On Equators in Asynchronous Name-passing Calculi without Matching – Extended Abstract –

Massimo Merro [1,2]

INRIA, 2004 Rue des Lucioles, B.P. 93, 06902 Sophia-Antipolis, France.

Abstract

We give a labeled characterization of barbed congruence in asynchronous π-calculus, which, unlike previous characterizations, does not use the matching construct. In absence of matching the observer cannot directly distinguish two names. In asynchronous π-calculus the fact that two names are indistinguishable can be modeled by means of Honda and Yoshida's notion of *equator*. Our labeled characterization is based on such a notion. As an application of our theory we provide a fully abstract encoding w.r.t. barbed congruence of *external mobility* (communication of free names) in terms of *internal mobility* (communication of private names).

1 Introduction

The π-calculus [13,14] is a calculus of *mobile processes*, i.e., processes with a dynamic linkage reconfiguration. Central to the π-calculus is the notion of *name*. Two processes sharing a name can use it to interact with each other. The interaction between two processes is synchronous, i.e., the send and receive events happen at the same time.

On the other hand asynchronous communications are easier to implement and closer to the communication primitives offered by available distributed systems. Honda and Tokoro [8], and independently Boudol [5], have introduced the *asynchronous π-calculus*, in short π_{a}, as a variant of π-calculus where message emission is non-blocking. The asynchronous π-calculus is a very expressive fragment of π-calculus and it is at the core of high-level concurrent programming languages such as Pict [17].

[1] Funded by the European Union, under the Marie Curie TMR programme
[2] E-mail: Massimo.Merro@sophia.inria.fr

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

The theory of π-calculus is well-developed and focuses on bisimulation-based behavioral equivalences. In the present document, we investigate *barbed congruence* [15]. This equates processes that, very roughly, in all contexts give rise to the same patterns of interactions. The main inconvenience of barbed congruence is that it uses quantification over contexts in its definition, and this can make proofs of process equalities heavy. For this reason it is important to find *direct* characterizations without context quantification. In presence of *matching*, a construct for testing equality between names, barbed congruence coincides with the closure under substitutions of *early* bisimilarity (on image-finite processes) [1,18]. In the proof of the above characterization, the matching construct plays a central role. This operator gives the observer the capability of looking at the names emitted by the processes to ensure that the labels of matching transitions are syntactically the same. Unfortunately, the matching construct, like testing equality between pointers in imperative languages, prevents many useful program optimizations and transformations. Also from a programming point of view the usefulness of matching is questionable. So, as already pointed out in the literature, it would be interesting to find a direct characterization of barbed congruence in absence of matching. The challenge becomes particularly interesting in asynchronous π-calculus where, in absence of matching, early bisimilarity is a congruence relation [7], and coincides with *ground*, *late* and *open bisimilarities* [19]. This is strongly related to the fact that, in asynchronous π-calculus without matching, the *substitution* can be encoded, up-to barbed congruence, in terms of *restriction*, *parallel composition* and a special process, called *equator*, first defined by Honda and Yoshida [9] as

$$\mathcal{E}(a,b) \stackrel{\text{def}}{=} !a(x).\,\overline{b}x \mid !b(x).\,\overline{a}x.$$

If we denote with $\cong_{\pi_a}$ the barbed congruence w.r.t. π_a-contexts, the following law holds:

$$P\{b/c\} \cong_{\pi_a} \boldsymbol{\nu}c\,(\mathcal{E}(b,c) \mid P) \tag{1}$$

Law 1 has already been proved with respect to the *maximum sound theory* [9]. More recently, Fournet and Gonthier have proved that, in asynchronous π-calculus, maximum sound theory and barbed congruence coincide [6]. Notice that the presence of either output prefix or matching would invalidate Law 1. As suggested by their name, equators induce a sort of equivalence relation over names. The notion of equator is very close to that of *fusion* of names introduced by Parrow and Victor in their *Fusion calculus* [16]. Actually these two concepts are more than similar; in [10] we already showed that, in an asynchronous scenario, it is possible to encode fusions in terms of equators.

In this paper we show that the notion of equator can be successfully used to provide a direct co-inductive characterization of barbed congruence in asynchronous π-calculus in absence of matching. Our characterization, called *synonymous bisimilarity*, is a congruence relation and therefore does not have to be closed under substitutions to yield barbed congruence. Synonymous bisimilarity extends the notion of *asynchronous ground bisimilarity* introduced by Amadio, Castellani, and Sangiorgi [1]. Roughly, synonymous bisimilarity coincides with asynchronous ground bisimilarity when dealing with input and silent moves, but admits a more permissive behavior when dealing with output actions. More precisely, it allows us to match output actions with the same subject part but different object parts by making the latter indistinguishable. We use the notion of equator to model the fact that two names cannot be distinguished by the environment. For simplicity we develop the theory for a monadic calculus; the generalization to the polyadic case is straightforward.

As an application of our theory we give an encoding of *external mobility* (when communicating free names) in terms of *internal mobility* [20] (when communicating private names). We prove that this encoding is fully-abstract with respect to barbed congruence. So, at least in an asynchronous scenario, we make precise Sangiorgi's claim [20] that internal mobility is responsible for the expressiveness of π-calculus.

In this extended abstract proofs are just sketched; detailed proofs can be found in the full version of the paper [11].

