Probabilistic Mobile Ambients

Marta Kwiatkowska[a], Gethin Norman[*,a], David Parker[a], Maria Grazia Vigliotti[b]

[a] *Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD*
[b] *Department of Computing, Imperial College, London SW7 2BZ, United Kingdom*

Abstract

The calculus of Mobile Ambients has been introduced for expressing mobility and mobile computation. In this paper we present a probabilistic version of Mobile Ambients by augmenting the syntax of the original Ambient Calculus with a (guarded) probabilistic choice operator. To allow for the representation of both the probabilistic behaviour introduced through the new probabilistic choice operator and the nondeterminism present in the original Ambient Calculus we use probabilistic automata as the underpinning semantic model. The Ambient logic is a logic for Mobile Ambients that contains a novel treatment of both locations and hidden names. To specifying properties of Probabilistic Mobile Ambients, we extend this logic to specify probabilistic behaviour. In addition, to show the utility of our approach we present an example of a virus infecting a network.

Key words: Mobile Ambients, Ambient Logic, Probabilistic Verification.

1. Introduction

Computer networks, multiprocessors and parallel algorithms, though quite different, all provide examples of complex concurrent systems. All benefit from parallelism, yet require careful design to ensure that they function correctly. Process algebra [1, 2, 3, 4] has proved a useful abstraction in order to model complex concurrent systems, and has provided formal tools to verify correctness. In cases where the behaviour of large distributed systems involves random events, for instance electronic coin flipping in network coordination algorithms and communication protocols (e.g. [5, 6, 7, 8]), link failures, message loss or the arrival of requests from external users, the introduction of probabilities is necessary.

Over the last twenty years, significant effort has been directed towards augmenting process algebra with probabilities in order to obtain formal specifications of randomised systems in both the discrete-time [9, 10, 11, 12, 13, 14, 15] and continuous-time setting [16, 17, 18, 19, 20, 21]. In the discrete-time setting, the approach chosen in this paper, we can roughly divide this work into two categories: those that replace the nondeterministic

[*] Corresponding author
Email addresses: marta.kwiatkowska@comlab.ox.ac.uk (Marta Kwiatkowska), gethin.norman@comlab.ox.ac.uk (Gethin Norman), david.parker@comlab.ox.ac.uk (David Parker), mgv98@doc.ic.ac.uk (Maria Grazia Vigliotti)

behaviour with probabilistic behaviour, e.g. [9, 14], and those that keep the nondeterministic behaviour while enriching the calculus with a probabilistic choice operator, e.g. [11, 13]. Since asynchronous behaviour introduced through parallel composition is fundamental to Mobile Ambient behaviour, we take the latter approach and enrich the Ambient Calculus with a probabilistic choice operator. The semantics for our calculus is given in terms of probabilistic automata [22] which extend classical labelled transition systems by allowing both probabilistic and nondeterministic behaviour to be modelled.

The aim of our work is to find a suitable probabilistic model in order to realistically model the behaviour of distributed and mobile systems. This has led us to Probabilistic Mobile Ambients (*PMA*), a probabilistic extension of the Mobile Ambients (*MA*) [23, 24] devised to represent mobile computation. In *MA*, *Ambient* is the key concept. Ambients are meant to represent administrative domains as well as physical locations and mobile devices. Ambients have a tree structure - possibly containing sub-ambients. The main advantage of *MA*, with respect to other calculi, is the simple constructs of the language and the inherent hierarchical structure of the processes. In the community of programming languages, *MA* has already become very popular [23, 24, 25, 26, 27, 28, 29, 30] to describe many issues, from access control to security protocols, from systems biology [31] to implementation of distributed languages [32]. Yet, quantitative aspects of computation in *MA* have only been studied in [31, 15] within the boundaries of continuous-time Markov chains.

We define a probabilistic bisimulation relation over *PMA*. This definition differs from the standard probabilistic bisimulation defined on labelled probabilistic transition systems [33, 11] as, similarly to the Ambient Calculus, the labels have to be second order, i.e. they are not simple actions but also include processes. It is known from the literature on process algebra that, in this kind of labelled transition system, it is difficult to define bisimulation [34, 35] and the Ambient calculus is no exception. *Barbed bisimulation* [36], however, takes into account only reductions via synchronisation, and uses a special predicate that entails the point of view of an observer. For *CCS*, it has been proved that labelled and barbed bisimulation are equivalent [36]. We therefore adapt barbed bisimulation to the probabilistic setting.

MA serves as a model for a spatial logic [24], called Ambient Logic, which expresses, on top of standard modal logic, predicates regarding location and secret names. A lot of work has been carried out in the context of Ambient Logic, with particular focus on decidable fragments and characterisation of equivalence relations induced by the logic [37, 38, 39]. Our work augments the Ambient Logic with a probabilistic operator [9] to obtain Probabilistic Ambient Logic. The latter will serve as a tool to express properties regarding random events in the model. Probabilistic Ambient Logic is conservative with respect to the original Ambient Logic. It is an open question whether the equivalence relation induced by the logic over *PMA* matches structural congruence as in the standard Ambient Logic [37, 38, 40].

Finally, we shall present an example of a virus infecting a network to demonstrate the utility of our approach.

Contributions.. This paper makes three main contributions:

1. We extend the syntax and semantics of the original Mobile Ambients [24] to allow for probabilistic behaviour. In this development we follow the approach taken in [11, 22] by defining a semantics where probabilities and nondeterminism co-exist.

2. We define the probabilistic semantics as a rewriting system also known in the literature as reduction semantics [24, 36, 41]. We show that, as far as internal actions i.e. τ actions are concerned, the labelled semantics and the reduction semantics coincide.
3. We extend the Ambient Logic with a probabilistic operator, and define the satisfaction relation with respect to Probabilistic Mobile Ambients.

Outline of paper.. The remainder of the paper is organised as follows. In Section 2 we give the background on probability theory and probabilistic automata which is required in the remainder of the paper, while Section 3 reviews both *MA* and Ambient Logic. Section 4 introduces the syntax and the semantics of *PMA* and gives some small examples, and Section 4.6 demonstrates that the probabilistic extension of asynchronous *CCS* can be encoded into *PMA*. The syntax and semantics of the probabilistic Ambient Logic is given in Section 5 and Section 6 presents an example concerning a virus spreading through a network. Section 7 concludes the paper.

2. Preliminaries

In this section we present the preliminary concepts required in the remainder of the paper, namely probability distributions and probabilistic automata.

Definition 2.1. *A* probability distribution *over a countable set X is a function $\mu : X \to [0,1]$ such that $\sum_{x \in X} \mu(x) = 1$. We write $\mathsf{Distr}(X)$ to denote the set of all probability distributions over X. For any $x \in X$, the* point distribution *at x, written η_x, is defined as $\eta_x(y) = 1$ if $x{=}y$ and $\eta_x(y) = 0$ otherwise.*

For any countable set X, distribution $\mu \in \mathsf{Distr}(X)$ and subset $V \subseteq X$, we let $\mu(V) = \sum_{x \in V} \mu(x)$.

Probabilistic automata [22, 42] extend classical automata by allowing probabilistic as well as nondeterministic behaviour. They are essentially equivalent to Markov decision processes [43] and probabilistic-nondeterministic systems [44].

Definition 2.2. *A probabilistic automaton is a tuple $(S, \mathcal{A}ct, \to)$ where*

- *S is a set of states;*
- *$\mathcal{A}ct$ is a set of actions;*
- *$\to \subseteq S \times \mathcal{A}ct \times \mathsf{Distr}(S)$ is a* probabilistic transition relation.

As a special case, one can consider probabilistic automata without actions, i.e. automata of the form $(S, \to)$ where $\to \subseteq S \times \mathsf{Distr}(S)$.

We write $s \xrightarrow{a} \mu$ for $(s, a, \mu) \in \to$ and $s \xrightarrow{a,\mu} t$ if $s \xrightarrow{a} \mu$ and $\mu(t) > 0$. A *path*, representing a particular resolution of both nondeterminism *and* probability, is a non-empty finite or infinite sequence of transitions:

$$\pi = s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} s_2 \xrightarrow{a_2,\mu_2} \cdots$$

and we denote by $\pi(i)$ the ith state appearing in the path π.

In contrast, an *adversary* (or scheduler) represents a particular resolution of nondeterminism *only.* More precisely, an adversary A is a function mapping every finite path π to an over action-distribution pair (a, μ) such that if s is the last state of π, then $s \xrightarrow{a} \mu$.

For any state s and adversary A we denote by $Path^A(s)$ the set of infinite paths that have s as an initial state and correspond to the adversary A, that is, paths $\pi = s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} s_2 \xrightarrow{a_2,\mu_2} \cdots$ where $s_0 = s$, $A(s_0) = (a_0, \mu_0)$ and $A(s_0 \xrightarrow{a_0,\mu_0} \cdots \xrightarrow{a_n,\mu_n} s_{n+1}) = (a_{n+1}, \mu_{n+1})$ for all $n \in \mathbb{N}$. For each adversary A and state s, using standard techniques [45, 22], one can construct the probability measure $Prob_s^A$ over the set of infinite paths $Path^A(s)$.

3. Mobile Ambients and the Ambient Logic

Mobile Ambients (*MA*) [46, 24] aim to represent, in a general way, process mobility. The calculus introduces an abstract framework which allows one to describe both *mobile computing* (i.e. mobile hardware) and *mobile computation* (i.e. mobile software). The advantage of *MA* is the simple underlying, unifying concept of *ambients*, which are meant to represent bounded places for computation such as concrete locations, concrete domains or abstract domains. The main features of mobile ambients can be summarised as follows.

- An ambient defines a perimeter (boundary) that establishes what is inside the ambient and what is outside.
- An ambient has a *name.*
- Ambients can move around: they can enter or exit other ambients.
- An ambient is a collection of *local agents*, i.e. processes, which run directly inside the ambient. The syntax $n[\,P\,]$ denotes process P running inside an ambient with name n .
- An ambient may have other ambients inside, creating a hierarchy of nested ambients, which could be represented as a tree. Each sub-ambient has its own name and behaves as an independent ambient.
- When an ambient moves, all the sub-ambients and processes inside move with it.

3.1. Syntax and semantics of MA

We assume the existence of a set of names N and set of identifiers Id. The definition of the syntax of the calculus, given below, includes two syntactic categories: processes (including both agents and ambients themselves) and capabilities (which enable ambients and agents to perform operations). Note that we replace replication as used in the original Ambient Calculus with recursion [47].

Definition 3.1. *The set of process terms of MA is given by the syntax:*

$$\begin{array}{rcll} M & ::= & \mathsf{in}\, n \;\big|\; \mathsf{out}\, n \;\big|\; \mathsf{open}\, n & \text{(capabilities)} \\ P, Q & ::= & \mathbf{0} \;\big|\; n[\,P\,] \;\big|\; P \mid Q \;\big|\; (\mathsf{new}\, n)\, P \;\big|\; \mathsf{A} \;\big|\; \mathit{fix}_{\mathsf{A}} P \;\big|\; M.P & \text{(processes)} \end{array}$$

where $n \in \mathsf{N}$ *and* $\mathsf{A} \in \mathsf{Id}$.

Nil, written $\mathbf{0}$, represents the inactive process, i.e. the process that does not reduce. An *ambient*, written $n[\,P\,]$, is composed of two parts: n is the name of the ambient and P is the active process inside. The square brackets around P indicate the perimeter of the ambient. If the ambient moves, everything inside moves with it. *Parallel composition*, written $P \mid Q$, means that P and Q are running in parallel and can compute independently from each other. *Restriction*, written $(\mathsf{new}\, n)\, P$, of the name n makes that name private and unique to P. No other process can use this name for interacting with P. Restriction is a binder and P is its scope. Given a process P, a name n appears *bound* within P if it appears within a subexpression of the form $(\mathsf{new}\, n)\, Q$. Names that are not bound are said to appear *free* in P and we denote the set of bound and free names by $\mathsf{bn}(P)$ and $\mathsf{fn}(P)$ respectively. *Recursion* is introduced through identifiers A and the recursion operator $\mathit{fix}_{\mathsf{A}}P$. An identifier A is *bound* in a process P if it appears within a subexpression of the form $\mathit{fix}_{\mathsf{A}}Q$. *Prefix*, written $M.P$, represents a process where P is enabled only if the prefix M has been consumed. Capabilities can be thought of as terms that enable the ambients to perform some actions. An ambient gains the ability to go inside another ambient whose name is n with the '$\mathsf{in}\, n$' capability. An ambient gains the ability to leave a parent ambient whose name is n with the '$\mathsf{out}\, n$' capability. An ambient named n can be dissolved by means of the '$\mathsf{open}\, n$' capability.

The operational semantics of *MA* is defined through a *structural congruence* between processes and a *reduction relation*. Structural congruence equates processes that are equivalent up to a simple syntactic rearrangement without any computational significance. This relation was developed from the metaphor present in the 'Chemical Abstract Machine' [41].

Definition 3.2. *The* structural congruence relation $\equiv$ *is the smallest congruence (equivalence relation preserved by all algebraic contexts) over MA terms that satisfies the equations:*

$$
\boxed{
\begin{array}{lr}
P \equiv P \mid \mathbf{0} & (\textsc{struc par zero}) \\
P \mid Q \equiv Q \mid P & (\textsc{struc par com}) \\
(P \mid Q) \mid R \equiv P \mid (Q \mid R) & (\textsc{struc par assoc}) \\
(\mathsf{new}\, n)\, \mathbf{0} \equiv \mathbf{0} & (\textsc{struc zero res}) \\
(\mathsf{new}\, m)\, (\mathsf{new}\, n)\, P \equiv (\mathsf{new}\, n)\, (\mathsf{new}\, m)\, P & (\textsc{struc res res}) \\
(\mathsf{new}\, n)\, (P \mid Q) \equiv P \mid (\mathsf{new}\, n)\, Q \ \textit{if}\ n \notin \mathsf{fn}(P) & (\textsc{struc res par}) \\
(\mathsf{new}\, m)\, n[\,P\,] \equiv n[\,(\mathsf{new}\, m)\, P\,] \ \ \textit{if}\ n \neq m & (\textsc{struc res amb}) \\
\mathit{fix}_{\mathsf{A}}P \equiv P\{\mathit{fix}_{\mathsf{A}}P/\mathsf{A}\} & (\textsc{struc rec})
\end{array}
}
$$

where, for any identifier A *and processes* $P, Q \in MA$*, the process* $P\{Q/\mathsf{A}\}$ *is obtained by substituting each free occurrence of* A *in* P *with* Q *and changing the bound identifiers of* P *to avoid clashes.*

The meaning of a computation in *MA* is given by the basic movement that ambients are able to make: entering, exiting and dissolving an ambient. Formally, steps of computation are represented by the reduction relation defined below.

Definition 3.3. *The* reduction relation $\rightarrow \subseteq \mathsf{MA} \times \mathsf{MA}$ *is the smallest binary relation over MA terms satisfying the set of rules:*

$$\begin{array}{lll}
m[\,\mathsf{in}\,n.P \mid Q\,] \mid n[\,R\,] & \rightarrow\ n[\,m[\,P \mid Q\,] \mid R\,] & (\textsc{red in}) \\
n[\,m[\,\mathsf{out}\,n.P \mid Q\,] \mid R\,] & \rightarrow\ m[\,P \mid Q\,] \mid n[\,R\,] & (\textsc{red out}) \\
\mathsf{open}\,n.P \mid n[\,Q\,] & \rightarrow\ P \mid Q & (\textsc{red open})
\end{array}$$

$$\frac{P \rightarrow P'}{P \mid R \rightarrow P' \mid R}\ (\textsc{red par}) \qquad \frac{P \rightarrow P'}{(\mathsf{new}\,n)\,P \rightarrow (\mathsf{new}\,n)\,P'}\ (\textsc{red restr})$$

$$\frac{P \rightarrow P'}{n[\,P\,] \rightarrow n[\,P'\,]}\ (\textsc{red amb}) \qquad \frac{Q \equiv P \rightarrow P' \equiv Q'}{Q \rightarrow Q'}\ (\textsc{red cong})$$

Furthermore, let $\rightarrow^*$ be the reflexive and transitive closure of $\rightarrow$.

The observational predicate ($\downarrow$) expresses what can be observed during computation. In the case of *MA* the name of a top level ambient has traditionally been chosen as a basic observation. Formally we have the following definitions.

Definition 3.4. *A process P* exhibits a barb *n, written $P \Downarrow n$, if and only if*

$$P \equiv (\mathsf{new}\,k_1)\ \ldots (\mathsf{new}\,k_n)\,(n[\,Q\,] \mid R)$$

for some processes $Q, R \in \mathsf{MA}$ and name $n \in \mathsf{N} \setminus \{k_1 \ldots k_n\}$. Furthermore, a process P eventually *exhibits a barb n, written as $P \Downarrow^* n$, if and only if $P \rightarrow^* Q$ and $Q \Downarrow n$.*

Definition 3.5. *Let $P \in \mathsf{MA}$. The process Q is* a step away *from P, written $P \downarrow Q$, if and only if there exists a name $n \in \mathsf{N}$ and process $R \in \mathsf{MA}$ such that $P \equiv n[\,Q\,] \mid R$. Furthermore, let $\downarrow^*$ be the reflexive and transitive closure of $\downarrow$.*

A context $\mathcal{C}$ is a process containing one occurrence of a 'hole' $(\cdot)$. We write $\mathcal{C}(P)$ for the process given by replacing the hole in $\mathcal{C}$ by P. Formally, we define *MA* contexts as follows.

Definition 3.6. *The set of MA* process contexts *is given by the syntax:*

$$\mathcal{C} \ ::= \ (\cdot) \ \big|\ \mathcal{C} \mid P \ \big|\ P \mid \mathcal{C} \ \big|\ n[\mathcal{C}\,] \ \big|\ (\mathsf{new}\,n)\,\mathcal{C}$$

where $P \in \mathsf{MA}$ and $n \in \mathsf{N}$.

Definition 3.7. Barbed bisimulation *is the largest symmetric relation $\simeq \subseteq \mathsf{MA} \times \mathsf{MA}$ such that $P \simeq Q$ implies:*

- *for each $n \in \mathsf{N}$, $P \Downarrow^* n$ if and only if $Q \Downarrow^* n$;*
- *for any context $\mathcal{C}$, if $\mathcal{C}(P) \rightarrow^* P'$, then there exists $Q' \in \mathsf{MA}$ such that $\mathcal{C}(Q) \rightarrow^* Q'$ and $P' \simeq Q'$.*

3.2. Ambient Logic

We introduce the standard Ambient Logic of Cardelli and Gordon [24, 48]. The kinds of properties that can be expressed are of the form: 'is process P located at the ambient named n?' or 'is there a secret name shared between two processes P and Q?' The syntax of the Ambient Logic is given below where Var denotes a set of variables.

Definition 3.8. *The Ambient Logic, written* AL*, is given by the syntax:*

$$
\begin{array}{lcll}
\phi, \psi & ::= & \mathbf{T} & \textit{true} \\
 & \mid & \neg\phi & \textit{negation} \\
 & \mid & \phi \vee \phi & \textit{disjunction} \\
 & \mid & \mathbf{0} & \textit{void} \\
 & \mid & \eta[\phi] & \textit{location} \\
 & \mid & \phi \mid \psi & \textit{composition} \\
 & \mid & \text{⌑}\phi & \textit{some where} \\
 & \mid & \Diamond\phi & \textit{some time} \\
 & \mid & \phi@\eta & \textit{location adjunct} \\
 & \mid & \phi \triangleright \psi & \textit{composition adjunct} \\
 & \mid & \forall x.\, \phi & \textit{universal quantification} \\
 & \mid & \eta \circledR \phi & \textit{revelation} \\
 & \mid & \phi \oslash \eta & \textit{revelation adjunct}
\end{array}
$$

where $x \in \mathsf{Var}$ *and* $\eta \in \mathsf{N} \cup \mathsf{Var}$*.*

The first three connectives are standard in propositional logic. The remaining connectives are tailored to express properties about ambient processes relative to both *time* and *space*. For example:

- $n[\phi]$ expresses that *here and now* there is an ambient called n inside which ϕ holds;
- $\text{⌑}\,\phi$ expresses that *some where*, i.e. after traversing down through a number of ambients, ϕ holds;
- $\Diamond\phi$ expresses that *some time*, i.e. after a finite number of reductions, ϕ holds;
- $\phi@n$ expresses that *in context* n, i.e. after being placed inside the ambient n, ϕ holds;
- $\phi \triangleright \psi$ expresses that ψ holds in *in any context satisfying* ϕ.

The set of free variables $\mathsf{fv}(\phi)$ is defined in the standard manner, bearing in mind that the only binding operator is $\forall x\phi$. Furthermore, a formula is closed if all the variables appearing in the formula are bound.

The Ambient Calculus serves as a model for the Ambient Logic. The relationship between the calculus and the language is expressed by the following satisfaction relation.

