

# On Barbs and Labels in Reactive Systems\*

Filippo Bonchi

Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands

Filippo.Bonchi@cwi.nl

Fabio Gadducci

Dipartimento di Informatica, Università di Pisa, Italy

fabio@di.unipi.it

Giacoma Valentina Monreale

Dipartimento di Informatica, Università di Pisa, Italy

vale@di.unipi.it

Reactive systems (RSs) represent a meta-framework aimed at deriving behavioral congruences for those computational formalisms whose operational semantics is provided by reduction rules. RSs proved a flexible specification device, yet so far most of the efforts dealing with their behavioural semantics focused on idem pushouts (IPOs) and saturated (also known as dynamic) bisimulations. In this paper we introduce a novel, intermediate behavioural equivalence: *L-bisimilarity*, which is able to recast both its IPO and saturated counterparts. The equivalence is parametric with respect to a set  $L$  of RSs labels, and it is shown that under mild conditions on  $L$  it is indeed a congruence. Furthermore, *L-bisimilarity* can also recast the notion of barbed semantics for RSs, proposed by the same authors in a previous paper. In order to provide a suitable test-bed, we instantiate our proposal by addressing the semantics of (asynchronous) CCS and of the calculus of mobile ambients.

## 1 Introduction

*Reactive systems* (RSs) [12] are an abstract formalism for specifying the dynamics of a computational device. Indeed, the usual specification technique is based on a reduction system, comprising a set of possible states of the device and a relation among them, representing the possible evolutions of the device. The relation is often given inductively, freely instantiating relatively few rewriting rules: despite its ease of use, the main drawback of reduction-based solutions is poor compositionality, since the dynamic behaviour of arbitrary stand-alone terms can be interpreted only by inserting them in appropriate contexts, where a reduction may take place. The theoretical appeal of RSs is their ability to distill labelled transition systems (LTSs), hence, behavioural equivalences, for devices specified by a reduction system.

The idea underlying RSs is simple: whenever a device specified by a term  $C[P]$  (i.e., a sub-term  $P$  inserted into a unary context  $C[-]$ ) may evolve to a state  $Q$ , the associated LTS has a transition  $P \xrightarrow{C[-]} Q$  (i.e., the state  $P$  evolves into  $Q$  with a label  $C[-]$ ). If all contexts are admitted, the resulting semantics is called saturated, and the standard bisimilarity on the derived LTS is a congruence. However, it is unfeasible to check the bisimulation game under all contexts, and usually it suffices to consider a subset of contexts that guarantees that the distilled behavioural semantics is a congruence. Such a set, the “minimal” contexts allowing a reduction to occur, was identified in [12] by the notion of *relative pushout*: the resulting strong bisimilarity is a congruence, even if it often does not coincide with the saturated one.

Several attempts have been made to encode various specification formalisms (Petri nets [15, 20], logic programming [6], etc.) as RSs, either hoping to recover the standard observational equivalences, whenever such a behavioural semantics exists (CCS [13], pi-calculus [14], etc.), or trying to distill a

---

\*Research partly supported by the EU within the FP6-IST IP 16004 SENSORIA (*Software Engineering for Service-Oriented Overlay Computers*) and carried out during the first author’s tenure of an ERCIM “Alain Bensoussan” Fellowship Programme.

meaningful new semantics. The results are often not fully satisfactory: bisimilarity via minimal contexts is usually too fine-grained; while saturated semantics are often too coarse (the standard CCS strong bisimilarity is e.g. strictly included in the saturated one). As for process calculi, the standard way out of the impasse is to consider *barbs* [16] (i.e., predicates on the states of a system) and barbed equivalences (i.e., adding the check of such predicates in the bisimulation game). The flexibility of the definition allows for recasting a variety of observational, bisimulation-based equivalences. Indeed, the methodological contribution of [5] is the introduction of suitable notions of barbed saturated semantics for RSs.

In this paper we move one step further, and we propose a novel behavioural equivalence for RSs, namely,  $L$ -bisimulation: a flexible tool, parametric with respect to a set of minimal labels  $L$ . Also in this case the idea is very simple, and it just asymmetrically refines the standard bisimulation game. If the minimal LTS has a transition  $P \xrightarrow{C[-]} Q$ , then a bisimilar  $P'$  has to react via a minimal transition  $P' \xrightarrow{C[-]} Q'$ , whenever  $C[-] \in L$ ; or it must ensure that  $C[P']$  may evolve into  $Q'$  (thus requiring no minimality for  $C[-]$  with respect to  $P'$ ), otherwise. The associated bisimilarity is intermediate between the standard semantics (i.e., minimal and saturated) for RSs: indeed, it is able to recover both of them, by simply varying the set  $L$  and exploiting the so-called semi-saturated semantics. It can be proved that, under mild closure conditions on the set  $L$ ,  $L$ -bisimilarity is a congruence; and moreover, it can be shown that barbed saturated semantics can be recast, as long as  $L$  satisfies suitable barb-capturing properties.

With respect to barbed saturated semantics,  $L$ -bisimilarity admits a streamlined definition, where state predicates play no role. It is thus of simpler verification, and its introduction may have far reaching consequences over the usability of the RS formalism. However, as for any newly proposed semantics, its adequacy and ease of use have to be tested against suitable case studies. We thus consider a recently introduced, minimal context semantics for mobile ambients (MAs), as distilled in [4]; as well as two minimally labelled transition systems for CCS and its asynchronous variant, reminiscent of those proposed in [3]. We show that in those cases, a set  $L$  of minimal labels can be identified, such that  $L$ -bisimilarity precisely captures the standard semantics of the calculus at hand.

The paper is organized as follows. Section 2 recalls the basic notions of RSs, while Section 3 and Section 4 perform the same for MAs and (asynchronous) CCS, respectively. Section 5 presents the technical core of the paper: the introduction of  $L$ -bisimilarity for RSs, the proof that (under mild conditions on  $L$ ) it is indeed a congruence, and moreover its correspondence with barbed semantics. Finally, Section 6 and Section 7 prove that, suitably varying the set  $L$ , the newly defined  $L$ -bisimilarity captures the standard equivalences for MAs and for CCS and its asynchronous variant, respectively.

## 2 Reactive Systems

This section summarizes the main results concerning (the theory of) reactive systems (RSs) [12]. The formalism aims at deriving labelled transition systems (LTSs) and bisimulation congruences for a system specified by a reduction semantics, and it is centered on the concepts of *term*, *context* and *reduction rule*: contexts are arrows of a category, terms are arrows having as domain  $0$ , a special object that denotes groundness, and reduction rules are pairs of (ground) terms.

**Definition 1** (Reactive System). *A reactive system  $\mathbb{C}$  consists of*

1. *a category  $\mathbf{C}$ ;*
2. *a distinguished object  $0 \in |\mathbf{C}|$ ;*
3. *a composition-reflecting subcategory  $\mathbf{D}$  of reactive contexts;*
4. *a set of pairs  $\mathfrak{R} \subseteq \bigcup_{I \in |\mathbf{C}|} \mathbf{C}(0, I) \times \mathbf{C}(0, I)$  of reduction rules.*

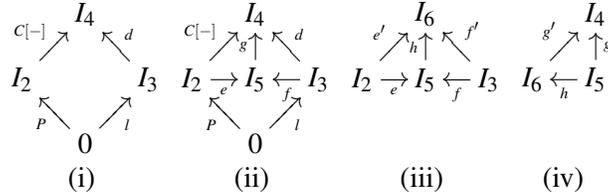


Figure 1: Redex Square and RPO

Intuitively, reactive contexts are those in which a reduction may occur. By composition-reflecting we mean that  $d' \circ d \in \mathbf{D}$  implies  $d, d' \in \mathbf{D}$ . Note that the rules have to be ground, i.e., left-hand and right-hand sides have to be terms without holes and, moreover, with the same codomain.

The reduction relation is generated from the reduction rules by closing them under all reactive contexts. Formally, the *reduction relation* is defined by taking  $P \rightsquigarrow Q$  if there is  $\langle l, r \rangle \in \mathfrak{R}$  and  $d \in \mathbf{D}$  such that  $P = d \circ l$  and  $Q = d \circ r$ .

