Behavioural Theory for Mobile Ambients

MASSIMO MERRO
Dipartimento di Informatica, Università di Verona, Italy
and
FRANCESCO ZAPPA NARDELLI
INRIA Rocquencourt, France

We study a behavioural theory of *Mobile Ambients*, a process calculus for modelling mobile agents in wide-area networks, focussing on *reduction barbed congruence*. Our contribution is threefold. (1) We prove a *context lemma* which shows that only parallel and nesting contexts need be examined to recover this congruence. (2) We characterise this congruence using a *labelled bisimilarity*: this requires novel techniques to deal with asynchronous movements of agents and with the invisibility of migrations of secret locations. (3) We develop refined proof methods involving *up-to proof techniques*, which allow us to verify a set of *algebraic laws* and the correctness of more complex examples.

Categories and Subject Descriptors: F.3.2 [**Theory of Computation**]: Logics and Meanings of Programs—*Operational semantics*; *Process models*

General Terms: Languages, Theory

Additional Key Words and Phrases: Behavioural theories, bisimulation, concurrency, process calculi, programming languages

Introduction

Programming *wide-area networks* is inherently different from programming distributed applications over local networks [Cardelli 1999], and requires novel and specialised programming techniques. Wide-area networks are characterised by the existence of separate *locations*, offering different services and having different properties. In particular, locations are protected by *barriers* (e.g. administrative domains, firewalls, etc), which control access to the local resources. As different locations have different properties, programs need to move between them and thus cross those barriers. Mobility and barrier crossing seem inevitable requirements of wide-area computing infrastructure.

Cardelli and Gordon designed the *process calculus* of *Mobile Ambients* [Cardelli and Gordon 2000], abbreviated MA, as an abstract model of computation over wide-area networks, by focussing on the concepts of barriers and barrier crossing. An *ambient* process, denoted by $n[P]$, represents a place, named n, delimited by a boundary, which encloses the multi-threaded computation P. Ambients can be nested within other ambients, forming a tree structure reminiscent of the hier-

archical organisation of administrative domains. Mobility is then represented as navigation across a hierarchy of ambients: in the term $n[P]$, the computation P can exercise a *capability* to instruct the enclosing ambient n to move. There are three kinds of capabilities. The first, the $\texttt{in\_}m$ capability, causes the enclosing ambient to enter into a sibling ambient named m. An example of this activity is formally rendered as follows:

$$n[\texttt{in\_}m.P] \mid m[Q] \;\rightarrow\!\triangleright\; m[n[P] \mid Q]$$

where the construct "|" denotes *parallel composition* of processes, "." denotes *prefixing* (which here blocks the execution of P until the capability is consumed), and "$\rightarrow\!\triangleright$" denotes the dynamics of the terms using a *reduction relation* showing the state change. The second, $\texttt{out\_}m$, causes the enclosing ambient to exit from its parent ambient, if the parent is named m:

$$m[n[\texttt{out\_}m.P] \mid Q] \;\rightarrow\!\triangleright\; m[Q] \mid n[P] \;.$$

The third, $\texttt{open\_}m$, dissolves the boundaries of an ambient named m:

$$\texttt{open\_}m.P \mid m[Q] \;\rightarrow\!\triangleright\; P \mid Q \;.$$

Ambient names, such as n and m, play a central role in the computational model of MA, as they are used to control access to the ambient's interior: a process must refer to the ambient it wants to interact with by name. As in the π-calculus [Milner et al. 1992], the construct $(\boldsymbol{\nu} n)P$ dynamically creates a new name whose scope is initially limited to the process P.

Since their introduction in 1998, MA attracted strong interest from the concurrency theory community. In particular, the development of effective semantics theories for MA has been a long-standing open problem.

A central concern for process calculi is to establish when two processes have the same *observable behaviour*, that is, they are indistinguishable in any environment. *Behavioural equivalences* are fundamental for relating implementations to specifications, and for justifying program transformations performed either by programmers, during system development, or by the optimising phases of compilers. While several notions of behavioural equivalences can be found in the literature, they all share some key properties:

—two terms are equivalent only if they offer identical interactions to any environment, that is, they expose the same *observables*;

—the equivalence is preserved by some key constructs of the calculus: in this case, proving the equivalence of two large processes can be reduced to proving the equivalence of their components.

Other properties may vary, such as sensitivity to deadlock, the class of contexts that preserve the equivalence, and the definition of observable.

In this paper we focus on (weak) *reduction barbed congruence*, a behavioural equivalence defined as the largest equivalence that:

—is preserved by all the constructs of the language;

—is preserved (in a sense we will make precise later) by the reduction semantics of the language;

—preserves *barbs*, which are simple observables of terms.

By definition, reduction barbed congruence is both a branching-time equivalence that preserves the observables of the language, and a congruence; we will point out later the advantages of this formulation. Reduction barbed congruence was first studied by Honda and Yoshida [1995] under the name of *maximum sound theory*, and it is also known as *open barbed bisimilarity* [Sangiorgi and Walker 2001b].

The definition of reduction barbed equivalence is simple and intuitive. In practise, however, it is difficult to use: the quantification on all contexts is a heavy proof obligation. Simpler proof techniques are based on *labelled bisimilarities* [Park 1981; Milner 1989], which are co-inductive relations that characterise the behaviour of processes using a *labelled transition system* (abbreviated LTS). An LTS consists of a collection of relations of the form

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

The judgement above means that the process P can realise the *action* α, and becomes Q: intuitively, the action α represents some small context with which the process P can interact. The reduction semantics of a process is easily encoded in an LTS because a reduction step can be seen as an interaction with an empty context: this is traditionally called a τ-action. More generally, an LTS records the fact that a process can interact with a context that makes some specific resource available. In this case, the action codifies the minimal context needed to realise such interaction.

We can define an equivalence for processes from the LTS, by requiring that the observable actions of one process can be mimicked by the actions of the other. Labelled bisimilarity is the co-inductively closed form of this equivalence. If the LTS is sufficiently rich, the resulting bisimilarity will be contained in reduction barbed congruence, and therefore the former becomes a proof technique for the latter. In practise, this is useful because the quantification over all actions is easier to verify than the quantification over all contexts. We can further simplify the verification, by employing *up-to proof techniques*. These techniques allow us to abstract some details of the processes being tested, for example by stripping off a common context after which only the remaining parts need be compared. If the labelled bisimilarity coincides with reduction barbed congruence, then the set of actions captures exactly the observable interactions that processes can have with arbitrary contexts. In this case, the LTS captures the fundamental properties of interactions between terms and contexts, providing a deep understanding of equivalent processes.

Although the idea of labelled bisimilarity is very general and does not rely on the specific syntax of the calculus, the definition of an appropriate LTS and associated weak bisimilarity for Mobile Ambients turned out to be harder than expected. The reasons can be summarised as follows:

— *Ambient mobility is asynchronous*: no synchronisation is required to migrate into an ambient. As noticed by Sangiorgi [2001], this causes a *stuttering* phenomenon originated by ambients that may repeatedly enter and exit another am-

bient. As an example, the two processes

$$P \stackrel{\text{def}}{=} \texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.R$$
$$Q \stackrel{\text{def}}{=} \texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n.R + \texttt{in\_}n.R$$

where "+" denotes nondeterministic guarded choice[1] *à la* CCS [Milner 1989], cannot be distinguished by reduction barbed congruence. Process Q can obviously mimic the behaviour of P. Perhaps surprisingly, P can simulate Q too: for instance, P can mimic the reduction $k[Q] \mid n[\,] \rightarrow n[k[R]]$ by performing three consecutive reductions: $k[P] \mid n[\,] \rightarrow \rightarrow \rightarrow n[k[R]]$. If we think in terms of labelled transitions, we see that when P undergoes the action $\texttt{in\_}n$, then Q can match this exactly, with one $\texttt{in\_}n$ action; while if Q undergoes the right $\texttt{in\_}n$ action, P can only match it with a sequence of three actions $\texttt{in\_}n.\texttt{out\_}n.\texttt{in\_}n$. This is challenging to accommodate with bisimilarity. However, since stuttering cannot be observed by reduction barbed congruence, a complete labelled characterisation of reduction barbed congruence is obliged to be insensitive to stuttering as well.

— *The movement of private ambients cannot be observed*: consider the *perfect firewall equation* [Cardelli and Gordon 2000], a well-known algebraic law of MA:

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

This law states that a *private ambient* n whose internal code does not refer to the name of the ambient itself is equivalent to the inactive process. The idea is that a context cannot know the name of the private ambient, and, consequently, it cannot interact with it, while the condition "n does not appear in P" ensures that the computation P cannot move outside the ambient n. The subtle point is that the ambient n can freely move around the network without being observed. Again, as the law above is captured by reduction barbed congruence, a complete labelled characterisation of reduction barbed congruence must not observe the movements of private ambients.

Merro and Hennessy [2002; 2005] introduced a weak labelled bisimilarity for a simpler variant of MA, called SAP, equipped with (i) *synchronous mobility*, as in Levi and Sangiorgi's *Safe Ambients* [Levi and Sangiorgi 2000], and (ii) *passwords* to exercise control over, and differentiate between, different ambients that wish to exercise a capability. Synchronous mobility ensures that stuttering cannot happen and prevents private ambients from moving: the two difficulties highlighted above are ruled out by changing the syntax and the reduction semantics of MA. Their main result is a sound and complete characterisation of reduction barbed congruence in terms of a labelled bisimilarity. The result does not apply to MA because it relies crucially on features (i) and (ii) mentioned above.

This paper is the natural continuation of Merro and Hennessy investigations, where we tackle the original problem: *to provide bisimulation proof methods for Mobile Ambients.*

[1]The guarded choice construct is not part of the syntax of MA; however, a similar, but more complex, example can be exhibited using only the operators of MA.

Contribution. The aim of this work is to provide a labelled characterisation of weak *reduction barbed congruence*. This is achieved by a careful study of the behavioural theory of a class of processes, called *systems*. We outline the main contributions of this paper, highlighting how they fit together.

Section 1. First, we divide MA terms into two categories: *processes* and *systems*. Systems are the subclass of processes consisting of parallel compositions of ambients (which may share the knowledge of ambient names). A system thus exposes the units of mobility to an observer, but does not expose directly the threads of computation. As we will see, this allows us to derive a simple LTS, while retaining enough observational power to extend our characterisation easily to all processes.

Section 2. We define an LTS for systems, which captures precisely and concisely the mobility interactions that a system can perform with a context. For instance, the term $m[\mathtt{in}\_n.P]$ can reduce in a context which provides an ambient named n, for example $- \mid n[R]$, where "$-$" denotes the hole in the context. The reduction

$$m[\mathtt{in}\_n.P] \mid n[R] \twoheadrightarrow n[m[P] \mid R]$$

is then captured in the LTS by the transition

$$m[\mathtt{in}\_n.P] \xrightarrow{m.\mathtt{enter}\_n} n[m[P] \mid R] \ .$$

Similarly, the other labels capture the all possible interactions:

— a system $m[\mathtt{out}\_n.P]$ can reduce in the context $n[- \mid R]$, yielding respectively the reduction and the transition

$$n[m[\mathtt{out}\_n.P]] \twoheadrightarrow m[P] \mid n[R] \qquad \text{and} \qquad m[\mathtt{out}\_n.P] \xrightarrow{m.\mathtt{exit}\_n} m[P] \mid n[R] \ ;$$

— a system $m[P]$ can interact with a context that opens it, as $n[- \mid \mathtt{open}\_m.R]$, yielding

$$n[m[P] \mid \mathtt{open}\_m.R] \twoheadrightarrow n[P \mid R] \qquad \text{and} \qquad m[P] \xrightarrow{n.\mathtt{open}\_m} n[P \mid R] \ ;$$

— a system $m[P]$ can interact with a context that provides an ambient that enters into it, as $- \mid n[\mathtt{in}\_m.R]$, yielding

$$m[P] \mid n[\mathtt{in}\_m.R] \twoheadrightarrow m[P \mid n[R]] \qquad \text{and} \qquad m[P] \xrightarrow{m.\overline{\mathtt{enter}}\_n} m[P \mid n[R]] \ .$$

In all the examples above, the process R inside the ambient n is an arbitrary process provided by the context. The LTS leaves it unspecified, and it will be instantiated *later*, in the bisimulation.

Section 3. From the LTS, we define a weak labelled bisimilarity over systems. The bisimilarity relation ensures that equivalent systems can mimic their observable actions. While doing so, it is also responsible for specifying the arbitrary process provided by the context; in this respect, it resembles the formulation of Sangiorgi's *context bisimulation* for HOπ [Sangiorgi 1996a]. However, as highlighted above, the contexts used in the co-inductive step are very simple, unlike in Sangiorgi's context bisimulation. Depending on the position of the quantification of processes in the definition of bisimulation, we can define both *late* and *early* bisimilarity [Sangiorgi and Walker 2001a]. As in HOπ [Sangiorgi 1996a], we show that the two formulations coincide; thereafter we concentrate on the late version, $\approx$, which is

easier to manipulate. The definition of our labelled bisimilarity is similar to the asynchronous bisimilarity of Amadio, Castellani and Sangiorgi for asynchronous π-calculus [Amadio et al. 1998]. More precisely, our bisimilarity does not observe the movements of secret ambients, in the same way as asynchronous bisimilarity does not observe input actions. We prove that the relation $\approx$ completely characterises *reduction barbed congruence over systems*, $\cong_{\mathsf{s}}$, that is, for all systems M and N it holds that

$$M \approx N \text{ iff } M \cong_{\mathsf{s}} N \ .$$

Section 4. We provide two *up-to proof techniques*, along the lines of [Milner and Sangiorgi 1992; Sangiorgi 1998; Sangiorgi and Walker 2001a]. More precisely, we develop both *up-to expansion* and *up-to context* proof techniques for $\approx$, and prove their soundness. These techniques are useful to reduce the size of the candidate bisimulation and turn the labelled bisimilarity into a very effective proof method. In particular, the up-to context proof technique is fundamental for factoring out the universally quantified processes provided by the environment. As far as we know, this is the first application of up-to proof techniques to higher-order process languages.

Section 5. We then use the theory developed for systems to characterise *reduction barbed congruence over processes*, $\cong_{\mathsf{p}}$, in terms of $\approx$. More precisely, we show that:

$$\cong_{\mathsf{p}} = \ \{(P,Q) : k[P \mid R] \approx k[Q \mid R] \text{ for all } k,\, R\}$$

where P and Q range over processes. This result relies crucially on a *context lemma* for $\cong_{\mathsf{p}}$, which allows us to consider only contexts for concurrency and locality.

When restricting our attention to systems, a stronger results holds: for all systems M and N we have that

$$M \approx N \text{ iff } M \cong_{\mathsf{p}} N \ .$$

Section 6. We extend our results to the full calculus of Mobile Ambients processes equipped with asynchronous communication of capabilities. A consequence of building our proof methods on top of the behaviour of systems rather than processes is that communication cannot be observed directly, and thus few modifications are required to accommodate it.

Section 7. We apply our bisimulation proof methods to checking a collection of *algebraic laws* (including the *perfect firewall equation*) with respect to $\cong_{\mathsf{p}}$. The proofs are pleasantly simple: the size of the candidate bisimulations is small thanks to the up-to context proof technique. We also prove the correctness of a protocol, introduced in [Cardelli and Gordon 2000], for controlling access through a firewall.

The paper ends with a comparison with related work.

1. MOBILE AMBIENTS IN TWO LEVELS

In Table I we report the syntax of MA, where $\mathbf{N}$ denotes a countable infinite set of names.

Unlike the original definitions of MA, our syntax is defined in a two-level structure, a lower one for *processes*, and an upper one for *systems*. Systems are collections of ambients running in parallel, that may share knowledge of ambient names. As

Table I. Mobile Ambients in Two Levels

Names:	$a, b, \ldots, k, l, m, n, \ldots \in \mathbf{N}$	
Systems:		
M, N ::=	$\mathbf{0}$	inactive system
	$M_1 \mid M_2$	parallel composition
	$(\boldsymbol{\nu} n)M$	restriction
	$n[P]$	ambient
Processes:		
P, Q, R ::=	$\mathbf{0}$	inactive process
	$P_1 \mid P_2$	parallel composition
	$(\boldsymbol{\nu} n)P$	restriction
	$C.P$	prefixing
	$n[P]$	ambient
	$!C.P$	replication
Capabilities:		
C ::=	$\mathtt{in\_}n$	may enter into n
	$\mathtt{out\_}n$	may exit out of n
	$\mathtt{open\_}n$	may open n

regards processes, the constructs for inactivity, parallel composition, restriction and replicated prefixing are inherited from mainstream concurrent calculi, most notably the π-calculus [Milner et al. 1992]. The inactive process, $\mathbf{0}$, does nothing. Parallel composition is denoted by the commutative and associative operator $|$. The restriction operator, $(\boldsymbol{\nu} n)P$, creates a new fresh name n within a scope P. We have replicated prefixing, $!C.P$, (rather than full replication) to create as many parallel replicas of a guarded process as needed. Since the copies of the guarded process cannot interact among themselves, working with replicated prefixing simplifies the definition of the LTS and most of the proofs.

The specific features of the ambient calculus are the *ambient* construct, $n[P]$, and the *prefixing* of capabilities, $C.P$. In $n[P]$, n is the name of the ambient and P is the process running inside the ambient. The process $C.P$ performs an action regulated by the capability C, and then continues as the process P. Capabilities are constructed from names; given a name n, the capability $\mathtt{in\_}n$ allows entering into n, the capability $\mathtt{out\_}n$ allows exiting out of n, and the capability $\mathtt{open\_}n$ allows destructing the boundary of the ambient n. To avoid unnecessary complications at this stage, we omit *communication*; it will be added in Section 6.

A (monadic) *context* $\mathcal{C}[-]$ is a process with a hole, denoted by $-$. A *static context* is a context where the hole does not appear under a prefix or a replication.

The class of systems is not closed under arbitrary contexts: as an example, the context $\mathcal{C}[-] = - \mid \mathtt{open\_}n$ sends a system M into a process $M \mid \mathtt{open\_}n$. We restrict our attention to the class of contexts, called *system contexts*, that sends a system to a system and that retains the distinguishing power of arbitrary contexts (as shown in Section 5). Formally, system contexts are those static contexts that

transform systems into systems. They are generated by the grammar below:

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

where M is an arbitrary system, and P is an arbitrary process. The contexts $n[\mathcal{C}[-] \mid P]$ and $n[P \mid \mathcal{C}[-]]$ allow testing a term by running it in parallel with a process: they are key elements to retain the distinguishing power of arbitrary contexts. We always specify if we mean an arbitrary context or a system context when we write $\mathcal{C}[-]$.

We use a number of notational conventions. Parallel composition has the lowest precedence among the operators. $\prod_{i\in I} P_i$ means the parallel composition of all processes P_i, for $i \in I$. $\tilde{n}$ denotes a tuple $n_1, \ldots, n_k$ of names. The process $C.C'.P$ is read as $C.(C'.P)$. We omit trailing dead processes, writing C for $C.\mathbf{0}$, and $n[\,]$ for $n[\mathbf{0}]$. Occasionally, we omit inactive processes when they are in parallel with processes, writing P for $P \mid \mathbf{0}$. The operator $(\boldsymbol{\nu} n)$ is a binder for names, leading to the usual notions of free and bound occurrences of names, $\mathrm{fn}(\cdot)$ and $\mathrm{bn}(\cdot)$, and α-conversion, $\equiv_\alpha$. We write $(\boldsymbol{\nu}\tilde{n})P$ as an abbreviation for $(\boldsymbol{\nu} n_1)\ldots(\boldsymbol{\nu} n_k)P$. We will identify processes up to α-conversion. More formally we will view process terms as representatives of their equivalence class with respect to $\equiv_\alpha$, and these representatives will always be chosen so that bound names are distinct from free names. Unless otherwise stated, contexts are monadic.

Operational semantics. The dynamics of the calculus is specified by the *reduction relation* over processes, $\rightarrowtriangle$, described in Table II. As systems are processes with a special structure, the rules of Table II also describe the evolution of systems. The *reduction semantics* relies on an auxiliary relation called *structural congruence* that brings the participants of a potential interaction into contiguous positions. It is easy to check that the class of systems is closed under the reduction relation, that is, systems always reduce to systems. The symbol $\rightarrowtriangle^*$ denotes the reflexive and transitive closure of $\rightarrowtriangle$.

Behavioural semantics. We now introduce our reference equivalence, reduction barbed congruence.

***Definition* 1.1** A relation $\mathcal{R}$ over processes is *reduction closed* if $P\ \mathcal{R}\ Q$ and $P \rightarrowtriangle P'$ imply the existence of some Q' such that $Q \rightarrowtriangle^* Q'$ and $P'\ \mathcal{R}\ Q'$.

***Definition* 1.2** A relation $\mathcal{R}$ over processes is *preserved by contexts (*resp. *system contexts)* if $P\ \mathcal{R}\ Q$ implies $\mathcal{C}[P]\ \mathcal{R}\ \mathcal{C}[Q]$ for all contexts (resp. system contexts) $\mathcal{C}[-]$.

In MA, given a process P, a simple observable is the presence at top-level of an ambient whose name (say n) is not restricted: the observation predicate $P \downarrow n$ captures exactly this observable. Formally, we write $P \downarrow n$ if $P \equiv (\boldsymbol{\nu}\tilde{m})(n[P_1] \mid P_2)$ where $n \notin \{\tilde{m}\}$. We write $P \Downarrow n$ if there exists P' such that $P \rightarrowtriangle^* P'$ and $P' \downarrow n$.

***Definition* 1.3** We say that a relation $\mathcal{R}$ over processes is *barb preserving* if $P\ \mathcal{R}\ Q$ and $P \downarrow n$ implies $Q \Downarrow n$.

We are ready to define the contextual equivalences of interest:

Definition* 1.4 (*Reduction barbed congruence)

Table II. Structural Congruence and Reduction Rules

$P \mid Q \equiv Q \mid P$	(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 satisfies the axioms and rules above, and is preserved by contexts.

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

$\rightarrow$ is the least relation which satisfies the rules above and is preserved by static contexts.

—*Reduction barbed congruence over systems*, written $\cong_{\mathsf{s}}$, is the largest symmetric relation over systems which is reduction closed, barb preserving, and preserved by system contexts.

—*Reduction barbed congruence over processes*, written $\cong_{\mathsf{p}}$, is the largest symmetric relation over processes which is reduction closed, barb preserving, and preserved by all contexts.

When comparing two processes, reduction barbed congruence allows the context surrounding the processes being compared to be changed at any point in the bisimulation game. An alternative contextual equivalence, called *barbed congruence* [Milner and Sangiorgi 1992], is defined as the context closure of the largest symmetric relation which is reduction closed and barb preserving. Since barbed congruence fixes the observer once for all at the beginning of the bisimulation game, it might be argued that it is a more natural equivalence. However, we choose reduction barbed congruence as our main equivalence because the the power to change the context surrounding the systems being tested makes possible the proof of characterisation theorem of Section 3.

