

MODELLING PROBABILISTIC WIRELESS NETWORKS

ANDREA CERONE AND MATTHEW HENNESSY

e-mail address: acerone@scss.tcd.ie, matthew.hennessy@scss.tcd.ie

School of Computer Science and Statistics, Trinity College Dublin, Ireland

ABSTRACT. We propose a process calculus to model high level wireless systems, where the topology of a network is described by a digraph. The calculus enjoys features which are proper of wireless networks, namely broadcast communication and probabilistic behaviour.

We first focus on the problem of composing wireless networks, then we present a compositional theory based on a probabilistic generalisation of the well known may-testing and must-testing preorders. Also, we define an extensional semantics for our calculus, which will be used to define both simulation and deadlock simulation preorders for wireless networks. We prove that our simulation preorder is sound with respect to the may-testing preorder; similarly, the deadlock simulation preorder is sound with respect to the must-testing preorder, for a large class of networks.

We also provide a counterexample showing that completeness of the simulation preorder, with respect to the may testing one, does not hold. We conclude the paper with an application of our theory to probabilistic routing protocols.

1. INTRODUCTION

Wireless networks have spread worldwide in the last decades; nowadays they are used in many areas, from domestic appliances to mobile phone networks, to the newer sensor infrastructures. One of the main problems of wireless networks is that of defining and implementing protocols for providing to users the services for which the network has been designed; also, because of their distributed nature, a more challenging problem is that of ensuring in a rigorous, mathematical way, the correct behaviour of a network with respect to some specification.

This problem becomes even more difficult to tackle if we consider that often wireless networks run protocols whose behaviour is probabilistic. Such protocols are indeed very useful for improving the performance of wireless networks, examples being the use of probabilistic *routing protocols* [6] or probabilistic protocols for *collision avoidance* at the *MAC-sublayer* of the *TCP/IP* reference model [20]. Further, problems for which there is no solution in a deterministic setting can be solved (in unbounded time) by introducing probabilistic behaviour in wireless networks [1].

Many different formal frameworks have been developed in the literature for defining and analysing the behaviour of wireless networks [25, 16, 13, 21, 32, 23, 11, 33, 36]; these differ in many details, the most important being the level of abstraction used to represent a wireless network, the computational power of stations of wireless networks and the mathematical structure used to represent the topology of wireless networks. However, each of these calculi have the following features in common: wireless networks are represented as a collection of stations (also called nodes, or locations) running code, and local broadcast is used as the only way of communication. Roughly

Received by the editors August 14, 2013.

The financial support of SFI is gratefully acknowledged.

speaking in local broadcast communication, whenever a node broadcasts a message only the nodes in its range of transmission are affected.

In this paper we propose another process calculus for modelling probabilistic wireless networks; the main concepts underlying our calculus can be summarised as follows:

- (i) The topology of a wireless network is static, that is mobility is not considered in our model. This restriction has been done to allow a more clear presentation of the topics treated in this paper; however, we could have used the approach described in [10] to introduce node mobility. The network topology is described by a digraph Γ ; intuitively vertices in this graph represent network locations, while an edge from a node to another is used for expressing that the latter is in the range of transmission of the former.
- (ii) A probabilistic process calculus is defined for assigning code to locations. The basic constructs allowed in our calculus are messages' broadcast and reception, internal actions, matching and process definitions; further, we allow a special clause ω whose role will be presented shortly. The mapping that assigns code to locations is partial, meaning that locations can have no code assigned. At least informally, such nodes, which will be called *external*, can be seen as terminals at which users can place code to test the behaviour of the network.
- (iii) Communication between nodes is reliable; a message broadcast by a node along a channel c will be received by all the nodes in the sender's range of transmission, provided that they are waiting to receive a message along such a channel. In other words, our calculus is designed for describing wireless networks at the *network layer* of the *TCP/IP reference model*; reliable communication is not ensured at lower levels, where issues such as the possibility of collisions [20] and synchronisation between nodes [29] arise.

One of the main goals of the paper is that of defining a compositional behavioural theory of wireless networks; given two wireless networks \mathcal{M} and \mathcal{N} , we want to establish whether they can be distinguished by an external user. To accomplish this task, we need to address several different topics. First, it is necessary to define how two wireless networks can be composed together. This topic has already been addressed, for different process calculi, in [23, 11, 5, 2, 33]. Here we define an asymmetric operator \sharp which can be used to extend one network with another. Despite being asymmetric, we show that the choice of the operator \sharp is driven by some natural requirements we require in general from a composition operator between networks. We remark that our theory of composition is restricted to a particular class of networks, which we call *well-formed*.

Once we have chosen a suitable composition operator \sharp , we can define a compositional theory for wireless networks. In this paper we have chosen to focus on a probabilistic generalisation of the well-known *De Nicola and Hennessy's testing preorders*, whose theory has been defined in [7] for a probabilistic version of *CSP*.

Informally speaking, we can test a wireless network \mathcal{M} via another wireless network \mathcal{T} which can be composed with the former (with respect to the operator \sharp); that is, the network $(\mathcal{M} \sharp \mathcal{T})$ is defined. Intuitively, the network $\mathcal{M} \sharp \mathcal{T}$ can be considered as an experiment in which the role of the testing component \mathcal{T} is that of determining whether \mathcal{M} satisfies some property for which the test has been designed for. The success of an experiment is denoted by the special construct of our calculus ω mentioned above.

Having this in mind, each computation of the network $(\mathcal{M} \sharp \mathcal{T})$ induces a success outcome, denoting the probability of reaching a configuration in which the special clause ω is enabled in such a computation. This induces a set of success outcomes for the network $(\mathcal{M} \sharp \mathcal{T})$ by quantifying over all the possible computations for such a network.

Knowing how to associate a set of success outcomes to a network, we can compare two networks \mathcal{M}, \mathcal{N} by quantifying over all possible tests \mathcal{T} , and comparing the sets of success outcomes

of the experiments $(\mathcal{M} \# \mathcal{T})$ and $(\mathcal{N} \# \mathcal{T})$, provided that they are both defined. This leads to the definition of two testing preorders, the *may-testing* preorder \sqsubseteq_{may} and *must-testing* preorder $\sqsubseteq_{\text{must}}$, according to the way in which the sets of success outcomes for the two experiments above are compared.

It is important to note that determining directly whether the statement $\mathcal{M} \sqsubseteq_{\text{may}} \mathcal{N}$ ($\mathcal{M} \sqsubseteq_{\text{must}} \mathcal{N}$) is true is not easy, due to the quantification over all tests. Therefore there is the need to define a proof methodology for establishing if two networks can be related via the \sqsubseteq_{may} ($\sqsubseteq_{\text{must}}$) preorder. This is the main topic of our paper. To this end, we define an extensional semantics for our calculus of wireless networks; the actions in this semantics correspond to activities that can be observed by the external nodes. The main idea here is that of defining sound coinductive proof methods for the testing preorders, based on the extensional behaviour of networks.

Since our calculus is equipped with local broadcast communication, we need to take care of some issues in the development of such proof methods; roughly speaking, the broadcast of a message to a set of external nodes can be simulated by a multicast of the same message which can be detected by the same set of external nodes. This leads to a non-standard definition of weak extensional actions, which will be used to define two coinductive relations between networks. The first one is the well-known *simulation preorder* [7]; the second one is a novel preorder, called the *deadlock simulation preorder*, which is obtained from the previous one by adding sensitivity to deadlock configurations. The main results of the papers are that, for a large class of networks, the simulation preorder is sound with respect to the may-testing preorder, while the inverse of the deadlock simulation preorder is sound with respect to the must-testing preorder. However, we provide a counterexample that shows that such proof methods fail to be complete.

The rest of the paper is organised as follows: in Section 2 we recall the mathematical tools needed for the development of our theory.

In Section 3 we define the syntax and intensional semantics of our calculus of wireless networks, and we prove some basic properties of our calculus.

In Section 4 we give the formal definition of the behavioural preorders between networks. This depends on how tests are applied to networks or more generally how networks are composed to form larger networks. So we first define our composition operator $\#$, which is asymmetric, in Section 4.1 and then use it to develop the behavioural preorders \sqsubseteq_{may} and $\sqsubseteq_{\text{must}}$ between networks. In Section 4.4 we return to our choice of composition operator $\#$, justifying it as the largest one which satisfies three natural requirements. In addition, somewhat surprisingly, we show that any symmetric composition operator satisfying the natural requirements generates a degenerate behavioural theory.

In Section 5 we define the extensional semantics of our calculus of wireless networks; here we also give the non-standard definition of weak extensional actions and we prove composition and decomposition results for them, with respect to the composition operator $\#$.

In Section 6 we define the notions of simulation and deadlock simulation preorders and we prove the main results of the paper, namely that the simulation preorder is sound with respect to the may-testing preorder, and the inverse of the deadlock simulation preorder is sound with respect to the must-testing preorder. Much of the technical development underlying these soundness results is relegated to the separate Section 7; this may be safely skipped by the uninterested reader.

In Section 8 we show that our proof methods fail to be complete; we also show the impossibility of defining a coinductive relation based on our notion of extensional actions, which characterises the may-testing relation.

In Section 9 we consider an application of our theory by analysing a simple probabilistic, connectionless routing protocol, showing that it is behaviourally equivalent to a formal specification.

We conclude our paper by summarising the topics we have covered and by illustrating the related work in Section 10. The topics covered in this paper were also the subject of an extended abstract [4].

2. BACKGROUND

In this Section we summarise the mathematical concepts, taken from [7], that will be needed throughout the paper. First we introduce some basic concepts from probability theory; then we show how these can be used to model concurrent systems which exhibit both probabilistic and non-deterministic behaviour.

Let S be a set; a function $\Delta : S \rightarrow [0, 1]$ is called a (probability) sub-distribution over S if $\sum_{s \in S} \Delta(s) \leq 1$. This quantity, $\sum_{s \in S} \Delta(s)$, is called the mass of the sub-distribution, denoted as $|\Delta|$. If $|\Delta| = 1$, then we say that Δ is a (full) distribution. The support of a sub-distribution Δ , denoted $\lceil \Delta \rceil$, is the subset of S consisting of all those elements which contribute to its mass, namely $\lceil \Delta \rceil = \{s \in S \mid \Delta(s) > 0\}$.

For any set S , the empty sub-distribution $\varepsilon \in \mathcal{D}_{sub}(S)$ is the only sub-distribution with empty support, that is $\lceil \varepsilon \rceil = \emptyset$. For each $s \in S$, the point distribution \bar{s} is defined to be the distribution which takes value 1 at s , and 0 elsewhere. The set of sub-distributions and distributions over a set S are denoted by $\mathcal{D}_{sub}(S)$ and $\mathcal{D}(S)$, respectively.

Given a family of sub-distributions $\{\Delta_k\}_{k \in K}$, $(\sum_{k \in K} \Delta_k)$ is the partial real-valued function in $S \rightarrow \mathbb{R}$ defined by $(\sum_{k \in K} \Delta_k)(s) := \sum_{k \in K} \Delta_k(s)$. This is a partial operation because for a given $s \in S$ this sum might not exist; it is also a partial operation on sub-distributions because even if the sum does exist it may be greater than 1.

Similarly, if $p \leq 1$ and Δ is a sub-distribution, then $p \cdot \Delta$ is the sub-distribution over S such that $(p \cdot \Delta)(s) = p \cdot \Delta(s)$.

It is not difficult to show that if $\{p_k\}_{k \in K}$ is a sequence of positive real numbers such that $\sum_{k \in K} p_k \leq 1$, and $\{\Delta_k\}_{k \in K}$ is a family of sub-distributions over a set S , then $(\sum_{k \in K} p_k \cdot \Delta_k)$ always defines a sub-distribution over S . Further, if $\sum_{k \in K} p_k = 1$ and each Δ_k is a distribution, then $(\sum_{k \in K} p_k \cdot \Delta_k)$ is a distribution.

Finally, if $f : X \rightarrow Y$ and Δ is a sub-distribution over X then we use $f(\Delta)$ to be the sub-distribution over Y defined by:

$$f(\Delta)(y) = \sum_{x \in X} \{\Delta(x) \mid f(x) = y\}. \quad (2.1)$$

This definition can be generalised to two arguments functions; if $f : X_1 \times X_2 \rightarrow Y$ is a function, and Δ, Θ are two sub-distributions respectively over X_1 and X_2 , then $f(\Delta, \Theta)$ denotes the sub-distribution over Y defined as

$$f(\Delta, \Theta)(y) = \sum_{x_1 \in X_1, x_2 \in X_2} \{\Delta(x_1) \cdot \Theta(x_2) \mid f(x_1, x_2) = y\}. \quad (2.2)$$

Now we turn our attention to probabilistic concurrent systems. The formal model we use to represent them is a generalisation to a probabilistic setting of Labelled Transition Systems (LTSs) [24].

Definition 2.1. A *probabilistic labelled transition system* (pLTS) is a 4-tuple $\langle S, \text{Act}_\tau, \rightarrow, \omega \rangle$, where

- (i) S is a set of states,
- (ii) Act_τ is a set of transition labels with a distinguished label τ ,
- (iii) the relation \rightarrow is a subset of $S \times \text{Act}_\tau \times \mathcal{D}(S)$,

(iv) $\omega : S \mapsto \{ \text{true}, \text{false} \}$ is a (success) predicate over the states S .

As usual, we will write $s \xrightarrow{\mu} \Delta$ in lieu of $(s, \alpha, \Delta) \in \longrightarrow$. \square

Before discussing pLTSs, some definitions first: a pLTS whose state space is finite is said to be finite state; further, we say that a pLTS $\langle S, \text{Act}_\tau, \rightarrow, \omega \rangle$ is finite branching if, for every $s \in S$, the set $\{ \Delta \mid s \xrightarrow{\mu} \Delta \text{ for some } \mu \in \text{Act}_\tau \}$ is finite. Finally, a finitary pLTS is one which is both finite state and finite branching.

We have included in the definition of a pLTS a success predicate ω over states, which will be used when testing processes. Apart from this, the only difference between LTSs and pLTSs is given by the definition of the transition relation; in the latter this is defined to be a relation (parametric in some action μ) between states and distributions of states, thus capturing the concept of probabilistic behaviour.

However, this modification introduces some difficulties when sequences of transitions performed by a given pLTS have to be considered, as the domain and the image of the transition relation do not coincide. To avoid this problem, we will focus only on distributions of states by defining transitions for them. The following Definition serves to this purpose:

Definition 2.2 (Lifted Relations). Let $\mathcal{R} \subseteq S \times \mathcal{D}_{\text{sub}}(S)$ be a relation from states to sub-distributions. Then $\bar{\mathcal{R}} \subseteq \mathcal{D}_{\text{sub}}(S) \times \mathcal{D}_{\text{sub}}(S)$ is the smallest relation which satisfies

- $s \mathcal{R} \Delta$ implies $\bar{s} \bar{\mathcal{R}} \Delta$
- If I is a finite index set and $\Delta_i \bar{\mathcal{R}} \Theta_i$ for each $i \in I$ then $(\sum_{i \in I} p_i \cdot \Delta_i) \mathcal{R} (\sum_{i \in I} p_i \cdot \Theta_i)$ whenever $(\sum_{i \in I} p_i) \leq 1$. \square

Lifting can also be defined for relations from states to probability distributions, by simply requiring $\sum_{i \in I} p_i = 1$ in the last constraint of the definition above.

Sometimes it will be convenient to consider also the lifting of relations of the form $\mathcal{R} \subseteq S \times S$; this is defined by first lifting the relation \mathcal{R} to $\mathcal{R}^e \subseteq S \times \mathcal{D}_{\text{sub}}(S)$, by letting $s \mathcal{R}^e \Theta$ iff $\Theta = \bar{t}$ for some $t \in S$ such that $s \mathcal{R} t$. Then we obtain the relation $\bar{\mathcal{R}}^e$ by applying Definition 2.2 to \mathcal{R}^e .

In a pLTS $\langle S, \text{Act}_\tau, \rightarrow, \omega \rangle$, each transition relation $\xrightarrow{\mu} \subseteq S \times \mathcal{D}(S)$ can be lifted to $(\xrightarrow{\mu}) \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$. With an abuse of notation, the latter will still be denoted as $\xrightarrow{\mu}$.

Lifted transition relations allow us to reason about the behaviour of pLTSs in terms of sequences of transitions; here we are mainly interested in the behaviour of a pLTS in the long run; that is, given a pLTS $\langle S, \text{Act}_\tau, \rightarrow, \omega \rangle$ and a sub-distribution $\Delta \subseteq \mathcal{D}_{\text{sub}}(S)$, we are interested in the sub-distributions $\Theta \subseteq \mathcal{D}_{\text{sub}}(S)$ which can be reached from Δ after an unbounded number of transitions.

For the moment we will focus only on internal actions of a pLTS, in which case the behaviour of a pLTS in the long run is captured by the concept of hyper-derivation:

Definition 2.3. [Hyper-derivations] In a pLTS a hyper-derivation consists of a collection of sub-distributions $\Delta, \Delta_k^{\rightarrow}, \Delta_k^{\times}$, for $k \geq 0$, with the following properties:

$$\begin{array}{rcl}
 \Delta & = & \Delta_0^{\rightarrow} + \Delta_0^{\times} \\
 \Delta_0^{\rightarrow} & \xrightarrow{\tau} & \Delta_1^{\rightarrow} + \Delta_1^{\times} \\
 & \vdots & \\
 \Delta_k^{\rightarrow} & \xrightarrow{\tau} & \Delta_{k+1}^{\rightarrow} + \Delta_{k+1}^{\times} \\
 & \vdots &
 \end{array}$$

If $\omega(s) = \text{false}$ for each $s \in [\Delta_k^{\rightarrow}]$ and $k \geq 0$ we call $\Delta' = \sum_{k=0}^{\infty} \Delta_k^{\times}$ a *hyper-derivative* of Δ , and write $\Delta \Longrightarrow \Delta'$. \square

We will often write $s \Longrightarrow \Delta$ in lieu of $\bar{s} \Longrightarrow \Delta$.

Example 2.4. Let us illustrate how hyper-derivations can be inferred in a pLTS via a simple example. A central role in hyper-derivations will be played by the empty sub-distribution ε . Note that, in any pLTS $\langle S, \text{Act}_\tau, \rightarrow, \omega \rangle$, for any action $\alpha \in \text{Act}_\tau$ we have that $\varepsilon \xrightarrow{\alpha} \varepsilon$.

Let us consider a pLTS whose state space is given by the set $\{h, t\}$, with the only transition $s \xrightarrow{\tau} 1/2 \cdot \bar{h} + 1/2 \cdot \bar{t}$ and with $\omega(t) = \text{true}$. This pLTS models a probabilistic experiment in which we continuously toss a fair coin until we obtain the outcome tail (represented by the state t), in which case we decree that the experiment succeeded; this last constraint is represented by letting $\omega(t) = \text{true}$. It is well-known, from elementary probability arguments, that the probability of obtaining a success before the coin has been tossed k times is $\frac{2^k - 1}{2^k}$, while in the long run the experiment will succeed with probability 1. This behaviour can be inferred by using hyper-derivations. For example, for any $k \geq 0$ we can consider the infinite sequence of transitions

$$\begin{array}{rclclcl}
\Delta_0^{\rightarrow} & = & \bar{h} & \xrightarrow{\tau} & \frac{1}{2} \cdot \bar{h} & + & \frac{1}{2} \cdot \bar{t} \\
\Delta_1^{\rightarrow} & = & \frac{1}{2} \cdot \bar{h} & \xrightarrow{\tau} & \frac{1}{2^2} \cdot \bar{h} & + & \frac{1}{2^2} \cdot \bar{t} \\
& \vdots & & \xrightarrow{\tau} & \vdots & & \vdots \\
\Delta_{k-2}^{\rightarrow} & = & \frac{1}{2^{k-2}} \cdot \bar{h} & \xrightarrow{\tau} & \frac{1}{2^{k-1}} \cdot \bar{h} & + & \frac{1}{2^{k-1}} \cdot \bar{t} \\
\Delta_{k-1}^{\rightarrow} & = & \frac{1}{2^{k-1}} \bar{h} & \xrightarrow{\tau} & \varepsilon & + & \frac{1}{2^k} \cdot \bar{h} + \frac{1}{2^k} \cdot \bar{t} \\
\Delta_k^{\rightarrow} & = & \varepsilon & \xrightarrow{\tau} & \varepsilon & + & \varepsilon \\
& \vdots & & \xrightarrow{\tau} & \vdots & & \vdots \\
\Delta_{k+n}^{\rightarrow} & = & \varepsilon & \xrightarrow{\tau} & \varepsilon & + & \varepsilon \\
& \vdots & & \xrightarrow{\tau} & \vdots & & \vdots
\end{array}$$

Note that the sequence of transitions above models a situation in which the experiment is stopped after the coin has been tossed k times. This is done by letting $\Delta_k^{\rightarrow} = \varepsilon$; at least informally this means that the computation proceeds with probability 0 after the k -th τ -transition has been performed. The sequence of transitions above leads to the hyper-derivation

$$\begin{aligned}
h & \Longrightarrow \left(\sum_{i=1}^{k-1} \frac{1}{2^i} \cdot \bar{t} \right) + \left(\frac{1}{2^k} \bar{h} + \frac{1}{2^k} \bar{t} \right) + \left(\sum_{i=k+1}^{\infty} \varepsilon \right) = \\
& = \frac{1}{2^k} \cdot \bar{h} + \left(\sum_{i=1}^k \frac{1}{2^i} \cdot \bar{t} \right) = \\
& = \frac{1}{2^k} \cdot \bar{h} + \frac{2^k - 1}{2^k} \cdot \bar{t}
\end{aligned}$$

That is, after k transitions have been performed the probability of having successfully terminated the experiment is $(2^k - 1)/2^k$.

Further, note that we can use hyper-derivations to describe the limiting behaviour of the experiment. In fact, we can consider the infinite sequence of transitions

$$\begin{array}{rccccccc}
 \bar{h} & & \xrightarrow{\tau} & & \frac{1}{2} \cdot \bar{h} & + & \frac{1}{2} \cdot \bar{t} \\
 \frac{1}{2} \cdot \bar{h} & & \xrightarrow{\tau} & & \frac{1}{2^2} \cdot \bar{h} & + & \frac{1}{2^2} \cdot \bar{t} \\
 \vdots & & \xrightarrow{\tau} & & \vdots & & \vdots \\
 \frac{1}{2^k} \cdot \bar{h} & & \xrightarrow{\tau} & & \frac{1}{2^{k+1}} \cdot \bar{h} & + & \frac{1}{2^{k+1}} \cdot \bar{t} \\
 \vdots & & \xrightarrow{\tau} & & \vdots & & \vdots
 \end{array}$$

which leads to the hyper-derivation

$$h \Longrightarrow \left(\sum_{i=1}^{\infty} \frac{1}{2^i} \cdot \bar{t} \right) = \bar{t}$$

□

Hyper-derivations can be seen as the probabilistic counterpart of the weak $\xrightarrow{\tau}$ action in LTSs; see [7] for a detailed discussion. Intuitively speaking, they represent fragments of computations obtained by performing only internal actions. The last constraint in Definition 2.3 is needed since we introduced a success predicate in our model; as we see pLTSs as nondeterministic, probabilistic experiments, we require that a computation stops when the experiment succeeds, that is when a state s such that $\omega(s) = \text{true}$ has been reached. States in which the predicate $\omega(\cdot)$ is true are called ω -successful.

Further, we are mainly interested in maximal computations of distributions. That is, we require a computation to proceed as long as some internal activity can be performed. To this end, we say that $\Delta \Longrightarrow \Delta'$ if

- $\Delta \Longrightarrow \Delta'$,
- for every $s \in \lceil \Delta' \rceil$, $s \xrightarrow{\tau}$ implies $\omega(s) = \text{true}$.

□

This is a mild generalisation of the notion of *extreme derivative* from [7]. Note that the last constraint models exactly the requirement of performing some internal activity whenever it is possible; In other words extreme derivatives correspond to a probabilistic version of maximal computations.

Example 2.5. Consider again the pLTS of Example 2.4. Here we have that the hyper-derivation $h \Longrightarrow \Theta = \frac{1}{2^k} \cdot \bar{h} + \frac{2^k-1}{2^k} \cdot \bar{t}$, where $k \geq 0$, is not an extreme derivation, since $\omega(h) = \text{false}$ and $h \xrightarrow{\tau}$. On the other hand, the hyper-derivation $h \Longrightarrow \bar{t}$ is also an extreme derivation, since $\omega(\bar{t}) = \text{true}$; therefore $h \Longrightarrow \bar{t}$. □

Theorem 2.6. In an arbitrary pLTS

- (1) \Longrightarrow is reflexive and transitive,
- (2) if $\Delta \Longrightarrow \Delta'$ and $\Delta' \Longrightarrow \Delta''$, then $\Delta \Longrightarrow \Delta''$; this is a direct consequence of the previous statement, and the definition of extreme derivatives,
- (3) suppose $\Delta = (\sum_{i \in I} p_i \cdot \Delta_i)$, where I is an index set and $\sum_{i \in I} p_i \leq 1$. If for any $i \in I$, $\Delta_i \Longrightarrow \Theta_i$ for some Θ_i , then $\Delta \Longrightarrow \Theta$, where $\Theta = (\sum_{i \in I} p_i \cdot \Theta_i)$,
- (4) for all sub-distributions Δ , there exists a sub-distribution Θ such that $\Delta \Longrightarrow \Theta$.