Related works The works which are most closely related to ours are [4,12,3]. In [4] Boreale and Sangiorgi study barbed congruence in synchronous π-calculus with capability types and no matching. Our characterization is much simpler than that in [4], but the latter is more general because it can be applied to several π-calculus languages (although the authors say that the extension to asynchronous π-calculus is not obvious). In [12] Merro and Sangiorgi give two direct characterizations of barbed congruence in Lπ, a variant of π_{a} where only the *output capability* of names may be transmitted (i.e., the recipient of a name may only use such name in output actions). Also in this case our characterization is simpler. Finally, in [3] Boreale introduces an encoding of external mobility in terms of the internal one proving its adequacy w.r.t. barbed bisimilarity. Our encoding is inspired by Boreale's encoding. Most of the proof techniques presented here are inspired by those in [3,12].

2 The asynchronous π-calculus

In this section the asynchronous π-calculus and its standard bisimulations will be briefly reviewed.

Let $\mathcal{N}$ be a countable set of *names* and $a, b, \ldots, x, y$ range over it. The grammar of π_{a} has operators of *inaction*, *input prefix*, *asynchronous output*, *parallel composition*, *restriction* and *recursive definitions*:

$$P ::= \mathbf{0} \;\Big|\; a(x).\, P \;\Big|\; \overline{a}b \;\Big|\; P \mid P \;\Big|\; \boldsymbol{\nu} a\, P \;\Big|\; A(a_1, \ldots, a_n)$$

Each constant A has a unique defining equation of the form $A(\widetilde{a}) \stackrel{\text{def}}{=} P$ where $\widetilde{a} = a_1, \ldots, a_n$, with the a_i pairwise distinct, and $\text{fn}(P) \subseteq \{a_1, \ldots, a_n\}$. In a constant definition $A(\widetilde{a}) \stackrel{\text{def}}{=} P$, all free occurrences of names of $\widetilde{a}$ in P are bound. We use σ for substitutions; $P\sigma$ is the result of applying σ to P, with the usual renaming convention to avoid captures. Parallel composition has the lowest precedence among the operators. The labeled transition system is the usual one (in the early style [1]). We also have a notion of *structural congruence*, written $\equiv$ and defined as usual (see [13]), to ignore certain structural differences between processes. Transitions are of the form $P \xrightarrow{\mu} P'$, where *action* μ can be: τ (interaction), ab (input), $\overline{a}b$ (free output) and $\overline{a}(b)$ (bound output, that is the emission of a private name b at a). In these actions, a is the *subject* and b the *object*. Free and bound names (fn, bn) of actions and processes are defined as usual. Relation $\Longrightarrow$ is the reflexive and transitive closure of $\xrightarrow{\tau}$; moreover, $\stackrel{\mu}{\Longrightarrow}$ stands for $\Longrightarrow\xrightarrow{\mu}\Longrightarrow$, and $\stackrel{\hat{\mu}}{\Longrightarrow}$ for $\stackrel{\mu}{\Longrightarrow}$ if $\mu \neq \tau$, and for $\Longrightarrow$ if $\mu = \tau$.

We focus on bisimulation-based behavioral equivalences, precisely on *barbed congruence* [15]. It is well-known that barbed congruence represents a uniform mechanism for defining behavioral equivalence in any process calculus possessing (i) an *interaction relation* (the τ-steps in π-calculus) modeling the evolution of the system, and (ii) an *observability predicate* which simply detects the possibility of performing some observable action. In asynchronous calculi it is natural to restrict the observation to the output commitments [1]. We write $P \downarrow_a$ if P can make an output action whose subject is a, that is if there exist P', b such that $P \xrightarrow{\overline{a}b} P'$ or $P \xrightarrow{\overline{a}(b)} P'$. We write $P \Downarrow_a$ if $P \Longrightarrow P'$ and $P' \downarrow_a$. We recall that a *context* $C[\cdot]$ is a process with exactly one hole where a process may be plugged in.

Definition 2.1 (barbed bisimilarity and congruence) *Barbed bisimilarity*, written $\dot{\approx}$, is the largest symmetric relation on π-calculus processes such that $P \dot{\approx} Q$ implies:

(i) If $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \Longrightarrow Q'$ and $P' \dot{\approx} Q'$.

(ii) If $P \downarrow_a$ then $Q \Downarrow_a$.

Let $\mathcal{L}$ be a set of processes in π_{a}, and $P, Q \in \mathcal{L}$. We say that P and Q are *barbed congruent in* $\mathcal{L}$, written $P \cong_{\mathcal{L}} Q$, if for each $\mathcal{L}$-context $C[\cdot]$, it holds that $C[P] \dot{\approx} C[Q]$.

The main inconvenience of barbed congruence is that it uses quantification over contexts in the definition, and this can make proofs of process equalities heavy. For this reason it is important to find *labeled* characterizations without context quantification.

Definition 2.2 (ground bisimilarities) A symmetric relation $\mathcal{S}$ on π-terms is an *oτ-bisimulation* if $P \mathrel{\mathcal{S}} Q$, $P \xrightarrow{\mu} P'$, μ is not an input and $\text{bn}(\mu) \cap \text{fn}(Q) = \emptyset$, implies that there exists Q' such that $Q \stackrel{\widehat{\mu}}{\Longrightarrow} Q'$ and $P' \mathrel{\mathcal{S}} Q'$.

