



- preserves *barbs*, that is preserves some simple observational property of terms.

However, context-based behavioural equalities, such as reduction barbed congruence, suffer from the universal quantification on contexts. This quantification makes very hard to prove process equalities, and makes mechanical checking impossible. Simpler proof techniques are based on *abe ed b s art es* whose definitions do not use context quantification. These bisimilarities should imply, or (better) coincide with, reduction barbed congruence [24, 1, 11]. The behaviour of processes is characterised using co-inductive relations defined over a *abe ed trans t on syste*, or  $L$ , a collection of relations of the form

$$P \longrightarrow Q.$$

Intuitively the action in the judgement  $P \longrightarrow Q$  represents some small context with which  $P$  can interact; if the labelled bisimilarity coincides with the reduction barbed congruence then this collection of small contexts, codified as actions, is sufficient to capture all possible interactions that processes can have with arbitrary contexts.

Even if the idea of bisimulation is very general and does not rely on the specific syntax of the calculus, the definition of an appropriate notion of bisimilarity for Mobile Ambients revealed to be harder than expected. The reasons of that can be resumed as follows:

- It is difficult for an ambient  $n$  to control interferences that may originate either from other ambients in its environment or from the computation running at  $n$  itself, [17].
- Ambient mobility is asynchronous — no permission is required to migrate into an ambient. As noticed in [28], this may cause a *stutter ng* phenomenon originated by ambients that may repeatedly enter and exit another ambient. Any successful bisimilarity for MA should not observe stuttering [28].
- One of the main algebraic laws of MA is the *perfect rewa equat on*, [7]:

$$(\ n)n[P] = \mathbf{0} \quad \text{for } n \text{ not in } P.$$

If you suppose  $P = \text{in}_k.\mathbf{0}$ , it is evident that a bisimilarity that want to capture this law must not observe the movements of *secret a b ents*, that is those ambients, like  $n$ , whose names are not known by the rest of the system.

In [18], it is introduced a labelled bisimilarity for an “easier” variant of MA, called SAP, equipped with (i) *synchronous ob ty*, as in Levi and Sangiorgi’s *afe A b ents* [17], and (ii) *passwords* to exercise control over, and differentiate between, different ambients which may wish to exercise a capability. The main result in [18] is the characterisation of reduction barbed congruence in terms of the labelled bisimilarity. The result holds only in SAP and heavily relies on the two features (i) and (ii) mentioned above.

This work is the natural continuation of [18] where, now, we tackle the original problem: *to prov de b s u at on proof ethods for Mob e A b ents*

**Contribution** First of all, as in the Distributed  $\pi$ -calculus [14], we rewrite the syntax of MA in two levels: *processes* and *systems*. This is because we are interested in studying systems rather than processes. So, our behavioural equalities are defined over systems. This little expedient allows us (i) to focus on higher-order actions, where movement of code is involved, and (ii) to model stuttering in terms of standard  $\pi$ -actions.

We give a new labelled transition system for MA which is used to define a labelled bisimilarity over systems. The resulting bisimilarity can be defined either in *late* or in *early*

---

**Table 1** The Mobile Ambients in Two Levels

---

*Names*  $a, b, \dots, k, l, m, n, \dots$   $\mathbf{N}$

*Systems*

$\mathbf{M}, \mathbf{N} ::= \mathbf{0}$   
          |  $\mathbf{M}_1 \mid \mathbf{M}_2$   
          |  $(n)\mathbf{M}$   
          |  $n$

termination  
parallel composition  
restriction

---

**Table 2 Structural Congruence and Reduction Rules**

---

$P \mid Q \equiv P \mid Q$	(Struct Par Comm)
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	(Struct Par Assoc)
$P \mid \mathbf{0} \equiv P$	(Struct Zero Par)
$(n)\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Res)
$!C.P \equiv C.P \mid !C.P$	(Struct Repl Par)
$(n)(m)P \equiv (m)(n)P$	(Struct Res Res)
$n \text{ fn}(P) \text{ implies } (n)(P \mid Q) \equiv P \mid (n)Q$	(Struct Res Par)
$n = m \text{ implies } (n)(m[P]) \equiv m[(n)P]$	(Struct Res Amb)

$\equiv$  is the least equivalence relation which i) satisfies the axioms and rules above and ii) is preserved by all contexts.

$n[in\_m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$	(Red In)
$m[n[out\_m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$	(Red Out)
$open\_n.P \mid n[Q] \rightarrow P \mid Q$	(Red Open)
$P \equiv Q \text{ and } Q \rightarrow R \text{ implies } P \rightarrow R$	(Red Struct)

$\rightarrow$  is the least equivalence relation which i) satisfies the rules above and ii) is preserved by all static contexts.