In the remainder of the paper, when working with a relation $\mathcal{R}$ over processes and/or systems, we write $\mathcal{R}^{=}$ to denote the symmetric closure of $\mathcal{R}$.

2. A LABELLED TRANSITION SEMANTICS FOR SYSTEMS

Along standard lines, [Milner 1989], prefixes C give rise to transitions of the form $P \xrightarrow{C} Q$. For example we have

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

However, similarly to what happens in [Merro and Hennessy 2002] and in [Merro and Hennessy 2005] each of the capability C induces different and more complicated actions. The LTS is defined over processes, although in the labelled bisimilarity we only consider actions going from systems to systems. We make a distinction

Table III. Pre-actions, Env-actions, Actions, Concretions, and Outcomes

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

Table IV. Labelled Transition System - Pre-actions

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

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

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

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

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 formal definition of the LTS is given below, and is followed by several examples illustrating its use (starting from page 12).

The *pre-actions*, defined in Table IV, are of the form $P \xrightarrow{\pi} O$ where the ranges of π and of O, the *outcomes*, are reported in Table III. An outcome may be a simple process Q, if for example π is a capability, or a *concretion*, of the form $(\boldsymbol{\nu}\tilde{m})\langle P\rangle Q$, when an ambient boundary is somehow involved. In this case, P represents the code that may enter to, reside at, or exit from an ambient; Q represents the derivative which is not affected by the action, 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 $r \notin \text{fn}(P)$, 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 $\text{fn}(R) \cap \{\tilde{m}\} = \emptyset$; similarly $R \mid K$ is the concretion

Table V. Labelled Transition System - τ-actions

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

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

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

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

(*) In rule (τ Enter) we require

$$((\text{fn}(k[P_1]) \cup \text{fn}(P_2)) \cap \{\tilde{q}\}) = ((\text{fn}(Q_1) \cup \text{fn}(Q_2)) \cap \{\tilde{p}\}) = \emptyset \ .$$

$(\boldsymbol{\nu}\tilde{m})\langle P\rangle(R \mid Q)$.

The rules (π Pfx), (π Repl Pfx), (π Res), and (π Par) are standard. The rule (π Enter) results in a concretion containing the ambient willing to enter n. The rule (π Exit) is similar, but the resulting concretion contains the ambient willing to exit from n. The rule (π Amb) records in a concretion the code residing at n.

The *τ-actions*, formally defined in Table V, model the internal evolution of processes. The rule (τ Enter) models an ambient migrating into a sibling ambient n. The rule (τ Exit) models an ambient k exiting from an ambient n. The rule (τ Open) describes the opening of an ambient n. Structural rules (τ Amb), (τ Res), and (τ Par) are straightforward.

The *env-actions*, formally defined in Table VI, are of the form $M \xrightarrow{\sigma} M'$, where the range of σ is given in Table III. Env-actions turn concretions into running systems by explicitly introducing the environment's ambient interacting with the process in question. The content of this ambient will be instantiated later, in the definition of the bisimilarity, with a process. A special process variable, denoted $\circ$, (also called *placeholder*) is used to pinpoint those ambients whose content will be instantiated later.

***Definition* 2.1 (*Extended syntax*)** We call *extended syntax* the grammar of Table 1 extended with the production $P ::= \ ... \mid \circ$.

We will specify if P represents a process or a process over the extended syntax whenever it is not clear from the context.

The LTS is defined over processes over the extended syntax. However, all the equivalences defined in this paper relate only processes that do not contain the special process variable $\circ$.

Note that, unlike pre-actions and τ-actions, env-actions do not have structural

Table VI. Labelled Transition System - Env-actions

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

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

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

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

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

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

rules; this is because env-actions are supposed to be performed by systems that can directly interact with the environment. In the rules (Enter) and (Exit) an ambient k enters, respectively exit from, an ambient n provided by the environment. The rules (Enter Shh) and (Exit Shh) are similar and model the migration of private ambients. In the rule (Co-Enter) an ambient k, provided by the environment, migrates into an ambient n of the process. In the rule (Open) the environment opens an ambient n of the process; the opening is performed inside an ambient k provided by the environment.

We call *actions* the set of env-actions extended with τ. Actions, denoted by α, always go from systems over the extended syntax to systems over the extended syntax. As our bisimilarity will be defined over systems, we will only consider actions (and not pre-actions) in its definition.

Proposition 2.2 *If T is a system (resp. a process) over the extended syntax, and $T \xrightarrow{\alpha} T'$, then T' is a system (resp. a process) over the extended syntax.*

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

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

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

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

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

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

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

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 where the execution started. Of course the transition fires only 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 the process Q is inside n and is affected by the action, while the process $\mathbf{0}$ is outside and is not affected. Finally, the rule (τ Enter) allows these two complementary actions to occur simultaneously, executing 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:

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

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

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

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

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

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

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

Now, let us explain the rules for *emigration* with an example. A typical example of an ambient m emigrating from an ambient n follows:

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

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

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

Here, when exercising this capability, the code Q remains inside the ambient n while the ambient $m[P_1 \mid P_2]$ moves outside. However, to complete 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:

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

As for immigration, env-actions $m.\texttt{exit\_}n$ and $*.\texttt{exit\_}n$ model the exiting of global and private ambients from an ambient n provided by the environment.

Whenever a system offers a public ambient n at top-level, a context can interact with the system by providing an ambient willing to enter inside n. For instance, the system $n[P] \mid M$ can interact with the system context $\mathcal{C}[-] = - \mid k[\texttt{in\_}n.R]$ (where R is an arbitrary process), yielding the system $n[k[R] \mid P] \mid M$. The rule (Co-Enter) captures this interaction between system and environment. In fact, it holds that

$$n[P] \mid M \xrightarrow{n.\overline{\texttt{enter}}\_k} n[k[\,\circ\,] \mid P] \mid M \ .$$

A system that offers a public ambient n at top-level can also interact with a system context willing to open it, like $\mathcal{C}[-] = k[\texttt{open\_}n.R \mid -]$. The rule (Open) captures this interaction. For instance, we have

$$n[P] \mid M \xrightarrow{k.\texttt{open\_}n} k[\circ \mid P \mid M] \ .$$

We end this section with several technical lemmas, and a theorem that asserts that the LTS-based semantics coincides with the reduction semantics of Section 1.

For any process P, outcome O and pre-action π such that $P \xrightarrow{\pi} O$, the structure of P and O can be determined up to structural congruence.

Lemma 2.3

—*If* $P \xrightarrow{C} O$, *with* $C \in \{\texttt{in\_}n, \texttt{out\_}n, \texttt{open\_}n\}$, *then there exist* $\tilde{p}, P_1, P_2$, *with* $n \notin \tilde{p}$, *such that*

$$P \equiv (\boldsymbol{\nu}\tilde{p})(C.P_1 \mid P_2) \quad \textit{and} \quad O \equiv (\boldsymbol{\nu}\tilde{p})(P_1 \mid P_2) \ .$$

—*If* $P \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{p})\langle P'\rangle P''$ *then there exist* k, P_1, P_2, *with* $n \notin \tilde{p}$, *such that*

$$P \equiv (\boldsymbol{\nu}\tilde{p})(k[\texttt{in\_}n.P_1 \mid P_2] \mid P'') \quad \textit{and} \quad P' \equiv k[P_1 \mid P_2] \ .$$

—*If* $P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{p})\langle P'\rangle P''$ *then there exist* k, P_1, P_2, *with* $n \notin \tilde{p}$, *such that*

$$P \equiv (\boldsymbol{\nu}\tilde{p})(k[\texttt{out\_}n.P_1 \mid P_2] \mid P'') \quad \textit{and} \quad P' \equiv k[P_1 \mid P_2] \ .$$

—*If* $P \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{p})\langle P'\rangle P''$, *with* $n \notin \tilde{p}$, *then* $P \equiv (\boldsymbol{\nu}\tilde{p})(n[P'] \mid P'')$.

Proof By induction on the transition rules of Tables IV and V. □

Transitions are preserved by structural congruence:

Lemma 2.4 *If $P \equiv Q$ and $P \xrightarrow{\ell} P'$ for $\ell \in \sigma \cup \{\tau\}$, then there is Q' such that $Q \xrightarrow{\ell} Q'$ and $P' \equiv Q'$.*

The correspondence between reductions and τ-transitions is stated in the theorem below:

Theorem 2.5

(1) If $P \xrightarrow{\tau} P'$ then $P \rightarrowtriangle P'$

(2) If $P \rightarrowtriangle P'$ then $P \xrightarrow{\tau} \equiv P'$.

The proof of these two results is standard, and is postponed to Appendix A.

3. CHARACTERISATION OF REDUCTION BARBED CONGRUENCE OVER SYSTEMS

In this section we define a labelled bisimilarity that completely characterises reduction barbed congruence over systems.

In the previous section we said that env-actions introduce a special process variable $\circ$ to pinpoint those ambients whose content must be instantiated in the bisimilarity. We write $P \bullet R$ to denote the name-capture avoiding substitution of the process R for the occurrences of $\circ$ in P.

***Definition* 3.1** Let P, Q be processes over the extended syntax. Let R be a process. We define:

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

It should be pointed out that in what follows, whenever we write $P \bullet R$, there is only one occurrence of $\circ$ in P. In some proofs, we use an extended definition of $\bullet$ allowing R to range over processes involving $\circ$.

Everything is now 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 $M \xrightarrow{\alpha} M'$, $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there is a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and for all processes P it holds that $M' \bullet P \mathcal{R} N' \bullet P$;

—if $M \xrightarrow{*.\texttt{enter}\_n} M'$ then there is a system N' such that $N \mid n[\,\circ\,] \Rightarrow N'$ and for all processes P it holds that $M' \bullet P \mathcal{R} N' \bullet P$;

—if $M \xrightarrow{*.\texttt{exit}\_n} M'$ then there is a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all processes P it holds that $M' \bullet P \mathcal{R} N' \bullet P$.

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

The first clause applies to both observable and silent transitions. It ensures that whenever the system being tested realises an observable action, then the matching system realises the same observable action, possibly preceded and/or followed by internal transitions. It also ensures that the outcomes are equivalent. In the case of silent transitions (i.e., when $\alpha = \tau$), the outcome M' does not contain the special process variable $\circ$, as there is no interaction with the environment. As a consequence, for $\alpha = \tau$, we could simply write

—if $M \xrightarrow{\tau} M'$ then there is a system N' such that $N \Rightarrow N'$ and $M' \mathcal{R} N'$.

When α is an env-action, there is a universal quantification over the process P (provided by the environment) which replaces the placeholder $\circ$ generated by the env-action.

The second and third clauses define the matching requirements when a system interacts with a context by the movement of a secret ambient. The bisimulation is not defined in the standard way, that is, as a symmetric relation $\mathcal{R}$ over systems such that whenever $M \mathcal{R} N$ and $M \xrightarrow{\alpha} M'$, there is a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$, and for all processes P it holds that $M' \bullet P \mathcal{R} N' \bullet P$. A standard bisimilarity would yield a sound proof technique, but would not be a complete characterisation of $\cong_{\mathsf{s}}$. In fact, the two systems

$$(\boldsymbol{\nu} n) n[\mathtt{in}\_k.\mathbf{0}] \quad \text{and} \quad \mathbf{0}$$

are reduction barbed congruent, but are distinguished by the standard bisimilarity. In particular, the system $(\boldsymbol{\nu} n) n[\mathtt{in}\_k.\mathbf{0}]$ can perform a $*.\mathtt{enter}\_k$ action while $\mathbf{0}$ cannot. This example shows that a labelled characterisation of reduction barbed congruence should treat actions $*.\mathtt{enter}\_n$ and $*.\mathtt{exit}\_n$ separately, asking for weaker matching requirements: like input actions in the asynchronous π-calculus [Honda and Tokoro 1991; Boudol 1992], these actions cannot be observed by a context.

According to the π-calculus terminology [Sangiorgi and Walker 2001a], 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 $\approx_e$ to denote the 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 [Sangiorgi 1996a], we will prove that late and early bisimilarity coincide. We choose late bisimilarity as our main labelled bisimilarity because the derivatives N' do not depend on the environment's contribution P. The π-calculus experience suggests that late bisimulations may fail to be transitive. However, processes reveals to be more 'tractable' than names, and in our framework late bisimilarity turns out to be an equivalence relation.

3.1 Soundness

We show that late and early bisimilarity are two proof techniques for reduction barbed congruence over systems. More precisely, we prove that they are both contained in reduction barbed congruence over systems.

Theorem 3.3 *Late bisimilarity is preserved by system contexts.*

Table VII. System Contexts for Visible Actions

$$\mathcal{C}_{k.\mathtt{enter}\_n}[-] \stackrel{\text{def}}{=} n[\mathsf{done}[\mathtt{in}\_k.\mathtt{out}\_k.\mathtt{out}\_n] \mid \circ] \mid -$$

$$\mathcal{C}_{k.\mathtt{exit}\_n}[-] \stackrel{\text{def}}{=} (\boldsymbol{\nu} a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid n[\circ \mid -]$$

$$\mathcal{C}_{n.\overline{\mathtt{enter}}\_k}[-] \stackrel{\text{def}}{=} (\boldsymbol{\nu} a)a[\mathtt{in}\_n.k[\mathtt{out}\_a.(\circ \mid (\boldsymbol{\nu} b)b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]])]] \mid -$$

$$\mathcal{C}_{k.\mathtt{open}\_n}[-] \stackrel{\text{def}}{=} k[\circ \mid (\boldsymbol{\nu} a, b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathsf{done}[\mathtt{out}\_k] \mid a[- \mid \mathtt{open}\_n.b[\mathtt{out}\_a]])]$$

where a,b and done are fresh names.

Proof We show that the closure of $\approx$ under system contexts is a bisimulation. This requires a long induction on the structure of the system contexts, reported in Appendix B. The result then follows by co-induction. □

In general, proving the congruence of a bisimilarity that involves higher-order terms is difficult. In MA, however, the mobility model is not based on process variables and substitutions, and the bisimilarity only relates closed processes. This avoids several difficulties that occur, for instance, in the proof of congruence of applicative bisimilarity for the λ-calculus [Howe 1996], and in Sangiorgi's proof of congruence of contextual bisimilarity for HOπ [Sangiorgi 1996a].

It is easy to adapt the proof of the theorem above to show that also early bisimilarity is preserved by system contexts.

Proposition 3.4 *Early bisimilarity is preserved by system contexts.*

In the following lemma we point out a close relationship between the observation predicate $M \downarrow n$ and a specific action that M can emit.

Lemma 3.5

(1) *If* $M \xrightarrow{n.\overline{\mathtt{enter}}\_k} M'$ *then* $M \downarrow n$*;*

(2) *if* $M \downarrow n$ *then there exists a system* M' *such that* $M \xrightarrow{n.\overline{\mathtt{enter}}\_k} M'$ *for some name* k*.*

It is thus easy to prove that both late and early bisimilarity imply reduction barbed congruence over systems.

Theorem 3.6 (Soundness) *The following chain of inclusions holds:* $\approx \subseteq \approx_e \subseteq \cong_{\mathsf{s}}$.

Proof The first inclusion holds by definition. The second one comes from the fact that early bisimilarity is reduction closed (immediate consequence of Theorem 2.5), barb-preserving (by Lemma 3.5), and preserved by system contexts (by Proposition 3.4). □

3.2 Completeness

We now prove that late and early bisimilarity are more than proof techniques. They actually characterise reduction barbed congruence over systems. The main challenge here is to design the system contexts capable of observing our visible actions.

The definition of these contexts, denoted $\mathcal{C}_\alpha[-]$, where α ranges over visible actions, is given in Table VII. Each context uses the ambient done as a *fresh* barb to signal that the action α has occurred. We now elucidate the intuitions

behind these contexts. The context for $k.\texttt{enter\_}n$ offers an ambient n, containing the ambient done. This interior ambient can consume the initial $\texttt{in\_}k$ capability and then migrate to top-level if and only if an ambient k enters into n, i.e. if the system being tested realises the action $k.\texttt{enter\_}n$. The context for $k.\texttt{exit\_}n$ uses a private ambient a, different from done, to test if an ambient k exits from n (that is, if the system being tested realises the action $k.\texttt{exit\_}n$). As a result, the barb done can be observed at top-level only if the exit action has been detected. The context for $n.\overline{\texttt{enter}}\texttt{\_}k$ is more complicated. Instead of moving the ambient k directly into n, it encapsulates k inside a private ambient a. This ensures that a Trojan horse hidden in the system being tested cannot use the ambient k to enter into the ambient n (thus failing to capture the desired behaviour of the test and contradicting Lemma 3.12). The barb done is then released only after that the ambient k goes inside the ambient n. The use of the private ambient b is optional, but allows a uniform formulation of Lemma 3.12 by ensuring that if the ambient done arrives at top-level, then it is empty. Finally, in the context for $k.\texttt{open\_}n$, the private ambients a and b guarantee that the barb done is unleashed only when an ambient n is opened.

To prove our characterisation result we will show that reduction barbed congruence over systems is contained in the late bisimilarity. Then, by Theorem 3.6, we can prove that late bisimilarity, early bisimilarity, and reduction barbed congruence over systems, they all coincide. To prove that reduction barbed congruence over systems implies late bisimilarity we must spell out the correspondence between visible actions α and their corresponding system contexts $\mathcal{C}_\alpha[-]$.

We begin with a simple result that allows to garbage collect empty ambients whose name is secret.

Lemma 3.7 $(\boldsymbol{\nu} n)n[\,] \cong_{\mathsf{s}} \mathbf{0}$.

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

Lemma 3.8 *Let* $\alpha \in \{k.\texttt{enter\_}n, k.\texttt{exit\_}n, n.\overline{\texttt{enter}}\texttt{\_}k, k.\texttt{open\_}n\}$ *and let* M *be a system. For all processes* P*, if* $M \xrightarrow{\alpha} M'$ *then* $\mathcal{C}_\alpha[M] \bullet P \Rightarrow\cong_{\mathsf{s}} (M' \bullet P) \mid \mathsf{done}[\,]$.

Proof The proof is by case analysis on α. We detail here the case $\alpha = k.\texttt{enter\_}n$, and we report all the other cases in Appendix B.

Case $\alpha = k.\texttt{enter\_}n$**.** 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

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

Now,

$$\begin{aligned}
& \mathcal{C}_{k.\texttt{enter_}n}[M] \bullet P \\
\equiv\ & (\boldsymbol{\nu}\tilde{m})(n[\mathsf{done}[\texttt{in_}k.\texttt{out_}k.\texttt{out_}n] \mid P] \mid k[\texttt{in_}n.M_1 \mid M_2] \mid M_3) \\
\xrightarrow{\tau}\ & (\boldsymbol{\nu}\tilde{m})(n[\mathsf{done}[\texttt{in_}k.\texttt{out_}k.\texttt{out_}n] \mid P \mid k[M_1 \mid M_2]] \mid M_3) \\
\xrightarrow{\tau}\ & (\boldsymbol{\nu}\tilde{m})(n[P \mid k[M_1 \mid M_2 \mid \mathsf{done}[\texttt{out_}k.\texttt{out_}n]]] \mid M_3)
\end{aligned}$$

Table VIII. Spy Contexts

$\mathsf{spy}_\alpha\langle i,j,-\rangle$	$\stackrel{\text{def}}{=}$	$(i[\mathtt{out}\_n] \mid -) \oplus (j[\mathtt{out}\_n] \mid -)$ if $\alpha \in \{k.\mathtt{enter}\_n, k.\mathtt{exit}\_n, k.\mathtt{open}\_n, *.\mathtt{enter}\_n, *.\mathtt{exit}\_n\}$
$\mathsf{spy}_\alpha\langle i,j,-\rangle$	$\stackrel{\text{def}}{=}$	$(i[\mathtt{out}\_k.\mathtt{out}\_n] \mid -) \oplus (j[\mathtt{out}\_k.\mathtt{out}\_n] \mid -)$ if $\alpha \in \{n.\overline{\mathtt{enter}}\_k\}$

$$
\begin{aligned}
&\xrightarrow{\tau} (\boldsymbol{\nu}\tilde{m})(n[P \mid \mathsf{done}[\mathtt{out}\_n] \mid k[M_1 \mid M_2]] \mid M_3)\\
&\xrightarrow{\tau} (\boldsymbol{\nu}\tilde{m})(\mathsf{done}[\,] \mid n[P \mid k[M_1 \mid M_2]] \mid M_3)\\
&\equiv\ (\boldsymbol{\nu}\tilde{m})(n[\circ \mid k[M_1 \mid M_2] \mid M_3]) \bullet P \mid \mathsf{done}[\,]\\
&=\ M' \bullet P \mid \mathsf{done}[\,]
\end{aligned}
$$

By Lemma 2.4 and transitivity of $\equiv$, there exists a system O such that $\mathcal{C}_{k.\mathtt{enter}\_n}[M] \bullet P \Rightarrow O$, and $O \equiv M' \bullet P \mid \mathsf{done}[\,]$. The result follows because structural congruence restricted to systems is contained in reduction barbed congruence over systems, and $O \cong_{\mathsf{s}} M' \bullet P \mid \mathsf{done}[\,]$.

The remaining cases are detailed in Appendix B. □

To complete the correspondence proof between actions α and their contexts $\mathcal{C}_\alpha[-]$, we have to prove the converse of Lemma 3.8, formalised in Lemma 3.12. The proof of this result uses some special contexts $\mathsf{spy}_\alpha\langle i,j,-\rangle$, defined in Table VIII, as 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 on both sides. Formally, the $\mathsf{spy}_\alpha\langle i,j,-\rangle$ contexts are multi-hole contexts, as the same hole occurs more than once (in this case, twice). The $\mathsf{spy}_\alpha\langle i,j,-\rangle$ contexts use *internal choice* encoded as:

$$P \oplus Q \stackrel{\text{def}}{=} (\boldsymbol{\nu}o)(o[\,] \mid \mathtt{open}\_o.P \mid \mathtt{open}\_o.Q)\ .$$

This encoding satisfies the following properties:

Lemma 3.9 *$P \oplus Q \xrightarrow{\tau}\cong_{\mathsf{s}} P$ and $P \oplus Q \xrightarrow{\tau}\cong_{\mathsf{s}} Q$.*

The ability of $\mathsf{spy}_\alpha\langle i,j,P\rangle$ to 'spy' on P stems from the fact that one of the two fresh barbs i and j is lost when P performs any action. The key properties of $\mathsf{spy}_\alpha\langle i,j,-\rangle$ are captured by the lemma below, proved in Appendix B.

Lemma 3.10

(1) *Let M be a system over the extended syntax. If $M \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \xrightarrow{\tau} O$ and $O \Downarrow i,j$, where i,j are fresh for P and M, then there exists a system M' over the the extended syntax such that:*
   (a) *$O = M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$;*
   (b) *$M \xrightarrow{\tau} M'$.*

(2) *For all ambients n and processes R, if $\{i,j\} \cap \mathrm{fn}(P) = \emptyset$, then*

$$n[(\boldsymbol{\nu}i,j)\mathsf{spy}_\alpha\langle i,j,P\rangle \mid R] \cong_{\mathsf{s}} n[P \mid R]\ .$$

The second statement illustrates that when the barbs i and j cannot be observed, $\mathsf{spy}_\alpha\langle i,j,-\rangle$ contexts can be garbage collected. We also need a simple result on arbitrary contexts (proved in Appendix B), reminiscent of the perfect firewall mentioned in the introduction.