- *synchronous ground bisimilarity*, denoted by $\approx$, is the largest $o\tau$-bisimulation $\mathcal{S}$ such that $P \; \mathcal{S} \; Q$ and $P \xrightarrow{aw} P'$, with $w \notin \mathrm{fn}(P \mid Q)$, implies that there exists Q' such that $Q \stackrel{aw}{\Longrightarrow} Q'$ and $P' \; \mathcal{S} \; Q'$.
- *asynchronous ground bisimilarity*, denoted by $\approx_{\mathrm{a}}$, is the largest $o\tau$-bisimulation $\mathcal{S}$ such that $P \; \mathcal{S} \; Q$ and $P \xrightarrow{aw} P'$, with $w \notin \mathrm{fn}(P \mid Q)$, implies that there exists Q' such that:

(i) either $Q \stackrel{aw}{\Longrightarrow} Q'$ and $P' \; \mathcal{S} \; Q'$
(ii) or $Q \Longrightarrow Q'$ and $P' \; \mathcal{S} \; (Q' \mid \overline{a}w)$.

We recall that in the asynchronous π-calculus without matching ground bisimilarities coincide with early, late, and open bisimilarities [19]. All these relations are congruences and imply barbed congruence.

3 A labeled characterization in absence of matching

In this section we provide a direct co-inductive characterization of barbed congruence for the asynchronous π-calculus without matching. This characterization generalizes the asynchronous ground bisimilarity by allowing us to match, under certain hypotheses, output actions with the same subject part and different object part.

The theory we are going to develop is based on a special process, called *equator*, first defined by Honda and Yoshida [9] as follows [3]:

$$\mathcal{E}(a,b) \stackrel{\mathrm{def}}{=} a(x).\,\overline{b}x \mid b(x).\,\overline{a}x \mid \mathcal{E}(a,b)$$

As suggested by their name, equators induce a sort of equivalence relation over names.

Lemma 3.1

(i) $\boldsymbol{\nu} a\, \mathcal{E}(a,b) \;\cong_{\pi_{\mathrm{a}}}\; \mathcal{E}(b,b) \;\cong_{\pi_{\mathrm{a}}}\; \mathbf{0}$
(ii) $\mathcal{E}(a,b) \;\cong_{\pi_{\mathrm{a}}}\; \mathcal{E}(b,a)$
(iii) $\boldsymbol{\nu} c\, (\mathcal{E}(a,c) \mid \mathcal{E}(c,b)) \;\cong_{\pi_{\mathrm{a}}}\; \mathcal{E}(a,b)$.

Proof. By exhibiting an appropriate synchronous or asynchronous ground bisimulation. Since both bisimularities imply barbed congruence we can conclude. ■

Equators play a central role in the asynchronous π-calculus without matching. As shown by the following Laws 2 and 3, they permit to equate output processes emitting different names along the same channel.

$$\overline{a}b \mid \mathcal{E}(b,c) \;\cong_{\pi_{\mathrm{a}}}\; \overline{a}c \mid \mathcal{E}(b,c) \tag{2}$$
$$\boldsymbol{\nu} c\, (\overline{a}c \mid \mathcal{E}(c,b)) \;\cong_{\pi_{\mathrm{a}}}\; \overline{a}b \tag{3}$$

[3] Equators were originally defined using replication instead of recursion.

The two laws above have already been proved in [9] and explain why, in absence of matching, a characterization of barbed congruence must match output actions with the same subject part but different object part. Laws 2 and 3 also suggest a natural candidate for a labeled characterization of barbed congruence in π_{a}.

Definition 3.2 (synonymous bisimilarity) A symmetric relation $\mathcal{S}$ on π_{a}-terms is a *synonymous bisimulation* if whenever $P \ \mathcal{S}\ Q$ the following holds:

(i) If $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \Longrightarrow Q'$ and $P' \ \mathcal{S}\ Q'$

(ii) If $P \xrightarrow{aw} P'$, with $w \notin \mathrm{fn}(P \mid Q)$, then there exists Q' such that
  (a) either $Q \stackrel{aw}{\Longrightarrow} Q'$ and $P' \ \mathcal{S}\ Q'$
  (b) or $Q \Longrightarrow Q'$ and $P' \ \mathcal{S}\ (Q' \mid \overline{a}w)$

(iii) If $P \xrightarrow{\overline{a}b} P'$ then there exists Q' such that
  (a) either $Q \stackrel{\overline{a}c}{\Longrightarrow} Q'$ and for some $w \notin \mathrm{fn}(P \mid Q)$ it holds $(\mathcal{E}(w,b) \,|\, P') \ \mathcal{S}\ (\mathcal{E}(w,c) \,|\, Q')$
  (b) or $Q \stackrel{\overline{a}(w)}{\Longrightarrow} Q'$, with $w \notin \mathrm{fn}(P)$, and $(\mathcal{E}(w,b) \,|\, P') \ \mathcal{S}\ Q'$

(iv) If $P \xrightarrow{\overline{a}(w)} P'$, with $w \notin \mathrm{fn}(Q)$, then there exists Q' such that
  (a) either $Q \stackrel{\overline{a}(w)}{\Longrightarrow} Q'$ and $P' \ \mathcal{S}\ Q'$
  (b) or $Q \stackrel{\overline{a}b}{\Longrightarrow} Q'$, for some b, and $P' \ \mathcal{S}\ (\mathcal{E}(w,b) \,|\, Q')$

We write $P \approx_{\varepsilon} Q$ if $P \ \mathcal{S}\ Q$ for some synonymous bisimulation $\mathcal{S}$.