Thus the behaviour of an RS is expressed as an unlabelled transition system. In order to obtain a LTS, we can plug a term  $P$  into some context  $C[-]$  and observe if a reduction occurs. In this case we have that  $P \xrightarrow{C[-]}$ . Categorically speaking, this means that  $C[-] \circ P$  matches  $d \circ l$  for some rule  $\langle l, r \rangle \in \mathfrak{R}$  and some reactive context  $d$ . This situation is formally depicted by diagram (i) in Fig. 1: a commuting diagram like this is called a *redex square*.

**Definition 2** (Saturated Transition System). *The saturated transition system (STS) is defined as follows*

- *states: arrows  $P : 0 \rightarrow I$  in  $\mathbf{C}$ , for arbitrary  $I$ ;*
- *transitions:  $P \xrightarrow{C[-]}_{SAT} Q$  if  $C[P] \rightsquigarrow Q$ .*

Note that  $C[P]$  stands for  $C[-] \circ P$ : the same notation is used in Definitions 3 and 7 below, in order to allow for an easier comparison with the process calculi notation, to be adopted in the following sections.

**Definition 3** (Saturated Bisimulation). *A symmetric relation  $\mathcal{R}$  is a saturated bisimulation if whenever  $P \mathcal{R} Q$  then  $\forall C[-]$*

- *if  $C[P] \rightsquigarrow P'$  then  $C[Q] \rightsquigarrow Q'$  and  $P' \mathcal{R} Q'$ .*

Saturated bisimilarity  $\sim^S$  is the largest saturated bisimulation.

It is obvious that  $\sim^S$  is a congruence. Indeed, it is the coarsest symmetric relation satisfying the bisimulation game on  $\rightsquigarrow$  that is also a congruence.

Note that STS is often infinite-branching since all contexts allowing reductions may occur as labels. Moreover, it has intuitively redundant transitions. For example, consider the term  $a.0$  of CCS. We have both the transitions  $a.0 \xrightarrow{\bar{a}.0| -}_{SAT} 0|0$  and  $a.0 \xrightarrow{P|\bar{a}.0| -}_{SAT} P|0|0$ , yet  $P$  does not “concur” to the reduction. We thus need a notion of “minimal context allowing a reduction”, captured by *idem pushouts*.

**Definition 4** (RPO, IPO). *Let the diagrams in Fig. 1 be in a category  $\mathbf{C}$ , and let (i) be a commuting diagram. A candidate for (i) is any tuple  $\langle I_5, e, f, g \rangle$  making (ii) commute. A relative pushout (RPO) is the smallest such candidate, i.e., such that for any other candidate  $\langle I_6, e', f', g' \rangle$  there exists a unique morphism  $h : I_5 \rightarrow I_6$  making (iii) and (iv) commute. A commuting square such as diagram (i) of Fig. 1 is called idem pushout (IPO) if  $\langle I_4, c, d, id_{I_4} \rangle$  is its RPO.*

Hereafter, we say that an RS has redex RPOs (IPOs) if every redex square has an RPO (IPO) as candidate. For a better understanding of these two notions, we refer the reader to [6].

**Definition 5** (IPO Transition System). *The IPO transition system (ITS) is defined as follows*

- *states:  $P : 0 \rightarrow I$  in  $\mathbf{C}$ , for arbitrary  $I$ ;*
- *transitions:  $P \xrightarrow{C[-]}_{IPO} d \circ r$  if  $d \in \mathbf{D}$ ,  $\langle l, r \rangle \in \mathfrak{R}$ , and  $(i)$  in Fig. 1 is an IPO.*

In other words, if inserting  $P$  into the context  $C[-]$  matches  $d \circ l$ , and  $C[-]$  is the “smallest” such context, then  $P$  evolves to  $d \circ r$  with label  $C[-]$ .

Bisimilarity on ITS is referred to as *IPO-bisimilarity* ( $\sim^I$ ). Leifer and Milner have shown that if the RS has redex RPOs, then it is a congruence.

**Proposition 1.** *Let us consider an RS with redex RPOs. Then,  $\sim^I$  is a congruence.*

Clearly,  $\sim^I \subseteq \sim^S$ . In [2] the first author shows that this inclusion is strict for many formalisms. In particular, it turns out that in some interesting cases  $\sim^I$  is too strict, while  $\sim^S$  is too coarse. This fact is the reason for introducing *barbed bisimilarities* [16]. Barbs are predicates (representing some basic observations) on the states of a system. For instance, in [16] the authors use for CCS barbs  $\downarrow_a$  and  $\downarrow_{\bar{a}}$  representing the ability of a process to perform an input, respectively an output, on channel  $a$ .

In the following we fix a family  $O$  of barbs, and we write  $P \downarrow_o$  if  $P$  satisfies  $o \in O$ .

**Definition 6** (Barbed Saturated Bisimulation). *A symmetric relation  $\mathcal{R}$  is a barbed saturated bisimulation if whenever  $P \mathcal{R} Q$  then  $\forall C[-]$*

- *if  $C[P] \downarrow_o$  then  $C[Q] \downarrow_o$ ;*
- *if  $C[P] \rightsquigarrow P'$  then  $C[Q] \rightsquigarrow Q'$  and  $P' \mathcal{R} Q'$ .*

Barbed saturated bisimilarity  $\sim^{BS}$  is the largest barbed saturated bisimulation.

It is easy to see that  $\sim^{BS}$  is the largest barbed bisimulation that is also a congruence.

## 2.1 An Efficient Characterization of (Barbed) Saturated Bisimilarity

Since the definition of saturated bisimulation involves a quantification over all possible contexts, it is usually hard to (automatically) prove the equivalence of two systems. For this reason, the first author, with König and Montanari, introduced *semi-saturated bisimilarity* [6].

**Definition 7** (Semi-Saturated Bisimulation). *A symmetric relation  $\mathcal{R}$  is a semi-saturated bisimulation if whenever  $P \mathcal{R} Q$  then*

- *if  $P \xrightarrow{C[-]}_{IPO} P'$  then  $C[Q] \rightsquigarrow Q'$  and  $P' \mathcal{R} Q'$ .*

Semi-saturated bisimilarity  $\sim^{SS}$  is the largest barbed semi-saturated bisimulation.

**Proposition 2.** *Let us consider an RS with redex IPOs. Then,  $\sim^{SS} = \sim^S$ .*

Reasoning on  $\sim^{SS}$  is easier than on  $\sim^S$  because instead of looking at the reductions in all contexts, only IPO transitions are considered.

In [5], the authors extended this technique to barbed saturated bisimilarity.

**Definition 8** (Barbed Semi-Saturated Bisimulation). *A symmetric relation  $\mathcal{R}$  is a barbed semi-saturated bisimulation if whenever  $P \mathcal{R} Q$  then*

- *$\forall C[-]$ , if  $C[P] \downarrow_o$  then  $C[Q] \downarrow_o$ ;*
- *if  $P \xrightarrow{C[-]}_{IPO} P'$  then  $C[Q] \rightsquigarrow Q'$  and  $P' \mathcal{R} Q'$ .*

Barbed semi-saturated bisimilarity  $\sim^{BSS}$  is the largest barbed semi-saturated bisimulation.

**Proposition 3.** *Let us consider an RS with redex IPOs. Then,  $\sim^{BSS} = \sim^{BS}$ .*

Also in this case, it is more convenient to work with  $\sim^{BSS}$  instead of  $\sim^{BS}$ . Even if barbs are still quantified over all contexts, for many formalisms (as for MAs) it is actually enough to check if  $P \downarrow_o$  implies  $Q \downarrow_o$ , since this condition implies that  $\forall C[-]$ , if  $C[P] \downarrow_o$  then  $C[Q] \downarrow_o$ . Barbs satisfying this property are called *contextual* barbs.

**Definition 9** (Contextual Barbs). *A barb  $o$  is a contextual barb if whenever  $P \downarrow_o$  implies  $Q \downarrow_o$  then  $\forall C[-]$ ,  $C[P] \downarrow_o$  implies  $C[Q] \downarrow_o$ .*

### 3 Mobile Ambients

In this section we first introduce the finite, communication-free fragment of mobile ambients (MAs) [8] and its reduction semantics. Then, we recall the IPO transition system for MAs presented in [4].

Fig. 2 shows the syntax of the calculus. We assume a set  $\mathcal{N}$  of *names* ranged over by  $m, n, u, \dots$ . Besides the standard constructors, we include a set  $\{X, Y, \dots\}$  of *process variables* and a set  $\{x, y, \dots\}$  of *name variables*. We let  $P, Q, R, \dots$  range over the set of *pure* processes, containing neither process nor name variables; while  $P_\varepsilon, Q_\varepsilon, R_\varepsilon, \dots$  range over the set of *well-formed* processes, i.e., such that no process or ambient variable occurs twice.

