Bisimulation Proof Methods for Mobile Ambients

Massimo Merro
Università di Verona, Italy

Francesco Zappa Nardelli
LIENS, Paris, France

University of Sussex TR 01:2003; Revised February 2003

Abstract

We study the behavioural theory of Cardelli and Gordon's Mobile Ambients. We give an *lts* based operational semantics, and a labelled *bisimulation* based equivalence that coincides with reduction barbed congruence. We also provide two *up-to proof techniques* that we use to prove a set of *algebraic laws*, including the perfect firewall equation.

Introduction

The calculus of *Mobile Ambients* [7], abbreviated MA, has been introduced as a process calculus for describing *mobile agents*.

In MA, the term $n[P]$ represents an agent, or *ambient*, named n, executing the code P. Intuitively, the ambient n is a bounded, protected, and (potentially) mobile space where the computation P takes place. In turn P may contain other ambients, may perform (local) *communications*, or may exercise *capabilities*, that allow entry to or exit from named ambients. *Ambient names*, such as n, are used to control access to the ambient's computation space and may be dynamically created as in the π-calculus, [21], using the construct $(\boldsymbol{\nu} n)P$. A *system* in MA consists of a collection of ambients running in parallel where the knowledge of certain ambient names may be restricted.

A crucial notion in a process calculus, and therefore also in MA, is that of *behavioural equality* between processes. For instance, behavioural equalities are used to verify that an implementation respects its specification, or that a code optimisation is correct. Roughly speaking, two processes are behavioural equivalent if no difference can be detected by interacting with them. In this paper, we focus on *bisimulation-based* behavioural equivalences. The notion of bisimulation was originally proposed in [23] and since then it has been used to define semantic equivalences for a variety of process calculi such as CCS, [20], and the π-calculus, [21].

Our *touchstone equivalence* is a generalisation of the *reduction barbed congruence* of [16]. Reduction barbed congruence is the largest equivalence relation which

- is a congruence for the language, that is is preserved by the constructs of the language
- preserves, in some sense, the reduction semantics of the language

- 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 *labelled bisimilarities* 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 *labelled transition system*, or *LTS*, a collection of relations of the form

$$P \xrightarrow{\boldsymbol{\alpha}} Q.$$

Intuitively the action α in the judgement $P \xrightarrow{\boldsymbol{\alpha}} Q$ 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 *stuttering* 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 firewall equation*, [7]:

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

  If you suppose $P = \texttt{in\_}k.\mathbf{0}$, it is evident that a bisimilarity that want to capture this law must not observe the movements of *secret ambients*, 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 mobility*, as in Levi and Sangiorgi's *Safe Ambients* [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 provide bisimulation proof methods for Mobile Ambients.*

Contribution First of all, as in the Distributed π-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 τ-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* style. However, as in HOπ [25], the two formulations coincide, and we concentrate on the easier late version, denoted by $\approx$. The definition of $\approx$ reminds us that of the asynchronous bisimilarity found in [1]. Indeed, as for inputs in asynchronous π, our bisimilarity does not observe the movements of secret ambients.

We prove that in MA the relation $\approx$ completely characterises reduction barbed congruence over systems. Then, we enhance our proof methods by defining two *up-to proof techniques*, along the lines of [22, 27, 30]. More precisely, we develop both *up-to-expansion* and *up-to-context* proof techniques and prove their soundness. We are not aware of other forms of up-to proof techniques for higher-order calculi. Finally, we apply our bisimulation proof methods to prove a collection of both old and new *algebraic laws* (among which the perfect firewall equation); then we also prove the correctness of the protocol, introduced in [7], for controlling access through a firewall.

The paper ends with Section 7, containing a discussion of our results and a comparison with related work.

1 Mobile Ambients in Two Levels

In Table 1 we give the syntax of MA, where $\mathbf{N}$ denotes an infinite set of names. Unlike other definitions of MA in the literature, our syntax is defined in a two-level structure, a lower one for *processes*, and an upper one for *systems*.

As regards processes, the constructs for inactivity, parallel composition, restriction and replicated prefixing are inherited from mainstream concurrent calculi, most notably the π-calculus [21]. The inactive process, $\mathbf{0}$, does nothing. Parallel composition is denoted by a binary operator, $P \mid Q$, that is commutative and associative. The restriction operator, $(\boldsymbol{\nu} n)P$, creates a new (unique) name n within a scope P. We have replicated prefixing, $!C.P$, (rather than full replication) to create as many parallel replicas as needed. Specific of the ambient calculus are the *ambient*, $n[P]$, and the *prefix* via capabilities, $C.P$. In $n[P]$, n is the name of the ambient and P is the process running inside the ambient. The process $C.P$ executes an action regulated by the capability C, and then continues as the process P. Capabilities are obtained from names; given a name n, the capability $\texttt{in\_}n$ allows entry into n, the capability $\texttt{out\_}n$ allows exit out of n, and the capability $\texttt{open\_}n$ allows the destruction of the boundary of ambient n. For the sake of simplicity, at this stage, we omit *communication*; it will be added in Section 5.

Systems are just a collection of ambients running in parallel where the knowledge of certain ambient names may be restricted among two or more ambients.

We use a number of notational conventions. Parallel composition has the lowest precedence among the operators. The process $C.C'.P$ is read as $C.(C'.P)$. We omit trailing dead

Table 1 The Mobile Ambients in Two Levels

Names: $a, b, \ldots, k, l, m, n, \ldots \in \mathbf{N}$

Systems:

M, N	$::= \mathbf{0}$	termination
	$\mid M_1 \mid M_2$	parallel composition
	$\mid (\boldsymbol{\nu} n)M$	restriction
	$\mid n[P]$	ambient

Capabilities:

C	$::= \texttt{in\_}n$	may enter into n
	$\mid \texttt{out\_}n$	may exit out of n
	$\mid \texttt{open\_}n$	may open n

Processes:

P, Q, R	$::= \mathbf{0}$	nil process
	$\mid P_1 \mid P_2$	parallel composition
	$\mid (\boldsymbol{\nu} n)P$	restriction
	$\mid C.P$	prefixing
	$\mid n[P]$	ambient
	$\mid !C.P$	replication

processes, writing C for $C.\mathbf{0}$, and $n[\ ]$ for $n[\mathbf{0}]$. Restriction $(\boldsymbol{\nu} n)P$ acts as binder for name n, and the set of *free names* of P, $\mathrm{fn}(P)$, is defined accordingly.

Operational semantics The dynamics of the calculus is given in the form of a *reduction relation* over processes as described in Table 2. However, as systems are processes with a special structure, the rules of Table 2 also describe the evolution of systems. The *reduction semantics* relies on an auxiliary relation called *structural congruence* which brings the participants of a potential interaction into contiguous positions. The definitions of structural congruence, $\equiv$, and of the reduction relation, $\rightarrowtriangle$, can be found in the in Table 2. It is easy to check that the reduction relation is closed under systems, that is, systems always reduce to systems.

Behavioural semantics One of the main motivation of our work is the definition of a notion of labelled bisimilarity for MA. Rather than simply defining an ad-hoc bisimulation based equivalence over systems we first introduce a basic equivalence by considering natural desirable properties.We choose to start from a generalisation of the reduction barbed congruence of [16].

Definition 1.1 *A relation $\mathcal{R}$ over systems is* reduction closed *if $M\ \mathcal{R}\ N$ and $M \rightarrowtriangle M'$ implies the existence of some N' such that $N \rightarrowtriangle^* N'$ and $M'\ \mathcal{R}\ N'$, where $\rightarrowtriangle^*$ denotes the reflexive and transitive closure of $\rightarrowtriangle$.*

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)
$(\boldsymbol{\nu} n)\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Res)
$!C.P \equiv C.P \mid !C.P$	(Struct Repl Par)
$(\boldsymbol{\nu} n)(\boldsymbol{\nu} m)P \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu} n)P$	(Struct Res Res)
$n \notin \mathrm{fn}(P)$ implies $(\boldsymbol{\nu} n)(P \mid Q) \equiv P \mid (\boldsymbol{\nu} n)Q$	(Struct Res Par)
$n \neq m$ implies $(\boldsymbol{\nu} n)(m[P]) \equiv m[(\boldsymbol{\nu} n)P]$	(Struct Res Amb)

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

$n[\mathtt{in\_}m.P \mid Q] \mid m[R] \rightarrowtriangle m[\, n[P \mid Q] \mid R\,]$	(Red In)
$m[\, n[\mathtt{out\_}m.P \mid Q] \mid R\,] \rightarrowtriangle n[P \mid Q] \mid m[R]$	(Red Out)
$\mathtt{open\_}n.P \mid n[\, Q\,] \rightarrowtriangle P \mid Q$	(Red Open)
$P \equiv Q \quad Q \rightarrowtriangle R \quad R \equiv S$ implies $P \rightarrowtriangle S$	(Red Struct)

$\rightarrowtriangle$ 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 grammar:*

$$C[-] \quad ::= \quad - \quad \mid \quad C[-] \mid M \quad \mid \quad (\boldsymbol{\nu} n)C[-] \quad \mid \quad n[C[-] \mid P]$$

where M is an arbitrary system, and P is an arbitrary process.

Definition 1.3 *A relation $\mathcal{R}$ over systems is* contextual *if $M\ \mathcal{R}\ N$ implies $C[M]\ \mathcal{R}\ C[N]$ for all system contexts $C[-]$.*

In Mobile Ambients the observation predicate $M \downarrow_n$ denotes the possibility of the system M of interacting with the environment via the ambient n. We write $M \downarrow_n$ if $M \equiv (\boldsymbol{\nu} \tilde{m})(n[P] \mid M')$ where $n \notin \{\tilde{m}\}$. We write $M \Downarrow_n$ if there exists M' such that $M \rightarrowtriangle^* M'$ and $M' \downarrow_n$.

Definition 1.4 *We say that a relation $\mathcal{R}$ over systems is* barb preserving *if $M\ \mathcal{R}\ N$ and $M \downarrow_n$ implies $N \Downarrow_n$.*

Definition 1.5 (Reduction barbed congruence) *Reduction barbed congruence, written $\cong$, is the largest symmetric relation over systems which is reduction closed, contextual, and barb preserving.*

Table 3 Pre-actions, Env-actions, Actions, Concretions, and Outcomes

Pre-actions:	π	$::=$		*Outcomes:*	O	$::=$	$P \mid K$
		$\mid$	$\texttt{in\_}n \mid \texttt{out\_}n$				
		$\mid$	$\texttt{open\_}n \mid \texttt{enter\_}n$				
		$\mid$	$\texttt{amb\_}n \mid \texttt{exit\_}n$				
Env-actions:	σ	$::=$		*Concretions:*	K	$::=$	$(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$
		$\mid$	$k.\texttt{enter\_}n \mid k.\texttt{exit\_}n$				
		$\mid$	$*.\texttt{enter\_}n \mid *.\texttt{exit\_}n$				
		$\mid$	$n.\overline{\texttt{enter}}\_k \mid k.\texttt{open\_}n$				
Actions:	α	$::=$	$\sigma \cup \tau$				

2 A Labelled Transition Semantics

The capabilities or prefixes C in our language give rise, in the standard manner, [20], to transitions of the form $P \xrightarrow{C} Q$; for example we have

$$\texttt{in\_}n.P_1 \mid P_2 \xrightarrow{\texttt{in\_}n} P_1 \mid P_2.$$

However, similarly to [18], each of the capability C induces different and more complicated actions. Our actions are defined over processes, although in the labelled bisimilarity we only consider actions going from systems to systems. We make a distinction between *pre-actions* and *env-actions*: the former denote the possibility to exercise certain capabilities whereas the latter model the interaction of a system with its environment. As usual, we also have τ-*actions* to model internal computations. Only env-actions and τ-actions model the evolution of a system at run-time.

The pre-actions, defined in Table 4, are of the form $P \xrightarrow{\boldsymbol{\pi}} O$ where the range of π and of O, the *outcomes*, are given in Table 3. An outcome may be a simple process Q, if for example π is a prefix of the language, or a *concretion*, of the form $(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$, when an ambient boundary is somehow involved. Here, intuitively, P represents the part of the system affected by the action, while Q is not affected, and $\tilde{m}$ is the set of private names shared by P and Q. We adopt the convention that if K is the concretion $(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$, then $(\boldsymbol{\nu}r)K$ is a shorthand for $(\boldsymbol{\nu}\tilde{m})\langle P\rangle(\boldsymbol{\nu}r)Q$, if $r \notin \mathrm{fn}(P)$, and the concretion $(\boldsymbol{\nu}r\tilde{m})\langle P\rangle Q$ otherwise. We have a similar convention for the rule (π Par): $K \mid R$ is defined to be the concretion $(\boldsymbol{\nu}\tilde{m})\langle P\rangle(Q \mid R)$, where $\tilde{m}$ are chosen, using α-conversion if necessary, so that $\mathrm{fn}(R) \cap \{\tilde{m}\} = \emptyset$.

The τ-actions, formally defined in Table 5, model the internal evolution of processes. Basically, there are three possible interactions: entering, exiting , and opening of ambients. Then, we also have the structural rules.

The env-actions, formally defined in Table 6, are of the form $M \xrightarrow{\boldsymbol{\sigma}} M'$, where the range of σ is given in Table 3. In practise, env-actions turn concretions in running systems

Table 4 Labelled Transition System - Pre-actions

$$(\pi \text{ Pfx}) \ \frac{-}{\pi.P \xrightarrow{\ \pi\ } P} \qquad (\pi \text{ Repl Pfx}) \ \frac{-}{!\pi.P \xrightarrow{\ \pi\ } P \mid !\pi.P}$$

$$(\pi \text{ Enter}) \ \frac{P \xrightarrow{\ \mathtt{in\_}n\ } P_1}{m[P] \xrightarrow{\ \mathtt{enter\_}n\ } \langle m[P_1]\rangle \mathbf{0}} \qquad (\pi \text{ Amb}) \ \frac{-}{n[P] \xrightarrow{\ \mathtt{amb\_}n\ } \langle P\rangle \mathbf{0}}$$

$$(\pi \text{ Exit}) \ \frac{P \xrightarrow{\ \mathtt{out\_}n\ } P_1}{m[P] \xrightarrow{\ \mathtt{exit\_}n\ } \langle m[P_1]\rangle \mathbf{0}} \qquad (\pi \text{ Res}) \ \frac{P \xrightarrow{\ \pi\ } O \qquad n \notin \mathrm{fn}(\pi)}{(\boldsymbol{\nu} n)P \xrightarrow{\ \pi\ } (\boldsymbol{\nu} n)O}$$

$$(\pi \text{ Par}) \ \frac{P \xrightarrow{\ \pi\ } O}{\begin{array}{l} P \mid Q \xrightarrow{\ \pi\ } O \mid Q \\ Q \mid P \xrightarrow{\ \pi\ } Q \mid O \end{array}}$$

by explicitly introducing the environment's ambient interacting with the process in question. The content of this ambient will be instantiated later, in the bisimilarity, with a process. For convenience, we extend the syntax of processes with the special process $\circ$ to pinpoint those ambients whose content will be instantiated later. The process $\circ$ does not reduce: it is simply a placeholder. Notice that, unlike pre-actions and τ-actions, env-actions do not have structural rules; this is because env-actions are supposed to be performed by complete systems that can directly interact with the environment.