---

**Definition 1.2 (Contexts)** A **static context** is a context where the hole does not appear under a prefix or a replication. A **system context** is a context generated by the following

---

**Table 3**

---

**Table 4** Labelled Transition System - Pre-actions

---

$$( \text{ Pfx} ) \frac{-}{.P \longrightarrow P}$$

$$( \text{ Repl Pfx} ) \frac{-}{! .P \longrightarrow P \mid ! .P}$$

$$( \text{ Enter} ) \frac{P \xrightarrow{a.n} P_1}{m[P] \xrightarrow{a.n} \langle m[P_1] \rangle \mathbf{0}}$$

$$( \text{ Amb} ) \frac{-}{n[P] \xrightarrow{-n} \langle P \rangle \mathbf{0}}$$

$$( \text{ Exit} ) \frac{P \xrightarrow{o.t.n} P_1}{m[P] \xrightarrow{o.t.n} \langle m[P_1] \rangle \mathbf{0}}$$

$$( \text{ Res} ) \frac{P \longrightarrow \mathbf{0} \quad n \text{ fn}(\cdot)}{(n)P \longrightarrow (n)\mathbf{0}}$$

$$( \text{ Par} ) \frac{P \longrightarrow \mathbf{0}}{P \mid Q \longrightarrow \mathbf{0} \mid Q}$$

$$Q \mid P \longrightarrow Q \mid \mathbf{0}$$


---

by explicitly introducing the environment's ambient interacting with the process in question.

Pt The content of this ambient will be e P

ecrocg w

---

**Table 5** Labelled Transition System - -actions

---

( Enter) <sup>P</sup>



---

**Table 6** Labelled Transition System - Env-actions

---

$$\text{(Enter)} \frac{P \xrightarrow{\text{nt } r.n} (\tilde{m})\langle k[P_1] \rangle P_2^{(\dagger)}}{P \xrightarrow{k. \text{nt } r.n} (\sim)}$$

$$n[m[\text{out}_n.P_1 \mid P_2] \mid Q] \longrightarrow n[Q \mid m[P_1 \mid P_2]].$$

Again, env-actions can model the exiting of both private and global ambients from an ambient provided by the environment.

Finally, we leave the rules which control the *opening* as an easy exercise for the reader.

We end this section with a theorem which asserts that the LTS-based semantics coincides with the reduction semantics of Section 1.

**Theorem 2.2**

*If  $P \longrightarrow P'$  then  $P \rightarrow P'$*

*If  $P \rightarrow P'$  then  $P \longrightarrow P'$*

**Proof** By transition induction. Part 1 is the most difficult. It requires a result describing the structure of a process  $P$  and the outcome  $O$  for any pre-action

### 3 Characterising Reduction Barbed Congruence

In this section we define a labelled bisimilarity for MA that completely characterises reduction barbed congruence.

Since we are interested in *weak bis-artes*, that abstract over  $\tau$ -actions, we introduce the notion of weak action. The definition is standard:  $\Rightarrow$  denotes the reflexive and transitive closure of  $\longrightarrow$ ;  $\Longrightarrow$  denotes  $\Rightarrow \longrightarrow \Rightarrow$ ;  $\hat{\Rightarrow}$  denotes  $\Rightarrow$  if  $\tau = \tau$  and  $\Longrightarrow$  otherwise.

In the previous section we said that actions (and more precisely env-actions) introduce a special process  $\tau$  to pinpoint those ambients whose content will be instantiated in the bisimilarity. It should be pointed out that we allow structural congruence to rearrange terms containing  $\tau$ : with respect to structural congruence,  $\tau$  behaves like the inactive process  $\mathbf{0}$ . Before defining the bisimilarity we explain how  $\tau$  is instantiated.

**Definition 3.1** *Let  $T, T_1$  and  $T_2$  range over both systems and processes then given a process  $P$  we define*

$$\begin{array}{ll} \mathbf{0} \bullet P & \stackrel{\text{def}}{=} \mathbf{0} & (T_1 \mid T_2) \bullet P & \stackrel{\text{def}}{=} (T_1 \bullet P) \mid (T_2 \bullet P) \\ n[R] \bullet P & \stackrel{\text{def}}{=} n[R \bullet P] & (\nu n)T \bullet P & \stackrel{\text{def}}{=} (\nu n)(T \bullet P) \text{ fn } \text{fn}(P) \\ \tau \bullet P & \stackrel{\text{def}}{=} P & C.R \bullet P & \stackrel{\text{def}}{=} C.(R \bullet P) \\ !C.R \bullet P & \stackrel{\text{def}}{=} !C.(R \bullet P). \end{array}$$

Now, everything is in place to define our bisimilarity.

