), (1-f), (2-a) and (2-b), we can deduce that $[\![\bar{a}]\!]$ *must* $[\![a.\omega.0]\!]$, implying that $\bar{a}$ *must* $a.\omega.0$. This contradicts the must-preservation hypothesis on $[\![\cdot]\!]$.

Consider the case (1-c). This implies (up to α-equivalence) $[\![\bar{a}]\!] \mid [\![a.a.\omega.0]\!] \equiv \langle !\bar{x} \mid x.C_1[\langle x.C_1[[\![\omega.0]\!]] \mid B_1\rangle] \mid B_0 \mid B_1\rangle \longrightarrow \langle !\bar{x} \mid C_1[\langle x.C_1[[\![\omega.0]\!]] \mid B_1\rangle] \mid B_0 \mid B_1\rangle \equiv \langle !\bar{x} \mid (!)(x.C_1[[\![\omega.0]\!]] \mid B_1) \mid B_2 \mid B_0 \mid B_1\rangle$. Suppose $(!)(x.C_1[[\![\omega.0]\!]] \mid B_1) = x.C_1[[\![\omega.0]\!]] \mid B_1$ (the other case is similar). It follows that $\langle !\bar{x} \mid x.C_1[[\![\omega.0]\!]] \mid B_1 \mid B_2 \mid B_0 \mid B_1\rangle \longrightarrow \langle !\bar{x} \mid C_1[[\![\omega.0]\!]] \mid B_1 \mid B_2 \mid B_0 \mid B_1\rangle \equiv \langle !\bar{x} \mid [\![\omega.0]\!] \mid B_2 \mid B_1 \mid B_2 \mid B_0 \mid B_1\rangle$. This implies that $[\![\bar{a}]\!]$ *must* $[\![a.a.\omega.0]\!]$, that is $\bar{a}$ *must* $a.a.\omega.0$, contradicting the must-preservation hypothesis on $[\![\cdot]\!]$.

The case (1-d) implies that $[\![\bar{a} \mid \bar{a}]\!]$ *must* $[\![a.a.\omega.0]\!]$, that is $\bar{a} \mid \bar{a}$ *must* $a.a.\omega.0$, again contradicting the must-preservation hypothesis on $[\![\cdot]\!]$. □