We call *actions* the set of env-actions to which τ has been added. Actions always go from systems to systems and, in general, from processes to processes, even if the outcome may possibly involve the special process $\circ$. As our bisimilarity will be defined over systems, we will only consider actions (and not pre-actions) in its definition.

Proposition 2.1 *If T is a system (resp. a process), and $T \xrightarrow{\ \alpha\ } T'$ then T' is a system (resp. a process), possibly containing the special process $\circ$.*

Now, we explain the rules induced by the the prefix `in`, the *immigration* of ambients. A typical example of an ambient m migrating into an ambient n is as follows:

$$(\boldsymbol{\nu} m)(m[\,\mathtt{in\_}n.P_1 \mid P_2\,] \mid M) \mid n[Q] \ \rightarrowtriangle \ (\boldsymbol{\nu} m)(M \mid n[\,m[\,P_1 \mid P_2\,] \mid Q])$$

The driving force behind the migration is the activation of the prefix $\mathtt{in\_}n$, within the ambient m. It induces a capability in the ambient m to migrate into n, which we formalise as a new action $\mathtt{enter\_}n$. Thus an application of (π Enter) gives

$$m[\mathtt{in\_}n.P_1 \mid P_2] \xrightarrow{\ \mathtt{enter\_}n\ } \langle m[P_1 \mid P_2]\rangle \mathbf{0}$$

and more generally, using the structural rules (π Res) and (π Par),

$$(\boldsymbol{\nu} m)(m[\mathtt{in\_}n.P_1 \mid P_2] \mid M) \xrightarrow{\ \mathtt{enter\_}n\ } (\boldsymbol{\nu} m)\langle m[P_1 \mid P_2]\rangle M.$$

Table 5 Labelled Transition System - τ-actions

$$(\tau \text{ Enter}) \quad \frac{P \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle P_2 \quad Q \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle Q_2 \; {}^{(*)}}{\begin{array}{c} P \mid Q \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(n[P_1 \mid Q_1] \mid P_2 \mid Q_2) \\ Q \mid P \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}\tilde{q})(\boldsymbol{\nu}\tilde{p})(n[Q_1 \mid P_1] \mid Q_2 \mid P_2) \end{array}}$$

$$(\tau \text{ Exit}) \quad \frac{P \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2}{n[P] \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}\tilde{m})(k[P_1] \mid n[P_2])} \qquad (\tau \text{ Amb}) \quad \frac{P \xrightarrow{\boldsymbol{\tau}} Q}{n[P] \xrightarrow{\boldsymbol{\tau}} n[Q]}$$

$$(\tau \text{ Open}) \quad \frac{P \xrightarrow{\texttt{open\_}\boldsymbol{n}} P_1 \quad Q \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle Q_1\rangle Q_2}{\begin{array}{c} P \mid Q \xrightarrow{\boldsymbol{\tau}} P_1 \mid (\boldsymbol{\nu}\tilde{m})(Q_1 \mid Q_2) \\ Q \mid P \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}\tilde{m})(Q_1 \mid Q_2) \mid P_1 \end{array}} \qquad (\tau \text{ Res}) \quad \frac{P \xrightarrow{\boldsymbol{\tau}} P'}{(\boldsymbol{\nu}n)P \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}n)P'}$$

$$(\tau \text{ Par}) \quad \frac{P \xrightarrow{\boldsymbol{\tau}} P'}{\begin{array}{c} P \mid Q \xrightarrow{\boldsymbol{\tau}} P' \mid Q \\ Q \mid P \xrightarrow{\boldsymbol{\tau}} Q \mid P' \end{array}}$$

(*) In rule (τ Enter) we require $((\mathrm{fn}(P_1) \cup \mathrm{fn}(P_2)) \cap \{\tilde{q}\}) = ((\mathrm{fn}(Q_1) \cup \mathrm{fn}(Q_2)) \cap \{\tilde{p}\}) = \emptyset$

This means that the ambient $m[\texttt{in\_}n.P_1 \mid P_2]$ has the capability to enter an ambient n; if the capability is exercised, the ambient $m[P_1 \mid P_2]$ will enter n while M will be the residual at the point of execution. Of course the action can only be executed if there is an ambient n in parallel. The rule (π Amb) allows to check for the presence of ambients. So for example, we have

$$n[Q] \xrightarrow{\texttt{amb\_}n} \langle Q\rangle \mathbf{0}.$$

Here, the concretion $\langle Q\rangle\mathbf{0}$ says that Q is in n, while $\mathbf{0}$ is outside. Finally, the communication (τ Enter) allows these two complementary actions to occur simultaneously, effecting the migration of the ambient $m[P_1 \mid P_2]$ from its current computation space into the ambient n, giving rise to the original move above:

$$(\boldsymbol{\nu}m)(m[\,\texttt{in\_}n.P_1 \mid P_2\,] \mid M) \mid n[Q] \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}m)(M \mid n[\,m[\,P_1 \mid P_2\,] \mid Q]).$$

Note that this is a *higher-order* interaction, as the ambient $m[P_1 \mid P_2]$ is transferred between two computation spaces.

We have not said yet what env-actions are useful for. They model the interaction of mobile agents with their environment. So, for instance, using the rule (Enter Shh), we derive from

$$(\boldsymbol{\nu}m)(m[\texttt{in\_}n.P_1 \mid P_2] \mid M) \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}m)\langle m[P_1 \mid P_2]\rangle M.$$

the transition

$$(\boldsymbol{\nu}m)(m[\texttt{in\_}n.P_1 \mid P_2] \mid M) \xrightarrow{\texttt{*.enter\_}\boldsymbol{n}} (\boldsymbol{\nu}m)(n[\circ \mid m[P_1 \mid P_2]] \mid M).$$

This transition denotes a private (and therefore *unknown*) ambient entering an ambient n provided by the environment. The computation running at n can be added later by instantiating the placeholder $\circ$.

Table 6 Labelled Transition System - Env-actions

$$\text{(Enter)}\ \frac{P \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2{}^{(\dagger)}}{P \xrightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[P_1]] \mid P_2)}$$

$$\text{(Co-Enter)}\ \frac{P \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle P_1\rangle P_2{}^{(\dagger)}}{P \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter\_}\boldsymbol{k}}} (\boldsymbol{\nu}\tilde{m})(n[P_1 \mid k[\,\circ\,]] \mid P_2)}$$

$$\text{(Exit)}\ \frac{P \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2{}^{(\dagger)}}{P \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})(k[P_1] \mid n[\circ \mid P_2])}$$

$$\text{(Open)}\ \frac{P \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle P_1\rangle P_2}{P \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} k[\circ \mid (\boldsymbol{\nu}\tilde{m})(P_1 \mid P_2)]}$$

$$\text{(Enter Shh)}\ \frac{P \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2{}^{(\ddagger)}}{P \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[P_1]] \mid P_2)}$$

$$\text{(Exit Shh)}\ \frac{P \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[P_1]\rangle P_2{}^{(\ddagger)}}{P \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})(k[P_1] \mid n[\circ \mid P_2])}$$

(†) In rules (Enter), (Co-Enter), and (Exit) we require $k \notin \tilde{m}$

(‡) In rules (Enter Shh) and (Exit Shh) we require $k \neq n$ and $k \in \tilde{m}$

Had the ambient name m not been restricted, we would have used the rule (Enter) to derive

$$m[\texttt{in\_}n.P_1 \mid P_2] \mid M \xrightarrow{\boldsymbol{m}.\texttt{enter\_}\boldsymbol{n}} n[\circ \mid m[P_1 \mid P_2]] \mid M$$

to model a global ambient m which enters an ambient n provided by the environment.

The rules of *emigration* are along the same lines. A typical example of ambient m emigrating from ambient n is as follows:

$$n[\,m[\texttt{out\_}n.P_1 \mid P_2] \mid Q] \rightarrowtriangle n[Q] \mid m[P_1 \mid P_2].$$

The driving force behind the emigration is the activation of the prefix $\texttt{out\_}n$ within the ambient m. It induces a capability in the ambient m to emigrate from n, which we formalise as a new action $\texttt{exit\_}n$. Thus an application of the rule (π Exit), followed by (π Par), gives

$$m[\texttt{out\_}n.P_1 \mid P_2] \mid Q \xrightarrow{\texttt{exit\_}\boldsymbol{n}} \langle m[P_1 \mid P_2]\rangle Q.$$

Here when this capability is exercised the code Q will remain inside the ambient n while the ambient $m[P_1 \mid P_2]$ will move outside. However to actually effect the emigration of m we need a further context, namely the ambient n from which to emigrate. This leads to the rule (τ Exit); an application of which gives the original move above:

$$n[\, m[\mathtt{out}\_n.P_1 \mid P_2] \mid Q] \xrightarrow{\tau} n[Q] \mid m[P_1 \mid P_2].$$

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

1. *If* $P \xrightarrow{\tau} P'$ *then* $P \rightarrowtriangle P'$
2. *If* $P \rightarrowtriangle P'$ *then* $P \xrightarrow{\tau} \equiv P'$.

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 π such that $P \xrightarrow{\pi} O$. For instance,

- If $P \xrightarrow{\mathtt{enter}\_n} O$ then there exist $\tilde{p}, m, P_1, P_2, P_3$, with $n \notin \tilde{p}$, such that

$$P \equiv (\boldsymbol{\nu}\tilde{p})(m[\mathtt{in}\_n.P_1 \mid P_2] \mid P_3) \text{ and } O \equiv (\boldsymbol{\nu}\tilde{p})\langle m[P_1 \mid P_2]\rangle P_3.$$

- If $P \xrightarrow{\mathtt{exit}\_n} O$ then there exist $\tilde{p}, m, P_1, P_2, P_3$, with $n \notin \tilde{p}$, such that

$$P \equiv (\boldsymbol{\nu}\tilde{p})(m[\mathtt{out}\_n.P_1 \mid P_2] \mid P_3) \text{ and } O \equiv (\boldsymbol{\nu}\tilde{p})\langle m[P_1 \mid P_2]\rangle P_3.$$

Similar results are necessary for the remaining pre-actions. The proof of these results is standard. □

Corollary 2.3 *If* $M \equiv N$ *and* $M \xrightarrow{\tau} M'$, *then there is* N' *such that* $N \xrightarrow{\tau} N'$ *and* $M' \equiv N'$.

From the results above, it is easy to establish that if $M \cong N$ then

- $M \Downarrow n$ iff $N \Downarrow n$
- $M \Rightarrow M'$ implies there is N' such that $N \Rightarrow N'$ and $M' \cong N'$.

In the sequel we will use these properties without comment.

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 bisimilarities*, that abstract over τ-actions, we introduce the notion of weak action. The definition is standard: $\Longrightarrow$ denotes the reflexive and transitive closure of $\xrightarrow{\boldsymbol{\tau}}$; $\overset{\boldsymbol{\alpha}}{\Longrightarrow}$ denotes $\Longrightarrow \xrightarrow{\boldsymbol{\alpha}} \Longrightarrow$; $\overset{\hat{\boldsymbol{\alpha}}}{\Longrightarrow}$ denotes $\Longrightarrow$ if $\alpha = \tau$ and $\overset{\boldsymbol{\alpha}}{\Longrightarrow}$ otherwise.

In the previous section we said that actions (and more precisely env-actions) introduce a special process $\circ$ 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 $\circ$: with respect to structural congruence, $\circ$ behaves like the inactive process $\mathbf{0}$. Before defining the bisimilarity we explain how $\circ$ 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:*

$$\begin{array}{rcl@{\qquad}rcl}
\mathbf{0} \bullet P & \stackrel{\text{def}}{=} & \mathbf{0} & (T_1 \mid T_2) \bullet P & \stackrel{\text{def}}{=} & (T_1 \bullet P) \mid (T_2 \bullet P) \\
n[R] \bullet P & \stackrel{\text{def}}{=} & n[R \bullet P] & (\boldsymbol{\nu} n)T \bullet P & \stackrel{\text{def}}{=} & (\boldsymbol{\nu} n)(T \bullet P) \text{ if } n \notin \mathrm{fn}(P) \\
\circ \bullet P & \stackrel{\text{def}}{=} & P & C.R \bullet P & \stackrel{\text{def}}{=} & C.(R \bullet P) \\
!C.R \bullet P & \stackrel{\text{def}}{=} & !C.(R \bullet P). & & &
\end{array}$$

Now, everything is in place to define our bisimilarity.

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

- *if $M \xrightarrow{\boldsymbol{\alpha}} M'$, $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there is a system N' such that $N \overset{\hat{\boldsymbol{\alpha}}}{\Longrightarrow} N'$ and for all processes P it holds $M' \bullet P \mathcal{R} N' \bullet P$;*
- *if $M \xrightarrow{*.\texttt{enter}\_\boldsymbol{n}} M'$ then there is a system N' such that $N \mid n[\,\circ\,] \Longrightarrow N'$ and for all processes P it holds $M' \bullet P \mathcal{R} N' \bullet P$;*
- *if $M \xrightarrow{*.\texttt{exit}\_\boldsymbol{n}} M'$ then there is a system N' such that $n[\circ \mid N] \Longrightarrow N'$ and for all processes P it holds $M' \bullet P \mathcal{R} N' \bullet P$.*

M and N are late bisimilar*, written $M \approx N$, if $M \mathcal{R} N$ for some late bisimulation $\mathcal{R}$.*

The bisimilarity above has a universal quantification over the process P provided by the environment. This process instantiates the special process $\circ$ 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 $\approx_{\mathrm{e}}$ to denote this early variant. By definition, every late bisimulation is also a early one, while the converse, in general, does not hold. However, in our case, as in HOπ [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 $*.\texttt{enter}\_n$ and $*.\texttt{exit}\_n$ 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 π-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 $\mathcal{S}$ be a contextual symmetric relation between systems. Let $(M, N) \in \mathcal{S}$ be a pair satisfying the bisimulation conditions in $\mathcal{S}$, that is,*

- *if $M \xrightarrow{\alpha} M'$, $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there is a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$;*
- *if $M \xrightarrow{*.\texttt{enter}\_n} M'$ then there is a system N' such that $N \mid n[\,\circ\,] \Longrightarrow N'$ and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$;*
- *if $M \xrightarrow{*.\texttt{exit}\_n} M'$ then there is a system N' such that $n[\circ \mid N] \Longrightarrow N'$ and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$.*

Then, all the pairs $(C[M], C[N])$, for any system context $C[-]$, also satisfy the bisimulation conditions in $\mathcal{S}$.

Proof The relation $\mathcal{S}$ is contextual, and as such it is the smallest relation between systems such that:

- if $M \;\mathcal{S}\; N$, then $M \mid H \;\mathcal{S}\; N \mid H$ for all systems H;
- if $M \;\mathcal{S}\; N$, then $(\boldsymbol{\nu} m)M \;\mathcal{S}\; (\boldsymbol{\nu} m)N$ for all names m;
- if $M \;\mathcal{S}\; N$, then $m[M \mid P] \;\mathcal{S}\; m[N \mid P]$ for all names m and processes P.

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

- $C[-] = -$.

  This case holds because $M \;\mathcal{S}\; N$ satisfies the bisimulation conditions in SS.

- $C[-] = (\boldsymbol{\nu} m)D[-]$.

  We know that $D[M] \;\mathcal{S}\; D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$, and we want to prove that $(\boldsymbol{\nu} m)D[M] \;\mathcal{S}\; (\boldsymbol{\nu} m)D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$ as well. Suppose $(\boldsymbol{\nu} m)D[M] \xrightarrow{\alpha}$. We perform a case analysis on α.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{\tau} O_1$.