Intuitively, an impure process such as  $x[P]|X$  represents an underspecified system, where either the process  $X$  or the name of the ambient  $x[-]$  can be further instantiated. These extended processes are needed later for the presentation of the LTS. We use the standard definitions for the set of free names of a pure process  $P$ , denoted by  $fn(P)$ , and for  $\alpha$ -convertibility, with respect to the restriction operators  $(\nu n)$ . We moreover assume that  $fn(X) = \emptyset$  and  $fn(x[P]) = fn(P)$ . We also consider a family of *substitutions*, which may replace a process/name variable with a pure process/name, respectively. Substitutions avoid name capture: for a pure process  $P$ , the expression  $(\nu n)(\nu m)(X|x[\mathbf{0}])\{^m/x, ^{n[P]}/X\}$  corresponds to the pure process  $(\nu p)(\nu q)(n[P]|m[\mathbf{0}])$ , for names  $p, q \notin \{m\} \cup fn(n[P])$ .

The semantics of the calculus exploits a *structural congruence*, denoted by  $\equiv$ , which is the least equivalence on pure processes that satisfies the axioms in Fig. 3. The *reduction relation*, denoted by  $\rightsquigarrow$ , describes the evolution of pure processes. It is the smallest relation closed under the congruence  $\equiv$  and inductively generated by the set of axioms and inference rules in Fig. 4.

As already said, a *barb*  $o$  is a predicate over the states of a system, with  $P \downarrow_o$  denoting that  $P$  satisfies  $o$ . In MAs,  $P \downarrow_n$  denotes the presence at top-level of an unrestricted ambient  $n$ . Formally, for a pure process  $P$ ,  $P \downarrow_n$  if  $P \equiv (\nu A)(n[Q]|R)$  and  $n \notin A$ , for processes  $Q$  and  $R$  and a set of restricted names  $A$ .

**Definition 10** (Reduction Barbed Congruences [18]). *Reduction barbed congruence  $\sim^{MA}$  is the largest symmetric relation  $\mathcal{R}$  such that whenever  $P \mathcal{R} Q$  then*

- if  $P \downarrow_n$  then  $Q \downarrow_n$ ;
- if  $P \rightsquigarrow P'$  then  $Q \rightsquigarrow Q'$  and  $P' \mathcal{R} Q'$ ;
- $\forall C[-], C[P] \mathcal{R} C[Q]$ .

A labelled characterization of the reduction barbed congruence over MAs processes was first presented by Rathke and Sobociński in [18], and then by the authors of this paper in [5]. In the latter we exploit the notion of barbed saturated bisimilarity over RSs that is able to capture the behavioural semantics for MAs defined above. Moreover, we give an efficient characterization of such a semantics through the IPO transition system presented in the next section, and the semi-saturated game.

---


$$P ::= \mathbf{0}, n[P], M.P, (\nu n)P, P_1 | P_2, X, x[P] \qquad M ::= in\ n, out\ n, open\ n$$


---

Figure 2: (Extended) Syntax of mobile ambients.

---

if $P \equiv Q$ then $P R \equiv Q R$	$P \mathbf{0} \equiv P$
if $P \equiv Q$ then $(\nu n)P \equiv (\nu n)Q$	$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$
if $P \equiv Q$ then $n[P] \equiv n[Q]$	$(\nu n)(P Q) \equiv P (\nu n)Q$ if $n \notin fn(P)$
if $P \equiv Q$ then $M.P \equiv M.Q$	$(\nu n)m[P] \equiv m[(\nu n)P]$ if $n \neq m$
$P Q \equiv Q P$	$(\nu n)M.P \equiv M.(\nu n)P$ if $n \notin fn(M)$
$(P Q) R \equiv P (Q R)$	$(\nu n)P \equiv (\nu m)(P\{^m/n\})$ if $m \notin fn(P)$

---

Figure 3: Structural congruence.

---

$n[in\ m.P Q] m[R] \rightsquigarrow m[n[P Q] R]$	if $P \rightsquigarrow Q$ then $(\nu n)P \rightsquigarrow (\nu n)Q$
$m[n[out\ m.P Q] R] \rightsquigarrow n[P Q] m[R]$	if $P \rightsquigarrow Q$ then $n[P] \rightsquigarrow n[Q]$
$open\ n.P n[Q] \rightsquigarrow P Q$	if $P \rightsquigarrow Q$ then $P R \rightsquigarrow Q R$

---

Figure 4: Reduction relation on pure processes.

**An ITS for Mobile Ambients.** Here we present the ITS  $M$  for MAs proposed in [4]. The inference rules describing this LTS are obtained from an analysis of a LTS over (processes as) graphs, derived by the borrowed context mechanism [10], which is an instance of the theory of RSs [21]. The labels of the transitions are unary contexts, i.e., terms of the extended syntax with a hole  $-$ . Note that they are minimal contexts, that is, they represent the exact amount of context needed by a system to react. We denote them by  $C_\varepsilon[-]$ . The formal definition of the LTS is presented in Fig. 5.

The rule TAU represents the  $\tau$ -actions modeling internal computations. Notice that the labels of the transitions are identity contexts composed of just a hole  $-$ , while the resulting states are pure processes.

The other rules in Fig. 5 model the interactions of a process with its environment. Note that both labels and resulting states contain process and name variables. We define the LTS  $M_I$  for processes over the standard syntax of MAs by instantiating all the variables of the labels and of the resulting states.

**Definition 11.** *Let  $P, Q$  be pure processes and let  $C[-]$  be a pure context. Then, we have that  $P \xrightarrow{C[-]}_{M_I} Q$  if there exists a transition  $P \xrightarrow{C_\varepsilon[-]}_M Q_\varepsilon$  and a substitution  $\sigma$  such that  $Q_\varepsilon\sigma \equiv Q$  and  $C_\varepsilon[-]\sigma = C[-]$ .*

In the above definition recall that substitutions replace process variables by pure processes and that they do not capture bound names.

The rule OPEN models the opening of an ambient provided by the environment. In particular, it enables a process  $P$  with a capability  $open\ n$  at top level, for  $n \in fn(P)$ , to interact with a context providing an ambient  $n$  containing some process  $X_1$ . Note that the label  $-|n[X_1]$  of the rule represents the minimal context needed by the process  $P$  for reacting. The resulting state is the process over the extended syntax  $(\nu A)(P_1|X_1|P_2)$ , where  $X_1$  represents a process provided by the environment. Note that the instantiation of the process variable  $X_1$  with a process containing a free name that belongs to the bound names in  $A$  is possible only  $\alpha$ -converting the resulting process  $(\nu A)(P_1|X_1|P_2)$  into a process that does not contain that name among its bound names at top level.

The rule COOPEN instead models an environment that opens an ambient of the process. The rule INAMB enables an ambient of the process to migrate into a sibling ambient provided by the environment,

---

$\text{(TAU)} \frac{P \rightsquigarrow Q}{P \rightarrow Q}$	$\text{(OUT)} \frac{P \equiv (vA)(out\ m.P_1 P_2) \quad m \notin A}{P \xrightarrow{m[x -X_1] X_2}(vA)(m[X_2] x[P_1 P_2 X_1])}$
$\text{(IN)} \frac{P \equiv (vA)(in\ m.P_1 P_2) \quad m \notin A}{P \xrightarrow{x[-X_1] m[X_2]}(vA)m[x[P_1 P_2 X_1] X_2]}$	$\text{(OUTAMB)} \frac{P \equiv (vA)(n[out\ m.P_1 P_2] P_3) \quad m \notin A}{P \xrightarrow{m[-X_1]}(vA)(m[P_3 X_1] n[P_1 P_2])}$
$\text{(INAMB)} \frac{P \equiv (vA)(n[in\ m.P_1 P_2] P_3) \quad m \notin A}{P \xrightarrow{m[-X_1]}(vA)(m[n[P_1 P_2] X_1] P_3)}$	$\text{(OPEN)} \frac{P \equiv (vA)(open\ n.P_1 P_2) \quad n \notin A}{P \xrightarrow{m[-X_1]}(vA)(P_1 P_2 X_1)}$
$\text{(COIN)} \frac{P \equiv (vA)(m[P_1] P_2) \quad m \notin A}{P \xrightarrow{x[in\ m.X_1 X_2]}(vA)(m[x[X_1 X_2] P_1] P_2)}$	$\text{(COOPEN)} \frac{P \equiv (vA)(n[P_1] P_2) \quad n \notin A}{P \xrightarrow{m[-open\ n.X_1]}(vA)(P_1 X_1 P_2)}$

---

Figure 5: The LTS  $M$ .

while in the rule IN both ambients are provided by the environment. In the rule COIN an ambient provided by the environment enters an ambient of the process. The rule OUTAMB models an ambient of the process exiting from an ambient provided by the environment, while in the rule OUT both ambients are provided by the environment.