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

We can finally prove the correspondence between actions and contexts.

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

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

$$\mathcal{C}_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \Rightarrow\equiv N \mid \mathsf{done}[\,]$$

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 $\mathsf{spy}_\alpha\langle i,j,P\rangle$ is found intact after the reduction. We proceed by case analysis on α. We detail here the case $\alpha = n.\overline{\mathtt{enter}}\_k$, and we report all the other cases in Appendix B.

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

$$\begin{array}{l}\mathcal{C}_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \equiv \\ \quad (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) a[\mathtt{in}\_n.k[\mathtt{out}\_a.(\mathsf{spy}_\alpha\langle i,j,P\rangle \mid b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]])]] \mid M\ .\end{array}$$

To unleash the ambient done, the ambient a must use its $\mathtt{in}\_n$ capability, and the ambient k must use its $\mathtt{out}\_a$ capability. Moreover, the ambient b must exit from k and n, and the ambient done must exit from b. More precisely, there must exist a system M_1 and system contexts $\mathcal{D}[-]$, $\mathcal{D}'[-]$, and $\mathcal{D}''[-_1,-_2,-_3]$ (for convenience we use a ternary context) such that

$$\begin{array}{rl}
& \mathcal{C}_{n.\overline{\mathtt{enter}}_k}[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \\
\equiv & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) a[\mathtt{in}_n.k[\mathtt{out}_a.(\mathsf{spy}_\alpha\langle i,j,P\rangle \mid b[\mathtt{out}_k.\mathtt{out}_n.\mathsf{done}[\mathtt{out}_b]])]] \mid M \\
\Rightarrow & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) a[\mathtt{in}_n.k[\mathtt{out}_a.(\mathsf{spy}_\alpha\langle i,j,P\rangle \mid b[\mathtt{out}_k.\mathtt{out}_n.\mathsf{done}[\mathtt{out}_b]])]] \mid M_1 \\
\xrightarrow{\tau} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) \mathcal{D}[a[k[\mathtt{out}_a.(\mathsf{spy}_\alpha\langle i,j,P\rangle \mid b[\mathtt{out}_k.\mathtt{out}_n.\mathsf{done}[\mathtt{out}_b]])]] \\
\stackrel{\tau}{\Longrightarrow} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) \mathcal{D}'[k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid b[\mathtt{out}_k.\mathtt{out}_n.\mathsf{done}[\mathtt{out}_b]]] \mid a[\,]] \quad (\star) \\
\Rightarrow & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b) \mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid a[\,], \mathsf{done}[\,], b[\,]] \quad (\star\star) \\
\equiv & N \mid \mathsf{done}[\,]
\end{array}$$

We know that the ambient done must end up at top level (up to $\equiv$). This implies that we first consume the capability $\mathtt{out}\_a$ (in the reduction sequence $(\star)$) and then the capabilities $\mathtt{out}\_k$, $\mathtt{out}\_n$, and $\mathtt{out}\_b$ (in the reduction sequence $(\star\star)$). Moreover, as $N \Downarrow i, j$, by Lemma 3.10, the process $\mathsf{spy}_\alpha\langle i,j,P\rangle$ must remain intact inside ambient k which cannot be opened (although some ambients may enter inside k). By examining the above reductions sequence from $\mathcal{C}_{n.\overline{\mathtt{enter}}\_k}[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$ we conclude that

$$M \Rightarrow M_1 \xrightarrow{n.\overline{\mathtt{enter}}\_k} \mathcal{D}[k[\,\circ\,]] \Rightarrow \mathcal{D}'[k[\,\circ\,]]\ .$$

As names a, b, and done are all fresh, by Lemma 3.11 applied to the sequence of transitions

$$(\boldsymbol{\nu} a)(\boldsymbol{\nu} b)\mathcal{D}'[k[\circ \mid b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]]] \mid a[\,]] \Rightarrow (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)\mathcal{D}''[\circ \mid a[\,], \mathsf{done}[\,], b[\,]]$$

we derive that there is M' such that

$$\mathcal{D}'[k[\,\circ\,]] \Rightarrow M' \equiv \mathcal{D}''[\circ \mid \mathbf{0}, \mathbf{0}, \mathbf{0}] \ .$$

As a and b are fresh and, by Lemma 3.7 $(\boldsymbol{\nu} n)n[\,] \cong_{\mathsf{s}} \mathbf{0}$, it holds that

$$\begin{array}{rcl} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle & \equiv & \mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid \mathbf{0},\mathbf{0},\mathbf{0}] \\ & \cong_{\mathsf{s}} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)\mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid a[\,],\mathbf{0},b[\,]] \end{array}$$

and hence also that:

$$\begin{array}{rcl} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] & \cong_{\mathsf{s}} & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)\mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid a[\,],\mathbf{0},b[\,]] \mid \mathsf{done}[\,] \\ & \equiv & (\boldsymbol{\nu} a)(\boldsymbol{\nu} b)\mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid a[\,],\mathsf{done}[\,],b[\,]] \\ & \equiv & N \mid \mathsf{done}[\,] \ . \end{array}$$

As done is a fresh name and $\cong_{\mathsf{s}}$ is closed under restriction, we conclude $M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \cong_{\mathsf{s}} N$, as desired. □

When proving the completeness result we implicitly use a standard property of reduction barbed congruence.

Proposition 3.13 *If $P \cong_{\mathsf{s}} Q$ then*

—$P \Downarrow n$ iff $Q \Downarrow n$
—$P \Rightarrow P'$ implies there is Q' such that $Q \Rightarrow Q'$ and $P' \cong_{\mathsf{s}} Q'$.

Similar results hold for reduction barbed congruence over processes. In the sequel we will use these properties without comment.

Theorem 3.14 (Completeness) *Reduction barbed congruence over systems is contained in late bisimilarity.*

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

— Suppose that $M \mathcal{R} N$ and that $M \xrightarrow{\alpha} M'$ where $\alpha \in \{k.\mathtt{enter}\_n, k.\mathtt{exit}\_n, n.\overline{\mathtt{enter}}\_k, k.\mathtt{open}\_n\}$. We must find a system N' such that $N \stackrel{\alpha}{\Longrightarrow} N'$ and for all P, $M' \bullet P \cong_{\mathsf{s}} N' \bullet P$.
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

$$\mathcal{D}_\alpha\langle P\rangle[-] \ = \ (\mathcal{C}_\alpha[-] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{Flip}$$

where $\mathcal{C}_\alpha[-]$ are the contexts in Table VII and Flip is the system:

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

where 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 $\cong_{\mathsf{s}}$ is preserved by system contexts, $M \cong_{\mathsf{s}} N$ implies that, for all processes P, it holds that

$$\mathcal{D}_\alpha\langle P\rangle[M] \;\cong_{\mathsf{s}}\; \mathcal{D}_\alpha\langle P\rangle[N]\ .$$

By Lemma 3.8 and 3.10(1), we can build the following reduction sequence:

$$\mathcal{D}_\alpha\langle P\rangle[M] = (\mathcal{C}_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{Flip} \;\Rightarrow M_1 \mid \mathsf{Flip} \;\Rightarrow O_1$$

with $M_1 \equiv \mathcal{D}'[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,] \cong_{\mathsf{s}} (M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{done}[\,]$, for some system context $\mathcal{D}'[-]$, and by Lemma 3.9 $O_1 \cong_{\mathsf{s}} (M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$ with $O_1 \Downarrow i, j, \mathsf{succ} \not\Downarrow \mathsf{fail}$.
This reduction must be matched by a corresponding reduction sequence

$$\mathcal{D}_\alpha\langle P\rangle[N] \;\Rightarrow\; O_2$$

where $O_1 \cong_{\mathsf{s}} O_2$ and hence $O_2 \Downarrow i, j, \mathsf{succ} \not\Downarrow \mathsf{fail}$.
The constrains on the barbs allow us to deduce the structure of the above reduction sequence. That is:

$$\mathcal{D}_\alpha\langle P\rangle[N] = (\mathcal{C}_\alpha[N] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \mid \mathsf{Flip} \;\Rightarrow N_1 \mid \mathsf{Flip} \;\Rightarrow O_2$$

with $N_1 \equiv \mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,]$, and $O_2 \cong_{\mathsf{s}} \mathcal{D}'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$ for some system contexts $\mathcal{D}''[-]$ and $\mathcal{D}'''[-]$ such that $\mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \Rightarrow \mathcal{D}'''[\mathsf{spy}_\alpha\langle i,j,P\rangle]$.
As $\mathcal{C}_\alpha[N] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;\Rightarrow N_1 \equiv \mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,]$, by Lemma 3.12 there is a system N' such that $N \stackrel{\alpha}{\Longrightarrow} N'$ and $\mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \cong_{\mathsf{s}} N' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$. As $\mathcal{D}''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \;\Rightarrow \mathcal{D}'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \Downarrow i,j$ there is N'' such that $N' \;\Rightarrow N''$ and $\mathcal{D}'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \cong_{\mathsf{s}} N'' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$.
Summarising, there is N'' such that $N \stackrel{\alpha}{\Longrightarrow} N''$ and:
—$O_1 \cong_{\mathsf{s}} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$
—$O_2 \cong_{\mathsf{s}} \mathcal{D}'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \mid \mathsf{done}[\,] \mid \mathsf{succ}[\,]$
—$\mathcal{D}'''[\mathsf{spy}_\alpha\langle i,j,P\rangle] \cong_{\mathsf{s}} N'' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$
—$O_1 \cong_{\mathsf{s}} O_2$.
As barbed congruence is preserved by restriction, we have

$$(\boldsymbol{\nu}\mathsf{done},\mathsf{succ})O_1 \cong_{\mathsf{s}} (\boldsymbol{\nu}\mathsf{done},\mathsf{succ})O_2\ .$$

By Lemma 3.7 $(\boldsymbol{\nu}\mathsf{done})\mathsf{done}[\,] \cong_{\mathsf{s}} (\boldsymbol{\nu}\mathsf{succ})\mathsf{succ}[\,] \cong_{\mathsf{s}} \mathbf{0}$, which implies

$$M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;\cong_{\mathsf{s}}\; N'' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle.$$

Observe that the placeholder must be located inside an ambient, and cannot be under a prefix. Since $\cong_{\mathsf{s}}$ is preserved by restriction, is transitive, and is closed under system contexts, we can apply Lemma 3.10(2) and we can finally derive $M' \bullet P \;\mathcal{R}\; N'' \bullet P$, for all processes P.

— Suppose now $M \,\mathcal{R}\, N$ and $M \xrightarrow{*.\mathtt{enter}\_n} M'$. We must find a system N' such that $N \mid n[\,\circ\,] \;\Rightarrow N'$ and for all P, $M' \bullet P \cong_{\mathsf{s}} N' \bullet P$.
We consider the context

$$\mathcal{C}\langle P\rangle[-] \;=\; - \mid n[\mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle]\ .$$

Since $\cong_{\mathsf{s}}$ is preserved by system contexts, for all processes P it holds that

$$\mathcal{C}\langle P\rangle[M] \;\cong_{\mathsf{s}}\; \mathcal{C}\langle P\rangle[N]\ .$$

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

$$\mathcal{C}\langle P\rangle[M] \;\Rightarrow\; M' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle$$

where $M' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle \Downarrow i,j$. Call this outcome O_1.
This reduction must be matched by a corresponding reduction

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

where $O_1 \cong_{\mathsf{s}} O_2$ and $O_2 \Downarrow i,j$. By Lemma 3.10(1) it follows that there is a system N' such that $O_2 = N' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle$ and $N \mid n[\,\circ\,] \;\Rightarrow N'$. Again, as $\cong_{\mathsf{s}}$ is preserved by restriction, from $O_1 \cong_{\mathsf{s}} O_2$ and Lemma 3.10(2) we can derive $M' \bullet P \cong_{\mathsf{s}} N' \bullet P$, for all P, as required.

— Suppose $M\ \mathcal{R}\ N$ and $M \xrightarrow{*.\mathtt{exit}\_n} M'$. In this case we must find a system N' such that $n[\circ \mid N] \;\Rightarrow N'$ and for all P, $M' \bullet P \cong_{\mathsf{s}} N' \bullet P$.
We consider the context

$$\mathcal{C}\langle P\rangle[-] \;=\; n[- \mid \mathsf{spy}_{*.\mathtt{exit}\_n}\langle i,j,P\rangle]\ .$$

Since $\cong_{\mathsf{s}}$ is preserved by system contexts, for all processes P it holds that

$$\mathcal{C}\langle P\rangle[M] \;\cong_{\mathsf{s}}\; \mathcal{C}\langle P\rangle[N]\ .$$

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

$$\mathcal{C}\langle P\rangle[M] \;\Rightarrow\; M' \bullet \mathsf{spy}_{*.\mathtt{exit}\_n}\langle i,j,P\rangle$$

where $M' \bullet \mathsf{spy}_{*.\mathtt{exit}\_n}\langle i,j,P\rangle \Downarrow i,j$. Call this outcome O_1.
This reduction must be matched by a corresponding reduction

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

where $O_1 \cong_{\mathsf{s}} O_2$ and $O_2 \Downarrow i,j$. By Lemma 3.10(1) it follows that there is a system N' such that $O_2 = N' \bullet \mathsf{spy}_{*.\mathtt{enter}\_n}\langle i,j,P\rangle$ and $n[\circ \mid N] \;\Rightarrow N'$. Again, as $\cong_{\mathsf{s}}$ is preserved by restriction, from $O_1 \cong_{\mathsf{s}} O_2$ and Lemma 3.10(2) we can derive $M' \bullet P \cong_{\mathsf{s}} N' \bullet P$, for all P, as required.

This concludes the analysis. □

As a consequence:

Theorem 3.15 (Characterisation of $\cong_{\mathsf{s}}$) *Late bisimilarity, early bisimilarity, and reduction barbed congruence over systems coincide.*

Proof Theorem 3.6 states that $\approx \subseteq \approx_e$ and $\approx_e \subseteq \cong_{\mathsf{s}}$. Theorem 3.14 states the reduction barbed congruence over systems is contained in late bisimilarity, that is $\cong_{\mathsf{s}} \subseteq \approx$. We hence have the following chain of inclusions $\cong_{\mathsf{s}} \subseteq \approx \subseteq \approx_e \subseteq \cong_{\mathsf{s}}$. □

A remark on transitivity of (late) bisimilarity. Giving a direct proof that $\approx$ is a transitive relation is difficult. At the same time, the characterisation result does not rely on the transitivity of $\approx$. As $\cong_{\mathsf{s}}$ is an equivalence relation, late and early bisimilarity are also equivalence relations.

4. UP-TO PROOF TECHNIQUES

In the previous section we presented a labelled characterisation of reduction barbed congruence. To prove that two systems are equivalent using the labelled characterisation, it is necessary to exhibit a relation $\mathcal{R}$ and to show that it is a bisimulation. If we ignore for a moment the asynchronous actions and the instantiation of the placeholder, the proof obligation consists of verifying that if $M\ \mathcal{R}\ N$ and $M \stackrel{\alpha}{\longrightarrow} M'$, there exists a system N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and $M'\ \mathcal{R}\ N'$. The idea behind *up-to proof techniques* [Sangiorgi and Milner 1992; Sangiorgi 1998] is to replace the heavy proof condition $M'\ \mathcal{R}\ N'$ with a weaker condition of the form $M'\ \mathcal{S}\mathcal{R}\mathcal{S}\ N'$, where $\mathcal{S}$ is another relation on systems. In this case we talk of *bisimulation up to* $\mathcal{S}$, and of *up-to* $\mathcal{S}$ *proof technique*. The role of the $\mathcal{S}$ relation is to abstract some details of the systems M' and N' being tested. For instance, the up-to $\equiv$ proof technique replaces the condition $M'\ \mathcal{R}\ N'$ by $M' \equiv\mathcal{R}\equiv N'$; the size of the relation $\mathcal{R}$ can be greatly reduced since $\mathcal{R}$ we need only consider representatives of the equivalence classes of structurally equivalent terms. In general a bisimulation up to $\mathcal{S}$ is not a bisimulation. However, for some well-chosen relations $\mathcal{S}$, it can be shown that if two processes are related by a bisimulation up to $\mathcal{S}$, then there exists also a bisimulation relating them (this result is called soundness of the up-to $\mathcal{S}$ technique). For instance, the soundness of the up-to $\equiv$ proof technique follows easily from Lemma 2.4.

In this section we focus on two powerful up-to techniques: *up-to expansion* [Sangiorgi and Milner 1992], and *up-to context* [Sangiorgi 1996b]. As in the π-calculus, these techniques can be merged.

When proving that two systems are bisimilar it is often useful to abstract from their internal behaviour. Whereas bisimulation up to bisimilarity would be useful, this proof technique is unsound [Sangiorgi and Milner 1992] for weak equivalences. However, a variation, namely bisimulation up to expansion, is indeed sound. The *expansion relation* [Arun-Kumar and Hennessy 1992], written $\lesssim$, is an asymmetric variant of bisimilarity which allows us to count the number of silent moves performed by a system. Intuitively, $M \lesssim N$ holds if M and N are bisimilar and N has at least as many τ-moves as M. This constraint on the number of internal reductions allows to recover the soundness of the up-to expansion proof technique, and in many practical cases when $M \approx N$ holds, M and N are ordered by expansion. To define expansion we introduce the following notation: $\stackrel{\hat{\tau}}{\longrightarrow}$ is $\stackrel{\tau}{\longrightarrow} \cup \mathcal{I}$, where $\mathcal{I}$ is the identity relation; if $\alpha \neq \tau$ then $\stackrel{\hat{\alpha}}{\longrightarrow}$ is $\stackrel{\alpha}{\longrightarrow}$. The expansion relation is then defined as follows.

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

—if $M \stackrel{\alpha}{\longrightarrow} M'$, $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there exists a system N' such that $N \stackrel{\alpha}{\Longrightarrow} N'$ and for all processes P it holds that $M' \bullet P\ \mathcal{R}\ N' \bullet P$;

—if $M \xrightarrow{*.\texttt{enter}\_n} M'$ then there exists a system N' such that $N \mid n[\circ] \Rightarrow N'$ and for all processes P it holds that $M' \bullet P \mathcal{R} N' \bullet P$;

—if $M \xrightarrow{*.\texttt{exit}\_n} M'$ then there exists a system N' such that $n[\circ \mid N] \Rightarrow N'$ and for all processes P it holds that $M' \bullet P \mathcal{R} N' \bullet P$;

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

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

—if $N \xrightarrow{*.\texttt{exit}\_n} N'$ 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 context and up to* $(\gtrsim, \approx)$)** A symmetric relation $\mathcal{R}$ over systems is a *bisimulation up to context and up to* $(\gtrsim, \approx)$ if $M \mathcal{R} N$ implies:

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

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

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

To prove that the bisimulation up to context and up to $(\gtrsim, \approx)$ is a sound proof technique we first need a technical lemma.

Lemma 4.3 *Let $\mathcal{R}$ be a bisimulation up to context and up to $(\gtrsim, \approx)$. If $M \mathcal{R} N$ and for some system context $\mathcal{C}[-]$ and system M'' it holds that $\mathcal{C}[M] \xrightarrow{\alpha} M''$ for $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there exists a system N'' such that $\mathcal{C}[N] \xRightarrow{\hat{\alpha}} N''$ and for all processes P there are a system context $\mathcal{C}'[-]$ and systems M', N' such that $M'' \bullet P \gtrsim \mathcal{C}'[M']$, $N'' \bullet P \approx \mathcal{C}'[N']$ and $M' \mathcal{R} N'$.*

The proof demands an analysis of the interactions between M and $\mathcal{C}[-]$, and is reported in Appendix C. The lemma above generalises to weak transitions.

Corollary 4.4 *Let $\mathcal{R}$ be a bisimulation up to context and up to $(\gtrsim, \approx)$. If $M \mathcal{R} N$ and for some system context $\mathcal{C}[-]$ and system M'' it holds that $\mathcal{C}[M] \xRightarrow{\hat{\alpha}} M''$ for $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$, then there exists a system N'' such that $\mathcal{C}[N] \xRightarrow{\hat{\alpha}} N''$ and for all processes P there are a system context $\mathcal{C}'[-]$ and systems M', N' such that $M'' \bullet P \gtrsim \mathcal{C}'[M']$, $N'' \bullet P \approx \mathcal{C}'[N']$ and $M' \mathcal{R} N'$.*

Proof The results follows by induction on the length of the transition $\mathcal{C}[M] \xRightarrow{\hat{\alpha}} M''$, using Lemma 4.3 and standard reasoning on the expansion relation. □

Theorem 4.5 *If $\mathcal{R}$ is a bisimulation up to context and up to $(\gtrsim, \approx)$, then $\mathcal{R} \subseteq \approx$.*

Proof We show that the relation

$$\mathcal{S} \;=\; \{(M,N) :\; \exists \mathcal{C}[-], M', N' \text{ such that } \mathcal{C}[M'] \approx M, \mathcal{C}[N'] \approx N, \text{ and } M' \;\mathcal{R}\; N'\}$$

is a bisimulation.

— Suppose $(M,N) \in \mathcal{S}$ and $M \xrightarrow{\alpha} M_1$ where $\alpha \notin \{*.\texttt{enter}\_n, *.\texttt{exit}\_n\}$. Since $(M,N) \in \mathcal{S}$, there exist $\mathcal{C}[-], M', N'$ such that $M \approx \mathcal{C}[M']$, and $N \approx \mathcal{C}[N']$, and $M' \;\mathcal{R}\; N'$. The definition of bisimilarity ensures that there exists M_1' such that $\mathcal{C}[M'] \stackrel{\hat{\alpha}}{\Longrightarrow} M_1'$, and that for all P it holds that $M_1 \bullet P \approx M_1' \bullet P$. Corollary 4.4 tells us that there exist $N_1', \mathcal{C}'[-], M_2, N_2$ such that $\mathcal{C}[N'] \stackrel{\hat{\alpha}}{\Longrightarrow} N_1'$, and $M_1' \bullet P \gtrsim \mathcal{C}'[M_2]$, and $N_1' \bullet P \approx \mathcal{C}'[N_2]$, where $M_2 \;\mathcal{R}\; N_2$. As $N \approx \mathcal{C}[N']$, this implies that there exists a system N_1 such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N_1$ and $N_1' \bullet P \approx N_1 \bullet P$. In turn, we have that $M_1 \bullet P \approx \mathcal{C}'[M_2]$ and $N_1 \bullet P \approx \mathcal{C}'[N_2]$ where $M_2 \;\mathcal{R}\; N_2$. The construction of $\mathcal{S}$ ensures that $M_1 \bullet P \;\mathcal{S}\; N_1 \bullet P$, as required.