Definition 3.9. *The satisfaction relation* $\models\subseteq \mathsf{MA} \times \mathsf{AL}$*, written* $P \models \phi$ *is defined as* *follows:*

$$
\begin{array}{rcl}
P \models \mathbf{T} & \text{for all } P \in \mathsf{MA} & \\
P \models \neg\phi & \Leftrightarrow & P \not\models \phi \\
P \models \phi \vee \psi & \Leftrightarrow & P \models \phi \vee P \models \psi \\
P \models \mathbf{0} & \Leftrightarrow & P \equiv \mathbf{0} \\
P \models n[\phi] & \Leftrightarrow & \exists Q \in \mathsf{MA}.\, (P \equiv n[\,Q\,]) \wedge (Q \models \phi) \\
P \models \phi \mid \psi & \Leftrightarrow & \exists Q, R \in \mathsf{MA}.\, (P \equiv (Q \mid R)) \wedge (Q \models \phi) \wedge (R \models \psi) \\
P \models \text{⌑}\phi & \Leftrightarrow & \exists Q \in \mathsf{MA}.\, (P \downarrow^{*} Q) \wedge (Q \models \phi) \\
P \models \Diamond\phi & \Leftrightarrow & \exists Q \in \mathsf{MA}.\, (P {\rightarrow^{*}} Q) \wedge (Q \models \phi) \\
P \models \phi @ n & \Leftrightarrow & n[\,P\,] \models \phi \\
P \models \phi \triangleright \psi & \Leftrightarrow & \forall Q \in \mathsf{MA}.\, (Q \models \phi) \Rightarrow (P \mid Q \models \psi) \\
P \models \forall x.\, \phi & \Leftrightarrow & \forall n \in \mathsf{N}.\, P \models \phi\{n/x\} \\
P \models n \circledR \phi & \Leftrightarrow & \exists Q \in \mathsf{MA}.\, (P \equiv (\mathsf{new}\, n)\, Q) \wedge (Q \models \phi) \\
P \models \phi \oslash n & \Leftrightarrow & (\mathsf{new}\, n)\, P \models \phi
\end{array}
$$

where $\phi\{n/x\}$ denotes the standard substitution of every free occurrence of x in ϕ by n.

The equivalence relation over *MA* induced by this logic is given by the following definition.

Definition 3.10. *For any $P, Q \in \mathsf{MA}$, we write $P =_{\mathsf{AL}} Q$ when, for any closed formula $\phi \in \mathsf{AL}$, we have $P \models \phi$ if and only if $Q \models \phi$.*

One interesting question is whether or not the equivalence relation $=_{\mathsf{AL}}$ coincides with any of the known relations, such as structural congruence (Definition 3.2) or barbed bisimulation (Definition 3.7). In the Ambient Logic, the relation induced by the logic coincides with structural congruence, at least for finite processes (the fragment of *MA* without recursion or replication).

Theorem 3.11 ([39])**.** *Let P, Q be Mobile Ambient processes.*

1. *If $P \equiv Q$, then $P =_{\mathsf{AL}} Q$.*
2. *If P and Q are* finite *processes and $P =_{\mathsf{AL}} Q$, then $P \equiv Q$.*

3.3. Example: Agent Crossing a Firewall

To show which kind of situations are naturally modelled in *MA*, we present the example of an agent crossing a firewall [24]. The idea is that a client requesting services over a network has to authenticate using security measures such as passwords, which are represented here as the secret names k, k', k''. Formally, the processes representing the firewall and agent are given by:

$$
\begin{array}{rcl}
\mathit{Firewall} & \stackrel{\text{def}}{=} & (\mathsf{new}\, w)\, (w[\, k[\, \mathsf{out}\, w.\mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid \mathsf{open}\, k'.\mathsf{open}\, k''.P\,]) \\
\mathit{Agent} & \stackrel{\text{def}}{=} & k'[\, \mathsf{open}\, k.k''[\, Q\,]\,]
\end{array}
$$

Using the structural congruence and reduction rules given in Definition 3.2 and Definition 3.3 respectively we have that:

$$\begin{array}{rl}
\multicolumn{2}{l}{\mathit{Firewall} \mid \mathit{Agent}} \\
= & (\mathsf{new}\, w)\ (w[\, k[\,\mathsf{out}\, w.\mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid \mathsf{open}\, k'.\mathsf{open}\, k''.P\,]) \mid k'[\,\mathsf{open}\, k.k''[\,Q\,]\,] \\
\equiv & (\mathsf{new}\, w)\ (w[\, k[\,\mathsf{out}\, w.\mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid \mathsf{open}\, k'.\mathsf{open}\, k''.P\,] \mid k'[\,\mathsf{open}\, k.k''[\,Q\,]\,]) \\
\rightarrow & (\mathsf{new}\, w)\ (w[\,\mathsf{open}\, k'.\mathsf{open}\, k''.P\,] \mid k[\,\mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid k'[\,\mathsf{open}\, k.k''[\,Q\,]\,]) \\
\rightarrow & (\mathsf{new}\, w)\ (w[\,\mathsf{open}\, k'.\mathsf{open}\, k''.P\,] \mid k'[\, k[\,\mathsf{in}\, w.\mathbf{0}\,] \mid \mathsf{open}\, k.k''[\,Q\,]\,]) \\
\rightarrow & (\mathsf{new}\, w)\ (w[\,\mathsf{open}\, k'.\mathsf{open}\, k''.P\,] \mid k'[\,\mathsf{in}\, w.\mathbf{0} \mid k''[\,Q\,]\,]) \\
\rightarrow & (\mathsf{new}\, w)\ (w[\,\mathsf{open}\, k'.\mathsf{open}\, k''.P \mid k'[\,\mathbf{0} \mid k''[\,Q\,]\,]\,]) \\
\equiv & (\mathsf{new}\, w)\ (w[\,\mathsf{open}\, k'.\mathsf{open}\, k''.P \mid k'[\, k''[\,Q\,]\,]\,]) \\
\rightarrow & (\mathsf{new}\, w)\ (w[\,\mathsf{open}\, k''.P \mid k''[\,Q\,]\,]) \\
\rightarrow & (\mathsf{new}\, w)\ (w[\,P \mid Q\,])\ .
\end{array}$$

which can be interpreted as the agent Q, who knew the passwords k, k', k'', successfully crossed the firewall.

4. Probabilistic Mobile Ambients (*PMA*)

In this section we define the Probabilistic Ambient Calculus which extends the classical ambient calculus to allow for probabilistic behaviour. Following the spirit of the Ambient Calculus where there is no choice operator, instead of adding a separate probabilistic choice operator we modify the prefix operator to incorporate probabilistic behaviour. More precisely, with respect to the syntax of the Mobile Ambients given in Definition 3.1, we replace the prefix operator $M.P$ with the (guarded) probabilistic choice operator [11]:

$$M.{\textstyle\sum_{i \in I}} p_i.P_i$$

where I is an indexing set and p_i is a real number in the interval $(0, 1]$ denoting the probability that after the action corresponding to the capability M is performed the process becomes P_i. In the case when the indexing set is finite, that is $I = \{i_1, \ldots, i_n\}$ for some $n \in \mathrm{I\!N}$, then we also write this operator as:

$$M.\,(p_{i_1}.P_{i_1} + p_{i_2}.P_{i_2} + \cdots + p_{i_n}.P_{i_n})\ .$$

The definition of the syntax of the Probabilistic Ambient Calculus is given below. Note that the other operators are the same as for the Ambient Calculus and are discussed in Section 3.1.

Definition 4.1. *The set of* process terms, PMA, *of PMA is given by the syntax:*

$$\begin{array}{rcl}
M & ::= & \mathsf{in}\, n \ \big|\ \mathsf{out}\, n \ \big|\ \mathsf{open}\, n \\
P, Q & ::= & \mathbf{0} \ \big|\ n[\,P\,] \ \big|\ P \mid Q \ \big|\ (\mathsf{new}\, n)\, P \ \big|\ \mathsf{A} \ \big|\ \mathit{fix}_{\mathsf{A}} P \ \big|\ M.{\textstyle\sum_{i \in I}} p_i.P_i
\end{array}$$

where $n \in \mathsf{N}$, $\mathsf{A} \in \mathsf{Id}$ and $\sum_{i \in I} p_i$ is a summation over a countable indexing set I such that $p_i \in (0, 1]$ for all $i \in I$ and $\sum_{i \in I} p_i = 1$.

For example, the following process of *PMA* represents a system that, after entering an ambient m, will with probability 0.25 become the process P while with probability 0.75 become the process Q:

$$\mathsf{open}\, m.\left(\tfrac{1}{4}.P + \tfrac{3}{4}.Q\right) \;.$$

We now introduce some notation that is required in the remainder of the paper.

- For any summation $\sum_{i\in I} p_i.P_i$ over a countable indexing set I such that $p_i \in (0,1]$, $\sum_{i\in I} p_i{=}1$ and $P_i \in \mathsf{PMA}$, let $[\![\sum_{i\in I} p_i.P_i]\!]$ denote the distribution over PMA where for any $T \in \mathsf{PMA}$: $[\![\sum_{i\in I} p_i.P_i]\!](T) = \sum_{i\in I \wedge P_i = T} p_i$.
- For any $\mu,\nu \in \mathsf{Distr}(\mathsf{PMA})$ and $Q \in \mathsf{PMA}$, let $(\mu \mid Q)$, $(Q \mid \mu)$ and $(\mu \mid \nu)$ denote the distributions over PMA where for any $T \in \mathsf{PMA}$:
$$(\mu \mid Q)(T) = \begin{cases} \mu(T') & \text{if } T = T' \mid Q \\ 0 & \text{otherwise} \end{cases} \qquad (Q \mid \mu)(T) = \begin{cases} \mu(T') & \text{if } T = Q \mid T' \\ 0 & \text{otherwise} \end{cases}$$
$$\text{and } \;(\mu \mid \nu)(T) = \begin{cases} \mu(T_1)\cdot\nu(T_2) & \text{if } T = T_1 \mid T_2 \\ 0 & \text{otherwise.} \end{cases}$$
- For any $\mu \in \mathsf{Distr}(\mathsf{PMA})$ and $n \in \mathsf{N}$, let $(\mathsf{new}\, n)\,\mu$ and $n[\,\mu\,]$ denote the distributions over PMA such that for any $T \in \mathsf{PMA}$:
$$((\mathsf{new}\, n)\,\mu)(T) = \begin{cases} \mu(T') & \text{if } T{=}(\mathsf{new}\, n)\,T' \\ 0 & \text{otherwise} \end{cases} \qquad (n[\,\mu\,])(T) = \begin{cases} \mu(T') & \text{if } T{=}n[\,T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

4.1. Reduction Semantics

We define the reduction semantics for the Probabilistic Mobile Ambients in terms of probabilistic automata [22, 42]. The basic idea is to represent steps of computation as a relation from processes to *probability distributions.* The main reason for working with reduction semantics is that a labelled transition system semantics for *MA* is very complicated, while the reduction semantics provides a means to deal with computation in a simple way. However, proofs in the reduction semantics are very difficult to handle due to the presence of structural congruence, while proofs in the labelled transition systems semantics are more straightforward as the definition is syntax-directed. Therefore we will also introduce the labelled transition system semantics in Section 4.3 and in Section 4.4 demonstrates that, up to τ actions, the two semantics coincide.

The definition of structural congruence for *MA* (see Definition 3.2) is extended to equate processes defined through the guarded probabilistic choice operator of *PMA*. More precisely, we add a single new structural congruence rule (STRUC PROB) which equates two processes defined through the probabilistic choice operator when the corresponding capabilities are the same and the structure of their distribution is the same, that is the probability of evolving into any process is the same. To take into account the possible replication of processes in the index set, for example in a process $M.\sum_{i\in I} p_i.P_i$ there may exist $i \neq j$ such that $P_i = P_j$, we sum the probabilities over the indices if the corresponding processes are equal. This rule will also mean that processes that are equivalent up to a permutation of the indexed set will be equated by structural congruence. For example, the following *PMA* processes will be equivalent under structural congruence:

$$\mathsf{open}\, m.\left(\tfrac{1}{4}.P + \tfrac{3}{4}.Q\right),\; \mathsf{open}\, m.\left(\tfrac{3}{4}.Q + \tfrac{1}{4}.P\right) \text{ and } \mathsf{open}\, m.\left(\tfrac{1}{4}.P + \tfrac{1}{4}.Q + \tfrac{1}{2}.Q\right) \;.$$

Formally we have the following definition for structural congruence over *PMA*.

Definition 4.2. *The* structural congruence *over the Probabilistic Mobile Ambients is the smallest congruence (equivalence relation preserved by all algebraic contexts) satisfying the equations:*

$$\boxed{\begin{array}{ll}
P \equiv P \mid \mathbf{0} & (\textsc{struc par zero}) \\
P \mid Q \equiv Q \mid P & (\textsc{struc par com}) \\
(P \mid Q) \mid R \equiv P \mid (Q \mid R) & (\textsc{struc par assoc}) \\
(\mathsf{new}\, n)\, \mathbf{0} \equiv \mathbf{0} & (\textsc{struc zero res}) \\
(\mathsf{new}\, m)\, (\mathsf{new}\, n)\, P \equiv (\mathsf{new}\, n)\, (\mathsf{new}\, m)\, P & (\textsc{struc res res}) \\
(\mathsf{new}\, n)\, (P \mid Q) \equiv P \mid (\mathsf{new}\, n)\, Q \textit{ if } n \notin \mathsf{fn}(P) & (\textsc{struc res par}) \\
(\mathsf{new}\, m)\, n[\, P\,] \equiv n[\, (\mathsf{new}\, m)\, P\,] \textit{ if } n \neq m & (\textsc{struc res amb}) \\
\mathit{fix}_{\mathsf{A}} P \equiv P\{\mathit{fix}_{\mathsf{A}} P / \mathsf{A}\} & (\textsc{struc rec}) \\
M.\sum_{i\in I} p_i.P_i \equiv M.\sum_{j\in J} q_j.Q_j \textit{ if } [\![\sum_{i\in I} p_i.P_i]\!] = [\![\sum_{i\in I} q_i.Q_i]\!] & (\textsc{struc prob})
\end{array}}$$

Furthermore, in standard manner we lift the relation $\equiv$ *to distributions over* PMA*:* $\mu \equiv \nu$ *if and only if* $\mu([P]_\equiv) = \nu([P]_\equiv)$ *for all equivalence classes* $[P]_\equiv \subseteq \mathsf{PMA}$ *of* $\equiv$*.*

Combining the fact that $\equiv$ is congruence and (STRUC PROB) it follows that:

$$M.\sum_{i\in I} p_i.P_i \equiv M.\sum_{j\in J} q_j.Q_j \quad \text{if} \quad \sum_{i\in I \wedge P_i \equiv T} p_i = \sum_{j\in J \wedge Q_j \equiv T} q_j \quad \text{for all } \; T \in \mathsf{PMA}$$

i.e. structural congruence identifies processes defined through the probabilistic choice operator when their capabilities are the same and the structure of their distributions are the same up to structural congruence. Note that this matches the equivalence over distributions induced by structural congruence given in Definition 4.2. For example the following processes are structurally congruent.

$$\mathsf{open}\, m.\left(\tfrac{1}{4}.P + \tfrac{3}{4}.Q\right) \quad \text{and} \quad \mathsf{open}\, m.\left(\tfrac{1}{4}.P + \tfrac{1}{4}.Q + \tfrac{1}{2}.(Q \mid \mathbf{0})\right) \; .$$

We next define the reduction semantics for the Probabilistic Ambient Calculus.

Definition 4.3. *The* reduction semantics *for PMA is the probabilistic automaton* (PMA, $\rightarrow$) *where the* probabilistic reduction relation $\rightarrow \subseteq \mathsf{PMA} \times \mathsf{Distr}(\mathsf{PMA})$ *is the smallest relation satisfying the rules in Figure 1.*

The main difference between the reduction rules for the Probabilistic Ambient Calculus and the original reduction rules (Definition 3.3) is that in the probabilistic case processes evolve into distributions over processes as opposed to single processes. For the rules (RED IN), (RED OUT) and (RED OPEN), because of possible replication of processes in the index set, there is a summation over the indices which correspond to the same process. For example applying the rule (RED OPEN) we have:

$$\mathsf{open}\, m.\left(\tfrac{1}{4}.P + \tfrac{1}{4}.Q + \tfrac{1}{2}.Q\right) \mid m[\, \mathbf{0}\,] \rightarrow \mu$$

$$\text{where} \;\; \mu(T) = \begin{cases} \frac{1}{4} & \text{if } T = P \mid \mathbf{0} \\ \frac{3}{4} & \text{if } T = Q \mid \mathbf{0} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{array}{rl}
(\textsc{red in}) & m\Big[\,\mathsf{in}\,n.\sum_{i\in I}p_i.P_i\;\Big|\;Q\,\Big]\;\Big|\;n[\,R\,]\rightarrow n[\,m[\,[\![\sum_{i\in I}p_i.P_i]\!]\mid Q\,]\mid R\,]\\
(\textsc{red out}) & n\Big[\,m\Big[\,\mathsf{out}\,n.\sum_{i\in I}p_i.P_i\;\Big|\;Q\,\Big]\;\Big|\;R\,\Big]\rightarrow m[\,[\![\sum_{i\in I}p_i.P_i]\!]\mid Q\,]\mid n[\,R\,]\\
(\textsc{red open}) & \mathsf{open}\,n.\sum_{i\in I}p_i.P_i\;\Big|\;n[\,Q\,]\rightarrow[\![\sum_{i\in I}p_i.P_i]\!]\mid Q\\
(\textsc{red par}) & P\mid Q\rightarrow\mu\mid Q\ \text{ if }\ P\rightarrow\mu'\\
(\textsc{red restr}) & (\mathsf{new}\,n)\,P\longrightarrow(\mathsf{new}\,n)\,\mu\ \text{ if }\ P\rightarrow\mu\\
(\textsc{red amb}) & n[\,P\,]\rightarrow n[\,\mu\,]\ \text{ if }\ P\rightarrow\mu\\
(\textsc{red cong}) & P\rightarrow\mu\ \text{ if }\ P\equiv P',\ \ P'\rightarrow\mu'\ \text{ and }\ \mu'\equiv\mu
\end{array}$$

Figure 1: Reduction Rules for *PMA*

The remaining rules, namely (RED PAR), (RED RESTR), (RED AMB) and (RED CONG), are the generalised versions of the rules for *MA* given in Definition 3.3 to distributions. For example, applying (RED PAR) to the example above we have:

$$\big(\mathsf{open}\,m.\,\big(\tfrac{1}{4}.P+\tfrac{1}{4}.Q+\tfrac{1}{2}.Q\big)\mid m[\,\mathbf{0}\,]\big)\mid Q\rightarrow\mu$$

$$\text{where }\ \mu(T)=\begin{cases}\frac{1}{4} & \text{if } T=(P\mid\mathbf{0})\mid Q\\ \frac{3}{4} & \text{if } T=(Q\mid\mathbf{0})\mid Q\\ 0 & \text{otherwise.}\end{cases}$$

The original Ambient Calculus can be encoded in the Probabilistic Ambient Calculus, by simply mapping any term of the form $M.P$ to $M.1.P$. Under this encoding structural congruence and the reduction semantics given here and for the Ambient Calculus given in Section 3.1 are equivalent. Note that to simplify the presentation, we use the original Ambient Calculus notation and reduction rules where applicable, that is we write probabilistic ambient components of the form $M.1.P$ as $M.P$.

4.2. Examples of PMA specifications

Below we present probabilistic extensions of some Mobile Ambient examples given in [24].