## 4 On Synchronous and Asynchronous CCS

This section introduces the ITSs for CCS and for its asynchronous variant. For the sake of space, we do not present the standard CCS, while we indeed recall the syntax and the semantics of Asynchronous CCS (ACCS). We then show an ITS for both CCS and ACCS: the former was introduced in [3], while the latter is original. Finally, we show that the IPO-bisimilarity coincides with the ordinary bisimilarity for CCS; while IPO-bisimilarity is strictly contained in asynchronous bisimilarity.

**Asynchronous CCS.** Differently from synchronous calculi, where messages are simultaneously sent and received, in asynchronous communication the messages are sent and travel through some media until they reach destination. Thus sending is non blocking (i.e., a process may send even if the receiver is not ready to receive), while receiving is (processes must wait until a message becomes available). Observations reflect the asymmetry: since sending is non blocking, receiving is unobservable.

Here we shortly introduce the finite fragment of ACCS. We adopt a presentation reminiscent of asynchronous  $\pi$  [1] that allows the non deterministic choice for input prefixes (a feature missing in [7, 9]).

Fig. 6 shows the syntax of the calculus. We assume a set  $\mathcal{N}$  of *names* ranged over by  $a, b, c, \dots$ . As for MAs, we included a set  $\{X, Y, \dots\}$  of *process variables*. These are needed for the presentation of the LTS in Fig. 11. We let  $P, Q, R, \dots$  range over the set of *pure* processes, containing no process variables. Substitution of process variables is defined analogously to MAs. Note that here we let  $M, N, O, \dots$  range over the set of summation, while in MAs we used those symbols for capabilities.

The main difference with respect to the standard CCS [13] is the absence of output prefixes. The occurrence of an unguarded  $\bar{a}$  indicates a message that is available on some communication media named  $a$ , and it disappears whenever it is received.

$$P ::= M, X, \bar{a}, (\nu a)P, P_1|P_2 \qquad M ::= \mathbf{0}, \tau.P, a.P, M_1 + M_2$$

Figure 6: (Extended) Syntax of Asynchronous CCS.

if $P \equiv Q$ then $P R \equiv Q R$	$P \mathbf{0} \equiv P$
if $P \equiv Q$ then $(\nu a)P \equiv (\nu a)Q$	$(\nu a)(\nu b)P \equiv (\nu b)(\nu a)P$
if $P \equiv Q$ then $\tau.P \equiv \tau.Q$	$(\nu a)(P Q) \equiv P (\nu a)Q$ if $a \notin fn(P)$
if $P \equiv Q$ then $a.P \equiv a.Q$	$M + N \equiv N + M$
if $M \equiv N$ then $M + O \equiv N + O$	$(M + N) + O \equiv M + (N + O)$
$P Q \equiv Q P$	$M + \mathbf{0} \equiv M$
$(P Q) R \equiv P (Q R)$	$(\nu a)P \equiv (\nu b)(P\{b/a\})$ if $b \notin fn(P)$

Figure 7: Structural congruence.

$(a.P + M) \bar{a} \rightsquigarrow P$	if $P \rightsquigarrow Q$ then $(\nu a)P \rightsquigarrow (\nu a)Q$
$\tau.P + M \rightsquigarrow P$	if $P \rightsquigarrow Q$ then $P R \rightsquigarrow Q R$

Figure 8: Reduction relation on pure processes.

$a.P + M \xrightarrow{a} P$	if $P \xrightarrow{\mu} Q$ then $(\nu a)P \xrightarrow{\mu} (\nu a)Q$ if $a \notin n(\mu)$
$\tau.P + M \xrightarrow{\tau} P$	if $P \xrightarrow{\mu} Q$ then $P R \xrightarrow{\mu} Q R$
$\bar{a} \xrightarrow{\bar{a}} \mathbf{0}$	if $P \xrightarrow{a} P_1$ and $Q \xrightarrow{\bar{a}} Q_1$ then $P Q \xrightarrow{\tau} P_1 Q_1$

Figure 9: Labelled transition system.

*Structural equivalence* ( $\equiv$ ) is the smallest congruence induced by the axioms in Fig. 7. The behaviour of a process  $P$  is then described as a reaction relation ( $\rightsquigarrow$ ) over processes up to  $\equiv$ , obtained by closing the rules in Fig. 8. For ACCS, there exists also an interactive semantics expressed by an LTS. This is the transition relation over processes up to  $\equiv$ , obtained by the rules in Fig. 9. Here we use  $\mu$  to range over the set of labels  $\{\tau, a, \bar{a} \mid a \in \mathcal{N}\}$ . The names of  $\mu$ , denoted by  $n(\mu)$ , are defined as usual.

The main difference with respect to the synchronous calculus lies in the notion of *observation*. Since sending messages is non-blocking, an external observer can just send messages to a system without knowing if they will be received or not. For this reason receiving should not be observable and thus barbs take into account only outputs. Formally,  $P \downarrow \bar{a}$  if there exists process  $Q$  such that  $P \xrightarrow{\bar{a}} Q$ . This is reflected in the notion of asynchronous bisimilarity [1].

**Definition 12** (Asynchronous Bisimulation). *A symmetric relation  $\mathcal{R}$  is an asynchronous bisimulation if whenever  $P \mathcal{R} Q$  then*

- if  $P \xrightarrow{\tau} P'$  then  $Q \xrightarrow{\tau} Q'$  and  $P' \mathcal{R} Q'$ ,
- if  $P \xrightarrow{\bar{a}} P'$  then  $Q \xrightarrow{\bar{a}} Q'$  and  $P' \mathcal{R} Q'$ ,
- if  $P \xrightarrow{a} P'$  then either  $Q \xrightarrow{a} Q'$  and  $P' \mathcal{R} Q'$  or  $Q \xrightarrow{\tau} Q'$  and  $P' \mathcal{R} Q'|\bar{a}$ .

Asynchronous bisimilarity  $\sim^A$  is the largest asynchronous bisimulation.

For example, the processes  $a.\bar{a} + \tau.\mathbf{0}$  and  $\tau.\mathbf{0}$  are asynchronous bisimilar. If  $a.\bar{a} + \tau.\mathbf{0} \xrightarrow{a} \bar{a}$ , then  $\tau.\mathbf{0} \xrightarrow{\tau} \mathbf{0}$  and clearly  $\bar{a} \sim^A \mathbf{0}|\bar{a}$ .

$$\begin{array}{l}
(\text{TAU}) \frac{P \rightsquigarrow Q}{P \twoheadrightarrow Q} \\
(\text{RCV}) \frac{P \equiv (vA)(a.Q+M|R) \quad a \notin A}{P \xrightarrow{-|\bar{a}.X_1} (vA)(Q|R|X_1)} \\
(\text{SND}) \frac{P \equiv (vA)(\bar{a}.Q+M|R) \quad a \notin A}{P \xrightarrow{-|a.X_1} (vA)(Q|R|X_1)}
\end{array}$$

Figure 10: The LTS  $C$ 

$$\begin{array}{l}
(\text{TAU}) \frac{P \rightsquigarrow Q}{P \twoheadrightarrow Q} \\
(\text{RCV}) \frac{P \equiv (vA)(a.Q+M|R) \quad a \notin A}{P \xrightarrow{-|\bar{a}} (vA)(Q|R)} \\
(\text{SND}) \frac{P \equiv (vA)(\bar{a}.Q) \quad a \notin A}{P \xrightarrow{-|a.X_1} (vA)(Q|X_1)}
\end{array}$$

Figure 11: The LTS  $A$ 

**An ITS for CCS.** In [3], the first and the second author together with König derived an ITS for the ordinary CCS by employing the borrowed context mechanism [10].