Proof. See [7] for detailed proofs. □

The last definition we need is that of convergent pLTSs. Intuitively these are pLTSs whose infinite computations have a negligible probability.

Definition 2.7. [Convergence] A pLTS $\langle S, \text{Act}_\tau, \longrightarrow, \omega \rangle$ is said to be convergent if $s \Longrightarrow \varepsilon$ for no state $s \in S$.

At least informally, $s \Longrightarrow \varepsilon$ means that there exists a computation rooted in s which contains only probability sub-distributions which can always perform a τ -action. See [7], Section 6 for a detailed discussion on divergence in pLTSs. The main property we will require from convergent pLTSs is the following:

Proposition 2.8. Let Δ be a distribution in a convergent pLTS. If $\Delta \Longrightarrow \Theta$ then $|\Theta| = 1$.

Proof. This is an immediate consequence of *Distillation of Divergence*, Theorem 6.20 of [7]. □

3. THE CALCULUS

In this Section we introduce our calculus for modelling wireless networks. In this calculus, a wireless network is modelled as a pair of the form $(\Gamma \triangleright M)$, where Γ is a digraph representing the topology of a wireless network and M is a term which assigns code to nodes.

The syntax of our calculus is presented in Section 3.1; here we also give some basic examples of wireless networks. In Section 3.2 we formalise how networks evolve by introducing an intensional semantics for our calculus; finally we prove some basic properties of our calculus in Section 3.3.

3.1. Syntax. The calculus we present is designed to model broadcast systems, particularly wireless networks, at a high level. We do not deal with low level issues, such as collisions of broadcast messages or multiplexing mechanisms [35]; instead, we assume that network nodes use protocols at the *MAC level* [20] to achieve a reliable communication between nodes.

Basically, the language will contain both primitives for sending and receiving messages and will enjoy the following features:

- (i) communication can be obtained through the use of different channels; although the physical medium for exchanging messages in wireless networks is unique, it is reasonable to assume that network nodes use some multiple access technique, such as *TDMA* or *FDMA* [35], to setup and communicate through virtual channels,
- (ii) communication is broadcast; whenever a node in a given network sends a message, it can be detected by all nodes in its range of transmission,
- (iii) communication is reliable; whenever a node broadcasts a message and a neighbouring node (that is, a node in the sender's range of transmission) is waiting to receive a message on the same channel, then the message will be delivered to the receiver. This is not ensured if low level issues are considered, as problems such as message collisions [20] and nodes synchronisation [29] arise .

The language for system terms, ranged over by M, N, \dots is given in Figure 1. Basically a system consists of a collection of named nodes at each of which there is some running code. The syntax for this code is a fairly straightforward instance of a standard process calculus, augmented by a probabilistic choice; code descriptions have the usual constructs for channel based communication, with input $c?(x).p$ being the unique binder. This gives rise to the standard notions of alpha-conversion, free and bound occurrences of variables in system terms and closed system terms.

$M, N ::=$	$n[s]$ $M N$ $\mathbf{0}$	Systems Nodes Composition Identity
$p, q ::=$	s $p_p \oplus q$	(probabilistic) Processes probabilistic choice
$s, t ::=$	$c!(e).p$ $c?(x).p$ $\omega.\mathbf{0}$ $s + t$ if b then s else t $\tau.p$ $A(\vec{x})$ $\mathbf{0}$	States broadcast receive test choice branch preemption definitions terminate

Figure 1: Syntax

We only consider the sub-language of well-formed system terms in which all node names have at most one occurrence. We use \mathbf{sSys} to range over all closed well-formed terms. A (well-formed) system term can be viewed as a mapping that assigns to node names the code they are executing. A sub-term $n[s]$ appearing in a system term M represents node n running code s . In the following we make use of standard conventions; we omit trailing occurrences of $\mathbf{0}$ and we use $\prod_{i=1}^n m_i[s_i]$ to denote the system term $m_1[s_1] | \dots | m_n[s_n]$.

Additional information such as the connectivity between nodes of a network is needed to formalise communications between nodes. Network connectivity is represented by a graph $\Gamma = \langle \Gamma_V, \Gamma_E \rangle$; here Γ_V is a finite set of nodes and $\Gamma_E \subseteq (\Gamma_V \times \Gamma_V)$ is an irreflexive relation between nodes in Γ_V . Intuitively, $(m, n) \in \Gamma_E$ models the possibility for node n to detect broadcasts fired by m .

We use the more graphic notation $\Gamma \vdash v$ to mean $v \in \Gamma_V$ and $\Gamma \vdash m \rightarrow n$ for $(m, n) \in \Gamma_E$. Similarly we use $\Gamma \vdash n \leftarrow m$ to denote $\Gamma \vdash m \rightarrow n$. Sometimes we also use the notations $\Gamma \vdash m \leftrightarrow n$ for $\{(n, m), (m, n)\} \subseteq \Gamma_E$ and $\Gamma \vdash m \rightleftarrows n$ to denote either $\Gamma \vdash m \rightarrow n$ or $\Gamma \vdash m \leftarrow n$.

A *network* consists of a pair $(\Gamma \triangleright M)$, representing the system M , from \mathbf{sSys} , executing relative to the connectivity graph Γ . All nodes occurring in M , $\text{nodes}(M)$, will appear in Γ and the effect of running the code at $n \in \text{nodes}(M)$ will depend on the connectivity of n in Γ . But in general there will be nodes in Γ which do not occur in M ; let $\text{Intf}(\Gamma \triangleright M) = \Gamma_V \setminus \text{nodes}(M)$; we call this set the *interface* of the network $\Gamma \triangleright M$, and its elements are called *external nodes*. Intuitively these are nodes which may be used to compose the network $\Gamma \triangleright M$ with other networks, or to place code for testing the behaviour of M .

In the following we use the meta-variables $\mathcal{M}, \mathcal{N}, \dots$ to range over networks. Also, the notation introduced for system terms and connectivity graphs is extended to networks in the obvious way; for example, if $\mathcal{M} = (\Gamma \triangleright M)$, $\text{nodes}(\mathcal{M}) = \text{nodes}(M)$, $\mathcal{M}_V = \Gamma_V$ and $\mathcal{M} \vdash m \rightarrow n$ denotes $\Gamma \vdash m \rightarrow n$.

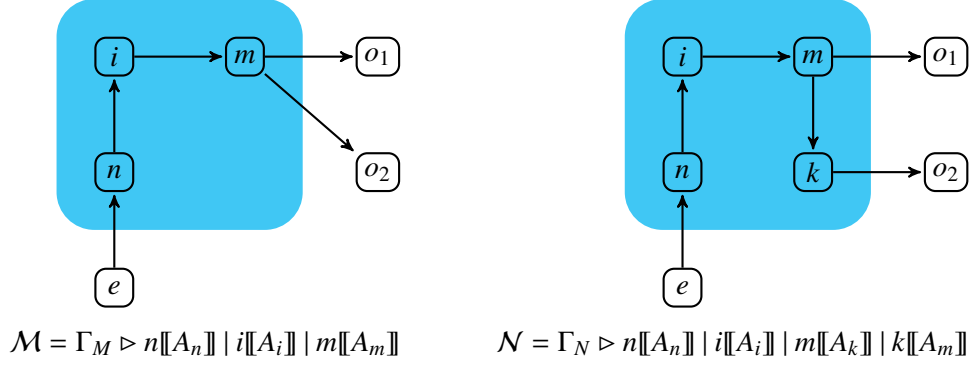


Figure 2: Example networks

Example 3.1. Consider \mathcal{M} described in Figure 2. There are six nodes, three occupied by code n , i and m , and three in the interface $\text{Intf}(\mathcal{M})$, e , o_1 and o_2 . Here, and in future examples, we differentiate between the interface and the occupied (internal) nodes using shading. Suppose the code at nodes is given by

$$A_n \Leftarrow c?(x) . d!\langle x \rangle . \mathbf{0} \quad A_i \Leftarrow d?(x) . d!\langle f(x) \rangle . \mathbf{0} \quad A_m \Leftarrow d?(x) . (d!\langle x \rangle . \mathbf{0}_{0.8} \oplus \mathbf{0})$$

Then \mathcal{M} can receive input from node e at its interface along the channel c ; this is passed on to the internal node i using channel d , where it is transformed in some way, described by the function f^1 , and then forwarded to node m , where 80% of the time it is broadcast to the external nodes o_1 and o_2 . The remainder of the time the message is lost.

The network \mathcal{N} has the same interface as \mathcal{M} , but has an extra internal node k connected to o_2 , and m is only connected to one interface node o_1 and the internal node k . The nodes i and n have the same code running as in \mathcal{M} , while nodes m and k will run the code

$$A_k \Leftarrow d?(x) . (d!\langle x \rangle . \mathbf{0}_{0.9} \oplus \mathbf{0})$$

Intuitively, the behaviour of \mathcal{N} is more complex than that of \mathcal{M} ; indeed, there is the possibility for a computation of \mathcal{N} to deliver a value only to one between the external nodes o_1 and o_2 , while this is not possible in \mathcal{M} . However, 81% of the times this message will be delivered to both these nodes, and thus it is more reliable than \mathcal{M} . Suppose now that we change the code at the intermediate node m in \mathcal{M} ,

$$\mathcal{M}_1 = \Gamma_M \triangleright \dots \mid m[[B_m]] \quad \text{where } B_m \Leftarrow d?(x) . (\tau . (d!\langle x \rangle . \mathbf{0}_{0.5} \oplus \mathbf{0}) + \tau . d!\langle x \rangle . \mathbf{0})$$

In \mathcal{M}_1 the behaviour at the node m is non-deterministic; it may act like a perfect forwarder, or one which is only 50% reliable. Optimistically it could be more reliable than \mathcal{M} , or pessimistically it could be less reliable than the latter. Further, there is no possibility for the network \mathcal{M}_1 to forward the message to only one of the external nodes o_1 , o_2 , so that its behaviour is somewhat less complex than that of \mathcal{N} .

As a further variation let \mathcal{M}_2 be the result of replacing the code at m with

$$C_m \Leftarrow d?(x) . D \\ D \Leftarrow \tau . (d!\langle x \rangle . \mathbf{0}_{0.5} \oplus \tau . D)$$

¹For example, if we assume the set of closed values to be \mathbb{Z} , f could be the mapping $f : z \mapsto z^2$.

<div style="text-align: center;">(s-SND)</div> $\frac{}{c!\langle e \rangle . p \xrightarrow{c!\llbracket e \rrbracket} \mathbb{P}(p)}$ <div style="text-align: center;">(s-τ)</div> $\frac{}{\tau . p \xrightarrow{\tau} \mathbb{P}(p)}$ <div style="text-align: center;">(s-SUMR)</div> $\frac{t \xrightarrow{\alpha} \Delta}{s + t \xrightarrow{\alpha} \Delta}$ <div style="text-align: center;">(s-ELSE)</div> $\frac{t \xrightarrow{\alpha} \Delta}{\text{if } b \text{ then } s \text{ else } t \xrightarrow{\alpha} \Delta} \llbracket b \rrbracket = \text{false}$	<div style="text-align: center;">(s-RCV)</div> $\frac{}{c?(x) . p \xrightarrow{c?v} \mathbb{P}(p\{v/x\})}$ <div style="text-align: center;">(s-SUML)</div> $\frac{s \xrightarrow{\alpha} \Delta}{s + t \xrightarrow{\alpha} \Delta}$ <div style="text-align: center;">(s-THEN)</div> $\frac{s \xrightarrow{\alpha} \Delta}{\text{if } b \text{ then } s \text{ else } t \xrightarrow{\alpha} \Delta} \llbracket b \rrbracket = \text{true}$ <div style="text-align: center;">(s-CALL)</div> $\frac{s\{\tilde{e}/\tilde{x}\} \xrightarrow{\mu} \Delta}{A(\tilde{e}) \xrightarrow{\mu} \Delta} A(\tilde{x}) \leftarrow s$
---	---

Figure 3: Pre-semantics of states

Here the behaviour is once more deterministic, with the probability that the message will be eventually transmitted successfully through node k approaching 1 in the limit. Thus, this network is as reliable as \mathcal{M}_1 , when the latter is viewed optimistically. \square

3.2. Intensional Semantics. We now turn our attention on the operational semantics of networks. Following [7], processes will be interpreted as probability distributions of states; such an interpretation is encoded by the function $\mathbb{P}(\cdot)$ defined below:

$$\begin{aligned} \mathbb{P}(s) &= \bar{s} \\ \mathbb{P}(p_1 \oplus p_2) &= p \cdot \mathbb{P}(p_1) + (1 - p) \cdot \mathbb{P}(p_2). \end{aligned}$$

Sometimes we will need to consider the probability distribution associated to system terms; this is done by letting

$$\begin{aligned} \mathbb{P}(\mathbf{0}) &= \bar{\mathbf{0}} \\ \mathbb{P}(n\llbracket p \rrbracket) &= n\llbracket \mathbb{P}(p) \rrbracket \\ \mathbb{P}(M_1 \mid M_2) &= \mathbb{P}(M_1) \mid \mathbb{P}(M_2) \end{aligned}$$

where $n\llbracket \mathbb{P}(p) \rrbracket$ represents a distribution over \mathbf{sSys} , obtained by a direct application of Equation (2.1) to the function $n\llbracket \cdot \rrbracket$ which maps states into system terms. Similarly, $\mathbb{P}(M_1 \mid M_2)$ is obtained by applying Equation (2.2) to the function $(\cdot \mid \cdot) : \mathbf{sSys} \times \mathbf{sSys} \rightarrow \mathbf{sSys}$.

The intensional semantics of networks is defined incrementally. We first define a pre-semantics for states, which is then used for giving the judgements of (state based) networks.

The pre-semantics for states takes the form

$$s \xrightarrow{\alpha} \Delta$$

$\frac{\text{(B-BROAD)}}{s \xrightarrow{c!v} \Delta} \quad \Gamma \triangleright n[[s]] \xrightarrow{n.c!v} n[[\Delta]]$	$\frac{\text{(B-REC)}}{s \xrightarrow{c?v} \Delta} \quad \Gamma \triangleright n[[s]] \xrightarrow{m.c?v} n[[\Delta]] \quad \Gamma \vdash m \rightarrow n$
$\frac{\text{(B-DEAF)}}{s \not\xrightarrow{c?v}} \quad \Gamma \triangleright n[[s]] \xrightarrow{m.c?v} \overline{n[[s]]}$	$\frac{\text{(B-DISC)}}{\Gamma \triangleright n[[s]] \xrightarrow{m.c?v} \overline{n[[s]]}} \quad \Gamma \vdash m \rightarrow n$
$\frac{\text{(B-0)}}{\mathbf{0} \xrightarrow{m.c?v} \overline{\mathbf{0}}}$	
$\frac{\text{(B-}\tau\text{)}}{s \xrightarrow{\tau} p} \quad \mathbb{P}(p) = \Delta \quad \Gamma \triangleright n[[s]] \xrightarrow{n.\tau} n[[\Delta]]$	$\frac{\text{(B-}\tau\text{.PROP)}}{\Gamma \triangleright M \xrightarrow{n.\tau} \Delta} \quad \Gamma \triangleright M N \xrightarrow{n.\tau} \Delta \overline{N}$
$\frac{\text{(B-PROP)}}{\Gamma \triangleright M \xrightarrow{m.c?v} \Delta, \Gamma \triangleright N \xrightarrow{m.c?v} \Theta} \quad \Gamma \triangleright M N \xrightarrow{m.c?v} \Delta \Theta$	$\frac{\text{(B-SYNC)}}{\Gamma \triangleright M \xrightarrow{m.c!v} \Delta, \Gamma \triangleright N \xrightarrow{m.c?v} \Theta} \quad \Gamma \triangleright M N \xrightarrow{m.c!v} \Delta \Theta$

Figure 4: Intensional semantics of networks

where s is a closed state, that is containing no free occurrences of a variable, Δ is a distribution of states and μ can take one of the forms $c!v$, $c?v$ or τ . The deductive rules for inferring these judgements are given in Figure 3 and should be self-explanatory. It assumes some mechanism for evaluating closed data-expressions e to values $[[e]]$. Note that we assume that definitions have the form $A(\tilde{x}) \Leftarrow s$, where s is a state; this is because actions for definitions are inherited by the state that is associated to them, and judgements are not defined for (probabilistic) processes. Also, note that we have a special state ω for which no rule has been defined. The role of this construct will become clear in Section 4.

Judgements in the intensional semantics of networks take the form

$$\Gamma \triangleright M \xrightarrow{\mu} \Delta$$

where Γ is a network connectivity, M is a system from sSys , and Δ is a distribution over sSys ; intuitively this means that relative to the connectivity Γ the system M can perform the action μ , and with probability $\Delta(N)$ be transformed into the system N , for every $N \in [\Delta]$. The action labels can take the form

- (i) receive, $n.c?v$, meaning that the value v is detected on channel c by all nodes in $\text{nodes}(M)$ which are reachable from n in Γ ,
- (ii) broadcast, $n.c!v$: meaning the node n (occurring in $\text{nodes}(M)$, and therefore in Γ) broadcasts the value v on channel c to all nodes directly connected to n in Γ
- (iii) internal activity, $n.\tau$, meaning some internal activity performed by node n .

The rules for inferring judgements are given in Figure 4. Here we have omitted the symmetric counterparts of rules (B-SYNC) and (B-TAUPROP). Rule (B-BROAD) models the capability for a node to broadcast a value v through channel c , assuming the code running there is capable of broadcasting along c . Here the distribution Δ is in turn obtained from the residual of the state s after the broadcast action.

Example 3.2. Consider the simple network $\Gamma \triangleright n[[s]]$ where Γ is an arbitrary connectivity graph and the code s has the form $c!(v).(s_1 \text{ } 1/4 \oplus s_2)$.

The pre-semantics of states determines that $s \xrightarrow{c!v} \mathbb{P}(s_1 \text{ } 1/4 \oplus s_2) = 1/4 \cdot \overline{s_1} + 3/4 \cdot \overline{s_2}$, using Rule (B-SND) Thus according to the rule (B-BROAD) we have the judgement

$$\Gamma \triangleright n[[s]] \xrightarrow{n.c!v} \frac{1}{4} \cdot n[[s_1]] + \frac{3}{4} \cdot n[[s_2]]$$

□

Rules (B-REC), (B-DEAF) and (B-DISC) express how a node reacts when a message is broadcast; the first essentially models the capability of a node which is listening to a channel c , and which appears in the sender's range of transmission, to receive the message correctly. The other two rules model situations in which a node is not listening to the channel used to broadcast a message, or it is not in the range of transmission of the sender; In both cases this node cannot detect the transmission at all.

The rules (B- τ) and (B- τ .PROP) model internal activities performed by some node of a system term; the latter (together with its symmetric counterpart) expresses the inability for a node which performs an internal activity to affect other nodes in a system term. Here again, $\Delta|\Theta$ is a distribution over sSys , this time obtained by instantiating Equation (2.2) to the function $(\cdot | \cdot) : \text{sSys} \times \text{sSys} \rightarrow \text{sSys}$.

Finally, rules (B-SYNC) and (B-PROP) describe how communication between nodes of a network is handled; here the result of a synchronisation between an output and an input is again an output, thus modelling broadcast communication [28].

3.3. Properties of the Calculus. We conclude this section by summarising the main properties enjoyed by the intensional semantics of our calculus.

Here (and in the rest of the paper) it will be convenient to identify networks and distributions of networks up to a structural congruence relation \equiv . This is first defined for states as the smallest equivalence relation which is a commutative monoid with respect to $+$ and $\mathbf{0}$, and which satisfies the equations if b then s else $t \equiv s$ if $[[b]] = \text{true}$, if b then s else $t \equiv t$ if $[[b]] = \text{false}$ and $A(\tilde{z}) \equiv s\{\tilde{z}/\tilde{x}\}$ if $A(\tilde{x}) \leftarrow s$. For system terms, we let \equiv be the smallest equivalence relation which is a commutative monoid with respect to $(\cdot | \cdot)$ and $\mathbf{0}$, and which satisfies the equation $s \equiv t$ implies $m[[s]] \equiv m[[t]]$ for any node m . Finally, we let $(\Gamma_M \triangleright M) \equiv (\Gamma_N \triangleright N)$ iff $\Gamma_1 = \Gamma_2$ and $M \equiv N$. Structural congruence is also defined for distributions of networks via the lifted relation $\overline{\equiv}^e$. With an abuse of notation, the latter is still denoted as \equiv .

The properties that we prove in this section give an explicit form to the structure of a network $(\Gamma \triangleright M)$ and a distribution Δ in the case that an action $(\Gamma \triangleright M) \xrightarrow{\mu} \Delta$ can be inferred in the intensional semantics presented in Section 3.2.

Proposition 3.3 (Tau-actions). *Let $\Gamma \triangleright M$ be a network; then $\Gamma \triangleright M \xrightarrow{m.\tau} \Delta$ if and only if*

- (i) $M \equiv m[(\tau.p) + s] | N$,
- (ii) $\Delta \equiv \mathbb{P}(m[[p]])|\overline{N}$

Outline of the Proof. We first need to prove a similar statement for states. Let t be a state; then $t \xrightarrow{\tau} \Theta$ if and only if $t \equiv \tau.p + s$ for some p, s such that $\mathbb{P}(p) = \Theta$. The two implications of this statement are proved separately.

To prove Proposition 3.3 suppose first that $\Gamma \triangleright M \xrightarrow{m.\tau} \Delta$ for some distribution Δ . We show that $M \equiv m[\tau.p + s] \mid N$, $\Delta \equiv \mathbb{P}(m[p]) \mid \bar{N}$ by structural induction on the proof of the derivation above.

If the last rule applied is (B- τ), then $M = m[t]$ for some t such that $t \xrightarrow{\tau} \Theta$, $\Delta = m[\Theta]$.

Since $t \xrightarrow{\tau} \Theta$ then $t \equiv \tau.p + s$ for some process p such that $\mathbb{P}(p) = \Theta$, which also gives $\Delta = \mathbb{P}(m[p])$. By definition of structural congruence $M \equiv m[\tau.p + s] \equiv m[\tau.p + s] \mid \mathbf{0}$. Further, $\Delta \equiv \mathbb{P}(m[p]) \mid \bar{\mathbf{0}}$, and there is nothing left to prove.

If the last rule applied is (B- τ .PROP), then $M \equiv N \mid L$ for some N, L such that $N \xrightarrow{m.\tau} \Delta_N$; further, $\Delta = \Delta_N \mid \bar{L}$. By inductive hypothesis we have that $N \equiv m[\tau.p + s] \mid N'$ and $\Delta_N \equiv \mathbb{P}(m[p]) \mid \bar{N}'$. By performing some simple calculations we find that $M \equiv N \mid L \equiv m[\tau.p + s] \mid (N' \mid L)$ and $\Delta \equiv \Delta_N \mid \bar{L} \equiv \mathbb{P}(m[p]) \mid (\bar{N}' \mid \bar{L}) \equiv \mathbb{P}(m[p]) \mid (\bar{N}' \mid \bar{L})$.

Conversely, suppose that $M \equiv m[\tau.p + s] \mid N$; in this case it suffices to perform a rule induction on the proof of the equivalence above to show that $\Gamma \triangleright M \xrightarrow{m.\tau} \Delta$, where $\Delta \equiv \mathbb{P}(m[p]) \mid \bar{N}$. \square

Proposition 3.4 (Input). *For any network $\Gamma \triangleright M$ we have that $\Gamma \triangleright M \xrightarrow{m.c?v} \Delta$ iff*

- (i) $m \notin \text{nodes}(M)$,
- (ii) $M \equiv \prod_{i \in I} n_i \llbracket (c?(x).p_i) + s_i \rrbracket \mid \prod_{j \in J} n_j \llbracket s_j \rrbracket$,
- (iii) for any $i \in I$, $\Gamma \vdash m \rightarrow n_i$,
- (iv) for any $j \in J$, either $\Gamma \vdash m \rightarrow n_j$ or $s_j \xrightarrow{c?v}$,
- (v) $\Delta \equiv \mathbb{P}(\prod_{i \in I} n_i \llbracket p_i\{v/x\} \rrbracket) \mid \prod_{j \in J} n_j \llbracket s_j \rrbracket$. \square

Proposition 3.5 (Broadcast). *Let $\Gamma \triangleright M$ be a network; then $\Gamma \triangleright M \xrightarrow{m.c!v} \Delta$ for some Δ iff*

- (i) $M \equiv m \llbracket (c!\langle e \rangle.p + s) \rrbracket \mid N$, where $\llbracket e \rrbracket = v$,
- (ii) $\Gamma \triangleright N \xrightarrow{m.c?v} \Theta$,
- (iii) $\Delta \equiv \mathbb{P}(m[p]) \mid \Theta$. \square

An immediate consequence of the results above is that actions in the intensional semantics are preserved by structurally congruent networks.

Corollary 3.6. *Let M, N be two networks such that $M \equiv N$. If $M \xrightarrow{\mu} \Delta$ then $N \xrightarrow{\mu} \Theta$ for some Θ such that $\Delta \equiv \Theta$. \square*

Another trivial consequence that follows from the results above is that the intensional semantics does not change the structure of a network.