— Suppose $(M,N) \in \mathcal{S}$ and $M \xrightarrow{\alpha} M_1$, where $\alpha = *.\texttt{enter}\_n$. We want to prove that there is N_1 such than $N \mid n[\,\circ\,] \Rightarrow N_1$ and that for all processes P, we have $M_1 \bullet P \;\mathcal{S}\; N_1 \bullet P$. Let P be an arbitrary process. Since $(M,N) \in \mathcal{S}$, there exist $\mathcal{C}[-], M', N'$ such that $M \approx \mathcal{C}[M']$, $\mathcal{C}[N'] \approx N$, and $M' \;\mathcal{R}\; N'$. The definition of bisimilarity ensures that there exists M_1' such that $\mathcal{C}[M'] \mid n[\,\circ\,] \Rightarrow M_1'$ and, for all processes R, it holds that $M_1 \bullet R \approx M_1' \bullet R$. In particular, $M_1 \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \approx M_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$, for i and j fresh. As the placeholder $\circ$ does not reduce, we have $\mathcal{C}[M'] \mid n[\mathsf{spy}_\alpha\langle i,j,P\rangle] \Rightarrow M_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$. We use Corollary 4.4 (the context we consider is $\mathcal{C}[-] \mid n[\mathsf{spy}_\alpha\langle i,j,P\rangle]$) and the presence of fresh barbs i and j to deduce that there exist $N_1', \mathcal{C}'[-], M_2, N_2$ such that $\mathcal{C}[N'] \mid n[\mathsf{spy}_\alpha\langle i,j,P\rangle] \Rightarrow N_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$, with $M_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \gtrsim \mathcal{C}'[M_2]$ and $N_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \approx \mathcal{C}'[N_2]$, where $M_2 \;\mathcal{R}\; N_2$. Now, since $\mathcal{C}[N'] \approx N$ and the bisimilarity is closed under parallel composition with systems, we have that $\mathcal{C}[N'] \mid n[\mathsf{spy}_\alpha\langle i,j,P\rangle] \approx N \mid n[\mathsf{spy}_\alpha\langle i,j,P\rangle]$. The definition of bisimilarity and the presence of fresh barbs i and j ensures that there exists a system N_1 such that $N \mid n[\mathsf{spy}_\alpha\langle i,j,P\rangle] \Rightarrow N_1 \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$ and $N_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \approx N_1 \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$. From the above weak transition we derive $N \mid n[\,\circ\,] \Rightarrow N_1$. Moreover, by Lemma 3.10(2) and because both $\approx$ and $\gtrsim$ are preserved by restriction, it holds that $M_1 \bullet P \approx (\boldsymbol{\nu} i,j)(M_1 \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \approx (\boldsymbol{\nu} i,j)(M_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \gtrsim (\boldsymbol{\nu} i,j)\mathcal{C}'[M_2]$. Similarly, $(\boldsymbol{\nu} i,j)\mathcal{C}'[N_2] \approx (\boldsymbol{\nu} i,j)(N_1' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \approx (\boldsymbol{\nu} i,j)(N_1 \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle) \approx N_1 \bullet P$. Thus, $M_1 \bullet P \;\mathcal{S}\; N_1 \bullet P$, as required.

— The case $\alpha = *.\texttt{exit}\_n$ is analogous to the previous one. □

5. A SEMANTIC THEORY FOR PROCESSES

In this section we characterise reduction barbed congruence over processes, $\cong_{\mathsf{p}}$, in terms of our labelled bisimilarity over systems, $\approx$.

The relation $\cong_{\mathsf{p}}$ is closed under arbitrary process contexts: reducing the number of contexts in the quantification is a first step towards the definition of a useful proof technique, and, broadly speaking, towards an understanding of the behavioural theory of processes.

We show that it is possible to work with a lighter definition of contextuality. In particular it suffices to require closure under the two crucial operators of MA: parallel composition (to model concurrency) and ambient construct (to model locality).

***Definition* 5.1** *Reduction barbed equivalence over processes*, written $\cong_{\mathsf{p}}^{\mathsf{e}}$, is the largest symmetric relation over processes which is reduction closed, barb preserving, and closed under parallel composition and ambient construct.

Theorem 5.2 (Context Lemma) *The relations $\cong_{\mathsf{p}}$ and $\cong_{\mathsf{p}}^{\mathsf{e}}$ coincide.*

Reduction barbed equivalence over processes still requires a universal quantification on non-trivial contexts. More than that, a direct proof of the above context lemma is surprisingly difficult. To overcome this difficulty, we first develop a characterisation of $\cong_{\mathsf{p}}^{\mathsf{e}}$ in terms of the labelled bisimulation of Section 3, and we postpone the proof of the context lemma after Theorem 5.3.

Theorem 5.3 (Characterisation of $\cong_{\mathsf{p}}^{\mathsf{e}}$) *Let*

$$\mathcal{S} = \{(P,Q) :\ k[P \mid R] \approx k[Q \mid R],\ \textit{for all } k, R\}\ .$$

The relations $\cong_{\mathsf{p}}^{\mathsf{e}}$ and $\mathcal{S}$ coincide.

To prove Theorem 5.3 we need some technical lemmas. The next two lemmas (their proofs are reported in the Appendix D) are necessary for proving the completeness part of Theorem 5.3. In particular Lemma 5.4 says that reduction barbed equivalence over processes is preserved by restriction. This result will be also useful when proving the context lemma.

Lemma 5.4 *If $P \cong_{\mathsf{p}}^{\mathsf{e}} Q$, then $(\boldsymbol{\nu} n)P \cong_{\mathsf{p}}^{\mathsf{e}} (\boldsymbol{\nu} n)Q$.*

Lemma 5.5 *$\cong_{\mathsf{p}}^{\mathsf{e}} \cap (\mathcal{M} \times \mathcal{M}) \subseteq \cong_{\mathsf{s}}$, where $\mathcal{M}$ is the set of all systems.*

The three lemmas below are important tools used to prove that the relations $\mathcal{S}$ and $\cong_{\mathsf{p}}^{\mathsf{e}}$ coincide.

Lemma 5.6 *Let P, Q be two processes such that $k[P] \approx k[Q]$. Let r be a name fresh for P and Q, and different from k. Then $k[\mathtt{open}\_r.P] \approx k[\mathtt{open}\_r.Q]$.*

Proof The argument proceeds by contradiction. Suppose that $k[\mathtt{open}\_r.P] \not\approx k[\mathtt{open}\_r.Q]$. By Theorem 3.15, $k[\mathtt{open}\_r.P] \not\cong_{\mathsf{s}} k[\mathtt{open}\_r.Q]$. As a consequence, there must exist a system context $\mathcal{C}[-]$ that tells $k[\mathtt{open}\_r.P]$ and $k[\mathtt{open}\_r.Q]$ apart. In doing so $\mathcal{C}[-]$ must interact with the processes P and Q. This implies that the context must necessarily consume the $\mathtt{open}\_r$ capability of both $\mathtt{open}\_r.P$ and $\mathtt{open}\_r.Q$. Without any loss of generality, the context $\mathcal{C}[-]$ can be assumed to be of the form $\mathcal{D}[- \mid r[\mathtt{in}\_k]]$, where $\mathcal{D}[-]$ is a system context with $r \notin \mathrm{fn}(\mathcal{D}[-])$; the ambient $r[\mathtt{in}\_k]$ has the (exclusive) role of entering inside the ambient k and consuming the capability $\mathtt{open}\_r$. It is now clear that if the context $\mathcal{D}[- \mid r[\mathtt{in}\_k]]$ tells $k[\mathtt{open}\_r.P]$ and $k[\mathtt{open}\_r.Q]$ apart, then the context $\mathcal{D}[-]$ tells $k[P]$ and $k[Q]$ apart. In turn, this implies that $k[P] \not\cong_{\mathsf{s}} k[Q]$. By Theorem 3.15, this contradicts the hypothesis that $k[P] \approx k[Q]$. □

Lemma 5.7 *Let r be a name fresh for the process P. It holds that*

$$n[P \mid R] \approx n[(\boldsymbol{\nu} r)(\mathtt{open}\_r.P \mid r[\,]) \mid R] \ .$$

Proof The relation

$$\mathcal{R} = \{\ (k[P \mid Q], n[(\boldsymbol{\nu} r)(\mathtt{open}\_r.P \mid r[\,]) \mid Q])\ :\ \forall k, Q\ \}^{=}\ \cup\ \mathcal{I}$$

is a bisimulation up to context and up to $\equiv$. □

Lemma 5.8 *Let P and Q be two processes, and k an ambient name. If $k[P] \approx k[Q]$, then for all n, R it holds that $n[P \mid R] \approx n[Q \mid R]$.*

Proof Let n and R be respectively a name and a process. Let r be a fresh name (that is, $r \notin \mathrm{fn}(P, Q, R)$ and $r \neq n, k$). By Lemma 5.6, we have

$$k[\mathtt{open}\_r.P] \approx k[\mathtt{open}\_r.Q] \ .$$

The definition of bisimulation assures us that if $k[\mathtt{open}\_r.P] \xrightarrow{k.\mathtt{open}\_n} n[\mathtt{open}\_r.P \mid \circ]$, then there is a matching transition $k[\mathtt{open}\_r.Q] \xrightarrow{k.\mathtt{open}\_n} n[\mathtt{open}\_r.Q \mid \circ]$, and that for all processes R' it holds that $n[\mathtt{open}\_r.P \mid R'] \approx n[\mathtt{open}\_r.Q \mid R']$. Observe that the matching transition must be strong, because the prefix $\mathtt{open}\_r$ prevents Q from reducing. By taking $R' = r[\,] \mid R$, we have

$$n[\mathtt{open}\_r.P \mid r[\,] \mid R] \approx n[\mathtt{open}\_r.Q \mid r[\,] \mid R] \ .$$

As $\approx$ is preserved by restriction and $r \notin \mathrm{fn}(R)$, we have

$$n[(\boldsymbol{\nu} r)(\mathtt{open}\_r.P \mid r[\,]) \mid R] \approx n[(\boldsymbol{\nu} r)(\mathtt{open}\_r.Q \mid r[\,]) \mid R] \ .$$

By Lemma 5.7 and transitivity of bisimulation, we conclude $n[P \mid R] \approx n[Q \mid R]$. □

Everything is now in place to prove Theorem 5.3.

Proof of Theorem 5.3. We first show that $P \cong_{\mathsf{p}}^{\mathsf{e}} Q$ implies $P\ \mathcal{S}\ Q$. For that, we must show that for all k, R, it holds that $k[P \mid R] \approx k[Q \mid R]$. Both $k[P \mid R]$ and $k[Q \mid R]$ are systems, and it holds that $k[P \mid R] \cong_{\mathsf{p}}^{\mathsf{e}} k[Q \mid R]$ because $\cong_{\mathsf{p}}^{\mathsf{e}}$ is closed under parallel composition and ambient construct. The result follows from Lemma 5.5 and Theorem 3.15.

It remains to prove that $\mathcal{S} \subseteq \cong_{\mathsf{p}}^{\mathsf{e}}$. For that, we must show that $\mathcal{S}$ is reduction closed, barb preserving, and closed under parallel composition and ambient construct.

(1) *$\mathcal{S}$ is reduction closed.* Suppose $P\ \mathcal{S}\ Q$ and $P \rightarrow P'$. Let n be a name such that $n \notin \mathrm{fn}(P, Q)$. We have $n[P] \approx n[Q]$, by definition of $\mathcal{S}$. As $n \notin \mathrm{fn}(P, Q)$, and because of the correspondence between τ-transitions and reductions, there is a system M such that $n[P] \xrightarrow{\tau} M \equiv n[P']$. As $n[P] \approx n[Q]$, there is N such that $n[Q] \Rightarrow N$ and $M \approx N$. As $n \notin \mathrm{fn}(P, Q)$, there must be Q' such that $Q \rightarrow^{*} Q'$ and $N \equiv n[Q']$; thus $n[P'] \approx n[Q']$. Lemma 5.8 allows us to derive $P'\ \mathcal{S}\ Q'$, as desired.

(2) *$\mathcal{S}$ is barb preserving.* Suppose that $P\ \mathcal{S}\ Q$ and $P \Downarrow n$. Consider the context

$$\mathcal{C}[-] = b[- \mid a[\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{ok}[\mathtt{out}\_a.\mathtt{out}\_b]]]$$

where a, b and ok are fresh for both P and Q. Then $\mathcal{C}[P] \approx \mathcal{C}[Q]$ by definition of $\mathcal{S}$. As $P \Downarrow n$, the construction of $\mathcal{C}[-]$ assures that $\mathcal{C}[P] \Downarrow \mathsf{ok}$. Bisimilarity is barb preserving and $\mathcal{C}[Q] \Downarrow \mathsf{ok}$ must hold. The construction of $\mathcal{C}[-]$ guarantees that $Q \Downarrow n$.

(3) *$\mathcal{S}$ is closed under parallel composition and ambient construct.* We first show that $P \mathcal{S} Q$ implies $P \mid R \mathcal{S} Q \mid R$. By definition of $\mathcal{S}$ we have $k[P \mid R'] \approx k[Q \mid R']$ for all k, R'. By taking $R' = R \mid R''$ for arbitrary R'' we have $k[P \mid R \mid R''] \approx k[Q \mid R \mid R'']$ for all R''. This implies $P \mid R \mathcal{S} Q \mid R$. We then show that $P \mathcal{S} Q$ implies $n[P] \mathcal{S} n[Q]$. By definition of $\mathcal{S}$ we have $n[P] \approx n[Q]$ for all n. The result follows from the closure of $\approx$ under static contexts. □

The characterisation of $\cong^{\mathsf{e}}_{\mathsf{p}}$ is a fundamental tool to reason about processes. As a first application, we prove the context lemma.

Proofof Theorem 5.2. We have to show that $\cong^{\mathsf{e}}_{\mathsf{p}} = \cong_{\mathsf{p}}$. The inclusion $\cong_{\mathsf{p}} \subseteq \cong^{\mathsf{e}}_{\mathsf{p}}$ is straightforward. For the converse we must prove that

(1) $\cong^{\mathsf{e}}_{\mathsf{p}}$ is reduction closed;
(2) $\cong^{\mathsf{e}}_{\mathsf{p}}$ is barb preserving;
(3) $\cong^{\mathsf{e}}_{\mathsf{p}}$ is closed under arbitrary contexts.

Conditions 1 and 2 hold by definition of $\cong^{\mathsf{e}}_{\mathsf{p}}$. It remains to show that the relation $\cong^{\mathsf{e}}_{\mathsf{p}}$ is preserved by all process contexts. The relation $\cong^{\mathsf{e}}_{\mathsf{p}}$ is preserved by parallel composition and ambient constructor by definition. It is also preserved by restriction by Lemma 5.4. It remains to prove that it is preserved by prefixing and replicated prefixing. We report the proof that $\cong^{\mathsf{e}}_{\mathsf{p}}$ is preserved by prefixing in the Appendix, and we focus on replicated prefixing.

We have to prove that if $P \cong^{\mathsf{e}}_{\mathsf{p}} Q$, then $!C.P \cong^{\mathsf{e}}_{\mathsf{p}} !C.Q$. Rather than working directly with $\cong^{\mathsf{e}}_{\mathsf{p}}$, we use Theorem 5.3 and we prove that $!C.P \mathcal{S} !C.Q$. For that, we show that $k[!C.P \mid R] \approx k[!C.Q \mid R]$ for all k and R. We perform a case analysis on C.

Suppose that $C = \mathtt{in}\_o$. We show that the relation

$$\mathcal{R} = \{(n[!\mathtt{in}\_o.P \mid R], n[!\mathtt{in}\_o.Q \mid R]) \ : \ P \cong^{\mathsf{e}}_{\mathsf{p}} Q\}^{=} \cup \approx$$

is a *bisimulation up to context and up to* $(\gtrsim, \approx)$.

The most interesting case is when the process $!\mathtt{in}\_o.P$ exercises the capability $\mathtt{in}\_o$. Suppose

$$n[!\mathtt{in}\_o.P \mid R] \xrightarrow{n.\mathtt{enter}\_o} o[n[P \mid !\mathtt{in}\_o.P \mid R] \mid \circ] \ .$$

We have a matching transition

$$n[!\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\mathtt{enter}\_o} o[n[Q \mid !\mathtt{in}\_o.Q \mid R] \mid \circ] \ .$$

Since $P \cong^{\mathsf{e}}_{\mathsf{p}} Q$, we have $P \mathcal{S} Q$ and in turn, for all R', we have $n[P \mid R'] \approx n[Q \mid R']$. As $\approx$ is preserved by system contexts, for all instantiations of $\circ$ it holds that $o[n[P \mid R'] \mid \circ] \approx o[n[Q \mid R'] \mid \circ]$. By taking $R' = !\mathtt{in}\_o.Q \mid R$, we obtain

$$o[n[!\mathtt{in}\_o.Q \mid R \mid P] \mid \circ] \approx o[n[Q \mid !\mathtt{in}\_o.Q \mid R] \mid \circ] \ .$$

Then, for all processes S, the following hold:

$$o[n[P \mid !\mathtt{in}\_o.P \mid R] \mid \circ] \bullet S \;\gtrsim\; \mathcal{C}[n[!\mathtt{in}\_o.P \mid R \mid P]]$$
$$o[n[Q \mid !\mathtt{in}\_o.Q \mid R] \mid \circ] \bullet S \;\approx\; \mathcal{C}[n[!\mathtt{in}\_o.Q \mid R \mid P]]$$

where $\mathcal{C}[-] = o[- \mid S]$ (we can rearrange the terms using structural congruence because $\equiv \subseteq \gtrsim$ and $\equiv \subseteq \approx$). By construction of $\mathcal{R}$ we have

$$n[!\mathtt{in}\_o.P \mid R \mid P] \;\mathcal{R}\; n[!\mathtt{in}\_o.Q \mid R \mid P]$$

and we can conclude that up to context and up to $(\gtrsim, \approx)$ we are still in $\mathcal{R}$.

The cases $C = \mathtt{out}\_o$ and $C = \mathtt{open}\_o$ follow along similar lines. □

The result below is a consequence of Theorems 5.2 and 5.3.

Theorem 5.9 (Characterisation of $\cong_{\mathsf{p}}$) *The relations $\mathcal{S}$ and $\cong_{\mathsf{p}}$ coincide.*

The relation $\mathcal{S}$ still involves a universal quantification over all the processes R. Yet, it is built on top of $\approx$ and it can be coupled with the up-to proof techniques. In turn, it reveals a useful tool to reason about processes, as illustrated by the proof of the context lemma and by the other examples given in Section 7.

Systems revisited. In Section 3, we conjectured that reduction barbed congruence over systems ($\cong_{\mathsf{s}}$) *is* "the right" equality when working with systems. We are now in measure to close the conjecture. In fact, if we restrict our attention to systems, we can show that system contexts have the same discriminating power as arbitrary contexts.

Theorem 5.10 *Let M and N be two systems, then $M \cong_{\mathsf{s}} N$ if and only if $M \cong_{\mathsf{p}} N$.*

Proof By definition, $M \cong_{\mathsf{p}} N$ implies $M \cong_{\mathsf{s}} N$. For the converse, by Theorem 3.15, if $M \cong_{\mathsf{s}} N$ then $M \approx N$. As $\approx$ is preserved by system contexts, for all n and R $n[M \mid R] \approx n[N \mid R]$. By Theorems 5.3 and 5.2 it follows that $M \cong_{\mathsf{p}} N$. □

This in turn implies a strong result: $\approx$ *completely characterises* $\cong_{\mathsf{p}}$ *on systems.*

Theorem 5.11 *Let M and N be two systems, then $M \cong_{\mathsf{p}} N$ if and only if $M \approx N$.*

A preliminary investigation suggests that it might be possible to define directly an LTS and associated bisimulation for processes, extending the approach of Sections 2 and 3. The idea is that a process that exercises the $\mathtt{in}\_n$ capability, such as $\mathtt{in}\_n.P_1 \mid P_2$, can realise a τ-action by interacting with a context of the form $k[- \mid R_1] \mid n[R_2]$, where the name k and the processes R_1 and R_2 are arbitrary. The case of the $\mathtt{out}\_n$ capability is similar, while the one for the $\mathtt{open}\_n$ capability might be simpler. However, extending our LTS with the env-actions that capture these interactions introduces the necessity of dealing with two arbitrary processes provided by the context, instead of only one. On the one hand this increases greatly the number of labels that must be checked when proving the equivalence of processes, and on the other hand this adds a tremendous complexity to the proof of the congruence of bisimilarity, of full-abstraction, and of soundness of the up-to proof technique.

6. ADDING COMMUNICATION

In this section we adapt our characterisation to the calculus extended with communication of capabilities. For that we introduce a (countable, infinite) set of variables,

Table IX. Message-passing Mobile Ambients

Names:	$a, b, \ldots, k, l, m, n, \ldots \in \mathbf{N}$		*Systems:*		
Variables:	$x, y, \ldots \in \mathbf{V}$		M, N ::=	$\mathbf{0}$	termination
Capabilities:				$M_1 \mid M_2$	parallel composition
C ::=	$\mathtt{in\_}n$	may enter into n		$(\boldsymbol{\nu} n)M$	restriction
	$\mathtt{out\_}n$	may exit out of n		$n[P]$	ambient
	$\mathtt{open\_}n$	may open n			
			Processes:		
Expressions:			P, Q, R ::=	$\mathbf{0}$	nil process
E, F ::=	x	variable		$P_1 \mid P_2$	parallel composition
	C	capability		$(\boldsymbol{\nu} n)P$	restriction
	$E.F$	path		$G.P$	prefixing
	ε	empty path		$n[P]$	ambient
				$!G.P$	replication
Guards:				$\langle E \rangle$	output
G ::=	E	expression			
	(x)	input			

denoted by $\mathbf{V}$ and ranged over by $x, y, \ldots$, disjoint from the set of names and from the placeholder $\circ$. Variables are bound by the input capability (x). The basic idea is to have an *output process* $\langle E \rangle$, which outputs the message E, and an input process $(x).Q$ where the variable x is bound in the continuation Q. An output action can synchronise with an input action provided that both take place inside the same ambient, and the result is that the message E is bound to the variable x in Q. A messages is a sequence of capabilities.

Unlike [Cardelli and Gordon 2000; Levi and Sangiorgi 2000], we do not allow ambient names to be transmitted. This has been a deliberate choice, for several reasons. First of all, if communication of names is allowed, then reduction can generate nonsensical terms such as $\mathtt{in\_}n[\ldots]$. While it is possible to rule out such terms using a type-system, studying the behavioural theory of typed ambients is out of the scope of this paper. Also, as suggested by Cardelli and Gordon in their seminal paper [Cardelli and Gordon 2000], communicating ambient names is a dangerous operation, as, a priori, when the name is transmitted the recipient gets considerable control over that ambient. Lastly, a technical point. It is well-known that in π-calculus a weak late bisimilarity that matches strong actions against weak ones fails to be an equivalence relation. As a consequence, great care would be required to ensure that, if communication of names is added to Mobile Ambients, the resulting late bisimilarity is still an equivalence relation.