**Definition 3.2 (Late bisimilarity)** *A symmetric relation  $R$  over systems is a late bisimulation if  $M R N$  implies*

*if  $M \longrightarrow M' \{ \text{.enter}_n, \text{.exit}_n \}$  then there is a system  $N'$  such that  $N \hat{\Rightarrow} N'$  and for a process  $P$  it holds  $M' \bullet P R N' \bullet P$*

*if  $M \xrightarrow{\text{.nt } \tau_n} M'$  then there is a system  $N'$  such that  $N \mid n[\ ] \Rightarrow N'$  and for a process  $P$  it holds  $M' \bullet P R N' \bullet P$*

*if  $M \xrightarrow{\text{.nt } \tau_n} M'$  then there is a system  $N'$  such that  $n[\ ] \mid N \Rightarrow N'$  and for a process  $P$  it holds  $M' \bullet P R N' \bullet P$*

$M$  and  $N$  are late bisimilar written  $M \sim N$  if  $M R N$  for some late bisimulation  $R$

The bisimilarity above has a universal quantification over the process  $P$  provided by the environment. This process instantiates the special process  $\tau$  generated via env-actions. The bisimilarity is defined in a *late* style as the existential quantification precedes the universal one. Another possibility would be to define the bisimilarity in *early* style where the universal quantification over the environment's contribution  $P$  precedes that over the derivative  $N'$ . We write  $\sim_e$  to denote this early variant. By definition, every late bisimulation is also an early one, while the converse, in general, does not hold. However, in our case, as in HO [25], we will prove that late and early bisimilarity actually coincide. As a consequence, late

bisimilarity will be our main labelled bisimilarity because the derivatives  $N'$  do not depend on processes  $P$ .

Finally, notice that, in the definition of bisimilarity, actions  $.\text{enter}_n$  and  $.\text{exit}_n$  are treated apart asking for weaker matching requirements. This is because both actions are not observable. Somehow, this is very similar to what happens with input actions in the asynchronous  $\pi$ -calculus [15, 3].

### 3.1 Soundness

Late and early bisimilarity represent two proof techniques for reduction barbed congruence. More precisely we prove that they are both contextual and contained in reduction barbed congruence.

The following lemma is crucial for proving that  $\approx$  is contextual. This lemma will be also used for proving the soundness of the up-to-context proof techniques in Section 4.

**Lemma 3.3** *Let  $S$  be a contextual symmetric relation between systems. Let  $(M, N) \in S$  be a pair satisfying the bisimulation conditions in  $S$  that is*

*if  $M \xrightarrow{\alpha} M'$  for  $\alpha \in \{.\text{enter}_n, .\text{exit}_n\}$  then there is a system  $N'$  such that  $N \xrightarrow{\alpha} N'$  and for all processes  $P$  it holds  $M' \cdot P \in S N' \cdot P$*

*if  $M \xrightarrow{.\text{nt}_r, n} M'$  then there is a system  $N'$  such that  $N \upharpoonright n \in S N'$  and for all processes  $P$  it holds  $M' \cdot P \in S N' \cdot P$*

*if  $M \xrightarrow{.\text{xt}_n} M'$  then there is a system  $N'$  such that  $n \notin N$  and for all processes  $P$  it holds  $M' \cdot P \in S N' \cdot P$*

*then all the pairs  $(C[M], C[N])$  for any system context  $C[-]$  also satisfy the bisimulation conditions in  $S$*

**Proof** The relation  $S$  is contextual, and as such it is the smallest relation between systems such that:

- if  $M \in S N$ , then  $M \upharpoonright H \in S N \upharpoonright H$  for all systems  $H$ ;
- if  $M \in S N$ , then  $(\nu m)M \in S (\nu m)N$  for all names  $m$ ;
- if  $M \in S N$ , then  $m[M \upharpoonright P] \in S m[N \upharpoonright P]$  for all names  $m$  and processes  $P$ .

We prove the closure of  $C[M] \in S C[N]$  under the conditions for being a bisimulation by induction on the structure of  $C[-]$ .

- $C[-] = -$ .

This case holds because  $M \in S N$

–  $(\ m)D[M] \longrightarrow O_1$ .

This can only be derived from  $D[M] \longrightarrow O_1$ , where  $O_1 = (\ m)O_1$ . The induction hypothesis tells us that there exists a system  $O_2$  such that  $D[N] \Rightarrow O_2$  and  $O_1 \ S \ O_2$ . We can derive  $(\ m)D[N] \Rightarrow (\ m)O_2$  and conclude  $(\ m)O_1 \ S \ (\ m)O_2$  because  $S$  is closed under restriction.

–  $(\ m)D[M] \xrightarrow{k.\ nt\ r.n} O_1$ .

Observe that this must have been derived from

$$\frac{\frac{D[M] \xrightarrow{nt\ r.n} (\ \tilde{r})\langle k[M_1] \rangle M_2}{(\ m)D[M] \xrightarrow{nt\ r.n} (\ m)(\ \tilde{r})\langle k[M_1] \rangle M_2}}{(\ m)D[M] \xrightarrow{k.\ nt\ r.n} O_1 \quad (\ m)(\ \tilde{r})(n[ \ | k[M_1] ] | M_2)}$$

for some process  $M_1$  and system  $M_2$ . Remark that this implies  $m = n$  and  $m = k$ .

Ask.

$(m)N' = O_2$ . We can conclude that for all processes  $P$ , it holds  $O_1 \bullet P \text{ S } O_2 \bullet P$  up to structural congruence, because  $S$  is closed under restriction.

–  $(m)D[M] \xrightarrow{n.\overline{nt} \ r.k} O_1$ .