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

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\texttt{enter\_}n} O_1$.

Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\texttt{enter\_}n} O_1 \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2)}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{k.\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\texttt{enter\_}n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{k.\texttt{enter\_}n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid k[N_1]] \mid N_2)$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{k.\texttt{enter\_}n} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid k[N_1]] \mid N_2)$. We obtain $(\boldsymbol{\nu} m)D[N] \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{k.\texttt{enter\_}n} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\texttt{exit\_}n} O_1$.

Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{k.\texttt{exit\_}n} O_1 \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1])}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{k.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{k.\texttt{exit\_}n} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{k.\texttt{exit\_}n} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid k[N_1])$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{k.\texttt{exit\_}n} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid k[N_1])$. We obtain $(\boldsymbol{\nu} m)D[N] \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{k.\texttt{exit\_}n} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call

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

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} O_1$.
Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1 \rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle M_1 \rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} O_1 \;\equiv\; (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2)}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1 \rangle M_2$ then $D[M] \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Longrightarrow A \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} B \Longrightarrow N'$, and for all processes P it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$. As $A \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[k[\,\circ\,] \mid N_1] \mid N_2)$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle N_1 \rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle N_1 \rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[k[\,\circ\,] \mid N_1] \mid N_2)$. We obtain $(\boldsymbol{\nu} m)D[N] \Longrightarrow (\boldsymbol{\nu} m)A \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} C \Longrightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \; \mathcal{S} \; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} O_1$.
Observe that this must have been derived from

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1 \rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle M_1 \rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} O_1 \;\equiv\; k[\circ \mid (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]}$$

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $D[M] \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1 \rangle M_2$ then $D[M] \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)] = M'$. Also observe that $O_1 \equiv (\boldsymbol{\nu} m)k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)] = (\boldsymbol{\nu} m)M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Longrightarrow A \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} B \Longrightarrow N'$, and for all processes P it holds $M' \bullet P \; \mathcal{S} \; N' \bullet P$. As $A \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} B$, the system B must be of the form $k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle N_1 \rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\texttt{amb}\_\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle N_1 \rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} C \equiv k[\circ \mid (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)] \equiv (\boldsymbol{\nu} m)k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)] = (\boldsymbol{\nu} m)N'$. We obtain $(\boldsymbol{\nu} m)D[N] \Longrightarrow (\boldsymbol{\nu} m)A \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} C \Longrightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \; \mathcal{S} \; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} O_1$.

Observe that there are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} O_1 \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid m[M_1]] \mid M_2)}$$

where $m \notin \tilde{r}$, for some process M_1 and system M_2. Remark that this implies $n \notin r$. As $D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2$ then $D[M] \xrightarrow{\boldsymbol{m}.\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid m[M_1]] \mid M_2) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{\boldsymbol{m}.\texttt{enter\_}\boldsymbol{n}} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{\boldsymbol{m}.\texttt{enter\_}\boldsymbol{n}} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid m[N_1]] \mid N_2)$, for some process N_1 and system N_2, where $m \notin \tilde{s}$. It also holds $A \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \mid n[\,\circ\,] \xrightarrow{\boldsymbol{\tau}} C \equiv (\boldsymbol{\nu} m)B = (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid m[N_1])$. We obtain $(\boldsymbol{\nu} m)(D[N] \mid n[\,\circ\,]) \equiv (\boldsymbol{\nu} m)D[N] \mid n[\,\circ\,] \Rightarrow (\boldsymbol{\nu} m)A \mid n[\,\circ\,] \xrightarrow{\boldsymbol{\tau}} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} O_1 \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2)}$$

where $k \in \tilde{r}$, for some process M_1 and system M_2. Remark that $n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2) = M'$. The induction hypothesis then tells us that there exist a system N' such that $D[N] \mid n[\,\circ\,] \Rightarrow N'$, and for all processes P it holds $M' \bullet P \mathcal{S} N' \bullet P$. We can derive $(\boldsymbol{\nu} m)D[N] \mid n[\,\circ\,] \equiv (\boldsymbol{\nu} m)(D[N] \mid n[\,\circ\,]) \Rightarrow (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

– $(\boldsymbol{\nu} m)D[M] \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} O_1$.

Observe that there are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}{(\boldsymbol{\nu} m)D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2}}{(\boldsymbol{\nu} m)D[M] \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} O_1 \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid m[M_1])}$$

where $m \notin \tilde{r}$, for some process M_1 and system M_2. Remark that this implies $n \notin r$. As $D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2$ then $D[M] \xrightarrow{\boldsymbol{m}.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid m[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{\boldsymbol{m}.\texttt{exit\_}\boldsymbol{n}} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. As $A \xrightarrow{\boldsymbol{m}.\texttt{exit\_}\boldsymbol{n}} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid m[N_1])$, for some process N_1 and system N_2, where $m \notin \tilde{s}$. It also holds $A \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu}m)A \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu}m)n[\circ \mid A] \xrightarrow{\boldsymbol{\tau}} C \equiv (\boldsymbol{\nu}m)B = (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid m[N_1])$. We obtain $(\boldsymbol{\nu}m)(D[N] \mid n[\,\circ\,]) \equiv (\boldsymbol{\nu}m)D[N] \mid n[\,\circ\,] \Rightarrow (\boldsymbol{\nu}m)A \mid n[\,\circ\,] \xrightarrow{\boldsymbol{\tau}} C \Rightarrow\equiv (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{(\boldsymbol{\nu}m)D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}}{(\boldsymbol{\nu}m)D[M] \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} O_1 \;\equiv\; (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1])}$$

where $k \in \tilde{r}$, for some process M_1 and system M_2. Remark that $n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist a system N' such that $n[\circ \mid D[N]] \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. We can derive $(\boldsymbol{\nu}m)D[N] \mid n[\circ] \equiv (\boldsymbol{\nu}m)(D[N] \mid n[\circ]) \Rightarrow (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

- $C[-] = D[-] \mid H$.

  We know that $D[M] \;\mathcal{S}\; D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$, and we want to prove that $D[M] \mid H \;\mathcal{S}\; D[N] \mid H$ satisfies the bisimulation conditions in $\mathcal{S}$ as well. We perform a case analysis on the transition $D[M] \mid H \xrightarrow{\boldsymbol{\alpha}} O_1$.

  We consider first the cases when there is no interaction between $D[M]$ and H.

  – $D[M] \mid H \xrightarrow{\boldsymbol{\tau}} O_1$, because $D[M] \xrightarrow{\boldsymbol{\tau}} M'$ and $O_1 \equiv M' \mid H$. The induction hypothesis tells us that there exists a N' such that $D[N] \Rightarrow N'$ and $M' \;\mathcal{S}\; N'$. Thus, $D[N] \mid H \Rightarrow O_2 \equiv N' \mid H$ and $O_1 \equiv M' \mid H \;\mathcal{S}\; N' \mid H \equiv O_2$ because $\mathcal{S}$ is closed under parallel composition.

  – $D[M] \mid H \xrightarrow{\boldsymbol{\tau}} O_1$, because $H \xrightarrow{\boldsymbol{\tau}} H'$ and $O_1 \equiv D[M] \mid H'$. Let $O_2 = D[N] \mid H'$: it holds $D[N] \mid H \xrightarrow{\boldsymbol{\tau}} O_2$, and $O_1 \;\mathcal{S}\; O_2$ because $D[M] \;\mathcal{S}\; D[N]$ and $\mathcal{S}$ is closed under parallel composition.

– $D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} O_1$.

There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} O_1 \equiv (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2 \mid H)}$$

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

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} O_1 \equiv (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[H_1]] \mid H_2 \mid M)}$$

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

$$\dfrac{\dfrac{H \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[H_1]] \mid H_2 \mid D[N]) = O_2}$$

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

– $D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} O_1$.

There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2 \mid H] \mid k[M_1])}$$

for some process M_1 and system M_2. Remark that $k \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Rightarrow A \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} B \Rightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. Remark that $N' \equiv (\boldsymbol{\nu}\tilde{h})n[\circ \mid N_3] \mid N_4$, for some N_3, N_4. As $A \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} B$, the system B must be of the form $(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid k[N_1])$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $A \mid H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2 \mid H] \mid k[N_1]) \equiv B \bullet (\circ \mid H)$. We obtain $D[N] \mid H \Rightarrow A \mid H \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} B \bullet (\circ \mid H) \Rightarrow\equiv N' \bullet (\circ \mid H)$. Call $N' \bullet (\circ \mid H) = O_2$. As for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$, we can conclude that for all processes Q, it holds $O_1 \bullet Q \;\mathcal{S}\; O_2 \bullet Q$ up to structural congruence, because $O_1 \bullet Q \equiv M' \bullet (Q \mid H) \;\mathcal{S}\; N' \bullet (Q \mid H) \equiv O_2 \bullet Q$.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[M]] \mid k[H_1])}$$

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

$$\dfrac{\dfrac{H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[N]] \mid k[H_1]) \;=\; O_2}$$

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

– $D[M] \mid H \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter\_}}\boldsymbol{k}} O_1$.

There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid M_1] \mid M_2 \mid H)}$$

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

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} O_1 \;\equiv\; (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid H_1] \mid H_2 \mid D[M])}$$

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

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle H_1\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} (\boldsymbol{\nu}\tilde{r})(n[k[\,\circ\,] \mid H_1] \mid H_2 \mid D[N]) \;=\; O_2}$$

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

– $D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{open}\_\boldsymbol{n}} O_1$.

There are two possible derivations.

∗ Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} O_1 \;\equiv\; k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid H]}$$

for some process M_1 and system M_2. Remark that $k, n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ then $D[M] \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]$. The induction hypothesis then tells us that there exist systems N', A, B such that $D[N] \Longrightarrow A \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} B \Longrightarrow N'$, and for all processes P it holds $M' \bullet P \;\mathcal{S}\; N' \bullet P$. As $A \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} B$, the system B must be of the form $k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$, for some process N_1 and system N_2. It also holds $A \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies $A \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2) \mid H] \;\equiv\; B \bullet (\circ \mid H)$. We obtain $D[N] \mid H \Longrightarrow A \mid H \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} \equiv B \bullet (\circ \mid H) \Longrightarrow \equiv N' \bullet (\circ \mid H)$. Call $N' \bullet (\circ \mid H) = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \;\mathcal{S}\; O_2 \bullet P$ up to structural congruence, because for all processes P it holds $M' \bullet (P \mid H) \;\mathcal{S}\; N' \bullet (P \mid H)$.

∗ Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} O_1 \;\equiv\; k[\circ \mid (\boldsymbol{\nu}\tilde{h})(H_1 \mid H_2) \mid D[M]]}$$

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

$$\dfrac{\dfrac{H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2 \mid D[N]}}{D[N] \mid H \xrightarrow{\boldsymbol{k}.\texttt{open\_}\boldsymbol{n}} k[\circ \mid (\boldsymbol{\nu}\tilde{h})(H_1 \mid H_2) \mid D[N]] \;=\; O_2}$$

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

– $D[M] \mid H \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} O_1$.

There are two possible derivations.

* Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{\texttt{*.enter\_}n} O_1 \equiv (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2 \mid H)}$$

where $k \in \tilde{r}$, for some process M_1 and system M_2. Remark that $n \notin \tilde{r}$. As $D[M] \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{\texttt{*.enter\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[M_1]] \mid M_2) = M'$. The induction hypothesis then tells us that there exist a system N' such that $D[N] \mid n[\,\circ\,] \Rightarrow N'$, and for all processes P it holds $M'\bullet P \mathrel{\mathcal{S}} N'\bullet P$. We can derive $D[N] \mid n[\circ] \mid H \Rightarrow N' \mid H$. Call $N' \mid H = O_2$. We can conclude that for all processes P, it holds $O_1 \bullet P \mathrel{\mathcal{S}} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under parallel composition.

* Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{\texttt{*.enter\_}n} O_1 \equiv (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[H_1]] \mid H_2 \mid D[M])}$$

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

$$\dfrac{\dfrac{H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]} \quad n[\,\circ\,] \xrightarrow{\texttt{amb\_}n} \langle\,\circ\,\rangle \mathbf{0}}{D[N] \mid H \mid n[\,\circ\,] \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid k[H_1]] \mid H_2 \mid D[N]) = O_2}$$

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

– $D[M] \mid H \xrightarrow{\texttt{*.exit\_}n} O_1$.

There are two possible derivations.

* Suppose:

$$\dfrac{\dfrac{D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2}{D[M] \mid H \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2 \mid H}}{D[M] \mid H \xrightarrow{\texttt{*.exit\_}n} O_1 \equiv (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2 \mid H] \mid k[M_1])}$$

for some process M_1 and system M_2. Remark that $k \in \tilde{r}$. As $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $D[M] \xrightarrow{\texttt{*.exit\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N' such that

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

* Suppose:

$$\dfrac{\dfrac{H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[M] \mid H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[M]}}{D[M] \mid H \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} O_1 \ \equiv \ (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[M]] \mid k[H_1])}$$

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

$$\dfrac{\dfrac{H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2}{D[N] \mid H \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[H_1]\rangle H_2 \mid D[N]}}{n[\circ \mid D[N] \mid H] \xrightarrow{\tau} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid H_2 \mid D[N]] \mid k[H_1]) = O_2}$$

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

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

– $D[M] \mid H \xrightarrow{\tau} O_1$, because

$$D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[M_1]\rangle M_2 \text{ and } H \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{h})\langle H_1\rangle H_2.$$

Then $O_1 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{m})(n[k[M_1] \mid H_1] \mid M_2 \mid H_2)$. We distinguish the cases $k \in \tilde{m}$, and $k \notin \tilde{m}$.

* $k \notin \tilde{m}$. As $D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[M_1]\rangle M_2$, it also holds $D[M] \xrightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} M' \equiv (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[M_1]] \mid M_2)$. The induction hypothesis tells us that there exists a system N' such that $D[N] \xRightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} N' \equiv (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[N_1]] \mid N_2)$, and for all processes P, it holds $M' \bullet P \ \mathcal{S} \ N' \bullet P$. But if $D[N] \xRightarrow{\boldsymbol{k}.\texttt{enter\_}\boldsymbol{n}} N'$, then $D[N] \xRightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[N_1]\rangle N_2$. This implies that $D[N] \mid H \xRightarrow{\tau} O_2 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{n})(n[k[N_1] \mid H_1] \mid N_2 \mid H_2)$. Since for all processes P, $M' \bullet P \ \mathcal{S} \ N' \bullet P$, it also holds $M' \bullet H_1 \ \mathcal{S} \ N' \bullet H_1$, and $O_1 \ \mathcal{S} \ O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.
* $k \in \tilde{m}$. As $D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle k[M_1]\rangle M_2$, it also holds $D[M] \xrightarrow{\texttt{enter\_}\boldsymbol{n}} M' \equiv (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[M_1]] \mid M_2)$. The induction hypothesis tells us that there

exists a system N' such that $D[N] \mid n[\circ] \Rightarrow N' \equiv (\boldsymbol{\nu}\tilde{n})(n[\circ \mid N_1] \mid N_2)$, and for all processes P, it holds $M' \bullet P \mathcal{S} N' \bullet P$. We can derive $D[N] \mid H \Rightarrow O_2 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{n})(n[N_1 \mid H_1] \mid N_2 \mid H_2)$. Since for all processes P, $M' \bullet P \mathcal{S} N' \bullet P$, it also holds $M' \bullet H_1 \mathcal{S} N' \bullet H_1$, and $O_1 \mathcal{S} O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.