Fig. 10 shows the LTS  $C$ . The labels of  $C$  are minimal contexts, i.e., they represent the exact amount of context needed by a process to react. The reactive semantics of CCS (denoted by  $\rightsquigarrow$ ) can be found in [14]. Note that both the labels and the resulting states contain the process variable  $X_1$ . For the sake of space, we avoided to report here the (extended) syntax of CCS: this is just the ordinary syntax of CCS, together with process variables (analogously to MAs and ACCS).

Following Definition 11 for MAs, we define the LTS  $C_I$  for processes over the standard syntax by instantiating the process variable of the labels and of the resulting states.

Now let us consider the rule RCV. If a process is ready to receive on some unrestricted channel  $a$ , then an interaction takes place whenever it is embedded in an environment of the shape  $-|\bar{a}.X_1$ <sup>1</sup>. Recall that the instantiation of the process variable  $X_1$  with a process containing a free name that belongs to the bound names in  $A$  is possible only  $\alpha$ -converting the resulting process  $(vA)(Q|R|X_1)$ .

Hereafter we use  $\xrightarrow{\mu}$  (with  $\mu \in \{\tau, a, \bar{a} \mid a \in \mathcal{N}\}$ ) to denote the ordinary LTS of CCS [13]. By comparing the latter with the LTS  $C$ , it is easy to see that  $P \xrightarrow{\tau} Q$  if and only if  $P \twoheadrightarrow Q$ . Moreover  $P \xrightarrow{a} Q$  iff  $P \xrightarrow{-|\bar{a}.X_1} Q|X_1$  and  $P \xrightarrow{\bar{a}} Q$  iff  $P \xrightarrow{-|a.X_1} Q|X_1$ . From these facts, the main result of [3] follows: the ordinary bisimilarity of CCS (denoted by  $\sim^{CCS}$ ) coincides with IPO bisimilarity. Instead, saturated bisimilarity is too coarse: the (recursive) processes  $P = rec_z \tau.z$  and  $P|a.\mathbf{0}$  are e.g. saturated bisimilar.

**An ITS for ACCS.** Following [3], we propose an ITS for ACCS. Fig. 11 shows the LTS  $A$ . The LTS  $A_I$  is defined by instantiating the process variable of the labels and of the resulting states.

The main difference between  $A$  and  $C$  is in the rule RCV: since outputs have no continuation in ACCS, then the process variable  $X_1$  (that occurs in  $C$ ) is not needed in  $A$ .

It is easy to see that also for ACCS there is a close correspondence between the ordinary LTS semantics (in Fig. 9) and  $A$ :  $P \xrightarrow{\tau} Q$  iff  $P \twoheadrightarrow Q$ ,  $P \xrightarrow{a} Q$  iff  $P \xrightarrow{-|\bar{a}} Q$  and  $P \xrightarrow{\bar{a}} Q$  iff  $P \xrightarrow{-|a.X_1} Q|X_1$ .

However, in the asynchronous case, IPO-bisimilarity is too fine grained. Indeed, the processes  $a.\bar{a} + \tau.\mathbf{0}$  and  $\tau.\mathbf{0}$  are asynchronously bisimilar, but they are not IPO-bisimilar. In the next section we will introduce a new semantics for RSs that generalizes both  $\sim^{CCS}$  and  $\sim^A$ .

<sup>1</sup>The LTS derived in [3] slightly differs from  $C$ . Besides dropping some not-engaged transitions (i.e., transitions that do not play any role in the notion of bisimulation), we simplified the labels for SND and RCV: these were, respectively,  $-|\bar{a}.X_1 + M_1$  and  $-|\bar{a}.X_1 + M_1$  for  $M_1$  a summation variable. Since these variables do not occur in the resulting states, they also play no role in the derived bisimilarity, and thus we avoided to consider them in the labels.



*Proof.* In order to prove this theorem we will use the composition and decomposition properties of IPOs, namely Proposition 2.1 and Proposition 2.2 of [12]. We have to prove that if  $P \sim^L Q$  then  $C[P] \sim^L C[Q]$ . We show that  $\mathcal{R} = \{(C[P], C[Q]) \text{ s.t. } P \sim^L Q\}$  is an  $L$ -bisimulation.

Suppose that  $C[P] \xrightarrow{J[-]}_{IPO} P'$ . Then there exists an IPO square like diagram (i) above, where  $\langle L, R \rangle \in \mathfrak{R}$ ,  $D[-] \in \mathbf{D}$  and  $P' = D[R]$ . Since, by hypothesis, the RS has redex RPOs, then we can construct an RPO as the one in diagram (ii) above. In this diagram, the lower square is an IPO, since RPOs are also IPOs (Proposition 1 of [12]). Since the outer square is an IPO and the lower square is an IPO, by IPO decomposition property, it follows that also the upper square is an IPO.

Since  $\mathbf{D}$  is composition-reflecting, then both  $D_1[-]$  and  $D_2[-]$  belong to  $\mathbf{D}$ , and then  $P \xrightarrow{J'[-]}_{IPO} D_1[R]$ . Now there are two cases: either  $J[-] \in L$  or  $J[-] \notin L$ .

If  $J[-] \in L$ , then also  $J'[-] \in L$ , because  $L$  is IPO-closed, by hypothesis. Since  $P \sim^L Q$ , then  $Q \xrightarrow{J'[-]}_{IPO} Q''$  and  $D_1[R] \sim^L Q''$ . This means that there exists an IPO square like the lower square of diagram (iii) above, where  $\langle L', R' \rangle \in \mathfrak{R}$ ,  $E[-] \in \mathbf{D}$  and  $E[R] = Q''$ . Now recall by the previous observation that the upper square of diagram (iii) is also an IPO and then, by IPO composition, also the outer square is an IPO. This means that  $C[Q] \xrightarrow{J'[-]}_{IPO} D_2[Q'']$ . Since  $D_1[R] \sim^L Q''$ , then  $P' = D[R] = D_2[D_1[R]] \mathcal{R} D_2[Q'']$ .

If  $J[-] \notin L$ , then either  $J'[-] \in L$  or  $J'[-] \notin L$ . In both cases, from  $P \xrightarrow{J'[-]}_{IPO} D_1[R]$  we derive that  $J'[Q] \rightsquigarrow Q''$  and  $D_1[R] \sim^L Q''$ . This means that the lower square of diagram (iii) above commutes. Since also the upper square commutes, then also the outer square commutes. This means that  $C[Q] \rightsquigarrow D_2[Q'']$ . Since  $D_1[R] \sim^L Q''$ , then  $P' = D[R] = D_2[D_1[R]] \mathcal{R} D_2[Q'']$ .  $\square$

## 5.1 Barbed Saturated Bisimilarity via $L$ -bisimilarity

Here we show that  $L$ -bisimilarity can also characterize barbed saturated bisimilarity, whenever the family of barbs and the set of labels  $L$  satisfy suitable conditions. This result will be used in later sections in order to show that  $L$ -bisimilarity captures the correct equivalences for MAs and ACCS.

In order to guarantee that  $\sim^L \subseteq \sim^{BS}$ , we need some conditions ensuring that the checking of barbs of  $\sim^{BS}$  is already done in  $\sim^L$  by the labels in  $L$ .

**Definition 15.** Let  $L$  be a set of labels and let  $O$  be a set of barbs. We say that  $L$  is  $O$ -capturing if for each barb  $o$  there exists a label  $C[-] \in L$  such that for each process  $P$ ,  $P \downarrow_o$  if and only if  $P \xrightarrow{C[-]}_{IPO} P'$ .

The next two definitions are needed to ensure that  $\sim^{BS} \subseteq \sim^L$ .

**Definition 16.** Let  $\mathcal{R}$  be a relation and let  $\mathcal{P}(X, Y)$  be a predicate on processes. We say that  $\mathcal{P}(X, Y)$  is stable under  $\mathcal{R}$  if whenever  $P \mathcal{R} Q$  and  $\mathcal{P}(P, P')$  there exists  $Q'$  such that  $\mathcal{P}(Q, Q')$  and  $P' \mathcal{R} Q'$ .

For example, the predicates in Fig. 12 and Fig. 13 are stable under  $\sim^{BS}$ .

**Definition 17.** Let  $\mathcal{R}$  be a relation and let  $C[-]$  be a label. We say that  $C[-]$  is stable under  $\mathcal{R}$  if the predicate  $\mathcal{P}(X, Y) = X \xrightarrow{C[-]}_{IPO} Y$  is stable under  $\mathcal{R}$ .