Observe that this must have been derived from

$$D[M] \xrightarrow{\tau^n} (\tilde{r})\langle$$

$$- (m)D[M] \xrightarrow{\cdot \text{nt } r_n} O_1.$$

Observe that there are two possible derivations.

Suppose:

$$\frac{\frac{D[M] \xrightarrow{\text{nt } r_n} (\tilde{r})\langle m[M_1] \rangle M_2}{(m)D[M] \xrightarrow{\text{nt } r_n} (m)(\tilde{r})\langle m[M_1] \rangle M_2}}{(m)D[M] \xrightarrow{\cdot \text{nt } r_n} O_1} \quad (m)(\tilde{r})(n[ | m[M_1] ] | M_2)$$

where m

where  $m \leq \tilde{r}$ , for some process  $M_1$  and system  $M_2$ . Remark that this implies  $n \leq r$ . As





Suppose:

$$\frac{\frac{D[M] \xrightarrow{\text{t.n}} (\tilde{r})\langle k[M_1] \rangle M_2}{D[M] | H \xrightarrow{\text{t.n}} (\tilde{r})\langle k[M_1] \rangle M_2 | H}}{D[M] | H \xrightarrow{\text{k.t.n}} O_1 \quad (\tilde{r})(n[ \quad | M_2 | H ] | k[M_1])}$$

for some process  $M_1$  and system  $M_2$ . Remark that  $k \tilde{r}$ . As  $D[M] \xrightarrow{\text{t.n}} (\tilde{r})\langle k[M_1] \rangle M_2$  then  $D[M] \xrightarrow{\text{k.t.n}} (\tilde{r})(n[ \quad | M_2 ] | k[M_1]) = M'$ . The induction hypothesis then tells us that there exist systems  $N', A, B$  such that  $D[N] \Rightarrow A \xrightarrow{\text{k.t.n}} B \Rightarrow N'$ , and for all processes  $P$  it holds  $M' \bullet P \leq N'$

Suppose:

$$\frac{\frac{D[M] \xrightarrow{-n} (\tilde{r})\langle M_1 \rangle M_2}{D[M] | H \xrightarrow{-n} (\tilde{r})\langle M_1 \rangle M_2 | H}}{D[M] | H \xrightarrow{n.\overline{nt} \tilde{r}.k} O_1 \quad (\tilde{r})(n[k[] | M_1] | M_2 | H)}$$

for some process  $M_1$  and system  $M_2$ . Remark that  $k, n \tilde{r}$ . As  $D[M] \xrightarrow{-n} (\tilde{r})\langle M_1 \rangle M_2$  then  $D[M] \xrightarrow{n.\overline{nt} \tilde{r}.k} (\tilde{r})(n[k[] | M_1] | M_2) = M'$ . The induction hypothesis then tells us that there exist systems  $N', A, B$  such that  $D[N] \Rightarrow A \xrightarrow{n.\overline{nt} \tilde{r}.k} B \Rightarrow N'$ , and for all processes  $P$  it holds  $M' \cdot P \leq N' \cdot P$ . As  $A \xrightarrow{n.\overline{nt} \tilde{r}.k} B$ , the system  $B$  must be of the form  $(\tilde{s})(n[k[] | N_1] | N_2)$ , for some process  $N_1$  and system  $N_2$ . It also holds  $A \xrightarrow{-n} (\tilde{s})\langle N_1 \rangle N_2$ . This implies  $A | H \xrightarrow{-n} (\tilde{s})\langle N_1 \rangle N_2 | H$ , from which we can derive  $A | H \xrightarrow{n.\overline{nt} \tilde{r}.k} (\tilde{s})(n[k[] | N_1] | N_2 | H) = B | H$ . We obtain  $D[N] | H \Rightarrow A | H \xrightarrow{n.\overline{nt} \tilde{r}.k} B | H \Rightarrow N' | H$ . Call  $N' | H = O_2$ . We can conclude that for all processes  $P$ , it holds  $O_1 \cdot P \leq O_2 \cdot P$  up to structural congruence, because  $\leq$  is closed under parallel composition.

Suppose:

$$\frac{\frac{H \xrightarrow{-n} (\tilde{r})\langle H_1 \rangle H_2}{D[M] | H \xrightarrow{-n} (\tilde{r})\langle H_1 \rangle H_2 | D[M]}}{D[M] | H \xrightarrow{n.\overline{nt} \tilde{r}.k} O_1 \quad (\tilde{r})(n[k[] | H_1] | H_2 | D[M])}$$

for some process  $H_1$  and system  $H_2$ . Remark that  $k \tilde{r}$ . We can construct the following derivation:

$$\frac{\frac{H \xrightarrow{-n} (\tilde{r})\langle H_1 \rangle H_2}{D[N] | H \xrightarrow{-n} (\tilde{r})\langle H_1 \rangle H_2 | D[N]}}{D[N] | H \xrightarrow{n.\overline{nt} \tilde{r}.k} (\tilde{r})(n[k[] | H_1] | H_2 | D[N]) = O_2}$$

We can conclude that for all processes  $P$ , it holds  $O_1 \cdot P \leq O_2 \cdot P$  up to structural congruence, because  $D[M] \leq D[N]$  and  $\leq$  is closed under parallel composition.