Client-Server. We consider here a scenario in distributed systems, where a client can probabilistically choose to use one of several servers. In the example below, the client probabilistically chooses to use $Server_1$ with probability $\frac{1}{3}$ and $Server_2$ with probability $\frac{2}{3}$:

$$\begin{array}{rcl}
Client & \overset{\text{def}}{=} & c\big[\,\mathsf{in}\,s.\,\big(\tfrac{1}{3}.\mathsf{open}\,s_1.Client+\tfrac{2}{3}.\mathsf{open}\,s_2.Client\big)\,\big]\\
Servers & \overset{\text{def}}{=} & s[\,s_1[\,Server_1\,]\mid s_2[\,Server_2\,]\mid\mathsf{open}\,c.\mathbf{0}\,]\\
System & \overset{\text{def}}{=} & Client\mid Servers
\end{array}$$

Then:

$$System=c\big[\,\mathsf{in}\,s.\,\big(\tfrac{1}{3}.\mathsf{open}\,s_1.Client+\tfrac{2}{3}.\mathsf{open}\,s_2.Client\big)\,\big]\mid Servers\rightarrow\mu$$

where for any $T\in\mathsf{PMA}$:

$$\mu(T)=\begin{cases}\frac{1}{3} & \text{if } T=s[\,c[\,\mathsf{open}\,s_1.Client\,]\mid s_1[\,Server_1\,]\mid s_2[\,Server_2\,]\mid\mathsf{open}\,c.\mathbf{0}\,]\\ \frac{2}{3} & \text{if } T=s[\,c[\,\mathsf{open}\,s_2.Client\,]\mid s_1[\,Server_1\,]\mid s_2[\,Server_2\,]\mid\mathsf{open}\,c.\mathbf{0}\,]\\ 0 & \text{otherwise.}\end{cases}$$

Then, for example when server 1 is chosen we have the following reduction which reaches a situation where the client can interact with the first server.

$$\begin{aligned}
& s[\, c[\, \mathsf{open}\, s_1 . \mathit{Client}\,] \mid s_1[\, \mathit{Server}_1\,] \mid s_2[\, \mathit{Server}_2\,] \mid \mathsf{open}\, c.\mathbf{0}\,] \\
\equiv\ & s[\, c[\, \mathsf{open}\, s_1 . \mathit{Client}\,] \mid \mathsf{open}\, c.\mathbf{0} \mid s_1[\, \mathit{Server}_1\,] \mid s_2[\, \mathit{Server}_2\,]\,] \\
\rightarrow\ & s[\, \mathsf{open}\, s_1 . \mathit{Client} \mid s_1[\, \mathit{Server}_1\,] \mid s_2[\, \mathit{Server}_2\,]\,] \\
\rightarrow\ & s[\, \mathit{Client} \mid \mathit{Server}_1 \mid s_2[\, \mathit{Server}_2\,]\,]\ .
\end{aligned}$$

We can extend this model to the case where the client can (nondeterministically) either use a coin biased towards the first or the second server by amending the definition of the client in the following way:

$$\begin{aligned}
\mathit{Client}_1 &\stackrel{\text{def}}{=} o\big[\, \mathsf{in}\, s.\, \big(\tfrac{1}{3}.\mathsf{open}\, s_1 . \mathit{Client} + \tfrac{2}{3}.\mathsf{open}\, s_2 . \mathit{Client}\big)\,\big] \\
\mathit{Client}_2 &\stackrel{\text{def}}{=} o\big[\, \mathsf{in}\, s.\, \big(\tfrac{2}{3}.\mathsf{open}\, s_1 . \mathit{Client} + \tfrac{1}{3}.\mathsf{open}\, s_2 . \mathit{Client}\big)\,\big] \\
\mathit{Client} &\stackrel{\text{def}}{=} c[\, \mathsf{open}\, o.\mathbf{0} \mid \mathit{Client}_2 \mid \mathit{Client}_2\,]\ .
\end{aligned}$$

Agent Crossing a Firewall. Here we consider a probabilistic version of the Agent and Firewall example presented in Section 3.3. We assume that the passwords k and k' are public and that the remaining password k'' is secret. We modify the firewall such that it reacts to attempts to pass through the firewall in two ways: if the final password offered is correct then it allows the agent presenting the password to cross the firewall, if the password is incorrect then it will be isolated. The *PMA* processes representing this example are given below.

$$\begin{aligned}
\mathit{Firewall} &\stackrel{\text{def}}{=} (\mathsf{new}\, wi)\, (w[\, k[\, \mathsf{out}\, w.\mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation}) \\
\mathit{Allow} &\stackrel{\text{def}}{=} \mathsf{open}\, k''.P \\
\mathit{Block} &\stackrel{\text{def}}{=} \mathsf{enter}\, i.\mathbf{0} \\
\mathit{Isolation} &\stackrel{\text{def}}{=} i[\, \mathsf{open}\, w.\mathsf{open}\, l''.\mathbf{0}\,] \\
\mathit{Agent} &\stackrel{\text{def}}{=} k'[\, \mathit{Guess}\,] \\
\mathit{Guess} &\stackrel{\text{def}}{=} \mathsf{open}\, k.\, \big(\tfrac{1}{50}.k''[\, Q\,] + \tfrac{49}{50}.l''[\, Q\,]\big)
\end{aligned}$$

Following similar reduction steps to those given in in Section 3.3 we have that:

$$\begin{aligned}
& \mathit{Firewall} \mid \mathit{Agent} \equiv \\
& (\mathsf{new}\, wi)\ (w[\, k[\, \mathsf{out}\, w.\mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k'[\, \mathit{Guess}\,]) \\
& \rightarrow\ (\mathsf{new}\, wi)\ (w[\, \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k[\, \mathsf{in}\, k'.\mathsf{in}\, w.\mathbf{0}\,] \mid k'[\, \mathit{Guess}\,]) \\
& \rightarrow\ (\mathsf{new}\, wi)\ (w[\, \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k'[\, k[\, \mathsf{in}\, w.\mathbf{0}\,] \mid \mathit{Guess}\,]) \\
& \rightarrow\ \mu
\end{aligned}$$

where, for any $T \in \mathsf{PMA}$, $\mu(T)$ equals

$$\begin{cases}
\frac{1}{50} & \text{if } T = (\mathsf{new}\, wi)\ (w[\, \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k'[\, \mathsf{in}\, w.\mathbf{0} \mid k''[\, Q\,]\,]) \\
\frac{49}{50} & \text{if } T = (\mathsf{new}\, wi)\ (w[\, \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k'[\, \mathsf{in}\, w.\mathbf{0} \mid l''[\, Q\,]\,]) \\
0 & \text{otherwise.}
\end{cases}$$

Considering each case in turn:

- With probability 0.02, the system can evolve to a situation where the agent has successfully passed through the firewall. In this case we have the following reduction steps:

$$\begin{aligned}
&(\mathsf{new}\, wi)\ (w[\,\mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k'[\,\mathsf{in}\, w.\mathbf{0} \mid k''[\,Q\,]\,])\\
\equiv\ &(\mathsf{new}\, wi)\ (k'[\,\mathsf{in}\, w.\mathbf{0} \mid k''[\,Q\,]\,] \mid w[\,\mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation})\\
\rightarrow\ &(\mathsf{new}\, wi)\ (w[\,k'[\,\mathbf{0} \mid k''[\,Q\,]\,] \mid \mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation})\\
\equiv\ &(\mathsf{new}\, wi)\ (w[\,\mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block}) \mid k'[\,k''[\,Q\,]\,]\,] \mid \mathit{Isolation})\\
\rightarrow\ &(\mathsf{new}\, wi)\ (w[\,(\mathit{Allow} \mid \mathit{Block}) \mid k''[\,Q\,]\,] \mid \mathit{Isolation})\\
=\ &(\mathsf{new}\, wi)\ (w[\,\mathsf{open}\, k''.P \mid \mathit{Block} \mid k''[\,Q\,]\,] \mid \mathit{Isolation})\\
\equiv\ &(\mathsf{new}\, wi)\ (w[\,\mathsf{open}\, k''.P \mid k''[\,Q\,] \mid \mathit{Block}\,] \mid \mathit{Isolation})\\
\rightarrow\ &(\mathsf{new}\, wi)\ (w[\,P \mid Q \mid \mathit{Block}\,] \mid \mathit{Isolation})\ .
\end{aligned}$$

- With probability 0.98, the system can evolve to the situation where the agent Q does not gain access through the firewall and is put in isolation. For this scenario we have the following reduction steps (we have omitted the steps that are the same as the case above):

$$\begin{aligned}
&(\mathsf{new}\, wi)\ (w[\,\mathsf{open}\, k'.(\mathit{Allow} \mid \mathit{Block})\,] \mid \mathit{Isolation} \mid k'[\,\mathsf{in}\, w.\mathbf{0} \mid l''[\,Q\,]\,])\\
&\qquad\qquad\qquad \cdots\\
\rightarrow\ &(\mathsf{new}\, wi)\ (w[\,(\mathit{Allow} \mid \mathit{Block}) \mid l''[\,Q\,]\,] \mid \mathit{Isolation})\\
=\ &(\mathsf{new}\, wi)\ (w[\,(\mathit{Allow} \mid \mathsf{enter}\, i.\mathbf{0}) \mid l''[\,Q\,]\,] \mid i[\,\mathsf{open}\, w.\mathsf{open}\, l''.\mathbf{0}\,])\\
\equiv\ &(\mathsf{new}\, wi)\ (w[\,\mathsf{enter}\, i.\mathbf{0} \mid \mathit{Allow} \mid l''[\,Q\,]\,] \mid i[\,\mathsf{open}\, w.\mathsf{open}\, l''.\mathbf{0}\,])\\
\rightarrow\ &(\mathsf{new}\, wi)\ (i[\,w[\,\mathbf{0} \mid \mathit{Allow} \mid l''[\,Q\,]\,] \mid \mathsf{open}\, w.\mathsf{open}\, l''.\mathbf{0}\,])\\
\equiv\ &(\mathsf{new}\, wi)\ (i[\,\mathsf{open}\, w.\mathsf{open}\, l''.\mathbf{0} \mid w[\,l''[\,Q\,] \mid \mathit{Allow}\,]\,])\\
\rightarrow\ &(\mathsf{new}\, wi)\ (i[\,\mathsf{open}\, l''.\mathbf{0} \mid l''[\,Q\,] \mid \mathit{Allow}\,])\\
\rightarrow\ &(\mathsf{new}\, wi)\ (i[\,\mathbf{0} \mid Q \mid \mathit{Allow}\,])\\
\equiv\ &(\mathsf{new}\, wi)\ (i[\,Q \mid \mathit{Allow}\,])\ .
\end{aligned}$$

4.3. Probabilistic Labelled Transition System Semantics

Due to the nature of the Mobile Ambients, in order to specify the computational steps that represent one ambient moving into another, or an ambient moving outside another, processes are put on the actions of the transition relation as in [15, 27]. This kind of labelled transition system is called a *second order labelled transition system* [34, 35]. However, since we are in the probabilistic setting, distributions over processes are required in the actions of the transition relations as opposed to processes. Formally we have the following definition.

Definition 4.4. *The set of first order actions $\mathcal{A}ct$ is defined by the syntax:*

$$\alpha ::= \mathsf{in}\, n \mid \mathsf{out}\, n \mid \mathsf{open}\, n \mid \overline{\mathsf{open}\, n}$$

where $n \in \mathsf{N}$. Furthermore, the set of second order actions $\mathcal{A}ct^{\star}$ is defined by the syntax:

$$\gamma ::= \tilde{k}\, \mathsf{enter}\, n\, (\nu) \mid \overline{\mathsf{enter}\, n}\, (\nu) \mid \tilde{k}\, \mathsf{exit}\, n\, (\nu)$$

where $\tilde{k}$ is a (possibly empty) sequence of names in N, $n \in \mathsf{N}$ and $\nu \in \mathsf{Distr}(\mathsf{PMA})$.

(ACT PFX) $M.\sum_{i\in I} p_i.P_i \xrightarrow{M} [\![\sum_{i\in I} p_i.P_i]\!]$
(ACT $\overline{\text{OPEN}}$) $n[\,P\,] \xrightarrow{\overline{\mathsf{open}\,n}} \eta_P$
(ACT PAR$_1$) $P \mid Q \xrightarrow{\alpha} \mu \mid Q$ if $P \xrightarrow{\alpha} \mu$
(ACT PAR$_2$) $P \mid Q \xrightarrow{\alpha} P \mid \mu$
(ACT REST) $(\mathsf{new}\,k)\,P \xrightarrow{\alpha} (\mathsf{new}\,k)\,\mu$ if $k \neq \mathsf{name}(\alpha)$ and $P \xrightarrow{\alpha} \mu$
(ACT REC) $\mathit{fix}_{\mathsf{A}}P \xrightarrow{\alpha} \mu$ if $P\{\mathit{fix}_{\mathsf{A}}P/\mathsf{A}\} \xrightarrow{\alpha} \mu$

Figure 2: Labelled Transition System: First Order Actions

Before we give the labelled transition semantics for *PMA* we require the following definitions. First, let $\mathsf{name} : \mathcal{A}ct \cup \mathcal{A}ct^\star \to \mathsf{N}$ be the function where for any name n, sequence of names $\tilde{k}$ and distribution ν:

$$\mathsf{name}(\mathsf{in}\,n) = \mathsf{name}(\mathsf{out}\,n) = \mathsf{name}(\mathsf{open}\,n) = \mathsf{name}(\overline{\mathsf{open}\,n}) = n$$

and

$$\mathsf{name}(\tilde{k}\,\mathsf{enter}\,n\,(\nu)) = \mathsf{name}(\overline{\mathsf{enter}\,n}\,(\nu)) = \mathsf{name}(\tilde{k}\,\mathsf{exit}\,n\,(\nu)) = n\ .$$

Definition 4.5. *For any (possibly empty) sequence of names $\tilde{k}$ we denote by $\{\tilde{k}\}$ the set of names appearing in $\tilde{k}$. Furthermore, for any $P \in \mathsf{PMA}$:*

$$(\mathsf{new}\,\tilde{k})\,P \overset{\text{def}}{=} \begin{cases} (\mathsf{new}\,k_1)\cdots(\mathsf{new}\,k_n)\,P & \text{if } \tilde{k} = k_1\cdots k_n \text{ for some } n{>}0 \\ P & \text{otherwise} \end{cases}$$

and, for any set of names $N \subseteq \mathsf{N}$, let $\tilde{k}\!\restriction_N$ denote the sequence $\tilde{k}$ restricted to only those names in N.

We are now in a position to define the *labelled probabilistic transition semantics* for *PMA*.

Definition 4.6. *The* probabilistic labelled transition system semantics *for PMA is the probabilistic automaton* $(\mathsf{PMA}, \mathcal{A}ct\cup\mathcal{A}ct^\star\cup\{\tau\}, \rightarrow)$ *where the* labelled probabilistic transition relation $\rightarrow \subseteq \mathsf{PMA} \times (\mathcal{A}ct \cup \mathcal{A}ct^\star \cup \{\tau\}) \times \mathsf{Distr}(\mathsf{PMA})$ *is the smallest relation satisfying the rules in Figures 2–4.*

Note that, as in the reduction semantics, due to possible replication of processes in the index set there is again a sum over the indices which correspond to the same process.

We now explain the rules of Figures 2–4 induced by the capability in (entering an ambient). An example of entry into an ambient is given by the following reduction:

$$(\mathsf{new}\,m)\,(\mathsf{new}\,k)\,\big(m\big[\,\mathsf{in}\,n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q\,\big] \mid R\big) \mid n[\,S\,] \rightarrow \rho$$

where supposing $k \notin \mathsf{fn}(Q)$ and $k \notin \mathsf{fn}(P_i)$ for any $i \in I$, for any $T \in \mathsf{PMA}$:

$$\rho(T) = \begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,m)\,(n[\,m[\,T' \mid Q\,] \mid S\,] \mid (\mathsf{new}\,k)\,R) \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

$$\begin{array}{l}
(\textsc{act}^\star\ \textsc{enter})\ m[\,P\,] \xrightarrow{\langle\rangle \mathsf{enter}\, n(m[\,\mu\,])} \eta_{\mathbf{0}}\ \text{ if }\ P \xrightarrow{\mathsf{in}\, n} \mu \\
(\textsc{act}^\star\ \textsc{exit})\ m[\,P\,] \xrightarrow{\langle\rangle \mathsf{exit}\, n(m[\,\mu\,])} \eta_{\mathbf{0}}\ \text{ if }\ P \xrightarrow{\mathsf{out}\, n} \mu \\
(\textsc{act}^\star\ \overline{\textsc{enter}})\ n[\,P\,] \xrightarrow{\overline{\mathsf{enter}\, n}(\nu)} n[\,P \mid \nu\,] \\
(\textsc{act}^\star\ \textsc{par}_1)\ P \mid Q \xrightarrow{\tilde{k}\beta(\nu)} \mu \mid Q\ \text{ if }\ \mathsf{fn}(Q){\cap}\{\tilde{k}\}{=}\emptyset \text{ and } P \xrightarrow{\tilde{k}\beta(\nu)} \mu \\
(\textsc{act}^\star\ \textsc{par}_2)\ P \mid Q \xrightarrow{\tilde{k}\beta(\nu)} P \mid \mu\ \text{ if }\ \mathsf{fn}(P){\cap}\{\tilde{k}\}{=}\emptyset \text{ and } Q \xrightarrow{\tilde{k}\beta(\nu)} \mu \\
(\textsc{act}^\star\ \textsc{rest}_1)\ (\mathsf{new}\, k)\, P \xrightarrow{k\tilde{k}\beta(\nu)} \mu\ \text{ if } k{\neq}\mathsf{name}(\beta),\ k{\in}\mathsf{fn}(\nu) \text{ and } P \xrightarrow{\tilde{k}\beta(\nu)} \mu \\
(\textsc{act}^\star\ \textsc{rest}_2)\ (\mathsf{new}\, k)\, P \xrightarrow{\tilde{k}\beta(\nu)} (\mathsf{new}\, k)\, \mu\ \text{ if } k{\neq}\mathsf{name}(\beta),\ k{\notin}\mathsf{fn}(\nu) \text{ and } P \xrightarrow{\tilde{k}\beta(\nu)} \mu \\
(\textsc{act}^\star\ \textsc{rec})\ \mathit{fix}_{\mathsf{A}} P \xrightarrow{\tilde{k}\beta(\nu)} \mu\ \text{ if }\ P\{\mathit{fix}_{\mathsf{A}} P/\mathsf{A}\} \xrightarrow{\tilde{k}\beta(\nu)} \mu
\end{array}$$

Figure 3: Labelled Transition System: Second Order Actions

$$\begin{array}{l}
(\tau\ \textsc{enter}_1)\ P \mid Q \xrightarrow{\tau} (\mathsf{new}\, \tilde{k}\tilde{l})\, (\mu_1 \mid \mu_2) \text{ if } P \xrightarrow{\tilde{k}\, \mathsf{enter}\, n(\nu)} \mu_1 \text{ and } Q \xrightarrow{\tilde{l}\, \overline{\mathsf{enter}\, n}(\nu)} \mu_2 \\
(\tau\ \textsc{enter}_2)\ P \mid Q \xrightarrow{\tau} (\mathsf{new}\, \tilde{k}\tilde{l})\, (\mu_1 \mid \mu_2)\ \text{ if }\ P \xrightarrow{\tilde{k}\, \overline{\mathsf{enter}\, n}(\nu)} \mu_1 \text{ and } Q \xrightarrow{\tilde{l}\, \mathsf{enter}\, n(\nu)} \mu_2 \\
(\tau\ \textsc{exit})\ n[\,P\,] \xrightarrow{\tau} (\mathsf{new}\, \tilde{k})\, (\nu \mid n[\,\mu\,])\ \text{ if }\ P \xrightarrow{\tilde{k}\mathsf{exit}\, n(\nu)} \mu \\
(\tau\ \textsc{open}_1)\ P \mid Q \xrightarrow{\tau} \mu_1 \mid \mu_2\ \text{ if }\ P \xrightarrow{\mathsf{open}\, n} \mu_1 \text{ and } Q \xrightarrow{\overline{\mathsf{open}\, n}} \mu_2 \\
(\tau\ \textsc{open}_2)\ P \mid Q \xrightarrow{\tau} \mu_1 \mid \mu_2\ \text{ if }\ P \xrightarrow{\overline{\mathsf{open}\, n}} \mu_1 \text{ and } Q \xrightarrow{\mathsf{open}\, n} \mu_2 \\
(\tau\ \textsc{par}_1)\ P \mid Q \xrightarrow{\tau} \mu \mid Q\ \text{ if }\ P \xrightarrow{\tau} \mu \\
(\tau\ \textsc{par}_2)\ P \mid Q \xrightarrow{\tau} P \mid \mu\ \text{ if }\ Q \xrightarrow{\tau} \mu \\
(\tau\ \textsc{rest})\ (\mathsf{new}\, k)\, P \xrightarrow{\tau} (\mathsf{new}\, k)\, \mu\ \text{ if }\ P \xrightarrow{\tau} \mu \\
(\tau\ \textsc{amb})\ n[\,P\,] \xrightarrow{\tau} n[\,\mu\,]\ \text{ if }\ P \xrightarrow{\tau} \mu \\
(\tau\ \textsc{rec})\ \mathit{fix}_{\mathsf{A}} P \xrightarrow{\tau} \mu\ \text{ if }\ P\{\mathit{fix}_{\mathsf{A}} P/\mathsf{A}\} \xrightarrow{\tau} \mu
\end{array}$$

Figure 4: Labelled Transition System: τ Actions

First considering the process inside the ambient m and applying ($\textsc{act pfx}$), ($\textsc{act par}$) and ($\textsc{act}^\star$ $\textsc{enter}$), we have:

$$m\big[\,\mathsf{in}\, n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q\,\big] \xrightarrow{\langle\rangle \mathsf{enter}\, n(\nu)} \mu_1$$

where for any $T \in \mathsf{PMA}$:

$$\nu(T) = \begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = m[\,T' \mid Q\,] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mu_1(T) = \begin{cases} 1 & \text{if } T = \mathbf{0} \\ 0 & \text{otherwise.} \end{cases}$$

In this labelled transition, the distribution ν represents the change caused by executing the capability represented by $\mathsf{enter}\, n$ (i.e. entering an ambient n). On the other hand, the distribution μ_1 represents the part of the process which is not affected by the capability. In this case, since the entire process moves under this capability, μ_1 is given by the point distribution $\eta_{\mathbf{0}}$. In general, since the distribution represents the part of the process that is not affected it will always be a point distribution, (this is clarified in Lemma 4.13). Next, applying $(\textsc{act}^{\star}\ \textsc{par}_2)$:

$$m\big[\,\mathsf{in}\; n.\textstyle\sum_{i\in I} p_i.P_i \mid Q\,\big] \mid R \xrightarrow{\langle\rangle \mathsf{enter}\; n(\nu)} \mu_1$$

where for any $T \in \mathsf{PMA}$:

$$\nu(T) = \begin{cases} \sum\limits_{i\in I\wedge P_i = T'} p_i & \text{if } T = m[\,T' \mid Q\,] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mu_1(T) = \begin{cases} 1 & \text{if } T = \mathbf{0} \mid R \\ 0 & \text{otherwise.} \end{cases}$$

The result is that the distribution μ_1 now also includes the process R, since it is unaffected. Using the fact that $k \notin \mathsf{fn}(Q)$ and $k \notin \mathsf{fn}(P_i)$ for any $i \in I$, applying $(\textsc{act}^{\star}\ \textsc{res}_1)$ followed by $(\textsc{act}^{\star}\ \textsc{res}_2)$ we have:

$$(\mathsf{new}\, m)\, (\mathsf{new}\, k)\ \big(m\big[\,\mathsf{in}\; n.\textstyle\sum_{i\in I} p_i.P_i \mid Q\,\big] \mid R\big) \xrightarrow{\langle m\rangle \mathsf{enter}\; n(\nu)} \mu_1$$

where for any $T \in \mathsf{PMA}$:

$$\nu(T) = \begin{cases} \sum\limits_{i\in I\wedge P_i = T'} p_i & \text{if } T = m[\,T' \mid Q\,] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mu_1(T) = \begin{cases} 1 & \text{if } T = (\mathsf{new}\, k)\, (\mathbf{0} \mid R) \\ 0 & \text{otherwise.} \end{cases}$$

Because k does not appear free in the part of the process that moves, the restriction $(\mathsf{new}\, k)$ only appears in the distribution μ_1. On the other hand, m does appears free (it is the ambient name of the process that moves), and hence it is included in the action.