Definition 3.7 (Stable distributions). A (node)-stable sub-distribution $\Delta \in \mathcal{D}_{\text{sub}}(\text{sSys})$ is one for which whenever $M, N \in [\Delta]$ it follows that $\text{nodes}(M) = \text{nodes}(N)$. A distribution over networks is said to be (node)-stable if it has the form $\Gamma \triangleright \Delta$, and Δ is a stable sub-distribution in $\mathcal{D}_{\text{sub}}(\text{sSys})$.

Corollary 3.8. *Whenever $\Gamma \triangleright M \xrightarrow{\mu} \Delta$ then Δ is node-stable; further, for any $N \in [\Delta]$ we have that $\text{nodes}(M) = \text{nodes}(N)$. \square*

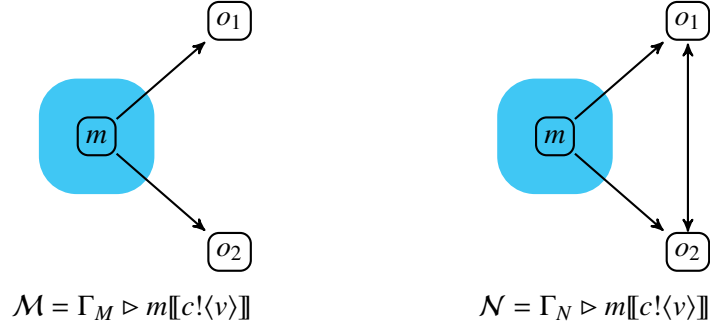


Figure 5: A well-formed and an ill-formed network

4. COMPOSITIONAL REASONING FOR NETWORKS

The aim of this Section is to develop preorders of the form

$$\mathcal{M} \sqsubseteq_{behav} \mathcal{N} \quad (4.1)$$

Intuitively this means that the network \mathcal{M} can be replaced by \mathcal{N} , as a part of a larger overall network, without any loss of behaviour. The intention is that the internal structure of the networks \mathcal{M} , \mathcal{N} should play no role in this comparison; the names used to identify their internal stations and the their communication topology should not be important. Intuitively the only behaviour to be taken into account in this extensional comparison is the reception of values at the their interface, the values subsequently broadcast at the interface.

To formalise this concept we need to say how networks are composed to form larger networks. In Section 4.1 we propose a specific composition operator, \sharp for this purpose, and briefly discuss its properties. We then use this operator in Section 4.2 to say how network behaviour is determined. In Section 4.3, we give the formal definition of the behavioural preorders, in a relatively standard manner following [27]; this section also treats some examples. The nature of these preorders depends on our particular choice of composition operator \sharp . In Section 4.4 we return to this point and offer a justification for our choice; this section may be safely ignored by the reader who is uninterested in this subtlety.

However first let us reconsider the informal requirements of the proposed behavioural preorder (4.1) above. We have already mentioned that it should not depend on the internal structure of \mathcal{N} , \mathcal{M} . But equally well \mathcal{N} , \mathcal{M} should not be able to make any assumptions about the topology of their external environment.

Example 4.1. Consider the networks \mathcal{M} and \mathcal{N} depicted in Figure 5. At least intuitively the extensional behaviour of these two networks is the same: a broadcast of value v along channel c can be detected by the external nodes o_1 and o_2 . However \mathcal{N} makes an assumption about the external environment, namely that there is a connection between the external nodes o_1 and o_2 . This slight difference can be exploited to distinguish between them behaviourally. Suppose that we place the code $c?(x).c!\langle w \rangle$ at node o_1 and the code $c?(x).c?(y).\omega$ at node o_2 to test the behaviour of both networks. In practice, let $T = o_1[[c?(x).c!\langle w \rangle]] \mid o_2[[c?(x).c?(y).\omega]]$, and consider the networks $\mathcal{M}' = \Gamma_M \triangleright m[[c!\langle v \rangle]] \mid T$ and $\mathcal{N}' = \Gamma_N \triangleright m[[c!\langle v \rangle]] \mid T$. In the first network the node o_2 can detect both the broadcast fired by node m and node o_1 , leading to a state in which the special action ω is enabled. However, the same is not possible in \mathcal{N}' , since there is no connection between the external nodes o_1 and o_2 . That is, node o_2 can only detect the broadcast fired by node m , ending in a state in

which the special action ω remains guarded by an input. As we will see later, the clause ω plays a crucial role in distinguishing networks. \square

The problem in Example 4.1 is caused by the presence of a connection between the two external nodes in the network \mathcal{N} of Figure 5; intuitively this represents an assumption of \mathcal{N} about its external environment. To avoid this problem, we focus on a specific class of networks in which connections between external nodes are not allowed. Also, we require external nodes to have at least a connection with some internal node.

Definition 4.2. [Well-Formed Networks] A network \mathcal{M} is well-formed iff

- (i) whenever $\mathcal{M} \vdash m \rightleftharpoons n$ then either $m \in \text{nodes}(\mathcal{M})$ or $n \in \text{nodes}(\mathcal{M})$,
- (ii) whenever $\mathcal{M} \vdash m$ and $\mathcal{M} \vdash m \rightleftharpoons n$ for no $n \in (\mathcal{M})_V$, then $m \in \text{nodes}(\mathcal{M})$. \square

Henceforth we will only focus on well-formed networks, unless stated otherwise. We denote the set of well-formed networks as **Nets**.

Finally let us provide some definitions which will be useful in the sequel. Let \mathcal{M} be a (well-formed) network. We say that a node $i \in \text{Intf}(\mathcal{M})$ is an *input node* if $\mathcal{M} \vdash m \leftarrow i$ for some m^2 ; conversely, if a node $o \in \text{Intf}(\mathcal{M})$ is such that $\Gamma \vdash m \rightarrow o$ we say that o is an *output node*. If we let $\text{In}(\mathcal{M}) = \{i \mid i \text{ is an input node in } \mathcal{M}\}$ and $\text{Out}(\mathcal{M}) = \{o \mid o \text{ is an output node in } \mathcal{M}\}$ it is easy to check that $\text{Intf}(\mathcal{M}) = \text{In}(\mathcal{M}) \cup \text{Out}(\mathcal{M})$.

4.1. Composing Networks. One use of composition operators is to enable compositional reasoning. For example the task of establishing

$$\mathcal{N}_1 \sqsubseteq_{\text{behav}} \mathcal{N}_2 \tag{4.2}$$

can be simplified if we can discover a common component, that is some \mathcal{N} such that $\mathcal{N}_1 = \mathcal{M}_1 \parallel \mathcal{N}$ and $\mathcal{N}_2 = \mathcal{M}_2 \parallel \mathcal{N}$ for some composition operator \parallel ; then (4.2) can be reduced to establishing

$$\mathcal{M}_1 \sqsubseteq_{\text{behav}} \mathcal{M}_2,$$

assuming that the behavioural preorder in question, $\sqsubseteq_{\text{behav}}$, is preserved by the composition operator \parallel .

However another use of a composition operator is in the definition of the behavioural preorder $\sqsubseteq_{\text{behav}}$ itself. Intuitively we can define

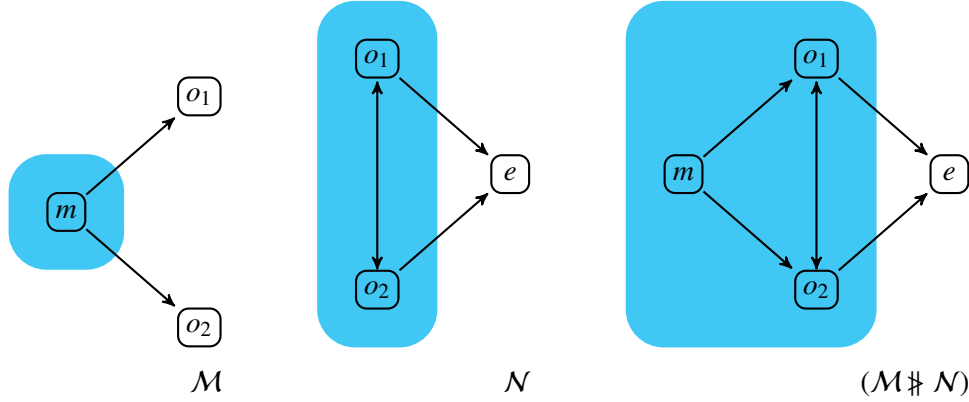
$$\mathcal{N}_1 \sqsubseteq_{\text{behav}} \mathcal{N}_2 \tag{4.3}$$

to be true if for every component \mathcal{T} which can be composed with both \mathcal{N}_1 and \mathcal{N}_2 , the external observable behaviour of the composite networks $\mathcal{N}_1 \parallel \mathcal{T}$ and $\mathcal{N}_2 \parallel \mathcal{T}$ are related in some appropriate way. \mathcal{T} is a *testing network* which is probing \mathcal{N}_1 and \mathcal{N}_2 for behavioural differences, along the lines used informally in Example 4.1. Intuitively this should be *black-box testing*, in which the tester, namely \mathcal{T} , should have no access to the internal stations of the networks being tested, namely \mathcal{N}_1 and \mathcal{N}_2 . All it can do is place code at their external interfaces, to transmit values and examine the subsequent effects, as seen again at the interfaces.

Definition 4.3. [Composing networks] For any two networks $\mathcal{M} = \Gamma_M \triangleright M$ and $\mathcal{P} = \Gamma_P \triangleright P$ let $\mathcal{M} \sharp \mathcal{P}$ be given by:

$$(\Gamma_M \triangleright M) \sharp (\Gamma_P \triangleright P) = \begin{cases} (\Gamma_M \cup \Gamma_P) \triangleright (M \mid P), & \text{if } \text{nodes}(\mathcal{M}) \cap (\mathcal{P})_V = \emptyset \\ \text{undefined}, & \text{otherwise} \end{cases}$$

²Note that by well-formedness this implies $m \in \text{nodes}(\mathcal{M})$.


 Figure 6: Network composition via $\#$

The composed connectivity graph $\Gamma_M \cup \Gamma_P$ is defined by letting $(\Gamma_M \cup \Gamma_P)_V = (\Gamma_M)_V \cup (\Gamma_P)_V$ and $(\Gamma_M \cup \Gamma_P)_E = (\Gamma_M)_E \cup (\Gamma_P)_E$. \square

The intuition here is that the composed network $\mathcal{M} \# \mathcal{P}$ is constructed by *extending* the network under test, \mathcal{M} , allowing code to be placed at its interface, and allowing completely fresh stations to be added. These fresh stations can be used by the tester to compute the results of probes made on \mathcal{M} .

Proposition 4.4. *Suppose $\mathcal{M}, \mathcal{N}, \mathcal{P} \in \text{Nets}$. Then*

- (1) $\mathcal{M} \# \mathcal{P} \in \text{Nets}$, whenever it is defined
- (2) $(\mathcal{M} \# \mathcal{N}) \# \mathcal{P} = \mathcal{M} \# (\mathcal{N} \# \mathcal{P})$, whenever both are defined

Proof. The two statements are proved separately. The proofs are given in a separate appendix, Appendix A; see pages 58 and 59. \square

There is an inherent asymmetry in the definition of our composition operator; in $\mathcal{M} \# \mathcal{P}$ we allow \mathcal{P} to place code at the interface nodes of \mathcal{M} but not the converse. As a result the operator is not in general symmetric, as can be seen from Example 4.5.

Example 4.5. [$\#$ is asymmetric] Let \mathcal{M}, \mathcal{N} be the networks depicted in Figure 6. Here the network $\mathcal{M} \# \mathcal{N}$ is well-defined and depicted on the right of Figure 6. Intuitively, this is the network obtained by extending \mathcal{M} with the information provided by \mathcal{N} ; these include the code running at nodes o_1, o_2 , a connection between such nodes, and a connection from o_1, o_2 , respectively, to a fresh node e , whose running code is left unspecified. The network $\mathcal{M} \# \mathcal{N}$ is well-defined because none of the connections specified in \mathcal{N} involve the node $m \in \text{nodes}(\mathcal{M})$; that is, when extending \mathcal{M} with \mathcal{N} it is ensured that the latter can interact with the node m only via the nodes o_1, o_2 .

On the other hand, the composition $\mathcal{N} \# \mathcal{M}$ is not defined. Intuitively, in \mathcal{N} nodes o_1 and o_2 can interact with the external environment only via the external node e . This is in contrast with the definition of \mathcal{M} , where node m can be used to broadcast messages to such nodes. \square

One might wonder if a symmetric composition operator could be used in place of our $\#$; this point is discussed at length in Section 4.4. Nevertheless $\#$ is a natural operator, and the next result shows that it can be used to construct all non-degenerate networks starting from single nodes. Let \mathbb{G} be the collection of networks which contain exactly one occupied node, that is $\mathbb{G} = \{(\Gamma \triangleright G) \mid G \equiv m[s]\}$ for some s, m .

Proposition 4.6. Suppose $\mathcal{M} \in \mathbf{Nets}$ is a network such that $\text{nodes}(\mathcal{M})$ is not empty. Then $\mathcal{M} \equiv \mathcal{N} \sharp \mathcal{G}$ for some $\mathcal{N} \in \mathbf{Nets}$ and $\mathcal{G} \in \mathbb{G}$.

Proof. See Appendix A, Page 59. □

We use the term *generating networks* to refer to elements of \mathbb{G} . The import of Proposition 4.6 is that all non-trivial well-formed networks can be constructed from basic generating networks, using \sharp .

4.2. Testing Structures. The introduction of the composition operator \sharp allows the development of a behavioural theory based on a probabilistic generalisation of the *De Nicola* and *Hennessy* testing preorders [27]. In order to develop such a framework, we will exploit the mathematical tools introduced in Section 2; our aim is to be able to relate networks with different connection topologies.

In our framework, as has already been indicated, testing can be summarised as follows: the network to be tested is composed with another one, usually called a *testing network*. The composition of these two networks is then isolated from the external environment, in the sense that no external agent (in our case nodes in the interface of the composed network) can interfere with its behaviour; we will shortly present how such a task can be accomplished. The composition of the two networks isolated from the external environment takes the name of *experiment*.

Once these two operations (composition with a test and isolation from the external environment) have been performed, the behaviour of the resulting experiment is analysed to check whether there exists a computation that yields a state which is successful. This task can be accomplished by relying on testing structures, which will be presented shortly.

At an informal level, successful states in our language coincide with those associated with networks where at least the code running at one node has the special *action* ω enabled. Since networks have probabilistic behaviour, each computation will be associated with the probability of reaching a successful state; thus, every experiment will be associated with a set of success probabilities, by quantifying over all its computation.

Let us now look at how the procedure explained above can be formalised; the topic of composing networks has already been addressed in detail in Section 4.1, in which we defined the operator \sharp and proved basic properties for it. To model experiments and their behaviour, we rely on the following mathematical structure.

Definition 4.7. A *Testing Structure* (TS) is a triple $\langle S, \rightarrow, \omega \rangle$ where

- (i) S is a set of states,
- (ii) the relation \rightarrow is a subset of $S \times \mathcal{D}(S)$,
- (iii) ω is a success predicate over S , that is $\omega : S \rightarrow \{\text{true}, \text{false}\}$. □

Testing structures can be seen as (degenerate) pLTSs where the only possible action corresponds to the internal activity τ , and the transition $\xrightarrow{\tau}$ is defined to coincide with the reduction relation \rightarrow . Conversely every pLTS automatically determines a testing structure, by concentrating on the relation $\xrightarrow{\tau}$.

Our goal is to turn a network into a testing structure. This amounts to defining, for networks, a reduction relation and the success predicate ω . As we have mentioned in the beginning of this section, when converting a network into a testing structure, we want to make it isolated from the external environment.

When considering simpler process languages, like *CCS* or *CSP* (and, more generally, their probabilistic counterparts), processes are converted into testing structures by identifying the reduction relation with the internal activity τ ; that is, processes are not allowed to synchronise with some external agent via a visible action.

Networks, however, are more complicated objects; here the nature of broadcast is non-blocking, meaning that a broadcast can be fired by a node in a network without requiring any synchronisation with a (possibly external) node. Thus we expect broadcast actions to induce reductions when converting a network into a testing structure.

On the other hand, input actions always originate from non-internal nodes. Hence they can be seen as external activities which can influence the behaviour of a network. Therefore, input actions should not be included in the definition of the reduction relation for networks.

Finally, the success predicate ω is defined to be true for exactly those networks in which the success clause ω is enabled in at least one node.

Example 4.8. The main example of a TS is given by

$$\langle \text{Nets}, \rightarrow, \omega \rangle$$

where

- (i) $(\Gamma \triangleright M) \rightarrow (\Gamma \triangleright \Delta)$ whenever
 - (a) $\Gamma \triangleright M \xrightarrow{m.\tau} \Delta$ for some $m \in \text{nodes}(M)$
 - (b) or, $\Gamma \triangleright M \xrightarrow{m.c!v} \Delta$ for some value v , node name m and channel c
- (ii)

$$\omega(M) = \begin{cases} \text{true}, & \text{if } M \equiv M' \mid n[\omega + s], \text{ for some } s, n, M' \\ \text{false}, & \text{otherwise} \end{cases}$$

If $\omega(M) = \text{true}$ for some system term M , we say that a network $\Gamma \triangleright M$, where Γ is an arbitrary connectivity graph, is ω -successful, or simply successful. Note that when recording an ω -success we do not take into account the node involved. \square

As TSs can be seen as pLTSs, we can use in an arbitrary TS the various constructions introduced in Section 2. Thus the reduction relation \rightarrow can be lifted to $\mathcal{D}_{\text{sub}}(\text{sSys}) \times \mathcal{D}_{\text{sub}}(\text{sSys})$ and we can make use of the concepts of hyper-derivatives and extreme-derivatives, introduced in Section 2, to model fragments of executions and maximal executions of a testing structure, respectively. Hyper-derivations in testing structures are denoted with the symbol \Longrightarrow , while we use the symbol \Longrightarrow for extreme derivations.

Below we provide two simple examples that show how to reason about the behaviour of the testing structures presented in Example 4.8.

Example 4.9. Consider the testing structure associated with the network \mathcal{N} in the center of Figure 11, where the code q is given by the definition $q \Leftarrow q_{0.5} \oplus c!\langle v \rangle . \mathbf{0}$. We can show that, in the long run, this network will broadcast message v to the external location o by exhibiting a hyper derivation for it which terminates in the point distribution $\overline{\Gamma_N \triangleright k[[c!\langle v \rangle . \mathbf{0}]]}$. If we let \mathcal{N}_1 denote the configuration $\Gamma_N \triangleright k[[c!\langle v \rangle . \mathbf{0}]]$, we have the following hyper-derivation:

$$\begin{aligned} \overline{\mathcal{N}} &= \frac{1}{2} \cdot \overline{\mathcal{N}} + \frac{1}{2} \cdot \overline{\mathcal{N}_1} \\ \frac{1}{2} \cdot \overline{\mathcal{N}} &\rightarrow \frac{1}{2^2} \cdot \overline{\mathcal{N}} + \frac{1}{2^2} \cdot \overline{\mathcal{N}_1} \\ &\vdots \\ \frac{1}{2^n} \cdot \overline{\mathcal{N}} &\rightarrow \frac{1}{2^{n+1}} \cdot \overline{\mathcal{N}} + \frac{1}{2^{n+1}} \cdot \overline{\mathcal{N}_1} \\ &\vdots \end{aligned}$$

Let $\Delta' = \sum_{n=1}^{\infty} \frac{1}{2^n} \cdot \overline{\mathcal{N}_1}$. It is straightforward to check that $\Delta' = \overline{\mathcal{N}_1}$ and therefore we have the hyper-derivation $\mathcal{N} \Longrightarrow \overline{\mathcal{N}_1}$. \square

An arbitrary network \mathcal{N} can be tested by another (testing) network \mathcal{T} provided $\mathcal{N} \sharp \mathcal{T}$ is well-defined. Executions of the resulting testing structure will then be checked to establish whether the network \mathcal{M} satisfies a property the test was designed for; in such a case, the testing component of an experiment will reach a ω -successful state.

Executions, or maximal computations, correspond to extreme derivatives in the testing structure associated with $(\mathcal{N} \sharp \mathcal{T})$, as defined in Section 2. Since the framework is probabilistic, each execution (that is extreme derivative) will be associated with a probability value, representing the probability that it will lead to an ω -successful state. Since the framework is also nondeterministic the possible results of this test application is given by a non-empty set of probability values.

Definition 4.10. [Tabulating results] The *value* of a sub-distribution in a TS is given by the function $\mathcal{V} : \mathcal{D}_{sub}(S) \rightarrow [0, 1]$, defined by $\mathcal{V}(\Delta) = \sum \{ \Delta(s) \mid \omega(s) = \text{true} \}$. Then the set of possible results from a sub-distribution Δ is defined by $O(\Delta) = \{ \mathcal{V}(\Delta') \mid \Delta \Longrightarrow \Delta' \}$. \square

Example 4.11. Let \mathcal{N} be the network from Example 4.9 and consider the testing network \mathcal{T} given in Figure 11, where the code is determined by $t \leftarrow c?(x).\omega.\mathbf{0}$. It is easy to check that $\mathcal{N} \sharp \mathcal{T}$ is well-defined and is equal to $\Gamma \triangleright k[[q]] \mid o[[t]]$, where Γ is the connectivity graph containing the three nodes k, o, l and having the connections from $\Gamma \vdash k \rightarrow o$ and $\Gamma \vdash o \leftarrow t$. So consider the testing structure associated with it; recall that we have the definition $q \leftarrow q_{0.5} \oplus c!\langle v \rangle.\mathbf{0}$. For convenience let $\mathcal{N}_1 = \Gamma_N \triangleright k[[c!\langle v \rangle.\mathbf{0}]]$ as in the previous example, $\mathcal{N}_2 = \Gamma_N \triangleright k[[\mathbf{0}]]$ and $\mathcal{T}_\omega = \Gamma_T \triangleright o[[\omega.\mathbf{0}]]$. Then we have the following hyper-derivation for $\mathcal{N} \sharp \mathcal{T}$:

$$\begin{array}{lcl} \overline{\mathcal{N} \sharp \mathcal{T}} & \rightarrow & (\frac{1}{2} \cdot \overline{\mathcal{N} \sharp \mathcal{T}} + \frac{1}{2} \cdot \overline{\mathcal{N}_1 \sharp \mathcal{T}}) \quad + \quad \varepsilon \\ \frac{1}{2} \cdot \overline{\mathcal{N} \sharp \mathcal{T}} + \frac{1}{2} \cdot \overline{\mathcal{N}_1 \sharp \mathcal{T}} & \rightarrow & (\frac{1}{2^2} \cdot \overline{\mathcal{N} \sharp \mathcal{T}} + \frac{1}{2^2} \cdot \overline{\mathcal{N}_1 \sharp \mathcal{T}}) \quad + \quad \frac{1}{2} \cdot \overline{\mathcal{N}_2 \sharp \mathcal{T}_\omega} \\ \vdots & & \vdots \\ \frac{1}{2^n} \cdot \overline{\mathcal{N} \sharp \mathcal{T}} + \frac{1}{2^n} \cdot \overline{\mathcal{N}_1 \sharp \mathcal{T}} & \rightarrow & (\frac{1}{2^{n+1}} \cdot \overline{\mathcal{N} \sharp \mathcal{T}} + \frac{1}{2^{n+1}} \cdot \overline{\mathcal{N}_1 \sharp \mathcal{T}}) \quad + \quad \frac{1}{2^n} \cdot \overline{\mathcal{N}_2 \sharp \mathcal{T}_\omega} \\ \vdots & & \vdots \end{array}$$

where we recall that ε denotes the empty sub-distribution, that is the one with $[\varepsilon] = \emptyset$. We have therefore the hyper-derivation

$$\overline{\mathcal{N} \sharp \mathcal{T}} \Longrightarrow \varepsilon + \sum_{n=1}^{\infty} \frac{1}{2^n} \overline{\mathcal{N}_2 \sharp \mathcal{T}_\omega} = \overline{\mathcal{N}_2 \sharp \mathcal{T}_\omega}.$$

Further, the above hyper-derivation satisfies the constraints required by \Longrightarrow , defined in Section 2, and therefore we have the extreme derivative $\overline{\mathcal{N} \sharp \mathcal{T}} \Longrightarrow \overline{\mathcal{N}_2 \sharp \mathcal{T}_\omega}$. Since $\mathcal{V}(\overline{\mathcal{N}_2 \sharp \mathcal{T}_\omega}) = 1$ we can therefore deduce that $1 \in O(\mathcal{N} \sharp \mathcal{T})$. \square

4.3. The behavioural preorders. We now combine the concepts of the previous two sections to obtain our behavioural preorders. We have seen how to associate a non-empty set of probabilities, tabulating the possible outcomes from applying the test \mathcal{T} to the network \mathcal{N} . As explained in [7] there are two natural ways to compare such sets, optimistically or pessimistically.

Definition 4.12. [Relating sets of outcomes] Let O_1, O_2 be two sets of values in $[0, 1]$.