– $D[M] \mid H \xrightarrow{\boldsymbol{\tau}} O_1$, because

$$D[M] \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle M_1 \rangle M_2 \text{ and } H \xrightarrow{\texttt{enter\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{h})\langle k[H_1] \rangle H_2.$$

Then $O_1 \equiv (\boldsymbol{\nu}\tilde{h}, \tilde{m})(n[k[H_1] \mid M_1] \mid M_2 \mid H_2)$. As $D[M] \xrightarrow{\texttt{amb\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{m})\langle M_1 \rangle M_2$, it also holds $D[M] \xrightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} M' \equiv (\boldsymbol{\nu}\tilde{m})(n[k[\circ] \mid M_1] \mid M_2)$. The induction hypothesis tells us that there exists a system N' such that $D[N] \xRightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} N' \equiv (\boldsymbol{\nu}\tilde{n})(n[k[\circ] \mid N_1] \mid N_2)$, and for all processes P, it holds $M' \bullet P \mathcal{S} N' \bullet P$. As $D[N] \xRightarrow{\boldsymbol{n}.\overline{\texttt{enter}}\_\boldsymbol{k}} N'$, we can derive $D[N] \xRightarrow{\texttt{amb\_}\boldsymbol{k}} (\boldsymbol{\nu}\tilde{n})\langle N_1 \rangle N_2$. It follows $D[N] \mid H \Rightarrow (\boldsymbol{\nu}\tilde{h}, \tilde{n})(n[k[H_1] \mid N_1] \mid N_2 \mid H_2) = O_2$. Since for all processes P, it holds $M' \bullet P \mathcal{S} N' \bullet P$, we have $M' \bullet h[H_1] \mathcal{S} N' \bullet h[H_1]$, and $O_1 \mathcal{S} O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.

- $C[-] = n[D[-] \mid P]$, where P is an arbitrary process.

We know that $D[M] \mathcal{S} D[N]$ satisfies the bisimulation conditions in $\mathcal{S}$, and we want to prove that $n[D[M] \mid P] \mathcal{S} n[D[N] \mid P]$ behaves as a bisimulation as well. We perform a case analysis on the transition $n[D[M] \mid P] \xrightarrow{\boldsymbol{\alpha}} O_1$.

– $n[D[M] \mid P] \xrightarrow{\boldsymbol{\tau}} O_1$, because $D[M] \xrightarrow{\boldsymbol{\tau}} M'$. Then $O_1 \equiv n[M' \mid P]$. The induction hypothesis tells us that there exists a system N' such that $D[N] \Rightarrow N'$ and $M' \mathcal{S} N'$. We can derive $n[D[N] \mid P] \Rightarrow n[N' \mid P]$ and conclude $n[M' \mid P] \mathcal{S} n[N' \mid P]$ because $\mathcal{S}$ is closed under ambient.

– $n[D[M] \mid P] \xrightarrow{\boldsymbol{\tau}} O_1$, because $P \xrightarrow{\boldsymbol{\tau}} P'$. Then $O_1 \equiv n[D[M] \mid P']$. Call $O_2 = n[D[N] \mid P']$. Then $O_1 \mathcal{S} O_2$ because $D[M] \mathcal{S} D[N]$, and $\mathcal{S}$ is closed under the contexts of the form $C[-] = n[- \mid Q]$ where Q is a process.

– $n[D[M] \mid P] \xrightarrow{\boldsymbol{\tau}} O_1$, because $D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1] \rangle M_2$. Then $O_1 \equiv (\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[M_2 \mid P])$. We distinguish the two cases $k \in \tilde{r}$ and $k \notin \tilde{r}$.

∗ $k \notin \tilde{r}$. From $D[M] \xrightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})\langle k[M_1] \rangle M_2$ we can derive $D[M] \xrightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[\circ \mid M_2])$. The induction hypothesis tells us that there exists a system N' such that $D[N] \xRightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} N' \equiv (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[\circ \mid N_2])$ and for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$. But $D[N] \xRightarrow{\boldsymbol{k}.\texttt{exit\_}\boldsymbol{n}} N'$ can only be derived from $D[N] \xRightarrow{\texttt{exit\_}\boldsymbol{n}} (\boldsymbol{\nu}\tilde{s})\langle k[N_1] \rangle N_2$ and thus $n[D[N] \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$, we can derive $(\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[P \mid M_2]) \mathcal{S} (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[P \mid N_2])$, as required.

* $k \in \tilde{r}$. From $D[M] \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ we can derive $D[M] \xrightarrow{*\texttt{.exit\_}n} (\boldsymbol{\nu}\tilde{r})(k[M_1] \mid n[\circ \mid M_2])$. The induction hypothesis tells us that there exists a system N' such that $n[\circ \mid D[N]] \Rightarrow N' \equiv (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[\circ \mid N_2])$, and for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$. We can instantiate the placeholder $\circ$ with the process P, thus obtaining the transition $n[D[N] \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$, we have $O_1 = (\boldsymbol{\nu}\tilde{m})(k[M_1] \mid n[P \mid M_2]) \equiv M' \bullet P \mathcal{S} N' \bullet P \equiv (\boldsymbol{\nu}\tilde{s})(k[N_1] \mid n[P \mid N_2]) = O_2$, as required.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, because $P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[P_1]\rangle P_2$. This implies $O_1 \equiv (\boldsymbol{\nu}\tilde{r})(k[P_1] \mid n[D[M] \mid P_2])$. It also holds $n[D[N] \mid P] \xrightarrow{\tau} \equiv (\boldsymbol{\nu}\tilde{r})(k[P_1] \mid n[D[N] \mid P_2])$. Call this last term O_2. The relation $O_1 \mathcal{S} O_2$ follows because $D[M] \mathcal{S} D[N]$ and from the closure properties of $\mathcal{S}$.

– $n[D[M] \mid P] \xrightarrow{\tau} O_1$, and the τ action is generated by an interaction between $D[M]$ and P. There are three cases.

* $D[M] \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ and $P \xrightarrow{\texttt{open\_}m} P'$. Then $O_1 = n[(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid P']$. It holds $D[M] \xrightarrow{n\texttt{.open\_}m} n[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]$. The induction hypothesis tells us that there exists a system N' such that $D[N] \xrightarrow{n\texttt{.open\_}m} N'$, and for all processes Q it holds $M' \bullet Q \mathcal{S} N' \bullet Q$. The system N' must be of the form $n[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$. The transition $D[N] \xRightarrow{n\texttt{.open\_}m} N'$ must have been derived from $D[N] \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies that $n[D[N] \mid P] \Rightarrow n[(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2) \mid P']$. Call this last term O_2. We can instantiate the placeholder $\circ$ with the process P', thus obtaining the transition $n[D[N] \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds $M' \bullet Q \mathcal{S} N' \bullet Q$, we have $O_1 = n[(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid P'] \equiv M' \bullet P' \mathcal{S} N' \bullet P' \equiv n[(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2) \mid P'] = O_2$, as required.

* $D[M] \xrightarrow{\texttt{enter\_}m}$ and $P \xrightarrow{\texttt{amb\_}m}$, or $D[M] \xrightarrow{\texttt{amb\_}m}$ and $P \xrightarrow{\texttt{enter\_}m}$. Call A_1 the outcome of the interaction between $D[M]$ and P. In both cases, by an analysis carried on previously, we know that there is a process A_2 such that $D[N] \mid P \Rightarrow A_2$, with $A_1 \mathcal{S} A_2$. We obtain $n[D[M] \mid P] \xrightarrow{\tau} n[A_1] = O_1$, and $n[D[N] \mid P] \Rightarrow n[A_2]$. The relation $n[A_1] \mathcal{S} n[A_2]$ follows from the closure of $\mathcal{S}$ under ambient.

– $n[D[M] \mid P] \xrightarrow{n\texttt{.}\overline{\texttt{enter}}\texttt{\_}k} O_1$. Then $O_1 \equiv n[k[\circ] \mid D[M] \mid P]$. But $n[D[N] \mid P] \xrightarrow{n\texttt{.}\overline{\texttt{enter}}\texttt{\_}k} O_2$, where $O_2 \equiv n[k[\circ] \mid D[N] \mid P]$. For all processes Q, $O_1 \bullet Q \mathcal{S} O_2 \bullet Q$ follows from $D[M] \mathcal{S} D[N]$ because of the closure properties of $\mathcal{S}$.

– $n[D[M] \mid P] \xrightarrow{n\texttt{.exit\_}m} m[\circ] \mid n[D[M] \mid P'] = O_1$, because $P \xrightarrow{\texttt{out\_}m} P'$. It also holds $n[D[N] \mid P] \xrightarrow{n\texttt{.exit\_}m} m[\circ] \mid n[D[N] \mid P']$. Call this last term O_2. Then, for all processes Q, the relation $O_1 \bullet Q \mathcal{S} O_2 \bullet Q$ follows from $D[M] \mathcal{S} D[N]$ because of the closure properties of $\mathcal{S}$.

□

Theorem 3.4 *Late bisimilarity is contextual.*

Proof Let $\mathcal{S}$ be the smallest binary relation between systems such that:

1. $\approx \subseteq \mathcal{S}$;
2. if $M \mathcal{S} N$, then $C[M] \mathcal{S} C[N]$ for all system contexts $C[-]$.

Remark that $\mathcal{S}$ is symmetric because of the symmetry of $\approx$. We prove that $\mathcal{S}$ is a late bisimilarity up to $\equiv$ by induction on the definition of $\mathcal{S}$.

- $M \mathcal{S} N$ because $M \approx N$.

  Immediate.

- $C[M] \mathcal{S} C[N]$ because $M \mathcal{S} N$.

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

□

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 [Sketch] The only non-trivial property is transitivity. We basically need Theorem 3.4 to say that $\approx$ is preserved by parallel composition and ambient nesting. These two properties are necessary to deal with the env-actions $*.\texttt{enter}\_n$ and $*.\texttt{exit}\_n$. □

Proposition 3.6 *Early bisimilarity is contextual, and it is an equivalence relation.*

As stated in the following Lemma, there is a close relationship between the observation predicate $M \downarrow_n$ and a particular action that M can emit.

Lemma 3.7

1. *If* $M \xrightarrow{n.\overline{\texttt{enter}}\_k} M'$ *then* $M \downarrow_n$;
2. *if* $M \downarrow_n$ *then there exists a system* M' *such that* $M \xrightarrow{n.\overline{\texttt{enter}}\_k} M'$, *for some* k.

We conclude that both late and early bisimilarity are contained in the reduction barbed congruence.

Theorem 3.8 (Soundness) *The following inclusions hold* $\approx \subseteq \approx_{\mathrm{e}} \subseteq \cong$.

Proof The first inclusion holds by definition. The second one comes from Proposition 3.6 and the fact that early bisimilarity is reduction closed and barb-preserving. □

Table 7 Contexts for visible actions

$\alpha = k.\texttt{enter\_}n$	$C_\alpha[-] = n[\circ \mid \mathsf{done}[\texttt{in\_}k.\texttt{out\_}k.\texttt{out\_}n]] \mid -$
$\alpha = k.\texttt{exit\_}n$	$C_\alpha[-] = (\boldsymbol{\nu} a)a[\texttt{in\_}k.\texttt{out\_}k.\mathsf{done}[\texttt{out\_}a]] \mid n[\circ \mid -]$
$\alpha = n.\overline{\texttt{enter\_}}k$	$C_\alpha[-] = (\boldsymbol{\nu} a)a[\texttt{in\_}n.k[\texttt{out\_}a.(\circ \mid (\boldsymbol{\nu} b)b[\texttt{out\_}k.\texttt{out\_}n.\mathsf{done}[\texttt{out\_}b]])]] \mid -$
$\alpha = k.\texttt{open\_}n$	$C_\alpha[-] = k[\circ \mid (\boldsymbol{\nu} a, b)(\texttt{open\_}b.\texttt{open\_}a.\mathsf{done}[\texttt{out\_}k] \mid a[- \mid \texttt{open\_}n.b[\texttt{out\_}a]])]$ where a and b are fresh.

3.2 Completeness

We now prove that late and early bisimilarity are more than proof techniques. They actually characterise reduction barbed congruence. The main challenge here is to design the contexts capable to observe our visible actions. The definition of these contexts, $C_\alpha[-]$, for every visible action α, is given in Table 7. The special ambient name done is used as *fresh* barb to signal the consumption of actions.

To prove our characterisation result it suffices to show that reduction barbed congruence is contained in the late bisimilarity. Then, by Theorem 3.8 we can conclude that late, early, and reduction barbed congruence, they all coincide. The proof that reduction barbed congruence implies the late bisimilarity requires the correspondence between visible actions α and their corresponding contexts $C_\alpha[-]$.

The following lemma says that the defining contexts are sound, that is, they can successfully mimic the execution of visible actions.

Lemma 3.9 *Let M be a system, and let $\alpha \in \{k.\texttt{enter\_}n, k.\texttt{exit\_}n, n.\overline{\texttt{enter\_}}k, k.\texttt{open\_}n\}$. For all processes P, if $M \xrightarrow{\alpha} M'$ then $C_\alpha[M] \bullet P \Rightarrow \cong M' \bullet P \mid \mathsf{done}[\,]$.*

Proof The proof is by case analysis on α.

$\alpha = k.\texttt{enter\_}n$ Let P be a process. We know that $M \xrightarrow{k.\texttt{enter\_}n} M'$. Then

$$M \equiv (\boldsymbol{\nu}\tilde{m})(k[\texttt{in\_}n.M_1 \mid M_2] \mid M_3)$$

where $(\{n, k\} \cup \mathrm{fn}(P)) \cap \{\tilde{m}\} = \emptyset$, and

$$M' \equiv (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[M_1 \mid M_2]] \mid M_3).$$

Now,

$$\begin{array}{rl}
 & C_{k.\texttt{enter_}n}[M] \bullet P \\
\equiv & (\boldsymbol{\nu}\tilde{m})(n[P \mid \mathsf{done}[\texttt{in_}k.\texttt{out_}k.\texttt{out_}n]] \mid k[\texttt{in_}n.M_1 \mid M_2] \mid M_3) \\
\rightarrow & (\boldsymbol{\nu}\tilde{m})(n[P \mid \mathsf{done}[\texttt{in_}k.\texttt{out_}k.\texttt{out_}n] \mid k[M_1 \mid M_2]] \mid M_3) \\
\rightarrow & (\boldsymbol{\nu}\tilde{m})(n[P \mid k[M_1 \mid M_2 \mid \mathsf{done}[\texttt{out_}k.\texttt{out_}n]]] \mid M_3) \\
\rightarrow & (\boldsymbol{\nu}\tilde{m})(n[P \mid \mathsf{done}[\texttt{out_}n] \mid k[M_1 \mid M_2]] \mid M_3) \\
\rightarrow & (\boldsymbol{\nu}\tilde{m})(\mathsf{done}[\,] \mid n[P \mid k[M_1 \mid M_2]] \mid M_3) \\
\equiv & (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[M_1 \mid M_2] \mid M_3]) \bullet P \mid \mathsf{done}[\,] \\
= & M' \bullet P \mid \mathsf{done}[\,]
\end{array}$$

This implies $C_{k.\texttt{enter\_}n}[M] \bullet P \Rightarrow\!\cong M' \bullet P \mid \mathsf{done}[\,]$.