The only difference with respect to the asynchronous ground bisimilarity is in the clauses for output actions where it is possible to send to the environment different names along the same channel. In clause iii(a), for instance, processes P and Q may emit along a two names b and c. To model the fact that b and c cannot be distinguished by the environment, the bisimulation game continues on the derivatives in parallel with two equators $\mathcal{E}(w,b)$ and $\mathcal{E}(w,c)$ making b and c "equal" to a fresh name w. From the environment point of view it is as if both processes P and Q had emitted the same fresh name w instead of b and c. We also allow the matching between free output actions and bound output actions. As in the previous case we need to record, in the derivatives, that we are sending different names. We believe that, when $b = c$, clause iii(a) can be optimized by avoiding to add the equators in the derivatives.

The remainder of this section is devoted to prove that the relation $\approx_{\varepsilon}$ is a labeled characterization of barbed congruence in π_{a}.

3.1 $\approx_{\varepsilon}$ is contained in barbed congruence

We prove that $\approx_{\varepsilon}$ is contained in barbed congruence by exploiting a compositional encoding $[\![\cdot]\!]$ which is an homomorphism on all operators except output, for which we have:

$$[\![\overline{a}b]\!] \stackrel{\mathrm{def}}{=} \boldsymbol{\nu} c\,(\overline{a}c \mid \overrightarrow{\mathcal{E}}\,(b,c))$$

where $\vec{\mathcal{E}}(b,c) \stackrel{\text{def}}{=} b(x).\,[\![\overline{c}x]\!] \mid c(x).\,[\![\overline{b}x]\!] \mid \vec{\mathcal{E}}(b,c)$. The process $\vec{\mathcal{E}}(a,b)$ is closely related to Honda's equator $\mathcal{E}(a,b)$. It holds that $[\![\mathcal{E}(a,b)]\!] = \vec{\mathcal{E}}(a,b)$.

The encoding $[\![\cdot]\!]$ is inspired by Boreale's encoding of free outputs in terms of bound outputs [3]. We denote with $\mathcal{P}_{\mathrm{a}}$ the image of our encoding, that is, the calculus obtained from π_{a} by replacing in the syntax the output particle $\overline{a}b$ with $[\![\overline{a}b]\!]$. If $P \in \mathcal{P}_{\mathrm{a}}$ then each derivative of P is still in $\mathcal{P}_{\mathrm{a}}$. We write $\approx_{\mathrm{A}}$ to denote the variant of asynchronous ground bisimulation in which, in the asynchronous clause for input, we replace, in the requirement for the derivatives, the output particle $\overline{a}w$ with its translation $[\![\overline{a}w]\!]$. We also write $\lesssim_{\mathrm{A}}$ to denote the *expansion* [2] version of $\approx_{\mathrm{A}}$. $\lesssim_{\mathrm{A}}$ refines $\approx_{\mathrm{A}}$ by requiring that if $Q \lesssim_{\mathrm{A}} P$, then all weak transitions in the behavior of P have at least as many τ-moves as the corresponding weak transitions in the behavior of Q.

The following lemma says that the new form of equator behaves as standard equators.

Lemma 3.3

(i) $\boldsymbol{\nu} a\, \vec{\mathcal{E}}(a,b) \gtrsim_{\mathrm{A}} \vec{\mathcal{E}}(b,b) \gtrsim_{\mathrm{A}} \mathbf{0}$

(ii) $\vec{\mathcal{E}}(a,b) \approx \vec{\mathcal{E}}(b,a)$

(iii) $\boldsymbol{\nu} c\, (\vec{\mathcal{E}}(a,c) \mid \vec{\mathcal{E}}(c,b)) \approx \vec{\mathcal{E}}(a,b)$

Proof. By exhibiting the appropriate expansions and/or bisimulations. In part (iii), as for a similar result in [3], the proof requires up-to context and up-to expansion techniques. ∎

Notice that in part (ii) and (iii), instead of $\approx_{\mathrm{A}}$ we have ground bisimilarity.

Lemma 3.4 *The relation $\approx_{\mathrm{A}}$ is a congruence in $\mathcal{P}_{\mathrm{a}}$.*

Proof. The proof that $\approx_{\mathrm{A}}$ is preserved by restriction follows from the fact that, on $\mathcal{P}_{\mathrm{a}}$-processes, output actions are always bound. The proof that $\approx_{\mathrm{A}}$ is preserved by parallel composition is based on Lemma 3.3(iii) and details can be found in [11]. Note that in general to prove that a ground bisimulation is preserved by parallel composition is quite difficult. Our case is easier because we consider $\mathcal{P}_{\mathrm{a}}$-processes. The proof that $\approx_{\mathrm{A}}$ is preserved by recursion does not present particular problems. The proof that $\approx_{\mathrm{A}}$ is preserved by input prefixing follows from the fact that $\approx_{\mathrm{A}}$ is a ground bisimulation and it is trivially preserved by injective substitutions. ∎

Now we investigate the encoding showing that, as for Boreale's encoding, there is an operational correspondence between processes P and $[\![P]\!]$.

Lemma 3.5

(i) *For each process $P \in \mathcal{P}_{\mathrm{a}}$ there exists a process $P' \in \pi_{\mathrm{a}}$ s.t. $[\![P']\!] = P$*

(ii) *For all processes $P \in \pi_{\mathrm{a}}$ and all substitutions σ it holds $[\![P\sigma]\!] = [\![P]\!]\sigma$.*

Proof. By structural induction. ∎

The following lemma shows a technical result which is fundamental for proving the operational correspondence between processes P and $[\![P]\!]$.