From $(\textsc{act}^{\star}\ \overline{\textsc{enter}})$ it follows that $n[\,S\,] \xrightarrow{\overline{\mathsf{enter}\; n}(\nu)} \mu_2$ where

$$\mu_2(T) = \begin{cases} \nu(T') & \text{if } T = n[\,S \mid T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

In this case, the distribution μ_2 represents the outcome of a process entering the ambient $n[\,S\,]$ and then evolving according to the distribution ν. Now applying the rule $(\tau\ \textsc{enter}_1)$:

$$(\mathsf{new}\, m)\, (\mathsf{new}\, k)\ \big(m\big[\,\mathsf{in}\; n.\textstyle\sum_{i\in I} p_i.P_i \mid Q\,\big] \mid R\big)\ \mid n[\,S\,] \xrightarrow{\tau} \mu$$

where, using the definitions of ν, μ_1 and μ_2, for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \sum\limits_{i\in I\wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\, m)\, ((\mathsf{new}\, k)\, (\mathbf{0} \mid R) \mid n[\,S \mid m[\,T' \mid Q\,]\,]) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Since for any process $T' \in \mathsf{PMA}$:

$$\begin{aligned} &(\mathsf{new}\, m)\, ((\mathsf{new}\, k)\, (\mathbf{0} \mid R) \mid n[\,S \mid m[\,T' \mid Q\,]\,]) \\ &\quad \equiv\ (\mathsf{new}\, m)\, (n[\,m[\,T' \mid Q\,] \mid S\,] \mid (\mathsf{new}\, k)\, R), \end{aligned}$$

it follows that the distribution obtained through the labelled transition system rules (see (2)) and that obtained through the reduction rules (see (1)) are structurally congruent.

The cases for out and open follow a similar structure. For further details on the general structure of the transitions constructed through the first order rules in Figure 2 see Lemma 4.11 and Lemma 4.13, while Lemma 4.12 and Lemma 4.14 present the general structure of the transitions relating to second order rules given in Figure 3.

4.4. Relationship Between the Semantics

In this section we demonstrate that the reduction rules and the labelled transition system rules for Probabilistic Mobile Ambients are equivalent. More precisely, we demonstrate that, up to structural congruence, the reduction semantics coincides with the labelled transition system semantics restricted to only τ actions.

Theorem 4.7. *Let P be a PMA process.*

1. *If $P \rightarrow \mu$, then $P \xrightarrow{\tau} \nu$ for some $\nu \in \mathsf{Distr}(\mathsf{PMA})$ such that $\nu \equiv \mu$.*
2. *If $P \xrightarrow{\tau} \mu$, then $P \rightarrow \mu$.*

Before proceeding with the proof we require a number of preliminary lemmas and the following remark.

Remark 4.8. *We make the standard assumption that, for any process $P \mid Q$, both the bound names of P and free names of Q, and the free names of P and the bound names of Q are disjoint.*

Lemma 4.9. *For any $P, Q \in \mathsf{PMA}$ and $n \in \mathsf{N}$, the following sets encode sets of equivalence classes of $\equiv$:*

- $\{T \,|\, T \in \mathsf{PMA} \wedge (T \mid P) \equiv Q\}$;
- $\{T \,|\, T \in \mathsf{PMA} \wedge n[T\,] \equiv Q\}$;
- $\{T \,|\, T \in \mathsf{PMA} \wedge (\mathsf{new}\, n)\, T \equiv P\}$.

Proof. The proof follows from the fact that $\equiv$ is a congruence, i.e. is preserved by all algebraic contexts. For example, if $T, T' \in \mathsf{PMA}$ are in the same equivalence class of $\equiv$, since $\equiv$ is a congruence, we have $T \mid P \equiv T' \mid P$, and therefore, due to the transitivity of $\equiv$, it follows that $T \mid P \equiv Q$ if and only if $T' \mid P \equiv Q$. □

Lemma 4.10. *Let $P, P' \in \mathsf{PMA}$ such that $P \equiv P'$.*

- *If $P \xrightarrow{\gamma} \mu$ for some $\mu \in \mathsf{Distr}(\mathsf{PMA})$ and $\gamma \in \mathcal{A}ct \cup \mathcal{A}ct^{\star} \cup \{\tau\}$, then there exists $\nu \in \mathsf{Distr}(\mathsf{PMA})$ such that $P' \xrightarrow{\gamma} \nu$ and $\mu \equiv \nu$.*
- *If $P' \xrightarrow{\gamma} \nu$ for some $\nu \in \mathsf{Distr}(\mathsf{PMA})$ and $\gamma \in \mathcal{A}ct \cup \mathcal{A}ct^{\star} \cup \{\tau\}$, then there exists $\mu \in \mathsf{Distr}(\mathsf{PMA})$ such that $P \xrightarrow{\gamma} \mu$ and $\nu \equiv \mu$.*

Proof. The proof is by induction on $\equiv$. Note that we only consider the first half of the lemma since the second half follows similarly.

(struc par zero) In this case $P' = P \mid \mathbf{0}$. Below we consider the case when $\gamma \in \mathcal{A}ct$; the cases when $\gamma \in \mathcal{A}ct^{\star}$ or $\gamma = \tau$ follow similarly using either (act* par$_1$) or (τ par$_1$). Therefore, supposing that $\gamma \in \mathcal{A}ct$ and $P \xrightarrow{\gamma} \mu$, using (act par$_1$) we have $P \mid \mathbf{0} \xrightarrow{\gamma} \mu'$, where for any $T \in \mathsf{PMA}$

$$\mu'(T) = \begin{cases} \mu(T') & \text{if } T = T' \mid \mathbf{0} \\ 0 & \text{otherwise.} \end{cases}$$

Now applying (struc par zero) it follows that $\mu \equiv \mu'$ as required.

(STRUC PAR COM) In this case $P = Q \mid R$ and $P' = R \mid Q$ for some processes Q and R. The result follows by considering the possible derivations of the transition $P \xrightarrow{\gamma} \mu$, i.e. either (ACT PAR$_1$), (ACT PAR$_2$), (ACT* PAR$_1$), (ACT* PAR$_2$) ($\tau$ ENTER$_1$), (τ ENTER$_2$), (τ OPEN$_1$), (τ OPEN$_2$), (τ PAR$_1$) or (τ PAR$_2$), and then applying the symmetric rule to P' to construct a transition $P' \xrightarrow{\gamma} \mu'$ such that $\mu \equiv \mu'$.

For example, if $P \xrightarrow{\gamma} \mu$ is derived through the rule (τ OPEN$_1$), then $Q \xrightarrow{\mathsf{open}\, n} \mu_1$ and $R \xrightarrow{\overline{\mathsf{open}\, n}} \mu_2$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \mu_1(T_1) \cdot \mu_2(T_2) & \text{if } T = T_1 \mid T_2 \\ 0 & \text{otherwise.} \end{cases}$$

Now applying (τ OPEN$_2$), we have $P' \xrightarrow{\gamma} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \mu_2(T_2) \cdot \mu_1(T_1) & \text{if } T = T_2 \mid T_1 \\ 0 & \text{otherwise.} \end{cases}$$

The fact that $\mu \equiv \mu'$ then follows from the definitions of μ and μ' and applying (STRUC PAR COM).

(STRUC PAR ASSOC) In this case $P = (Q \mid R) \mid T$ and $P' = Q \mid (R \mid T)$ for some processes Q, R and T. The proof follows by considering the different ways that the transition $P \xrightarrow{\gamma} \mu$ is derived and then applying where necessary the appropriate symmetric rule to obtain a transition $P' \xrightarrow{\gamma} \mu'$ such that $\mu \equiv \mu'$.

(STRUC ZERO RES) In this case $P = (\mathsf{new}\, n)\, \mathbf{0}$ and $P' = \mathbf{0}$ for some name n, and since one cannot derive $P \xrightarrow{\gamma} \mu$ for any $\gamma \in \mathcal{A}ct \cup \mathcal{A}ct^\star \cup \{\tau\}$ and $\mu \in \mathsf{Distr}(\mathsf{PMA})$, the result holds in this case.

(STRUC RES RES) In this case $P = (\mathsf{new}\, m)\, (\mathsf{new}\, n)\, Q$ and $P' = (\mathsf{new}\, n)\, (\mathsf{new}\, m)\, Q$ for some names n and m and process Q. The result follows from the fact that the transition $P \xrightarrow{\gamma} \mu$ is derived via two rule applications from some transition $Q \xrightarrow{\gamma'} \rho$ and that, if one applies these rules to $Q \xrightarrow{\gamma'} \rho$ in the reverse order, then $P' \xrightarrow{\gamma} \mu'$ for some distribution μ' such that $\mu \equiv \mu'$.

(STRUC RES PAR) In this case $P = (\mathsf{new}\, n)\, (Q \mid R)$ and $P' = Q \mid (\mathsf{new}\, n)\, R$ for some processes Q and R and name n such that $n \notin \mathsf{fn}(Q)$. Similarly to the case above, this result follows from the fact that $P \xrightarrow{\gamma} \mu$ is derived by two rule applications from some transition $Q \xrightarrow{\gamma'} \rho$ and, applying these rules to $Q \xrightarrow{\gamma'} \rho$ in the reverse order, one obtains a transition $P' \xrightarrow{\gamma} \mu'$ such that $\mu \equiv \mu'$.

(STRUC RES AMB) In this case $P = (\mathsf{new}\, m)\, n[\, Q\,]$ and $P' = n[\, (\mathsf{new}\, m)\, Q\,]$ for some process Q and names n and m such that $n \neq m$. Again $P \xrightarrow{\gamma} \mu$ is derived through two rule applications from a transition $Q \xrightarrow{\gamma'} \rho$ and one can construct a transition $P' \xrightarrow{\gamma} \mu'$ such that $\mu \equiv \mu'$ by applying these rules in the reverse order to $Q \xrightarrow{\gamma'} \rho$.

(STRUC REC) In this case $P = \mathit{fix}_{\mathsf{A}} Q$ and $P' = Q\{\mathit{fix}_{\mathsf{A}} Q/\mathsf{A}\}$ for some identifier A and process Q and the result follows from the rules (ACT REC), (ACT* REC) and (τ REC). depending on whether $\gamma \in \mathcal{A}ct$, $\gamma \in \mathcal{A}ct^\star$ or $\gamma = \tau$.

(STRUC PROB) In this case $P = M.\sum_{i \in I} q_i.Q_i$ and $P' = M.\sum_{j \in J} r_j.R_j$ for some processes $M.\sum_{i \in I} q_i.Q_i$ and $M.\sum_{j \in J} r_j.R_j$ such that for any $T \in \mathsf{PMA}$:

$$\textstyle\sum_{i \in I \wedge Q_i = T} q_i = \sum_{j \in J \wedge R_j = T} r_j \ . \tag{3}$$

If $P \xrightarrow{\gamma} \mu$, then since the only rule that can be applied is (ACT PFX), $\gamma = M$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \textstyle\sum_{i \in I \wedge Q_i = T} q_i \ .$$

On the other hand, applying (ACT PFX) we have $P' \xrightarrow{M} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \textstyle\sum_{j \in J \wedge R_j = T} r_j \ .$$

It then follows from (3) and the definitions of μ and μ' that $\mu \equiv \mu'$ as required.

(STRUC REFL) If $P \equiv P'$ is derived through the fact that $\equiv$ is reflexive, then $P' = P$ and the result follows.

(STRUC SYMM) If $P \equiv P'$ is derived through the fact that $\equiv$ is symmetric, then $P' \equiv P$ and the result follows by induction.

(STRUC TRANS) If $P \equiv P'$ is derived through the fact that $\equiv$ is transitive, then there exists a process $Q \in \mathsf{PMA}$ such that $P \equiv Q$ and $Q \equiv P'$. Now, supposing $P \xrightarrow{\gamma} \mu$, using the fact that $P \equiv Q$, by induction there exists $\rho \in \mathsf{Distr}(\mathsf{PMA})$ such that $Q \xrightarrow{\gamma} \rho$ and $\mu \equiv \rho$. Furthermore, since $Q \equiv P'$ by induction there exists $\nu \in \mathsf{Distr}(\mathsf{PMA})$ such that $P' \xrightarrow{\gamma} \nu$ and $\rho \equiv \nu$. The result then follows from the fact that if $\mu_1 \equiv \mu_2$ and $\mu_2 \equiv \mu_3$, then $\mu_1 \equiv \mu_3$.

(STRUC CONG) It remains to consider the cases when $P \equiv P'$ is derived through the fact that $\equiv$ is preserved by all algebraic contexts, that is the cases where:

- $P = Q \mid R$ and $P' = Q' \mid R$ for some process Q, Q' and R where $Q \equiv Q'$;
- $P = (\mathsf{new}\, n)\, Q$ and $P' = (\mathsf{new}\, n)\, Q'$ for some process Q and Q' where $Q \equiv Q'$;
- $P = n[\, Q\,]$ and $P' = n[\, Q'\,]$ for some process Q and Q' where $Q \equiv Q'$;
- $P = \mathit{fix}_{\mathsf{A}} Q$ and $P' = \mathit{fix}_{\mathsf{A}} Q'$ for some identifier A and processes Q and Q' such that $Q \equiv Q'$;
- $P = M.\sum_{i \in I} p_i.P_i$ and $P' = M.\sum_{i \in I} p_i.P'_i$ for some sequences of processes $\langle P_i \rangle_{i \in I}$ and $\langle P'_i \rangle_{i \in I}$ where $P_i \equiv P'_i$ for all $i \in I$.

In each case the proof follows by induction, the derivation rule used in the transition $P \xrightarrow{\gamma} \mu$ and employing Lemma 4.9. For example, in the case when $P = n[\, Q\,]$ and $\gamma = \tau$, through (τ AMB) we have $Q \xrightarrow{\tau} \rho$ where for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if } T = n[\, T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

Since $Q \equiv Q'$, by induction $Q' \xrightarrow{\tau} \rho'$ where $\rho \equiv \rho'$ and applying (τ AMB) we have $P' = n[\, Q'\,] \xrightarrow{\tau} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \rho'(T') & \text{if } T = n[\, T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

Now, since $\rho \equiv \rho'$, using Lemma 4.9 it follows that $\mu \equiv \mu'$ as required. □

Lemma 4.11. *Let $P \in \mathsf{PMA}$. If $P \xrightarrow{M} \mu$ for some capability M, then:*

$$P \equiv (\mathsf{new}\, \tilde{k})\ \big(M.{\textstyle\sum_{i \in I}} p_i.Q_i \mid Q\big) \quad \textit{and} \quad \mu \equiv \mu'$$

for some processes $M.\sum_{i \in I} p_i.Q_i$ and Q, sequence of names $\tilde{k}$ and distribution μ' such that $\mathsf{name}(M) \notin \{\tilde{k}\}$ *and for any $T \in \mathsf{PMA}$:*

$$\mu'(T) = \begin{cases} \sum\limits_{i \in I \wedge Q_i = T'} p_i & \textit{if}\, T = (\mathsf{new}\, \tilde{k})\, (T' \mid Q) \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. The proof is by induction on derivation of $P \xrightarrow{M} \mu$. Below we only consider the case when $M = \mathsf{in}\, n$ for some $n \in \mathsf{N}$ as the cases when $M = \mathsf{out}\, n$ and $M = \mathsf{open}\, n$ for some $n \in \mathsf{N}$ follow similarly.

(act pfx) In this case P is of the form $\mathsf{in}\, n.\sum_{i \in I} p_i.P_i$ and $\mu(T) = \sum_{i \in I \wedge P_i = T} p_i$. Now from (struc par zero) and Definition 4.5 if follows that $T \equiv (\mathsf{new}\, \langle\rangle)\, (T \mid \mathbf{0})$ for all $T \in \mathsf{PMA}$, and hence

$$P \equiv (\mathsf{new}\, \langle\rangle)\ \big(\mathsf{in}\, n.{\textstyle\sum_{i \in I}} p_i.Q_i \mid \mathbf{0}\big)$$

and $\mu \equiv \mu'$, where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \sum\limits_{i \in I \wedge Q_i = T'} p_i & \text{if}\ T = (\mathsf{new}\, \langle\rangle)\, (T' \mid \mathbf{0}) \\ 0 & \text{otherwise} \end{cases}$$

as required.

(act par$_1$) In this case P is of the form $Q \mid R$, $Q \xrightarrow{\mathsf{in}\, n} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if}\ T = T' \mid R \\ 0 & \text{otherwise} \end{cases}$$

Now, by induction we have

$$Q \equiv (\mathsf{new}\, \tilde{k})\ \big(\mathsf{in}\, n.{\textstyle\sum_{i \in I}} p_i.P_i \mid Q'\big) \ \text{ and } \ \rho \equiv \rho'$$

for some processes $\mathsf{in}\, n.\sum_{i \in I} p_i.P_i$ and Q', sequence of names $\tilde{k}$ and distribution ρ' such that $\mathsf{name}(\mathsf{in}\, n) \notin \{\tilde{k}\}$ and for any $T \in \mathsf{PMA}$:

$$\rho'(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if}\ T = (\mathsf{new}\, \tilde{k})\, (T' \mid Q') \\ 0 & \text{otherwise.} \end{cases}$$

Now since $\equiv$ is a congruence:

$$\begin{aligned} P = Q \mid R \ &\equiv\ (\mathsf{new}\, \tilde{k})\ \big(\mathsf{in}\, n.{\textstyle\sum_{i \in I}} p_i.P_i \mid Q'\big) \mid R \\ &\equiv\ (\mathsf{new}\, \tilde{k})\ \big(\big(\mathsf{in}\, n.{\textstyle\sum_{i \in I}} p_i.P_i \mid Q'\big) \mid R\big) \quad \text{by (STRUC RES PAR) and Remark 4.8} \\ &\equiv\ (\mathsf{new}\, \tilde{k})\ \big(\mathsf{in}\, n.{\textstyle\sum_{i \in I}} p_i.P_i \mid (Q' \mid R)\big) \quad \text{by (STRUC PAR ASSOC).} \end{aligned}$$

It therefore remains to show that $\mu \equiv \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\,(T' \mid (Q' \mid R)) \\ 0 & \text{otherwise.} \end{cases}$$

Now, for any equivalence class $[T]_\equiv \subseteq \mathsf{PMA}$ of $\equiv$, by construction of μ:

$$\begin{aligned}
&\mu([T]_\equiv) \; = \; \rho\{T' \mid T' \in \mathsf{PMA} \wedge T' \mid R \equiv T\} \\
&= \rho'\{T' \mid T' \in \mathsf{PMA} \wedge T' \mid R \equiv T\} && \text{by Lemma 4.9 and since } \rho \equiv \rho' \\
&= \textstyle\sum\{\!|p_i \mid i \in I \wedge (\mathsf{new}\,\tilde{k})\,(P_i \mid Q') \mid R \equiv T|\!\} && \text{by definition of } \rho' \\
&= \textstyle\sum\{\!|p_i \mid i \in I \wedge (\mathsf{new}\,\tilde{k})\,((P_i \mid Q') \mid R) \equiv T|\!\} \\
& && \text{by (STRUC RES PAR) and Remark 4.8} \\
&= \textstyle\sum\{\!|p_i \mid i \in I \wedge (\mathsf{new}\,\tilde{k})\,(P_i \mid (Q' \mid R)) \equiv T|\!\} && \text{by (STRUC PAR ASSOC)} \\
&= \mu'([T]_\equiv) && \text{by definition of } \mu'.
\end{aligned}$$

Now since the equivalence class $[T]_\equiv$ was arbitrary, by definition $\mu \equiv \mu'$ as required.

(ACT PAR$_2$) This case is symmetric to (ACT PAR$_1$).