$\alpha = k.\texttt{exit\_}n$ Let P be a process. We know that $M \xrightarrow{\boldsymbol{k.\texttt{exit\_}n}} M'$. Then

$$M \equiv (\boldsymbol{\nu}\tilde{m})(k[\texttt{out\_}n.M_1 \mid M_2] \mid M_3)$$

where $(\{n,k\} \cup \mathrm{fn}(P)) \cap \{\tilde{m}\} = \emptyset$, and

$$M' \equiv (\boldsymbol{\nu}\tilde{m})(k[M_1 \mid M_2] \mid n[\circ \mid M_3]).$$

Now,

$$\begin{array}{rl}
& C_{k.\texttt{exit_}n}M \bullet P \\
\equiv & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\texttt{in_}k.\texttt{out_}k.\mathsf{done}[\texttt{out_}a]] \mid n[P \mid k[\texttt{out_}n.M_1 \mid M_2] \mid M_3)] \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\texttt{in_}k.\texttt{out_}k.\mathsf{done}[\texttt{out_}a]] \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)k[a[\texttt{out_}k.\mathsf{done}[\texttt{out_}a]] \mid M_1 \mid M_2] \mid n[P \mid M_3]) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\mathsf{done}[\texttt{out_}a]] \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)(\mathsf{done}[\,] \mid a[\,]) \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\
\cong & (\boldsymbol{\nu}\tilde{m})(k[M_1 \mid M_2] \mid n[\circ \mid P_3]) \bullet P \mid \mathsf{done}[\,] \\
= & M' \bullet P \mid \mathsf{done}[\,]
\end{array}$$

This implies $C_{k.\texttt{exit\_}n}[M] \bullet P \Rightarrow\!\cong M' \bullet P \mid \mathsf{done}[\,]$.

$\alpha = n.\overline{\texttt{enter}}\texttt{\_}k$ Let P be a process. We know that $M \xrightarrow{\boldsymbol{n.\overline{\texttt{enter}}\texttt{\_}k}} M'$. Then

$$M \equiv (\boldsymbol{\nu}\tilde{m})(n[M_1] \mid M_2)$$

where $(\{n,k\} \cup \mathrm{fn}(P)) \cap \{\tilde{m}\} = \emptyset$, and

$$M' \equiv (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid k[\,\circ\,]] \mid M_2).$$

Now,

$$\begin{array}{rl}
& C_{n.\overline{\texttt{enter}}\texttt{_}k}[M] \bullet P \\
\equiv & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\texttt{in_}n.k[\texttt{out_}a.(P \mid (\boldsymbol{\nu}b)b[\texttt{out_}k.\texttt{out_}n.\mathsf{done}[\texttt{out_}b]])]] \mid n[M_1] \mid M_2) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[k[\texttt{out_}a.(P \mid (\boldsymbol{\nu}b)b[\texttt{out_}k.\texttt{out_}n.\mathsf{done}[\texttt{out_}b]])]]] \mid M_2) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P \mid (\boldsymbol{\nu}b)b[\texttt{out_}k.\texttt{out_}n.\mathsf{done}[\texttt{out_}b]]]] \mid M_2) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P] \mid (\boldsymbol{\nu}b)b[\texttt{out_}n.\mathsf{done}[\texttt{out_}b]]] \mid M_2) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P]] \mid (\boldsymbol{\nu}b)b[\mathsf{done}[\texttt{out_}b]] \mid M_2) \\
\rightarrowtriangle & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P]] \mid (\boldsymbol{\nu}b)b[] \mid \mathsf{done}[] \mid M_2) \\
\cong & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid k[\,\circ\,]] \mid M_2) \bullet P \mid \mathsf{done}[\,] \\
= & M' \bullet P \mid \mathsf{done}[\,]
\end{array}$$

This implies $C_{n.\overline{\texttt{enter}}\texttt{\_}k}[M] \bullet P \Rightarrow\!\cong M' \bullet P \mid \mathsf{done}[\,]$.

Table 8 Auxiliary contexts and processes

$$
\begin{array}{lcl}
-_1 \oplus -_2 & = & (\boldsymbol{\nu} o)(o[\,] \mid \mathtt{open}\_o.-_1 \mid \mathtt{open}\_o.-_2) \\
\mathrm{SPY}_\alpha\langle i,j,-\rangle & = & (i[\mathtt{out}\_n] \mid -) \oplus (j[\mathtt{out}\_n] \mid -) \\
 & & \quad \text{if } \alpha \in \{k.\mathtt{enter}\_n, k.\mathtt{exit}\_n, k.\mathtt{open}\_n, *.\mathtt{enter}\_n, *.\mathtt{exit}\_n\} \\
\mathrm{SPY}_\alpha\langle i,j,-\rangle & = & (i[\mathtt{out}\_k.\mathtt{out}\_n] \mid -) \oplus (j[\mathtt{out}\_k.\mathtt{out}\_n] \mid -) \text{ if } \alpha \in \{n.\overline{\mathtt{enter}}\_k\}
\end{array}
$$

$\alpha = k.\mathtt{open}\_n$ Let P be a process. We know that $M \xrightarrow{k.\mathtt{open}\_n} M'$. Then $M \equiv (\boldsymbol{\nu}\tilde{m})(n[M_1] \mid M_2)$, where $n \in \{\tilde{m}\}$, and $M' \equiv k[\circ \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)]$. Names a and b are fresh for M. Now,

$$
\begin{array}{rl}
 & C_{k.\mathtt{open}\_n}[M] \bullet P \\
\equiv & k[P \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathtt{done}[\mathtt{out}\_k] \mid \\
 & \quad a[(\boldsymbol{\nu}\tilde{m})(n[M_1] \mid M_2) \mid \mathtt{open}\_n.b[\mathtt{out}\_a]])] \\
\rightarrowtriangle & k[P \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathtt{done}[\mathtt{out}\_k] \mid a[(\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2) \mid b[\mathtt{out}\_a]])] \\
\rightarrowtriangle & k[P \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathtt{done}[\mathtt{out}\_k] \mid a[(\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)] \mid b[])] \\
\rightarrowtriangle & k[P \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_a.\mathtt{done}[\mathtt{out}\_k] \mid a[(\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)])] \\
\rightarrowtriangle & k[P \mid (\boldsymbol{\nu} a,b)(\mathtt{done}[\mathtt{out}\_k] \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2))] \\
\rightarrowtriangle & k[P \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2))]\mathtt{done}[] \\
\equiv & k[\circ \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)] \bullet P \mid \mathtt{done}[\,] \\
= & M' \bullet P \mid \mathtt{done}[\,]
\end{array}
$$

This implies $C_{k.\mathtt{open}\_n}[M] \bullet P \Rightarrow\cong M' \bullet P \mid \mathtt{done}[\,]$.

□

To complete the correspondence proof between actions α and their contexts $C_\alpha[-]$, we have to prove the converse of Lemma 3.9, formalised in Lemma 3.12. Such result requires a few technical definitions given in Table 8.

The symbol $\oplus$ denotes a form of internal choice, whereas the context $\mathrm{SPY}_\alpha\langle i,j,-\rangle$ is a technical tool to guarantee that the process P provided by the environment does not perform any action. This is necessary when proving completeness to guarantee that the contribution P is the same in both sides. The ability of $\mathrm{SPY}_\alpha\langle i,j,P\rangle$ to "spy" on P derives from the fact that one the two fresh barbs i and j is lost when P performs any action. The property of $\mathrm{SPY}_\alpha\langle i,j,-\rangle$ is captured in the following lemma.

Lemma 3.10 *Let M be a system which may possibly contain an occurrence of the special process $\circ$. If $M \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} O$ and $O \Downarrow_{i,j}$, where i,j are fresh for P and M, then there exists a system M' such that:*

1. $O = M' \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$;
2. $M \xrightarrow{\tau} M'$.

Proof For 1), the definition of $\bullet$ assures that there exists an arbitrary context $C[-]$ such that $C[\mathrm{SPY}_\alpha\langle i,j,P\rangle] = M \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$, and names in P are not bound in $C[-]$. The construction of $\mathrm{SPY}_\alpha\langle i,j,P\rangle$ assures that if $C[\mathrm{SPY}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} Q$, then either there is an arbitrary context C' such that $Q = C'[\mathrm{SPY}_\alpha\langle i,j,P\rangle]$, or $Q = C[P']$ where $\mathrm{SPY}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} P'$. But if $\mathrm{SPY}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} P'$, then $P' \Downarrow i \not\Downarrow j$, or $P' \Downarrow j \not\Downarrow i$. As $O \Downarrow_{i,j}$, O must be the outcome of the first reduction, and as such there exists an arbitrary context $C'[-]$ such that $O = C'[\mathrm{SPY}_\alpha\langle i,j,P\rangle]$. Let $M' = C'[\circ]$. As $C[\mathrm{SPY}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} C'[\mathrm{SPY}_\alpha\langle i,j,P\rangle]$, names in P cannot be bound in $C'[-]$. This implies $O = C'[\mathrm{SPY}_\alpha\langle i,j,P\rangle] = M' \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$, as required for 1).

For 2), $M \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle = C[\mathrm{SPY}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} C'[\mathrm{SPY}_\alpha\langle i,j,P\rangle] = M' \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ implies $M = C[\circ] \xrightarrow{\tau} C'[\circ] = M'$, as required. □

We also need a simple result on arbitrary contexts.

Lemma 3.11 *Let $C[-]$ and $C'[-]$ be arbitrary contexts, P, P' processes, and r a fresh name for $C[-]$ and P, such that $C[r[P]] \xrightarrow{\tau} C'[r[P']]$. Then $C[\mathbf{0}] \xrightarrow{\tau} C'[\mathbf{0}]$.*

We can finally prove the correspondence between actions and contexts.

Lemma 3.12 *Let M be a system, let $\alpha \in \{k.\mathtt{enter}\_n, k.\mathtt{exit}\_n, n.\overline{\mathtt{enter}}\_k, k.\mathtt{open}\_n\}$, and let i, j be fresh names for M. For all processes P with $\{i,j\} \cap \mathrm{fn}(P) = \emptyset$, if $C_\alpha[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \Rightarrow\cong O \mid \mathsf{done}[\,]$ and $O \Downarrow_{i,j}$, then there exists a system M' such that $O \cong M' \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ and $M \overset{\alpha}{\Longrightarrow} M'$.*

Proof The proof depends on the precise definition of the context. The main argument is that in the reduction

$$C_\alpha[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \Rightarrow\cong O \mid \mathsf{done}[\,]$$

the fresh ambient $\mathsf{done}[\,]$ can only be unleashed if M performs the action α, possibly preceded or followed by some internal actions. The fresh barbs i, j assure that the process P does not take part in the reduction, and that the component $\mathrm{SPY}_\alpha\langle i,j,P\rangle$ is found intact after the reduction. The barbed congruence is used to garbage collect a possible $(\boldsymbol{\nu} a)a[\,]$ ambient. We proceed by case analysis on α.

$\alpha = k.\mathtt{enter}\_n$**.** Observe that

$$C_\alpha[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle = n[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\mathtt{in}\_k.\mathtt{out}\_k.\mathtt{out}\_n]] \mid M\ .$$

As $O \Downarrow_{i,j}$ and done is fresh, by several applications of Lemma 3.10, there must be a system $D[-]$ such that $O \mid \mathsf{done}[\,] \equiv D[\mathsf{done}[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ and $C_\alpha[M] \Rightarrow D[\mathsf{done}[\,]]$. As P cannot reduce and done is fresh, the ambient n does not migrate during the reduction. Moreover, as M is a system, the ambient n cannot be opened. Also observe that the ambient done must consume the prefix $\mathtt{in}\_k$, thus requiring the presence of an ambient k inside the ambient n during the reduction. More precisely,

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

$$\begin{array}{rl} & C_\alpha[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\ = & n[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\mathtt{in\_}k.\mathtt{out\_}k.\mathtt{out\_}n]] \mid M \\ \Rightarrow \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\mathtt{in\_}k.\mathtt{out\_}k.\mathtt{out\_}n] \mid M_1] \mid M_2) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid C[\mathsf{done}[\mathtt{out\_}k.\mathtt{out\_}n]]] \mid M_2) \\ \Rightarrow & D[\mathsf{done}[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\ \equiv & D[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] \\ \equiv & O \mid \mathsf{done}[\,] \end{array}$$

Examining the above reductions sequence from $C_\alpha[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ we conclude that

$$M \Rightarrow \xrightarrow{k.\mathtt{enter\_}n} (\boldsymbol{\nu}\tilde{m})(n[\circ \mid M_1] \mid M_2).$$

As the name done is fresh for M, by several applications of Lemma 3.11, we also have that

$$(\boldsymbol{\nu}\tilde{m})(n[\circ \mid \mathbf{0} \mid M_1] \mid M_2) \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \Rightarrow D[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle.$$

Repeated application of Lemma 3.10(2) gives $(\boldsymbol{\nu}\tilde{m})(n[\circ \mid \mathbf{0} \mid M_1] \mid M_2) \Rightarrow D[\mathbf{0}]$, and therefore, as $\equiv$ is closed under reduction, there is a M', $M' \equiv D[\mathbf{0}]$, such that $M \xrightarrow{k.\mathtt{enter\_}n} M'$, as desired.

$\alpha = k.\mathtt{exit\_}n$. Observe that

$$C_{k.\mathtt{exit\_}n}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle = (\boldsymbol{\nu}a)a[\mathtt{in\_}k.\mathtt{out\_}k.\mathsf{done}[\mathtt{out\_}a]] \mid n[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid M]\ .$$

To unleash the ambient done, the ambient a must perform both its capabilities, and as its name is restricted the ambient a will be empty at the end of reduction. As P cannot reduce, and M is a system, the ambient n does not migrate during the reduction. Also observe that the ambient a must consume the prefix $\mathtt{in\_}k$, thus requiring the presence of an ambient k at top-level. More precisely, there exist a system M_1 and a static contexts $D[-]$ and $E[-]$ such that:

$$\begin{array}{rll} & C_{k.\mathtt{exit\_}n}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle & \\ = & (\boldsymbol{\nu}a)a[\mathtt{in\_}k.\mathtt{out\_}k.\mathsf{done}[\mathtt{out\_}a]] \mid n[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid M] & \\ \Rightarrow & (\boldsymbol{\nu}a)a[\mathtt{in\_}k.\mathtt{out\_}k.\mathsf{done}[\mathtt{out\_}a]] \mid M_1 \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle & \\ \xrightarrow{\tau} & (\boldsymbol{\nu}a)D[a[\mathtt{out\_}k.\mathsf{done}[\mathtt{out\_}a]]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle & \\ \Rightarrow & (\boldsymbol{\nu}a)E[\mathsf{done}[\,] \mid a[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle & (\star) \\ \cong & E[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] & \\ \equiv & O \mid \mathsf{done}[\,] & \end{array}$$

Examining the above reductions sequence from $C_{k.\mathtt{exit\_}n}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ we conclude that

$$M \Rightarrow \xrightarrow{k.\mathtt{exit\_}n} M_1.$$

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

$$\begin{array}{l}(\boldsymbol{\nu} a)a[\mathtt{in\_}k.\mathtt{out\_}k.\mathbf{0}] \mid M_1 \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\ \Rightarrow (\boldsymbol{\nu} a)E[\mathbf{0} \mid a[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle.\end{array}$$

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

$$\begin{array}{l}(\boldsymbol{\nu} a)(\mathbf{0} \mid M_1) \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\ \Rightarrow (\boldsymbol{\nu} a)E[\mathbf{0} \mid \mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle.\end{array}$$

Summarising,

$$M_1 \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \equiv (\boldsymbol{\nu} a)(\mathbf{0} \mid M_1) \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \Rightarrow (\boldsymbol{\nu} a)E[\mathbf{0} \mid \mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$$

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

$$M_1 \Rightarrow\!\equiv E[\mathbf{0}].$$

So, assuming $M' = E[\mathbf{0}]$, we can conclude.