We can finally state a first correspondence result.

**Proposition 5.** Let us consider an RS with redex RPOs, a set  $O$  of contextual barbs and a set  $L$  of labels. If  $L$  is  $O$ -capturing and its labels are stable under  $\sim^{BS}$ , then  $\sim^{BS}$  coincides with  $\sim^L$ .

*Proof.* In order to prove that  $\sim^{BS} \subseteq \sim^L$ , we show that  $\mathcal{R} = \{(P, Q) \text{ s.t. } P \sim^{BS} Q\}$  is an  $L$ -bisimulation.

Suppose that  $P \xrightarrow{C[-]}_{IPO} P'$ . We have two cases: either  $C[-] \in L$  or  $C[-] \notin L$ . If  $C[-] \in L$ , then  $C[-]$  is stable under  $\sim^{BS}$  and thus, since  $P \sim^{BS} Q$ ,  $Q \xrightarrow{C[-]}_{IPO} Q'$  and  $P' \sim^{BS} Q'$ . For the case that  $C[-] \notin L$ , it is enough to note that, since  $P \xrightarrow{C[-]}_{IPO} P'$ , then  $C[P] \rightsquigarrow P'$ . Since  $P \sim^{BS} Q$ , then  $C[Q] \rightsquigarrow Q'$  and  $P' \sim^{BS} Q'$ .

Now we show that  $\mathcal{R} = \{(P, Q) \text{ s.t. } P \sim^L Q\}$  is a barbed semi-saturated bisimulation (i.e.,  $\sim^L \subseteq \sim^{BS}$ ) and thus, since the RS has redex IPOs, by Proposition 3 it follows that  $\sim^L \subseteq \sim^{BS}$ .

At first, we note that, since  $O$  is a set of contextual barbs, in order to show that  $\mathcal{R}$  satisfies the first condition of Definition 8 it suffices to show that  $P \downarrow_o$  implies  $Q \downarrow_o$ . Since  $L$  is  $O$ -capturing, if  $P \downarrow_o$  then there is a label  $C[-] \in L$  such that  $P \downarrow_o$  if and only if  $P \xrightarrow{C[-]}_{IPO}$ . Since  $P \sim^L Q$ , then also  $Q \xrightarrow{C[-]}_{IPO}$  and  $Q \downarrow_o$ .

In order to prove the second condition of Definition 8, it is enough to note that if  $P \xrightarrow{C[-]}_{IPO} P'$  then, for either  $C[-] \in L$  or  $C[-] \notin L$ ,  $C[Q] \rightsquigarrow Q'$  with  $P' \sim^L Q'$ .  $\square$

As a corollary of the previous definition, we obtain the following property that allows to check whenever IPO-bisimilarity coincides with barbed saturated one.

**Lemma 1.** *Let us consider an RS with redex IPOs and a set  $O$  of contextual barbs. If the set of all labels is  $O$ -capturing and each label is stable under  $\sim^{BS}$ , then  $\sim^L$  coincides with  $\sim^{BS}$ .*

## 6 $L$ -Bisimilarity for Mobile Ambients

This section proposes a new labelled characterization of the reduction barbed congruence for MAs, presented in Section 3. In particular, by using the ITS  $M_I$  (also in Section 3) we define an  $L$ -bisimilarity that captures barbed saturated bisimilarity for MAs, coinciding with reduction barbed congruence.

**Proposition 6** (see [5], Theorem 3). *Reduction barbed congruence over MAs  $\sim^{MA}$  coincides with barbed saturated bisimilarity  $\sim_M^{BS}$ .*

As shown in Section 5.1, we can characterize barbed saturated bisimilarity on a set of contextual barbs  $O$  through the IPO transition system and a set of labels  $L$ . In particular, as required by Proposition 5, the set  $L$  must be  $O$ -capturing and each  $C[-] \in L$  must be stable under the barbed saturated bisimilarity.

We denote by  $O_M$  the set of barbs of MAs, recalling that MAs barbs are contextual barbs [5].

**Proposition 7** (see [5], Proposition 6).  *$O_M$  is a set of contextual barbs.*

Therefore, we can characterize reduction barbed congruence over MAs by instantiating Definitions 13 with the ITS  $M_I$  and a set  $L$  of labels having the two properties said above.

First of all, we find some labels of  $M_I$  that capture the barbs of MAs. This ensures that the checking of barbs of the barbed saturated bisimilarity is done in the  $L$ -bisimilarity by the first condition of its definition. It is easy to note that a MAs process  $P$  observes a unrestricted ambient  $n$  at top-level, in symbols  $P \downarrow_n$ , if and only if it can execute a transition labelled with  $-|open\ n.T_1$  or with  $-|m[in\ n.T_1|T_2]$ . Therefore,  $L$  is  $O_M$ -capturing if it contains at least one kind of these labels. We choose to consider labels of the first type, that is, having the shape  $-|open\ n.T_1$ , for  $n$  ambient name and  $T_1$  pure process.

It is possible to prove that these labels are stable under  $\sim_M^{BS}$ . Therefore, if we consider the set  $L$  defined below, we obtain an  $L$ -bisimilarity for MAs that is able to characterize  $\sim_M^{BS}$ .

**Proposition 8.** *Let  $L_M$  be the set of all labels of the ITS  $M_I$  having the shape  $-|open\ n.T_1$ , for  $n$  ambient name and  $T_1$  pure process. Then,  $L_M$  is  $O_M$ -capturing.*

$$\mathcal{P}^{-|open\ n.T_1}(X, Y) \quad \exists P'' \text{ and } m \notin fn(X) \text{ s.t. } P'' \downarrow_m, C'[X] \rightsquigarrow P'' \rightsquigarrow Y \text{ and } Y \not\downarrow_m \\ \text{with } C'[-] = -|open\ n.(m[\mathbf{0}]|open\ m.T_1)$$

Figure 12: Predicate for the label  $-|open\ n.T_1$ .

*Proof.* We have to show that for each barb  $n \in O_M$  there exists a label  $C[-] \in L_M$  such that for each process  $P$ ,  $P \downarrow_n$  if and only if  $P \xrightarrow{C[-]}_{M_I} P'$ .

It is easy to note that, given a barb  $n \in O_M$ , we have that for each process  $P$ ,  $P \downarrow_n$  if and only if  $P \xrightarrow{-|open\ n.T_1}_{M_I} P'$ , with  $T_1$  pure process. Since we know that  $L_M$  contains all labels having the shape  $-|open\ n.T_1$ , for  $n$  ambient name and  $T_1$  pure process, we can conclude that  $L_M$  is  $O_M$ -capturing.  $\square$

Now, in order to prove that each  $C[-] \in L_M$  is stable under  $\sim_M^{BS}$ , we exploit a predicate such that it is stable under  $\sim_M^{BS}$  and equivalent to the one of Definition 17.

**Lemma 2.** *Let  $\mathcal{P}^{-|open\ n.T_1}(X, Y)$  be the binary predicate on MAs processes shown in Fig. 12, for  $n$  ambient name and  $T_1$  pure process. Then,  $\mathcal{P}^{-|open\ n.T_1}(X, Y)$  is stable under  $\sim_M^{BS}$  and for each  $P$  and  $P'$ ,  $\mathcal{P}^{-|open\ n.T_1}(P, P')$  if and only if  $P \xrightarrow{-|open\ n.T_1}_{M_I} P'$ .*

*Proof.* We begin by proving that the predicate  $\mathcal{P}^{-|open\ n.T_1}(X, Y)$  is stable under  $\sim_M^{BS}$ .

Assume that  $P \sim_M^{BS} Q$  and  $\mathcal{P}^{-|open\ n.T_1}(P, P')$  holds. Since  $\mathcal{P}^{-|open\ n.T_1}(P, P')$  holds, then there exists a process  $P''$  and an ambient  $m$  fresh for  $P$  and  $Q$ , such that  $C'[P] \rightsquigarrow P''$ ,  $P'' \downarrow_m$ ,  $P'' \rightsquigarrow P'$  and  $P' \not\downarrow_m$ , with  $C'[-] = -|open\ n.(m[\mathbf{0}]|open\ m.T_1)$ .