The syntax of the extended language is given in Table IX. We assume an understanding of free and bound variables ($\mathrm{fv}(\cdot)$ and $\mathrm{bv}(\cdot)$), and of *substitutions*. Processes are identified up to α-conversion of bound variables. A process P is said to be *closed* if $\mathrm{fv}(P) = \emptyset$; otherwise is said to be *open*. Observe that the definition of open process (and of closed process) does not take into account the set of free names of the process, but only its free variables.

The structural and reduction rules below define the semantics of communication:

$$E.(F.P) \;\equiv\; (E.F).P \qquad \varepsilon.P \;\rightarrowtriangle\; P \qquad (x).P \mid \langle E \rangle \;\rightarrowtriangle\; P\{^{E}/_{x}\} \;.$$

Table X. Pre-actions, Concretions and Labelled Transition System for Communication

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

$$(\pi\ \text{Output})\ \frac{-}{\langle E\rangle \xrightarrow{\langle - \rangle} \langle E\rangle \mathbf{0}} \qquad (\pi\ \text{Input})\ \frac{-}{(x).P \xrightarrow{(E)} P\{^{E}/_{x}\}} \qquad (\pi\ \text{Path})\ \frac{E.(F.P) \xrightarrow{\pi} Q}{(E.F).P \xrightarrow{\pi} Q}$$

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

The LTS is extended by the introduction of two new pre-actions (E) for input, $\langle - \rangle$ for output, and a new form of concretion $(\boldsymbol{\nu}\tilde{m})\langle E\rangle Q$; intuitively the message E is buffered in the concretion, Q is the outcome of the output action, and $\tilde{m}$ are the names shared by E and Q. In Table X we give the rules that should be added to those of Table IV and Table V to define the LTS for the closed processes of the extended calculus. Note that in the structural rules of Table IV 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 pre-action for output allows a uniform treatment of extrusion of names. Definition 3.2 and the extended LTS induce a bisimilarity relation, still denoted by $\approx$, over the closed systems of the message passing calculus.

We define the *open extension* $\mathcal{R}^{\mathsf{o}}$ of a relation $\mathcal{R}$ as: $P\ \mathcal{R}^{\mathsf{o}}\ Q$ if and and only if for every closing substitution σ mapping variables into expressions, we have $P\sigma\ \mathcal{R}\ Q\sigma$.

Theorem 6.1 (Characterisation of $\cong_{\mathsf{s}}^{\mathsf{o}}$) *In the message-passing calculus, the relations $\approx^{\mathsf{o}}$ and $\cong_{\mathsf{s}}^{\mathsf{o}}$ coincide.*

Proof The extension of Theorem 3.6 (soundness of bisimilarity) to the message-passing calculus is straightforward. The extension of Theorem 3.14 (completeness of bisimilarity) follows because these relations are defined over systems and communication cannot be observed at top-level. □

The open extension of the relation $\mathcal{S}$, written $\mathcal{S}^{\mathsf{o}}$ can be shown equivalent to the relation

$$\mathcal{S}^{\mathsf{o}} = \{(P,Q) : k[P \mid R] \approx^{\mathsf{o}} k[Q \mid R],\ \text{for all } k, R \text{ closed}\}\ .$$

Our characterisation of reduction barbed equivalence over processes lifts smoothly to the message passing calculus.

Theorem 6.2 (Characterisation of $\cong_{\mathsf{p}}^{\mathsf{e}\,\mathsf{o}}$) *The relations $\cong_{\mathsf{p}}^{\mathsf{e}\,\mathsf{o}}$ and $\mathcal{S}^{\mathsf{o}}$ coincide over processes in the message-passing calculus.*

Proof It is an easy extension of the proof of Theorem 5.3 to the closed terms of the message passing calculus. The result then follows from the definition of open extension. □

The context lemma can be rephrased for the message passing calculus.

Theorem 6.3 *Relations $\cong_{\mathsf{p}}^{\mathsf{e}\,\mathsf{o}}$ and $\cong_{\mathsf{p}}^{\mathsf{o}}$ coincide over processes in the message-passing calculus.*

Proof The proof is an extension of the proof in the case without communication. We detail the case of closure under input prefix and replicated input prefix (for all the other cases it is enough to consider closed terms).

Suppose that $P \cong^{\mathsf{e}\,\mathsf{o}}_{\mathsf{p}} Q$ and that $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\}$. We want to show that $(x).P \cong^{\mathsf{e}}_{\mathsf{p}} (x).Q$. For that we use our characterisation of $\cong^{\mathsf{e}}_{\mathsf{p}}$ and we prove that for all n, R closed it holds that $n[(x).P \mid R] \approx n[(x).Q \mid R]$. In particular, we prove that the relation

$$\mathcal{R} = \{(n[(x).P \mid R], n[(x).Q \mid R]) \ : \\ P \cong^{\mathsf{e}\,\mathsf{o}}_{\mathsf{p}} Q,\ \mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\},\ \text{for all } n, R \text{ closed}\}^{=} \cup \approx$$

is a bisimulation up to context and up to structural congruence. The most interesting case is when $n[(x).P \mid R] \xrightarrow{\tau} n[(\boldsymbol{\nu}\tilde{r})(P\{^{E}/_{x}\} \mid R')] \equiv (\boldsymbol{\nu}\tilde{r})n[P\{^{E}/_{x}\} \mid R']$, where $n \notin \tilde{r}$. Observe that R sends the message E and resumes as R'. So we have a matching transition $n[(x).Q \mid R] \xrightarrow{\tau} \equiv (\boldsymbol{\nu}\tilde{r})n[Q\{^{E}/_{x}\} \mid R']$. Since $P \cong^{\mathsf{e}\,\mathsf{o}}_{\mathsf{p}} Q$, it holds that $P\{^{E}/_{x}\} \cong^{\mathsf{e}}_{\mathsf{p}} Q\{^{E}/_{x}\}$. The characterisation of $\cong^{\mathsf{e}}_{\mathsf{p}}$ guarantees that $n[P\{^{E}/_{x}\} \mid R'] \approx n[Q\{^{E}/_{x}\} \mid R']$ and this allows us to conclude that up to context we are still in $\mathcal{R}$.

Suppose that $P \cong^{\mathsf{e}\,\mathsf{o}}_{\mathsf{p}} Q$ and that $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\}$. Now we want to show that $!(x).P \cong^{\mathsf{e}}_{\mathsf{p}} !(x).Q$. Reasoning as before, we prove that for all n, R closed it holds that $n[!(x).P \mid R] \approx n[!(x).Q \mid R]$. In particular, we prove that the relation

$$\mathcal{R} = \{(n[!(x).P \mid R], n[!(x).Q \mid R)] \ : \\ P \cong^{\mathsf{e}\,\mathsf{o}}_{\mathsf{p}} Q,\ \mathrm{fn}(P) \cup \mathrm{fn}(Q) \subseteq \{x\},\ \text{for all } n, R \text{ closed}\}^{=} \cup \approx$$

is a bisimulation up to context and up to $(\gtrsim, \approx)$. The most interesting case is when $n[!(x).P \mid R] \xrightarrow{\tau} n[(\boldsymbol{\nu}\tilde{r})(P\{^{E}/_{x}\} \mid !(x).P \mid R)] \equiv (\boldsymbol{\nu}\tilde{r})n[P\{^{E}/_{x}\} \mid !(x).P] \mid R'$, where $n \notin \tilde{r}$ and $\tilde{r} \cap \mathrm{fn}(P) = \emptyset$. Observe that R sends the message E and resumes as R'. So we have a matching transition $n[!(x).Q \mid R] \xrightarrow{\tau} \equiv (\boldsymbol{\nu}\tilde{r})n[Q\{^{E}/_{x}\} \mid !(x).Q \mid R']$, where $\tilde{r} \cap \mathrm{fn}(Q) = \emptyset$. By construction of $\mathcal{R}$ we have $n[P\{^{E}/_{x}\} \mid !(x).P \mid R'] \ \mathcal{R}\ n[P\{^{E}/_{x}\} \mid !(x).Q \mid R']$. Since $P \cong^{\mathsf{e}\,\mathsf{o}}_{\mathsf{p}} Q$, it holds that $P\{^{E}/_{x}\} \cong^{\mathsf{e}}_{\mathsf{p}} Q\{^{E}/_{x}\}$. The characterisation of $\cong^{\mathsf{e}}_{\mathsf{p}}$ guarantees that $n[P\{^{E}/_{x}\} \mid !(x).Q \mid R'] \approx n[Q\{^{E}/_{x}\} \mid !(x).Q \mid R']$. Since bisimilarity is closed under restriction we have $(\boldsymbol{\nu}\tilde{r})n[P\{^{E}/_{x}\} \mid !(x).Q \mid R'] \approx (\boldsymbol{\nu}\tilde{r})n[Q\{^{E}/_{x}\} \mid !(x).Q \mid R']$. This allows us to conclude that up to context (we factor out the context $(\boldsymbol{\nu}\tilde{r})(-)$) and up to $(\gtrsim, \approx)$ we are still in $\mathcal{R}$. □

Corollary 6.4 *In the message-passing calculus, the relations $\mathcal{S}^{\mathsf{o}}$ and $\cong^{\mathsf{o}}_{\mathsf{p}}$ coincide.*

A crucial aspect of working with systems deserves to be pointed out. Bisimilarity is defined over systems, and as such it cannot directly observe the exercise of communications capabilities (apart from internal communications). This allow us to avoid any special treatment for asynchronous communication. More than that, we can easily extend our results to a calculus equipped with *synchronous* communication (e.g., $\langle E \rangle.P$).

7. ALGEBRAIC PROPERTIES

In this section we prove a collection of algebraic laws using our bisimulation proof methods. Then, we prove the correctness of a protocol for controlling access through

a *firewall*, first proposed in [Cardelli and Gordon 2000].

Laws on systems. We briefly comment on the laws of Theorem 7.1. We recall that M, N range over systems and P, Q, R over processes. 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 7.1

(1) $(\boldsymbol{\nu} n)n[\langle W\rangle \mid (x).Q \mid M] \;\cong_{\mathsf{p}}\; (\boldsymbol{\nu} n)n[Q\{^{W}/_{x}\} \mid M] \quad \textit{if } n \notin \mathrm{fn}(M)$

(2) $(\boldsymbol{\nu} n)n[\langle W\rangle \mid (x).Q \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j] \;\cong_{\mathsf{p}}\; (\boldsymbol{\nu} n)n[Q\{^{W}/_{x}\} \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j]$

(3) $(\boldsymbol{\nu} n)n[P] \;\cong_{\mathsf{p}}\; \mathbf{0} \quad \textit{if } n \notin \mathrm{fn}(P)$

(4) $(\boldsymbol{\nu} n)((\boldsymbol{\nu} m)m[\mathtt{in}\_n.P] \mid n[M]) \;\cong_{\mathsf{p}}\; (\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)m[P] \mid M] \quad \textit{if } n \notin \mathrm{fn}(M)$

(5) $(\boldsymbol{\nu} m, n)(m[\mathtt{in}\_n.P] \mid n[\prod_{j\in J} \mathtt{open}\_k_j.R_j]) \;\cong_{\mathsf{p}}$
$(\boldsymbol{\nu} m, n)n[m[P] \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j]$

(6) $(\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)m[\mathtt{out}\_n.P] \mid M] \;\cong_{\mathsf{p}}\; (\boldsymbol{\nu} n)((\boldsymbol{\nu} m)m[P] \mid n[M]) \quad \textit{if } n \notin \mathrm{fn}(M)$

(7) $(\boldsymbol{\nu} n)n[m[\mathtt{out}\_n.P] \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j] \;\cong_{\mathsf{p}}\; (\boldsymbol{\nu} n)(m[P] \mid n[\prod_{j\in J} \mathtt{open}\_k_j.R_j])$
if $m \neq k_j$, *for* $j \in J$

(8) $n[(\boldsymbol{\nu} m)(\mathtt{open}\_m.P \mid m[N]) \mid Q] \;\cong_{\mathsf{p}}\; n[(\boldsymbol{\nu} m)(P \mid N) \mid Q]$
if $Q \equiv M \mid \prod_{j\in J}(x_j).R_j$ *and* $m \notin \mathrm{fn}(N)$

(9) $(\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)(\mathtt{open}\_m.P \mid m[Q]) \mid R] \;\cong_{\mathsf{p}}\; (\boldsymbol{\nu} n)n[(\boldsymbol{\nu} m)(P \mid Q) \mid R]$
if $R \equiv \prod_{i\in I}(x_i).S_i \mid \prod_{j\in J} \mathtt{open}\_k_j.R_j$ *and* $m, n \notin \mathrm{fn}(Q)$.

Proof To prove the laws above, except (3) and (9), we exhibit a bisimulation that relates them: the results will follow from Theorem 5.11. In all cases the bisimulation follows a similar pattern:

$$\mathcal{S} = \{(\mathit{lhs}, \mathit{rhs})\}^{=} \cup \approx$$

where *lhs* and *rhs* denote respectively the left hand side and the right hand side of the equation, parametrised over names, processes and systems. For proving the laws (3) and (9) we show that the above $\mathcal{S}$ is a bisimulation up to context and up to structural congruence. We illustrate the proof of the law (3). Let $\mathcal{S} = \{((\boldsymbol{\nu} n)n[Q], \mathbf{0}) \mid \forall Q \text{ s.t. } n \notin \mathrm{fn}(Q)\}^{=}$. We show that $\mathcal{S}$ is a bisimulation up to context and up to structural congruence. The most delicate cases are those regarding the silent moves $*.\mathtt{enter}\_k$ and $*.\mathtt{exit}\_k$. For instance, if

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

then

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

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

Laws on processes. In Theorem 7.2 we give a collection of algebraic laws involving processes. Law 1 says that the opening of private ambients, possibly containing arbitrary messages, cannot be observed. Law 2 says that realising the same capability several times sequentially or in parallel has the same effect. Law 3 shows that processes prefixed by private capabilities are garbage. Law 4 says that two processes that differ only for having received different private capabilities cannot be distinguished. An instance of this law is

$$(\boldsymbol{\nu} n)\langle C_n\rangle \cong_{\mathsf{p}} (\boldsymbol{\nu} n)\langle D_n\rangle$$

for $C_n, D_n \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$. Notice that the above private outputs are not equivalent to $\mathbf{0}$ (use context $(x).a[\ ]$, for a fresh). Law 5 is the Mobile Ambient variant of the *asynchrony law* [Amadio et al. 1998] due to asynchronous communication. Finally, Law 6 equates two different outputs by adding a special process. While this law reminds us of Honda and Yoshida's *equator* [Honda and Yoshida 1995], it should be pointed out that Honda and Yoshida's equators hide the difference between two channels, whereas we equate messages.

Theorem 7.2 (Process Laws)

(1) $(\boldsymbol{\nu} n)(n[\prod_{j\in J}\langle E_j\rangle] \mid \mathtt{open}\_n.P) \cong_{\mathsf{p}} \prod_{j\in J}\langle E_j\rangle \mid P$ *if* $n \notin \mathrm{fn}(P, E_j)$ *for all* j;

(2) $C.P \cong_{\mathsf{p}} C.\mathbf{0} \mid P$ *if* P *is of the form* $C.\cdots.C.\mathbf{0}$;

(3) $(\boldsymbol{\nu} n)C_n.P \cong_{\mathsf{p}} \mathbf{0}$ *if* $C_n \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$;

(4) $(\boldsymbol{\nu} n)P\{^{C_n}/_x\} \cong_{\mathsf{p}} (\boldsymbol{\nu} n)P\{^{D_n}/_x\}$ *if* $C_n, D_n \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$, $\mathrm{fv}(P) \subseteq \{x\}$, *and* $n \notin \mathrm{fn}(P)$;

(5) $(x).\langle x\rangle \cong_{\mathsf{p}} \mathbf{0}$;

(6) $\langle E\rangle \mid \mathsf{Eq}(E, F) \cong_{\mathsf{p}} \langle F\rangle \mid \mathsf{Eq}(E, F)$ *where* $\mathsf{Eq}(E, F) \stackrel{\mathsf{def}}{=} !(x).\langle E\rangle \mid !(x).\langle F\rangle$.

Proof By Theorems 5.2 and 5.3, it suffices to show that

$$k[lhs \mid R] \approx k[rhs \mid R]$$

for all k and R, where lhs and rhs denote the left hand side, right hand side, of each law. In all cases, except 2 and 4, this can be proved by showing that the relation

$$\mathcal{R} = \{(k[lhs \mid R], k[rhs \mid R]) : \text{ for all } k \text{ and } R\}^{=} \cup \mathcal{I}$$

is a bisimulation up to context and up to $(\gtrsim, \approx)$, where $\mathcal{I}$ represent the identity relation over systems.

The candidate bisimulation for law (2) is

$$\mathcal{R} = \{(k[C.\underbrace{C.\cdots.C}_{n \text{ times}}.\mathbf{0} \mid R], k[C.\mathbf{0} \mid \underbrace{C.\cdots.C}_{n \text{ times}}.\mathbf{0} \mid R]) : \text{for all } n \geq 0, \text{ and for all } k \text{ and } R\}^{=} \cup \mathcal{I} .$$

In Law 4, the equality to prove is $k[(\boldsymbol{\nu} n)P\{^{C_n}/_x\} \mid R] \approx k[(\boldsymbol{\nu} n)P\{^{D_n}/_x\} \mid R]$, for all k and R. This can be proved by showing that the relation

$$\mathcal{R} = \{((\boldsymbol{\nu} n)M\{^{C_n}/_x\}, (\boldsymbol{\nu} n)M\{^{D_n}/_x\}) : \mathrm{fv}(M) \subseteq \{x\} \text{ and } n \notin \mathrm{fn}(M)\}^{=}$$

is a bisimulation. Notice that, as R is closed, up to α-conversion (to avoid name-capture), we have $k[(\boldsymbol{\nu} n)P\sigma \mid R] \equiv (\boldsymbol{\nu} n)k[P \mid R]\sigma$. □

On stuttering. In [Sangiorgi 2001] it is argued that barbed equivalences are insensitive to *stuttering* phenomena, originated by processes that may repeatedly enter and exit an ambient. Using a sum operator *à la* CCS, the next example conveys some intuitions about stuttering. The systems

$$M = m[\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.R] \qquad \text{and} \qquad N = m[\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.R + \mathtt{in}\_n.R]$$

are indeed reduction barbed congruent. To see why the extra summand of N does not affect its behaviour, consider a reduction produced by this summand:

$$N \mid n[S] \;\rightarrowtriangle\; n[S \mid m[R]] \;.$$

The process M can match it using three reductions:

$$M \mid n[S] \;\rightarrowtriangle\; n[S \mid m[\mathtt{out}\_n.\mathtt{in}\_n.R]] \;\rightarrowtriangle\; n[S] \mid m[\mathtt{in}\_n.R] \;\rightarrowtriangle\; n[S \mid m[R]] \;.$$

The crucial point is that the exercise of the capability $\mathtt{in}\_n$ is matched by the exercise of three capabilities, $\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n$. Although it might seem that our bisimilarity matches each action with only one action (possibly preceded and/or followed by τ transitions), our bisimilarity is actually insensitive to stuttering. To illustrate why, we use a variant of the example above that does not rely on internal sum. Replication in the processes P and Q below implements a loop with an alternation between input/output and the path $\mathtt{in}\_n.\mathtt{out}\_n$. There is a 1-cycle shift, however, between the two loops. Stuttering makes the shift irrelevant.

Proposition 7.3 *The processes P and Q defined as*

$$P = (\boldsymbol{\nu} l)(\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])$$
$$Q = (\boldsymbol{\nu} l)(\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])$$

are reduction barbed congruent over processes.

Proof Let

$$\begin{aligned}\mathcal{R} = \{(\; & k[O \mid (\boldsymbol{\nu} l)(\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])] \;, \\ & k[O \mid (\boldsymbol{\nu} l)(\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])] \;) \\ & \qquad \mid \; k \text{ and } O \text{ are arbitrary}\}^{=} \;\cup\; \mathcal{I} \;.\end{aligned}$$

where $\mathcal{I}$ is the identity relation between systems. The relation $\mathcal{R}$ is a bisimulation up to context and up to structural congruence. We detail the most interesting case, where the exercise of one capability must be matched by the exercise of three capabilities. Suppose $M \;\mathcal{R}\; N$, with

$$M = k[O \mid (\boldsymbol{\nu} l)(\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])]$$

and

$$N = k[O \mid (\boldsymbol{\nu} l)(\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])] \;.$$

Also suppose that

$$M \xrightarrow{k.\mathtt{enter}\_n} n[\circ \mid k[O \mid (\boldsymbol{\nu} l)(l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])]] \;.$$

Then N can perform the following sequence of transitions:

$$\begin{aligned} N &\xrightarrow{k.\mathtt{enter}\_n} n[\circ \mid k[O \mid (\boldsymbol{\nu} l)(\mathtt{out}\_n.\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])]] \\ &\xrightarrow{\tau} n[\,\circ\,] \mid k[O \mid (\boldsymbol{\nu} l)(\mathtt{in}\_n.l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])] \\ &\xrightarrow{\tau} n[\circ \mid k[O \mid (\boldsymbol{\nu} l)(l[] \mid !\mathtt{open}\_l.\mathtt{out}\_n.\mathtt{in}\_n.l[])]] \ . \end{aligned}$$

For all instantiations of $\circ$ we can factor out the context $n[\circ \mid -]$ and up to context we are still in $\mathcal{R}$. □

The proof above clearly shows that the exercise of the sequence of three capabilities $\mathtt{in}\_n.\mathtt{out}\_n.\mathtt{in}\_n$ is needed to match the capability $\mathtt{in}\_n$ give rise to a $k.\mathtt{enter}\_n$ action followed by two internal transitions. The internal actions are subsequently absorbed by the weak formulation of the equivalence.