- (i) The *Hoare's Preorder* is defined by letting $O_1 \sqsubseteq_H O_2$ whenever for any $p_1 \in O_1$ there exists $p_2 \in O_2$ such that $p_1 \leq p_2$.
- (ii) The *Smith's Preorder* is defined by letting $O_1 \sqsubseteq_S O_2$ if for any $p_2 \in O_2$ there exists $p_1 \in O_1$ such that $p_1 \leq p_2$. \square

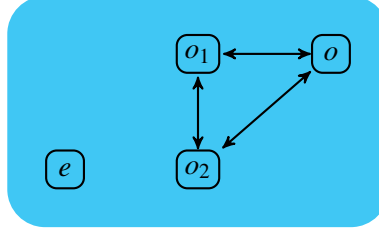


Figure 7: A test

Given two networks \mathcal{M}, \mathcal{N} we can relate their behaviour, when extended with a testing network \mathcal{T} , by comparing the success outcomes of $\mathcal{M} \sharp \mathcal{T}$ and $\mathcal{N} \sharp \mathcal{T}$ (provided both these networks are defined) via Definition 4.12. We can go further and consider what is the relationship between such sets of outcomes with respect to all possible tests \mathcal{T} which can be used to extend the networks \mathcal{M}, \mathcal{N} .

Definition 4.13. [Testing networks] For $\mathcal{M}_1, \mathcal{M}_2 \in \text{Nets}$ such that $\text{In}(\mathcal{M}_1) = \text{In}(\mathcal{M}_2)$, $\text{Out}(\mathcal{M}_1) = \text{Out}(\mathcal{M}_2)$, we write

- (1) $\mathcal{M}_1 \sqsubseteq_{\text{may}} \mathcal{M}_2$ iff for every (testing) network $\mathcal{T} \in \text{Nets}$ such that both $\mathcal{M}_1 \sharp \mathcal{T}$ and $\mathcal{M}_2 \sharp \mathcal{T}$ are defined, $\mathcal{O}(\mathcal{M}_1 \sharp \mathcal{T}) \sqsubseteq_H \mathcal{O}(\mathcal{M}_2 \sharp \mathcal{T})$.
- (2) $\mathcal{M}_1 \sqsubseteq_{\text{must}} \mathcal{M}_2$ iff for every (testing network) $\mathcal{T} \in \text{Nets}$ such that both $\mathcal{M}_1 \sharp \mathcal{T}$ and $\mathcal{M}_2 \sharp \mathcal{T}$ are defined, $\mathcal{O}(\mathcal{M}_1 \sharp \mathcal{T}) \sqsubseteq_S \mathcal{O}(\mathcal{M}_2 \sharp \mathcal{T})$

We use $\mathcal{M}_1 =_{\text{may}} \mathcal{M}_2$ as an abbreviation for $\mathcal{M}_1 \sqsubseteq_{\text{may}} \mathcal{M}_2$ and $\mathcal{M}_2 \sqsubseteq_{\text{may}} \mathcal{M}_1$. The relation $=_{\text{must}}$ is defined similarly. Finally, we say that $\mathcal{M}_1 \sqsubseteq \mathcal{M}_2$ iff both $\mathcal{M}_1 \sqsubseteq_{\text{may}} \mathcal{M}_2$ and $\mathcal{M}_1 \sqsubseteq_{\text{must}} \mathcal{M}_2$ hold, and $\mathcal{M}_1 \simeq \mathcal{M}_2$ iff $\mathcal{M}_1 \sqsubseteq \mathcal{M}_2$ and $\mathcal{M}_2 \sqsubseteq \mathcal{M}_1$. \square

Some explanation is necessary for the requirement on the interface of networks we have placed in Definition 4.13. This constraint establishes that two networks \mathcal{M} and \mathcal{N} are always distinguished if the sets of their input or output nodes differ. As we already mentioned, external nodes can be seen as terminals that can be accessed by the external environment to interact with the network. Roughly speaking, the constraint we have placed corresponds to the intuition that the external environment can distinguish two networks \mathcal{M} and \mathcal{N} by simply looking at the terminals that it can use to interact with these two networks.

Example 4.14. Consider the testing network

$$\mathcal{T} = \Gamma_T \triangleright e[[c!\langle 0 \rangle]] \mid o_1[[d?(x) \cdot c!\langle x \rangle]] \mid o_2[[d?(x) \cdot c!\langle x \rangle]] \mid o[[c?(x) \cdot c?(y) \cdot \omega]]$$

where the connectivity is described in Figure 7. This can be used to test the networks \mathcal{M}, \mathcal{N} from Example 3.1 in the testing structure of Example 4.8.

Intuitively the test sends the value 0 along the channel c at the node e , awaits for results along the channel d at the nodes o_1 and o_2 . These results are processed at node o , where success might be announced.

The combined network $(\mathcal{M} \sharp \mathcal{T})$ is deterministic in this TS, although probabilistic, and so has only one extreme derivative; $\mathcal{O}(\mathcal{M} \sharp \mathcal{T}) = \{0.8\}$.

A similar calculation shows that $\mathcal{O}(\mathcal{N} \sharp \mathcal{T}) = \{0.81\}$; it therefore follows that $\mathcal{N} \not\sqsubseteq_{\text{may}} \mathcal{M}$ and $\mathcal{M} \not\sqsubseteq_{\text{must}} \mathcal{N}$.

Consider now the networks $\mathcal{M}_1, \mathcal{M}_2$ from Example 3.1. Here $(\mathcal{M}_1 \sharp \mathcal{T})$ is both probabilistic and nondeterministic, and $\mathcal{O}(\mathcal{M}_1 \sharp \mathcal{T}) = \{p \mid 0.5 \leq p \leq 1\}$. Moreover we have $\mathcal{O}(\mathcal{M} \sharp \mathcal{T}) \sqsubseteq_H \mathcal{O}(\mathcal{M}_1 \sharp \mathcal{T})$.

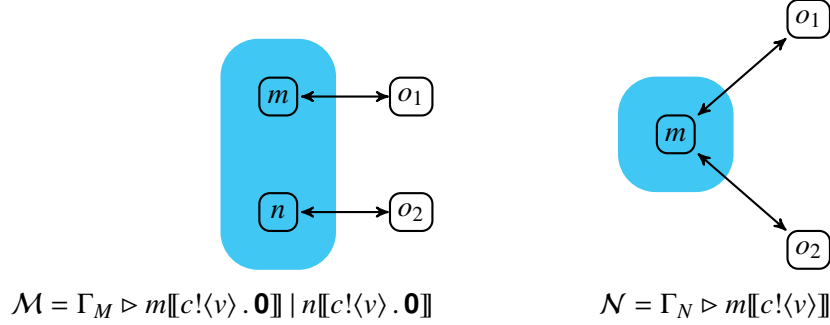


Figure 8: Broadcast vs Multicast

The combined network $(\mathcal{M}_2 \# \mathcal{T})$ is also deterministic, although it has *limiting behaviour*; $\mathcal{O}(\mathcal{M}_2 \# \mathcal{T}) = \{1\}$. Thus, in this case we have both $\mathcal{O}(\mathcal{M} \# \mathcal{T}) \sqsubseteq_H \mathcal{O}(\mathcal{M}_2 \# \mathcal{T})$ and $\mathcal{O}(\mathcal{M}_1 \# \mathcal{T}) \sqsubseteq_H \mathcal{O}(\mathcal{M}_2 \# \mathcal{T})$. Further, we have that $\mathcal{O}(\mathcal{M}_1 \# \mathcal{T}) \sqsubseteq_S \mathcal{O}(\mathcal{M}_2 \# \mathcal{T})$, but $\mathcal{O}(\mathcal{M}_2 \# \mathcal{T}) \not\sqsubseteq_S \mathcal{O}(\mathcal{M}_1 \# \mathcal{T})$. \square

Example 4.15. [Broadcast vs Multicast] Consider the networks \mathcal{M} and \mathcal{N} in Figure 8. Intuitively in \mathcal{N} the value v is (simultaneously) broadcast to both nodes o_1 and o_2 while in \mathcal{M} there is a multicast. More specifically o_1 receives v from mode m while in an independent broadcast o_2 receives it from n .

This difference in behaviour can be detected (when we compare the networks optimistically) by the testing network

$$\mathcal{T} = \Gamma_T \triangleright o_1[[c?(x).c!\langle w \rangle . \mathbf{0}]] \mid o_2[[c?(x).c?(y). \text{if } y = 0 \text{ then } \mathbf{0} \text{ else } \omega]]$$

assuming v is different than w ; here we assume Γ_T is the simple network which connects o_1 to o_2 . Both $\mathcal{M} \# \mathcal{T}$ and $\mathcal{N} \# \mathcal{T}$ are well-formed and note that they are both non-probabilistic.

Because \mathcal{N} simultaneously broadcasts to o_1 and o_2 the second value received by o_2 is always w and therefore the test never succeeds; $\mathcal{O}(\mathcal{N} \# \mathcal{T}) = \{0\}$. On the other-hand there is a possibility for the test succeeding when applied to \mathcal{M} , $1 \in \mathcal{O}(\mathcal{M} \# \mathcal{T})$. This is because in \mathcal{M} node m might first transmit v to o_1 after which n transmits w to o_2 ; now node n might transmit the value v to o_2 and assuming it is different than w we reach a success state. It follows that $\mathcal{M} \not\sqsubseteq_{\text{may}} \mathcal{N}$.

Note that we can slightly modify the test \mathcal{T} to show that we also have $\mathcal{N} \not\sqsubseteq_{\text{must}} \mathcal{N}$. To this end, let

$$\mathcal{T}' = \Gamma_T \triangleright o_1[[c?(x).c!\langle 0 \rangle . \mathbf{0}]] \mid o_2[[c?(x).c?(y). \text{if } y = 0 \text{ then } \omega \text{ else } \mathbf{0}]]$$

In this case we have that $\mathcal{O}(\mathcal{M} \# \mathcal{T}') = \{0, 1\}$, while $\mathcal{O}(\mathcal{N} \# \mathcal{T}') = \{1\}$, and by definition $\mathcal{O}(\mathcal{N} \# \mathcal{T}') \not\sqsubseteq_S \mathcal{O}(\mathcal{M} \# \mathcal{T}')$.

One might also think it possible to use the difference between broadcast and multicast to design a test \mathcal{T}'' for which $\mathcal{O}(\mathcal{N} \# \mathcal{T}'') \not\sqsubseteq_H \mathcal{O}(\mathcal{M} \# \mathcal{T}'')$ and $\mathcal{O}(\mathcal{M} \# \mathcal{T}'') \not\sqsubseteq_S \mathcal{O}(\mathcal{N} \# \mathcal{T}'')$.

For example, if we let $\mathcal{T}'' = \mathcal{T}'$ we obtain that $1 \in \mathcal{O}(\mathcal{N} \# \mathcal{T}')$. This is because in $\mathcal{N} \# \mathcal{T}'$ the second value received by o_2 is always w . However we also have that $1 \in \mathcal{O}(\mathcal{M} \# \mathcal{T}')$, since the simultaneous broadcast in \mathcal{N} can be simulated by a multicast in \mathcal{M} , by node m first broadcasting to o_1 followed by n broadcasting to o_2 . As this line of reasoning is independent from the test \mathcal{T}' , it also applies to all those networks that can be used to test the behaviour of \mathcal{M} and \mathcal{N} ; this leads to the intuition that $\mathcal{N} \sqsubseteq_{\text{may}} \mathcal{M}$, which will be proved formally later as a consequence of Example 6.4 and Theorem 6.2. Similarly, Theorem 6.14 shows that $\mathcal{M} \sqsubseteq_{\text{must}} \mathcal{N}$.

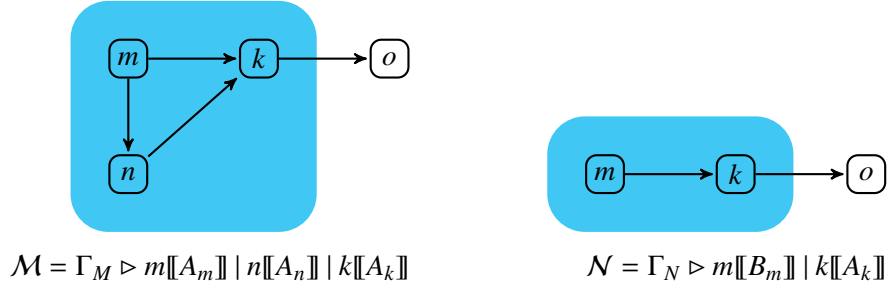
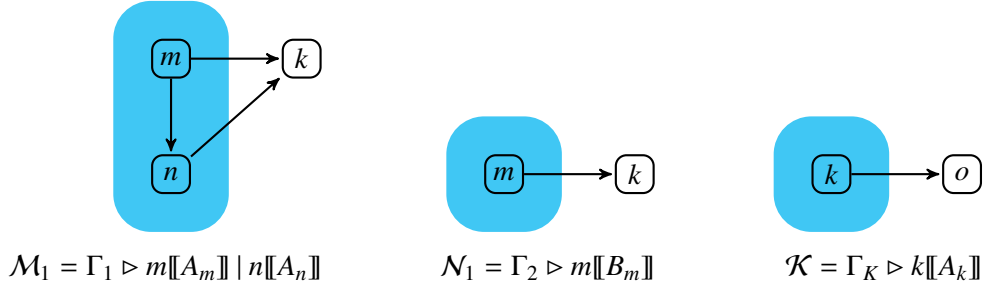


Figure 9: Two networks with a common sub-network


 Figure 10: Decomposition of the networks \mathcal{M} and \mathcal{N}

□

One pleasing property of the behavioural preorders is that they allow compositional reasoning over networks.

Proposition 4.16 (Compositionality). *Let $\mathcal{M}_1, \mathcal{M}_2$ be two networks such that $\mathcal{M}_1 \sqsubseteq_{\text{may}} \mathcal{M}_2$ ($\mathcal{M}_1 \sqsubseteq_{\text{must}} \mathcal{M}_2$), and let \mathcal{N} be another network such that both $(\mathcal{M}_1 \sharp \mathcal{N})$ and $(\mathcal{M}_2 \sharp \mathcal{N})$ are defined. Then $(\mathcal{M}_1 \sharp \mathcal{N}) \sqsubseteq_{\text{may}} (\mathcal{M}_2 \sharp \mathcal{N})$ ($(\mathcal{M}_1 \sharp \mathcal{N}) \sqsubseteq_{\text{must}} (\mathcal{M}_2 \sharp \mathcal{N})$).*

Proof. A direct consequence of \sharp being both associative and interface preserving. □

We end this section with an application of this compositionality result.

Example 4.17. Consider the networks \mathcal{M} and \mathcal{N} in Figure 9, where the codes at the various nodes are given by

$$\begin{aligned}
 A_m &\leftarrow c!\langle v \rangle . \mathbf{0} \\
 A_n &\leftarrow c?(x) . d!\langle w \rangle . \mathbf{0} \\
 A_k &\leftarrow c?(x) . d?(y) . e!\langle u \rangle . \mathbf{0} \\
 B_m &\leftarrow c!\langle v \rangle . d!\langle w \rangle . \mathbf{0}
 \end{aligned}$$

It is possible to write both of them respectively as $\mathcal{M}_1 \sharp \mathcal{K}$ and $\mathcal{N}_1 \sharp \mathcal{K}$, where the networks $\mathcal{M}_1, \mathcal{N}_1$ and \mathcal{K} are depicted in Figure 10. In order to prove that $\mathcal{M} \sqsubseteq_{\text{may}} \mathcal{N}$ ($\mathcal{N} \sqsubseteq_{\text{must}} \mathcal{M}$), it is therefore sufficient to focus on their respective sub-networks \mathcal{M}_1 and \mathcal{N}_1 , and prove $\mathcal{M}_1 \sqsubseteq_{\text{may}} \mathcal{N}_1$ ($\mathcal{N}_1 \sqsubseteq_{\text{must}} \mathcal{M}_1$). The equivalence of the two original networks will then follow from a direct application of Proposition 4.16. □

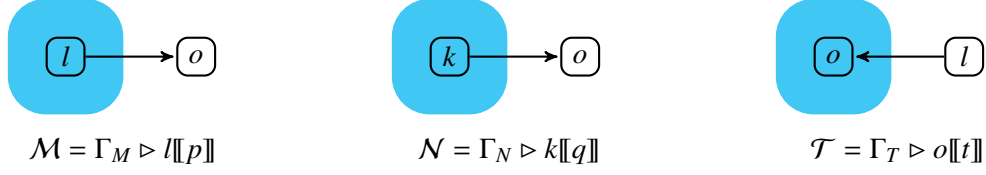


Figure 11: A problem with the composition operator

4.4. **Justifying the operator \sharp .** Here we revisit Definition 4.3 and in particular investigate the possibility of using alternative composition operators. The remainder of the paper is independent of this section and so it may be safely skipped by the uninterested reader.

Here we take a more general approach to composition; rather than give a particular operator we discuss natural properties we would expect of such operators. Let us just presuppose a *consistency predicate* \mathcal{P} on pairs of networks determining when their composition should be defined. The only requirement on $\mathcal{M}\mathcal{P}\mathcal{N}$ is that whenever it is defined the resulting composite network is well-formed. Since the composite network should be determined by that of its components this amounts to requiring that

$$\mathcal{P}(\mathcal{M}, \mathcal{N}) \text{ implies } \text{nodes}(\mathcal{M}) \cap \text{nodes}(\mathcal{N}) = \emptyset \quad \text{for any } \mathcal{M}, \mathcal{N} \in \text{Nets} \quad (4.4)$$

Given a consistency predicate satisfying (4.4) we can now generalise Definition 4.3 to give a range of different composition operators.

Definition 4.18. [General composition of networks] Let \mathcal{P} be a consistency predicate on networks in Nets satisfying (4.4). Then we define the associated partial composition relation by:

$$(\Gamma_M \triangleright M) \parallel_{\mathcal{P}} (\Gamma_N \triangleright N) = \begin{cases} (\Gamma_M \cup \Gamma_N) \triangleright (M \mid N), & \text{if } \mathcal{P}((\Gamma_M \triangleright M), (\Gamma_N \triangleright N)) \\ \text{undefined,} & \text{otherwise} \end{cases}$$

The connectivity graph $\Gamma_M \cup \Gamma_N$ is defined as in Definition 4.3. \square

Example 4.19. Let \mathcal{P}_s be the partial binary predicate defined by letting $\mathcal{P}_s(\mathcal{M}, \mathcal{N})$ whenever

- $\text{nodes}(\mathcal{M}) \cap \text{nodes}(\mathcal{N}) = \emptyset$
- $\mathcal{M} \vdash m \rightarrow n$ if and only if $\mathcal{N} \vdash m \rightarrow n$, for every $m \in \text{nodes}(\mathcal{M})$ and $n \in \text{nodes}(\mathcal{N})$,
- $\mathcal{M} \vdash m \leftarrow n$ if and only if $\mathcal{N} \vdash m \leftarrow n$, for every $m \in \text{nodes}(\mathcal{M})$ and $n \in \text{nodes}(\mathcal{N})$,

By definition this satisfies the requirement (4.4) above, and intuitively it only allows the composition whenever the two individual networks agree on the interconnections between internal and external nodes.

For notational convenience we denote the operator $\parallel_{\mathcal{P}_s}$ with \parallel . It is easy to check that this operator is both associative and commutative. \square

Thus a priori this composition operator \parallel could equally well be used to develop the testing theory in Section 4.3. Unfortunately the resulting theory would be degenerate.

Example 4.20. [Example 4.19 continued] Consider the networks \mathcal{M}, \mathcal{N} in Figure 11, where we choose $p = c!\langle v \rangle$ and $q = c!\langle w \rangle$ for two different values v, w . Then intuitively \mathcal{M} and \mathcal{N} should have different observable behaviour, observable by placing a test at the node o . However if the operator \parallel is used to combine a test with the network being observed they are indistinguishable.

This is because if there is a network \mathcal{T} such that $\mathcal{M} \parallel \mathcal{T}$ and $\mathcal{N} \parallel \mathcal{T}$ are well-defined then o can not be in $\text{nodes}(\mathcal{T})$. For if o were in $\text{nodes}(\mathcal{T})$, then since $\mathcal{M} \vdash l \rightarrow o$ the definition of the operator implies that $\mathcal{T} \vdash l \rightarrow o$. This in turn implies that $\mathcal{N} \vdash l \rightarrow o$, which is not true.

Now since no testing network which can be applied to both \mathcal{M} and \mathcal{N} can place any code at o , no difference can be discovered between them. \square

The question now naturally arises about which consistency predicate \mathcal{P} lead to reasonable composition operators $\parallel_{\mathcal{P}}$, in the sense that at least the resulting testing theories are not degenerate. We want to be able to compare networks with different connectivity graphs, and possibly different nodes, such as \mathcal{M} and \mathcal{N} in Figure 11. We also should not be able to change the connectivity of the internal nodes of a network when we test it; we wish to implement *black-box* testing, where the nodes containing running code cannot be accessed directly.

These informal requirements can be formulated as natural requirements on composition operators. The first says that the composed network is completely determined by the components:

(I) Merge: the operator \parallel should be determined by some predicate \mathcal{P} using Definition 4.3.

Intuitively the interface of a network is how their external behaviour is to be observed. Since our aim is to enable compositional reasoning over networks, we would expect composition to preserve interfaces:

(II) Interface preservation: If $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$ and \mathcal{T} can be composed with both, that is both $\mathcal{M} \parallel \mathcal{T}$ and $\mathcal{N} \parallel \mathcal{T}$ are well-defined, then $\text{In}(\mathcal{M} \parallel \mathcal{T}) = \text{In}(\mathcal{N} \parallel \mathcal{T})$, $\text{Out}(\mathcal{M} \parallel \mathcal{T}) = \text{Out}(\mathcal{N} \parallel \mathcal{T})$.

The final requirement captures the intuitive idea that reorganising the internal structure of a network should not affect the ability to perform a test; in fact the reorganisation is simply a renaming of nodes. Let σ be a permutation of node names. We use $(\Gamma \triangleright M)\sigma$ to denote the result of applying σ to the node names in M and in the connectivity graph Γ .

(III) Renaming: Suppose $\mathcal{M} \parallel \mathcal{T}$ is defined. Then $\mathcal{M}\sigma \parallel \mathcal{T}$ is also defined, provided σ is a node permutation which satisfies

- $\sigma(e) = e$ for every $e \in \text{Intf}(\mathcal{M})$
- no $n \in \text{nodes}(\mathcal{T})$ appears in the range of σ ; that is $n \in \text{nodes}(\mathcal{T})$ implies $\sigma(n) = n$.

Example 4.21. The operator \parallel does not satisfy **(III)**, as can be seen using the simple networks in Figure 11; $\mathcal{M} \parallel \mathcal{T}$ is obviously well-defined. However, consider the renaming σ which swaps node names l to k , which is valid with respect to \mathcal{T} ; the network $\mathcal{M}\sigma \parallel \mathcal{T}$ is not defined, as $\Gamma_{\mathcal{T}} \not\vdash k \rightarrow o$. A slight modification will demonstrate that interfaces are also not preserved by this operator. \square

Proposition 4.22. Suppose \parallel satisfies the conditions (I) - (III) above. Then $\text{nodes}(\mathcal{M}) \cap \text{Intf}(\mathcal{N}) = \emptyset$ whenever $\mathcal{M} \parallel \mathcal{N}$ is defined.

Proof. By contradiction; let $\mathcal{M} = \Gamma_M \triangleright M$ and $\mathcal{N} = \Gamma_N \triangleright N$. Assume that m is a node included in $\text{nodes}(\mathcal{M}) \cap \text{Intf}(\mathcal{N})$, and that $\mathcal{M} \parallel \mathcal{N}$ is defined. Finally, let σ be an arbitrary permutation such that $\mathcal{M}\sigma \parallel \mathcal{N}$ is defined. Note that the following statements are true:

- (1) $m \notin \text{nodes}(\mathcal{N})$,
- (2) $\mathcal{N} \vdash m$,
- (3) $\text{Intf}(\mathcal{M}) = \text{Intf}(\mathcal{M}\sigma)$.
- (4) $m \notin \text{Intf}(\mathcal{M} \parallel \mathcal{N})$.

For proving the last statement just note that $m \in \text{nodes}(\mathcal{M})$ by hypothesis, hence $m \in \text{nodes}(\mathcal{M} \parallel \mathcal{N})$. By definition of interface it follows that $m \notin \text{Intf}(\mathcal{M} \parallel \mathcal{N})$.

Let l be a node which is not contained in Γ_M , nor in Γ_N . Consider the permutation σ which swaps nodes m and l ; that is $\sigma(m) = l, \sigma(l) = m$ and $\sigma(k) = k$ for all $k \neq m, l$. Since $\mathcal{M} \not\vdash l$ we also have that $l \notin \text{nodes}(\mathcal{M})$, so that $\sigma(l) = m \notin \text{nodes}(\mathcal{M}\sigma)$. Further, the permutation σ is consistent with condition (III), renaming, when applied to networks \mathcal{M} and \mathcal{N} ; therefore $\mathcal{M}\sigma \parallel \mathcal{N}$ is defined.

Since $m \notin \text{nodes}(\mathcal{M}\sigma)$, by (1) above it follows that $m \notin \text{nodes}(\mathcal{M}\sigma \parallel \mathcal{N})$; by (2), we obtain that $(\Gamma_M)\sigma \cup \Gamma_N \vdash m$. These two statements ensure that $m \in \text{Intf}(\mathcal{M}\sigma \parallel \mathcal{N})$.