(ACT RES) In this case P is of the form $(\mathsf{new}\,k)\,Q$, $k \neq \mathsf{name}(\mathsf{in}\,n)$, $Q \xrightarrow{\mathsf{in}\,n} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if } T = (\mathsf{new}\,k)\,T' \\ 0 & \text{otherwise.} \end{cases}$$

Now, by induction we have

$$Q \equiv (\mathsf{new}\,\tilde{k})\,\left(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid R\right) \quad \text{and} \quad \rho \equiv \rho'$$

for some processes $\mathsf{in}\,n.\sum_{i\in I} p_i.P_i$ and R, sequence of names $\tilde{k}$ and distribution ρ' such that $\mathsf{name}(\mathsf{in}\,n) \notin \{\tilde{k}\}$ and for any $T \in \mathsf{PMA}$:

$$\rho'(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\,(T' \mid R) \\ 0 & \text{otherwise.} \end{cases}$$

Next, since $\equiv$ is a congruence:

$$\begin{aligned}
P = (\mathsf{new}\,k)\,Q \equiv &\; (\mathsf{new}\,k\tilde{k})\,\left(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.P_i\right) \mid R \\
\equiv &\; (\mathsf{new}\,k\tilde{k})\,\left(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid R\right)
\end{aligned}$$

by (STRUC RES PAR) and Lemma 4.8. Furthermore, since $k \neq \mathsf{name}(\mathsf{in}\,n)$ it follows that $\mathsf{name}(\mathsf{in}\,n) \notin \{k\tilde{k}\}$. Hence, it remains to show that $\mu \equiv \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,k\tilde{k})\,(T' \mid Q') \\ 0 & \text{otherwise.} \end{cases}$$

Therefore consider equivalence class $[T]_\equiv \subseteq \mathsf{PMA}$ of $\equiv$. By construction of μ:

$$\begin{aligned}
&\mu([T]_\equiv) = \rho\{T' \mid T' \in \mathsf{PMA} \wedge (\mathsf{new}\, k)\, T' \equiv T\} \\
&= \rho'\{T' \mid T' \in \mathsf{PMA} \wedge (\mathsf{new}\, k)\, T' \equiv T\} && \text{by Lemma 4.9 and since } \rho \equiv \rho' \\
&= \textstyle\sum\{\!|p_i \mid i \in I \wedge (\mathsf{new}\, k)\, (\mathsf{new}\, \tilde{k})\, (P_i \mid Q') \equiv T|\!\} && \text{by definition of } \rho' \\
&= \mu'([T]_\equiv) && \text{by definition of } \mu'
\end{aligned}$$

and since the equivalence class $[T]_\equiv$ was arbitrary, by definition $\mu \equiv \mu'$ as required.

(ACT REC) In this case $P = \mathit{fix}_{\mathsf{A}} Q$ for some identifier A and process Q such that $Q\{\mathit{fix}_{\mathsf{A}} Q/\mathsf{A}\} \xrightarrow{\mathsf{in}\, n} \mu$ and the result follows by induction on Q. □

Lemma 4.12. *Let P be an ambient process. If $P \xrightarrow{(\tilde{k})M(\nu)} \mu$ for some sequence of names $\tilde{k}$, capability M and distributions ν and μ, then*

$$P \equiv (\mathsf{new}\, \tilde{l}) \left(m\Big[(\mathsf{new}\, \tilde{l}')\, (M.\textstyle\sum_{i \in I} p_i.P_i \mid Q_1) \Big] \mid Q_2 \right), \quad \nu \equiv \nu' \quad \textit{and} \quad \mu \equiv \mu'$$

for some processes $M.\sum_{i \in I} p_i.P_i$, Q_1 and Q_2, sequences of names $\tilde{l}$, $\tilde{l}'$ and $\tilde{k}'$ and distributions ν' and μ' such that

- $\mathsf{name}(M) \notin \{\tilde{l}'\} \cup \{\tilde{l}\}$;
- $\tilde{l}\restriction_N = \tilde{k}$ *and* $\tilde{l}\restriction_{\mathsf{N} \setminus N} = \tilde{k}'$ *where* $N = \mathsf{fn}\left(m\Big[(\mathsf{new}\, \tilde{l}')\, (M.\sum_{i \in I} p_i.P_i \mid Q_1) \Big] \right)$;
- *for any* $T \in \mathsf{PMA}$:

$$\begin{aligned}
\nu'(T) &= \begin{cases} \displaystyle\sum_{i \in I \wedge P_i = T'} p_i & \textit{if } T = m\Big[(\mathsf{new}\, \tilde{l}')\, (T' \mid Q_1) \Big] \\ 0 & \textit{otherwise} \end{cases} \\
\mu'(T) &= \begin{cases} 1 & \textit{if } T = (\mathsf{new}\, \tilde{k}')\, Q_2 \\ 0 & \textit{otherwise.} \end{cases}
\end{aligned}$$

Proof. The proof is by induction on the derivation of $P \xrightarrow{(\tilde{k})M(\nu)} \mu$. Below we only consider the case when $M = \mathsf{in}\, n$ for some $n \in \mathsf{N}$ as the case for $M = \mathsf{out}\, n$ follows similarly.

(ACT* ENTER) In the case P is of the form $m[Q]$, $P \xrightarrow{\langle\rangle \mathsf{enter}\, n(\nu)} \mu$, $Q \xrightarrow{\mathsf{in}\, n} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} 1 & \text{if } T = \mathbf{0} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \nu(T) = \begin{cases} \rho(T') & \text{if } T = m[T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

Now using Lemma 4.11:

$$Q \equiv (\mathsf{new}\, \tilde{l}')\, \left(\mathsf{in}\, n.\textstyle\sum_{i \in I} p_i.P_i \mid Q'\right) \quad \text{and} \quad \rho \equiv \rho'$$

for some processes $\mathsf{in}\,n.\sum_{i\in I} p_i.P_i$ and Q, sequence of names $\tilde{l}'$ and distribution ρ' such that $\mathsf{name}(\mathsf{in}\,n) \notin \{\tilde{l}'\}$ and for any $T \in \mathsf{PMA}$:

$$\rho'(T) = \begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{l}')\,(T' \mid Q') \\ 0 & \text{otherwise.} \end{cases}$$

By construction and since $\equiv$ is a congruence, using (STRUC RES PAR), (STRUC PAR ZERO) and Definition 4.5:

$$\begin{aligned} P = m[\,Q\,] &\equiv m\Big[\,(\mathsf{new}\,\tilde{l}')\ \big(\mathsf{in}\,n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q'\big)\,\Big] \\ &\equiv\ m\Big[\,(\mathsf{new}\,\tilde{l}')\ \big(\mathsf{in}\,n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q'\big)\,\Big] \mid \mathbf{0} && \text{by (STRUC PAR ZERO)} \\ &=\ (\mathsf{new}\,\langle\rangle)\ \Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q')\,\Big] \mid \mathbf{0}\Big) && \text{by Definition 4.5.} \end{aligned}$$

It therefore remains to show that $\nu \equiv \nu'$ and $\mu \equiv \mu'$ where:

$$\begin{aligned} \nu'(T) &= \begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q')\,\Big] \\ 0 & \text{otherwise} \end{cases} \\ \mu'(T) &= \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\langle\rangle)\,\mathbf{0} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

which follows using Lemma 4.9 and the facts that $\rho \equiv \rho'$ and $\mathbf{0} = (\mathsf{new}\,\langle\rangle)\,\mathbf{0}$.

(ACT* PAR$_1$) In the case P is of the form $Q \mid R$, $\mathsf{fn}(Q) \cap \{\tilde{k}\} = \emptyset$, $Q \xrightarrow{\tilde{k}\beta(\nu)} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if } T = T' \mid Q \\ 0 & \text{otherwise.} \end{cases}$$

Now by induction on Q we have

$$Q \equiv (\mathsf{new}\,\tilde{l})\ \Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q_1)\,\Big] \mid Q_2\Big), \quad \nu \equiv \nu' \quad \text{and} \quad \rho \equiv \rho'$$

for some processes $\mathsf{in}\,n.\sum_{i\in I} p_i.P_i$, Q_1 and Q_2, sequences of names $\tilde{l}$, $\tilde{l}'$ and $\tilde{k}'$ and distributions ν' and ρ' such that:

- $\mathsf{name}(\mathsf{in}\,n) \notin \{\tilde{l}'\} \cup \{\tilde{l}\}$;
- $\tilde{l}\upharpoonright_N = \tilde{k}$ and $\tilde{l}\upharpoonright_{\mathsf{N}\setminus N} = \tilde{k}'$ where $N = \mathsf{fn}\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(M.\sum_{i\in I} p_i.P_i \mid Q_1)\,\Big]\Big)$;
- for any $T \in \mathsf{PMA}$:

$$\begin{aligned} \nu'(T) &= \begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\,\Big] \\ 0 & \text{otherwise} \end{cases} \\ \rho'(T) &= \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}')\,Q_2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Now, since $\equiv$ is a congruence:

$$\begin{aligned} P = Q \mid R &\equiv (\mathsf{new}\,\tilde{l})\,\Big(m\Big[(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i\in I}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big) \mid R \\ &\equiv (\mathsf{new}\,\tilde{l})\,\Big(\Big(m\Big[(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i\in I}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big) \mid R\Big) \\ &\equiv (\mathsf{new}\,\tilde{l})\,\Big(m\Big[(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i\in I}p_i.P_i \mid Q_1)\Big] \mid (Q_2 \mid R)\Big) \end{aligned}$$

where the second step follows from ($\textsc{struc res res}$) and Remark 4.8 and the final step from ($\textsc{struc res par}$). Since $\nu \equiv \nu'$ it remains to show that $\mu \equiv \mu'$ where

$$\mu'(T) = \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}')\,(Q_2 \mid R) \\ 0 & \text{otherwise.} \end{cases}$$

Using Remark 4.8 and ($\textsc{struc res par}$) we have $(\mathsf{new}\,\tilde{k}')\,(Q_2 \mid R) \equiv (\mathsf{new}\,\tilde{k}')\,Q_2 \mid R$, and hence the result follows from the fact that $\rho \equiv \rho'$.

$(\textsc{act}^\star\ \textsc{par}_2)$ This case is symmetric to $(\textsc{act}^\star\ \textsc{par}_1)$.

$(\textsc{act}^\star\ \textsc{rest}_1)$ In this case P is of the form $(\mathsf{new}\,k)\,Q$, $\tilde{k} = k\tilde{k}'$, $k \neq \mathsf{name}(\beta)$, $k \in \mathsf{fn}(\nu)$ and $Q \xrightarrow{\tilde{k}'\beta(\nu)} \rho$. Now by induction on Q we have

$$Q \equiv (\mathsf{new}\,\tilde{l})\,\Big(m\Big[(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i\in I}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big), \quad \nu \equiv \nu' \quad \text{and} \quad \rho \equiv \rho'$$

for some processes $\mathsf{in}\,n.\sum_{i\in I}p_i.P_i$, Q_1 and Q_2, sequences of names $\tilde{l}$, $\tilde{l}'$ and $\tilde{k}'$ and distributions ν' and ρ' such that

- $\mathsf{name}(\mathsf{in}\,n) \notin \{\tilde{l}'\} \cup \{\tilde{l}\}$;
- $\tilde{l}\restriction_N = \tilde{k}$ and $\tilde{l}\restriction_{\mathsf{N}\setminus N} = \tilde{k}'$ where $N = \mathsf{fn}\,\Big(m\Big[(\mathsf{new}\,\tilde{l}')\,(M.\sum_{i\in I}p_i.P_i \mid Q_1)\Big]\Big)$;
- for any $T \in \mathsf{PMA}$:

$$\begin{aligned} \nu'(T) &= \begin{cases} \displaystyle\sum_{i\in I \wedge P_i = T'} p_i & \text{if } T = m\Big[(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\Big] \\ \qquad 0 & \text{otherwise} \end{cases} \\ \rho'(T) &= \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}')\,Q_2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Since $\equiv$ is a congruence we have:

$$P = (\mathsf{new}\,k)\,Q \equiv (\mathsf{new}\,k\tilde{l})\,\Big(m\Big[(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i\in I}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big)$$

and the result follows from the fact that $k \in N$.

$(\textsc{act}^\star\ \textsc{rest}_2)$ In the case P is of the form $(\mathsf{new}\,k)\,Q$, $k \neq \mathsf{name}(\beta)$, $k \notin \mathsf{fn}(\nu)$ and $Q \xrightarrow{\tilde{k}\beta(\nu)} \rho$ where for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if } T = (\mathsf{new}\,k)\,T' \\ \ \ 0 & \text{otherwise.} \end{cases}$$

Now, by induction on Q we have

$$Q \equiv (\mathsf{new}\,\tilde{l})\ \left(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\right), \quad \nu \equiv \nu' \quad \text{and} \quad \rho \equiv \rho'$$

for some processes $\mathsf{in}\,n.\sum_{i\in I} p_i.P_i$, Q_1 and Q_2, sequences of names $\tilde{l}$, $\tilde{l}'$ and $\tilde{k}'$ and distributions ν' and ρ' such that

- $\mathsf{name}(M) \notin \{\tilde{l}'\} \cup \{\tilde{l}\}$;
- $\tilde{l}\restriction_N = \tilde{k}$ and $\tilde{l}\restriction_{\mathsf{N}\setminus N} = \tilde{k}'$ where $N = \mathsf{fn}\left(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(M.\sum_{i\in I} p_i.P_i \mid Q_1)\Big]\right)$;
- for any $T \in \mathsf{PMA}$:

$$\nu'(T) \ = \begin{cases} \sum\limits_{i\in I\wedge P_i = T'} p_i & \text{if } T = m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\Big] \\ \quad 0 & \text{otherwise} \end{cases}$$

$$\rho'(T) \ = \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}')\,Q_2 \\ 0 & \text{otherwise.} \end{cases}$$

Now since $\equiv$ is a congruence:

$$\begin{aligned} P = (\mathsf{new}\,k)\,Q \equiv\ & (\mathsf{new}\,k)\,(\mathsf{new}\,\tilde{l})\ \left(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\right) \\ \equiv\ & (\mathsf{new}\,\tilde{l})\,(\mathsf{new}\,k)\ \left(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\right) \\ \equiv\ & (\mathsf{new}\,\tilde{l})\ \left(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid (\mathsf{new}\,k)\,Q_2\right) \end{aligned}$$

where the second step follows from (STRUC RES RES) and the final step follows from (STRUC RES PAR) and the hypothesis that $k \neq \mathsf{name}(\beta)$, $k \notin \mathsf{fn}(\nu)$ and $\nu \equiv \nu'$. From the definition of ρ' and since $k \notin N$, it is sufficient to show that $\mu \equiv \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} 1 & \text{if } T = (\mathsf{new}\,k\tilde{k}')\,Q_2 \\ 0 & \text{otherwise} \end{cases}$$

which follows from the fact that $\rho \equiv \rho'$ and the definition of μ.

(ACT REC) In this case $P = \mathit{fix}_{\mathsf{A}}Q$ for some identifier A and process Q such that $Q\{\mathit{fix}_{\mathsf{A}}Q/\mathsf{A}\} \xrightarrow{(\tilde{k})M(\nu)} \mu$ and the result follows by induction on Q. □

Lemma 4.13. *Let P be an ambient process. If $P \xrightarrow{\mathsf{open}\,n} \mu$ for some $n \in \mathsf{N}$, then*

$$P \equiv (\mathsf{new}\,\tilde{k})\,(n[\,P_1\,] \mid P_2) \quad \textit{and} \quad \mu \equiv \mu'$$

for some processes P_1 and P_2, sequence of names $\tilde{k}$ and distribution μ' such that $n \notin \{\tilde{k}\}$ and for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} 1 & \textit{if } T = (\mathsf{new}\,\tilde{k})\,(P_1 \mid P_2) \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. The proof follows by induction on derivation tree for $P \xrightarrow{\overline{\mathsf{open}\, n}} \mu$ and is similar to Lemma 4.11. □

Lemma 4.14. *Let P be an ambient process. If $P \xrightarrow{\overline{\mathsf{enter}\, n}(\nu)} \mu$ for some distributions ν and μ, then:*

$$P \equiv n[\,Q_1\,] \mid Q_2, \quad \nu \equiv \nu' \quad and \quad \mu \equiv \mu'$$

for some processes Q_1 and Q_2 and distributions ν' and μ' such that for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \nu'(T') & \textit{if}\, T = n[\,T' \mid Q_1\,] \mid Q_2 \\ 0 & \textit{otherwise} \end{cases}$$

Proof. The proof follows by induction on derivation tree for $P \xrightarrow{\overline{\mathsf{enter}\, n}(\nu)} \mu$ in a manner similar to Lemma 4.12. □

We now proceed with the proof of Theorem 4.7.

Proof of Theorem 4.7. In the first half of the proof we show that if $P \to \mu$, then $P \xrightarrow{\tau} \mu'$ for some $\mu' \in \mathsf{Distr}(\mathsf{PMA})$ such that $\mu \equiv \mu'$. The proof is by structural induction on $P \to \mu$.

(red in) In this case P is of the form $m\big[\,\mathsf{in}\, n.\sum_{i\in I} p_i.P_i \mid Q\,\big] \;\mid n[\,R\,]$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \sum\limits_{i\in I\wedge P_i = T'} p_i & \text{if}\, T = n[\,m[\,T' \mid Q\,] \mid R\,] \\ 0 & \text{otherwise} \end{cases}$$

Now using (act pfx), (act par) and (act* enter) we have

$$m\big[\,\mathsf{in}\, n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q\,\big] \xrightarrow{\langle\rangle \mathsf{enter}\, n(\nu')} \mu_1'$$

where for any $T \in \mathsf{PMA}$:

$$\nu'(T) = \begin{cases} \sum\limits_{i\in I\wedge P_i = T'} p_i & \text{if}\, T = m[\,T' \mid Q\,] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mu_1'(T) = \begin{cases} 1 & \text{if}\, T = \mathbf{0} \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, using (act* $\overline{\text{enter}}$), $n[\,R\,] \xrightarrow{\overline{\mathsf{enter}\, n}(\nu')} \mu_2'$ where

$$\mu_2'(T) = \begin{cases} \nu'(T') & \text{if}\, T = n[\,R \mid T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

Combining these transitions through (τ enter$_1$) it follows that

$$m\big[\,\mathsf{in}\, n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q\,\big] \;\mid n[\,R\,] \xrightarrow{\tau} \mu'$$

where for any $T \in \mathsf{PMA}$:

$$\begin{aligned}
\mu'(T) &= \begin{cases} \mu'_1(T_1) \cdot \mu'_2(T_2) & \text{if } T = (\mathsf{new}\,\langle\rangle)\,(T_1 \mid T_2) \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \mu'_1(T_1) \cdot \mu'_2(T_2) & \text{if } T = T_1 \mid T_2 \\ 0 & \text{otherwise} \end{cases} && \text{by Definition 4.5} \\
&= \begin{cases} 1 \cdot \mu'_2(T_2) & \text{if } T = \mathbf{0} \mid T_2 \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \mu'_1 \\
&= \begin{cases} \nu'(T) & \text{if } T = \mathbf{0} \mid n[\,R \mid T'\,] \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \mu'_2 \\
&= \begin{cases} \sum_{i \in I \wedge P_i = T'} p_i & \text{if } T = \mathbf{0} \mid n[\,R \mid m[T' \mid Q\,]\,] \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \nu'.
\end{aligned}$$

Now using (STRUC PAR ZERO), for any $T' \in \mathsf{PMA}$:

$$\begin{aligned}
n[\,m[T' \mid Q\,] \mid R\,] &\equiv \mathbf{0} \mid n[\,m[T' \mid Q\,] \mid R\,] \\
&\equiv (\mathbf{0} \mid n[\,R \mid m[T' \mid Q\,]\,]) && \text{by (STRUC PAR COM)}
\end{aligned}$$

and hence it follows that $\mu \equiv \mu'$ as required.

(RED OUT) By definition $P = n\big[\,m\big[\,\mathsf{out}\,n.\sum_{i \in I} p_i.P_i \mid Q\,\big] \mid R\,\big] \rightarrow \mu$ where for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \sum_{i \in I \wedge P_i = T'} p_i & \text{if } T = m[T' \mid Q\,] \mid n[\,R\,] \\ 0 & \text{otherwise.} \end{cases}$$

Now using (ACT PFX), followed by (ACT PAR), (ACT* EXIT) and (ACT* PAR$_2$) we have

$$m\big[\,\mathsf{out}\,n.{\textstyle\sum_{i \in I}} p_i.P_i \mid Q\,\big] \mid R \xrightarrow{\langle\rangle\mathsf{exit}\,n(\nu')} \rho'$$

where for any $T \in \mathsf{PMA}$:

$$\nu'(T) = \begin{cases} \sum_{i \in I \wedge P_i = T'} p_i & \text{if } T = m[T' \mid Q\,] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \rho'(T) = \begin{cases} 1 & \text{if } T = \mathbf{0} \mid R \\ 0 & \text{otherwise.} \end{cases}$$

Next, applying (τ EXIT), $P \xrightarrow{\tau} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\begin{aligned}
\mu'(T) &= \begin{cases} \nu'(T_1) \cdot \rho'(T_2) & \text{if } T = T_1 \mid n[T_2\,] \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \nu'(T_1) \cdot 1 & \text{if } T = T_1 \mid n[\mathbf{0} \mid R\,] \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \rho' \\
&= \begin{cases} \sum_{i \in I \wedge P_i = T'} p_i & \text{if } T = m[T' \mid Q\,] \mid n[\mathbf{0} \mid R\,] \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \nu'.
\end{aligned}$$

Now by (STRUC PAR ZERO) for any $T' \in \mathsf{PMA}$:

$$m[T' \mid Q\,] \mid n[\,R\,] \equiv m[T' \mid Q\,] \mid n[\mathbf{0} \mid R\,]$$

and hence it follows that $\mu \equiv \mu'$ as required.