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

$$\begin{array}{l}C_\alpha[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle = \\ \quad (\boldsymbol{\nu} a)a[\mathtt{in\_}n.k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathsf{done}[\mathtt{out\_}b]])]] \mid M\end{array}$$

To unleash the ambient done, the ambient a must use its $\mathtt{in\_}n$ capability, and the ambient b must exit from a. Moreover the ambient b must unleash all its capabilities. This implies that at the end of the reduction both secret ambients a and b will be empty. Also observe that the prefix $\mathtt{in\_}n$ must be consumed, thus requiring the presence of an ambient n at top-level.

More precisely, there exist a system M_1 and static contexts $D[-]$ and $E[-]$ such that

$$\begin{array}{rl}& C_{n.\overline{\mathtt{enter}}\_k}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\ = & (\boldsymbol{\nu} a)a[\mathtt{in\_}n.(k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathsf{done}[\mathtt{out\_}b]])])] \mid M \\ \Rightarrow & (\boldsymbol{\nu} a)a[\mathtt{in\_}n.(k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathsf{done}[\mathtt{out\_}b]])])] \mid M_1 \\ \xrightarrow{\tau} & D[(\boldsymbol{\nu} a)a[k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathsf{done}[\mathtt{out\_}b]]])]] \\ \Rightarrow & E[\mathsf{done}[] \mid (\boldsymbol{\nu} b)b[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \quad (\star) \\ \cong & E[\mathsf{done}[]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\ \cong & E[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[] \\ = & O \mid \mathsf{done}[\,]\end{array}$$

Observe that,

$$D[(\boldsymbol{\nu}a)a[k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu}b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathsf{done}[\mathtt{out\_}b]]])]]$$
$$\cong D[k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu}b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathsf{done}[\mathtt{out\_}b]]]]$$

Thus, by examining the above reductions sequence from $C_{n.\overline{\mathtt{enter}}\_k}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ we conclude that

$$M \Rightarrow \xrightarrow{\boldsymbol{n.\overline{\mathtt{enter}}\_k}} \cong D[k[\circ]].$$

As the name done is fresh, several applications of Lemma 3.11 to the above reduction marked by $(\star)$ gives:

$$D[(\boldsymbol{\nu}a)a[k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu}b)b[\mathtt{out\_}k.\mathtt{out\_}n.\mathbf{0}])]]]$$
$$\Rightarrow E[\mathbf{0} \mid (\boldsymbol{\nu}b)b[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$$

Again, as b is fresh, by several applications of Lemma 3.11 and reducing underneath $(\boldsymbol{\nu}b)$, we have:

$$D[(\boldsymbol{\nu}a)a[k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathbf{0})]]]$$
$$\Rightarrow E[\mathbf{0} \mid \mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle.$$

Summarising,

$$D[k[\mathrm{SPY}_\alpha\langle i,j,P\rangle]] \cong D[(\boldsymbol{\nu}a)a[k[\mathtt{out\_}a.(\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathbf{0})]]] \Rightarrow\cong E[\mathbf{0}]\bullet\mathrm{SPY}_\alpha\langle i,j,P\rangle.$$

and, as $\cong$ is closed under reduction, this implies:

$$D[k[\circ]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle = D[k[\mathrm{SPY}_\alpha\langle i,j,P\rangle]] \Rightarrow\equiv E[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle.$$

Now, by applying Lemma 3.10 there must be a system $\hat{M}$ such that: $D[k[\circ]] \Rightarrow \hat{M}$ and $\hat{M} \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \cong E[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$. Finally, as

$$M \Rightarrow \xrightarrow{\boldsymbol{n.\overline{\mathtt{enter}}\_k}} \cong D[k[\circ]]$$

and $\cong$ is reduction closed, there must by M' such that $M \Rightarrow M'$ and

$$M' \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \cong E[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \cong O$$

as required.

$\alpha = k.\mathtt{open\_}n$**.** Observe that

$$C_{k.\mathtt{open}\_n}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle =$$
$$k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu}a,b)(\mathtt{open\_}b.\mathtt{open\_}a.\mathsf{done}[\mathtt{out\_}k] \mid a[M \mid \mathtt{open\_}n.b[\mathtt{out\_}a]])]$$

where a and b are fresh. To unleash the ambient done, the ambient a must use its $\mathtt{open\_}n$ capability, and the ambient b must exit from a. Moreover both the empty ambients a and b will be opened before done is activated. Also observe that the prefix $\mathtt{open\_}n$ must be consumed, thus requiring the presence of an ambient n inside the

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

$$\begin{array}{ll}
 & C_{k.\mathtt{open}_n}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\
= & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[M \mid \mathtt{open}_n.b[\mathtt{out}_a]])] \\
\Rightarrow & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[M_1 \mid \mathtt{open}_n.b[\mathtt{out}_a]])] \\
\xrightarrow{\tau} & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[Q \mid b[\mathtt{out}_a]])] \\
\Rightarrow & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[Q_1 \mid b[\mathtt{out}_a]])] \\
\xrightarrow{\tau} & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid b[\,] \mid a[Q_1])] \\
\Rightarrow & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid b[\,] \mid a[Q_2])] \\
\xrightarrow{\tau} & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid \mathbf{0} \mid a[Q_2])] \\
\Rightarrow & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid \mathbf{0} \mid a[Q_3])] \\
\Rightarrow & k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathsf{done}[\mathtt{out}_k] \mid \mathbf{0} \mid Q_3)] \\
\Rightarrow & D[\mathsf{done}[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \\
\equiv & D[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] \\
= & O \mid \mathsf{done}[\,]
\end{array}$$

Examining the above reductions sequence from $C_{k.\mathtt{open}\_n}[M] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$ we conclude that

$$M \Rightarrow \xrightarrow{\boldsymbol{k}.\mathtt{open}\_\boldsymbol{n}} k[\circ \mid Q].$$

As

$$\begin{array}{l}
k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathsf{done}[\mathtt{out}_k] \mid a[Q \mid b[\mathtt{out}_a]])] \\
\Rightarrow D[\mathsf{done}[\,]] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle
\end{array}$$

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

$$\begin{array}{l}
k[\mathrm{SPY}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}_b.\mathtt{open}_a.\mathbf{0} \mid a[Q \mid b[\mathtt{out}_a]])] \\
\Rightarrow D[\mathbf{0}] \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle.
\end{array}$$

By Lemma 3.10, this implies

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

Applying our proof techniques we can easily prove that:

$$k[\circ \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathbf{0} \mid a[Q \mid b[\mathtt{out}\_a]])] \cong k[\circ \mid Q].$$

As $\cong$ is closed under reduction, it follows that there is M' such that

$$k[\circ \mid Q] \Rightarrow M' \cong D[\mathbf{0}].$$

So, there is M' such that $M \Rightarrow M'$ and $O \cong M' \bullet \mathrm{SPY}_\alpha\langle i,j,P\rangle$, as desired.

□

Theorem 3.13 (Completeness) *Reduction barbed congruence is contained in late bisimilarity.*

Proof We prove that the relation $\mathcal{R} = \{(M, N) \mid M \cong N\}$ is a late bisimulation. The result will then follow by co-induction.

- Suppose $M \mathcal{R} N$ and $M \xrightarrow{\boldsymbol{\alpha}} M'$. Suppose also $\alpha \in \{k.\texttt{enter\_}n, k.\texttt{exit\_}n, n.\overline{\texttt{enter}}\texttt{\_}k, k.\texttt{open\_}n\}$. We must find a system N' such that $N \stackrel{\boldsymbol{\alpha}}{\Longrightarrow} N'$ and for all P, $M' \bullet P \cong N' \bullet P$.

  The idea of the proof is to use a particular context which mimics the effect of the action α, and also allows us to subsequently compare the residuals of the two systems. This context has the form

  $$D_\alpha\langle P\rangle[-] \;=\; (C_\alpha[-] \mid \mathsf{Flip}) \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle$$

  where $C_\alpha[-]$ are the contexts in Table 7 and Flip is the system:

  $$(\boldsymbol{\nu} k)k[\texttt{in\_done.out\_done.}(\mathsf{succ}[\texttt{out\_}k] \oplus \mathsf{fail}[\texttt{out\_}k])]$$

  with succ and fail are fresh names. Intuitively, the existence of the fresh barb fail indicates that the action α has not yet happened, whereas the presence of succ together with the absence of fail ensures that the action α has been performed, and has been reported via done.

  As $\cong$ is contextual, $M \cong N$ implies that, for all processes P, it holds

  $$D_\alpha\langle P\rangle[M] \;\cong\; D_\alpha\langle P\rangle[N]\ .$$

  By Lemma 3.9, and by inspecting the reduction of the Flip process, we observe that:

  $$\begin{array}{rll} D_\alpha\langle P\rangle[M] & \Longrightarrow\cong & M' \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle \mid \mathsf{done}[\,] \mid \mathsf{Flip} \\ & \Longrightarrow\cong & M' \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,] \end{array}$$

  where $M' \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,] \Downarrow_{i,j,\mathsf{succ}} \not\Downarrow_{\mathsf{fail}}$. Call this outcome O_1.

  This reduction must be matched by a corresponding reduction

  $$D_\alpha\langle P\rangle[N] \;\Longrightarrow\; O_2$$

  where $O_1 \cong O_2$. However, the possible matching reductions are constrained by the barbs of O_1, because it must hold $O_2 \Downarrow_{i,j,\mathsf{succ}} \not\Downarrow_{\mathsf{fail}}$.

  As $O_2 \Downarrow_{\mathsf{succ}} \not\Downarrow_{\mathsf{fail}}$, it must be $O_2 \cong \hat{N} \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$, for some systems $\hat{N}$.

  As $O_2 \Downarrow_{i,j}$, the previous observation can be combined with Lemma 3.12 to derive the existence of a system (over the extended process syntax) N' such that $\hat{N} \cong N' \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle$ and a weak action

  $$N \stackrel{\boldsymbol{\alpha}}{\Longrightarrow} N'.$$

To conclude we must establish that for all P, it holds $M' \bullet P \;\cong\; N' \bullet P$. As barbed congruence is preserved by restriction, we have $(\boldsymbol{\nu}\mathsf{done}, \mathsf{succ})O_1 \cong (\boldsymbol{\nu}\mathsf{done}, \mathsf{succ})O_2$. As $(\boldsymbol{\nu}\mathsf{done})\mathsf{done}[\,] \cong (\boldsymbol{\nu}\mathsf{succ})\mathsf{succ}[\,] \cong \mathbf{0}$, it follows that

$$M' \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle \;\cong\; N' \bullet \mathrm{SPY}_\alpha\langle i, j, P\rangle.$$

Again, $\cong$ is preserved by restriction and $(\boldsymbol{\nu} i, j)\mathrm{SPY}_\alpha\langle i, j, P\rangle \cong P$. So, we can finally derive $M' \bullet P \;\mathcal{R}\; N' \bullet P$, for all processes P.

- Suppose now $M \;\mathcal{R}\; N$ and $M \xrightarrow{*.\mathtt{enter}\_\boldsymbol{n}} M'$, We must find a system N' such that $N \mid n[\,\circ\,] \Rightarrow N'$ and for all P, $M' \bullet P \cong N' \bullet P$.

  We consider the context

$$C\langle P\rangle[-] \;=\; - \mid n[\mathrm{SPY}_{*.\mathtt{enter}\_}n\langle i, j, P\rangle]\;.$$

  Because $\cong$ is contextual, for all processes P it holds

$$C\langle P\rangle[M] \;\cong\; C\langle P\rangle[N]\;.$$

  By inspecting the reduction rules of $C\langle P\rangle[M]$ we observe that,

$$C\langle P\rangle[M] \;\Rightarrow\; M' \bullet \mathrm{SPY}_{*.\mathtt{enter}\_}n\langle i, j, P\rangle$$

  where $M' \bullet \mathrm{SPY}_{*.\mathtt{enter}\_}n\langle i, j, P\rangle \Downarrow_{i,j}$. Call this outcome O_1.

  This reduction must be matched by a corresponding reduction

$$C\langle P\rangle[N] \;\Rightarrow\; O_2$$

  where $O_1 \cong O_2$ and $O_2 \Downarrow_{i,j}$. By several applications of Lemma 3.10 it follows that there is a system N' such that $O_2 = N' \bullet \mathrm{SPY}_{*.\mathtt{enter}\_}n\langle i, j, P\rangle$ and $N \mid n[\,\circ\,] \Rightarrow N'$. Again, as $\cong$ is preserved by restriction and $(\boldsymbol{\nu} i, j)\mathrm{SPY}_{*.\mathtt{enter}\_}n\langle i, j, P\rangle \cong P$, from $O_1 \cong O_2$ and the freshness of i and j we can derive $M' \bullet P \cong N' \bullet P$, for all P, as required.

- Suppose at last $M\mathcal{R}N$ and $M \xrightarrow{*.\mathtt{exit}\_\boldsymbol{n}} M'$. In this case we must find a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all P, $M' \bullet P \cong N' \bullet P$.

  We consider the context

$$C\langle P\rangle[-] \;=\; n[- \mid \mathrm{SPY}_{*.\mathtt{exit}\_}n\langle i, j, P\rangle]\;.$$

  Because $\cong$ is contextual, for all processes P it holds

$$C\langle P\rangle[M] \;\cong\; C\langle P\rangle[N]\;.$$

  By inspecting the reduction rules of $C\langle P\rangle[M]$ we observe that,

$$C\langle P\rangle[M] \;\Rightarrow\; M' \bullet \mathrm{SPY}_{*.\mathtt{exit}\_}n\langle i, j, P\rangle$$

where $M' \bullet \mathrm{SPY}_{*.\mathtt{exit}\_}n\langle i,j,P\rangle \Downarrow_{i,J}$. Call this outcome O_1.

This reduction must be matched by a corresponding reduction

$$C\langle P\rangle[N] \;\Rightarrow\; O_2$$

where $O_1 \cong O_2$ and $O_2 \Downarrow_{A,B}$. By several applications of Lemma 3.10 it follows that there is a system N' such that $O_2 = N' \bullet \mathrm{SPY}_{*.\mathtt{enter}\_}n\langle i,j,P\rangle$ and $n[\circ \mid N] \Rightarrow N'$. Again, as $\cong$ is preserved by restriction and $(\boldsymbol{\nu} i,j)\mathrm{SPY}_{*.\mathtt{exit}\_}n\langle i,j,P\rangle \cong P$, from $O_1 \cong O_2$ and the freshness of i and j we can derive $M' \bullet P \cong N' \bullet P$, for all P, as required.

and in turn that $O_2 \equiv N' \bullet \mathrm{SPY}_{*.\mathtt{exit}\_}n\langle i,j,P\rangle$. We can derive that $n[\circ \mid N] \Rightarrow N'$, and conclude that for all P, $M' \bullet P \cong N' \bullet P$ because of Lemma 3.10.