Crossing a firewall. A protocol is discussed in [Cardelli and Gordon 2000] for controlling access through a firewall. The ambient w represents the firewall; the ambient m, a trusted agent containing a process Q that is supposed to cross the firewall. The firewall ambient sends into the agent a pilot ambient k with the capability $\mathtt{in}\_w$ 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:

$$\begin{aligned} AG &\stackrel{\mathsf{def}}{=} m[\mathtt{open}\_k.(x).x.Q] \\ FW &\stackrel{\mathsf{def}}{=} (\boldsymbol{\nu} w)w[\mathtt{open}\_m.P \mid k[\mathtt{out}\_w.\mathtt{in}\_m.\langle \mathtt{in}\_w \rangle]] \end{aligned}$$

The correctness (of a mild variant) of the protocol above is shown in [Cardelli and Gordon 2000] for may-testing [De Nicola and Hennessy 1984] proving that

$$(\boldsymbol{\nu} m, k)(AG \mid FW) \ \cong_{\mathsf{p}} \ (\boldsymbol{\nu} w)w[Q \mid P]$$

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 reasoning. 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 $\cong_{\mathsf{p}}$ is insensitive to all these τ-actions. The result follows from the algebraic laws of Theorem 7.1 and the following two laws:

Lemma 7.4 *Let P, Q, and R be processes. Then*

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

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

Proof By exhibiting the appropriate bisimulation. In both cases, the bisimulations we exhibit have a similar form:

$$\mathcal{S} = \{(lhs, rhs)\}^{=} \cup \mathcal{I}$$

where *lhs* and *rhs* denote respectively the left hand side and the right hand side of the equation. □

Theorem 7.5 *If* $w \notin \mathrm{fn}(Q)$ *and* $\{m,k\} \cap (\mathrm{fn}(P) \cup \mathrm{fn}(Q)) = \emptyset$, *then*

$$(\boldsymbol{\nu} m,k)(AG \mid FW) \cong_{\mathsf{p}} (\boldsymbol{\nu} w)w[Q \mid P] \ .$$

Proof It suffices to apply the algebraic laws of Theorem 7.1 and Lemma 7.4. More precisely, we apply Law (7) of Theorem 7.1, Law (1) of Lemma 7.4, Law (9) of Theorem 7.1, Law (2) of Lemma 7.4, and Laws (5) and (9) of Theorem 7.1. □

8. RELATED WORK

In this paper we study the behavioural theory of Cardelli and Gordon's Mobile Ambients.

A theory of Morris-style preserved by system contexts equivalence for Mobile Ambients has been developed by Gordon and Cardelli in [Gordon and Cardelli 2002]. In particular, they prove the perfect firewall equation, a simplified variant of Law 1 of Theorem 7.2, and the protocol to cross a firewall discussed above. Although their theory benefits of a context lemma which allows to consider only contexts of a particular form, we believe that the verification of algebraic laws still remains quite complicated. It should be noticed that all the laws proved in [Gordon and Cardelli 2002] relate processes that engage only in limited interactions with their context.

Higher-order LTSs for Mobile Ambients can be found in [Cardelli and Gordon 1996; Gordon and Cardelli 2002; Vigliotti 1999; Ferrari et al. 2001]. But we are not aware of any form of bisimilarity defined using these LTSs. Sewell [2002] addresses the problem of uniformly deriving LTSs and bisimulation congruences from the reduction rules of a calculus. The transitions generated for a fragment of Mobile Ambients require the same universal quantification on the content of the interacting ambient as ours. Sewell's techniques only apply to strong equivalences. A simple first-order LTS for MA without restriction and replication is proposed by Sangiorgi in [Sangiorgi 2001], and later extended to replication by Hirschkoff, Lozes and Sangiorgi in [Hirschkoff et al. 2002]. Using this LTS the authors defines an *intensional* bisimilarity for MA that separates terms on the basis of their internal structure.

Recently, Jensen and Milner [2004], based on previous work by Leifer and Milner [Leifer and Milner 2000], derived an LTS for Mobile Ambients starting from an encoding of Mobile Ambients into Bigraphs. They built a standard bisimilarity on top of this LTS. Their LTS is strikingly similar to ours, and, if we confine our attention to Mobile Ambients without restriction, we conjecture that their bisimilarity coincides with ours. In general, however, their equivalence is sensitive to the movement of secret ambients unlike ours. As a consequence their bisimilarity does not satisfy equations involving unobservable migrations, like the perfect firewall equation.

Our work is the natural continuation of [Merro and Hennessy 2002; 2005] where an LTS and a labelled characterisation of reduction barbed congruence are given for a more handful variant of Levi and Sangiorgi's Safe Ambients, called SAP. The main differences with respect to [Merro and Hennessy 2002] are the following:

— unlike MA, the calculus SAP is equipped with co-capabilities and passwords; both features are essential to prove the characterisation result in SAP. On the other hand in MA, (i) the asynchrony nature of ambient migration, and (ii) the non-observability of secret ambients, make the behavioural theory much more involved;

— our env-actions, unlike those in [Merro and Hennessy 2002], are truly late, because they do not mention the process provided by the environment. We add such process *later*, when playing the bisimulation game. This highlights how the contribution of the environment is limited to providing an ambient so that interaction can happen; the content of the ambient is irrelevant when building the matching actions. Following our experience, this approach has then been adopted in [Merro and Hennessy 2005];

— 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 in SAP 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. By contrast, in MA the movements of an ambient whose name is private cannot be observed. This is why the perfect firewall equation holds;

— we enhance our proof methods with up-to expansion and up-to context proof techniques.

Note that, although the labelled bisimilarity contains a universal quantification over processes, it is an effective proof technique, especially when coupled with the up-to expansion and up-to context proof techniques. The best illustration of this are the proofs of the algebraic laws of Section 7: in all cases, the definition and the verification of the candidate bisimulations are very simple. This should be contrasted with the proofs based on contextual reasoning developed in [Levi and Sangiorgi 2003; Gordon and Cardelli 2002].

Our work builds on Sangiorgi's seminal research on bisimulations for HOπ [Sangiorgi 1996a], and inherits from this work the idea that transitions in the LTS must correspond to interaction with some context. Sangiorgi achieves this by explicitly quantifying over all possible interacting contexts (in the so-called *contextual bisimulation*). The structure of contextual bisimulation as defined by Sangiorgi requires an LTS in *delay* style: τ-transitions are forbidden after the observable action. As a consequence, his contextual bisimulation for HOπ is sound but not complete with respect to reduction barbed congruence. Sangiorgi then refined his approach by replacing the interacting context by a term behaving as a fresh pointer; the pointer can then be exploited by an arbitrary context to interact with the term being tested. Sangiorgi's approach has been recently improved by Jeffrey and Rathke [2005], who obtained a full-abstraction result for HOπ: the pointers are now represented by special markers (instead of terms) and are integrated directly in the LTS. It not clear if Jeffrey and Rathke's approach to pinpoint the interactions can be extended to the complicated operational semantics of MA.

Higher-order features have been at the core of research on mobile processes. We point out that, apart from [Merro and Hennessy 2002], other forms of bisimilarity for higher-order distributed calculi, such as Distributed π-calculus [Hennessy and Riely 1998], Seal [Vitek and Castagna 1999], a Calculus for Mobile Resources [Godskesen et al. 2002], NBA [Bugliesi et al. 2005], SafeDpi [Hennessy et al. 2003], Homer [T. Hildebrandt 2004], and the Kell calculus [Schmitt and Stefani 2004] can be found in [Hennessy et al. 2004; Castagna et al. 2005; Godskesen et al. 2002; Bugliesi et al. 2005; Hennessy et al. 2003; Schmitt and Stefani 2004], but only [Hennessy et al. 2004; Godskesen et al. 2002; Bugliesi et al. 2005; Hennessy et al. 2003; Schmitt and Stefani 2004] prove labelled characterisations of a contextually-defined program equivalence (in [T. Hildebrandt 2004] completeness holds only for the strong equivalence). Unyapoth and Sewell [2001] take a different, more intensional approach to define an equivalence for Nomadic Pict. To establish correctness of a particular protocol, they identify a novel equivalence based on coupled simulation but tailored to accommodate code migration. Although this equivalence has many interesting properties, in particular it is a congruence, is not shown to coincide with any independent contextually defined equivalence.

APPENDIX

A. PROOFS FROM SECTION 2

Proof of Proposition 2.2 The LTS is defined over processes extended with the process variable $\circ$. As such, it sends a process over the extended syntax to a process over the extended syntax.

Let M be a system over the extended syntax, such that $M \xrightarrow{\alpha} P$. We must show that P is a system over the extended syntax. The env-action must have been derived from a pre-action of the form $M \xrightarrow{\pi} (\boldsymbol{\nu}\tilde{m})\langle P_1\rangle P_2$. If $\alpha = \tau$, then the result follows because the class of systems is closed under reductions, and a τ-transition can be assimilated to a reduction, as shown in Theorem 2.5. If $\alpha \in \{k.\mathtt{exit}\_n, k.\mathtt{open}\_n, *.\mathtt{exit}\_n\}$, then the rules (Exit), (Open), (Exit Shh) guarantee that the outcome is a system. In the remaining cases, observe that the env-action must have been derived from a pre-action of the form $M \xrightarrow{\pi} (\boldsymbol{\nu}\tilde{m})\langle P_1\rangle P_2$. Since M is a system, the P_2 component of the concretion must be a system, and the result follows. □

Proof of Lemma 2.4 We prove a stronger result stating that structural congruence preserves all the labels of the LTS (including pre-actions):

(1) if $P \equiv Q$ and $P \xrightarrow{\ell} P'$ for $\ell \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$, then there exists Q' such that $Q \xrightarrow{\ell} Q'$ and $P' \equiv Q'$;

(2) if $P \equiv Q$ and $P \xrightarrow{\ell} (\boldsymbol{\nu}\tilde{n})\langle P_1\rangle P_2$ for $\ell \in \{\mathtt{enter}\_n, \mathtt{exit}\_n, \mathtt{amb}\_n\}$, then there exist Q_1, Q_2 such that $Q \xrightarrow{\ell} (\boldsymbol{\nu}\tilde{n})\langle Q_1\rangle Q_2$, with $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$;

(3) if $P \equiv Q$ and $P \xrightarrow{\alpha} P'$, then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \equiv Q'$.

The three statements are proved by induction on the derivation of $P \equiv P'$, and by case analysis on the transition ℓ. As the $\equiv$ relation is symmetric, we prove directly

the symmetric case, where Q realises the ℓ action. It is instructive to detail the cases that deal with the (Struct Repl Par) and (Struct Res Par) rules.

1. **Case** (Struct Repl Par). Let $P = !C.R$ and $Q = C.R \mid !C.R$. In this case, the label ℓ must be a pre-action π. Suppose $!C.R \xrightarrow{\pi} R \mid !C.R$ (rule (πRepl Pfx)). We derive $C.R \mid !C.R \xrightarrow{\pi} R \mid !C.R$ using the rules (π Pfx) and (π Par). For the symmetric case, we must consider two subcases. If $C.R \mid !C.R \xrightarrow{\pi} R \mid !C.R$ (rules (π Pfx) and (π) Par), then we derive $!C.R \xrightarrow{\pi} R \mid !C.R$ by the rule (π Repl Pfx). If $C.R \mid !C.R \xrightarrow{\pi} C.R \mid R \mid !C.R$ (rules (π Repl Pfx) and (π Par)), then $!C.R \xrightarrow{\pi} R \mid !C.R$ by the rule (π Repl Pfx), and $C.R \mid R \mid !C.R \equiv R \mid !C.R$ by application of rules (Struct Repl Par), (Struct Par Comm), (Struct Par Assoc).
Observe that this structural rule must be considered only in the proof of the first statement, because if $P = !C.R$ and $Q = C.R \mid !C.R$, then the label ℓ must be a pre-action $\ell \in \{\mathtt{in}\_n, \mathtt{out}\_n, \mathtt{open}\_n\}$.

1. **Case** (Struct Res Par). Let $P = R \mid (\boldsymbol{\nu} n)S$ and $Q = (\boldsymbol{\nu} n)(R \mid S)$, where $n \notin \mathrm{fn}(R)$. If $P \xrightarrow{\pi} P'$, then either $R \xrightarrow{\pi} R'$ and $P' = R' \mid (\boldsymbol{\nu} n)S$, or $(\boldsymbol{\nu} n)S \xrightarrow{\pi} S'$ and $P' = R \mid S'$. In the first case, since $n \notin \mathrm{fn}(R)$ we have $n \notin \mathrm{fn}(\pi)$, and we can deduce $(\boldsymbol{\nu} n)(R \mid S) \xrightarrow{\pi} (\boldsymbol{\nu} n)(R' \mid S) \equiv P'$. A derivation for the latter can simply be obtained by switching the (π Res) and the (π Par) rules in the derivation of $P \xrightarrow{\pi} P'$. The cases when $Q \xrightarrow{\pi} Q'$ are similar.

2. **Case** (Struct Res Par). We detail the case when $P = R \mid (\boldsymbol{\nu} n)S \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu} \tilde{n})\langle P_1 \rangle P_2$. The $\mathtt{enter}\_n$ action is either realised by R and the outcome is $(\boldsymbol{\nu} \tilde{n}_1)\langle R_1 \rangle R_2 \mid (\boldsymbol{\nu} n)S$, or by $(\boldsymbol{\nu} n)S$ and the outcome is $(\boldsymbol{\nu} \tilde{n}_2)\langle S_1 \rangle R \mid S_2$. In the first case, since $n \notin \mathrm{fn}(R)$, we can derive the transition $Q = (\boldsymbol{\nu} n)(R \mid S) \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu} \tilde{n}_1)\langle R_1 \rangle (\boldsymbol{\nu} n)(R_2 \mid S)$, and it holds that $R_1 \equiv R_1$ and $R_2 \mid (\boldsymbol{\nu} n)S \equiv (\boldsymbol{\nu} n)(R_2 \mid S)$ because $n \notin \mathrm{fn}(R)$ implies $n \notin \mathrm{fn}(R_2)$. In the latter, if $n \in \mathrm{fn}(S_1)$, then the outcome is $(\boldsymbol{\nu} \tilde{n}_2)\langle S_1 \rangle R \mid S_2$ with $n \in \tilde{n}_2$ and the result follows. Instead, if $n \notin \mathrm{fn}(S_1)$, then the outcome is $(\boldsymbol{\nu} \tilde{n}_2)\langle S_1 \rangle (\boldsymbol{\nu} n)(R \mid S_2)$ and the result follows because $R \mid (\boldsymbol{\nu} n)S_2 \equiv (\boldsymbol{\nu} n)(R \mid S_2)$.

3. **Case** (Struct Res Par). We detail the case when the transition $P \xrightarrow{\tau} P'$ is derived from $R \xrightarrow{\mathtt{enter}\_m} (\boldsymbol{\nu} \tilde{r})\langle R_1 \rangle R_2$ and $(\boldsymbol{\nu} n)S \xrightarrow{\mathtt{amb}\_m} (\boldsymbol{\nu} \tilde{s})\langle S_1 \rangle S_2$. Observe that $m \neq n$. We distinguish two subcases. If $n \in \tilde{s}$, then $P' = (\boldsymbol{\nu} \tilde{r})(\boldsymbol{\nu} \tilde{s})(m[R_1 \mid S_1] \mid R_2 \mid S_2)$, and $S \xrightarrow{\mathtt{amb}\_m} (\boldsymbol{\nu} \tilde{s} \setminus n)\langle S_1 \rangle S_2$. By ($\tau$ Enter) and by (τ Res) we derive $(\boldsymbol{\nu} n)(R \mid S) \xrightarrow{\tau} (\boldsymbol{\nu} n)(\boldsymbol{\nu} \tilde{r})(\boldsymbol{\nu} \tilde{s} \setminus n)(m[R_1 \mid S_1] \mid R_2 \mid S_2)$, which is equal to P_1 modulo the order of names in the top-level restriction. If $n \notin \tilde{s}$, then $P' = (\boldsymbol{\nu} \tilde{r})(\boldsymbol{\nu} \tilde{s} \setminus n)(m[R_1 \mid S_1] \mid R_2 \mid (\boldsymbol{\nu} n)S_2)$, and $S \xrightarrow{\mathtt{amb}\_m} (\boldsymbol{\nu} \tilde{s})\langle S_1 \rangle S_2$, and by ($\tau$ Enter) and by (τ Res) we derive $(\boldsymbol{\nu} n)(R \mid S) \xrightarrow{\tau} (\boldsymbol{\nu} n)(\boldsymbol{\nu} \tilde{r})(\boldsymbol{\nu} \tilde{s})(m[R_1 \mid S_1] \mid R_2 \mid S_2)$, which is structurally congruent to P' by successive applications of rule (Struct Res Par).
More in general, we point out that complementary pre-actions must refer to the same name. Since the pre-action realised by R cannot mention the name n, the pre-action realised by S cannot either; this allows to derive the required transitions independently of the position of the restriction $(\boldsymbol{\nu} n)$. □

Proof of Theorem 2.5

Part 1. By induction on the derivation of $P \xrightarrow{\tau} P'$. Remark that τ-transitions can only be generated by the rules in Table V.

(τ Enter) We know that $P \xrightarrow{\texttt{enter\_}n} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle P_2$, and $Q \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle Q_2$. From Lemma 2.3 we deduce that $P \equiv (\boldsymbol{\nu}\tilde{s})(k[\texttt{in\_}n.P_3 \mid P_4] \mid P_2)$ for some names $\tilde{s}$ and processes P_3 and P_4. Let $\tilde{r}$ be the names in $\tilde{s}$ that do not appear in $\tilde{p}$; these names are not free in P_2. We can then write $P \equiv (\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{in\_}n.P_3 \mid P_4] \mid P_2)$. Also, $P_1 \equiv (\boldsymbol{\nu}\tilde{r})k[P_3 \mid P_4]$. From Lemma 2.3 we deduce that $Q \equiv (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2)$. Then,

$$\begin{array}{rcl} P \mid Q & \equiv & (\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{in\_}n.P_3 \mid P_4] \mid P_2) \mid (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})(\boldsymbol{\nu}\tilde{q})(k[\texttt{in\_}n.P_3 \mid P_4] \mid n[Q_1] \mid P_2 \mid Q_2) \\ & \rightarrowtriangle & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})(\boldsymbol{\nu}\tilde{q})(n[k[P_3 \mid P_4] \mid Q_1] \mid P_2 \mid Q_2) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(n[P_1 \mid Q_1] \mid P_2 \mid Q_2) \end{array}$$

as desired.

(τ Exit) We know that $P \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{p})\langle k[P_1]\rangle P_2$. From Lemma 2.3 we deduce that $P \equiv (\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{out\_}n.P_3 \mid P_4] \mid P_2)$, where $P_1 \equiv P_3 \mid P_4$, for some processes P_3, P_4 and names $\tilde{r}$. Then,

$$\begin{array}{rcl} n[P] & \equiv & n[(\boldsymbol{\nu}\tilde{p})((\boldsymbol{\nu}\tilde{r})k[\texttt{out\_}n.P_3 \mid P_4] \mid P_2)] \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})n[k[\texttt{out\_}n.P_3 \mid P_4] \mid P_2] \\ & \rightarrowtriangle & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{r})(n[P_2] \mid k[P_3 \mid P_4]) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(n[P_2] \mid (\boldsymbol{\nu}\tilde{r})k[P_3 \mid P_4]) \end{array}$$

as desired.

(τ Open) We know that $P \xrightarrow{\texttt{open\_}n} P_1$ and $Q \xrightarrow{\texttt{amb\_}n} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle Q_2$. From Lemma 2.3 we deduce that $P \equiv (\boldsymbol{\nu}\tilde{p})(\texttt{open\_}n.P_2 \mid P_3)$, where $P_1 \equiv (\boldsymbol{\nu}\tilde{p})(P_2 \mid P_3)$ for some processes P_2, P_3. Lemma 2.3 also guarantees that $Q \equiv (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2)$. Then,

$$\begin{array}{rcl} P \mid Q & \equiv & (\boldsymbol{\nu}\tilde{p})(\texttt{open\_}n.P_2 \mid P_3) \mid (\boldsymbol{\nu}\tilde{q})(n[Q_1] \mid Q_2) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(\texttt{open\_}n.P_2 \mid n[Q_1] \mid P_3 \mid Q_2) \\ & \rightarrowtriangle & (\boldsymbol{\nu}\tilde{p})(\boldsymbol{\nu}\tilde{q})(P_2 \mid Q_1 \mid P_3 \mid Q_2) \\ & \equiv & (\boldsymbol{\nu}\tilde{p})(P_2 \mid P_3) \mid (\boldsymbol{\nu}\tilde{q})(Q_1 \mid Q_2) \end{array}$$

as desired.

The other cases follows straightforwardly from the congruence rules of the reduction relation.

Part 2. By induction on the derivation of $P \rightarrowtriangle Q$. There are three base cases.

(Red In) We know that

$$n[\texttt{in\_}m.P \mid Q] \mid m[R] \;\rightarrowtriangle\; m[n[P \mid Q] \mid R]\ .$$

The derivation below is valid.

$$\dfrac{\dfrac{\dfrac{}{\mathtt{in}\_m.P \xrightarrow{\mathtt{in}\_m} P}}{\mathtt{in}\_m.P \mid Q \xrightarrow{\mathtt{in}\_m} P \mid Q}}{n[\mathtt{in}\_m.P \mid Q] \xrightarrow{\mathtt{enter}\_m} \langle n[P \mid Q]\rangle \mathbf{0}} \qquad m[R] \xrightarrow{\mathtt{amb}\_m} \langle R\rangle \mathbf{0}}{n[\mathtt{in}\_m.P \mid Q] \mid m[R] \xrightarrow{\tau} m[n[P \mid Q] \mid R] \mid \mathbf{0} \mid \mathbf{0}}$$

and $m[n[P \mid Q] \mid R] \mid \mathbf{0} \mid \mathbf{0} \equiv m[n[P \mid Q] \mid R]$.

(Red Out) We know that

$$m[n[\mathtt{out}\_m.P \mid Q] \mid R] \rightarrowtriangle n[P \mid Q] \mid m[R]$$

The derivation below is valid.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathtt{out}\_m.P \xrightarrow{\mathtt{out}\_m} P}}{\mathtt{out}\_m.P \mid Q \xrightarrow{\mathtt{out}\_m} P \mid Q}}{n[\mathtt{out}\_m.P \mid Q] \xrightarrow{\mathtt{exit}\_m} \langle n[P \mid Q]\rangle \mathbf{0}}}{n[\mathtt{out}\_m.P \mid Q] \mid R \xrightarrow{\mathtt{exit}\_m} \langle n[P \mid Q]\rangle R}}{m[n[\mathtt{out}\_m.P \mid Q] \mid R] \xrightarrow{\tau} n[P \mid Q] \mid m[R]}$$

(Red Open) We know that

$$\mathtt{open}\_n.P \mid n[Q] \rightarrowtriangle P \mid Q$$

The derivation below is valid

$$\dfrac{\mathtt{open}\_n.P \xrightarrow{\mathtt{open}\_n} P \qquad n[Q] \xrightarrow{\mathtt{amb}\_n} \langle Q\rangle \mathbf{0}}{\mathtt{open}\_n.P \mid n[Q] \xrightarrow{\tau} P \mid Q \mid \mathbf{0}}$$

and $P \mid Q \mid \mathbf{0} \equiv P \mid Q$.

For the inductive step, we must prove that:

— Rule (Red Struct). The induction hypothesis tells us that there exists R' such that $Q \xrightarrow{\tau} R' \equiv R$. Lemma 2.4 tells us that there exists R'' such that $P \xrightarrow{\tau} R''$ and $R'' \equiv R'$. The result follows from transitivity of $\equiv$.

— The reduction relation is preserved by static context: τ-transitions are preserved by static contexts too. □

B. PROOFS FROM SECTION 3

Proof of Theorem 3.3 Let $\mathcal{S}$ be the smallest relation such that:

(1) $\approx \subseteq \mathcal{S}$;
(2) if $M \mathcal{S} N$, then $(\boldsymbol{\nu} m)M \mathcal{S} (\boldsymbol{\nu} m)N$ for all names m;
(3) if $M \mathcal{S} N$, then $M \mid H \mathcal{S} N \mid H$ for all systems H;
(4) if $M \mathcal{S} N$, then $n[M \mid P] \mathcal{S} n[N \mid P]$ for all names n and processes P.