Since  $C'[P] \rightsquigarrow P''$  and  $P \sim_M^{BS} Q$ , then  $C'[Q] \rightsquigarrow Q''$  and  $P'' \sim_M^{BS} Q''$ . Therefore, it is obvious that also  $Q'' \downarrow_m$ . Now, we know that  $P'' \rightsquigarrow P'$ , hence we can say that  $Q'' \rightsquigarrow Q'$  and  $P' \sim_M^{BS} Q'$ . From this follows that, since  $P' \not\downarrow_m$ , then also  $Q' \not\downarrow_m$ . So, we can conclude that  $\mathcal{P}^{-|open\ n.T_1}(Q, Q')$  holds, hence  $\mathcal{P}^{-|open\ n.T_1}(X, Y)$  is stable under  $\mathcal{R}$ .

Now we show that for each  $P$  and  $P'$ ,  $\mathcal{P}^{-|open\ n.T_1}(P, P')$  iff  $P \xrightarrow{-|open\ n.T_1}_{M_I} P'$ .

Assume that  $\mathcal{P}^{-|open\ n.T_1}(P, P')$  holds. This means that there exists a process  $P''$  and an ambient  $m$  fresh for  $P$ , such that  $C'[P] \rightsquigarrow P''$ ,  $P'' \downarrow_m$ ,  $P'' \rightsquigarrow P'$  and  $P' \not\downarrow_m$ , with  $C'[-] = -|open\ n.(m[\mathbf{0}]|open\ m.T_1)$ . The fact that  $C'[P] \rightsquigarrow P''$  and  $P'' \downarrow_m$  means that the capability  $open\ n$  has been executed, hence there must be a unrestricted ambient  $n$  at top-level of  $P$ , i.e.,  $P \equiv (\nu A)(n[P_1]|P_2)$  and  $n \notin A$ . From this follows that  $P'' = (\nu A)(P_1|P_2)|m[\mathbf{0}]|open\ m.T_1$ , and since  $P' \not\downarrow_m$ , then  $P' \equiv (\nu A)(P_1|P_2)|T_1$ . Moreover, by knowing that  $P = (\nu A)(n[P_1]|P_2)$  and  $n \notin A$ , we can conclude that  $P \xrightarrow{-|open\ n.T_1}_{M_I} P'$ .

Assume that  $P \xrightarrow{-|open\ n.T_1}_{M_I} P'$ . This means that  $P \equiv Q$ , where  $Q = (\nu A)(n[P_1]|P_2)$ ,  $n \notin A$  and  $P' = (\nu A)(P_1|P_2)|T_1$ . We consider the context  $C'[-] = -|open\ n.(m[\mathbf{0}]|open\ m.T_1)$  with  $m \notin fn(P)$ . It is easy to note that  $C'[Q] \rightsquigarrow P''$  s.t.  $P'' = (\nu A)(P_1|P_2)|m[\mathbf{0}]|open\ m.T_1$  and  $P'' \downarrow_m$ . Therefore, since  $C'[P] \equiv C'[Q]$ , we also have that  $C'[P] \rightsquigarrow P''$ . Now, we can note that  $P'' \rightsquigarrow P'$  and, since  $m$  is fresh for  $P$ ,  $P' \not\downarrow_m$ .  $\square$

**Proposition 9.** *All labels in  $L_M$  are stable under  $\sim_M^{BS}$ .*

The proof of the proposition above trivially follows from Lemma 2.

We finally introduce the main characterization proposition.

**Proposition 10.**  $\sim_M^{BS} = \sim_{L_M}$ .

*Proof.* First of all, by Proposition 7, we know that MAs barbs are contextual. Moreover, by Propositions 8 and 9, we know that  $L$  is  $O_M$ -capturing and it contains only labels that are stable under  $\sim_M^{BS}$ . Therefore, thanks to Proposition 5, we can conclude that  $\sim_M^{BS} = \sim^{L_M}$ .  $\square$

The  $L$ -bisimilarity  $\sim^{L_M}$  presented above is not the only one which is able to characterize barbed saturated bisimilarity  $\sim_M^{BS}$ . For example, as said before, we can choose to consider all labels of the shape  $-|m[in\ n.T_1|T_2]$ : besides being able to capture MAs barbs, they are also stable under  $\sim_M^{BS}$ . However, generally, we can consider the sets  $L$  containing at least all the labels of the shape  $-|open\ n.T_1$  or  $-|m[in\ n.T_1|T_2]$  to capture barbs, and other labels of  $M_I$  that are stable under  $\sim_M^{BS}$ , i.e., labels such that it is possible to define a predicate analogous to the one we defined for the labels  $-|open\ n.T_1$ .

## 7 $L$ -Bisimilarity for (Asynchronous) CCS

Section 4 has shown that IPO-bisimilarity coincides with the ordinary bisimilarity of CCS ( $\sim^{CCS}$ ), while it is strictly contained in asynchronous bisimilarity. In this section, we first show that  $L$ -bisimilarity generalizes both cases and then we prove that these also coincide with their barbed saturated bisimilarities.

**$L$ -Bisimilarity for Asynchronous CCS.** In asynchronous bisimulation (Definition 12), transitions labelled with  $\tau$  and  $\bar{a}$  (corresponding to  $-$  and  $-|a.T_1$  in  $A_I$ , respectively) must be matched by transitions with the same labels. Moreover, when  $P \xrightarrow{a} P'$  (corresponding to  $P \xrightarrow{-|\bar{a}} P'$  in  $A_I$ ) then either  $Q \xrightarrow{a} Q'$  and  $P' \mathcal{R} Q'$  or  $Q \xrightarrow{\tau} Q'$  and  $P' \mathcal{R} Q'|\bar{a}$ . This is equivalent to require that  $Q|\bar{a} \rightsquigarrow Q'$  and  $P' \mathcal{R} Q'$ . Thus, in order to characterize  $\sim^A$  as  $L$ -bisimilarity, it suffices to choose as  $L$  the set of labels corresponding to  $\tau$  and  $\bar{a}$ .

**Proposition 11.** *Let  $L_A$  be the set containing the labels of the ITS  $A_I$  of the shape  $-$  and  $-|a.T_1$ , for a channel name and  $T_1$  pure process. Then,  $\sim^{L_A} = \sim^A$ .*

**$L$ -Bisimilarity for CCS.** Since IPO-bisimilarity coincides with  $\sim^{CCS}$ , in order to characterize  $\sim^{CCS}$  as  $L$ -bisimilarity, it is enough to include all the IPO-labels into  $L$ .

**Proposition 12.** *Let  $L_{CCS}$  be the set containing all the labels of the ITS  $C_I$ . Then,  $\sim^{L_{CCS}} = \sim^{CCS}$ .*

**From  $L$ -Bisimilarity to Barbed Saturated Bisimilarity.** It is important to note that the choice of  $L_{CCS}$  and  $L_A$  is not arbitrary. Indeed, in both cases,  $\sim^{L_{CCS}}$  and  $\sim^{L_A}$  coincide with barbed saturated bisimilarities. This is not a new result, but it is interesting to see that it can be easily proved by following the same approach that we have used for MAs in Section 6.

For the synchronous case, barbs are defined as  $P \downarrow a$  if and only if  $P \xrightarrow{a} Q$  and  $P \downarrow \bar{a}$  if and only if  $P \xrightarrow{\bar{a}} Q$ . Since  $L_{CCS}$  contains the labels  $-|\bar{a}.T_1$  and  $-|a.T_1$  (corresponding to  $a$  and  $\bar{a}$  in the ordinary LTS), then  $L_{CCS}$  is barb capturing.

It is also easy to see that the barbs are contextual. Then, in order to use Proposition 5, we only have to prove that all the labels in  $L_{CCS}$  are stable under barbed congruence. Analogously to MAs, we define some additional predicates. These are shown in Fig. 13. It is easy to see that for each label  $C[-]$ ,  $X \xrightarrow{C[-]} Y$  in  $C_I$  if and only if  $\mathcal{P}^{C[-]}(X, Y)$ . It is also easy to show that all of them are stable under  $\sim^{BS}$ .

For the asynchronous case, recall that  $L_A$  only contains labels of the form  $-$  and  $-|a.T_1$  (corresponding to labels  $\tau$  and  $\bar{a}$  in the ordinary LTS). Since only output barbs  $\downarrow \bar{a}$  are defined, then  $L_A$  is barb capturing. In order to prove that each label in  $L_A$  is stable under  $\sim^{BS}$  we can use for  $-$  and  $-|a.T_1$  the predicates that we have used in the synchronous case (Fig. 13).