Suppose:

$$\frac{\frac{D[M] \xrightarrow{-n} (\tilde{r})(M_1)M_2}{D[M] | H \xrightarrow{-n} (\tilde{r})(M_1)M_2 | H}}{D[M] | H \xrightarrow{k.op \ n.n} O_1 \quad k[ | (\tilde{r})(M_1 | M_2) | H]}$$

for some process  $M_1$  and system  $M_2$ . Remark that  $k, n \ \tilde{r}$ . As  $D[M] \xrightarrow{-n} (\tilde{r})$

Suppose:

$$D[M] \xrightarrow{nt \ r-n} ( \tilde{r} )$$

$n[ \mid D[N] ] \Rightarrow N'$ , and for all processes  $P$  it holds  $M' \cdot P \leq N' \cdot P$ . Remark that  $N' = (\tilde{s})n[ \mid N_3 ] \mid N_4$ , for some  $N_3, N_4$ . We can derive  $n[ \mid D[N] \mid H ] \Rightarrow (\tilde{s})n[ \mid N_3 \mid H ] \mid N_4$ . Call  $(\tilde{s})n[ \mid N_3 \mid H ] \mid N_4 = O_2$ . As for all processes  $P$  it holds  $M' \cdot P \leq N' \cdot P$ , we can conclude that for all processes  $Q$ , it holds  $O_1 \cdot Q \leq O_2 \cdot Q$  up to structural congruence, because  $O_1 \cdot Q = M' \cdot (Q \mid H) \leq N' \cdot (Q \mid H) = O_2 \cdot Q$ .

Suppose:

$$\frac{\frac{H \xrightarrow{\tilde{r}.t.n} (\tilde{r})\langle k[H_1] \rangle H_2}{D[M] \mid H \xrightarrow{\tilde{r}.t.n} (\tilde{r})\langle k[H_1] \rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{\tilde{r}.t.n} O_1 \quad (\tilde{r})(n[ \mid H_2 \mid D[M] ] \mid k[H_1])}$$

for some process  $H_1$  and system  $H_2$ . Remark that  $k = \tilde{r}$ . We can construct the following derivation:

$$\frac{\frac{H \xrightarrow{\tilde{r}.t.n} (\tilde{r})\langle k[H_1] \rangle H_2}{D[N] \mid H \xrightarrow{\tilde{r}.t.n} (\tilde{r})\langle k[H_1] \rangle H_2 \mid D[N]}}{n[ \mid D[N] \mid H ] \longrightarrow (\tilde{r})(n[ \mid H_2 \mid D[N] ] \mid k[H_1]) = O_2}$$

We can conclude that for all processes  $P$ , it holds  $O_1 \cdot P \leq O_2 \cdot P$  up to structural congruence, because  $D[M] \leq D[N]$  and  $\leq$  is closed under parallel composition and ambient.

Then, we consider the cases when there is interaction between  $D[M]$  and  $H$ .

–  $D[M] \mid H \longrightarrow O_1$ , because

$$D[M] \xrightarrow{nt \ r.n} (\tilde{m})\langle k[M_1] \rangle M_2 \text{ and } H \xrightarrow{\tilde{s}.n} (\tilde{h})\langle H_1 \rangle H_2.$$

Then  $O_1 = (\tilde{h}, \tilde{m})(n[k[M_1] \mid H_1] \mid M_2 \mid H_2)$ . We distinguish the cases  $k = \tilde{m}$ , and  $k \neq \tilde{m}$ .

$k = \tilde{m}$ . As  $D[M] \xrightarrow{nt \ r.n} (\tilde{m})\langle k[M_1] \rangle M_2$ , it also holds  $D[M] \xrightarrow{k. \ nt \ r.n} M' = (\tilde{m})(n[ \mid k[M_1] ] \mid M_2)$ . The induction hypothesis constr



exists a system  $N'$  such that  $D[N] \mid n[ ] \Rightarrow N' \quad (\tilde{n})(n[ \mid N_1 \mid N_2])$ , and

k ř



**Theorem 3.4** *Late bisimilarity is contextual*

**Proof** Let  $S$  be the smallest binary relation between systems such that:

1.  $M \sim N$ ;
2. if  $M \sim N$ , then  $C[M] \sim C[N]$  for all system contexts  $C[-]$ .

Remark that  $S$  is symmetric because of the symmetry of  $\sim$ . We prove that  $S$  is a late bisimilarity up to  $\sim$  by induction on the definition of  $S$ .

- $M \sim N$  because  $M \sim N$ .

Immediate.

- $C[M] \sim C[N]$  because  $M \sim N$ .

The induction hypothesis assures that  $(M, N) \sim$  is a pair satisfying the bisimulation conditions in  $S$ . Lemma 3.3 assures that the pair  $(C[M], C[N])$  satisfies the bisimulation conditions in  $S$ .