We prove that $\mathcal{S}$ is a bisimilarity up to $\equiv$[2], by induction on the definition of $\mathcal{S}$.

[2]The up-to $\equiv$ proof technique is introduced in Section 4.

Suppose that $M \mathcal{S} N$ **because** $M \approx N$**.** This case is straightforward.

Suppose that $(\boldsymbol{\nu} m)M \mathcal{S} (\boldsymbol{\nu} m)N$ **because** $M \mathcal{S} N$**.** Suppose $(\boldsymbol{\nu} m)M \xrightarrow{\alpha} O_1$. We perform a case analysis on α.

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

— $(\boldsymbol{\nu} m)M \xrightarrow{k.\mathtt{enter}\_n} O_1$.
Observe that this must have been derived from

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

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

— $(\boldsymbol{\nu} m)M \xrightarrow{k.\mathtt{exit}\_n} O_1$.
Observe that this must have been derived from

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

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

obtain $(\boldsymbol{\nu} m)N \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{exit}\_n} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds that $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

— $(\boldsymbol{\nu} m)M \xrightarrow{n.\overline{\mathtt{enter}}\_k} O_1$.
Observe that this must have been derived from

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

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

— $(\boldsymbol{\nu} m)M \xrightarrow{k.\mathtt{open}\_n} O_1$.
Observe that this must have been derived from

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

for some process M_1 and system M_2. Remark that this implies $m \neq n$ and $m \neq k$. As $M \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ then $M \xrightarrow{k.\mathtt{open}\_n} k[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)] = M'$. 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 $N \Rightarrow A \xrightarrow{k.\mathtt{open}\_n} B \Rightarrow N'$, and for all processes P it holds that $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{k.\mathtt{open}\_n} B$, 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 that $A \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. This implies $(\boldsymbol{\nu} m)A \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$, from which we can derive $(\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{open}\_n} C \equiv k[\circ \mid (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)] \equiv (\boldsymbol{\nu} m)k[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)] = (\boldsymbol{\nu} m)N'$. We obtain $(\boldsymbol{\nu} m)N \Rightarrow (\boldsymbol{\nu} m)A \xrightarrow{k.\mathtt{open}\_n} C \Rightarrow\equiv (\boldsymbol{\nu} m)N'$. Call $(\boldsymbol{\nu} m)N' = O_2$. We can conclude that for all processes P, it holds that $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

— $(\boldsymbol{\nu} m)M \xrightarrow{*.\mathtt{enter}_n} O_1$.

Observe that there are two possible derivations.

—Suppose:

$$\frac{M \xrightarrow{\mathtt{enter}_n} (\boldsymbol{\nu}\tilde{r})\langle m[M_1]\rangle M_2 \quad 2}{(\boldsymbol{\nu} m)M \xrightarrow{\mathtt{enter}_n} (\boldsymbol{\nu} m)(\boldsymbol{\nu}\tilde{r})\langle m[\quad \mathsf{M}_1.\rangle M}$$

N', A, B such that $N \Rightarrow A \xrightarrow{m.\mathtt{exit}\_n} B \Rightarrow N'$, and for all processes P it holds that $M' \bullet P \mathcal{S} N' \bullet P$. As $A \xrightarrow{m.\mathtt{exit}\_n} B$, 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 that $A \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$. This implies $(\boldsymbol{\nu}m)A \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})\langle m[N_1]\rangle N_2$, from which we can derive $(\boldsymbol{\nu}m)n[\circ \mid A] \xrightarrow{\tau} C \equiv (\boldsymbol{\nu}m)B = (\boldsymbol{\nu}m)(\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2] \mid m[N_1])$. We obtain $(\boldsymbol{\nu}m)n[\circ \mid N] \equiv n[\circ \mid (\boldsymbol{\nu}m)N] \Rightarrow n[\circ \mid (\boldsymbol{\nu}m)A] \xrightarrow{\tau} C \Rightarrow\equiv (\boldsymbol{\nu}m)N'$. Call $(\boldsymbol{\nu}m)N' = O_2$. We can conclude that for all processes P, it holds that $O_1 \bullet P \mathcal{S} O_2 \bullet P$ up to structural congruence, because $\mathcal{S}$ is closed under restriction.

—Suppose:

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

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

Suppose that $M \mid H \mathcal{S} N \mid H$ **because** $M \mathcal{S} N$**.** Suppose $M \mid H \xrightarrow{\alpha} O_1$. We perform a case analysis on α.

We first consider the cases when there is no interaction between M and H.

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

— $M \mid H \xrightarrow{\tau} O_1$, because $H \xrightarrow{\tau} H'$ and $O_1 \equiv M \mid H'$. Let $O_2 = N \mid H'$: it holds that $N \mid H \xrightarrow{\tau} O_2$, and $O_1 \mathcal{S} O_2$ because $M \mathcal{S} N$ and $\mathcal{S}$ is closed under parallel composition.

— $M \mid H \xrightarrow{k.\mathtt{enter}\_n} O_1$.

There are two possible derivations.

—Suppose:

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

for some process M_1 and system M_2. Remark that $k \notin \tilde{r}$. As $M \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $M \xrightarrow{k.\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{r})(n[k[M_1] \mid \circ] \mid M_2) = M'$. The

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

—Suppose:

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

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

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

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

— $M \mid H \xrightarrow{k.\mathtt{exit}\_n} O_1$.

There are two possible derivations.

—Suppose:

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

for some process M_1 and system M_2. Remark that $k \notin \tilde{r}$. As $M \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $M \xrightarrow{k.\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N', A, B such that $N \Rightarrow A \xrightarrow{k.\mathtt{exit}\_n} B \Rightarrow N'$, and for all processes P it holds that $M' \bullet P \mathcal{S} N' \bullet P$. Remark that $N' \equiv (\boldsymbol{\nu}\tilde{h})n[\circ \mid N_3] \mid N_4$, for some N_3, N_4. As $A \xrightarrow{k.\mathtt{exit}\_n} B$, 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 that $A \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2$. This implies $A \mid H \xrightarrow{\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})\langle k[N_1]\rangle N_2 \mid H$, from which we can derive $A \mid H \xrightarrow{k.\mathtt{exit}\_n} (\boldsymbol{\nu}\tilde{s})(n[\circ \mid N_2 \mid H] \mid k[N_1]) \equiv B \bullet (\circ \mid H)$. We obtain

$N \mid H \Rightarrow A \mid H \xrightarrow{k.\mathtt{exit}\_n} B \bullet (\circ \mid H) \Rightarrow\equiv N' \bullet (\circ \mid H)$. Call $N' \bullet (\circ \mid H) = O_2$. As for all processes P it holds that $M' \bullet P \mathcal{S} N' \bullet P$, we can conclude that for all processes Q, it holds that $O_1 \bullet Q \mathcal{S} O_2 \bullet Q$ 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:

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

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

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

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

— $M \mid H \xrightarrow{n.\overline{\mathtt{enter}}\_k} O_1$.

There are two possible derivations.

—Suppose:

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

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

—Suppose:

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

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

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

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

— $M \mid H \xrightarrow{k.\mathtt{open}\_n} O_1$.

There are two possible derivations.

—Suppose:

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

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

—Suppose:

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

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

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

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

— $M \mid H \xrightarrow{*.\texttt{enter\_}n} O_1$.

There are two possible derivations.

—Suppose:

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

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

—Suppose:

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

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

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

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

— $M \mid H \xrightarrow{*.\texttt{exit\_}n} O_1$.

There are two possible derivations.

—Suppose:

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

for some process M_1 and system M_2. Remark that $k \in \tilde{r}$. As $M \xrightarrow{\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})\langle k[M_1]\rangle M_2$ then $M \xrightarrow{*.\texttt{exit\_}n} (\boldsymbol{\nu}\tilde{r})(n[\circ \mid M_2] \mid k[M_1]) = M'$. The induction hypothesis then tells us that there exist systems N' such that $n[\circ \mid N] \Rightarrow N'$, and for all processes P it holds that $M' \bullet P \;\mathcal{S}\; N' \bullet P$. 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 $n[\circ N \mid H] \Rightarrow (\boldsymbol{\nu}\tilde{s})n[\circ \mid N_3 \mid H] \mid N_4$. Call $(\boldsymbol{\nu}\tilde{s})n[\circ \mid N_3 \mid H] \mid N_4 = O_2$. As for all processes P it holds that $M' \bullet P \;\mathcal{S}\; N' \bullet P$, we can conclude that for all processes Q, it holds that $O_1 \bullet Q \;\mathcal{S}\; O_2 \bullet Q$ 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:

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

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

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

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

Then, we consider the cases when there is interaction between M and H.

— $M \mid H \xrightarrow{\tau} O_1$, because

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

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

it also holds that $M' \bullet H_1 \ \mathcal{S} \ N' \bullet H_1$, and $O_1 \ \mathcal{S} \ O_2$ follows because $\mathcal{S}$ is closed under parallel composition and restriction.

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

— $M \mid H \xrightarrow{\tau} O_1$, because

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

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

Suppose that $n[M \mid P] \ \mathcal{S} \ n[N \mid P]$ **because** $M \ \mathcal{S} \ N$**.** Suppose $n[M \mid P] \xrightarrow{\alpha} O_1$. We perform a case analysis on α.

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

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

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

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

P, thus obtaining the transition $n[N \mid P] \Rightarrow N' \bullet P$. As for all processes Q, it holds that $M' \bullet Q \mathcal{S} N' \bullet Q$, 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.

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

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

—$M \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{r})\langle M_1\rangle M_2$ and $P \xrightarrow{\texttt{open\_}m} P'$. Then $O_1 \equiv n[(\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2) \mid P']$. It holds that $M \xrightarrow{n.\texttt{open\_}m} n[\circ \mid (\boldsymbol{\nu}\tilde{r})(M_1 \mid M_2)]$. The induction hypothesis tells us that there exist systems A, B, N' such that $N \Rightarrow A \xrightarrow{n.\texttt{open\_}m} B \Rightarrow N'$, and for all processes Q it holds that $M' \bullet Q \mathcal{S} N' \bullet Q$. The transition $A \xrightarrow{n.\texttt{open\_}m} B$ must have been derived from $A \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{s})\langle N_1\rangle N_2$. Then A must be of the form $(\boldsymbol{\nu}\tilde{s})(m[N_1] \mid N_2)$ and B must be of the form $n[\circ \mid (\boldsymbol{\nu}\tilde{s})(N_1 \mid N_2)]$. This implies that $n[N \mid P] \Rightarrow B \bullet P' \rightarrow N' \bullet P'$. As for all Q it holds that $M' \bullet Q \mathcal{S} N' \bullet Q$, we can deduce $M' \bullet P' \mathcal{S} N' \bullet P'$ and conclude by take $O_2 = N' \bullet P'$.

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

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

— $n[M \mid P] \xrightarrow{n.\texttt{exit\_}m} m[\,\circ\,] \mid n[M \mid P'] = O_1$, because $P \xrightarrow{\texttt{out\_}m} P'$. It also holds that $n[N \mid P] \xrightarrow{n.\texttt{exit\_}m} m[\,\circ\,] \mid n[N \mid P']$. 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 $M \mathcal{S} N$ because of the closure properties of $\mathcal{S}$. □

Proof of Lemma 3.7 The relation $\{((\boldsymbol{\nu}n)n[\,], \mathbf{0})\}$ is a bisimulation, and the result follows from the soundness of bisimulation. □

Proof of Lemma 3.8 – omitted cases In all the cases below, ambients of the form $(\boldsymbol{\nu}n)n[\,]$ are garbage collected by applying Lemma 3.7.

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

$$\begin{array}{rl} & \mathcal{C}_{k.\mathtt{exit}\_n} M \bullet P \\ \equiv & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid n[P \mid k[\mathtt{out}\_n.M_1 \mid M_2] \mid M_3)] \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)k[a[\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid M_1 \mid M_2] \mid n[P \mid M_3]) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\mathsf{done}[\mathtt{out}\_a]] \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)(\mathsf{done}[\,] \mid a[\,]) \mid k[M_1 \mid M_2] \mid n[P \mid M_3]) \\ \cong_{\mathsf{s}} & (\boldsymbol{\nu}\tilde{m})(k[M_1 \mid M_2] \mid n[\circ \mid P_3]) \bullet P \mid \mathsf{done}[\,] \\ = & M' \bullet P \mid \mathsf{done}[\,] \end{array}$$

This implies $\mathcal{C}_{k.\mathtt{exit}\_n}[M] \bullet P \Rightarrow\!\cong_{\mathsf{s}} M' \bullet P \mid \mathsf{done}[\,]$.

Case $\alpha = n.\overline{\mathtt{enter}}\_k$**.** Let P be a process. We know that $M \xrightarrow{n.\overline{\mathtt{enter}}\_k} M'$. 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,

$$\begin{array}{rl} & \mathcal{C}_{n.\overline{\mathtt{enter}}\_k}[M] \bullet P \\ \equiv & (\boldsymbol{\nu}\tilde{m})((\boldsymbol{\nu}a)a[\mathtt{in}\_n.k[\mathtt{out}\_a.(P \mid (\boldsymbol{\nu}b)b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]])]] \mid n[M_1] \mid M_2) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[k[\mathtt{out}\_a.(P \mid (\boldsymbol{\nu}b)b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]])]]] \mid M_2) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P \mid (\boldsymbol{\nu}b)b[\mathtt{out}\_k.\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]]]] \mid M_2) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P] \mid (\boldsymbol{\nu}b)b[\mathtt{out}\_n.\mathsf{done}[\mathtt{out}\_b]]] \mid M_2) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P]] \mid (\boldsymbol{\nu}b)b[\mathsf{done}[\mathtt{out}\_b]] \mid M_2) \\ \xrightarrow{\tau} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid (\boldsymbol{\nu}a)a[] \mid k[P]] \mid (\boldsymbol{\nu}b)b[] \mid \mathsf{done}[] \mid M_2) \\ \cong_{\mathsf{s}} & (\boldsymbol{\nu}\tilde{m})(n[M_1 \mid k[\,\circ\,]] \mid M_2) \bullet P \mid \mathsf{done}[\,] \\ = & M' \bullet P \mid \mathsf{done}[\,] \end{array}$$

This implies $\mathcal{C}_{n.\overline{\mathtt{enter}}\_k}[M] \bullet P \Rightarrow\!\cong_{\mathsf{s}} M' \bullet P \mid \mathsf{done}[\,]$.

Case $\alpha = k.\mathtt{open}\_n$**.** 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 $n \notin \{\tilde{m}\}$, and $M' \equiv k[\circ \mid (\boldsymbol{\nu}\tilde{m})(M_1 \mid M_2)]$. Names

a and b are fresh for M. Now,

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

This implies $\mathcal{C}_{k.\mathtt{open}\_n}[M] \bullet P \;\Rightarrow\cong_{\mathsf{s}}\; M' \bullet P \mid \mathsf{done}[\,]$. □

Proof of Lemma 3.10

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

For point b), $M \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle = \mathcal{C}[\mathsf{spy}_\alpha\langle i,j,P\rangle] \xrightarrow{\tau} \mathcal{C}'[\mathsf{spy}_\alpha\langle i,j,P\rangle] = M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$ implies $M = \mathcal{C}[\circ] \xrightarrow{\tau} \mathcal{C}'[\circ] = M'$, as required.

Part 2. It is easy to see that the relation

$$\mathcal{R} = \{(n[(\boldsymbol{\nu} i,j)\mathsf{spy}_\alpha\langle i,j,P\rangle \mid R], n[P \mid R]) \mid \text{for all } n, R\} \;\cup \mathcal{I}$$

is a bisimulation up-to context. Observe that the soundness of the up-to context proof technique does not depend on the completeness of the bisimilarity. □

Proof of Lemma 3.11 If $\mathcal{C}[r[P]] \xrightarrow{\tau} \mathcal{C}'[r[P']]$, then either $P \xrightarrow{\tau} P'$ (the process P performs the τ action without a contribution of the context $\mathcal{C}[-]$, or $\mathcal{C}[\mathbf{0}] \xrightarrow{\tau} \mathcal{C}'[\mathbf{0}]$ (the context $\mathcal{C}[-]$ performs the τ action without a contribution of the process $r[P]$), or $\mathcal{C}[-]$ and $r[P]$ interact. In the two first cases, the result follows easily. In the last case, since the name r is fresh, the only possible interactions are originated by $P \xrightarrow{\mathtt{in}\_m}$ or $P \xrightarrow{\mathtt{out}\_m}$, that is the ambient r moves in the ambient hierarchy of $\mathcal{C}[-]$, which is otherwise left unchanged. The result follows. □

Proof of Lemma 3.12 – omitted cases

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

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

As $N \Downarrow i, j$ and done is fresh, by Lemma 3.10(1), there must be a system context $\mathcal{D}[-]$ such that $N \mid \mathsf{done}[\,] \equiv \mathcal{D}[\mathsf{done}[\,]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$ and $\mathcal{C}_\alpha[M] \Rightarrow \mathcal{D}[\mathsf{done}[\,]]$. 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 $\mathcal{C}[-]$ such that:

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

Examining the above reductions sequence from $\mathcal{C}_\alpha[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle$ we conclude that

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

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

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

Repeated application of Lemma 3.10(1) gives $(\boldsymbol{\nu}\tilde{m})(n[\circ \mid \mathbf{0} \mid M_1] \mid M_2) \Rightarrow \mathcal{D}[\mathbf{0}]$, and therefore, as $\equiv$ is closed under reduction, there is a M', $M' \equiv \mathcal{D}[\mathbf{0}]$, such that $M \xRightarrow{k.\mathtt{enter}\_n} M'$, as desired.

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

$$\mathcal{C}_{k.\mathtt{exit}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle \equiv (\boldsymbol{\nu}a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid n[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid M]\ .$$

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 static contexts $\mathcal{D}[-]$ and $\mathcal{E}[-_1, -_2]$ such that:

$$\begin{array}{rll} & \mathcal{C}_{k.\mathtt{exit}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle & \\ = & (\boldsymbol{\nu}a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid n[\mathsf{spy}_\alpha\langle i, j, P\rangle \mid M] & \\ \Rightarrow & (\boldsymbol{\nu}a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]] \mid M_1 \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle & \\ \xrightarrow{\tau} & (\boldsymbol{\nu}a)\mathcal{D}[a[\mathtt{out}\_k.\mathsf{done}[\mathtt{out}\_a]]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle & \\ \Rightarrow & (\boldsymbol{\nu}a)\mathcal{E}[\mathsf{done}[\,], a[\,]] \bullet \mathsf{spy}_\alpha\langle i, j, P\rangle & (\star) \\ \equiv & N \mid \mathsf{done}[\,] & \end{array}$$

Examining the above reductions sequence from $\mathcal{C}_{k.\mathtt{exit}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$ 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:

$$(\boldsymbol{\nu} a)a[\mathtt{in}\_k.\mathtt{out}\_k.\mathbf{0}] \mid M_1 \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;\Rightarrow (\boldsymbol{\nu} a)\mathcal{E}[\mathbf{0}, a[\,]] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;.$$

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

$$(\boldsymbol{\nu} a)(\mathbf{0} \mid M_1) \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;\Rightarrow (\boldsymbol{\nu} a)\mathcal{E}[\mathbf{0},\mathbf{0}] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle \;.$$

Summarising,

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

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

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

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

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

$$\begin{array}{l}\mathcal{C}_{k.\mathtt{open}\_n}[M] \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle = \\ \quad k[\mathsf{spy}_\alpha\langle i,j,P\rangle \mid (\boldsymbol{\nu} a,b)(\mathtt{open}\_b.\mathtt{open}\_a.\mathsf{done}[\mathtt{out}\_k] \mid a[M \mid \mathtt{open}\_n.b[\mathtt{out}\_a]])]\end{array}$$

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 $\mathcal{D}[-]$ such that:

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

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

$$M \;\Rightarrow \xrightarrow{\;k.\mathtt{open}\_n\;} k[\circ \mid Q].$$

As

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

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

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

By Lemma 3.10, this implies

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

Applying our proof techniques we can easily prove that:

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

As $\cong_{\mathsf{s}}$ is closed under reduction, it follows that there is M' such that

$$k[\circ \mid Q] \;\Rightarrow M' \cong_{\mathsf{s}} \mathcal{D}[\mathbf{0}].$$

So, there is M' such that $M \;\Rightarrow M'$ and $N \cong_{\mathsf{s}} M' \bullet \mathsf{spy}_\alpha\langle i,j,P\rangle$, as desired. □

C. PROOFS FROM SECTION 4

Proof of Lemma 4.3 Let $\mathcal{R}$ be a bisimulation up to context and up to $(\gtrsim, \approx)$. Suppose that $(M,N) \in \mathcal{R}$ and that $\mathcal{C}[M] \xrightarrow{\alpha} M''$ for some system context $\mathcal{C}[-]$. We prove the result by induction on the structure of $\mathcal{C}[-]$.

—$\mathcal{C}[-] = -$. In this case the result follows directly from the definition of bisimulation up to context and up to $(\gtrsim, \approx)$.

—$\mathcal{C}[-] = - \mid H$, for some system H. We decompose the transition $\mathcal{C}[M] \xrightarrow{\alpha} M''$, distinguishing four cases.

(1) The transition is performed by M, that is, $M \xrightarrow{\alpha} M'$ and $M'' = M' \mid H$. As $(M,N) \in \mathcal{R}$ there is N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and for all processes P there is a systems context $\mathcal{D}_P[-]$ and systems M'''_P and N'''_P such that $M' \bullet P \gtrsim \mathcal{D}_P[M'''_P]$, $N' \bullet P \approx \mathcal{D}_P[N'''_P]$, and $M'''_P \;\mathcal{R}\; N'''_P$. As $\gtrsim$ is preserved by parallel composition, for all processes P we have $M'' \bullet P = (M' \mid H) \bullet P = (M' \bullet P) \mid H \gtrsim \mathcal{D}_P[M'''_P] \mid H = \mathcal{E}_P[M'''_P]$, where $\mathcal{E}_P[-] = \mathcal{D}_P[-] \mid H$. Moreover, $\mathcal{C}[N] = N \mid H \stackrel{\hat{\alpha}}{\Longrightarrow} N' \mid H = N''$. As $\approx$ is preserved by parallel composition $N'' \bullet P = (N' \mid H) \bullet P = (N' \bullet P) \mid H \approx \mathcal{D}_P[N'''_P] \mid H = \mathcal{E}_P[N'''_P]$, as required.

(2) The transition is performed by H, that is, $H \xrightarrow{\alpha} H'$ and $M'' = M \mid H'$. This case follows easily.