As a direct consequence of (3) and condition (II), interface preservation, we also have that $m \in \text{Intf}(\mathcal{M} \parallel \mathcal{N})$, but this contradicts (4). \square

Corollary 4.23. Let \parallel be any symmetric composition operator which satisfies the conditions (I) - (III). Suppose $\mathcal{M}_1 \parallel \mathcal{M}_2$ is well-defined, and of the form $\Gamma \triangleright M$. Then $\Gamma \vdash m_1 \not\approx m_2$ whenever $m_i \in \text{nodes}(\mathcal{M}_i)$.

Proof. A simple consequence of the previous result. \square

What this means is that if we use such a symmetric operator when applying a test to a network, as in Definition 4.7, then the resulting testing preorder will be degenerate; it will not distinguish between any pair of nets. In some sense this result is unsurprising. For \mathcal{T} to test \mathcal{M} in $\mathcal{M} \parallel \mathcal{T}$ it must have code running at the interface of \mathcal{M} . But, as we have seen, condition (III) more or less forbids \mathcal{T} to have code running at the interface of \mathcal{M} .

Thus we have ruled out the possibility of basing our testing theory on a symmetric composition operator. The question now remains what composition operator is the most appropriate? We have already stated that conditions (I)-(III) are natural, and since there are no further obvious requirements we could choose the operator with greatest expressive power among those that satisfy conditions (I)-(III). Here an operator $\parallel_{\mathcal{P}_1}$ is more expressive than another $\parallel_{\mathcal{P}_2}$ if, whenever $\mathcal{M} \parallel_{\mathcal{P}_1} \mathcal{N}$ is defined for any two arbitrary networks \mathcal{M}, \mathcal{N} , then so is $\mathcal{M} \parallel_{\mathcal{P}_2} \mathcal{N}$, and the result of the two compositions above is the same. The next Lemma shows that the operator which we are looking for is exactly \sharp .

Lemma 4.24. Let \mathcal{P} be any consistency predicate satisfying (4.4) above. Then if $\mathcal{M} \parallel_{\mathcal{P}} \mathcal{N}$ is defined so is $\mathcal{M} \sharp \mathcal{N}$ and moreover $\mathcal{M} \parallel_{\mathcal{P}} \mathcal{N} = \mathcal{M} \sharp \mathcal{N}$.

Proof. Obvious from the definition of \sharp in Definition 4.3. \square

5. EXTENSIONAL SEMANTICS

As explained in papers such as [30, 18], contextual equivalences and preorders are determined by so-called *extensional actions*, which consist of the observable activities which a system can have with its external environment. We present an extensional Semantics for probabilistic wireless networks in Section 5.1. The final two sections develop technical properties of these actions, which will be used in the later soundness proofs. They may be safely skipped at first reading. Section 5.2 is devoted to basic decomposition and composition results for extensional actions, while in Section 5.3 we relate the extensional actions we introduced with the reduction relation of the testing structures associated with networks.

5.1. Extensional Actions. Here we design a pLTS whose set of actions can be detected (hence tested) by the external environment. The intensional semantics in Section 3 already provides a pLTS and it is instructive to see why this is not appropriate.

Consider \mathcal{M} and \mathcal{N} from Figure 11, and suppose further that the code p and q , running at l and k respectively, is identical, $c!\langle v \rangle$. Then we would expect \mathcal{M} and \mathcal{N} to be behaviourally indistinguishable. However \mathcal{M} will have an output action, labelled $l.c!v$, which is not possible for \mathcal{N} . So output actions cannot record their source node. What turns out to be important is the set of target nodes. For example if in \mathcal{M} we added a new output node m to the interface, with a connection from l then we would be able to distinguish \mathcal{M} from \mathcal{N} ; the required test would simply place some appropriate testing code at the new node m .

Now we present an extensional semantics for networks; here the visible actions consist of activities which can be detected (hence tested) by placing code at the interface of a network. In this semantics we have internal, input and output actions.

Definition 5.1. [Extensional actions] The actions of the extensional semantics are defined as follows:

- (1) **internal**, $(\Gamma \triangleright M) \xrightarrow{\tau} (\Gamma \triangleright \Delta)$; some internal activity reduces the system M , relative to the connectivity Γ , to some system N , where $N \in [\Delta]$. Here the internal activity of a network coincides either with some node performing a silent move $m.\tau$ or broadcasting a value which cannot be detected by any node in the interface of the network itself.

Formally, $(\Gamma \triangleright M) \xrightarrow{\tau} (\Gamma \triangleright \Delta)$ whenever $M \not\equiv m[\omega + s] \mid N$ and

- (a) $\Gamma \triangleright M \xrightarrow{m.\tau} \Delta$
 (b) or $\Gamma \triangleright M \xrightarrow{n.c!v} \Delta$ for some value v , channel c and node name n satisfying $\Gamma \vdash n \rightarrow m$ implies $m \in \text{nodes}(M)$

Note that we are using the notation given in Section 3 for defining distributions. Here Δ is a distribution over \mathbf{sSys} and so $(\Gamma \triangleright \Delta)$ is a distribution over networks; however all networks in its support use the same network connectivity Γ .

- (2) **input**, $(\Gamma \triangleright M) \xrightarrow{n.c?v} (\Gamma \triangleright \Delta)$; an observer placed at node n can send the value v along the channel c to the network $(\Gamma \triangleright M)$. For the observer to be able to place the code at node n we must have $n \in \text{In}(\Gamma \triangleright M)$.

Formally $(\Gamma \triangleright M) \xrightarrow{n.c?v} (\Gamma \triangleright \Delta)$ whenever $M \not\equiv m[\omega + s] \mid N$ and

- (a) $\Gamma \triangleright M \xrightarrow{n.c?v} \Delta$
 (b) $n \in \text{In}(\Gamma \triangleright M)$

- (3) **output**, $(\Gamma \triangleright M) \xrightarrow{c!v\triangleright\eta} (\Gamma \triangleright \Delta)$, where η is a non-empty set of nodes; an observer placed at any node $n \in \eta$ can receive the value v along the channel c . For this to happen each node $n \in \eta$ must be in $\text{Out}(\Gamma \triangleright M)$, and there must be some code running at some node in M which can broadcast along channel c to each such n .

Formally, $(\Gamma \triangleright M) \xrightarrow{c!v\triangleright\eta} (\Gamma \triangleright \Delta)$ whenever $M \not\equiv m[\omega + s] \mid N$

- (i) $\Gamma \triangleright M \xrightarrow{m.c!v} \Delta$ for some node m
 (ii) $\eta = \{n \in \text{Out}(\Gamma \triangleright M) \mid \Gamma \vdash m \rightarrow n\} \neq \emptyset$. \square

In the following we will use the metavariable λ to range over extensional actions. These actions endow the set of networks with the structure of a pLTS. Thus the terminology used for pLTSs is extended to networks, so that in the following we will use terms such as finitary networks or finite branching networks; we use the symbol \Longrightarrow to denote hyper-derivations in the extensional pLTS

of networks, and \Longrightarrow to denote extreme derivations. Also note that we allow an extensional actions to be performed only in the case that a network is not ω -successful. As we have already stated, we see pLTSs as non-deterministic probabilistic experiments whose success is obtained by reaching an ω -successful state. When an ω -successful state is reached, we require the experiment to terminate.

A trivial application of Corollary 3.6 ensures that extensional actions are preserved by structurally congruent networks. Further, they do not change the topological structure of a network.

Proposition 5.2. *Suppose that $\Gamma \triangleright M \xrightarrow{\lambda} \Delta$, $\Delta \in \mathcal{D}(\text{Nets})$. Then $\Delta = \Gamma \triangleright \Theta$, and $\Theta \in \mathcal{D}(\text{sSys})$ is node stable. Further, for any $N \in [\Delta]$ we have that $\text{nodes}(N) = \text{nodes}(M)$.*

Proof. The definition of extensional actions ensures that whenever $\Gamma \triangleright M \xrightarrow{\lambda} \Delta$ then $\Delta = \Gamma \triangleright \Theta$ for some $\Theta \in \mathcal{D}(\text{sSys})$. The fact that for any $N \in [\Theta]$ we have that $\text{nodes}(M) = \text{nodes}(N)$ follows from Corollary 3.8. \square

Note that, if two distributions $\Delta, \Theta \in \mathcal{D}(\text{Nets})$ are node-stable, if $M \sharp N$ is defined for some $M \in [\Delta], N \in [\Theta]$, then $M' \sharp N'$ is defined for any $M' \in [\Delta], N' \in [\Theta]$. Thus, for node-stable distributions of networks it makes sense to lift the operator \sharp to distributions of networks, defined directly via an application of Equation 2.2. A similar argument holds for the symmetric operator \parallel .

In the following we will need *weak* versions of extensional actions, which abstract from internal activity, provided by the relation $\xrightarrow{\tau}$. Internal activity can be modelled by the hyper-derivation relation \Longrightarrow , which is a probabilistic generalisation of the more standard weak internal relation $\xrightarrow{\tau}^*$.

Definition 5.3. [Weak extensional actions]

- (1) Let $M \xrightarrow{\tau} \Delta$ whenever we have the hyper-derivation $M \Longrightarrow \Delta$
- (2) $M \xrightarrow{n.c?v} \Delta$ whenever $M \Longrightarrow \xrightarrow{n.c?v} \Longrightarrow \Delta$
- (3) Let $M \xrightarrow{c!v \triangleright \eta} \Delta$ be the least relation satisfying:
 - (a) $M \xrightarrow{c!v \triangleright \eta} \Longrightarrow \Delta$ implies $M \xrightarrow{c!v \triangleright \eta} \Delta$
 - (b) $M \xrightarrow{c!v \triangleright \eta_1} \Delta', \Delta' \xrightarrow{c!v \triangleright \eta_2} \Delta$, where $\eta_1 \cap \eta_2 = \emptyset$, implies $M \xrightarrow{c!v \triangleright (\eta_1 \cup \eta_2)} \Delta$ \square

These weak actions endow the set of networks Nets with the structure of another pLTS, called the *extensional pLTS* and denoted by $\text{pLTS}_{\text{Nets}}$.

Some explanation is necessary for the non-standard definition of output actions in Definition 5.3(3). Informally speaking, the definition of weak extensional output actions expresses the capability of simulating broadcast through multicast, which has already been observed in Example 4.15. A single (weak) broadcast action detected by a set of nodes η can be matched by a sequence of weak broadcast actions, detected respectively by $\eta_1, \dots, \eta_i \subseteq \eta$, provided that the collection $\{\eta_1, \dots, \eta_i\}$ is a partition of η . This constraint is needed to ensure that

- (i) every node in η will detect the transmitted value and
- (ii) no node in η will detect the value more than once.

As we will see in Section 6, the ability of a multicast to simulate a broadcast is captured by the testing preorders. Roughly speaking, in a generic network M a broadcast can be converted into a multicast³ leading to a network N such that $M \sqsubseteq_{\text{may}} N$. Conversely, in the must-testing setting a

³ Note that this operation would require a change in the internal topology of M .


 Figure 12: A problem with decomposition of actions with respect to \sharp

multicast in a network \mathcal{M} can be replaced by a broadcast leading to a network \mathcal{N} such that $\mathcal{M} \sqsubseteq_{\text{must}} \mathcal{N}$.

5.2. Composition and Decomposition Results. In this Section we prove decomposition and composition results for the extensional actions introduced in the previous section. In its most general form, the results we want to develop can be summarised as follows: given a network $\mathcal{M} \sharp \mathcal{N}$,

Strong Decomposition: for actions of the form $(\mathcal{M} \sharp \mathcal{N}) \xrightarrow{\lambda} \Delta$ we want to determine two actions of the form $\mathcal{M} \xrightarrow{\lambda_1} \Delta$ and $\mathcal{N} \xrightarrow{\lambda_2} \Theta$, where $(\Delta \sharp \Theta) = \Delta$,

Weak Composition: conversely, given two actions of the form $\mathcal{M} \xrightarrow{\lambda_1} \Delta$ and $\mathcal{N} \xrightarrow{\lambda_2} \Theta$, we want to determine an action of the form $(\Delta \sharp \Theta) \xrightarrow{\lambda} (\Delta \parallel \Theta)$.

Unfortunately, the following example shows that this task cannot be achieved by relying solely on the extensional semantics.

Example 5.4. Consider the networks \mathcal{M}, \mathcal{N} of Figure 12. It is straightforward to note that $\mathcal{M} \sharp \mathcal{N}$ is defined; further, $(\mathcal{M} \sharp \mathcal{N}) \xrightarrow{\tau} \overline{(\mathcal{M}' \sharp \mathcal{N}'})$, where $\mathcal{M}' = (\Gamma_M \triangleright m[[\mathbf{0}]])$, $\mathcal{N}' = (\Gamma_N \triangleright o[[\mathbf{0}]])$.

One could wish to be able to infer this action from the broadcast action of the form $\mathcal{M} \xrightarrow{c!v \triangleright \{o\}} \overline{\mathcal{M}'}$ and an input action of the form $\mathcal{N} \xrightarrow{m.c?v} \overline{\mathcal{N}'}$. Unfortunately, this last action cannot be inferred, since in \mathcal{N} node o cannot detect the broadcasts performed by node m . \square

The problem of Example 5.4 arises because the connection between from node m to node n , which is present in $\mathcal{M} \sharp \mathcal{N}$, is not present in the right hand side of the composition \mathcal{N} . As a consequence, the action $\mathcal{N} \xrightarrow{m.c?v} \overline{\mathcal{N}'}$ cannot be derived. However, note that we could still decompose the transition $\mathcal{M} \sharp \mathcal{N} \xrightarrow{\tau} \overline{\mathcal{M}' \sharp \mathcal{N}'}$ if we were to focus on the code run by node o in \mathcal{N} , rather than on the network \mathcal{N} itself.

Example 5.5. Consider again the networks \mathcal{M}, \mathcal{N} of Figure 12, and recall that $\mathcal{N} = \Gamma_N \triangleright o[[c?(x) . \mathbf{0}]]$. Given the action $\mathcal{M} \sharp \mathcal{N} \xrightarrow{\tau} \overline{\mathcal{M}' \sharp \mathcal{N}'}$, where we recall that $\mathcal{N}' = o[[\mathbf{0}]]$, we can now infer the extensional transition $\mathcal{M} \xrightarrow{c!v \triangleright \{o\}} \overline{\mathcal{M}'}$ and the process transition $c?(x) . \mathbf{0} \xrightarrow{c?v} \mathbf{0}$.

The first transitions says that some node in \mathcal{M} performs a broadcast which affects node o , while the second one says that how the code which node o is running reacts to a broadcast. Therefore, it is not surprising to note that the two transitions $\mathcal{M} \xrightarrow{c!v \triangleright \{o\}} \overline{\mathcal{M}'}$ and $P \xrightarrow{c?v} \mathbf{0}$ can be combined together to obtain the original transition $\mathcal{M} \sharp \mathcal{N} \xrightarrow{\tau} \overline{\mathcal{M}' \sharp \mathcal{N}'}$. \square

Example 5.5 leads us to the intuition that composition and decomposition results of extensional actions can be developed if we focus on composed networks of the form $\mathcal{M} \sharp \mathcal{G}$, where $\mathcal{G} \in \mathbb{G}$. Intuitively, such results can be obtained by reasoning on the extensional transitions of \mathcal{M} and the process transitions performed by the only internal node in \mathcal{G} .

Proposition 5.6. [Strong decomposition in $\text{pLTS}_{\text{Nets}}$] Let $\mathcal{M} \in \text{Nets}$, $\mathcal{G} \in \mathbb{G}$ be two networks such that $\mathcal{M} \sharp \mathcal{G}$ is defined. Further, let Γ_N, n, s be such that $\mathcal{G} = (\Gamma_N \triangleright n[[s]])$; Suppose $\mathcal{M} \sharp \mathcal{G} \xrightarrow{\lambda} \Lambda$, $\Lambda \in \mathcal{D}(\text{Nets})$. Then $\Lambda = \Delta \sharp (\Gamma_N \triangleright n[[\Theta]])$ for some $\Delta, \Theta \in \mathcal{D}(\text{Nets})$ such that

- (1) if $\lambda = \tau$ then either
 - (i) $\mathcal{M} \xrightarrow{\tau} \Delta$, $\Theta = \bar{s}$, or
 - (ii) $\Delta = \overline{\mathcal{M}}$ and $s \xrightarrow{\tau} \Theta$, or
 - (iii) $\mathcal{M} \xrightarrow{c!v \triangleright \{n\}} \Delta$, $s \xrightarrow{c?v} \Theta$, or
 - (iv) $\mathcal{M} \xrightarrow{c!v \triangleright \{n\}} \Delta$, $s \not\xrightarrow{c?v}$ and $\Theta = \bar{s}$, or
 - (v) $\text{Out}(\mathcal{G}) = \emptyset$, $\mathcal{M} \xrightarrow{n.c?v} \Delta$, $s \xrightarrow{c!v} \Theta$, or
 - (vi) $\text{Out}(\mathcal{G}) = \emptyset$, $n \notin \text{In}(\mathcal{M})$, $\Delta = \overline{\mathcal{M}}$ and $s \xrightarrow{c!v} \Theta$.
- (2) if $\lambda = c!v \triangleright \eta$ then either
 - (i) $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta$, $n \notin \eta$ and $\Theta = \bar{s}$, or
 - (ii) $\mathcal{M} \xrightarrow{c!v \triangleright \eta \cup \{n\}} \Delta$, $n \notin \eta$ and $s \xrightarrow{c?v} \Theta$, or
 - (iii) $\mathcal{M} \xrightarrow{c!v \triangleright \eta \cup \{n\}} \Delta$, $n \notin \eta$, $s \not\xrightarrow{c?v}$ and $\Theta = \bar{s}$, or
 - (iv) $\mathcal{M} \xrightarrow{n.c?v} \Delta$, $s \xrightarrow{c!v} \Theta$ and $\eta = \text{Out}(\mathcal{G})$, or
 - (v) $n \notin \text{In}(\mathcal{M})$, $\Delta = \overline{\mathcal{M}}$, $s \xrightarrow{c!v} \Theta$ and $\eta = \text{Out}(\mathcal{G})$,
- (3) if $\lambda = m.c?v$, where $m \neq n$, then either
 - (i) $\mathcal{M} \xrightarrow{m.c?v} \Delta$, $\Theta = \bar{s}$ and $m \notin \text{In}(\mathcal{G})$, or
 - (ii) $\Delta = \overline{\mathcal{M}}$, $m \notin \text{In}(\mathcal{M})$, $s \xrightarrow{c?v} \Theta$, or
 - (iii) $\Delta = \overline{\mathcal{M}}$, $m \notin \text{In}(\mathcal{M})$, $s \not\xrightarrow{c?v}$ and $\Theta = \bar{s}$, or
 - (iv) $\mathcal{M} \xrightarrow{m.c?v} \Delta$, $s \xrightarrow{c?v} \Theta$ and $m \in \text{In}(\mathcal{G})$, or
 - (v) $\mathcal{M} \xrightarrow{m.c?v} \Delta$, $s \not\xrightarrow{c?v}$ and $\Theta = \bar{s}$.

Proof. The proof of this Proposition is quite technical, and it is therefore relegated to Appendix B, Page 61. \square

Next we consider how the weak actions performed by a stable sub-distribution of the form $\Delta \sharp (\Gamma_N \triangleright n[[s]])$, can be inferred by a weak action performed by the node-stable distribution Δ and a process transition performed by the state s , respectively. For our purposes it will suffice to combine a weak extensional transition with a strong process transition.

Proposition 5.7. [Weak/Strong composition in $\text{pLTS}_{\text{Nets}}$] Let $\Delta \in \mathcal{D}_{\text{sub}}(\text{Nets})$ be a node-stable sub-distributions of networks. Let also Γ_N, n, s be such that $\Delta \sharp (\Gamma_N \triangleright n[[s]])$ is well-defined. Further, let $\mathcal{G} \in [\Gamma_N \triangleright n[[s]]]$; here note that the sets $\text{Out}(\mathcal{G})$ and $\text{In}(\mathcal{G})$ are completely determined by the connectivity graph Γ_N .

(1) Composition resulting in internal activity:

- (i) $\Delta \xrightarrow{\tau} \Delta'$ implies $\Delta \sharp (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta' \sharp (\Gamma_N \triangleright n[[\bar{s}]])$,
- (ii) $s \xrightarrow{\tau} \Theta$ implies $\Delta \sharp (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta \sharp (\Gamma_N \triangleright n[[\Theta]])$,
- (iii) $\Delta \xrightarrow{c!v \triangleright \{n\}} \Delta'$ and $s \xrightarrow{c?v} \Theta$ implies $\Delta \sharp (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta' \sharp (\Gamma_N \triangleright n[[\Theta']])$,
- (iv) $\Delta \xrightarrow{c!v \triangleright \{n\}} \Delta'$ and $s \not\xrightarrow{c?v}$ implies $\Delta \sharp (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta' \sharp (\Gamma_N \triangleright n[[\bar{s}]])$,
- (v) $\Delta \xrightarrow{n.c?v} \Delta'$, $\text{Out}(\mathcal{G}) = \emptyset$ and $s \xrightarrow{c!v} \Theta$, then, then $\Delta \sharp (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta' \sharp (\Gamma_N \triangleright n[[\Theta]])$,

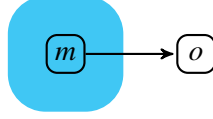


Figure 13: Relating reductions with extensional actions.

- (vi) for any $\mathcal{M} \in [\Delta]$, $n \notin \text{In}(\mathcal{M})$, $\text{Out}(\mathcal{G}) = \emptyset$ and $s \xrightarrow{c!v} \Theta$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{\tau} \Delta \# (\Gamma_N \triangleright n[\Theta])$.
- (2) Composition resulting in an extensional output:
- (i) $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$, $n \notin \eta$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$,
 - (ii) $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$, $\{n\} \subset \eta$ and $s \xrightarrow{c?v} \Theta$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta \setminus \{n\}} \Delta' \# (\Gamma_N \triangleright n[\Theta])$,
 - (iii) $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$, $\{n\} \subset \eta$ and $s \not\xrightarrow{c?v}$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta \setminus \{n\}} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$,
 - (iv) $s \xrightarrow{c!v} \Theta$, $\text{Out}(\mathcal{G}) \neq \emptyset$, $\Delta \xrightarrow{n.c?v} \Delta'$ implies that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \text{Out}(\mathcal{G})} \Delta' \# (\Gamma_N \triangleright n[\Theta])$,
 - (v) $n \notin \text{In}(\mathcal{M})$ for any $\mathcal{M} \in [\Delta]$ and $s \xrightarrow{c!v} \Theta'$, $\text{Out}(\mathcal{G}) \neq \emptyset$ implies that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \text{Out}(\mathcal{G})} \Delta \# (\Gamma_N \triangleright n[\Theta])$.
- (3) Composition resulting in an input:
- (i) $\Delta \xrightarrow{m.c?v} \Delta'$ and $m \notin \text{In}(\mathcal{G})$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{m.c?v} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$,
 - (ii) $m \notin \text{In}(\mathcal{M})$ for $\mathcal{M} \in [\Delta]$, $m \in \text{In}(\mathcal{G})$ and $s \xrightarrow{c?v} \Theta$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{m.c?v} \Delta \# (\Gamma_N \triangleright n[\Theta])$,
 - (iii) $m \notin \text{In}(\mathcal{M})$ for $\mathcal{M} \in [\Delta]$, $m \in \text{In}(\mathcal{G})$ and $s \not\xrightarrow{c?v}$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{m.c?v} \Delta \# (\Gamma_N \triangleright n[\bar{s}])$,
 - (iv) $\Delta \xrightarrow{m.c?v} \Delta'$, $m \in \text{In}(\mathcal{G})$ and $s \xrightarrow{c?v} \Theta$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{m.c?v} \Delta' \# (\Gamma_N \triangleright n[\Theta])$,
 - (v) $\Delta \xrightarrow{m.c?v} \Delta'$, $m \in \text{In}(\mathcal{G})$ and $s \not\xrightarrow{c?v}$ implies $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{m.c?v} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$.

Proof. See Appendix B, Page 63. □

5.3. Relating Extensional Actions and Reductions. In this section we investigate the relationship between extensional actions and reductions, defined in Section 4.2.

It follows immediately, from the definition of the reduction relation \rightarrow , that for any network \mathcal{M} we have $\mathcal{M} \rightarrow \Delta$ if and only if either $\mathcal{M} \xrightarrow{\tau} \Delta$ or $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta$. However, this result is not true anymore if we focus on the extensional transitions performed by a distribution of networks.

Example 5.8. [Reductions and Extensional Actions] Consider the network distribution $\Gamma \triangleright \Delta$, where Γ is depicted in Figure 13 and

$$\Delta = 1/2 \cdot \overline{m[\tau. \mathbf{0}]} + 1/2 \cdot \overline{m[c!\langle v \rangle. \mathbf{0}]}.$$

Note that we have the reductions $\Gamma \triangleright m[\tau. \mathbf{0}] \rightarrow \overline{\Gamma \triangleright m[\mathbf{0}]}$, and $\Gamma \triangleright m[c!\langle v \rangle. \mathbf{0}] \rightarrow \overline{\Gamma \triangleright m[\mathbf{0}]}$. These two reductions can be combined together to infer $\Gamma \triangleright \Delta \rightarrow \overline{\Gamma \triangleright m[\mathbf{0}]}$.