(RED OPEN) By definition $P = \mathsf{open}\, n. \sum_{i \in I} p_i.P_i \mid n[\,Q\,] \;\to \mu$ where for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = T' \mid Q \\ 0 & \text{otherwise.} \end{cases}$$

Now using (ACT PFX) and (ACT $\overline{\text{OPEN}}$)

$$\mathsf{open}\, n. \sum_{i \in I} p_i.P_i \xrightarrow{\mathsf{open}\, n} \mu'_1 \quad \text{and} \quad n[\,Q\,] \xrightarrow{\overline{\mathsf{open}\, n}} \mu'_2$$

where for any $T \in \mathsf{PMA}$:

$$\mu'_1(T) = \sum_{i \in I \wedge P_i = T'} p_i \;\text{ and }\; \mu'_2(T) = \begin{cases} 1 & \text{if } T = Q \\ 0 & \text{otherwise.} \end{cases}$$

Next, applying (τ OPEN$_1$) we have $P \xrightarrow{\tau} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\begin{aligned} \mu'(T) &= \begin{cases} \mu'_1(T_1) \cdot \mu'_2(T_2) & \text{if } T = T_1 \mid T_2 \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} \mu'_1(T_1) \cdot 1 & \text{if } T = T_1 \mid Q \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \mu'_2 \\ &= \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = T' \mid Q \\ 0 & \text{otherwise} \end{cases} && \text{by definition of } \mu'_1 \\ &= \mu(T) && \text{by definition of } \mu. \end{aligned}$$

Since $\mu = \mu'$ it follows that $\mu \equiv \mu'$ as required.

(RED PAR) In this case $P = Q \mid R$, $Q \to \nu$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \nu(T') & \text{if } T = T' \mid R \\ 0 & \text{otherwise.} \end{cases}$$

By induction $Q \xrightarrow{\tau} \nu'$ and $\nu \equiv \nu'$. Next applying (τ PAR$_1$) we have $P = Q \mid R \xrightarrow{\tau} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \nu'(T') & \text{if } T = T' \mid R \\ 0 & \text{otherwise} \end{cases}$$

Applying Lemma 4.9 it follows that $\mu \equiv \mu'$ as required.

(RED RESTR) In this case $P = (\mathsf{new}\, n)\, Q$, $Q \to \nu$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \nu(T') & \text{if } T = (\mathsf{new}\, n)\, T' \\ 0 & \text{otherwise} \end{cases}$$

By induction $Q \xrightarrow{\tau} \nu'$ for some distribution ν' such that $\nu \equiv \nu'$. Next applying (τ REST) we have $P = (\mathsf{new}\, n)\, Q \xrightarrow{\tau} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \nu'(T') & \text{if } T = (\mathsf{new}\, n)\, T' \\ 0 & \text{otherwise} \end{cases}$$

Applying Lemma 4.9 it follows that $\mu \equiv \mu'$ as required.

(RED AMB) $P = n[\,Q\,]$, $Q \to \nu$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \mu'(T') & \text{if } T = n[\,T'\,] \\ 0 & \text{otherwise} \end{cases}$$

By induction $Q \xrightarrow{\tau} \nu'$ for some distribution ν' such that $\nu \equiv \nu'$. Next applying (τ AMB) we have $P = n[\,Q\,] \xrightarrow{\tau} \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \nu'(T') & \text{if } T = n[\,T'\,] \\ 0 & \text{otherwise} \end{cases}$$

Applying Lemma 4.9 it follows that $\mu \equiv \mu'$ as required.

(RED CONG) In this case $P \equiv Q$, $Q \to \rho$ and $\rho \equiv \mu$. Now, by induction there exists ρ' such that $Q \xrightarrow{\tau} \rho'$ and $\rho \equiv \rho'$. Furthermore, applying Lemma 4.10, $P \xrightarrow{\tau} \mu'$ and $\mu' \equiv \rho'$. Therefore, by the transitivity of $\equiv$, it follows that $\mu \equiv \mu'$ as required.

Since these are the only cases to consider this completes the first half of the proof.

In the second half of the proof we show the reverse direction, that is, if $P \xrightarrow{\tau} \mu$, then $P \to \mu$. The proof is by induction on the derivation of $P \xrightarrow{\tau} \mu$.

(τ ENTER$_1$) In this case $P = Q \mid R$, $Q \xrightarrow{\tilde{k}\,\mathsf{enter}\,n(\nu)} \mu_1$, $R \xrightarrow{\overline{\mathsf{enter}}\,n(\nu)} \mu_2$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \mu_1(T_1) \cdot \mu_2(T_2) & \text{if } T = (\mathsf{new}\,\tilde{k})\,(T_1 \mid T_2) \\ 0 & \text{otherwise.} \end{cases}$$

By Lemma 4.12:

$$Q \equiv (\mathsf{new}\,\tilde{l})\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big)\,,\quad \nu \equiv \nu_1' \quad\text{and}\quad \mu_1 \equiv \mu_1'$$

for some processes $\mathsf{in}\,n.\sum_{i\in I}p_i.P_i$, Q_1 and Q_2, sequences of names $\tilde{l}$, $\tilde{l}'$ and $\tilde{k}'$ and distributions ν_1' and μ_1' such that:

- $\mathsf{name}(\mathsf{in}\,n) \notin \{\tilde{l}'\} \cup \{\tilde{l}\}$;
- $\tilde{l}\restriction_N = \tilde{k}$ and $\tilde{l}\restriction_{\mathsf{N}\setminus N} = \tilde{k}'$ where $N = \mathsf{fn}(\mathsf{in}\,n.\,\sum_{i\in I} p_i.P_i \mid Q_1)$;
- for any $T \in \mathsf{PMA}$:

$$\begin{aligned} \nu_1'(T) &= \begin{cases} \displaystyle\sum_{i\in I \wedge P_i = T'} p_i & \text{if } T = m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\Big] \\ 0 & \text{otherwise} \end{cases} \\ \mu_1'(T) &= \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}')\,Q_2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Furthermore, using Lemma 4.14 we have

$$R \equiv n[\,R_1\,] \mid R_2, \;\; \nu \equiv \nu_2' \;\text{ and }\; \mu_2 \equiv \mu_2'$$

for some processes R_1 and R_2 and distributions ν_2' and μ_2 such that for any $T \in \mathsf{PMA}$:

$$\mu_2'(T) = \begin{cases} \nu_2'(T') & \text{if } T = n[\,T' \mid R_1\,] \mid R_2 \\ 0 & \text{otherwise.} \end{cases}$$

Using Lemma 4.9 and the fact that $\mu_1 \equiv \mu_1'$, it follows from the definition of μ that $\mu \equiv \rho_1$ where for any $T \in \mathsf{PMA}$:

$$\rho_1(T) = \begin{cases} \mu_2(T_2) & \text{if } T = (\mathsf{new}\,\tilde{k})\,((\mathsf{new}\,\tilde{k}')\,Q_2 \mid T_2) \\ 0 & \text{otherwise.} \end{cases}$$

Combining Lemma 4.9 with the fact that $\mu_2' \equiv \mu_2$, we have $\rho_1 \equiv \rho_2$ where for any $T \in \mathsf{PMA}$:

$$\rho_2(T) = \begin{cases} \nu_2'(T_2) & \text{if } T = (\mathsf{new}\,\tilde{k})\,((\mathsf{new}\,\tilde{k}')\,Q_2 \mid (n[\,T' \mid R_1\,] \mid R_2)) \\ 0 & \text{otherwise.} \end{cases}$$

Next, since $\nu_2' \equiv \nu$ it follows that $\rho_2 \equiv \rho_3$ where for any $T \in \mathsf{PMA}$:

$$\rho_3(T) = \begin{cases} \nu(T_2) & \text{if } T = (\mathsf{new}\,\tilde{k})\,((\mathsf{new}\,\tilde{k}')\,Q_2 \mid (n[\,T' \mid R_1\,] \mid R_2)) \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, since $\nu \equiv \nu_1'$, $\rho_3 \equiv \rho_4$ where, for any $T \in \mathsf{PMA}$, $\rho_4(T)$ equals

$$\begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\,\Big((\mathsf{new}\,\tilde{k}')\,Q_2 | \Big(n\Big[\,m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T'|Q_1)\,\Big]\,|R_1\,\Big]\,|R_2\Big)\Big) \\ 0 & \text{otherwise.} \end{cases}$$

By (STRUC PAR COM) for any $T' \in \mathsf{PMA}$:

$$\begin{aligned} &(\mathsf{new}\,\tilde{k})\,\Big((\mathsf{new}\,\tilde{k}')\,Q_2 \mid \Big(n\Big[\,m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\,\Big] \mid R_1\,\Big] \mid R_2\Big)\Big) \\ &\quad \equiv\ (\mathsf{new}\,\tilde{k})\,\Big(n\Big[\,m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\,\Big] \mid R_1)\,\Big] \mid ((\mathsf{new}\,\tilde{k}')\,Q_2 \mid R_2)\Big) \end{aligned}$$

and hence $\rho_4 \equiv \rho_5$ where, for any $T \in \mathsf{PMA}$, $\rho_5(T)$ equals

$$\begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\,\Big(n\Big[\,m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T'|Q_1)\,\Big]\,|R_1\,\Big]\,|((\mathsf{new}\,\tilde{k}')\,Q_2|R_2)\Big) \\ 0 & \text{otherwise.} \end{cases}$$

Finally in this derivation, through the transitivity of $\equiv$, we have $\mu \equiv \rho_5$.

On the other hand, since $\equiv$ is a congruence:

$$\begin{aligned} P = Q \mid R\ &\equiv\ (\mathsf{new}\,\tilde{l})\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid Q_1)\,\Big] \mid Q_2\Big) \mid (n[\,R_1\,] \mid R_2) \\ &\equiv\ (\mathsf{new}\,\tilde{k})\,(\mathsf{new}\,\tilde{k}')\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid Q_1)\,\Big] \mid Q_2\Big) \mid (n[\,R_1\,] \mid R_2) \\ &\equiv\ (\mathsf{new}\,\tilde{k})\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid Q_1)\,\Big] \mid (\mathsf{new}\,\tilde{k}')\,Q_2\Big) \mid (n[\,R_1\,] \mid R_2) \\ &\equiv\ (\mathsf{new}\,\tilde{k})\,\Big(m\Big[\,(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.(\mathsf{new}\,\tilde{l}')\,(P_i \mid Q_1))\,\Big] \mid (\mathsf{new}\,\tilde{k}')\,Q_2\Big) \mid (n[\,R_1\,] \mid R_2) \\ &\equiv\ (\mathsf{new}\,\tilde{k})\,\Big(\Big(m\Big[\,(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.(\mathsf{new}\,\tilde{l}')\,(P_i \mid Q_1))\,\Big] \mid (\mathsf{new}\,\tilde{k}')\,Q_2\Big) \mid (n[\,R_1\,] \mid R_2)\Big) \\ &\equiv\ (\mathsf{new}\,\tilde{k})\,\Big(\Big(m\Big[\,(\mathsf{in}\,n.\textstyle\sum_{i \in I} p_i.(\mathsf{new}\,\tilde{l}')\,(P_i \mid Q_1))\,\Big] \mid n[\,R_1\,]\Big) \mid ((\mathsf{new}\,\tilde{k}')\,Q_2 \mid R_2)\Big) \end{aligned}$$

where the second step follows from (STRUC RES RES) and the definition of $\tilde{l}$, the third, fourth and fifth from (STRUC RES PAR) and the facts that $\tilde{l} \restriction_{\mathsf{N} \setminus N} = \tilde{k}'$, $\mathsf{name}(\mathsf{in}\,n) \notin \tilde{l}'$ and Remark 4.8, while the final step from applying (STRUC PAR COM) and (STRUC PAR ASSOC). Applying the structural congruence rules, (RED IN), (RED RESTR), (RED PAR) and (RED CONG), it follows that $P \rightarrow \mu'$ where, for any $T \in \mathsf{PMA}$, $\mu'(T)$ equals

$$\begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\,(n\Big[\,m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T'|Q_1)\Big]\,|R_1\Big]\,|((\mathsf{new}\,\tilde{k}')\,Q_2|R_2)) \\ 0 & \text{otherwise} \end{cases}$$

Now since $\mu' = \rho_5$ it follows that $\mu \equiv \mu'$, and hence using (RED CONG) we have $P \rightarrow \mu$ as required.

(τ ENTER$_2$) This case is symmetric to (τ ENTER$_1$).

(τ EXIT) In this case P is of the form $n[\,Q\,]$, $Q \xrightarrow{\tilde{k}\mathsf{exit}\,n(\nu)} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \nu(T_1)\cdot\rho(T_2) & \text{if } T = (\mathsf{new}\,\tilde{k})\,(T_1 \mid n[\,T_2\,]) \\ 0 & \text{otherwise.} \end{cases}$$

By Lemma 4.12:

$$Q \equiv (\mathsf{new}\,\tilde{l})\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{out}\,n.{\textstyle\sum_{i\in I}} p_i.P_i \mid Q_1)\Big] \mid Q_2\Big), \quad \nu \equiv \nu' \quad \text{and} \quad \rho \equiv \rho'$$

for some processes $\mathsf{out}\,n.\sum_{i\in I} p_i.P_i$, Q_1 and Q_2, sequences of names $\tilde{l}$, $\tilde{l}'$ and $\tilde{k}'$ and distributions ν' and ρ' such that

- $\mathsf{name}(\mathsf{out}\,n) \not\in \{\tilde{l}'\} \cup \{\tilde{l}\}$;
- $\tilde{l}\restriction_N = \tilde{k}$ and $\tilde{l}\restriction_{\mathsf{N}\setminus N} = \tilde{k}'$ where $N = \mathsf{fn}(\mathsf{out}\,n.\sum_{i\in I} p_i.P_i \mid Q_1)$;
- for any $T \in \mathsf{PMA}$:

$$\begin{aligned} \nu'(T) \;&= \begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\Big] \\ 0 & \text{otherwise} \end{cases} \\ \rho'(T) \;&= \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}')\,Q_2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

By definition of μ, Lemma 4.9 and the fact that $\rho \equiv \rho'$ it follows that $\mu \equiv \mu_1$ where for any $T \in \mathsf{PMA}$:

$$\mu_1(T) = \begin{cases} \nu(T_1) & \text{if } T = (\mathsf{new}\,\tilde{k})\,\Big(T_1 \mid n\Big[\,(\mathsf{new}\,\tilde{k}')\,Q_2\Big]\Big) \\ 0 & \text{otherwise.} \end{cases}$$

Since $\nu \equiv \nu'$ it follows that $\mu_1 \equiv \mu_2$ where, for any $T \in \mathsf{PMA}$, $\mu_2(T)$ equals

$$\begin{cases} \sum\limits_{i\in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\,\Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\Big] \mid n\Big[\,(\mathsf{new}\,\tilde{k}')\,Q_2\Big]\Big) \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, since $\equiv$ is a congruence:

$$\begin{aligned}
P = n[\,Q\,] \equiv{}& n\Big[\,(\mathsf{new}\,\tilde{l})\ \Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{out}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big)\Big]\\
\equiv{}& n\Big[\,(\mathsf{new}\,\tilde{k}\tilde{k}')\ \Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{out}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big)\Big]\\
\equiv{}& (\mathsf{new}\,\tilde{k})\,n\Big[\,(\mathsf{new}\,\tilde{k}')\ \Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{out}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid Q_2\Big)\Big]\\
\equiv{}& (\mathsf{new}\,\tilde{k})\,n\Big[\,m\Big[\,(\mathsf{new}\,\tilde{l}')\,(\mathsf{out}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q_1)\Big] \mid (\mathsf{new}\,\tilde{k}')\,Q_2\Big]\\
\equiv{}& (\mathsf{new}\,\tilde{k})\,n\Big[\,m\Big[\,(\mathsf{out}\,n.{\textstyle\sum_{i\in I}}p_i.(\mathsf{new}\,\tilde{l}')\,(P_i \mid Q_1))\Big] \mid (\mathsf{new}\,\tilde{k}')\,Q_2\Big]
\end{aligned}$$

where the second step follows from (STRUC RES RES) and the definition of $\tilde{l}$, the remaining steps follow from Remark 4.8, (STRUC RES PAR) and the following facts:

- $\mathsf{name}(\mathsf{out}\,n) \notin \tilde{l}$;
- $\tilde{l}\restriction_{\mathsf{N}} = \tilde{k}$;
- $\tilde{l}\restriction_{\mathsf{N}\setminus N} = \tilde{k}'$;
- $\mathsf{name}(\mathsf{out}\,n) \notin \tilde{l}'$.

Now, applying the structural congruence rules (RED OUT), (RED RESTR), (RED PAR) and (RED CONG), $n[\,Q\,] \to \mu'$ where, for any $T \in \mathsf{PMA}$, $\mu'(T)$ equals

$$\begin{cases} \sum\limits_{i\in I\wedge P_i=T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k})\ \Big(m\Big[\,(\mathsf{new}\,\tilde{l}')\,(T' \mid Q_1)\Big] \mid n\Big[\,((\mathsf{new}\,\tilde{k}')\,Q_2)\Big]\Big)\\ 0 & \text{otherwise.}\end{cases}$$

Now since $\mu \equiv \mu_2$ it follows that $\mu \equiv \mu'$ and therefore applying (RED CONG) we have $P \to \mu$ as required.

(τ OPEN$_1$) In this case P is of the form $Q \mid R$, $Q \xrightarrow{\mathsf{open}\,n} \mu_1$, $R \xrightarrow{\overline{\mathsf{open}\,n}} \mu_2$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \mu_1(T_1)\cdot\mu_2(T_2) & \text{if } T = T_1 \mid T_2\\ 0 & \text{otherwise.}\end{cases}$$

Now by Lemma 4.11:

$$Q \equiv (\mathsf{new}\,\tilde{k}_1)\ \big(\mathsf{open}\,n.{\textstyle\sum_{i\in I}}p_i.P_i \mid Q'\big) \quad\text{and}\quad \mu_1 \equiv \mu'_1$$

for some processes $\mathsf{open}\,n.\sum_{i\in I} p_i.P_i$ and Q', sequence of names $\tilde{k}_1$ and distribution μ'_1 such that $\mathsf{name}(\mathsf{open}\,n) \notin \{\tilde{k}\}$ for any $T \in \mathsf{PMA}$:

$$\mu'_1(T) = \begin{cases} \sum\limits_{i\in I\wedge P_i=T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k}_1)\,(T' \mid Q)\\ 0 & \text{otherwise.}\end{cases}$$

Furthermore, from Lemma 4.13:

$$R \equiv (\mathsf{new}\,\tilde{k}_2)\,(n[\,R_1\,] \mid R_2) \quad\text{and}\quad \mu_2 \equiv \mu'_2$$

for some processes R_1 and R_2, sequence of names $\tilde{k}_2$ and distribution μ' such that for any $T \in \mathsf{PMA}$:

$$\mu'_2(T) = \begin{cases} 1 & \text{if } T = (\mathsf{new}\,\tilde{k}_2)\,(R_1 \mid R_2) \\ 0 & \text{otherwise.} \end{cases}$$

Now, since $\mu_2 \equiv \mu'_2$, it follows that $\mu \equiv \rho_1$ where for any $T \in \mathsf{PMA}$:

$$\rho_1(T) = \begin{cases} \mu_1(T_1) & \text{if } T = T_1 \mid (\mathsf{new}\,\tilde{k}_2)\,(R_1 \mid R_2) \\ 0 & \text{otherwise.} \end{cases}$$

Next, using Lemma 4.9 together with the fact that $\nu \equiv \nu'$, $\rho_1 \equiv \rho_2$ where for any $T \in \mathsf{PMA}$:

$$\rho_2(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k}_1)\,(T' \mid Q') \mid (\mathsf{new}\,\tilde{k}_2)\,(R_1 \mid R_2) \\ 0 & \text{otherwise.} \end{cases}$$

From Remark 4.8 and (struc res par) it follows that for any $T \in \mathsf{PMA}$:

$$\begin{aligned} (\mathsf{new}\,\tilde{k}_1)\,(T' \mid Q') \mid (\mathsf{new}\,\tilde{k}_2)\,(R_1 \mid R_2) &\equiv (\mathsf{new}\,\tilde{k}_1\tilde{k}_2)\,((T' \mid Q') \mid (R_1 \mid R_2)) \\ &\equiv (\mathsf{new}\,\tilde{k}_1\tilde{k}_2)\,((T' \mid R_1) \mid (Q' \mid R_2)) \end{aligned}$$

by (struc par comm) and (struc par assoc). This fact together with Lemma 4.9 implies that $\rho_2 \equiv \rho_3$ where for any $T \in \mathsf{PMA}$:

$$\rho_3(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k}_1\tilde{k}_2)\,(T' \mid R_1) \mid (Q' \mid R_2) \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, by the transitivity of $\equiv$, $\mu \equiv \rho_3$.