Lemma 3.6 *If* $P \in \pi_{\mathrm{a}}$ *then* $\boldsymbol{\nu} a\, (\vec{\mathcal{E}}\,(a,b) \mid [\![P]\!]) \gtrsim_{\mathrm{A}} [\![P]\!]\{^{b}\!/\!_{a}\}$

Proof. As for a similar result in [3] the proof is by structural induction on P and uses Lemma 3.3. Details can be found in [11]. ■

Lemma 3.7 (Operational correspondence) *Let* P *be a process in* π_{a}.

(i) *Suppose that* $P \stackrel{\alpha}{\Longrightarrow} P'$. *Then we have:*
  (a) *if* $\alpha = ac$ *then* $[\![P]\!] \stackrel{ac}{\Longrightarrow} \gtrsim_{\mathrm{A}} [\![P']\!]$;
  (b) *if* $\alpha = \overline{a}b$ *then* $[\![P]\!] \stackrel{\overline{a}(c)}{\Longrightarrow} \gtrsim_{\mathrm{A}} (\vec{\mathcal{E}}\,(c,b) \mid [\![P']\!])$, *with* $c \notin fn(P')$;
  (c) *if* $\alpha = \overline{a}(b)$ *then* $[\![P]\!] \stackrel{\overline{a}(c)}{\Longrightarrow} \gtrsim_{\mathrm{A}} \boldsymbol{\nu} b\, (\vec{\mathcal{E}}\,(c,b) \mid [\![P']\!])$, *with* $c \notin fn(P')$;
  (d) *if* $\alpha = \tau$ *then* $[\![P]\!] \stackrel{\tau}{\Longrightarrow} \gtrsim_{\mathrm{A}} [\![P']\!]$.

(ii) *Suppose that* $[\![P]\!] \stackrel{\alpha}{\Longrightarrow} P_1$. *Then there exists* $P' \in \pi_{\mathrm{a}}$ *s.t.:*
  (a) *if* $\alpha = ac$ *then*
    either $P \stackrel{ac}{\Longrightarrow} P'$, *with* $P_1 \gtrsim_{\mathrm{A}} [\![P']\!]$
    or $P \Longrightarrow P'$, *with* $P_1 \gtrsim_{\mathrm{A}} [\![P']\!] \mid [\![\overline{a}c]\!]$;
  (b) *if* $\alpha = \overline{a}(c)$ *then:*
    either $P \stackrel{\overline{a}b}{\Longrightarrow} P'$, *with* $c \notin fn(P')$ *and* $P_1 \gtrsim_{\mathrm{A}} (\vec{\mathcal{E}}\,(c,b) \mid [\![P']\!])$,
    or $P \stackrel{\overline{a}(b)}{\Longrightarrow} P'$, *with* $c \notin fn(P')$ *and* $P_1 \gtrsim_{\mathrm{A}} \boldsymbol{\nu} b\, (\vec{\mathcal{E}}\,(c,b) \mid [\![P']\!])$;
  (c) *if* $\alpha = \tau$ *then* $P \stackrel{\tau}{\Longrightarrow} P'$ *with* $P_1 \gtrsim_{\mathrm{A}} [\![P']\!]$.

Proof. By transition induction. The only subtle points arise in the proofs of parts i(d) and ii(c) where also Lemma 3.6 is used. Details can be found in [11]. ■

The following three lemmas will allow us to prove the desired soundness result.

Lemma 3.8

(i) $P \Longrightarrow P'$ *implies* $[\![P]\!] \Longrightarrow\gtrsim_{\mathrm{A}} [\![P']\!]$;

(ii) $[\![P]\!] \Longrightarrow P_1$ *implies that there is* P' *such that* $P \Longrightarrow P'$ *and* $P_1 \gtrsim_{\mathrm{A}} [\![P']\!]$;

(iii) $P \Downarrow_a$ *iff* $[\![P]\!] \Downarrow_a$.

Proof. Parts 1 and 2 are proven by induction on the number of τ-moves by exploiting Lemma 3.7. Part iii is a consequence of parts i and ii, and Lemma 3.7. ■

Lemma 3.9 *Let* P *and* Q *be two* π_{a}*-processes. Then*

$$[\![P]\!] \stackrel{\cdot}{\approx} [\![Q]\!] \;\text{ implies }\; P \stackrel{\cdot}{\approx} Q.$$

Proof. We use Lemmas 3.7 and 3.8 and the fact that $\lesssim_{\mathrm{A}} \stackrel{\cdot}{\approx} \gtrsim_{\mathrm{A}} \subset \stackrel{\cdot}{\approx}$ to prove that the relation $\mathcal{R} = \{(P,Q) : [\![P]\!] \stackrel{\cdot}{\approx} [\![Q]\!]\}$ is a barbed bisimulation. ■

Lemma 3.10 *Let P and Q be two π_a-processes.*

$$P \approx_\varepsilon Q \ \text{ implies } \ [\![P]\!] \approx_{\mathrm{A}} [\![Q]\!].$$

Proof. Using Lemmas 3.6 and 3.7 we show that the relation $\mathcal{S} = \{([\![P]\!], [\![Q]\!]) : P \approx_\varepsilon Q\}$ is a $\approx_{\mathrm{A}}$-bisimulation up-to $\gtrsim_{\mathrm{A}}$. Details can be found in [11]. ■

Theorem 3.11 (Soundness) *Let P and Q be two π_a-processes. Then*

$$P \approx_\varepsilon Q \ \text{ implies } \ P \cong_{\pi_\mathrm{a}} Q.$$

Proof. By Lemma 3.10 we have $[\![P]\!] \approx_{\mathrm{A}} [\![Q]\!]$. By Lemma 3.4 $[\![P]\!] \cong_{\mathcal{P}_\mathrm{a}} [\![Q]\!]$. By Lemmas 3.9 and 3.5(i), and the compositionality of $[\![\cdot]\!]$ we have that $P \cong_{\pi_\mathrm{a}} Q$. ■