However, there exists no extensional action of the form $\Gamma \triangleright \Delta \xrightarrow{\lambda} \overline{\Gamma \triangleright m[\mathbf{0}]}$; in fact, the only possibility for the network $\Gamma \triangleright m[\tau. \mathbf{0}]$ is to perform a τ -extensional action, while the network

$\Gamma \triangleright m[[c!\langle v \rangle]. \mathbf{0}]]$ can only perform an output action of the form $c!v \triangleright \{o\}$. In order to infer the action $\Gamma \triangleright \Delta \xrightarrow{\lambda} \Gamma \triangleright \overline{m[[\mathbf{0}]}}$ we require that every network in $[\Gamma \triangleright \Delta]$ performs the same action λ ; but as we have just noted, this is not true for $\Gamma \triangleright \Delta$. \square

The problem in Example 5.8 arises because reductions have been identified with two different activities of networks; internal actions and broadcasts of messages. However, it is possible to avoid this problem if we modify a network \mathcal{M} by removing the nodes in its interface. As a consequence, the only activities allowed for such a network would be internal actions of the form τ .

Definition 5.9. [Closure of a Network] Let \mathcal{M} be a network; we define its closure $\text{cl}(\mathcal{M}) = \mathcal{M}'$ by letting

$$\begin{aligned} (\mathcal{M}')_V &= \mathcal{M}_V \setminus \text{Intf}(\mathcal{M}) \\ \mathcal{M}' \vdash m, n, \mathcal{M} \vdash m \rightarrow n &\text{ implies } \mathcal{M}' \vdash m \rightarrow n \end{aligned}$$

Obviously the operator $\text{cl}(\cdot)$ preserves well-formed networks.

The actions of a network of the form $\text{cl}(\mathcal{M})$ are completely determined by those performed by \mathcal{M} , as the following result shows.

Proposition 5.10. Suppose that $\mathcal{M} \xrightarrow{\lambda} \Delta$, where either $\lambda = \tau$ or $\lambda = c!v \triangleright \eta$; then $\text{cl}(\mathcal{M}) \xrightarrow{\tau} \text{cl}(\Delta)$.

Conversely, if $\text{cl}(\mathcal{M}) \xrightarrow{\lambda} \Theta$, then $\lambda = \tau$ and either

- (i) $\mathcal{M} \xrightarrow{\tau} \Delta$ and $\Theta = \text{cl}(\Delta)$,
- (ii) $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta$ and $\Theta = \text{cl}(\Delta)$.

Proof. Let $\mathcal{M} = \Gamma \triangleright M$. Suppose that $\Gamma \triangleright M \xrightarrow{c!v \triangleright \eta} \Gamma \triangleright \Delta$ for some Γ, M and $\Delta \in \mathcal{D}(\text{sSys})$. By definition of extensional actions there exists a node $m \in \text{nodes}(M)$ such that $\Gamma \triangleright M \xrightarrow{m.c!v} \Delta$, and $\{o \mid \Gamma \vdash m \rightarrow o\} = \eta$. By Proposition 3.5 it follows that $M \equiv m[[c!e.p + q]] \mid N$, with $[[e]] = v$. By definition of the operator $\text{cl}(\cdot)$, $\text{cl}(\Gamma \triangleright M) = \Gamma' \triangleright M$, where $\Gamma' \vdash m \rightarrow n$ whenever $\Gamma \vdash m \rightarrow n$ and $m, n \in \text{nodes}(M)$. By an application of propositions 3.5 and 3.4 we obtain that $\Gamma' \triangleright M \xrightarrow{m.c!v} \Delta^4$. By Definition 5.1(1) we get that $\Gamma' \triangleright M \xrightarrow{\tau} \Gamma' \triangleright \Delta$. But $\Gamma' \triangleright \Delta$ is exactly $\text{cl}(\Gamma \triangleright \Delta)$. The case $\Gamma \triangleright M \xrightarrow{\tau} \Gamma \triangleright \Delta$ is treated similarly.

Conversely, suppose that $\Gamma' \triangleright M \xrightarrow{\lambda} \Gamma' \triangleright \Theta$. Since $\text{Intf}(\Gamma' \triangleright M) = \emptyset$, it follows that it cannot be $\lambda = m.c!v$, nor $\lambda = c!v \triangleright \eta$. Therefore the only possibility is that $\Gamma' \triangleright M \xrightarrow{\tau} \Gamma' \triangleright \Theta$. By Definition 5.1(1) there are two possible cases.

- (1) $\Gamma' \triangleright M \xrightarrow{m.\tau} \Theta$. In this case we have that $M \equiv m[[\tau.p + q]] \mid N$ and $\Theta \equiv \mathbb{P}(m[[p]]) \mid \overline{N}$. It follows that $\Gamma \vdash M \xrightarrow{m.\tau} \Theta$, hence $\Gamma \triangleright M \xrightarrow{m.\tau} \Gamma \triangleright \Theta$.
- (2) $\Gamma' \triangleright M \xrightarrow{m.c!v} \Theta$; in this case we have that $\Gamma \triangleright M \xrightarrow{m.c!v} \Theta$, since whenever $\Gamma' \vdash m \rightarrow n$ it also follows that $\Gamma \vdash m \rightarrow n$. Let $\eta = \{o \in \text{Out}(\Gamma \triangleright M) \mid \Gamma \vdash m \rightarrow o\}$. If $\eta = \emptyset$, by Definition 5.1(1) we obtain that $\Gamma \triangleright M \xrightarrow{\tau} \Gamma \triangleright \Theta$, otherwise Definition 5.1(2) ensures that $\Gamma \triangleright M \xrightarrow{c!v \triangleright \eta} \Gamma \triangleright \Theta$.

\square

⁴In reality these propositions ensure that $\Gamma' \triangleright M \xrightarrow{m.c!v} \Delta'$, where $\Delta' \equiv \Delta$. However, since the system term M is not changed when performing the operation $\text{cl}(\Gamma \triangleright M)$, it can be proved that $\Delta = \Delta'$.

Thus only τ -actions are allowed in networks of the form $\text{cl}(\mathcal{M})$, and the extensional outputs performed by \mathcal{M} are converted in τ -actions in $\text{cl}(\mathcal{M})$. This relationship can be used to relate the reductions performed by the network \mathcal{M} with the extensional actions of the network $\text{cl}(\mathcal{M})$

Proposition 5.11. Let $\Gamma' \triangleright M = \text{cl}(\Gamma \triangleright M)$; then $\Gamma \triangleright M \rightarrow \Gamma \triangleright \Delta$ if and only if $\Gamma' \triangleright M \rightarrow \Gamma' \triangleright \Delta$.

Proof. Follows immediately from Proposition 5.10 and the definition of reductions. \square

Corollary 5.12. Let $\Gamma' \triangleright M = \text{cl}(\Gamma \triangleright M)$; then $\Gamma \triangleright M \Longrightarrow \Gamma \triangleright \Delta$ if and only if $\Gamma' \triangleright M \Longrightarrow \Gamma' \triangleright \Delta$. Further, $\Gamma \triangleright M \Longrightarrow\!\!\!\Longrightarrow \Gamma \triangleright \Delta$ if and only if $\Gamma' \triangleright M \Longrightarrow\!\!\!\Longrightarrow \Gamma' \triangleright \Delta$. \square

An important consequence of Corollary 5.12 is that the operator $\text{cl}(\cdot)$ does not affect the set of outcomes of a network.

Corollary 5.13. For any network \mathcal{M} , $O(\mathcal{M}) = O(\text{cl}(\mathcal{M}))$.

Proof. Let $\mathcal{M} = \Gamma \triangleright M$, $\text{cl}(\mathcal{M}) = \Gamma' \triangleright M$. Suppose that $p \in O(\mathcal{M})$. Then $\Gamma \triangleright M \Longrightarrow\!\!\!\Longrightarrow \Gamma \triangleright \Delta$, and $\sum_{\mathcal{N} \in [\Delta]} \mathcal{V}(\Delta) = p$. By Corollary 5.12 we have that $\Gamma' \triangleright M \Longrightarrow\!\!\!\Longrightarrow \Gamma' \triangleright \Delta$, hence $p = \sum_{\mathcal{N} \in [\Delta]} \mathcal{V}(\Delta) \in O(\Gamma' \triangleright M)$.

The converse implication is proved analogously. \square

Thus the operator $\text{cl}(\cdot)$ allows to relate weak extensional actions and reductions without affecting the set of outcomes of a network. As we will see in Section 6, this operator is very helpful when exhibiting sound proof methods for the testing preorders.

6. A SOUND PROOF METHOD FOR THE TESTING PREORDERS

In this Section we present the main results of the paper. Following [7] we introduce the notion of simulation between networks $\mathcal{M} \triangleleft_{sim} \mathcal{N}$, and we prove that it is a sound proof technique for the may-testing preorder. This topic is addressed in Section 6.1.

In Section 6.2 we give a similar result for the must testing relation. We introduce the concepts deadlocked network and terminal distributions. These will be used to define a novel coinductive relation for sub-distribution of networks, the *deadlock simulation* \sqsubseteq_{DS} . We prove that the inverse of this relation is sound with respect to the must-testing preorder \sqsubseteq_{must} . Here the use of sub-distributions is necessary, since the must-testing preorder is sensitive to divergence.

Finally, in Section 6.3 we focus on convergent networks; We show that for such networks a slight variation of deadlock simulations can be used as a sound proof method for both the may and must testing preorders.

6.1. The May Case. We begin this section by reviewing the standard definition of simulations for probabilistic systems, applied to our calculus of probabilistic networks.

Definition 6.1. [Simulation preorder] In $\text{pLTS}_{\text{Nets}}$ we let \triangleleft_{sim} denote the largest relation in $\text{Nets} \times \text{Nets}$ such that if $\mathcal{M} \triangleleft_{sim} \mathcal{N}$ then:

- $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$,
- if $\omega(\mathcal{M}) = \text{true}$, then $\mathcal{N} \xrightarrow{\tau} \Theta'$ such that for every $\mathcal{L} \in [\Theta']$, $\omega(\mathcal{L}) = \text{true}$,
- otherwise, whenever $\mathcal{M} \xrightarrow{\lambda} \Delta$, for $\lambda \in \text{Act}_\tau$, then there is a $\Theta \in \mathcal{D}(\text{Nets})$ such that $\mathcal{N} \xrightarrow{\lambda} \Theta$ and $\Delta \overline{(\triangleleft_{sim})^e} \Theta$. \square

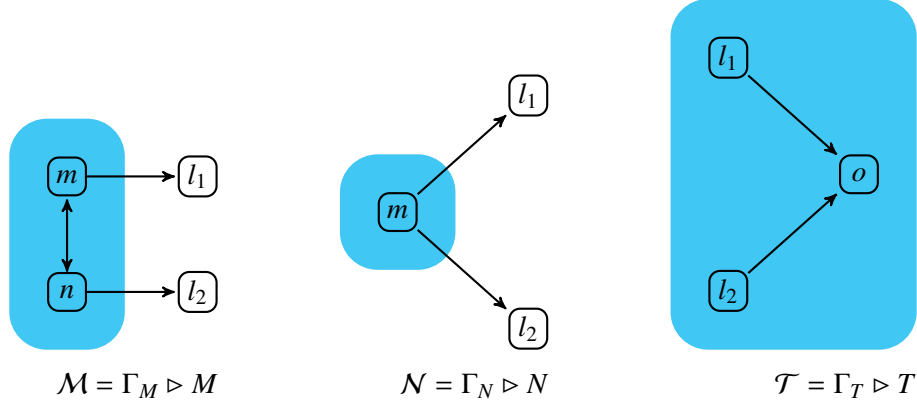


Figure 14: Ensuring soundness

This is a mild generalisation of the corresponding definition in [7] where we factor in the presence of the success predicate $\omega()$ and we compare only networks with the same input and output nodes.

Our aim in this Section is to prove the following theorem:

Theorem 6.2. [Soundness for May-testing] Suppose $\mathcal{N}_1, \mathcal{N}_2$ are finitary networks. Then $\mathcal{N}_1 \triangleleft_{sim} \mathcal{N}_2$ in $\text{pLTS}_{\text{Nets}}$ implies $\mathcal{N}_1 \sqsubseteq_{\text{may}} \mathcal{N}_2$.

Before proving Theorem 6.2, let us review, this time with formal arguments, why our definition of weak extensional actions had to be so complicated, and why we decided to focus on well-formed networks.

Example 6.3. [On Well-formed Networks] Consider again the networks \mathcal{M}, \mathcal{N} of Example 4.1. Here we recall that \mathcal{N} is not well-formed, as it has a connection between the two external nodes o_1, o_2 .

Let $(\triangleleft_{sim})'$ be the largest relation over (possibly non well-formed) networks which satisfies the requirements of Definition 6.1. It is immediate to show that $\mathcal{M}(\triangleleft_{sim})' \mathcal{N}$. However, $\mathcal{M} \not\sqsubseteq_{\text{may}} \mathcal{N}$. In fact, it suffices to consider the test $\mathcal{T} = \Gamma_T \triangleright o_1 \llbracket c?(x) \cdot c!\langle w \rangle \rrbracket \mid o_2 \llbracket c?(x) \cdot c?(y) \cdot \omega \rrbracket$, where $\Gamma_T \vdash o_1, o_2$ and $\Gamma_T \vdash o_1 \rightarrow o_2$ to distinguish these two networks. Specifically $\mathcal{O}(\mathcal{M} \sharp \mathcal{T}) = \{0\}$, $\mathcal{O}(\mathcal{N} \sharp \mathcal{T}) = \{1\}$, and $\{0\} \not\sqsubseteq_H \{1\}$. Therefore, Theorem 6.2 cannot be extended to non well-formed networks. \square

Example 6.4. Consider the networks \mathcal{M} and \mathcal{N} in Figure 8, discussed already in Example 4.15. It

is easy to show that both of them can perform the weak extensional action $\xrightarrow{c!v \triangleright \{o_1, o_2\}}$. However, the inference of the action is different for the individual networks; while in network \mathcal{N} it is implied by the execution of a single broadcast action, detected by both nodes o_1 and o_2 simultaneously, in \mathcal{M} this is implied by a sequence of weak extensional actions $\mathcal{M} \xrightarrow{c!v \triangleright \{o_1\}} \xrightarrow{c!v \triangleright \{o_2\}}$.

It is therefore possible to exhibit a simulation between \mathcal{N} and \mathcal{M} , thus showing that $\mathcal{N} \triangleleft_{sim} \mathcal{M}$; Theorem 6.2 shows that this implies $\mathcal{N} \sqsubseteq_{\text{may}} \mathcal{M}$. \square

Example 6.5. Soundness requires that the extensional output actions records the set of target nodes, rather than single nodes. Consider the networks \mathcal{M}, \mathcal{N} depicted in Figure 14, where $M = m \llbracket c!\langle v \rangle + c?(x) \cdot \mathbf{0} \rrbracket \mid n \llbracket c!\langle v \rangle + c?(x) \cdot \mathbf{0} \rrbracket$ and $N = m \llbracket c!\langle v \rangle \rrbracket$. Intuitively, network \mathcal{M} can broadcast value v to either node l_1 or node l_2 , but it cannot broadcast the message to both nodes. On the other hand, in \mathcal{N} node m can broadcast value v simultaneously to both nodes l_1 and l_2 .

Note that $\mathcal{N} \not\sqsubseteq_{\text{may}} \mathcal{M}$ because of the test \mathcal{T} given in Figure 14, where $T = l_1 \llbracket c?(x) . c!\langle x \rangle \rrbracket \mid l_2 \llbracket c?(x) . c!\langle x \rangle \rrbracket \mid o \llbracket c?(x) . c?(y) . \omega \rrbracket$. In fact, in $(\mathcal{N} \# \mathcal{T})$ both nodes l_1 and l_2 will receive the broadcast of value v along channel c performed by node m ; each of these nodes will forward the received value to node o . Therefore, node o will receive two values, one from node l_1 and one from node l_2 , after which it will reach a successful state. That is, $1 \in \mathcal{O}(\mathcal{N} \# \mathcal{T})$.

On the other hand, none of the computations of $(\mathcal{M} \# \mathcal{T})$ leads to a successful configuration. There are in fact two possibilities; either node m broadcasts value v to nodes n and l_1 , or node n broadcasts value v to nodes m and l_2 . Note that one of the effects of node m (respectively n) broadcasting value v is that of preventing node n (respectively m) from performing a second broadcast. As a consequence, only one among nodes l_1, l_2 will receive value v along channel c . When node l_1 (respectively l_2) receives value v , it will forward it to node o . After this broadcast has been performed the network reaches a configuration in which node o is still waiting to receive a value along channel c before entering a successful state; further, the computation of the network cannot proceed anymore, since none of its nodes can perform a broadcast. There are no other possible behaviours of $\mathcal{M} \# \mathcal{T}$, therefore we obtain $\mathcal{O}(\mathcal{M} \# \mathcal{T}) = \{0\}$.

Since $1 \in \mathcal{O}(\mathcal{N} \# \mathcal{T})$, but $1 \notin \mathcal{O}(\mathcal{M} \# \mathcal{T})$, it follows that $\mathcal{N} \not\sqsubseteq_{\text{may}} \mathcal{M}$.

We also have $\mathcal{N} \not\prec_{\text{sim}} \mathcal{M}$ because \mathcal{N} can perform the output action labelled $c!v \triangleright \{l_1, l_2\}$, which can not be matched by \mathcal{M} .

However suppose we were to restrict η in the definition of extensional output actions, part (3) of Definition 5.1, to be singleton sets of node names. Then in the resulting pLTS it is easy to check that \mathcal{M} can simulate \mathcal{N} . The broadcast of value v in network \mathcal{N} , which can be detected by both nodes l_1 and l_2 , can be matched by either the broadcast performed by node m (which can be detected by node l_1) or by the broadcast performed by node n (which can be detected by node l_2) in \mathcal{M} . In other words, with the proposed simplification the resulting simulations would not be sound; that is, Theorem 6.2 would no longer hold. \square

Let us now turn to the proof of Theorem 6.2; it relies on the following two technical results, whose proofs are developed in Section 7.

Theorem 6.6. [Compositionality] Let \mathcal{M}, \mathcal{N} be finitary networks such that $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$. Also, suppose \mathcal{L} is a network such that both $\mathcal{M} \# \mathcal{L}$ and $\mathcal{N} \# \mathcal{L}$ are defined. Then $\mathcal{M} \prec_{\text{sim}} \mathcal{N}$ implies $(\mathcal{M} \# \mathcal{L}) \prec_{\text{sim}} (\mathcal{N} \# \mathcal{L})$.

Proof. See Corollary 7.4 in Section 7.1 \square

Theorem 6.7. [Outcome preservation] In $\text{pLTS}_{\text{Nets}}$, $\Delta \overleftarrow{\prec}_{\text{sim}} \Theta$ implies $\mathcal{O}(\Delta) \sqsubseteq_H \mathcal{O}(\Theta)$.

Proof. See Corollary 7.11 in Section 7.3. \square

Proof of Theorem 6.2: This is now a straightforward application of Compositionality and Theorem 6.7.

Let us assume that $\mathcal{N}_1 \prec_{\text{sim}} \mathcal{N}_2$. To prove the conclusion, $\mathcal{N}_1 \sqsubseteq_{\text{may}} \mathcal{N}_2$, we must show that $\mathcal{O}(\mathcal{N}_1 \# \mathcal{T}) \sqsubseteq_H \mathcal{O}(\mathcal{N}_2 \# \mathcal{T})$ for an arbitrary testing network \mathcal{T} such that both $\mathcal{N}_1 \# \mathcal{T}$ and $\mathcal{N}_2 \# \mathcal{T}$ are defined. For such a \mathcal{T} Compositionality entails $(\mathcal{N}_1 \# \mathcal{T}) \prec_{\text{sim}} (\mathcal{N}_2 \# \mathcal{T})$, and now we can apply Theorem 6.7. \square

6.2. The Must Case. In this Section we give a sound proof method for the must-testing preorder. It has already been observed that, for standard probabilistic process calculi such as pCSP [7], the must-testing can be characterised by looking at the set of actions which are not enabled in processes. This is because outputs in such a calculus are blocking actions; in order for an action to be performed, a

synchronisation (either within the process or with the external environment) must occur. This leads to the notion of *failure simulations*.

This is not true for broadcast systems, where the nature of the broadcast action is non-blocking. It has been observed in [9] that, if broadcast communication is assumed, then the must-testing relation can be used to observe only if a computation of a process cannot proceed (that is, no internal actions nor broadcasts are possible).

Following this intuition, we readapt the notion of *failure simulation* given in [7].

Definition 6.8 (Deadlocked Networks, Terminal Distributions). The predicate $\delta : \text{Nets} \rightarrow \{\text{true}, \text{false}\}$ is defined by letting $\delta(\mathcal{M}) = \text{true}$ whenever the following conditions are met:

- (i) $\omega(\mathcal{M}) = \text{false}$,
- (ii) $\mathcal{M} \not\stackrel{\tau}{\rightarrow}$,
- (iii) $\mathcal{M} \not\stackrel{c!v \triangleright \eta}{\rightarrow}$ for any c, v, η . □

Networks for which the predicate δ is true are called *deadlock networks*, or *deadlocked*. Note that the term *deadlock network* makes sense only in the reduction semantics. Deadlock networks are those whose computation cannot proceed autonomously; however, it could be the case that an input from the external environment makes the network evolve in a distribution where the computation can proceed, thus resolving the deadlock.

A distribution Δ is said to be *terminal* if any network in its support is either deadlocked or successful.

Next we present a notion of simulation which is sensitive to deadlocked networks:

Definition 6.9 (Deadlock Simulations). The relation $\sqsubseteq_{\text{DS}} \subseteq \text{Nets} \times \mathcal{D}_{\text{sub}}(\text{Nets})$ is the largest relation such that whenever $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$

- (i) if $\delta(\mathcal{M}) = \text{true}$ then $\Theta \stackrel{\tau}{\Longrightarrow} \Theta'$ for some Θ' such that $\delta(\mathcal{N}) = \text{true}$ for any $\mathcal{N} \in \lceil \Theta' \rceil$,
- (ii) if $\mathcal{M} \stackrel{\lambda}{\Longrightarrow} \Delta$ for some $\Delta \in \mathcal{D}_{\text{sub}}(\text{Nets})$ then $\Theta \stackrel{\lambda}{\Longrightarrow} \Theta'$ for some Θ' such that $\Delta \overline{\sqsubseteq}_{\text{DS}} \Theta'$. □

We use the notation $\Theta \supseteq_{\text{DS}} \mathcal{M}$ for $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$. Note that deadlock simulations are sensitive to divergence. That is, whenever $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$ and $\mathcal{M} \stackrel{\tau}{\Longrightarrow} \varepsilon$, then $\Theta \stackrel{\tau}{\Longrightarrow} \varepsilon$. To prove this, note first that for any relation $\mathcal{R} \subseteq \text{Nets} \times \mathcal{D}_{\text{sub}}(\text{Nets})$ then whenever $\Delta \overline{\mathcal{R}} \Theta$ we have that $|\Delta| \geq |\Theta|$; this follows at once from Definition 2.2. Thus if $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$ and $\mathcal{M} \stackrel{\tau}{\Longrightarrow} \varepsilon$, by definition it follows that $\Theta \stackrel{\tau}{\Longrightarrow} \Theta'$ for some Θ' such that $\varepsilon \overline{\sqsubseteq}_{\text{DS}} \Theta'$; but this means that $0 = |\varepsilon| \geq |\Theta'|$, or equivalently $\Theta' = \varepsilon$.

Before discussing the soundness of deadlock simulations for must testing let us discuss briefly the definition of deadlock simulations. First, note that the deadlock simulation relation \sqsubseteq_{DS} is lifted to a relation between sub-distributions, rather than to a relation between (full) distributions. This is needed, since the Must-testing preorder is sensitive to divergence.

Example 6.10. [Divergence] Let $\mathcal{M} = \Gamma \triangleright m[\mathbf{0}]$, $\mathcal{N} = \Gamma \triangleright m[\text{Div}]$, where Γ is the connectivity graph containing the sole node m and no connections and $\text{Div} \Leftarrow \tau.\text{Div}$.

It is immediate to show that $\mathcal{M} \not\sqsubseteq_{\text{DS}} \mathcal{N}$, since the move $\mathcal{N} \stackrel{\tau}{\Longrightarrow} \varepsilon$ cannot be matched by \mathcal{M} .

Even more, it is straightforward to note that $\mathcal{M} \not\sqsubseteq_{\text{must}} \mathcal{N}$. Consider in fact the test $\mathcal{T} = \Gamma_T \triangleright n[\tau.\omega]$, where Γ_T is the connectivity graph consisting of the sole node n . Note that, since $\mathcal{N} \stackrel{\tau}{\Longrightarrow} \varepsilon$ we also have $\mathcal{N} \# \mathcal{T} \stackrel{\tau}{\Longrightarrow} \varepsilon$, and therefore $0 \in \mathcal{O}(\mathcal{N} \# \mathcal{T})$. Intuitively the last hyper-derivation can be inferred by always letting process Div perform a τ -action in $\mathcal{N} \# \mathcal{T}$. On the other hand we have that $\mathcal{O}(\mathcal{M} \# \mathcal{T}) = \{1\}$, since the only possible transition for $\mathcal{M} \# \mathcal{T}$ is $(\mathcal{M} \# \mathcal{T}) \stackrel{\tau}{\rightarrow}$

$\overline{\mathcal{M} \sharp (\Gamma_T \triangleright n[\omega])}$, and the latter is ω -successful. Since $0 \in O(\mathcal{N} \sharp \mathcal{T})$, but $0 \notin O(\overline{\mathcal{M} \sharp \mathcal{T}})$, it follows that $\mathcal{M} \not\sqsubseteq_{\text{must}} \mathcal{N}$.