□

Note that the above proof does not rely on the transitivity of the late bisimulation. Note also that it is easy to adapt Lemma 3.3 and the above proof to show that early bisimilarity is contextual.

**Proposition 3.5** *Late bisimilarity is an equivalence relation*

**Proof**

---

**Table 7** Contexts for visible actions

---

=  $k.\text{enter}_n$  C [-] =  $n[ \mid \text{or}[in\_k.out\_k.out\_n]] \mid -$

=  $k.\text{exit}_n$  C [-] =  $( a)a[in\_k.out\_k. \text{or}[out\_a]] \mid n[ \mid -]$

=  $n.\text{enter}_k$  C [-] =  $( a)a[in\_n.k[out\_a.( \mid ( b)b[out\_k.out\_n. \text{or}[out\_b]])]] \mid -$

=  $k.\text{open}_n$  C [-] =  $k[ \mid ( a, b)(\text{open}_b.\text{open}_a. \text{or}[out\_k] \mid a[- \mid \text{open}_n.b[out\_a]])]$

where a and b are

This implies  $C_{k, \text{nt}, r, n}[M] \cdot P \Rightarrow M' \cdot P \mid \text{on } []$ .

= k.exit\_n Let P be a process. We know that  $M \xrightarrow{k, \text{nt}, r, n} M'$

---

**Table 8** Auxiliary contexts and processes

---

$$^{-1} \quad ^{-2} \quad = \quad ( \quad o)(o[\quad b$$

**Proof**

there exist systems  $M_1$  and  $M_2$  and a static context  $C[-]$  such that:

$$C[M] \bullet \text{SPY } i, j, P \\ = \text{ n[SPY } i, j, P \mid \text{ on[in\_k.out\_k.out\_n]] } \mid M$$

As the name  $on$  is fresh for  $M$ , by several applications of Lemma 3.11 to the reduction marked by  $( )$  we have:

$$\begin{aligned} & ( a)a[in\_k.out\_k.0] | M_1 \bullet SPY \ i,j,P \\ \Rightarrow & ( a)E[0 | a[]] \bullet SPY \ i,j,P . \end{aligned}$$

Again, as  $a$  is fresh, by several applications of Lemma 3.11, and reducing underneath  $( a)$ , we obtain:

$$\begin{aligned} & ( a)(0 | M_1) \bullet SPY \ i,j,P \\ \Rightarrow & ( a)E[0 | 0] \bullet SPY \ i,j,P . \end{aligned}$$

Summarising,

$$M_1 \bullet SPY \ i,j,P \quad ( a)(0 | M_1) \bullet SPY \ i,j,P \Rightarrow ( a)E[0 | 0] \bullet SPY \ i,j,P$$

and, as  $\bullet$  is closed under reductions,

$$M_1 \Rightarrow E[0].$$

So, assuming  $M' = E[0]$ , we can conclude.

$= n.\overline{enter\_k}$ . Observe that

$$\begin{aligned} C [M] \bullet SPY \ i,j,P &= \\ & ( a)a[in\_n.k[out\_a.(SPY \ i,j,P | ( b)b[out\_k.out\_n. on [out\_b]])]] | M \end{aligned}$$

To unleash the ambient  $on$ , the ambient  $a$  must use its  $in\_n$  capability, and the  
 in the 2649 mathlib file  
 on 4.236.896(t)-0.147034(h)-0.31047032 reduction secrea

Observe that,

$$\begin{aligned} & D[(a) a[k_{out\_a} (SPY \ i, j, P \ | \ (b) b[k_{out\_k} \cdot out\_n \cdot on \ [out\_b]])]] \\ & = D[k[SPY \ i, j, P \ | \ (b) b[k_{out\_k} \cdot out\_n \cdot on \ [out\_b]]]] \end{aligned}$$

Thus, by examining the above reductions sequence from  $C_{n, \overline{nt} \ r, k}$



ambient  $a$ . More precisely, there exist a system  $M_1$ , processes  $Q_i$ , and a static context  $D[-]$  such that:

$$\begin{aligned}
& C_{k.op\ n.n}[M] \bullet SPY\ i, j, P \\
= & k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid a[M \mid open\_n.b[out\_a]])] \\
\Rightarrow & k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid a[M_1 \mid open\_n.b[out\_a]])] \\
\longrightarrow & k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid a[Q \mid b[out\_a]])] \\
\Rightarrow & k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid a[Q_1 \mid b[out\_a]])] \\
\longrightarrow & k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid b[] \mid a[Q_1])] \\
\Rightarrow & k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid b[] \mid a[Q_2])] \\
\longrightarrow & k[SPY\ i, j, P \mid (a, b)(open\_a.\ on\ [out\_k] \mid \mathbf{0} \mid a[Q_2])] \\
\Rightarrow & k[SPY\ i, j, P \mid (a, b)(open\_a.\ on\ [out\_k] \mid \mathbf{0} \mid a[Q_3])] \\
\Rightarrow & k[SPY\ i, j, P \mid (a, b)(\ on\ [out\_k] \mid \mathbf{0} \mid Q_3)] \\
\Rightarrow & D[\ on\ []] \bullet SPY\ i, j, P \\
& D[\mathbf{0}] \bullet SPY\ i, j, P \mid \ on\ [] \\
= & \mathbf{0} \mid \ on\ []
\end{aligned}$$