□

As a consequence:

Theorem 3.14 *Late bisimilarity, early bisimilarity, and reduction barbed congruence they all coincide.*

Proof Theorem 3.8 states that $\approx \subseteq \approx_{\mathrm{e}}$ and $\approx_{\mathrm{e}} \subseteq \cong$. Theorem 3.13 states the reduction barbed congruence is contained in late bisimilarity, that is $\cong \subseteq \approx$. We hence have the following chain of inclusions $\cong \subseteq \approx \subseteq \approx_{\mathrm{e}} \subseteq \cong$. □

4 Up-to Proof Techniques

In the previous section we gave a labelled characterisation of reduction barbed congruence to prove that two systems have the same behaviour. In this section we adapt some well-known *up-to* proof techniques [22, 27] to our setting. As usual, these techniques allow us to reduce the size of the relation $\mathcal{R}$ for proving that two processes are bisimilar. We focus on two forms of up-to techniques: the *up-to-expansion* [29] and the *up-to-context* technique [26]. As in the π-calculus, these two techniques can also be merged. As a consequence, we only prove the more general up to context and up to expansion proof-technique.

Roughly, the expansion [2], written $\lesssim$, is an asymmetric variant of the bisimilarity which allows us to count the number of silent moves performed by a process. More precisely, $M \lesssim N$ holds if M and N are bisimilar and N has at least as many τ-moves as M. Formally,

Definition 4.1 (Expansion) *A relation $\mathcal{R}$ over systems is an* expansion *if $M \mathcal{R} N$ implies:*

- *if $M \xrightarrow{\boldsymbol{\alpha}} M'$, $\alpha \notin \{\mathtt{enter}\_n, \mathtt{exit}\_n\}$, then there exists a system N' such that $N \stackrel{\hat{\boldsymbol{\alpha}}}{\Longrightarrow} N'$ and for all processes P it holds $M' \bullet P \;\mathcal{R}\; N' \bullet P$;*
- *if $M \xrightarrow{\mathtt{enter}\_\boldsymbol{n}} M'$ then there exists a system N' such that $N \mid n[\,\circ\,] \Rightarrow N'$ and for all processes P it holds $M' \bullet P \;\mathcal{R}\; N' \bullet P$;*
- *if $M \xrightarrow{\mathtt{exit}\_\boldsymbol{n}} M'$ then there exists a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all processes P it holds $M' \bullet P \;\mathcal{R}\; N' \bullet P$;*

- *if* $N \xrightarrow{\boldsymbol{\alpha}} N'$, $\alpha \notin \{\texttt{enter\_}n, \texttt{exit\_}n\}$, *then there exists a system* M' *such that* $M \xrightarrow{\hat{\boldsymbol{\alpha}}} M'$ *and for all processes* P *it holds* $M' \bullet P \mathcal{R} N' \bullet P$;

- *if* $N \xrightarrow{\texttt{enter\_}\boldsymbol{n}} N'$ *then* $(M \mid n[P]) \mathcal{R} N' \bullet P$, *for all processes* P;

- *if* $N \xrightarrow{\texttt{exit\_}\boldsymbol{n}} N'$ *then* $n[M \mid P] \mathcal{R} N' \bullet P$, *for all processes* P.

We write $M \lesssim N$, *if* $M \mathcal{R} N$ *for some expansion* $\mathcal{R}$.

Definition 4.2 (Bisimulation up to $\gtrsim$) *A relation* $\mathcal{R}$ *is a* bisimulation up to $\gtrsim$ and $\approx$ *if* $M \mathcal{R} N$ *implies:*

- *if* $M \xrightarrow{\boldsymbol{\alpha}} M'$, $\alpha \notin \{\texttt{enter\_}n, \texttt{exit\_}n\}$, *then there exists a system* N' *such that* $N \stackrel{\hat{\boldsymbol{\alpha}}}{\Longrightarrow} N'$ *and for all processes* P *it holds* $M' \bullet P \gtrsim \mathcal{R} \lesssim N' \bullet P$;

- *if* $M \xrightarrow{\texttt{enter\_}\boldsymbol{n}} M'$ *then there exists a system* N' *such that* $N \mid n[\,\circ\,] \Longrightarrow N'$ *and for all processes* P *it holds* $M' \bullet P \gtrsim \mathcal{R} \lesssim N' \bullet P$;

- *if* $M \xrightarrow{\texttt{exit\_}\boldsymbol{n}} M'$ *then there exists a system* N' *such that* $n[\circ \mid N] \Longrightarrow N'$ *and for all processes* P *it holds* $M' \bullet P \gtrsim \mathcal{R} \lesssim N' \bullet P$.

Theorem 4.3 *If* $\mathcal{R}$ *is a bisimulation up to* $\gtrsim$, *then it holds that* $\mathcal{R} \subseteq \approx$.

The proofs of Theorem 4.3 and of Theorem 4.5 below can be easily derived from the proof of Theorem 4.7.

Definition 4.4 (Bisimulation up to context) *A symmetric relation* $\mathcal{R}$ *is a* bisimulation up-to context *if* $P \mathcal{R} Q$ *implies:*

- *if* $M \xrightarrow{\boldsymbol{\alpha}} M''$, $\alpha \notin \{\texttt{enter\_}n, \texttt{exit\_}n\}$, *then there exists a system* N'' *such that* $N \stackrel{\hat{\boldsymbol{\alpha}}}{\Longrightarrow} N''$, *and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P = C[M']$, $N'' \bullet P = C[N']$, *and* $M' \mathcal{R} N'$;

- *if* $M \xrightarrow{\texttt{*.enter\_}\boldsymbol{n}} M''$ *then there exists a system* N'' *such that* $N \mid n[\,\circ\,] \Longrightarrow N''$, *and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P = C[M']$, $N'' \bullet P = C[N']$, *and* $M' \mathcal{R} N'$;

- *if* $M \xrightarrow{\texttt{*.exit\_}\boldsymbol{n}} M''$ *then there exists a system* N'' *such that* $n[\circ \mid N] \Longrightarrow N''$, *and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P = C[M']$, $N'' \bullet P = C[N']$, *and* $M' \mathcal{R} N'$.

Theorem 4.5 *If* $\mathcal{R}$ *is a bisimulation up to context, then it holds that* $\mathcal{R} \subseteq \approx$.

Definition 4.6 (Bisimulation up to context and up to $\gtrsim$) *A symmetric relation* $\mathcal{R}$ *is a* bisimulation up to context and up to $\gtrsim$ *if* $P \mathcal{R} Q$ *implies:*

- *if* $M \xrightarrow{\boldsymbol{\alpha}} M''$, $\alpha \notin \{\texttt{enter\_}n, \texttt{exit\_}n\}$, *then there exists a system* N'' *such that* $N \stackrel{\hat{\boldsymbol{\alpha}}}{\Longrightarrow} N''$, *and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, *and* $M' \mathcal{R} N'$;

- *if* $M \xrightarrow{\texttt{enter\_}\boldsymbol{n}} M''$ *then there exists a system* N'' *such that* $N \mid n[\circ] \Longrightarrow N''$, *and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, *and* $M' \mathcal{R} N'$;

- *if* $M \xrightarrow{\texttt{exit\_}\boldsymbol{n}} M''$ *then there exist a system* N'' *such that* $n[\circ \mid N] \Longrightarrow N''$, *and for all processes* P *there is a system context* $C[-]$ *and systems* M' *and* N' *such that* $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, *and* $M' \mathcal{R} N'$.

Theorem 4.7 *If* $\mathcal{R}$ *is a bisimulation up to context and up to* $\gtrsim$, *then* $\mathcal{R} \subseteq \approx$.

Proof We define the relation $\mathcal{S}$ as the smallest relation such that:

1. $M \mathcal{R} N$ implies $M \mathcal{S} N$;
2. $M \gtrsim A, A \mathcal{S} B, B \lesssim N$ implies $M \mathcal{S} N$;
3. $M \mathcal{S} N$ implies $C[M] \mathcal{S} C[N]$, for all system contexts $C[-]$.

We prove by induction on its definition, that $\mathcal{S}$ is a late bisimulation. This will assure the soundness of the relation $\mathcal{R}$, because $M \mathcal{R} N$ implies $M \mathcal{S} N$ which implies $M \approx N$. Observe that $\mathcal{S}$ is symmetric because $\mathcal{R}$ is.

- $M \mathcal{S} N$ because $M \mathcal{R} N$.

  Suppose that $M \xrightarrow{\boldsymbol{\alpha}} M''$, with $\alpha \notin \{*.\texttt{enter\_}n, *.\texttt{exit\_}n\}$. As $\mathcal{R}$ is a bisimulation up to context and up-to $\gtrsim$, we know that there exists a system N'' such that $N \stackrel{\boldsymbol{\alpha}}{\Longrightarrow} N''$. We also know that for all process P, there exist a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, and $M' \mathcal{R} N'$. This implies $M' \mathcal{S} N'$. By construction $\mathcal{S}$ is contextual and $C[M'] \mathcal{S} C[N']$ holds. By construction $\mathcal{S}$ is closed under expansion, and therefore $M'' \mathcal{S} N''$, as required.

  Suppose that $M \xrightarrow{*.\texttt{enter\_}\boldsymbol{n}} M''$. As $\mathcal{R}$ is a bisimulation up to context and up to $\gtrsim$, we know that there exists a system N'' such that $N \mid n[\circ] \stackrel{*.\texttt{enter\_}\boldsymbol{n}}{\Longrightarrow} N''$. We also know that for all process P, there exist a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, and $M' \mathcal{R} N'$. This implies $M' \mathcal{S} N'$. By construction, $\mathcal{S}$ is contextual, and $C[M'] \mathcal{S} C[N']$ holds. By construction $\mathcal{S}$ is closed under expansion, and therefore $M'' \mathcal{S} N''$, as required.

  Suppose that $M \xrightarrow{*.\texttt{exit\_}\boldsymbol{n}} M''$. As $\mathcal{R}$ is a bisimulation up to context and up to $\gtrsim$, we know that there exists a system N'' such that $n[\circ \mid N] \stackrel{*.\texttt{exit\_}\boldsymbol{n}}{\Longrightarrow} N''$. We also know that for all process P, there exist a system context $C[-]$ and systems M' and N' such that $M'' \bullet P \gtrsim C[M']$, $N'' \bullet P \gtrsim C[N']$, and $M' \mathcal{R} N'$. This implies $M' \mathcal{S} N'$. By construction, $\mathcal{S}$ is contextual, and $C[M'] \mathcal{S} C[N']$ holds. By construction $\mathcal{S}$ is closed under expansion, and we conclude $M'' \mathcal{S} N''$, as required.

- $M \mathcal{S} N$ because $M \gtrsim A, A \mathcal{S} B, B \lesssim N$.

  The induction hypothesis tells us that $A \mathcal{S} B$ behaves like a late bisimulation.

  Suppose $M \xrightarrow{\alpha} M'$, with $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$. A simple diagram chasing allows us to conclude that there are systems A', B', N' such that for all process P it holds $M' \bullet P \gtrsim A' \bullet P \ \mathcal{S} \ B' \bullet P \lesssim N' \bullet P$, and in turn, by construction of $\mathcal{S}$, $M' \bullet P \mathcal{S} N' \bullet P$.

  Suppose $M \xrightarrow{*.\texttt{enter}\_n} M'$. As $M \gtrsim A$, for all process P, it holds $M' \bullet P \gtrsim A \mid n[P]$. As $A \mathcal{S} B$, the closure properties of $\mathcal{S}$ assure that $A \mid n[P] \mathcal{S} B \mid n[P]$. The expansion relation is a congruence, and since $B \mathcal{S} N$ we conclude that $B \mid n[P] \lesssim N \mid n[P]$. But $N \mid n[P] \Rightarrow N \mid n[P]$, and $M' \bullet P \gtrsim \mathcal{S} \lesssim (N \mid n[\circ]) \bullet P$. This, by construction of $\mathcal{S}$, implies $M' \bullet P \mathcal{S} (N \mid n[\circ]) \bullet P$.

  Suppose $M \xrightarrow{*.\texttt{exit}\_n} M'$. As $M \gtrsim A$, for all process P, it holds $M' \bullet P \gtrsim n[P \mid A]$. As $A \mathcal{S} B$, the closure properties of $\mathcal{S}$ assure that $n[P \mid A] \mathcal{S} n[P \mid B]$. The expansion relation is a congruence, and since $B \mathcal{S} N$ we conclude that $n[P \mid A] \lesssim n[P \mid N]$. But $n[P \mid B] \Rightarrow n[P \mid N]$, and $M' \bullet P \gtrsim \mathcal{S} \lesssim n[\circ \mid N] \bullet P$. This, by construction of $\mathcal{S}$, implies $M' \bullet P \mathcal{S} n[\circ \mid N] \bullet P$.

- $C[M] \mathcal{S} C[N]$ because $M \mathcal{S} N$ and $C[-]$ is a system context.

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

□

5 Adding Communication

The basic idea is to have an *output process* such as $\langle E \rangle.P$, which outputs the message E and then continues as P, and an input process $(x)Q$ which on receiving a message binds it to x in Q which then executes; here occurrences of x in Q are bound. Notice that we have synchronous output; as discussed in [33, 28, 4] this is not unrealistic because communication in MA is always local. The syntax of our extended language, together with the reduction rule for communication, is given in Table 9.

The LTS is extended by the introduction of two new pre-actions (E) for input, $\langle - \rangle$ for output, and a new form of concretions $(\boldsymbol{\nu}\tilde{m})\langle E \rangle Q$. In Table 11 we give all the defining rules that should be added to those of Table 4 and Table 5 to obtain the LTS for the extended calculus. Note that in the structural rules of Table 4 we are now assuming that parallel composition and restriction distribute over the new form of concretions $(\boldsymbol{\nu}\tilde{m})\langle E \rangle Q$ in the same manner as $(\boldsymbol{\nu}\tilde{m})\langle P \rangle Q$. The slightly unusual pre-action for output allows a uniform treatment of extrusion of names.