However, suppose that Definition 6.9 is changed by only considering hyper-derivations of the form $\mathcal{M} \xrightarrow{\lambda} \Delta$, where Δ is a distribution. In this case we would have that the only possible (weak) move for the network \mathcal{N} above is $\mathcal{N} \xrightarrow{\tau} \overline{\mathcal{N}}$, which can be matched by $\mathcal{M} \xrightarrow{\tau} \overline{\mathcal{M}}$. Therefore we would have that $\mathcal{M} \sqsubseteq_{\text{DS}} \mathcal{N}$, and since we already proved that $\mathcal{M} \not\sqsubseteq_{\text{must}} \mathcal{N}$ Theorem 6.14 would no longer hold. \square

Also, deadlock simulation is defined as a relation between networks and sub-distributions of states, rather than a relation between networks. This is in contrast with the definition of simulation given in Section 6.1, which has been defined as a relation between networks. In fact, since deadlock simulation also considers sub-distributions the latter approach would have led to a less discriminating relation.

Remark 6.11. Note that, for any sub-distribution Δ we have that $\Delta \sqsubseteq_{\text{DS}} \varepsilon$; in fact, it is straightforward to show that $\varepsilon \xrightarrow{\lambda} \varepsilon$ for any extensional action λ , and since $[\varepsilon] = \emptyset$ we also have that $\delta(\mathcal{N}) = \text{true}$ for any $\mathcal{N} \in [\varepsilon]$.

Now suppose that deadlock simulation had been defined as a relation between networks, by letting $\mathcal{M} \sqsubseteq_{\text{DS}} \mathcal{N}$ be the largest relation such that

- (i) if $\delta(\mathcal{M}) = \text{true}$ then $\mathcal{N} \xrightarrow{\tau} \Theta$ for some Θ such that $\delta(\mathcal{L}) = \text{true}$ for any $\mathcal{L} \in [\Theta]$,
- (ii) whenever $\mathcal{M} \xrightarrow{\lambda} \Delta$ then $\mathcal{N} \xrightarrow{\lambda} \Theta$ with $\Delta \sqsubseteq_{\text{DS}} \Theta$.

While Theorem 6.14 would still hold with this definition of deadlock simulations, it is straightforward to show that if $\Delta \sqsubseteq_{\text{DS}} \Theta$ then $|\Delta| = |\Theta|$. As a consequence, whenever $\Delta \sqsubseteq_{\text{DS}} \varepsilon$ it would follow that $\Delta = \varepsilon$.

Therefore we have that, for any non-empty sub-distribution Δ , $\Delta \sqsubseteq_{\text{DS}} \varepsilon$, but not $\Delta \sqsubseteq_{\text{DS}} \varepsilon$; that is, the definition of deadlock simulation proposed above is less discriminating than the one given in Definition 6.9.

The proof of soundness of deadlock simulations follows the same structure as the corresponding proof for simulations in Section 6.1. It relies on the following two technical results.

Theorem 6.12. [Outcome preservation] If $\Delta \sqsubseteq_{\text{DS}} \Theta$ then $O(\Delta) \sqsubseteq_S O(\Theta)$.

Proof. See Corollary 7.14 in Section 7.3. \square

Theorem 6.13 (Compositionality). *Let \mathcal{M} be a network and Θ be a stable sub-distribution such that $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$. Then, for any network \mathcal{N} such that both $\mathcal{M} \sharp \mathcal{N}$ and $\Theta \sharp \mathcal{N}$ are defined it follows that $(\mathcal{M} \sharp \mathcal{N}) \sqsubseteq_{\text{DS}} (\Theta \sharp \mathcal{N})$.* \square

Proof. See Corollary 7.8 in Section 7.2. \square

Theorem 6.14. [Soundness for Must-testing] Let \mathcal{M}, \mathcal{N} be two finitary networks such that $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$. If $\overline{\mathcal{M}} \sqsubseteq_{\text{DS}} \mathcal{N}$ then $\mathcal{M} \sqsubseteq_{\text{must}} \mathcal{N}$.

Proof. Suppose that $\mathcal{M} \sqsubseteq_{\text{DS}} \mathcal{N}$, and suppose that $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$. Note also that $\overline{\mathcal{M}}$ is a stable distribution.

Let \mathcal{T} be a network such that both $\mathcal{M} \sharp \mathcal{T}$ and $\mathcal{N} \sharp \mathcal{T}$ are defined. Compositionality, Corollary 7.8 gives that $(\mathcal{M} \sharp \mathcal{T}) \sqsubseteq_{\text{DS}} (\mathcal{N} \sharp \mathcal{T})$, while Theorem 6.12 states that $O(\overline{\mathcal{M} \sharp \mathcal{T}}) \sqsubseteq_S O(\mathcal{N} \sharp \mathcal{T})$. Since the testing network \mathcal{T} has been chosen arbitrarily, it follows that $\mathcal{M} \sqsubseteq_{\text{must}} \mathcal{N}$. \square

6.3. Proof Methods for Convergent Networks. One of the main drawbacks of deadlock simulations is that they require the use of probability sub-distributions. As we have seen in example 6.10, using sub-distributions is necessary for ensuring the validity of Theorem 6.14. We have also emphasised that this constraint is necessary since the must-testing preorder is sensitive to divergence.

However, sub-distributions are no longer needed if we focus on convergent networks, that is those whose generated pLTS (with respect to the strong extensional semantics) does not contain a state \mathcal{M} for which $\mathcal{M} \xrightarrow{\tau} \varepsilon$ holds.

Definition 6.15. [Divergence-free Deadlock Simulations] The relation $\triangleleft_{ds} \subseteq \text{Nets} \times \text{Nets}$ is defined as the largest relation such that whenever $\mathcal{M} \triangleleft_{ds} \mathcal{N}$

- (i) if $\delta(\mathcal{M}) = \text{true}$ then $\mathcal{N} \xrightarrow{\tau} \Theta$ for some Θ such that $\delta(\mathcal{L}) = \text{true}$ whenever $\mathcal{L} \in [\Theta]$,
- (ii) if $\mathcal{M} \xrightarrow{\lambda} \Delta$ then $\mathcal{N} \xrightarrow{\lambda} \Theta$ for some Θ such that $\Delta \triangleleft_{ds}^e \Theta$.

We write $\mathcal{N} \triangleright_{ds} \mathcal{M}$ for $\mathcal{M} \triangleleft_{ds} \mathcal{N}$.

Theorem 6.16. [Soundness for Convergent Networks, Must-testing] Let \mathcal{M}, \mathcal{N} be two convergent networks such that $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$ and $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$; if $\mathcal{M} \triangleright_{ds} \mathcal{N}$ then $\mathcal{M} \sqsubseteq_{\text{must}} \mathcal{N}$.

Proof. It suffices to show that \triangleleft_{ds} is included in \sqsubseteq_{DS} . To this end, note that if $\mathcal{M} \xrightarrow{\lambda} \Delta$ and \mathcal{M} is a convergent network, then $|\Delta| = 1$, by Proposition 2.8. \square

Having a simpler sound proof technique is not the only advantage that we gain by focusing on convergent networks. In fact, if we make a further restriction and we compare networks whose codes running at nodes do not contain the success clause ω , it follows that the relation \triangleleft_{ds} is also included in the may-testing preorder \sqsubseteq_{may} . This restriction is justified since in general we require the tests applied to a network, rather than the networks to be tested, to contain the clause ω to denote the success of an experiment.

Theorem 6.17. [Soundness for Convergent Networks, May-testing] A network $\Gamma \triangleright M$ is proper if the term M does not contain any occurrence of the special clause ω .

Let \mathcal{M}, \mathcal{N} be convergent, proper networks such that $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$. If $\mathcal{M} \triangleleft_{ds} \mathcal{N}$ then $\mathcal{M} \sqsubseteq_{\text{may}} \mathcal{N}$.

Proof. It is trivial to note that the relation \triangleleft_{ds} is included in $\triangleleft_{\text{sim}}$ when restricted to convergent, proper networks. The result follows then from Theorem 6.2. \square

7. TECHNICAL DEVELOPMENT

In this section we collect the proofs of some technical results underlying our soundness results; it may safely be skipped by the uninterested reader. The first to results concern the compositionality of the simulation preorders. The last outlines the proofs about *Outcome preservation*.

7.1. Compositionality for simulations. This section is devoted to the proof of Theorem 6.6, namely that the simulation preorder is preserved by the extension operator \sharp . In general such compositionality results depends on *decomposition* and (*re-*)*composition* results for the actions used in the definition of simulations. Definition 6.1 uses *weak* extensional actions, and providing *decomposition* results for these would be a difficult undertaking. Instead we first give an alternative characterisation of the simulation preorder, for which *decomposition results* for *strong* actions is sufficient.

These (strong) decomposition results, and *(re)-composition* results for *weak* actions have already been given in Section 5.2.

Definition 7.1. [Simple simulations] In $\text{pLTS}_{\text{Nets}}$ we let \triangleleft^s denote the largest relation in $\text{Nets} \times \text{Nets}$ such that if $\mathcal{M} \triangleleft^s \mathcal{N}$ then:

- $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N}), \text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N}),$
 - if $\omega(\mathcal{M}) = \text{true}$ then $\mathcal{N} \xRightarrow{\tau} \Theta$ such that $\omega(\mathcal{L}) = \text{true}$ for any $\mathcal{L} \in [\Theta],$
 - otherwise,
- (i) whenever $\mathcal{M} \xrightarrow{\lambda} \Delta$ there is a $\Theta \in \mathcal{D}(\text{Nets})$ with $\mathcal{N} \xRightarrow{\lambda} \Theta$ and $\Delta \overline{(\triangleleft^s)^e} \Theta.$

Theorem 7.2. [Alternative characterisation] In $\text{pLTS}_{\text{Nets}},$ $\mathcal{M} \triangleleft_{\text{sim}} \mathcal{N}$ if and only if $\mathcal{M} \triangleleft^s \mathcal{N},$ provided that \mathcal{M} and \mathcal{N} are finitary networks.

Proof. (Outline) The proof is similar in style to the one of Theorem 7.20 of [7]; however, there are some extra complications, mainly because of the more complicated definition of weak extensional actions. Here we report a detailed outline of the proof; we first prove that $\triangleleft_{\text{sim}}$ is included in $\triangleleft^s,$ then we show that, for finitary networks, the converse inclusion also holds.

Showing that the relation $\triangleleft_{\text{sim}}$ is included in \triangleleft^s is straightforward. We only need to show that $\triangleleft_{\text{sim}}$ satisfies the constraints of Definition 7.1. Suppose that $\mathcal{M} \triangleleft_{\text{sim}} \mathcal{N}.$ Then this hypothesis ensures that

- $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N}), \text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$ and
- if $\omega(\mathcal{M}) = \text{true}$ then $\mathcal{N} \xRightarrow{\tau} \Theta$ such that $\omega(\mathcal{L}) = \text{true}$ for any $\mathcal{L} \in [\Theta].$

Suppose however that $\omega(\mathcal{M}) = \text{false}$ and $\mathcal{M} \xrightarrow{\lambda} \Delta$ for some $\Delta.$ Then we also have that $\mathcal{M} \xRightarrow{\lambda} \Delta,$ from which it follows from the hypothesis $\mathcal{M} \triangleleft_{\text{sim}} \mathcal{N}$ that there exists Θ such that $\mathcal{N} \xRightarrow{\tau} \Theta$ and $\Delta \overline{(\triangleleft_{\text{sim}})^e} \Theta,$ which is exactly what we wanted to show.

It remains to show that, for finitary networks, the relation \triangleleft^s is included in $\triangleleft_{\text{sim}}.$ Here the main difficulty consists in showing that, whenever $\mathcal{M} \triangleleft^s \mathcal{N}, \omega(\mathcal{M}) = \text{false}$ and $\mathcal{M} \xRightarrow{\lambda} \Delta,$ then $\mathcal{N} \xRightarrow{\lambda} \Theta$ for some Θ such that $\Delta \overline{(\triangleleft^s)^e} \Theta.$ The proof of this statement is performed by a case analysis on the action $\lambda;$

- (1) First suppose that $\lambda = \tau.$ This case can be proved in the analogous way of Theorem 7.20 of [7]. Note that we require networks to be finitary in this case, since the proof requires properties of hyper-derivations which in general are not satisfied by infinitary pLTSs; see Lemma 6.12 of [7].
- (2) Now suppose that $\lambda = i.c?v$ for some $i, c, v;$ then $\mathcal{M} \xRightarrow{i.c?v} \Delta$ implies that there exist Δ', Δ'' such that $\mathcal{M} \xRightarrow{\tau} \Delta' \xrightarrow{i.c?v} \Delta'' \xRightarrow{\tau} \Delta.$ Since $\mathcal{M} \triangleleft^s \mathcal{N}$ and $\mathcal{M} \xRightarrow{\tau} \Delta',$ by the previous case $\mathcal{N} \xRightarrow{\tau} \Theta'$ for some Θ' such that $\Delta' \overline{(\triangleleft^s)^e} \Theta'.$ Since $\Delta' \xrightarrow{i.c?v} \Delta'',$ we can conclude that $\Theta' \xRightarrow{i.c?v} \Theta''$ for some Θ'' such that $\Delta'' \overline{(\triangleleft^s)^e} \Theta''.$ Finally, since $\Delta'' \xRightarrow{\tau} \Delta,$ by the previous case⁵ we obtain that $\Theta'' \xRightarrow{\tau} \Theta$ for some Θ such that $\Delta \overline{(\triangleleft^s)^e} \Theta.$ Therefore we have shown that $\mathcal{N} \xRightarrow{\tau} \Theta' \xrightarrow{i.c?v} \Theta'' \xRightarrow{\tau} \Theta,$ or equivalently $\mathcal{N} \xRightarrow{i.c?v} \Theta,$ and $\Delta \overline{(\triangleleft^s)^e} \Theta,$ which is exactly what we needed to prove.

⁵Note that here it is necessary to decompose Δ'' as a sum of state-based networks, each of which can perform a weak τ -action.

(3) Finally, suppose that $\lambda = c!v \triangleright \eta$ for some c, v and non-empty set of nodes η . We perform an inner induction on the proof of the derivation $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta$;

- $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta$ because $\mathcal{M} \xrightarrow{\tau} \xrightarrow{c!v \triangleright \eta} \xrightarrow{\tau} \Delta$. This case is identical to the one $\lambda = i.c?v$,
- $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta$ because $\mathcal{M} \xrightarrow{c!v \triangleright \eta_1} \Delta' \xrightarrow{c!v \triangleright \eta_2} \Delta$, where $\eta_1 \cup \eta_2 = \eta$, $\eta_1 \cap \eta_2 = \emptyset$. Since $\mathcal{M} \triangleleft^s \mathcal{N}$, by the (inner) inductive hypothesis we have that $\mathcal{N} \xrightarrow{c!v \triangleright \eta_1} \Theta'$ for some Θ' such that $\Delta' \overline{(\triangleleft^s)^e} \Theta'$. A second application of the inductive hypothesis to the last statement gives that $\Theta' \xrightarrow{c!v \triangleright \eta_2} \Theta$ for some Θ such that $\Delta \overline{(\triangleleft^s)^e} \Theta$. Therefore we have shown that $\mathcal{N} \xrightarrow{c!v \triangleright \eta_1} \Theta' \xrightarrow{c!v \triangleright \eta_2} \Theta$, or equivalently $\mathcal{N} \xrightarrow{c!v \triangleright \eta} \Theta$ (recall that $\eta_1 \cup \eta_2 = \eta$ and $\eta_1 \cap \eta_2 = \emptyset$), and $\Delta \overline{(\triangleleft^s)^e} \Theta$, as we wanted to prove. \square

Theorem 7.2 enables us to exploit the results developed in Section 5.2 for proving the compositionality of \triangleleft_{sim} with respect to the extension operator \sharp . Since such results are valid only in the case that a network \mathcal{M} is composed with a generating network \mathcal{G} , we first focus on compositionality with respect to a generating network.

Theorem 7.3. Suppose $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N})$, $\text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$ and both $\mathcal{M} \sharp \mathcal{G}$ and $\mathcal{N} \sharp \mathcal{G}$ are defined. Then $\mathcal{M} \triangleleft^s \mathcal{N}$ implies $\mathcal{M} \sharp \mathcal{G} \triangleleft^s \mathcal{N} \sharp \mathcal{G}$.

Proof. It suffices to show that the relation

$$\mathcal{S} = \{(\mathcal{M} \sharp \mathcal{G}, \mathcal{N} \sharp \mathcal{G}) \mid \mathcal{M} \triangleleft^s \mathcal{N}\}$$

satisfies the requirements of Definition 7.1. We denote the network \mathcal{G} with $\Gamma_G \triangleright n[[s]]$.

Suppose that $\mathcal{M} \sharp \mathcal{G} \xrightarrow{c!v \triangleright \eta} \Delta$; note that the definition of extensional output ensures that $\eta \neq \emptyset$. We need to show that $\mathcal{N} \sharp \mathcal{G} \xrightarrow{c!v \triangleright \eta} \Theta$, and $\Delta \overline{\mathcal{S}^e} \Theta$. By Proposition 5.6 we know that $\Delta = \Delta_M \sharp (\Gamma_G \triangleright n[[\Theta_n]])$, where Δ_M, Θ_n are determined according to six different cases, which are considered below.

- (i) $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta_M$, $n \notin \eta$ and $\Theta_n = \bar{s}$; since $\mathcal{M} \triangleleft^s \mathcal{N}$ it follows that $\mathcal{N} \xrightarrow{c!v \triangleright \eta} \Theta_N$ for some Θ_N such that $\Delta_M \overline{(\triangleleft^s)^e} \Theta_N$. Since $n \notin \eta$, By Proposition 5.7 it follows that $\mathcal{N} \sharp (\Gamma_G \triangleright n[[s]]) \xrightarrow{c!v \triangleright \eta} \Theta_N \sharp (\Gamma_G \triangleright n[[\bar{s}]])$. Now it suffices to note that $(\Delta_M \sharp \Gamma_G \triangleright n[[\bar{s}]]) \overline{(\mathcal{S})^e} (\Delta_N \sharp \Gamma_G \triangleright n[[\bar{s}]])$, as we wanted to prove
- (ii) $\mathcal{M} \xrightarrow{c!v \triangleright \eta \cup \{n\}} \Delta_M$, $n \notin \eta$ and $s \xrightarrow{c?v} \Theta_n$. Since $\mathcal{M} \triangleleft^s \mathcal{N}$ it follows that $\mathcal{N} \xrightarrow{c!v \triangleright \eta \cup \{n\}} \Theta_N$, where $\Delta_M \overline{(\triangleleft^s)^e} \Theta_N$. Since $\eta \neq \emptyset$, that is $\{n\} \subset (\eta \cup \{n\})$, we can apply Proposition 5.7(2)(ii) to the weak extensional transition $\mathcal{N} \xrightarrow{c!v \triangleright \eta \cup \{n\}} \Theta_N$ and the process transition $s \xrightarrow{c?v} \Theta_n$ to infer $\mathcal{N} \sharp (\Gamma_G \triangleright n[[s]]) \xrightarrow{c!v \triangleright (\eta \cup \{n\}) \setminus \{n\}} \Theta_N \sharp (\Gamma_G \triangleright n[[\Theta_n]])$; since we are assuming that $n \notin \eta$, the latter can be rewritten as $\mathcal{N} \sharp (\Gamma_G \triangleright n[[s]]) \xrightarrow{c!v \triangleright \eta} \Theta_N \sharp (\Gamma_G \triangleright n[[\Theta_n]])$; again, since $\Delta_M \overline{(\triangleleft^s)^e} \Theta_N$, we have that $(\Delta_M \sharp \Gamma_G \triangleright n[[\Theta_n]]) \overline{\mathcal{S}^e} \Theta_N \sharp (\Gamma_G \triangleright n[[\Theta_n]])$
- (iii) $\mathcal{M} \xrightarrow{c!v \triangleright \eta \cup \{n\}} \Delta_M$, $s \xrightarrow{c?v}$ and $\Theta_n = \bar{s}$. This case is similar to the previous one, this time employing Proposition 5.7(2)(iii)

- (iv) $\mathcal{M} \xrightarrow{m.c?v} \Delta_M, s \xrightarrow{c!v} \Theta_n$ and $\eta = \text{Out}((\Gamma_G \triangleright n[s]))$. Since $\mathcal{M} \triangleleft^s \mathcal{N}$ we have that $\mathcal{N} \xrightarrow{m.c?v} \Theta_N$ for some Θ_N such that $\Delta_M \overline{(\triangleleft^s)^e} \Theta_N$. It follows from Proposition 5.7(2)(iv) that $\mathcal{N} \# (\Gamma_G \triangleright n[s]) \xrightarrow{c!v \triangleright \eta} \Theta_N \# (\Gamma_G \triangleright n[\Theta_n])$. Now note that $(\Delta_M \# \Gamma_G \triangleright n[\Theta_n]) \overline{(\mathcal{S})^e} (\Theta_N \# \Gamma_G \triangleright n[\Theta_n])$
- (v) $m \notin \text{In}(\mathcal{M}), \Delta = \overline{\mathcal{M}}$ and $s \xrightarrow{c!v} \Theta_n$. Since $\mathcal{M} \triangleleft_{sim} \mathcal{N}$ it follows that $m \notin \text{In}(\mathcal{N})$, hence by Proposition 5.7 we have that $\mathcal{N} \# (\Gamma_G \triangleright n[s]) \xrightarrow{m.c?v} \overline{\mathcal{N}} \# (\Gamma_G \triangleright n[\Theta_n])$. Now the hypothesis $\mathcal{M} \triangleleft^s \mathcal{N}$ ensures that $\overline{\mathcal{M}} \# (\Gamma_G \triangleright n[\Theta_n]) \text{lift}(\mathcal{S})^e \overline{\mathcal{N}} \# (\Gamma_G \triangleright n[\Theta_n])$.

The cases $(\mathcal{M} \# \mathcal{G}) \xrightarrow{\tau}$ and $(\mathcal{M} \# \mathcal{G}) \xrightarrow{i.c?v}$ are treated similarly, and are therefore left as an exercise for the reader. \square

However, since generating networks can be used to generate all the networks in **Nets**, Proposition 4.6, we can easily generalise Theorem 7.3 to arbitrary networks.

Corollary 7.4. [Compositionality: Theorem 6.6] Let \mathcal{M}, \mathcal{N} be finitary networks such that $\text{In}(\mathcal{M}) = \text{In}(\mathcal{N}), \text{Out}(\mathcal{M}) = \text{Out}(\mathcal{N})$. Also, suppose \mathcal{L} is a network such that both $\mathcal{M} \# \mathcal{L}$ and $\mathcal{N} \# \mathcal{L}$ are defined. Then $\mathcal{M} \triangleleft_{sim} \mathcal{N}$ implies $(\mathcal{M} \# \mathcal{L}) \triangleleft_{sim} (\mathcal{N} \# \mathcal{L})$.

Proof. First note that the only elements used to define simulations are network interfaces, the set of outcomes of networks and the extensional transitions that networks can perform. These definitions are preserved by the structural congruence between networks defined on page 13. As a consequence, simulations are identified up-to structural congruence; if $\mathcal{M} \equiv \mathcal{M}', \mathcal{M}' \triangleleft_{sim} \mathcal{N}'$ and $\mathcal{N}' \equiv \mathcal{N}$, it follows that $\mathcal{M} \triangleleft_{sim} \mathcal{N}$.

Suppose then that $\mathcal{M} \triangleleft_{sim} \mathcal{N}$, and let \mathcal{L} be a network such that both $(\mathcal{M} \# \mathcal{L})$ and $(\mathcal{N} \# \mathcal{L})$ are defined. We show that $(\mathcal{M} \# \mathcal{L}) \triangleleft_{sim} (\mathcal{N} \# \mathcal{L})$ by induction on $\text{nodes}(\mathcal{L})$.