Examining the above reductions sequence from  $C_{k.op\ n.n}[M] \bullet SPY\ i, j, P$  we conclude that

$$M \Rightarrow \xrightarrow{k.op\ n.n} k[ \mid Q].$$

As

$$\begin{aligned}
& k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\ on\ [out\_k] \mid a[Q \mid b[out\_a]])] \\
\Rightarrow & D[\ on\ []] \bullet SPY\ i, j, P
\end{aligned}$$

and the name  $on$  is fresh, by several applications of Lemma 3.11 we have

$$\begin{aligned}
& k[SPY\ i, j, P \mid (a, b)(open\_b.open\_a.\mathbf{0} \mid a[Q \mid b[out\_a]])] \\
\Rightarrow & D[\mathbf{0}] \bullet SPY\ i, j, P .
\end{aligned}$$

By Lemma 3.10, this implies

$$k[ \mid (a, b)(open\_b.open\_a.\mathbf{0} \mid a[Q \mid b[out\_a]])] \Rightarrow D[\mathbf{0}].$$

Applying our proof techniques we can easily prove that:

$$k[ \mid (a, b)(open\_$$

**Theorem 3.13 (Completeness)**

To conclude we must establish that for all  $P$ , it holds  $M' \cdot P = N' \cdot P$ . As barbed congruence is preserved by restriction, we have (

where  $M' \bullet \text{SPY}_{*}^{\text{nt r-n}} i, j, P_{i,j}$ . Call this outcome  $O_1$ .

This reduction must be matched by a corresponding reduction

$$C P [N] \Rightarrow O_2$$

where  $O_1 = O_2$  and  $O_2 \text{ A,B}$ . By several applications of Lemma 3.10 it follows that there is a system  $N'$  such that  $O_2 = N' \bullet \text{SPY}_{*}^{\text{nt r-n}} i, j, P$



- $f M \xrightarrow{\text{enter}_n, \text{exit}_n} M''$  then there exists a system  $N''$  such that  $N \hat{\Rightarrow} N''$  and for a process  $P$  there is a system context  $C[-]$  and systems  $M'$  and  $N'$  such that  $M'' \cdot P \approx C[M']$ ,  $N'' \cdot P \approx C[N']$  and  $M' R N'$
- $f M \xrightarrow{\text{nt}_r} M''$  then there exists a system  $N''$  such that  $N | n[] \Rightarrow N''$  and for a process  $P$  there is a system context  $C[-]$  and systems  $M'$  and  $N'$

- $M \leq N$  because  $M \approx A, A \leq B, B \approx N$ .

The induction hypothesis tells us that  $A \leq B$  behaves like a late bisimulation.

Suppose  $M \xrightarrow{\text{.enter}_n, \text{.exit}_n} M'$ . A simple diagram chasing allows us to conclude that there are systems  $A', B', N'$  such that for all process  $P$  it holds  $M' \cdot P \approx A' \cdot P \leq B' \cdot P \approx N' \cdot P$ , and in turn, by construction of  $S$ ,  $M' \cdot P \leq N' \cdot P$ .

Suppose  $M \xrightarrow{\text{.nt r}_n} M'$ . As  $M \approx A$ , for all process  $P$ , it holds  $M' \cdot P \approx A | n[P]$ . As  $A \leq B$ , the closure properties of  $S$  assure that  $A | n[P] \leq B | n[P]$ . The expansion relation is a congruence, and since  $B \leq N$  we have  $B | n[P] \leq N | n[P]$ .

**Table 9** The Message-passing Mobile Ambients in Two Levels

<i>Nam es</i>	$a, b, \dots, k, l, m, n, \dots$	$N$
<i>Capab ilit es</i>	$\mathbf{C} ::= \begin{array}{l} \text{in}_n \\ \text{out}_n \\ \text{open}_n \end{array}$	may enter into $n$ may exit out of $n$ may open $n$
<i>Express ions</i>	$\mathbf{E, F} ::= \begin{array}{l} x \\ \mathbf{C} \\ \mathbf{E.F} \end{array}$	variable capability path empty path
<i>Guards</i>	$\mathbf{G} ::= \begin{array}{l} \mathbf{E} \\ (x) \\ \mathbf{E} \end{array}$	expression input output
<i>Systems</i>	$\mathbf{M, N} ::= \begin{array}{l} \mathbf{0} \\ M_1 \mid M_2 \\ (n)M \\ n[P] \end{array}$	termination parallel composition restriction ambient
<i>Processes</i>	$\mathbf{P, Q, R} ::= \begin{array}{l} \mathbf{0} \\ P_1 \mid P_2 \\ (n)P \\ G.P \\ n[P] \\ !G.P \end{array}$	nil process parallel composition restriction prefixing ambient replication
<i>Structural and reduction rules for Communication</i>	$\begin{array}{l} \mathbf{E.(F.P)} \quad \mathbf{(E.F).P} \\ \mathbf{.P} \rightarrow \mathbf{P} \\ \mathbf{(x).P} \mid \mathbf{M} \quad \mathbf{.Q} \rightarrow \mathbf{P\{M/x\}} \mid \mathbf{Q} \end{array}$	(Struct Path) (Red Empty Path) (Red Comm)