On the other hand since $\equiv$ is a congruence:

$$\begin{aligned} P = Q \mid R \equiv\ & (\mathsf{new}\,\tilde{k}_1)\,\big(\mathsf{open}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid Q'\big) \mid (\mathsf{new}\,\tilde{k}_2)\,(n[\,R_1\,] \mid R_2) \\ \equiv\ & (\mathsf{new}\,\tilde{k}_2\tilde{k}_1)\,\big(\mathsf{open}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid Q'\big) \mid (n[\,R_1\,] \mid R_2) \\ \equiv\ & (\mathsf{new}\,\tilde{k}_2\tilde{k}_1)\,\big(\mathsf{open}\,n.\textstyle\sum_{i \in I} p_i.P_i \mid n[\,R_1\,]\big) \mid (Q' \mid R_2) \end{aligned}$$

where the first step follows by (struc res res) and Remark 4.8 and the second by (struc par comm) and (struc par assoc). Now, applying the structural congruence rules, (red open), (red restr) and (red cong), $Q \mid R \to \mu'$ where for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \sum\limits_{i \in I \wedge P_i = T'} p_i & \text{if } T = (\mathsf{new}\,\tilde{k}_1\tilde{k}_2)\,(T' \mid R_1) \mid (Q' \mid R_2) \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, since $\mu \equiv \rho'_3$, it follows that $\mu \equiv \mu'$, and therefore applying (red cong) we have $P \to \mu$ as required.

(τ open$_2$) This case is symmetric to (τ open$_1$).

(τ PAR$_1$) In this case P is of the form $Q \mid R$, $Q \xrightarrow{\tau} \rho'$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \mu'(T') & \text{if } T = T' \mid R \\ 0 & \text{otherwise} \end{cases}$$

By induction $Q \rightarrow \rho'$ for some distribution ρ' such that $\rho \equiv \rho'$. Applying (RED PAR) we have $P = Q \mid R \rightarrow \mu'$ and for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \rho'(T') & \text{if } T = T' \mid Q \\ 0 & \text{otherwise} \end{cases}$$

Using Lemma 4.9 it follows that $\mu \equiv \mu'$, and hence, by (RED CONG), $P \rightarrow \mu$ as required.

(τ PAR$_2$) This case is symmetric to (τ PAR$_1$).

(τ REST) In this case P is of the form $(\mathsf{new}\, k)\, Q$, $Q \xrightarrow{\tau} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if } T = (\mathsf{new}\, k)\, T' \\ 0 & \text{otherwise.} \end{cases}$$

By induction $Q \rightarrow \rho'$ for some distribution ρ' such that $\rho \equiv \rho'$. Applying (RED REST) we have $P = (\mathsf{new}\, n)\, Q \rightarrow \mu'$ and for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \rho'(T') & \text{if } T = (\mathsf{new}\, k)\, T' \\ 0 & \text{otherwise} \end{cases}$$

Using Lemma 4.9 it follows that $\mu \equiv \mu'$, and hence by (RED CONG) we have $P \rightarrow \mu$ as required.

(τ AMB) In this case P is of the form $n[\,Q\,]$, $Q \xrightarrow{\tau} \rho$ and for any $T \in \mathsf{PMA}$:

$$\mu(T) = \begin{cases} \rho(T') & \text{if } T = n[\,T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

By induction $Q \rightarrow \rho'$ for some distribution ρ' such that $\rho \equiv \rho'$. Applying (RED AMB) we have $P = n[\,Q\,] \rightarrow \mu'$ and for any $T \in \mathsf{PMA}$:

$$\mu'(T) = \begin{cases} \rho'(T') & \text{if } T = n[\,T'\,] \\ 0 & \text{otherwise.} \end{cases}$$

Using Lemma 4.9 it follows that $\mu \equiv \mu'$, and hence by (RED CONG) we have $P \rightarrow \mu$ as required.

(τ REC) In this case $P = \mathit{fix}_{\mathsf{A}} Q$ for some identifier $\mathsf{A} \in \mathsf{Id}$ and process $Q \in \mathsf{PMA}$ such that $Q\{\mathit{fix}_{\mathsf{A}} Q/\mathsf{A}\} \xrightarrow{\tau} \mu$ and the result follows by induction on Q and (STRUC REC).

This completes the proof of Theorem 4.7. □

4.5. Bisimulation for Probabilistic Mobile Ambients

To complete the semantics for *PMA* we define here probabilistic bisimulation for Probabilistic Mobile Ambients following the definition of barbed bisimulation [36]. Similarly to the case for *MA* (see Definition 3.7), the basic idea is that two processes are bisimilar if they exhibit the same observable behaviour and if they reduce in the same way in any context. The definitions of the observational predicate $\downarrow$ (exhibits a barb) and a process contexts $\mathcal{C}$ are as for Mobile Ambients (see Definition 3.4 and Definition 3.6 respectively).

Definition 4.15. Barbed probabilistic bisimulation *is the largest symmetric relation* $\simeq_p \subseteq \mathsf{PMA} \times \mathsf{PMA}$ *such that* $P \simeq_p Q$ *implies:*

- *for each* $n \in \mathsf{N}$*, if* $P \downarrow n$*, then* $Q \downarrow n$*;*
- *for any context* $\mathcal{C}$*, if* $\mathcal{C}(P) \xrightarrow{\tau} \mu$*, then* $\mathcal{C}(Q) \xrightarrow{\tau} \nu$ *for some distribution* ν *such that* $\mu([R]) = \nu([R])$ *for all* $[R] \in \mathsf{PMA}_{/\simeq_p}$*.*

Below are some examples of *PMA* processes that are barbed probabilistic bisimilar.

$$q\big[\,\mathsf{in}\,n.\left(\tfrac{1}{3}.\mathsf{open}\,r.\mathbf{0} + \tfrac{2}{3}.\mathsf{in}\,m.(\mathsf{new}\,k)\,\mathbf{0}\right)\big] \\ \simeq_p \;\; q\big[\,\mathsf{in}\,n.\left(\tfrac{1}{6}.\mathsf{open}\,r.\mathbf{0} + \tfrac{1}{6}.\mathsf{open}\,r.\mathbf{0} + \tfrac{2}{3}.\mathsf{in}\,m.\mathbf{0}\right)\big]$$

$$(\mathsf{new}\,q)\,(q\big[\,\mathsf{in}\,n.\left(\tfrac{1}{3}.\mathsf{open}\,r.\mathbf{0} + \tfrac{2}{3}.\mathsf{open}\,r.\mathbf{0}\right)\big] \\ \simeq_p \;\; (\mathsf{new}\,s)\,(\mathsf{new}\,t)\;\left(s\big[\,\mathsf{in}\,n.\left(\tfrac{1}{6}.\mathsf{open}\,r.\mathbf{0} + \tfrac{5}{6}.\mathsf{open}\,r.\mathsf{out}\,t.\mathbf{0}\right)\big]\right)$$

Moreover, structural congruence is included in barbed probabilistic bisimilarity.

Proposition 4.16. *Let* P *and* Q *be PMA processes. If* $P \equiv Q$*, then* $P \simeq_p Q$*.*

PROOF. The proof follows by induction on $\equiv$. □

As in the case of Mobile Ambients, where there exist processes that are barbed bisimilar but not structurally congruent, the reverse of Proposition 4.16 is not true. For example, we can encode the standard example of *MA* processes that are bisimilar but not structurally congruent as the following *PMA* processes:

$$\mathsf{in}\,n.1.(\mathsf{in}\,n.1.\mathbf{0}) \;\text{ and }\; (\mathsf{in}\,n.1.\mathbf{0}) \mid (\mathsf{in}\,n.1.\mathbf{0})$$

which are barbed probabilistic bisimilar but not structurally congruent.

4.6. Encoding of probabilistic asynchronous CCS into PMA

Asynchronous *CCS* can be faithfully encoded into *MA* [49]. In this section we will show that *probabilistic* asynchronous *CCS* can be faithfully encoded into *PMA*. Probabilistic asynchronous *CCS* is a variant of the probabilistic extension of *CCS* as defined in [11]. The difference is that, in the asynchronous setting [50, 51] the output operator has no continuation and there is no nondeterministic choice operator. For simplicity we call this calculus *PCCS*. We will show that, as in the non-probabilistic setting [49], the encoding to *PMA* is quite natural since it is homomorphic for parallel composition, sum, restriction and recursion.

We assume an enumerable set of input actions Act and corresponding set of output actions $\overline{\mathsf{Act}} = \{\overline{a} \mid a \in \mathsf{Act}\}$.

$$\begin{array}{rl}
(\textsc{input}) & a.\sum_{i\in I} p_i.P_i \xrightarrow{a} [\![\sum_{i\in I} p_i.P_i]\!] \\
(\textsc{output}) & \overline{a} \xrightarrow{\overline{a}} \eta_{\mathbf{0}} \\
(\textsc{par}_1) & P \mid Q \xrightarrow{\alpha} \mu \mid Q \ \text{ if } \ P \xrightarrow{\alpha} \mu \\
(\textsc{par}_2) & P \mid Q \xrightarrow{\alpha} P \mid \mu \ \text{ if } \ Q \xrightarrow{\alpha} \mu \\
(\textsc{comm}_1) & P \mid Q \xrightarrow{\tau} \mu_1 \mid \mu_2 \ \text{ if } \ P \xrightarrow{a} \mu_1 \ \text{ and } \ Q \xrightarrow{\overline{a}} \mu_2 \\
(\textsc{comm}_2) & P \mid Q \xrightarrow{\tau} \mu_1 \mid \mu_2 \ \text{ if } \ P \xrightarrow{\overline{a}} \mu_1 \ \text{ and } \ Q \xrightarrow{a} \mu_2 \\
(\textsc{rest}) & (\mathsf{new}\, b)\, P \xrightarrow{\alpha} (\mathsf{new}\, b)\, \mu \ \text{ if } \ \alpha \neq b, \ \alpha \neq \overline{b} \ \text{ and } \ P \xrightarrow{\alpha} \mu \\
(\textsc{rec}) & \mathit{fix}_{\mathsf{A}} P \xrightarrow{\tau} \mu \ \text{ if } \ P\{\mathit{fix}_{\mathsf{A}} P/\mathsf{A}\} \xrightarrow{\tau} \mu
\end{array}$$

Figure 5: Labelled Transition Semantics for *PCCS*

Definition 4.17. *The set PCCS of processes* PCCS *is given by the syntax:*

$$P, Q ::= \mathbf{0} \ \big| \ P \mid Q \ \big| \ (\mathsf{new}\, a)\, P \ \big| \ \mathsf{A} \ \big| \ \mathit{fix}_{\mathsf{A}} P \ \big| \ a.\textstyle\sum_{i\in I} p_i.P_i \ \big| \ \overline{a}$$

where $a \in \mathsf{Act}$, $\overline{a} \in \overline{\mathsf{Act}}$ *and* $\sum_{i\in I} p_i$ *is a summation over a countable index set* I *such that* $p_i \in (0,1]$ *for all* $i \in I$ *and* $\sum_{i\in I} p_i = 1$.

Definition 4.18. *Let* $\mathcal{L}ab = \mathsf{Act} \cup \overline{\mathsf{Act}} \cup \{\tau\}$ *be the set of labels. The* labelled transition system semantics *for PCCS is the probabilistic automaton* $(\mathsf{PCCS}, \mathcal{L}ab, \rightarrow)$ *where the probabilistic transition relation* $\rightarrow \subseteq \mathsf{PCCS} \times \mathcal{L}ab \times \mathsf{Distr}(\mathsf{PCCS})$ *is the smallest relation satisfying the rules in Figure 5.*

We next give our encoding of *PCCS* into *PMA*.

Definition 4.19. *The encoding* $[\![\cdot]\!] : \mathsf{PCCS} \rightarrow \mathsf{PMA}$ *of PCCS processes into PMA processes is defined as follows:*

$$\begin{array}{rcl}
[\![\mathbf{0}]\!] & = & \mathbf{0} \\
[\![\overline{a}]\!] & = & \mathsf{n}(a)[\,\mathbf{0}\,] \\
[\![a.\sum_{i\in I} p_i.P_i]\!] & = & \mathsf{open}\,\mathsf{n}(a).\sum_{i\in I} p_i.[\![P_i]\!] \\
[\![P \mid Q]\!] & = & [\![P]\!] \mid [\![Q]\!] \\
[\![(\mathsf{new}\, a)\, P]\!] & = & (\mathsf{new}\,\mathsf{n}(a))\,[\![P]\!] \\
[\![\mathit{fix}_{\mathsf{A}} P]\!] & = & \mathit{fix}_{\mathsf{A}}[\![P]\!]
\end{array}$$

where $\mathsf{n} : \mathsf{Act} \rightarrow \mathsf{N}$ *is an injective mapping from the set actions of PCCS to the set of names of PMA. We also extend the encoding to map labels of* PCCS *to the actions of PMA as follows:*

$$[\![a]\!] = \mathsf{open}\,\mathsf{n}(a), \quad [\![\overline{a}]\!] = \overline{\mathsf{open}\,\mathsf{n}(a)} \quad \textit{and} \quad [\![\tau]\!] = \tau$$

and lift the mapping to distributions such that for any distribution $\mu \in \mathsf{Distr}(\mathsf{PCCS})$ *and process* $T \in \mathsf{PMA}$*:*

$$[\![\mu]\!](T) = \sum_{P \in \mathsf{PCCS} \wedge [\![P]\!] = T} \mu(P) \ .$$

The following proposition demonstrates the above translation preserve the behaviour of *PCCS* processes.

Proposition 4.20. *For any process $P \in \mathsf{PCCS}$ and action $\alpha \in \mathcal{L}ab$:*

$$P \xrightarrow{\alpha} \mu \quad \Leftrightarrow \quad [\![P]\!] \xrightarrow{[\![\alpha]\!]} [\![\mu]\!]\,.$$

Proof. The proof follows by induction on the structure of P. □

5. Probabilistic Ambient Logic

In this section we extend the Ambient Logic [24, 48] (see Section 3.2) to the probabilistic setting. We take the standard approach in probabilistic temporal logics, see for example PML [33] a probabilistic extension of HML [52] and PCTL [9, 44] a probabilistic extension of CTL [53], and replace the some time operator $\Diamond\,\phi$ with a probabilistically quantified version of the form $\mathsf{IP}_{\sim p}(\Diamond\,\phi)$ where $\sim\, \in \{<, \leq, \geq, >\}$ and $p \in [0,1]$. Intuitively, a process satisfies such a formula $\mathsf{IP}_{\sim p}(\Diamond\,\phi)$ if the probability of reaching a process satisfying the formula ϕ in the underlying semantics (probabilistic automata) satisfies the condition $\sim\! p$. Because of the nondeterminism present in the underlying semantic model (see Section 2), we must quantify this condition over all the possible resolutions of the nondeterminism.

Formally, the syntax of the Probabilistic Ambient Logic (PAL) is defined below.

Definition 5.1. *The set of logical formulae of the* Probabilistic Ambient Logic PAL *is given by the syntax:*

$$\begin{aligned}\phi ::= \mathbf{T} \mid \neg\phi \mid \phi\vee\phi \mid \mathbf{0} \mid \eta[\phi] \mid \phi|\phi \mid \text{✧}\phi \mid \phi@\eta \mid \phi \triangleright \phi \mid \forall x.\,\phi \mid \eta\circledR\phi \mid \phi \oslash \eta \mid \\ \mathsf{IP}_{\sim p}(\Diamond\,\phi) \qquad \text{(probabilistic some time)}\end{aligned}$$

where $x \in \mathsf{Var}$ and $\eta \in \mathsf{N} \cup \mathsf{Var}$, $\sim\in \{<, \leq, \geq, >\}$ and $p \in [0,1]$.

Definition 5.2. *The satisfaction relation $\models\subseteq \mathsf{PMA} \times \mathsf{PAL}$ is defined in the same way as the satisfaction relation for AL (see Definition 3.9) except that for the probabilistic operator: $P \models \mathsf{IP}_{\sim p}(\Diamond\,\phi)$ if and only if*

$$\mathit{Prob}^A_P\{\pi \in \mathit{Path}^A(P) \mid \pi \models \Diamond\,\phi\} \sim p \;\; \textit{for all adversaries } A \textit{ of } (\mathsf{PMA}, \{\tau\}, \rightarrow)$$

where $\pi \models \Diamond\,\phi$ if and only if there exists $i \in \mathrm{IN}$ such that $\pi(i) \models \phi$.

Examples of PAL properties include:

- $n[\,\mathsf{IP}_{\leq 0}(\Diamond\,\mathbf{0})\,]$ which states that there exists an ambient n and the chance that the process inside this ambient becomes inactive is always 0;
- $\phi \triangleright \mathsf{IP}_{\geq 1}(\Diamond\,\psi)$ which states that in any context where ϕ holds one can always make, with probability 1, ψ hold in the future;
- $n\,\circledR\,\mathsf{IP}_{\geq 0.4}(\Diamond\,\phi)$ which states that once placed inside the ambient n, with probability at least 0.4, ϕ will eventually become true;

- $\mathsf{IP}_{\leq 0.75}(\Diamond\,(\text{⌑}\,\phi))$ which states that the greatest chance that ϕ holds at any time in any place is at most 0.75.

Note that the the original Ambient Logic (see Section 3.2) can be encoded in PAL by mapping any formula $\Diamond\phi$ to the formula $\neg\mathsf{IP}_{\leq 0}(\Diamond\phi)$ (which states that there exists an adversary under which the probability of ϕ holding some time in the future is greater than 0). Furthermore, the semantics of AL and PAL coincide for the subset of of *PMA* which correspond to *MA* processes (i.e. which the guarded probabilistic choice operator is restricted to $M.1.P$).

We have seen in Section 3.2 that the equivalence relation induced by the Ambient Logic coincides with structural congruence - for the finite fragment of Mobile Ambients [39].

Definition 5.3. *For any $P, Q \in \mathsf{PMA}$, we write $P =_{\mathsf{PAL}} Q$ when, for any formula $\phi \in \mathsf{PAL}$, we have $P \models \phi$ if and only if $Q \models \phi$.*

Proposition 5.4. *Let $P, Q \in \mathsf{PMA}$. If $P \equiv Q$, then $P =_{\mathsf{PAL}} Q$.*

Proof. The proof follows by showing that if $P, Q \in \mathsf{PMA}$, $P \equiv Q$ and $\phi \in \mathsf{PAL}$, then $P \models \phi$ if and only if $Q \models \phi$. The proof is by induction on the structure of $\phi \in \mathsf{PAL}$ and extends the case for Mobile Ambients and the Ambient logic [24]. The extension concerns the case when $\phi = \mathsf{IP}_{\sim p}(\Diamond\,\psi)$ for some $\sim \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$ and $\psi \in \mathsf{PAL}$ which we now describe.

By induction we have that if $P, Q \in \mathsf{PMA}$ and $P \equiv Q$, then $P \models \psi$ if and only if $Q \models \psi$. Using Lemma 4.10 we have that $\equiv$ is a probabilistic bisimulation relation [33], and hence using the results of [42] preserves formulae of the logic PCTL [9]. Furthermore, we can express the formula $\mathsf{IP}_{\sim p}(\Diamond\,\psi)$ in the logic PCTL by means of the formula $\mathcal{P}_{\sim p}[\ \mathbf{T}\ \mathcal{U}\ a_\psi\ ]$ where a_ψ is the atomic proposition labelling all processes satisfying ψ. Combining these results with the induction hypothesis it follows that for any $P, Q \in \mathsf{PMA}$ such that $P \equiv Q$: $P \models \mathsf{IP}_{\sim p}(\Diamond\,\psi)$ if and only if $Q \models \mathsf{IP}_{\sim p}(\Diamond\,\psi)$ as required. □

The reverse direction is an open question, as is the exact relationship between this equivalence and barbed probabilistic bisimulation. Recall that in the non-probabilistic setting logical equivalence coincides with structural equivalence, and hence it also follows that logical equivalence is stronger than barbed bisimulation. From these results it follows that there exist processes that are barbed probabilistic bisimilar but not logically equivalent, for example:

$$\mathbf{0} \simeq_p (\mathsf{new}\,n)\,n[\,\mathbf{0}\,] \quad \text{while} \quad \mathbf{0} \not\models n[\mathbf{T}] \oslash n, \quad \text{and} \quad (\mathsf{new}\,n)\,n[\,\mathbf{0}\,] \models n[\mathbf{T}] \oslash n\ .$$

Therefore, the possible relationship between logical equivalence and barbed probabilistic bisimulation is either that logical equivalence is strictly finer than barbed probabilistic bisimulation or that the equivalences are incomparable. On the other hand, from Proposition 5.4, we have that that either PAL equivalence is strictly coarser than structural congruence or the equivalences coincide.

6. Virus Spreading

In this section we demonstrate how the Probabilistic Ambient Calculus can be used for specifying how a virus might spread through a network. In addition, we give examples of properties expressed in the Probabilistic Ambient Logic and results obtained for this case study after a manual translation of the model and properties into the probabilistic model checker PRISM [54, 55]. Note that the feasibility of this translation follows from the fact that the semantics of the example has a finite state space. This example is inspired by the models presented in [56, 21, 57] but considers nondeterminism not treated in these models.

The model consists of a network in which one node has been infected by a virus. We suppose that the virus, after infecting a node, can attempt to enter any of the neighbouring nodes and, if successful, try to infect the neighbour. Furthermore, we suppose that both the events of the virus entering a node and infecting a node are probabilistic. In the case of entering, this means that there is chance that the virus may fail to pass through the node's firewall undetected, while in the case of infection, when the virus tries to infect a node it may be detected and quarantined by the node's local software. On the other hand, we suppose that the choice as to which node (out of neighbouring nodes that are not infected) the virus attempts to infect next is nondeterministic. Note that the actual choice as to which node to infect next could depend on the precise topology of nodes present in the network; for example, the choice may be based on that fact that some of the neighbouring nodes are closer than others, because the virus is following some route, or because the virus tries to attack certain types of nodes first.