The proof of Theorem 2.2 can be easily completed to take into account the extended calculus. A consequence of working with MA in two levels is that communication capabilities cannot be observed at top-level. Moreover, the free variables of a system cannot be

Table 9 The Message-passing Mobile Ambients in Two Levels

Names: $a, b, \ldots, k, l, m, n, \ldots \in \mathbf{N}$

Capabilities:

C	$::=$	$\texttt{in\_}n$	may enter into n
	$\mid$	$\texttt{out\_}n$	may exit out of n
	$\mid$	$\texttt{open\_}n$	may open n

Expressions:

E, F	$::=$	x	variable
	$\mid$	C	capability
	$\mid$	$E.F$	path
	$\mid$	ε	empty path

Guards:

G	$::=$	E	expression
	$\mid$	(x)	input
	$\mid$	$\langle E \rangle$	output

Systems:

M, N	$::=$	$\mathbf{0}$	termination
	$\mid$	$M_1 \mid M_2$	parallel composition
	$\mid$	$(\boldsymbol{\nu} n)M$	restriction
	$\mid$	$n[P]$	ambient

Processes:

P, Q, R	$::=$	$\mathbf{0}$	nil process
	$\mid$	$P_1 \mid P_2$	parallel composition
	$\mid$	$(\boldsymbol{\nu} n)P$	restriction
	$\mid$	$G.P$	prefixing
	$\mid$	$n[P]$	ambient
	$\mid$	$!G.P$	replication

Structural and Reduction rules for Communication:

$E.(F.P) \equiv (E.F).P$	(Struct Path)
$\varepsilon.P \rightarrowtriangle P$	(Red Empty Path)
$(x).P \mid \langle M \rangle.Q \rightarrowtriangle P\{^M/_x\} \mid Q$	(Red Comm)

Table 10 Pre-actions and Concretions for Communication

Pre-actions: π	$::=$	$\ldots$	*Concretions:* K	$::=$	$(\boldsymbol{\nu}\tilde{m})\langle P \rangle Q$
	$\mid$	(E) $\mid$ $\langle - \rangle$		$\mid$	$(\boldsymbol{\nu}\tilde{m})\langle E \rangle Q$

Table 11 Labelled Transition System - Communication

$$(\pi \text{ Output}) \ \frac{-}{\langle E\rangle.P \xrightarrow{\langle - \rangle} \langle E\rangle P} \qquad (\pi \text{ Input}) \ \frac{-}{(x).P \xrightarrow{(\boldsymbol{E})} P\{^{E}/_{x}\}}$$

$$(\pi \text{ Path}) \ \frac{E.(F.P) \xrightarrow{\boldsymbol{\pi}} Q}{(E.F).P \xrightarrow{\boldsymbol{\pi}} Q} \qquad (\tau \text{ Eps}) \ \frac{-}{\epsilon.P \xrightarrow{\boldsymbol{\tau}} P}$$

$$(\tau \text{ Comm}) \ \frac{P \xrightarrow{\langle - \rangle} (\boldsymbol{\nu}\tilde{m})\langle E\rangle P' \quad Q \xrightarrow{(\boldsymbol{E})} Q' \quad \text{fn}(Q') \cap \{\tilde{m}\} = \emptyset}{P \mid Q \xrightarrow{\boldsymbol{\tau}} (\boldsymbol{\nu}\tilde{m})(P' \mid Q')}$$

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 a collection of algebraic properties using our bisimulation proof methods. Then, we prove the correctness of a protocol for controlling access through a *firewall*, first proposed in [7].

We briefly comment on the laws of Theorem 6.1. The first two laws are two examples of local communication within private ambients without interference. The third law is the well-known perfect firewall law. The following four laws represent non-interference properties about movements of private ambients. Finally, the last two laws say when opening cannot be interfered.

Theorem 6.1

1. $(\boldsymbol{\nu}n)n[\langle W\rangle.P \mid (x).Q \mid M] \ \cong \ (\boldsymbol{\nu}n)n[P \mid Q\{^{W}/_{x}\} \mid M]$ *if* $n \notin \text{fn}(M)$
2. $(\boldsymbol{\nu}n)n[\langle W\rangle.P \mid (x).Q \mid \prod_{j\in J} \texttt{open\_}k_j.R_j] \ \cong \ (\boldsymbol{\nu}n)n[P \mid Q\{^{W}/_{x}\} \mid \prod_{j\in J} \texttt{open\_}k_j.R_j]$
3. $(\boldsymbol{\nu}n)n[P] \ \cong \ \mathbf{0}$ *if* $n \notin \text{fn}(P)$
4. $(\boldsymbol{\nu}n)((\boldsymbol{\nu}m)m[\texttt{in\_}n.P] \mid n[M]) \ \cong \ (\boldsymbol{\nu}n)n[(\boldsymbol{\nu}m)m[P] \mid M]$ *if* $n \notin \text{fn}(M)$
5. $(\boldsymbol{\nu}m,n)(m[\texttt{in\_}n.P] \mid n[\prod_{j\in J} \texttt{open\_}k_j.R_j]) \ \cong \ (\boldsymbol{\nu}m,n)n[m[P] \mid \prod_{j\in J} \texttt{open\_}k_j.R_j]$
6. $(\boldsymbol{\nu}n)n[(\boldsymbol{\nu}m)m[\texttt{out\_}n.P] \mid M] \ \cong \ (\boldsymbol{\nu}n)((\boldsymbol{\nu}m)m[P] \mid n[M])$ *if* $n \notin \text{fn}(M)$

7. $(\boldsymbol{\nu} n)n[m[\texttt{out\_}n.P] \mid \prod_{j\in J} \texttt{open\_}k_j.R_j] \;\cong\; (\boldsymbol{\nu} n)(m[P] \mid n[\prod_{j\in J} \texttt{open\_}k_j.R_j])$ *if* $m \neq k_j$, *for* $j \in J$

8. $n[(\boldsymbol{\nu} m)(\texttt{open\_}m.P \mid m[N]) \mid Q] \;\cong\; n[(\boldsymbol{\nu} m)(P \mid N) \mid Q]$ *if* $Q \equiv M \mid \prod_{j\in J}\langle W_j\rangle.R_j$ *and* $m \notin \mathrm{fn}(N)$

9. $(\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)(\texttt{open\_}m.P \mid m[Q]) \mid R] \;\cong\; (\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)(P \mid Q) \mid R]$ *if* $R \equiv \prod_{i\in I}\langle W_i\rangle.S_i \mid \prod_{j\in J} \texttt{open\_}k_j.R_j$ *and* $m, n \notin \mathrm{fn}(Q)$.

Proof The proofs of the above laws, except for (3) and (9), are by exhibiting the appropriate bisimulation. In all cases the bisimulation has a similar form:

$$\mathcal{S} = \{(LHS, RHS)\} \cup \approx$$

where LHS, RHS denote the left hand side, right hand side respectively of the equation, parameterised over names, processes and systems. For proving the laws (3) and (9) we need to show that the above $\mathcal{S}$ is a bisimulation up to context and up to structural congruence. The most delicate cases are those regarding the silent moves $*.\texttt{enter\_}k$ and $*.\texttt{exit\_}k$. For instance, if

$$(\boldsymbol{\nu} n)n[P] \xrightarrow{*.\texttt{enter\_}k} (\boldsymbol{\nu} n)k[\circ \mid n[P']] \equiv k[\circ \mid (\boldsymbol{\nu} n)n[P']]$$

then

$$\mathbf{0} \mid k[\,\circ\,] \Rightarrow\!\equiv k[\circ \mid \mathbf{0}]$$

and up to context and structural congruence we are still in $\mathcal{S}$. □

Crossing a firewall A protocol is discussed in [7] for controlling access through a firewall. An ambient w represents the firewall; an ambient m, a trusted agent inside which is a process Q that is supposed to cross the firewall. The firewall ambient sends into the agent a pilot ambient k with the capability $\texttt{in\_}w$ for entering the firewall. The agent acquires the capability by opening k. The process Q carried by the agent is finally liberated inside the firewall by the opening of ambient m. Names m and k act like passwords which guarantee the access only to authorised agents. Here is the protocol in MA:

$$\begin{array}{rcl} AG & \stackrel{\text{def}}{=} & m[\texttt{open\_}k.(x).x.Q] \\ FW & \stackrel{\text{def}}{=} & (\boldsymbol{\nu} w)w[\texttt{open\_}m.P \mid k[\texttt{out\_}w.\texttt{in\_}m.\langle\texttt{in\_}w\rangle]] \end{array}$$

The correctness (of a slight variant) of the protocol above is shown in [7] for may-testing [9] proving that

$$(\boldsymbol{\nu} m, k)(AG \mid FG) \;\cong\; (\boldsymbol{\nu} w)w[Q \mid P]$$

under the conditions that $w \notin \mathrm{fn}(Q)$, $x \notin \mathrm{fv}(Q)$, $\{m, k\} \cap (\mathrm{fn}(P) \cup \mathrm{fn}(Q)) = \emptyset$. The proof relies on non-trivial contextual reasonings. In what follows, we show how it can be established using our bisimulation proof methods.

The system on the right can be obtained from that one on the left by executing six τ-actions. So, it suffices to prove that $\cong$ is insensitive to all these τ-actions. The result follows from the algebraic laws of Theorem 6.1 and the following two laws:

Lemma 6.2 *Let P, Q, and R be processes. Then*

1. $(\boldsymbol{\nu} k, m, w)(k[\mathtt{in}\_m.P] \mid m[\mathtt{open}\_k.Q] \mid w[\mathtt{open}\_m.R])$
   $\cong\ (\boldsymbol{\nu} k, m, w)(m[k[P] \mid \mathtt{open}\_k.Q] \mid w[\mathtt{open}\_m.R])$

2. $(\boldsymbol{\nu} m, w)(m[\langle \mathtt{in}\_w \rangle \mid (x).P] \mid w[\mathtt{open}\_m.Q]) \cong (\boldsymbol{\nu} m, w)(m[P\{^{\mathtt{in}\_w}/_x\}] \mid w[\mathtt{open}\_m.Q])$

Proof By exhibiting the appropriate bisimulation. Again, in all cases the bisimulation has a similar form:

$$\mathcal{S} = \{(LHS, RHS)\} \cup \approx$$

where LHS, RHS denote the left hand side, right hand side respectively of the identity. □

Theorem 6.3 *If $w \notin \mathrm{fn}(Q)$, $x \notin \mathrm{fv}(Q)$, and $\{m, k\} \cap (\mathrm{fn}(P) \cup \mathrm{fn}(Q)) = \emptyset$, then*

$$(\boldsymbol{\nu} m, k)(AG \mid FG) \cong (\boldsymbol{\nu} w)w[Q \mid P].$$

Proof It suffices to apply the algebraic laws of Theorem 6.1 and Lemma 6.2. More precisely, we apply, in sequence, Law (7) of Theorem 6.1, Law (1) of Lemma 6.2, Law (9) of Theorem 6.1, Law (2) of Lemma 6.2, and Laws (5) and (9) of Theorem 6.1. □

7 Conclusion and Related Work

In this paper we study the behavioural theory of Cardelli and Gordon's Mobile Ambients. We rewrite the syntax of MA in two levels, thus distinguishing *processes* and *systems*. This little expedient allows us to gain new interesting algebraic laws without loosing expressive power.

The main results of the paper are:

- an LTS based operational semantics for MA,
- a bisimulation based equivalence over this LTS which coincides with reduction barbed congruence,
- up-to expansion and up-to context proof techniques.

We believe that interesting labelled characterisations of typed reduction barbed congruence can be derived along the lines of [19].

Higher-order LTSs for Mobile Ambients can be found in [6, 13, 32, 10]. But we are not aware of any form of bisimilarity defined using these LTSs. A simple first-order LTS for MA without restriction is proposed by Sangiorgi in [28]. Using this LTS the author defines an *intensional* bisimilarity for MA which separates terms on the basis of their internal structure.

Our work is the natural prosecution of [18] where an LTS and a labelled characterisation of reduction barbed congruence are given for a variant of Levi and Sangiorgi's Safe Ambients, called SAP. The main differences with respect to [18] are the following:

- SAP differs from MA for having co-capabilities and passwords, both features are essential to prove the characterisation result in SAP.

- 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 *late*, 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 $k.\texttt{enter}\_n$ we say that ambient k enters n. The knowledge of k is necessary to make the action observable for the environment. This is not needed in SAP, because movements can be observed by means of co-capabilities.

- Co-capabilities also allow the observation of the movement of an ambient whose name is private. As a consequence, the perfect firewall equation does not hold neither in SAP, nor in Safe Ambients. In MA, the movements of an ambient whose name is private cannot be observed. This is why the perfect firewall equation holds.

Finally, apart from [18], other forms of bisimilarity for higher-order calculi, such as Distributed π-calculus [14], Seal [33], Nomadic Pict [31], a Calculus for Mobile Resources [12], and NBA [5], can be found in [19, 8, 31, 12, 5], but only [19, 12, 5] prove a labelled characterisations of a contextually defined notion of equivalence. The perfect firewall equation as already been proved for Morris-style contextual equivalence in [13] using a context lemma.

Acknowledgements The authors would like to thank Vladimiro Sassone who spotted a problem in the proof of Theorem 4.7 in an early draft of the paper. The second author is grateful to the Foundations of Computing Group of University of Sussex, for the kind hospitality and support.

References

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

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

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

[4] M. Bugliesi, G. Castagna, and S. Crafa. Boxed ambients. In *Proc. 4th TACS*, volume 2215 of *LNCS*. Springer-Verlag, 2001.

[5] M. Bugliesi, S. Crafa, M. Merro, and V. Sassone. Communication interference in mobile boxed ambients. Forthcoming Technical Report. An extended abstract appeared in Proc. FSTTCS'02, LNCS, Springer-Verlag.

[6] L. Cardelli and A. Gordon. A commitment relation for the ambient calculus. Unpublished notes, 1996.

[7] L. Cardelli and A. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000. An extended abstract appeared in *Proc. of FoSSaCS '98*.

[8] G. Castagna and F. Zappa Nardelli. The seal calculus revisited: Contextual equivalence and bisimilarity. In *Proc. 22nd FSTTCS '02*, volume 2556 of *LNCS*. Springer-Verlag, 2002.

[9] R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.

[10] G. Ferrari, U. Montanari, and E. Tuosto. A LTS semantics of ambients via graph synchronization with mobility. In *Proc. ICTCS*, volume 2202 of *LNCS*, 2001.

[11] C. Fournet and G. Gonthier. A hierarchy of equivalences for asynchronous calculi. In *Proc. 25th ICALP*, pages 844–855, 1998.

[12] J.C. Godskesen, T. Hildebrandt, and V. Sassone. A calculus of mobile resources. In *Proc. 10th CONCUR '02*, volume 2421 of *LNCS*, 2002.

[13] A. D. Gordon and L. Cardelli. Equational properties of mobile ambients. *Journal of Mathematical Structures in Computer Science*, 12:1–38, 2002.

[14] M. Hennessy and J. Riely. A typed language for distributed mobile processes. In *Proc. 25th POPL*. ACM Press, 1998.

[15] K. Honda and M. Tokoro. An Object Calculus for Asynchronous Communications. In *Proc. ECOOP '91*, volume 512 of *LNCS*. Springer Verlag, 1991.

[16] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 152(2):437–486, 1995.

[17] F. Levi and D. Sangiorgi. Controlling interference in ambients. In *Proc. 27th POPL*. ACM Press, 2000.

[18] M. Merro and M. Hennessy. Bisimulation congruences in safe ambients. In *Proc. 29th POPL '02*. ACM Press, 2002.

[19] M. Hennessy M. Merro and J. Rathke. Towards a behavioural theory of access and mobility control in distributed system. To appear in Proc. 5th FoSSaCS '03, LNCS, 2003, Springer-Verlag.

[20] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

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

[22] R. Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. 19th ICALP*, volume 623 of *LNCS*, pages 685–695. Springer Verlag, 1992.

[23] D.M. Park. Concurrency on automata and infinite sequences. In P. Deussen, editor, *Conf. on Theoretical Computer Science*, volume 104 of *LNCS*. 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 *Proc. 28th POPL*. ACM Press, 2001.

[29] D. Sangiorgi and R. Milner. The problem of "Weak Bisimulation up to". In *Proc. CONCUR '92*, volume 630 of *LNCS*, pages 32–46. Springer Verlag, 1992.

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

[31] A. Unyapoth and P. Sewell. Nomadic Pict: Correct communication infrastructures for mobile computation. In *Proc. 28th POPL*. ACM, January 2001.

[32] M. G. Vigliotti. Transition systems for the ambient calculus. Master thesis, Imperial College of Science, Technology and Medicine (University of London), September 1999.

[33] J. Vitek and G. Castagna. Seal: A framework for secure mobile computations. In *Internet Programming Languages*, 1999.