**Table 10** Pre-actions and Concretions for Communication

<i>Pre actions</i>	$::= \dots$ $\mid (\mathbf{E}) \mid -$	<i>Concretions</i>	$\mathbf{K} ::= (\tilde{m})\langle \mathbf{P} \rangle \mathbf{Q}$ $\mid (\tilde{m})\langle \mathbf{E} \rangle \mathbf{Q}$
--------------------	--	--------------------	---



---

**Table 11** Labeled Transition System - Communication
 

---

$$\begin{array}{ll}
 \text{( Output)} \frac{}{E.P \xrightarrow{\langle E \rangle} P} & \text{( Input)} \frac{}{(x).P \xrightarrow{(E)} P\{E/x\}} \\
 \text{( Path)} \frac{E.(F.P) \longrightarrow Q}{(E.F).P \longrightarrow Q} & \text{( Eps)} \frac{}{.P \longrightarrow P} \\
 \text{( Comm)} \frac{P \xrightarrow{\langle \tilde{m} \rangle} (E)P' \quad Q \xrightarrow{(E)} Q' \quad \text{fn}(Q') \cap \{\tilde{m}\} = \emptyset}{P \mid Q \longrightarrow ( \tilde{m} )(P' \mid Q')} &
 \end{array}$$


---

instantiated by a system context, because in a system context the hole cannot appear under a prefix. This in turn implies that our bisimulations can be applied to the extended calculus, and all the results of Section 3 and Section 4 hold without modifications.

**Theorem 5.1** *Late bisimilarity, early bisimilarity and barbed congruence coincide in the Message Passing Calculus*

**Theorem 5.2** *The up to expansion, up to context and up to context and expansion proof techniques are sound proof techniques in the Message Passing Calculus*

## 6 Algebraic Theory

In this section we prove a collection of algebraic properties

$$P(\mathbf{m} | \prod_{j \in J} \text{open\_}k_j \cdot R_j) = P(\mathbf{m} | P) \prod_{j \in J} P(\text{open\_}k_j | R_j) \quad \text{if } m = k_j \text{ for } j \in J$$

$$P(\mathbf{m} | \text{open\_}m \cdot P | \mathbf{m} | N) | Q = P(\mathbf{m} | P | N) | Q \quad \text{if } Q = M | \prod_{j \in J} W_j \cdot R_j$$

and

**Lemma 6.2** *Let  $P$ ,  $Q$  and  $R$  be processes then*

$$\begin{aligned} & (k, m, w)(k[in\_m.P] \mid m[open\_k.Q] \mid w[open\_m.R]) \\ & = (k, m, w)(m[k[P] \mid open\_k.Q] \mid w[open\_m.R]) \end{aligned}$$

$$\_ (m, w)(m[in\_w \mid (x).P] \mid w[open\_m.Q]) = (m, w)(m[P \{^{n-w}/x\}] \mid w[open\_m.Q])$$

**Proof**

- Our env-actions, unlike those in [18], are truly late, as they do not mention the process provided by the environment. This process can be added *ate*, when playing the bisimulation game.
- Our actions for ambient's movement, unlike those in SAP, report the name of the migrating ambient. For instance, in `k.enter_n` we say that ambient `k` enters `n`. The knowledge of `k` is necessary to make the action observable for the environment. This

- [7] L. Cardelli and A. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000. An extended abstract appeared in

- [23] D.M. Park. Concurrency on automata and infinite sequences. In P. Deussen, editor, *Conf on Theoretical Computer Science*, volume 104 of *LNC*. Springer Verlag, 1981.
- [24] D. Sangiorgi. *Expressing Mobility in Process Algebras: First Order and Higher Order Paradigms*. PhD thesis CST-99-93, Department of Computer Science, University of Edinburgh, 1992.
- [25] D. Sangiorgi. Bisimulation for Higher-Order Process Calculi. *Information and Computation*, 131(2):141-178, 1996.
- [26] D. Sangiorgi. Locality and non-interleaving semantics in calculi for mobile processes. *Theoretical Computer Science*, 155:39-83, 1996.
- [27] D. Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8:447-479, 1998.
- [28] D. Sangiorgi. Extensionality and intensionality of the ambient logic. In *Proceedings of the 28th ACM Symposium on Principles of Programming Languages (POPL)*. ACM Press, 2001.
- [29] D. Sangiorgi and R. Milner. The problem of "Weak Bisimulation up to". In *Proceedings of the 13th ACM Symposium on Principles of Programming Languages (POPL)*, volume 630 of *LNC*, pages 32-46. Springer Verlag, 1992.