3.2 Barbed congruence is contained in $\approx_\varepsilon$

We prove that barbed congruence is contained in $\approx_\varepsilon$. Unlike the other characterizations of barbed congruence in the literature, we do not need to close $\approx_\varepsilon$ under substitutions and we do not need the matching construct for ensuring that, during the bisimulation game, an output actions is matched with the same output action. Our notion of bisimulation allows us, up-to equators, to match output actions with the same subject part but different object parts. In some sense, here, the role of the matching construct is played by the equators added in the derivatives. As usual, the theorem is shown for the class of image-finite processes (to which most of the processes one would like to write belong). We recall that the class of *image-finite processes* is the largest subset $\mathcal{I}$ of π-calculus processes which is derivation closed and such that $P \in \mathcal{I}$ implies that, for all μ, the set $\{P' \ : \ P \stackrel{\mu}{\Longrightarrow} P'\}$, quotiented by alpha conversion, is finite. First we show that synonymous bisimilarity satisfies Law 1.

Lemma 3.12 *Let P be a π_a-process. Then $\boldsymbol{\nu} c\,(\mathcal{E}(c,b) \mid P) \approx_\varepsilon P\{b/c\}$.*

Proof. By showing that $\mathcal{S} = \{(\boldsymbol{\nu} c\,(\mathcal{E}(c,b) \mid P), P\{b/c\}) \colon P \in \pi_\mathrm{a}\} \cup \{(\mathcal{E}(c,b) \mid P\,,\ \mathcal{E}(c,b) \mid P\{b/c\}) \colon P \in \pi_\mathrm{a}\}$ is a synonymous bisimulation. ■

The lemma below represents the core of the proof of completeness; its proof is a variant of that in [1] where no matching is used.

Lemma 3.13 *Let P and Q be two image-finite processes in π_a. Then*

$$\text{If } \ (\forall\, R \in \pi_\mathrm{a} \ \ P \,|\, R \stackrel{\bullet}{\approx} \ Q \,|\, R) \ \text{ then } P \approx_\varepsilon Q.$$

Proof. Let $\simeq_\varepsilon$ be the variant of $\approx_\varepsilon$ obtained by replacing $\stackrel{\mu}{\longrightarrow}$ with $\stackrel{\mu}{\Longrightarrow}$ in the hypothesis of each clause in Definition 3.2. Let $\stackrel{\bullet}{\simeq}$ be the variant of $\stackrel{\bullet}{\approx}$ obtained by replacing $\stackrel{\tau}{\longrightarrow}$ with $\stackrel{\tau}{\Longrightarrow}$ and $\downarrow_a$ with $\Downarrow_a$ in the hypothesis of each clause in Definition 2.1. It holds that $\simeq_\varepsilon = \approx_\varepsilon$ and $\stackrel{\bullet}{\simeq} \ = \stackrel{\bullet}{\approx}$. So, we prove that

$$(\forall\, R \in \pi_{\mathrm{a}}\ \ P \mid R \stackrel{\bullet}{\simeq}\ Q \mid R) \implies P \simeq_{\varepsilon} Q.$$

Let $\mathcal{F}$ be the monotone operator over $\mathcal{P}(Pr \times Pr)$ associated with the definition of $\simeq_{\varepsilon}$. Suppose $\simeq_{\varepsilon}^{0} = Pr \times Pr$, $\simeq_{\varepsilon}^{k+1} = \mathcal{F}(\simeq_{\varepsilon}^{k})$, and $\simeq_{\varepsilon}^{\omega} = \bigcap_{k<\omega} \simeq_{\varepsilon}^{k}$. On image finite lts the operator $\mathcal{F}$ preserves co-directed sets. In particular, $\mathcal{F}(\simeq_{\varepsilon}^{\omega}) = \simeq_{\varepsilon}^{\omega}$. This means that on image finite processes $\simeq_{\varepsilon} = \simeq_{\varepsilon}^{\omega}$. So, more precisely, we prove that

$$(\forall\, R \in \pi_{\mathrm{a}}\ \ P \mid R \stackrel{\bullet}{\simeq}\ Q \mid R) \ \text{ implies that }\ P \simeq_{\varepsilon}^{\omega} Q.$$

This will suffice to conclude.

We define a collection of tests $R(n, L)$ depending on $n \in \omega$ and L a finite set of channel names, and show by induction on n that:
$\exists\, L, L' : L \supseteq fn(P \mid Q)$, $L' \subseteq L$ such that $\boldsymbol{\nu} L'(P \mid R(n,L)) \stackrel{\bullet}{\simeq} \boldsymbol{\nu} L'(P \mid R(n,L))$ implies $P \simeq_{\varepsilon}^{n} Q$. If the property above holds then we can conclude the proof by observing that:

$$\begin{array}{rl}
\forall\, R \in \pi_{\mathrm{a}}\ (P \mid R \stackrel{\bullet}{\simeq}\ Q \mid R) \text{ implies} & \boldsymbol{\nu} L'\,(P \mid R(n,L)) \stackrel{\bullet}{\simeq} \boldsymbol{\nu} L'\,(Q \mid R(n,L)) \\
& \text{for each } n \in \omega \text{ with } L = fn(P \mid Q), \\
& \text{and } L' = \emptyset. \\
\text{implies} & \forall\, n \in \omega\ \ P \simeq_{\varepsilon}^{n} Q \\
\text{implies} & P \simeq_{\varepsilon}^{\omega} Q \\
\text{implies} & P \simeq_{\varepsilon} Q.
\end{array}$$