The network is a grid of $N \times N$ nodes with each node connected to four neighbours (the nodes that are above, below, to the left and to the right), except for the nodes on the border for which some of the neighbours are not present. Unlike [56, 21, 57], we model the situation in which the virus spawns/multiplies. That is, once a node is infected, the virus remains at that node and repeatedly tries to infect any of the neighbouring nodes while they remain uninfected.

The Probabilistic Ambient Calculus specification is given below.

$$\begin{aligned}
\mathit{System} &\stackrel{\text{def}}{=} (\mathit{Run} \mid \mathit{Node}_{1,1} \mid \mathit{Node}_{1,2} \mid \cdots \mid \mathit{Node}_{n,n-1} \mid \mathit{Virus}_{n,n}) \\
\mathit{Run} &\stackrel{\text{def}}{=} (\mathsf{open}\, \mathit{run}.\mathbf{0}) \mid \mathit{Run} \\
\mathit{Node}_{i,j} &\stackrel{\text{def}}{=} \mathit{node}_{i,j}[\,\mathbf{0}\,] \\
\mathit{Virus}_{i,j} &\stackrel{\text{def}}{=} v_{i,j}\Big[\mathsf{in}\, \mathit{node}_{i-1,j}.\big(p_{i-1,j}.\mathbf{0} + (1-p_{i-1,j}).\mathit{Infect}_{i,j}^{i-1,j}\,\big)\Big] \\
&\quad \mid v_{i,j}\Big[\mathsf{in}\, \mathit{node}_{i,j+1}.\big(p_{i,j+1}.\mathbf{0} + (1-p_{i,j+1}).\mathit{Infect}_{i,j}^{i,j+1}\,\big)\Big] \\
&\quad \mid v_{i,j}\Big[\mathsf{in}\, \mathit{node}_{i+1,j}.\big(p_{i+1,j}.\mathbf{0} + (1-p_{i+1,j}).\mathit{Infect}_{i,j}^{i+1,j}\,\big)\Big] \\
&\quad \mid v_{i,j}\Big[\mathsf{in}\, \mathit{node}_{i,j-1}.\big(p_{i,j-1}.\mathbf{0} + (1-p_{i,j-1}).\mathit{Infect}_{i,j}^{i,j-1}\,\big)\Big] \\
&\quad \mid \mathit{Virus}_{i,j} \\
\mathit{Infect}_{i,j}^{k,l} &\stackrel{\text{def}}{=} \mathit{run}\Big[\,\mathit{Activate}_{i,j}^{k,l}\,\Big] \mid \mathit{Virus}_{k,l} \\
\mathit{Activate}_{i,j}^{k,l} &\stackrel{\text{def}}{=} \mathsf{out}\, v_{i,j}.\big(\,q_{k,l}.\mathbf{0} + (1-q_{k,l}).(\,\mathsf{out}\, \mathit{node}_{k,l}.\mathsf{open}\, \mathit{node}_{k,l}.\mathsf{open}\, v_{k,l}.\mathbf{0}\,)\,\big)
\end{aligned}$$

The behaviour of the virus in the specification given above can be understood as follows. When the virus has infected any node, it can spawn and attempt to enter the ambient of any (uninfected) neighbouring node. If the virus has infected node $N_{i,j}$ and attempts to enter the neighbouring node $N_{k,l}$ by entering the ambient $\mathit{node}_{k,l}$, then

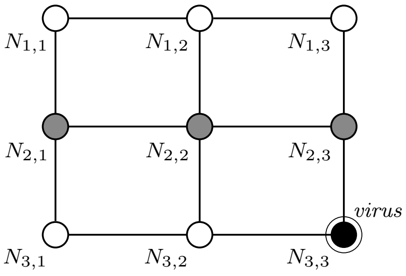


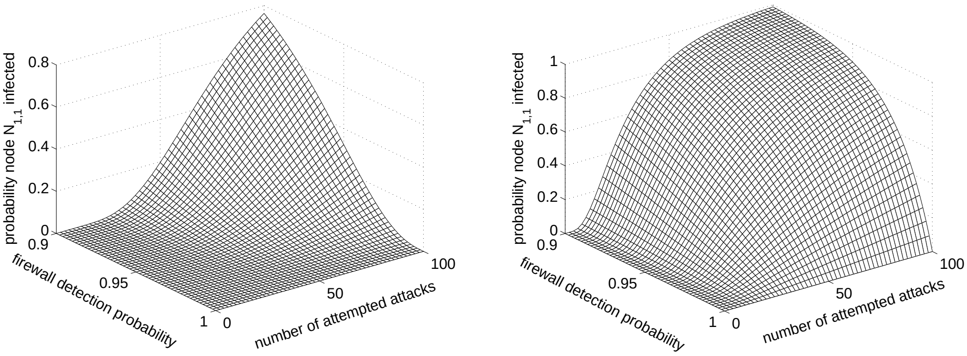
Figure 6: Network Configuration

with probability $p_{k,l}$ it will be blocked by the firewall, and with probability $1-p_{k,l}$ it will successfully pass through the node's firewall. When the virus succeeds, it next tries to infect the node. With probability $q_{k,l}$ the virus is detected and quarantined; with probability $1-q_{k,l}$, the infection succeeds. Infection by the virus opens (removes) the ambient $node_{i,j}$ preventing the virus from attacking an infected node in the future. In addition, it opens (removes) the ambient $v_{i,j}$ which enables the virus to spread to the uninfected neighbours of the newly infected node: before the ambient $v_{i,j}$ is opened, the virus present at the newly infected node ($Virus_{k,l}$) is inside the ambient $v_{i,j}$ (the virus has the form $v_{i,j}[\cdots\ v_{k,l}[\,\mathsf{in}\ node_{n,m}\ \cdots]\cdots]$), and hence is unable to attempt to spread to a neighbouring node $N_{n,m}$.

For our analysis we assume that the network is of size 3×3 and that the nodes $N_{2,1}$, $N_{2,2}$ and $N_{2,3}$ act as a barrier between the 'high' nodes ($N_{1,1}$, $N_{1,2}$ and $N_{1,3}$) and the 'low' nodes ($N_{3,1}$, $N_{3,2}$ and $N_{3,3}$) and are used to scan the traffic between these sets of nodes. A graphical representation of the network in its initial configuration is given in Figure 6. More precisely, we suppose that the probability $q_{i,j}$ equals 0.5 for each node, while the probability $p_{i,j}$ for any 'high' or 'low' node equals 0.5, and for the barrier nodes ($N_{2,1}$, $N_{2,2}$ and $N_{2,3}$) we vary $p_{i,j}$ from 1 to 0.9.

We consider properties relating to node $N_{1,1}$ getting infected by the virus which, since initially only the node $N_{3,3}$ is infected, requires that the virus passes through the barrier nodes. For example, consider the following Probabilistic Ambient Logic formulae:

- $⌑\, n_{2,2}[\neg\mathbf{0}]$ states that the virus has either infected node $N_{2,2}$ or entered the node without detection;
- $\mathbb{P}_{\geq 1}(\diamond(\mathbf{T} \mid v_{1,3}[\mathbf{T}]))$ states that with probability at least 1, the node $N_{1,3}$ is eventually infected by the virus;
- $\mathbf{T} \rhd \neg\mathbb{P}_{\leq 0.2}(\diamond(\mathbf{T} \mid v_{1,2}[\mathbf{T}]))$ states that, no matter what security measures are added to the network, it is possible that the node $N_{1,2}$ will become infected with probability greater than 0.2;
- $\mathbb{P}_{\leq 0}(\diamond(\mathbf{T} \mid node_{2,1}[\mathbf{0}] \mid node_{2,2}[\mathbf{0}] \mid node_{2,3}[\mathbf{0}] \mid v_{1,1}[\mathbf{T}]))$ states that the ('high') node $N_{1,1}$ cannot become infected if none of the 'barrier' nodes are infected;
- $\mathbb{P}_{\geq 0.5}(\diamond(\mathbf{T} \mid node_{2,1}[\mathbf{0}] \mid v_{2,2}[\mathbf{T}]))$ states that node $N_{2,2}$ becomes infected before node $N_{2,1}$ with probability at least 0.5.



(a) minimum probability

(b) maximum probability

Figure 7: Probability node $N_{3,3}$ gets infected after K attacks

Note that, by a simple adaptation of the model, one can incorporate a counter into the specification which counts the number of attempted attacks the virus undertakes, and then also specify properties relating to the minimum/maximum probability that a node is infected after at most k attacks.

Since this model is finite state, we have translated the semantics of this system into the probabilistic model checking tool PRISM [54, 55] and calculated both the minimum and maximum probability that node $N_{3,3}$ is eventually infected and is infected after the virus has performed at most k attacks, where k varies from 0 to 100. Both the minimum and maximum probability of node $N_{1,1}$ eventually becoming infected is 1, which is to be expected since in our model at any infected site the virus repeatedly tries to infect all neighbouring nodes.

The results concerning probability of infecting node $N_{1,1}$ after at most k attacks are presented in Figure 7. The first point to note is that, if the firewalls of the barrier nodes are completely secure ($p_{i,j}$ equals 1), then the virus cannot pass from the 'low' nodes to the 'high' nodes, and hence, no matter how many attacks are made, both the minimum and maximum probability of $N_{3,3}$ becoming infected is 0. The fact that there is a large difference between the minimum and maximum probabilities of infection is because the virus chooses nondeterministically which node to infect next, and in the maximum case the virus finds the quickest route to infect the node $N_{1,1}$, while in the minimum case the virus attempts to infect all other nodes before infecting the node $N_{1,1}$. Note that the minimum and maximum probabilities to infect *all* nodes after at most k attacks are the same; this is because the only choice the virus has is which neighbouring uninfected node to attack next.

7. Conclusions

We have introduced probabilistic versions of Mobile Ambients and Ambient Logic, with the aim of modelling randomised mobile distributed systems. We are able to carry out probabilistic model checking of probabilistic ambient processes against Ambient Logic specifications under the assumption that the underlying probabilistic automaton is finite state. In classical process calculi, such as *CCS* or the π-calculus, simple syntactic restrictions suffice to guarantee the finite-state property: namely, that recursive definition can occur only under prefix and cannot be composed with any other operator in the language. In the case of Mobile Ambients this restriction is too weak [58]. It is also shown in [58] that finite state ambient is achieved by means of a type system.

Future work will include investigation of how the finite control type system can be adapted to the probabilistic version of the calculus presented in this paper, with the aim of developing and implementing a probabilistic model checker for the Probabilistic Ambient Logic. In addition, we aim to answer the open question described in Section 5 regarding the relationship between structural congruence, barbed probabilistic bisimulation and logical equivalence.

Acknowledgments

The authors are supported in part by EPSRC grants GR/S11107, GR/S46727 and EP/D077273 and Microsoft Research Cambridge contract MRL 2005-44. The authors Kwiatkowska, Norman and Parker were in the School of Computer Science at the University of Birmingham when parts of this work were first carried out. We thank the anonymous referees for their valuable comments.

References

[1] R. Milner, Communication and concurrency, Prentice-Hall International, 1989.

[2] R. Milner, The polyadic π-calculus: a tutorial, in: F. Hamer, W. Brauer, H. Schwichtenberg (Eds.), Logic and Algebra of Specification, Springer-Verlag, 1993, pp. 203–246.

[3] C. Hoare, Communicating sequential processes, Prentice Hall, 1985.

[4] J. Baeten, W. Weijland, Process algebra, Cambridge University Press, 1990.

[5] Specification of the Bluetooth System (Version 1.2), Bluetooth SIG, 2003, `www.bluetooth.com`.

[6] S. Cheshire, B. Adoba, E. Guttman, Dynamic configuration of IPv4 link-local addresses (draft August 2002), zeroconf Working Group of the Internet Engineering Task Force (`www.zeroconf.org`).

[7] IEEE 1394-1995, High Performance Serial Bus Standard (1995).

[8] IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Standard (1997).

[9] H. Hansson, B. Jonsson, A logic for reasoning about time and reliability, Formal Aspects of Computing 6 (5) (1994) 512–535.

[10] R. van Glabbeek, S. Smolka, B. Steffen, Reactive, generative, and stratified models of probabilistic processes, Information and Computation 121 (1) (1995) 59–80.

[11] C. Baier, M. Kwiatkowska, Domain equations for probabilistic processes, Mathematical Structures in Computer Science 10 (6) (2000) 665–717.

[12] G. Lowe, Probabilistic and prioritized models of timed CSP, Theoretical Computer Science 138 (2) (1995) 315–352.

[13] O. Herescu, C. Palamidessi, Probabilistic asynchronous π-calculus, in: J. Tiuryn (Ed.), 3rd Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS'00), Vol. 1784 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 146–160.

[14] A. Di Pierro, C. Hankin, H. Wiklicky, Probabilistic KLAIM, in: R. De Nicola, G. Ferrari, G. Meredith (Eds.), Proc. Coordination Models and Languages, Vol. 2949 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 119–134.

[15] M. Vigliotti, P. Harrison, Stochastic mobile ambients, in: A. D. Pierro, H. Wiklicky (Eds.), Proc. 4th Int. Workshop Quantitative Aspects of Programming Languages (QAPL 2006), Vol. 164 (issue 3) of Electronic Notes in Theoretical Computer Science, Elsevier, 2006, pp. 169–186.

[16] C. Priami, A stochastic π-calculus, The Computer Journal 38 (7) (1995) 578–589.

[17] J. Hillston, A Compositional Approach to Performance Modelling, Cambridge University Press, 1996.

[18] M. Bernardo, R. Gorrieri, A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time, Theoretical Computer Science 202 (1–2) (1998) 1–54.

[19] H. Hermanns, U. Herzog, V. Mertsiotakis, Stochastic process algebras - between LOTOS and Markov chains, Computer Networks and ISDN (CNIS) 30 (9–10) (1998) 901–924.

[20] H. Hermanns, Interactive Markov Chains, Vol. 2428 of Lecture Notes in Computer Science, Springer-Verlag, 2002.

[21] R. De Nicola, D. Latella, M. Massink, Formal modeling and quantitative analysis of KLAIM-based mobile systems, in: Proc. ACM Symp. Applied Computing, ACM Press, 2005, pp. 428–435.

[22] R. Segala, Modelling and verification of randomized distributed real time systems, Ph.D. thesis, Massachusetts Institute of Technology (1995).

[23] L. Cardelli, A. Gordon, Types for mobile ambients, in: Proc. 26th ACM Symp. Principles of Programming Languages (POPL'99), ACM Press, 1999, pp. 79–92.

[24] L. Cardelli, A. Gordon, Anytime, anywhere. modal logic for mobile ambients, in: Proc. 27th ACM Symp. Principles of Programming Languages (POPL'00), ACM Press, 2000, pp. 365–377.

[25] H. Nielsen, F. Nielsen, Validating the firewalls in mobile ambients, in: J. Baeten, S. Mauw (Eds.), Proc. 10th Int. Conf. Concurrency Theory (CONCUR '99), Vol. 1664 of Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 463–477.

[26] F. Levi, D. Sangiorgi, Controlling interference for ambients, in: Proc. 28th ACM Symp. Principles of Programming Languages (POPL'01), ACM Press, 2000, pp. 352–364.

[27] M. Merro, M. Hennessy, Bisimulation congruences in safe ambients, in: Proc. 29th ACM Symp. Principles of Programming Languages (POPL'02), ACM Press, 2002, pp. 71–80.

[28] M. Bugliesi, G. Castagna, S. Crafa, Boxed ambients, in: N. Kobayashi, B. Pierce (Eds.), Proc. 4th Int. Symp. Theoretical Aspects of Computer Software (TACS 2001), Vol. 2215 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 38–63.

[29] L. Guan, Y. Yang, J. You, Making ambients more robust, in: N. Kurki-Suonio (Ed.), Proc. Int. Conf. Software: Theory and Practice, PHEI Press, 2000, pp. 377–384.

[30] I. Phillips, M. Vigliotti, On the reduction semantics for the push and pull ambient calculus, in: R. Baeza-Yates, U. Montanari, N. Santoro (Eds.), Proc. 2nd IFIP Int. Conf. Theoretical Computer Science (TCS 2002), Vol. 223 of IFIP Conference Proceedings, Kluwer, 2002, pp. 550–562.

[31] A. Regev, E. Panina, W. Silverman, L. Cardelli, E. Shapiro, Bioambients: an abstraction for biological compartments., Theoretical Computer Science 325 (1) (2004) 141–167.

[32] A. Phillips, N. Yoshida, S.Eisenbach, A distributed abstract machine for boxed ambient calculi, in: D. Schmidt (Ed.), Proc. 13th European Symp. Programming Languages and Systems (ESOP 2004), Vol. 2986 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 155–170.

[33] K. Larsen, A. Skou, Bisimulation through probabilistic testing, Information and Computation 94 (1) (1991) 1–28.

[34] B. Thomsen, Calculi for Higher Order Communicating Systems, Ph.D. thesis, Imperial College (1990).

[35] D. Sangiorgi, Expressing Mobility in Process Algebra: First-Order and Higher-Order Paradigms, Ph.D. thesis, University of Edinburgh (1993).

[36] R. Milner, D. Sangiorgi, Barbed bisimulation, in: W. Kuich (Ed.), Proc. 19th Int. Colloquium Automata, Languages and Programming (ICALP'92), Vol. 623 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 685–695.

[37] D. Hirschkoff, É. Lozes, D. Sangiorgi, Separability, expressiveness and decidability in the ambient logic, in: Proc. 17th Annual IEEE Symp. Logic in Computer Science (LICS'02), IEEE Computer Society Press, 2002, pp. 423–431.

[38] D. Hirschkoff, É. Lozes, D. Sangiorgi, On the expressiveness of the ambient logic., Logical Methods in Computer Science 2 (2).

[39] É. Lozes, Expressivité des logiques spatiales, Thèse de doctorat, Laboratoire de l'Informatique du Parallélisme, ENS Lyon, France (2004).

[40] D. Sangiorgi, Extensionality and intensionality of the ambient logics, in: Proc. 28th ACM Symp. Principles of Programming Languages (POPL'01), ACM Press, 2000, pp. 4–13.

[41] G. Berry, G. Boudol, Chemical abstract machine, in: Proc. 17th ACM Symp. Principles of Programming Languages (POPL'90), ACM Press, 1990, pp. 81–94.

[42] R. Segala, N. Lynch, Probabilistic simulations for probabilistic processes, Nordic Journal of Computing 2 (2) (1995) 250–273.

[43] C. Derman, Finite-State Markovian Decision Processes, New York: Academic Press, 1970.

[44] A. Bianco, L. de Alfaro, Model checking of probabilistic and nondeterministic systems, in: P. Thi-

agarajan (Ed.), Proc. 15th Conf. Foundations of Software Technology and Theoretical Computer Science, Vol. 1026 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 499–513.

[45] J. Kemeny, J. Snell, A. Knapp, Denumerable Markov Chains, 2nd Edition, Springer-Verlag, 1976.

[46] L. Cardelli, Abstraction for mobile computation, in: J. Vitek, C. Jensen (Eds.), Proc. Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Vol. 1603 of Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 51–94.

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

[48] L. Cardelli, A. Gordon, Logical properties of name restriction, in: S. Abramsky (Ed.), Proc. 5th Int. Conf. Typed Lambda Calculi and Applications (TLCA'01), Vol. 2044 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 46–60.

[49] N. Busi, G. Zavattaro, On the expressive power of movement and restriction in pure mobile ambients, Theoretical Computer Science 22 (3) (2004) 477–515.

[50] K. Honda, M. Tokoro, An object calculus for asynchronous communication, in: P. America (Ed.), Proc. European Conf. Object-Oriented Programming (ECOOP'91), Vol. 512 of Lecture Notes in Computer Science, Springer-Verlag, 1991, pp. 133–147.

[51] K. Honda, N. Yoshida, Noninterference through flow analysis, Journal Functial Programming 15 (2) (2005) 293–349.

[52] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, Journal of the ACM 32 (1) (1985) 137–161.

[53] E. Clarke, E. Emerson, A. Sistla, Automatic verification of finite-state concurrent systems using temporal logics, ACM Transactions on Programming Languages and Systems 8 (2) (1986) 244–263.

[54] A. Hinton, M. Kwiatkowska, G. Norman, D. Parker, PRISM: A tool for automatic verification of probabilistic systems, in: H. Hermanns, J. Palsberg (Eds.), Proc. 12th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06), Vol. 3920 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 441–444.

[55] PRISM web site, `www.prismmodelchecker.org`.

[56] A. Di Pierro, C. Hankin, H. Wiklicky, Continuous-time probabilistic KLAIM, in: R. Focardi, G. Zavattaro (Eds.), Proc. 2nd Int. Workshop Security Issues in Coordination Models, Languages, and Systems (SecCo 2004), Vol. 128 (issue 5) of Electronic Notes in Theoretical Computer Science, Elsevier, 2005, pp. 27–38.

[57] R. De Nicola, J.-P. Katoen, D. Latella, M. Massink, Towards a logic for performance and mobility, in: A. Cerone, H. Wiklicky (Eds.), Proc. 3rd Workshop Quantitative Aspects of Programming Languages (QAPL 2005), Vol. 153 (issue 2) of Electronic Notes in Theoretical Computer Science, Elsevier, 2006, pp. 161–175.

[58] W. Charatonik, A. Gordon, J. Talbot, Finite control ambients, in: D. L. Metayer (Ed.), Proc. 11th European Symp. Programming Languages and Systems (ESOP 2002), Vol. 2305 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 295–313.