---

$\mathcal{P}^{- \bar{a}.T_1}(X, Y)$	$\exists P' \text{ and } i \notin \text{fn}(X) \text{ s.t. } P' \downarrow \bar{i} \text{ and } X \bar{a}.(\bar{i} T_1) i \rightsquigarrow P' \rightsquigarrow Y$
$\mathcal{P}^{- a.T_1}(X, Y)$	$\exists P' \text{ and } i \notin \text{fn}(X) \text{ s.t. } P' \downarrow \bar{i} \text{ and } X a.(\bar{i} T_1) i \rightsquigarrow P' \rightsquigarrow Y$
$\mathcal{P}^{-}(X, Y)$	$X \rightsquigarrow Y$

---

Figure 13: Predicates for CCS

It is worth noting that labels of the form  $-\bar{a}$  are not stable under  $\sim^{BS}$ . Indeed, we cannot adopt the predicate used in the synchronous case (the first in Fig. 13), since outputs have no continuation in ACCS.

## 8 Conclusions and future work

The paper introduces a novel behavioural equivalence for RSs, namely,  $L$ -bisimulation: a flexible tool, parametric with respect to a set of labels  $L$ . The associated bisimilarity is proved to be a congruence, and it is shown to be intermediate between the standard IPO and saturated semantics for RSs: indeed, it is able to recover both of them, by simply varying the set of labels  $L$ . More importantly, also the more expressive barbed saturated semantics can be recast, as long as the set  $L$  satisfies suitable conditions.

As for any newly proposed semantics, its expressiveness and ease of use have to be tested against suitable case studies. We thus considered a recently introduced IPO transition system for MAs, and two other IPO transition systems for CCS and its asynchronous variant. We show that in all those cases, for a right choice of  $L$ ,  $L$ -bisimilarity precisely captures the standard semantics for the calculus at hand.

We can foresee three immediate extensions of our work. First of all, we would like to precisely understand the notion of IPO-closedness, which is required for the set of labels  $L$ , in order for  $L$ -bisimilarity to be a congruence. It would be important to establish suitable and more manageable conditions under which a set of arrows of a given category satisfies that property, especially for those RSs where IPOs have an inductive presentation (such as for those induced by the borrowed context mechanism).

Moreover, we would like to further elaborate on the connection between  $L$ -bisimilarity and barbed semantics, moving beyond the preliminary results presented in Section 5.1. As a start, in order to establish conditions ensuring that barbs satisfy the pivotal property of being contextual; and, more to the point, for checking whenever a set of labels is barb capturing and contains only labels stable under barbed saturated bisimilarity. As far as the specific MAs case study is concerned, most of the IPO labels occurring in our transition system are indeed stable, i.e., the relative labelled transitions can be characterized by a predicate which is stable under the barbed saturated bisimilarity. The only labels that are not stable are the ones of the shape  $-|m[P]$  and  $m[-|P]$  of the rule INAMB and OUTAMB, respectively. It seems intriguing that those same labels required the introduction of so-called Honda-Tokoro inference rules in [18] for capturing the reduction barbed congruence by means of standard bisimilarity.

Finally, we remark that so far in our methodology the choice of the “right” set  $L$ , as well as the identification of a meaningful set of barbs, is left to the ingenuity of the researcher. We would like to devise a general theory that relying only on the syntax of the calculus at hand and on the associated reduction semantics might allow to automatically derive either a suitable family of barbs or some kind of basic set of observations, along the lines of the proposals in [11, 17, 19].

**Acknowledgements.** We are indebted to the anonymous referees for their useful remarks, which helped us in improving the overall presentation of the paper.

## References

- [1] R. Amadio, I. Castellani & D. Sangiorgi (1998): *On Bisimulations for the Asynchronous  $\pi$ -Calculus*. *Theoretical Computer Science* 195(2), pp. 291–324.
- [2] F. Bonchi (2008): *Abstract Semantics by Observable Contexts*. Ph.D. thesis, Department of Informatics, University of Pisa.
- [3] F. Bonchi, F. Gadducci & B. König (2009): *Synthesising CCS Bisimulation Using Graph Rewriting*. *Information and Computation* 207(1), pp. 14–40.
- [4] F. Bonchi, F. Gadducci & G. V. Monreale (2009): *Labelled Transitions for Mobile Ambients (As Synthesized via a Graphical Encoding)*. In: T. Hildebrandt & D. Gorla, editors: *Expressiveness in Concurrency*, *Electr. Notes in Theor. Comp. Sci.* 242(1). Elsevier, pp. 73–98.
- [5] F. Bonchi, F. Gadducci & G. V. Monreale (2009): *Reactive Systems, Barbed Semantics, and the Mobile Ambients*. In: L. de Alfaro, editor: *Foundations of Software Science and Computational Structures*, *Lect. Notes in Comp. Sci.* 5504. Springer, pp. 272–287.
- [6] F. Bonchi, B. König & U. Montanari (2006): *Saturated Semantics for Reactive Systems*. In: *Logic in Computer Science*. IEEE Computer Society, pp. 69–80.
- [7] M. Boreale, R. De Nicola & R. Pugliese (1998): *Asynchronous Observations of Processes*. In: M. Nivat, editor: *Foundations of Software Science and Computation Structures*, *Lect. Notes in Comp. Sci.* 1378. Springer, pp. 95–109.
- [8] L. Cardelli & A. Gordon (2000): *Mobile Ambients*. *Theoretical Computer Science* 240(1), pp. 177–213.
- [9] I. Castellani & M. Hennessy (1998): *Testing Theories for Asynchronous Languages*. In: V. Arvind & R. Ramanujam, editors: *Foundations of Software Technology and Theoretical Computer Science*, *Lect. Notes in Comp. Sci.* 1530. Springer, pp. 90–101.
- [10] H. Ehrig & B. König (2006): *Deriving Bisimulation Congruences in the DPO Approach to Graph Rewriting with Borrowed Contexts*. *Mathematical Structures in Computer Science* 16(6), pp. 1133–1163.
- [11] K. Honda & N. Yoshida (1995): *On Reduction-Based Process Semantics*. *Theoretical Computer Science* 151(2), pp. 437–486.
- [12] J.J. Leifer & R. Milner (2000): *Deriving Bisimulation Congruences for Reactive Systems*. In: C. Palamidessi, editor: *Concurrency Theory*, *Lect. Notes in Comp. Sci.* 1877. Springer, pp. 243–258.
- [13] R. Milner (1989): *Communication and Concurrency*. Prentice Hall.
- [14] R. Milner (1999): *Communicating and Mobile Systems: the  $\pi$ -Calculus*. Cambridge University Press.
- [15] R. Milner (2004): *Bigraphs for Petri Nets*. In: J. Desel, W. Reisig & G. Rozenberg, editors: *Concurrency and Petri Nets*, *Lect. Notes in Comp. Sci.* 3098. Springer, pp. 686–701.
- [16] R. Milner & D. Sangiorgi (1992): *Barbed Bisimulation*. In: W. Kuich, editor: *Automata, Languages and Programming*, *Lect. Notes in Comp. Sci.* 623. Springer, pp. 685–695.
- [17] J. Rathke, V. Sassone & P. Sobociński (2007): *Semantic Barbs and Biorthogonality*. In: H. Seidl, editor: *Foundations of Software Science and Computation Structures*, *Lect. Notes in Comp. Sci.* 4423. Springer, pp. 302–316.
- [18] J. Rathke & P. Sobociński (2008): *Deriving Structural Labelled Transitions for Mobile Ambients*. In: F. van Breugel & M. Chechik, editors: *Concurrency Theory*, *Lect. Notes in Comp. Sci.* 5201. Springer, pp. 462–476.
- [19] J. Rathke & P. Sobociński (2009): *Making the Unobservable, Unobservable*. In: F. Bonchi, D. Grohmann, P. Spoletini, A. Troina & E. Tuosto, editors: *Interaction and Concurrency Experiences*, *Electr. Notes in Theor. Comp. Sci.* 229(3). Elsevier, pp. 131–144.
- [20] V. Sassone & P. Sobociński (2005): *A Congruence for Petri Nets*. In: H. Ehrig, J. Padberg & G. Rozenberg, editors: *Petri Nets and Graph Transformation*, *Electr. Notes in Theor. Comp. Sci.* 127(2). Elsevier, pp. 107–120.
- [21] V. Sassone & P. Sobociński (2005): *Reactive Systems over Cospans*. In: *Logic in Computer Science*. IEEE Computer Society, pp. 311–320.