For defining the tests $R(n, L)$ we introduce an internal choice operator $\oplus$. This is a derived operator defined as follows:

$$P_1 \oplus \ldots \oplus P_n \stackrel{\text{def}}{=} \boldsymbol{\nu} a\,(a.\,P_1 \mid \ldots \mid a.\,P_n \mid \overline{a}) \text{ with } a \notin fn(P_1 \mid \ldots \mid P_n)$$

We suppose that the collection of channel names Ch has been partitioned in two infinite well-ordered sets Ch' and Ch''. In the following we have $L' \subseteq L \subseteq Ch''$. We also assume the sequences $\{b_n, b'_n : n \in \omega\} \cup \{c_n^{\beta} : n \in \omega,\ \beta \in \{\tau, a, \overline{a} : a \in Ch''\}\}$ of distinct names in Ch'. The test $R(n, L)$ is defined by induction on n as follows, where we pick a' and a'' to be the first and the second name in the well-ordered set $Ch'' \setminus L$. When emitting or receiving a name which is not in L we work up to injective substitution to show that $P \simeq_{\varepsilon}^{n} Q$. The relation $R(n, L)$ is defined as follows:

$$\begin{aligned}
R(0, L) &= \overline{b_0} \oplus \overline{b'_0} \\
R(n, L) &= \overline{b_n} \oplus \overline{b'_n} \text{ (for } n > 0) \\
&\quad \oplus (\overline{c_n^{\tau}} \oplus R(n-1, L)) \\
&\quad \oplus \{\overline{c_n^{\overline{a}}} \oplus (\boldsymbol{\nu} a'\,(\overline{a} a' \mid R(n-1, L \cup \{a'\}))) \mid a \in L\} \\
&\quad \oplus \{\overline{c_n^{a}} \oplus a(x).\,(\mathcal{E}(a', x) \mid R(n-1, L \cup \{a'\})) \mid a \in L\}.
\end{aligned}$$

The proof is by induction on n by using Lemma 3.12. Details can be found in [11]. ■

Finally, we prove that barbed congruence is contained in synonymous bisimilarity.

Theorem 3.14 (Completeness) *Let P and Q be two image finite processes in $\pi_{\rm a}$. Then*

$$P \cong_{\pi_{\rm a}} Q \text{ implies } P \approx_{\varepsilon} Q$$

Proof. The proof follows by Lemma 3.13. ■

4 Encoding external mobility in terms of internal mobility

The mobility mechanism of π-calculus can be divided into *internal* mobility and *external* mobility [20]. The former arises when an input meets the output of a private name, the latter when an input meets the output of a free name. In [20] Sangiorgi introduces a subcalculus of π, called πI, where only private names may be emitted. He also claims that "internal mobility is responsible for much of the expressiveness of the π-calculus, whereas external mobility is responsible for many of the semantic complications".

In [3] Boreale shows that external mobility can be "programmed" via the internal one. He gives an encoding from $\pi_{\rm a}$ to πI which is adequate w.r.t. barbed bisimulation. This result is not fully satisfactory because barbed bisimulation is an extremely coarse relation, too coarse to be considered, per se, as an important behavioral equivalence (for instance, it is not even preserved by parallel composition). Boreale leaves as an open problem whether the encoding is fully abstract for some reasonable behavioral equivalence.

Boreale's encoding is composed by two encodings. The former maps $\pi_{\rm a}$ into Lπ, a variant of $\pi_{\rm a}$ where only the *output capability* of names may be transmitted [12]. The latter maps Lπ into πI. The adequacy of the whole encoding is derived by proving the adequacy of the two encodings separately. Unfortunately, as showed in [21,12], neither encoding is fully abstract w.r.t. barbed congruence.

In Subsection 3.1 we have introduced an encoding $[\![\cdot]\!]$, from $\pi_{\rm a}$ to the asynchronous variant of πI, which is inspired by and generalizes Boreale's second encoding. In this section we prove that $[\![\cdot]\!]$ is fully abstract w.r.t. barbed congruence. So, at least in an asynchronous scenario, we make precise and validate Sangiorgi's claim.

The syntax of asynchronous πI, in short $\pi{\rm I}_{\rm a}$, has operators of inaction, input prefix, asynchronous bound output, parallel composition, restriction, and recursive definitions.

$$P ::= \mathbf{0} \;\Big|\; a(x).\,P \;\Big|\; \boldsymbol{\nu} b\,(\overline{a}b \mid P) \;\Big|\; P \mid P \;\Big|\; \boldsymbol{\nu} a\, P \;\Big|\; A(a_1, \dots , a_n)$$

The lemma below will be crucial for proving the desired full-abstraction result.