(3) The systems M and H interact because $M \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{p})\langle k[P_1]\rangle M_2$ and $H \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{q})\langle Q_1\rangle H_2$, obtaining

$$M \mid H \xrightarrow{\tau} M'' \equiv (\boldsymbol{\nu}\tilde{p}\tilde{q})(n[k[P_1] \mid Q_1] \mid M_2 \mid H_2)\ .$$

We distinguish two subcases, because, depending on whether $k \in \tilde{p}$ or not, the system M may move either a free or a private ambient k into the ambient n. We detail the case $k \notin \tilde{p}$, the other being similar. We derive $M \xrightarrow{k.\mathtt{enter}\_n} M' = (\boldsymbol{\nu}\tilde{p})(n[k[P_1] \mid \circ] \mid M_2)$. As $(M,N) \in \mathcal{R}$, there is N' such that $N \xRightarrow{k.\mathtt{enter}\_n} N'$ and for all processes P there is a systems context $\mathcal{D}_P[-]$ and systems M'''_P and N'''_P such that $M' \bullet P \gtrsim \mathcal{D}_P[M'''_P]$, $N' \bullet P \approx \mathcal{D}_P[N'''_P]$, and $M'''_P \mathcal{R} N'''_P$. As $\gtrsim$ is preserved by restriction and parallel composition, it holds that $M'' \bullet P = M'' \equiv (\boldsymbol{\nu}\tilde{q})(M' \bullet Q_1 \mid H_2) \gtrsim (\boldsymbol{\nu}\tilde{q})(\mathcal{D}_{Q_1}[M'''_{Q_1}] \mid H_2) = \mathcal{E}_{Q_1}[M'''_{Q_1}]$, for $\mathcal{E}_{Q_1}[-] = (\boldsymbol{\nu}\tilde{q})(\mathcal{D}_{Q_1}[-] \mid H_2)$. Moreover, $\mathcal{C}[N] = N \mid H \Rightarrow (\boldsymbol{\nu}\tilde{q})(N' \bullet Q_1 \mid H_2) = N''$. As also $\approx$ is preserved by restriction and parallel composition, we obtain that $N'' \bullet P = N'' \approx (\boldsymbol{\nu}\tilde{q})(\mathcal{D}_{Q_1}[N'''_{Q_1}] \mid H_2) = \mathcal{E}_{Q_1}[N'''_{Q_1}]$, as required.

(4) The systems M and H interact since $M \xrightarrow{\mathtt{amb}\_n} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle M_2$ and $H \xrightarrow{\mathtt{enter}\_n} (\boldsymbol{\nu}\tilde{q})\langle k[Q_1]\rangle H_2$, obtaining

$$M \mid H \xrightarrow{\tau} M'' \equiv (\boldsymbol{\nu}\tilde{p}\tilde{q})(n[P_1 \mid k[Q_1]] \mid M_2 \mid H_2) \ .$$

This case is similar to the previous one.

—$\mathcal{C}[-] = n[- \mid R]$, for some name n and process R. We decompose the transition $\mathcal{C}[M] \xrightarrow{\alpha} M''$, distinguishing three cases.

(1) If $\alpha \neq \tau$ then the system M does not play any role in the transition, and it is easy to conclude.

(2) If $\alpha = \tau$ but M does not take part in the transition, it is easy to conclude.

(3) If $\alpha = \tau$ and M takes part in the transition, then we distinguish four cases.

(a) Suppose that $M \xrightarrow{k.\mathtt{exit}\_n} M'$, with $M'' \equiv M' \bullet R$. As $(M,N) \in \mathcal{R}$ there is N' such that $N \xRightarrow{k.\mathtt{exit}\_n} N'$ and for all processes P there exist a systems context $\mathcal{D}_P[-]$ and systems M'''_P and N'''_P such that $M' \bullet P \gtrsim \mathcal{D}_P[M'''_P]$, $N' \bullet P \approx \mathcal{D}_P[N'''_P]$, and $M'''_P \mathcal{R} N'''_P$. Thus $M'' \bullet P = M'' \equiv M' \bullet R \gtrsim \mathcal{D}_R[M'''_R]$. Moreover, $\mathcal{C}[N] \Rightarrow N' \bullet R = N''$ and $N'' \bullet P = N'' = N' \bullet R \approx \mathcal{D}_R[N'''_R]$, as required.

Suppose instead that $M \xrightarrow{*.\mathtt{exit}\_n} M'$, with $M'' \equiv M' \bullet R$. As $(M,N) \in \mathcal{R}$ there is N' such that $n[N \mid \circ] \Rightarrow N'$ and for all processes P there exist a systems context $\mathcal{D}_P[-]$ and systems M'''_P and N'''_P such that $M' \bullet P \gtrsim \mathcal{D}_P[M'''_P]$, $N' \bullet P \approx \mathcal{D}_P[N'''_P]$, and $M''' \mathcal{R} N'''$. Thus $M'' \bullet P = M'' \equiv M' \bullet R \gtrsim \mathcal{D}_R[M'''_R]$. Moreover, $\mathcal{C}[N] \Rightarrow N' \bullet R = N''$ and $N'' \bullet P = N'' = N' \bullet R \approx \mathcal{D}_R[N'''_R]$, as required.

(b) $M \xrightarrow{\mathtt{amb}\_k} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle M_2$ and $R \xrightarrow{\mathtt{open}\_k} R_1$, with $M'' = n[(\boldsymbol{\nu}\tilde{p})(P_1 \mid M_2) \mid R_1]$. This implies $M \xrightarrow{n.\mathtt{open}\_k} n[(\boldsymbol{\nu}\tilde{p})(P_1 \mid M_2) \mid \circ] = M'$. As $(M,N) \in \mathcal{R}$ there is N' such that $N \xRightarrow{n.\mathtt{open}\_k} N'$ and for all processes P there exist a system context $\mathcal{D}_P[-]$ and systems M'''_P and N'''_P such that $M' \bullet P \gtrsim \mathcal{D}_P[M'''_P]$, $N' \bullet P \approx \mathcal{D}[N'''_P]$, and $M''' \mathcal{R} N'''$. Thus $M'' \bullet P = M'' \equiv M' \bullet R_1 \gtrsim \mathcal{D}_{R_1}[M'''_{R_1}]$. Moreover, $\mathcal{C}[N] \Rightarrow N' \bullet R_1 = N''$ and $N'' \bullet P = N'' = N' \bullet R_1 \approx \mathcal{D}_{R_1}[N'''_{R_1}]$, as required.

(c) $M \xrightarrow{\mathtt{enter}\_m} (\boldsymbol{\nu}\tilde{p})\langle k[P_1]\rangle M_2$ and $R \xrightarrow{\mathtt{amb}\_m} (\boldsymbol{\nu}\tilde{q})\langle R_1\rangle R_2$, with

$$M'' \equiv n[(\boldsymbol{\nu}\tilde{p}\tilde{q})(m[k[P_1] \mid R_1] \mid M_2 \mid R_2)] \ .$$

This case is similar to case $\mathcal{C}[-] = - \mid H$, subcase 3.

(d) $M \xrightarrow{\texttt{amb\_}m} (\boldsymbol{\nu}\tilde{p})\langle P_1\rangle M_2$ and $R \xrightarrow{\texttt{enter\_}m} (\boldsymbol{\nu}\tilde{q})\langle k[R_1]\rangle R_2$, with

$$M'' \equiv n[(\boldsymbol{\nu}\tilde{p}\tilde{q})(m[P_1 \mid k[R_1]] \mid M_2 \mid R_2)] \ .$$

This case is similar to case $\mathcal{C}[-] = - \mid H$, subcase 4.

—$\mathcal{C}[-] = (\boldsymbol{\nu}n)-$. We do not detail this case, which follows easily. □

D. PROOFS FROM SECTION 5

Proof of Lemma 5.4 Let $\mathcal{R} = \{(P', Q') : P' \equiv (\boldsymbol{\nu}n)P,\ Q' \equiv (\boldsymbol{\nu}n)Q,\ P \cong^{\mathsf{e}}_{\mathsf{p}} Q\} \cup \ \cong^{\mathsf{e}}_{\mathsf{p}}$. We show that $\mathcal{R} \subseteq \cong^{\mathsf{e}}_{\mathsf{p}}$. The relation $\mathcal{R}$ is reduction closed because both $\cong^{\mathsf{e}}_{\mathsf{p}}$ and $\equiv$ are, and restriction does not influence internal reductions. $\mathcal{R}$ is also barb preserving because $\cong^{\mathsf{e}}_{\mathsf{p}}$ and $\equiv$ are. To prove that $\mathcal{R}$ is closed under ambient nesting, we have to show that if $P' \ \mathcal{R}\ Q'$, with $P' \equiv (\boldsymbol{\nu}n)P$ and $Q' \equiv (\boldsymbol{\nu}n)Q$, then $k[P'] \ \mathcal{R}\ k[Q']$. But $k[P'] \equiv k[(\boldsymbol{\nu}n)P] \equiv (\boldsymbol{\nu}n)k[P]$ and $k[Q'] \equiv k[(\boldsymbol{\nu}n)Q] \equiv (\boldsymbol{\nu}n)k[Q]$. Moreover, by definition of $\cong^{\mathsf{e}}_{\mathsf{p}}$, $k[P] \cong^{\mathsf{e}}_{\mathsf{p}} k[Q]$. The result follows from the construction of $\mathcal{R}$. The argument for parallel composition is similar. □

Proof of Lemma 5.5 To prove the inclusion $\cong^{\mathsf{e}}_{\mathsf{p}} \cap (\mathcal{M} \times \mathcal{M}) \subseteq \cong_{\mathsf{s}}$, observe that the relation $\cong^{\mathsf{e}}_{\mathsf{p}} \cap (\mathcal{M} \times \mathcal{M})$ is: reduction closed because $\cong^{\mathsf{e}}_{\mathsf{p}}$ is reduction closed and systems always reduce in systems; barb preserving because $\cong^{\mathsf{e}}_{\mathsf{p}}$ preserves barbs; closed under system contexts because $\cong^{\mathsf{e}}_{\mathsf{p}}$ is preserved by parallel composition, ambient, and, by Lemma 5.4, by restriction. □

Proof of Theorem 5.2 – omitted cases We prove that the relation $\cong^{\mathsf{e}}_{\mathsf{p}}$ is preserved by prefixing. We have to prove that if $P \cong^{\mathsf{e}}_{\mathsf{p}} Q$, then $C.P \cong^{\mathsf{e}}_{\mathsf{p}} C.Q$. Rather than working directly with $\cong^{\mathsf{e}}_{\mathsf{p}}$, we use Theorem 5.3 and we prove that $C.P \ \mathcal{S}\ C.Q$. For that, we must show that for all n, R, it holds that $n[C.P \mid R] \approx n[C.Q \mid R]$. We perform a case analysis on C.

—$C = \texttt{in\_}o$. We show that the relation

$$\mathcal{R} = \{(n[\texttt{in\_}o.P \mid R], n[\texttt{in\_}o.Q \mid R]) : P \cong^{\mathsf{e}}_{\mathsf{p}} Q,\ n, R \text{ arbitrary}\}^{=} \cup \ \approx$$

is a bisimulation up to context and up to structural congruence. Suppose that $n[\texttt{in\_}o.P \mid R] \xrightarrow{\alpha} M$. We perform a case analysis on α.

— $\alpha = \tau$. There are two sub-cases.

First case. $M \equiv n[\texttt{in\_}o.P \mid R']$ with $R \xrightarrow{\tau} R'$. It follows that $n[\texttt{in\_}o.Q \mid R] \xrightarrow{\tau} N$, where $N \equiv n[\texttt{in\_}o.Q \mid R']$, and $M \equiv \mathcal{R} \equiv N$.

Second case. $M \equiv (\boldsymbol{\nu}\tilde{r})(r[R_1] \mid n[\texttt{in\_}o.P \mid R_2])$, where $R \equiv (\boldsymbol{\nu}\tilde{r})(r[\texttt{out\_}n.R_1] \mid R_2)$. This implies $n[\texttt{in\_}o.Q \mid R] \xrightarrow{\tau} N$, where $N \equiv (\boldsymbol{\nu}\tilde{r})(r[R_1] \mid n[\texttt{in\_}o.Q \mid R_2])$. Now, we can factor out the system context $\mathcal{C}[-] = (\boldsymbol{\nu}\tilde{r})(r[R_1] \mid -)$ and the construction of $\mathcal{R}$ guarantees that we are still in $\mathcal{R}$ up to context and up to $\equiv$.

— $\alpha = m.\texttt{open\_}n$. Then $M \equiv m[\circ \mid \texttt{in\_}o.P \mid R]$. This implies $n[\texttt{in\_}o.Q \mid R] \xrightarrow{m.\texttt{open\_}n} N$, where $N \equiv m[\circ \mid \texttt{in\_}o.Q \mid R]$. Then, for all processes R' we have $M \bullet R' \equiv \mathcal{R} \equiv N \bullet R'$.

— $\alpha = n.\overline{\texttt{enter}}\_k$. Then $M \equiv n[\texttt{in\_}o.P \mid R \mid k[\,\circ\,]]$. This implies $n[\texttt{in\_}o.Q \mid R] \xrightarrow{n.\overline{\texttt{enter}}\_k} N$, where $N \equiv n[\texttt{in\_}o.Q \mid R \mid k[\,\circ\,]]$. Then for all processes R' we have $M \bullet R' \equiv \mathcal{R} \equiv N \bullet R'$.

— $\alpha = n.\mathtt{exit}\_k$. Then $M \equiv n[\mathtt{in}\_o.P \mid R'] \mid k[\,\circ\,]$ and R has unleashed the capability $\mathtt{out}\_k$ turning into R'. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\mathtt{exit}\_k} N$, where $N \equiv n[\mathtt{in}\_o.Q \mid R'] \mid k[\,\circ\,]$. Then, factoring out the context $\mathcal{C}[-] = - \mid k[S]$, for all processes S, the construction of $\mathcal{R}$ guarantees that we are still in $\mathcal{R}$ up to context and up to $\equiv$.

— $\alpha = n.\mathtt{enter}\_o$. There are two sub-cases.

First case. $M \equiv o[n[\mathtt{in}\_o.P \mid R'] \mid \circ]$ and R has unleashed the capability $\mathtt{in}\_o$ turning into R'. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\mathtt{enter}\_o} N$, where $N \equiv o[n[\mathtt{in}\_o.Q \mid R'] \mid \circ]$. Then, factoring out the context $\mathcal{C}[-] = o[- \mid S]$, for all processes S, the construction of $\mathcal{R}$ guarantees that we are still in $\mathcal{R}$ up to context and up to $\equiv$.

Second case. $M \equiv o[n[P \mid R] \mid \circ]$. This implies $n[\mathtt{in}\_o.Q \mid R] \xrightarrow{n.\mathtt{enter}\_o} N$, where $N \equiv o[n[Q \mid R] \mid \circ]$. As $P \cong_{\mathsf{p}}^{\mathsf{e}} Q$ it holds that $n[P \mid R] \cong_{\mathsf{p}}^{\mathsf{e}} n[Q \mid R]$. By Theorem 5.3 we get $M \bullet S \equiv\approx\equiv N \bullet S$ and hence $M \bullet S \,\mathcal{R}\, N \bullet S$.

— $\alpha = n.\mathtt{enter}\_k,\ k \neq o$. It is similar to the first part of the previous case.

—$C = \mathtt{out}\_o$. We show that the relation

$$\mathcal{R} = \{(n[\mathtt{out}\_o.P \mid R], n[\mathtt{out}\_o.Q \mid R]) :\ P \cong_{\mathsf{p}}^{\mathsf{e}} Q\}^{=} \ \cup\ \approx$$

is a bisimulation up to context and up to structural congruence. The only case different from the above is when the process $\mathtt{out}\_o.P$ exercises the capability $\mathtt{out}\_o$. Suppose $n[\mathtt{out}\_o.P \mid R] \xrightarrow{n.\mathtt{exit}\_o} M \equiv n[P \mid R] \mid o[\,\circ\,]$. This implies $n[\mathtt{out}\_o.Q \mid R] \xrightarrow{n.\mathtt{exit}\_o} N \equiv n[Q \mid R] \mid o[\,\circ\,]$. As $P \cong_{\mathsf{p}}^{\mathsf{e}} Q$ it holds that $n[P \mid R] \cong_{\mathsf{p}}^{\mathsf{e}} n[Q \mid R]$. By Lemma 5.5 and Theorem 3.15 it follows $n[P \mid R] \approx n[Q \mid R]$. As $\approx$ is preserved by system contexts, we have $M \bullet S \equiv\approx\equiv N \bullet S$. As a consequence, $M \bullet S \,\mathcal{R}\, N \bullet S$.

—$C = \mathtt{open}\_o$. We show that the relation

$$\mathcal{R} = \{(n[\mathtt{open}\_o.P \mid R], n[\mathtt{open}\_o.Q \mid R]) :\ P \cong_{\mathsf{p}}^{\mathsf{e}} Q\}^{=} \ \cup\ \approx$$

is a bisimulation up to context and up to structural congruence. The only case different from the above is when the process $\mathtt{open}\_o.P$ exercises the capability $\mathtt{open}\_o$. Suppose $n[\mathtt{open}\_o.P \mid R] \xrightarrow{\tau} n[P \mid R']$. This implies $n[\mathtt{open}\_o.Q \mid R] \xrightarrow{\tau} n[Q \mid R']$. As $P \cong_{\mathsf{p}}^{\mathsf{e}} Q$ it holds that $n[P \mid R'] \cong_{\mathsf{p}}^{\mathsf{e}} n[Q \mid R']$. By Lemma 5.5 and Theorem 3.15 it follows $n[P \mid R'] \approx n[Q \mid R']$. As a consequence, $n[P \mid R'] \,\mathcal{R}\, n[Q \mid R']$. □

ACKNOWLEDGMENTS

We would like to thank Vladimiro Sassone, who spotted a problem in the proof of Theorem 4.5 in an early draft of the paper, James Leifer, for his suggestions for improving the paper, and the anonymous referees, for their accurate reading and their constructive comments. The second author is grateful to the Foundations of Computing Group of University of Sussex for the kind hospitality and support, and to the Computer Laboratory of the University of Cambridge.

REFERENCES

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

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

Boudol, G. 1992. Asynchrony and the π-calculus. Tech. Rep. RR-1702, INRIA-Sophia Antipolis.

Bugliesi, M., Crafa, S., Merro, M., and Sassone, V. 2005. Communication and mobility control in boxed ambients. *Information and Computation*. In press. An extended abstract appeared in *Proc. FSTTCS'02*, LNCS, Springer Verlag.

Cardelli, L. 1999. Wide area computation. *Lecture Notes in Computer Science 1644*, 10–24.

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

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

Castagna, G., Vitek, J., and Zappa Nardelli, F. 2005. The seal calculus. *Information and Computation 201:1*, 1–54.

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

Ferrari, G., Montanari, U., and Tuosto, E. 2001. A LTS semantics of ambients via graph synchronization with mobility. In *Proc. ICTCS*. LNCS, vol. 2202. Springer Verlag.

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

Godskesen, J., Hildebrandt, T., and Sassone, V. 2002. A calculus of mobile resources. In *Proc. 10th CONCUR '02*. LNCS, vol. 2421. Springer Verlag.

Gordon, A. D. and Cardelli, L. 2002. Equational properties of mobile ambients. *Journal of Mathematical Structures in Computer Science 12*, 1–38. An extended abstract appeared in *Proc. FoSSaCs '99*.

Hennessy, M., Merro, M., and Rathke, J. 2004. Towards a behavioural theory of access and mobility control in distributed systems. *Theoretical Computer Science 322*, 615–669.

Hennessy, M., Rathke, J., and Yoshida, N. 2003. Safedpi: A language for controlling mobile code. Computer Science Report 2003:02, University of Sussex. An extended abstract appeared in the Proc. FOSSACS'04, volume 2987, Lecture Notes in Computer Science. Springer-Verlag 2004.

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

Hirschkoff, D., Lozes, E., and Sangiorgi, D. 2002. Separability, expressiveness, and decidability in the ambient logic. In *Proc. LICS*. IEEE Computer Society Press, 423–432.

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

Honda, K. and Yoshida, N. 1995. On reduction-based process semantics. *Theoretical Computer Science 152,* 2, 437–486.

Howe, D. J. 1996. Proving congruence of bisimulation in functional programming languages. *Information and Computation 124,* 2, 103–112.

Jeffrey, A. and Rathke, J. 2005. Contextual equivalence for higher-order π-calculus revisited. *Logical Methods in Computer Science 1,* 1-4.

Jensen, O. H. and Milner, R. 2004. Bigraphs and mobile processes (revised). Tech. Rep. 580, LFCS, Dept. of Comp. Sci., Edinburgh Univ. Feb. An extended abstract appeared in *Conference Record of 30th Symposium on Principles of Programming Languages*, ACM Press, 2003.

Leifer, J. J. and Milner, R. 2000. Deriving bisimulation congruences for reactive systems. In *Proc. CONCUR 2000*. LNCS, vol. 1877. Springer-Verlag, 243–258.

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

Levi, F. and Sangiorgi, D. 2003. Mobile safe ambients. *ACM Transactions on Programming Languages and Systems 25,* 1 (Jan.), 1–69.

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

Merro, M. and Hennessy, M. in press, 2005. A bisimulation-based semantic theory for safe ambients. *ACM Transactions on Programming Languages and Systems*.

Milner, R. 1989. *Communication and Concurrency*. Prentice Hall.

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

Milner, R. and Sangiorgi, D. 1992. Barbed bisimulation. In *Proc. 19th ICALP*. LNCS, vol. 623. Springer Verlag, 685–695.

Park, D. 1981. Concurrency on automata and infinite sequences. In *Conf. on Theoretical Computer Science*, P. Deussen, Ed. LNCS, vol. 104. Springer Verlag.

Sangiorgi, D. 1992. Expressing mobility in process algebras: First-order and higher-order paradigms. Ph.D. thesis, Dept. of Comp. Sci., Edinburgh University.

Sangiorgi, D. 1996a. Bisimulation for Higher-Order Process Calculi. *Information and Computation 131,* 2, 141–178.

Sangiorgi, D. 1996b. Locality and non-interleaving semantics in calculi for mobile processes. *Theoretical Computer Science 155*, 39–83.

Sangiorgi, D. 1998. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science 8*, 447–479.

Sangiorgi, D. 2001. Extensionality and intensionality of the ambient logic. In *Proc. 28th POPL*. ACM Press.

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

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

Sangiorgi, D. and Walker, D. 2001b. Some results on barbed equivalences in pi-calculus. In *Proc. CONCUR '01*. LNCS, vol. 2154. Springer Verlag.

Schmitt, A. and Stefani, J. 2004. The kell calculus: A family of higher-order distributed process calculi. In *LNCS*. Springer-Verlag. Workshop of Global Computing.

Sewell, P. 2002. From rewrite rules to bisimulation congruences. *TCS 274,* 1–2, 183–230.

T. Hildebrandt, J.C. Godskesen, M. B. 2004. Bisimulation congruences for homer. Technical Report TR-2004-52, ITU.

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

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

Vitek, J. and Castagna, G. 1999. Seal: A framework for secure mobile computations. In *Internet Programming Languages*. Number 1686 in LNCS. Springer Verlag, 47–77.