Lemma 4.1 *Let P be a process in π_a. Then*

$$P \approx_\varepsilon [\![P]\!].$$

Proof. The proof is by structural induction on P. The only interesting case is when $P \equiv \overline{a}b$. The proof follows by applying clause iii(b) (or iv(b)) of the definition of synonymous bisimulation. ∎

Theorem 4.2 (Full abstraction of $[\![\cdot]\!]$) *Let P and Q be two processes in π_a. Then*

$$P \cong_{\pi_\mathrm{a}} Q \;\text{ iff }\; [\![P]\!] \cong_{\pi\mathrm{I}_\mathrm{a}} [\![Q]\!]$$

Proof. ($\Longleftarrow$) Since $\pi\mathrm{I}_\mathrm{a} \supset \mathcal{P}_\mathrm{a}$ it holds that $[\![P]\!] \cong_{\mathcal{P}_\mathrm{a}} [\![Q]\!]$. By Lemma 3.9 and the compositionality of $[\![\cdot]\!]$ we have that $P \cong_{\pi_\mathrm{a}} Q$.
($\Longrightarrow$) By Lemma 4.1 and Theorem 3.11 we have $[\![P]\!] \cong_{\pi_\mathrm{a}} [\![Q]\!]$. Since $\pi\mathrm{I}_\mathrm{a} \subset \pi_\mathrm{a}$ it holds that $[\![P]\!] \cong_{\pi\mathrm{I}_\mathrm{a}} [\![Q]\!]$.
∎

The theorem above shows that in asynchronous π-calculus external mobility can be expressed in terms of internal mobility. Note that the encoding introduces divergence, so by no means it should be regarded as an implementation.

Acknowledgments
I benefited from discussions with by Gérard Boudol, Silvano Dal-Zilio, and Davide Sangiorgi. I thank Ilaria Castellani and António Ravara for detailed comments on a previous draft of this paper. I also thank the anonymous referees for helpful comments.

References

[1] Amadio, R., I. Castellani and D. Sangiorgi, *On bisimulations for the asynchronous π-calculus*, Theoretical Computer Science **195** (1998), pp. 291–324.

[2] Arun-Kumar, S. and M. Hennessy, *An efficiency preorder for processes*, Acta Informatica **29** (1992), pp. 737–760.

[3] Boreale, M., *On the expressiveness of internal mobility in name-passing calculi*, Theoretical Computer Science **195** (1998), pp. 205–226.

[4] Boreale, M. and D. Sangiorgi, *Bisimulation in name-passing calculi without matching*, in: *13th LICS Conf.* (1998).

[5] Boudol, G., *Asynchrony and the π-calculus*, Technical Report RR-1702, INRIA-Sophia Antipolis (1992).

[6] Fournet, C. and G. Gonthier, *A Hierarchy of Equivalences for Asynchronous Calculi*, in: *25th ICALP*, Lecture Notes in Computer Science **1443** (1998).

[7] Honda, K., *Two bisimilarities for the ν-calculus*, Technical Report 92-002, Keio University (1992).

[8] Honda, K. and M. Tokoro, *An Object Calculus for Asynchronous Communications*, in: M. Tokoro, O. Nierstrasz, P. Wegner and A. Yonezawa, editors, *ECOOP '91 Workshop on Object Based Concurrent Programming,* Geneva, Switzerland, 1991, Lecture Notes in Computer Science **512** (1991), pp. 133–147.

[9] Honda, K. and N. Yoshida, *On reduction-based process semantics*, Theoretical Computer Science **152** (1995), pp. 437–486.

[10] Merro, M., *On the Expressiveness of Chi, Update and Fusion calculi*, in: *Proc. Express'98*, Electronic Notes in Theoretical Computer Science **16.2** (1998).

[11] Merro, M., *On Equators in Asynchronous Name-passing calculi without Matching*, Technical Report available at http://www-sop.inria.fr/meije/personnel/Massimo.Merro.html.

[12] Merro, M. and D. Sangiorgi, *On asynchrony in name-passing calculi*, in: *25th ICALP*, Lecture Notes in Computer Science **1443** (1998).

[13] Milner, R., *The polyadic π-calculus: a tutorial*, Technical Report ECS–LFCS–91–180, LFCS, Dept. of Comp. Sci., Edinburgh Univ. (1991), Also in *Logic and Algebra of Specification*, ed. F.L. Bauer, W. Brauer and H. Schwichtenberg, Springer Verlag, 1993.

[14] Milner, R., J. Parrow and D. Walker, *A calculus of mobile processes, (Parts I and II)*, Information and Computation **100** (1992), pp. 1–77.

[15] Milner, R. and D. Sangiorgi, *Barbed bisimulation*, in: W. Kuich, editor, *19th ICALP*, Lecture Notes in Computer Science **623** (1992), pp. 685–695.

[16] Parrow, J. and B. Victor, *The fusion calculus: Expressiveness and symmetry in mobile processes*, in: *13th LICS Conf.*, 1998.

[17] Pierce, B. C. and D. N. Turner, *Pict: A programming language based on the pi-calculus.*, Technical Report CSCI 476, Indiana University (1997), to appear in *Proof, Language and Interaction: Essays in Honour of Robin Milner*, Gordon Plotkin, Colin Stirling, and Mads Tofte, editors, MIT Press.

[18] Sangiorgi, D., "Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms," PhD thesis CST–99–93, Department of Computer Science, University of Edinburgh (1992).

[19] Sangiorgi, D., *Lazy functions and mobile processes*, Technical Report RR-2515, INRIA-Sophia Antipolis (1995), to appear in *Proof, Language and Interaction: Essays in Honour of Robin Milner*, Gordon Plotkin, Colin Stirling, and Mads Tofte, editors, MIT Press.

[20] Sangiorgi, D., *π-calculus, internal mobility and agent-passing calculi*, Theoretical Computer Science **167** (1996), pp. 235–274.

[21] Yoshida, N., *Minimality and Separation Results on Asynchronous Mobile Processes: representability theorem by concurrent combinators*, in: *9th CONCUR*, Lecture Notes in Computer Science **1466** (1998).