- $\text{nodes}(\mathcal{L}) = 0$. In this case we have that

$$(\mathcal{M} \# \mathcal{L}) \equiv \mathcal{M} \triangleleft_{sim} \mathcal{N} \equiv (\mathcal{N} \# \mathcal{L}),$$

- $\text{nodes}(\mathcal{L}) > 0$. By Proposition 4.6 there exist two networks \mathcal{L}' and \mathcal{G} such that $\mathcal{L} \equiv (\mathcal{L}' \# \mathcal{G})$. In particular $(\mathcal{M} \# \mathcal{L}) \equiv \mathcal{M} \# (\mathcal{L}' \# \mathcal{G}) = (\mathcal{M} \# \mathcal{L}') \# \mathcal{G}$, where we have used the associativity of the operator $\#$, Proposition 4.4. Note that $\text{nodes}(\mathcal{L}') < \text{nodes}(\mathcal{L})$, hence by the inductive hypothesis it follows that $(\mathcal{M} \# \mathcal{L}') \triangleleft_{sim} (\mathcal{N} \# \mathcal{L}')$. By Theorem 7.2 we obtain that $(\mathcal{M} \# \mathcal{L}') \triangleleft^s (\mathcal{N} \# \mathcal{L}')$, and Theorem 7.3 gives that $((\mathcal{M} \# \mathcal{L}') \# \mathcal{G}) \triangleleft^s ((\mathcal{N} \# \mathcal{L}') \# \mathcal{G})$. Another application of 7.2 and the associativity of the operator $\#$ lead to $(\mathcal{M} \# (\mathcal{L}' \# \mathcal{G})) \triangleleft_{sim} (\mathcal{N} \# (\mathcal{L}' \# \mathcal{G}))$. Therefore we have that

$$(\mathcal{M} \# \mathcal{L}) \equiv (\mathcal{M} \# (\mathcal{L}' \# \mathcal{G})) \triangleleft_{sim} (\mathcal{N} \# (\mathcal{L}' \# \mathcal{G})) \equiv (\mathcal{N} \# \mathcal{L})$$

as we wanted to prove. \square

7.2. Compositionality for deadlock simulations. As we have already done in Section 7.1, here we also rely on an alternative characterisation of the \sqsubseteq_{DS} preorder, which is more amenable to *decomposition/composition* results.

Definition 7.5 (Simple Deadlock Simulation). Let $\sqsubseteq_{DS}^s \subseteq \text{Nets} \times \mathcal{D}_{sub}(\text{Nets})$ be the largest relation such that whenever $\mathcal{M} \sqsubseteq_{DS}^s \Theta$ then $\text{In}(\mathcal{M}) = \text{In}(\Theta), \text{Out}(\mathcal{M}) = \text{Out}(\Theta)$ for any $\Theta \in [\Theta]$; further

- if $\mathcal{M} \Longrightarrow \varepsilon$ then $\Theta \Longrightarrow \varepsilon$,
- otherwise,

- (i) if $\delta(\mathcal{M}) = \text{true}$ then $\Theta \Longrightarrow \Theta'$ such that $\delta(\mathcal{N}) = \text{true}$ for any $\mathcal{N} \in [\Theta']$,
- (ii) if $\mathcal{M} \xrightarrow{\lambda} \Delta'$, then $\Theta \xrightarrow{\lambda} \Theta'$ and $\Delta' \overline{\sqsubseteq}_{\text{DS}}^s \Theta'$.

Theorem 7.6. [Alternative Characterisation of Deadlock Simulations] If \mathcal{M} is a finitary network and Θ is a finitary sub-distribution of networks then $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$ if and only if $\mathcal{M} \overline{\sqsubseteq}_{\text{DS}}^s \Theta$.

Proof. The proof is identical to that of Theorem 7.2. \square

Theorem 7.7. [Single Node Compositionality for Deadlock Simulations] Let \mathcal{M} be a network and Θ be a stable sub-distribution such that $\mathcal{M} \overline{\sqsubseteq}_{\text{DS}}^s \Theta$. Then, for any network \mathcal{G} such that $\mathcal{M} \sharp \mathcal{G}$ and $\Theta \sharp \overline{\mathcal{G}}$ are well-defined it follows that $(\mathcal{M} \sharp \mathcal{G}) \sqsubseteq_{\text{DS}} (\Theta \sharp \overline{\mathcal{G}})$.

Proof. The proof is analogous to that of Theorem 7.3. There are, however, some extra statements that we need to check.

- If $\mathcal{M} \sharp \mathcal{G} \xrightarrow{\tau} \varepsilon$ then $\mathcal{N} \sharp \mathcal{G} \xrightarrow{\tau} \varepsilon$. This can be proved by rewriting by inspecting the infinite sequence of τ -moves which constitute the hyper-derivation $\mathcal{M} \sharp \mathcal{G} \xrightarrow{\tau} \varepsilon$ to build an hyper-derivation of the form $\mathcal{N} \sharp \mathcal{G} \xrightarrow{\tau} \varepsilon$. This step requires employing the composition and decomposition results developed in Section 5.2.
- For any $\mathcal{M} \in \text{Nets}, \mathcal{G} \in \mathbb{G}$ such that $\mathcal{M} \sharp \mathcal{G}$ is defined, $\delta(\mathcal{M} \sharp \mathcal{G}) = \text{true}$ implies $\delta(\mathcal{M}) = \text{true}$ and $\delta(\mathcal{G}) = \text{true}$,
- For any stable sub-distribution Δ and network $\mathcal{G} \in \mathbb{G}$, suppose that $\Delta \xrightarrow{\tau} \Delta'$, where Δ' is such that $\delta(\mathcal{M}) = \text{true}$ for any $\mathcal{M} \in [\Delta']$; further, suppose that $\mathcal{G} \xrightarrow{\tau} \Theta$ for some Θ such that $\delta(\mathcal{N}) = \text{true}$ for any $\mathcal{N} \in [\Theta]$. Then $\Delta \sharp \overline{\mathcal{G}} \xrightarrow{\tau} \Lambda$, and $\delta(\mathcal{L}) = \text{true}$ for any $\mathcal{L} \in [\Lambda]$.

The validity of the two statements above can be easily proved using the results developed in Section 5.2. \square

Corollary 7.8. [Theorem 6.13] Let \mathcal{M} be a network and Θ be a stable sub-distribution such that $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$. Then, for any network \mathcal{N} such that both $\mathcal{M} \sharp \mathcal{N}$ and $\Theta \sharp \mathcal{N}$ are defined it follows that $(\mathcal{M} \sharp \mathcal{N}) \sqsubseteq_{\text{DS}} (\Theta \sharp \mathcal{N})$. \square

Proof. By induction on the number of nodes contained in $\text{nodes}(\mathcal{N})$, using Theorem 7.6. The proof is analogous to that of Corollary 7.4. \square

7.3. Outcome preservation. The aim of this section is to prove Theorem 6.7 and Theorem 6.12, namely that simulations preserve, in the sense of Definition 4.12, the outcomes generated by networks. To this end, we first need two technical results.

Lemma 7.9. If $\mathcal{M} \triangleleft_{\text{sim}} \mathcal{N}$, then $\text{cl}(\mathcal{M}) \triangleleft_{\text{sim}} \text{cl}(\mathcal{N})$.

Proof. It suffices to show that the relation

$$\mathcal{S} = \{(\text{cl}(\mathcal{M}), \text{cl}(\mathcal{N})) \mid \mathcal{M} \triangleleft_{\text{sim}} \mathcal{N}\}$$

satisfies the requirements of Definition 6.1. Let \mathcal{M}, \mathcal{N} such that $\mathcal{M} \triangleleft_{\text{sim}} \mathcal{N}$. Since the only possibility for networks of the form $\text{cl}(\mathcal{M}), \text{cl}(\mathcal{N})$ is that of performing τ -extensional actions, we only have to check that if $\omega(\text{cl}(\mathcal{M})) = \text{true}$ then $\text{cl}(\mathcal{N}) \xrightarrow{\tau} \Theta$ for some Θ such that $\omega(\mathcal{L}) = \text{true}$ for any $\mathcal{L} \in [\Theta]$.

Suppose then that $\omega(\text{cl}(\mathcal{M})) = \text{true}$. It follows immediately that $\omega(\mathcal{M}) = \text{true}$, and since by hypothesis $\mathcal{M} \triangleleft_{sim} \mathcal{N}$, we have that $\mathcal{N} \xrightarrow{\tau} \Theta'$ for some Θ' such that $\omega(\mathcal{L}') = \text{true}$ for any $\mathcal{L}' \in [\Theta']$. Therefore we have that $\text{cl}(\mathcal{N}) \xrightarrow{\tau} \text{cl}(\Theta')$. It follows that $[\text{cl}(\Theta')] = \{\text{cl}(\mathcal{L}') \mid \mathcal{L}' \in [\Theta']\}$, hence any network $\mathcal{L} \in [\text{cl}(\Theta')]$ satisfies the predicate ω . \square

Lemma 7.10. Let Δ, Θ be stable distributions in $\text{pLTS}_{\text{Nets}}$ such that $\Delta \overline{(\triangleleft_{sim})^e} \Theta$; then $\Theta \Longrightarrow \Theta'$ such that $\mathcal{V}(\Delta) \leq \mathcal{V}(\Theta')$.

Proof. Note that in the statement of Lemma 7.10 we use the extreme derivative associated with the reduction relation \rightarrow defined for the testing structures associated with networks. We have seen in Section 5.3 that such extreme derivatives do not coincide with weak τ -actions in the extensional semantics, except in the case of closed distributions.

Therefore, let us first prove the statement for closed distributions, that is those whose networks in the support have an empty interface. Let then Δ, Θ be such that $\text{Intf}(\Delta) = \text{Intf}(\Theta) = \emptyset$.

We have two different cases.

- (i) First suppose Δ is a point distribution $\overline{\mathcal{M}}$. If the predicate $\omega(\mathcal{M})$ is equal to **false**, $\mathcal{V}(\overline{\mathcal{M}}) = 0$. In this case, we recall that Theorem 2.6 (4) ensures that there exists at least one extreme derivation $\Theta \Longrightarrow \Theta'$, for which $0 \leq \mathcal{V}(\Theta')$ trivially holds.

Otherwise the predicate $\omega(\mathcal{M})$ is satisfied and $\mathcal{V}(\overline{\mathcal{M}})$ has to be 1. Since $\overline{\mathcal{M}} \overline{(\triangleleft_{sim})^e} \Theta$ we know $\Theta \xrightarrow{\tau} \Theta'$, such that for all $\mathcal{N} \in \Theta'$, $\omega(\mathcal{N}) = \text{true}$. Also, since Θ is closed, the hyper-derivation above can be rewritten as $\Theta \Longrightarrow \Theta'$. This means that $\mathcal{V}(\Theta') = 1$; moreover, as every state in $[\Theta']$ is a successful state, we also have that $\Theta \Longrightarrow \Theta'$.

- (ii) Otherwise Θ can be written as $\sum_{\mathcal{M} \in [\Delta]} \Delta(\mathcal{M}) \cdot \Theta_{\mathcal{M}}$ where $\mathcal{M} \overline{(\triangleleft_{sim})^e} \Theta_{\mathcal{M}}$ for each \mathcal{M} in the support of Δ . By part (i) each $\Theta_{\mathcal{M}} \Longrightarrow \Theta'_{\mathcal{M}}$ such that $\mathcal{V}(\overline{\mathcal{M}}) \leq \mathcal{V}(\Theta'_{\mathcal{M}})$. As an extreme derivative is also a hyper-derivative, we can combine these to obtain a hyper derivation for Θ , using Theorem 2.6 (3). This leads to

$$\Theta = \sum_{\mathcal{M} \in [\Delta]} \Delta(\mathcal{M}) \cdot \Theta_{\mathcal{M}} \Longrightarrow \sum_{\mathcal{M} \in [\Delta]} \Theta'_{\mathcal{M}} = \Theta'$$

As for every $\mathcal{M} \in [\Delta]$, $\mathcal{N} \in [\Theta'_{\mathcal{M}}]$ we have that $\mathcal{N} \rightarrow$ implies $\omega \mathcal{N} = \text{true}$, this condition is respected also by all states in $[\Theta']$. Thus, the hyper derivation $\Theta \Longrightarrow \Theta'$ is also an extreme derivation. Finally, the quantity $\mathcal{V}(\Delta) = \sum \{\Delta(\mathcal{M}) \mid \omega(\mathcal{M}) = \text{true}\}$ can be rewritten as $\sum_{\mathcal{M} \in [\Delta]} \mathcal{V}(\overline{\mathcal{M}})$, leading to

$$\mathcal{V}(\Delta) = \sum_{\mathcal{M} \in [\Delta]} \mathcal{V}(\overline{\mathcal{M}}) \leq \sum_{\mathcal{M} \in [\Delta]} \mathcal{V}(\Theta'_{\mathcal{M}}) = \mathcal{V}(\Theta').$$

For more general distributions Δ, Θ it suffices to note that if $\Delta \overline{(\triangleleft_{sim})^e} \Theta$ then $\text{cl}(\Delta) \overline{(\triangleleft_{sim})^e} \text{cl}(\Theta)$. For such (closed) distributions of networks we have that $\text{cl}(\Theta) \Longrightarrow \Theta''$, and $\mathcal{V}(\text{cl}(\Delta)) \leq \mathcal{V}(\Theta'')$. Now Corollary 5.12 ensures that $\Theta'' = \text{cl}(\Theta')$ for some Θ' such that $\Theta \Longrightarrow \Theta'$. Finally, by Corollary 5.13 we obtain that

$$\mathcal{V}(\Delta) = \mathcal{V}(\text{cl}(\Delta)) \leq \mathcal{V}(\text{cl}(\Theta')) = \mathcal{V}(\Theta')$$

which concludes the proof. \square

Corollary 7.11 (Theorem 6.7). *In $\text{pLTS}_{\text{Nets}}$, $\Delta \overleftarrow{\text{sim}} \Theta$ implies $O(\Delta) \sqsubseteq_H O(\Theta)$.*

Proof. We first prove the result for closed distributions Δ, Θ . Suppose $\Delta \Longrightarrow \Delta'$. We have to find a derivation $\Theta \Longrightarrow \Theta'$ such that $\mathcal{V}(\Delta') \leq \mathcal{V}(\Theta')$. Since we are assuming that Δ is closed, then $\Delta \Longrightarrow \Delta'$ implies $\Delta \xRightarrow{\tau} \Delta'$, Corollary 5.12, which in turn gives $\Delta \xRightarrow{\tau} \Delta'$. We can use the definition of $\overleftarrow{\text{sim}}$ to find a derivation $\Theta \xRightarrow{\tau} \Theta''$ such that $\Delta' \overleftarrow{\text{sim}}^e \Theta''$. Applying the previous lemma we obtain $\Theta'' \xRightarrow{\tau} \Theta'$ such that $\mathcal{V}(\Delta') \leq \mathcal{V}(\Theta')$. By Theorem 2.6 we have that $\Theta \xRightarrow{\tau} \Theta'$, and Corollary 5.12 gives $\Theta \Longrightarrow \Theta'$.

Suppose now that Δ, Θ are not closed distributions. In this case Corollary 5.13 ensures that

$$O(\Delta) = O(\text{cl}(\Delta)) \sqsubseteq_H O(\text{cl}(\Theta)) = O(\Theta)$$

and there is nothing left to prove. \square

We now repeat the above argument to show that outcomes are also preserved by deadlock simulations; the details are quite similar.

Lemma 7.12. Whenever $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$ it follows that $\text{cl}(\mathcal{M}) \sqsubseteq_{\text{DS}} \text{cl}(\Theta)$.

Proof. It suffices to note that, for any network \mathcal{M} , $\delta(\mathcal{M}) = \text{true}$ if and only if $\delta(\text{cl}(\mathcal{M})) = \text{true}$. Using this fact the result can be proved as in Lemma 7.9. \square

The main use of Lemma 7.12 is that of showing that whenever $\Delta \sqsubseteq_{\text{DS}} \Theta$ then the sets of outcomes of Δ and Θ are related in some appropriate manner.

Lemma 7.13. *Suppose that $\Delta \overline{\sqsubseteq}_{\text{DS}} \Theta$ for some terminal distribution Δ . Then $\Theta \Longrightarrow \Theta'$ for some Θ' such that $\mathcal{V}(\Theta') \leq \mathcal{V}(\Delta)$.*

Proof. First suppose that Δ is a closed distribution, that is for any network $\mathcal{M} \in [\Delta]$ we have that $\text{Intf}(\Delta) = \emptyset$.

- (1) if $\Delta = \overline{\mathcal{M}}$ we have two possible cases:
 - (i) $\delta(\mathcal{M}) = \text{false}$; since we are assuming that $\overline{\mathcal{M}}$ is a terminal distribution then it has to be $\omega\mathcal{M} = \text{true}$, which implies $\mathcal{V}(\overline{\mathcal{M}}) = 1$. In this case we are ensured that $\Theta \Longrightarrow \Theta'$ for some Θ' , and $\mathcal{V}(\Theta') \leq 1$, trivially holds.
 - (ii) $\delta(\mathcal{M}) = \text{true}$; in this case $\omega(\mathcal{M}) = \text{false}$, hence $\mathcal{V}(\overline{\mathcal{M}}) = 0$. Therefore we have to show that $\Theta \Longrightarrow \Theta'$ for some Θ' such that $\mathcal{V}(\Theta') = 0$. Since $\mathcal{M} \sqsubseteq_{\text{DS}} \Theta$, we have that $\Theta \xRightarrow{\tau} \Theta'$ for some Θ' such that $\delta(\mathcal{N}) = \text{true}$ for any $\mathcal{N} \in [\Theta']$, which is equivalent to $\mathcal{V}(\Theta') = 0$. Further, $\Theta \xRightarrow{\tau} \Theta'$ implies $\Theta \Longrightarrow \Theta'$. Since $\mathcal{N} \xrightarrow{\tau} \mathcal{N}$, $\mathcal{N} \xrightarrow{c!v \triangleright \eta} \mathcal{N}$ for any $\mathcal{N} \in [\Theta']$, which is equivalent to $\mathcal{N} \not\rightarrow$ for any $\mathcal{N} \in [\Theta']$, we also have $\Theta \Longrightarrow \Theta'$.
- (2) otherwise, $\Delta = \sum_{i \in I} p_i \cdot \overline{\mathcal{M}}_i$, $\Theta = \sum_{i \in I} p_i \cdot \Theta_i$ with $\sum_{i \in I} p_i \leq 1$ and $\mathcal{M}_i \sqsubseteq_{\text{DS}} \Theta_i$ for any $i \in I$. This part of the Lemma can be proved as in Lemma 7.10.

Now let Δ be a general distribution, not necessarily closed. By Lemma 7.12 it follows that $\text{cl}(\Delta) \sqsubseteq_{\text{DS}} \text{cl}(\Theta)$, therefore there exists a sub-distribution Θ' such that $\text{cl}(\Theta) \Longrightarrow \text{cl}(\Theta')$, and $\mathcal{V}(\text{cl}(\Theta')) \leq \mathcal{V}(\text{cl}(\Delta))$. This in turn implies that $\Theta \Longrightarrow \Theta'$, and

$$\mathcal{V}(\Theta') = \mathcal{V}(\text{cl}(\Theta')) \leq \mathcal{V}(\text{cl}(\Delta)) = \mathcal{V}(\Delta).$$

\square

Corollary 7.14 (Theorem 6.12). *If $\Delta \overline{\sqsubseteq}_{DS} \Theta$ then $\mathcal{O}(\Delta) \sqsubseteq_S \mathcal{O}(\Theta)$.*

Proof. First suppose that Δ is a closed sub-distribution; let Δ' be a sub-distribution such that $\Delta \implies \Delta'$; we have to show that $\Theta \implies \Theta'$ for some Θ' such that $\mathcal{V}(\Theta') \leq \mathcal{V}(\Delta')$. Note that, since Δ is closed, this is equivalent to $\Delta \vdash \Delta'$. Further, it is straightforward to note that Δ' is terminal. Since $\Delta \overline{\sqsubseteq}_{DS} \Theta$ it follows that $\Theta \vdash \Theta''$ for some Θ'' (which also implies $\Theta \implies \Theta''$) such that $\Delta' \overline{\sqsubseteq}_{DS} \Theta''$. Since Δ' is terminal, by Lemma 7.13 we also have that $\Theta'' \implies \Theta'$ and $\mathcal{V}(\Theta') \leq \mathcal{V}(\Delta')$. Now we just need to combine the reductions $\Theta \implies \Theta''$ and $\Theta'' \implies \Theta'$ to obtain $\Theta \implies \Theta'$.

If Δ is not a closed sub-distribution, we have that $\text{cl}(\Delta) \overline{\sqsubseteq}_{DS} \text{cl}(\Theta)$; since $\text{cl}(\Delta)$ is closed it follows that $\mathcal{O}(\Delta) = \mathcal{O}(\text{cl}(\Delta)) \sqsubseteq_S \mathcal{O}(\text{cl}(\Theta)) = \mathcal{O}(\Theta)$. \square

8. FAILURE OF COMPLETENESS

Although the simulation preorder \triangleleft_{sim} provides a proof methodology for establishing that two networks are related via the testing preorder \sqsubseteq_{may} , it is not complete.

That is, it is possible to find two networks \mathcal{M}, \mathcal{N} such that $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$ holds, but \mathcal{M} cannot be simulated by \mathcal{N} . Similarly, for the must-testing preorder, we have that it is possible to exhibit two networks \mathcal{M}, \mathcal{N} such that $\mathcal{M} \sqsubseteq_{must} \mathcal{N}$, but $\overline{\mathcal{M}} \not\sqsubseteq_{DS} \mathcal{N}$. These results are quite surprising, as simulation preorder has been already proved to provide a characterisation of the may-testing preorder for more standard process calculi such as pCSP, while the must-testing preorder has been proved to be characterised by failure simulations [7]. Here, for simplicity, we discuss the failure of completeness for the sole \sqsubseteq_{may} preorder; however, the examples discussed here can be used to show that the relation \sqsubseteq_{must} is also incomplete.

The main problem that arises in our setting is that the mathematical basis of simulation preorders rely on (full) probability distributions, which are a suitable tool in a framework where a weak action from a process term has to be matched with the same action performed by a distribution of processes.

This is not true in our calculus; we have already shown that, due to the presence of local broadcast communication, it is possible to match a weak broadcast action with a sequence of outputs whose sets of target nodes are pairwise disjoint. This behaviour has been formalised by giving a non-standard definition of weak extensional actions in Definition 5.3.

Such a definition captures the possibility of simulating a broadcast through a multicast only when the former action is performed with probability 1.

However, when comparing distributions of networks we have to also match actions which are performed with probabilities less than 1, at least informally; here the simulation of broadcast using multicast runs into problems, as the following example shows.

Example 8.1. Consider the two networks $\Gamma_M \triangleright M, \Gamma_N \triangleright N$ depicted in Figure 15; let

$$\begin{aligned} M &= m[\tau.(c!\langle v \rangle)_{0.9} \oplus \mathbf{0}] \\ N &= m[c!\langle v \rangle] \mid n[c?(x).(c!\langle x \rangle)_{0.9} \oplus \mathbf{0}] \end{aligned}$$

In $\Gamma_M \triangleright M$ a message is broadcast to nodes o_1, o_2 with probability 0.9, while in $\Gamma_N \triangleright N$ two different broadcasts happen in sequence. The first broadcast, which can be detected by node o_1 , happens with probability 1. The second broadcast, detectable by node o_2 happens with probability 0.9. As a result, the overall probability of message v to be detected by both nodes o_1, o_2 is again 0.9.

We first show that $\Gamma_M \triangleright M \sqsubseteq_{may} \Gamma_N \triangleright N$, then we prove that $\Gamma_M \triangleright M \not\triangleleft_{sim} \Gamma_N \triangleright N$.

For the first statement, we only supply informal details, as a complete proof would be rather long and technical. Consider a test $\Gamma_T \triangleright T$, such that both $(\Gamma_M \triangleright M) \# (\Gamma_T \triangleright T)$ and $\Gamma_N \triangleright N \# (\Gamma_T \triangleright T)$

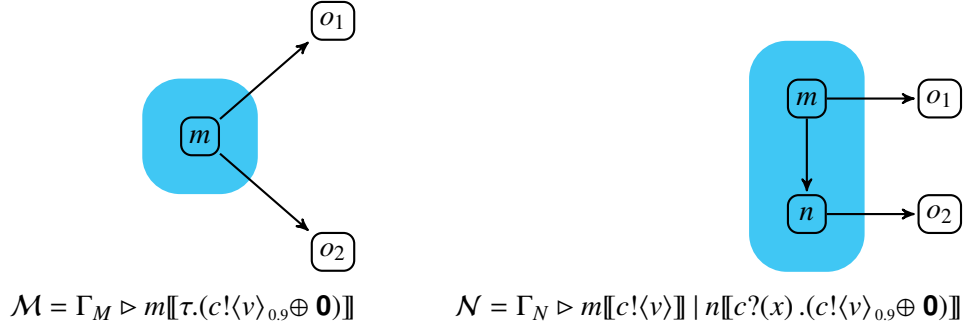


Figure 15: Two testing related networks

are defined. Without loss of generality, suppose that both $o_1, o_2 \in \text{nodes}(\Gamma_T \triangleright T)$, that is $T \equiv o_1[t_1] \mid o_2[t_2] \mid T'$. We consider only the most interesting case, that is when the testing component reaches (with some probability p) an ω -successful configuration after network $\Gamma_M \triangleright M$ broadcasts the message v . In this case, a computation fragment of $(\Gamma_M \triangleright M) \sharp (\Gamma_T \triangleright T)$ can be summarised as follows:

- (1) The testing component $\Gamma_T \triangleright T$ performs some internal activity, thus leading to $\Gamma_T \triangleright T \xrightarrow{\tau} \Gamma_T \triangleright o_1[\Lambda_1] \mid o_2[\Lambda_2] \mid \Lambda_T$,
- (2) At this point, the network $\Gamma_M \triangleright M$ performs a τ -extensional action, specifically $\Gamma_M \triangleright M \xrightarrow{\tau} \Gamma_M \triangleright \Delta$, where

$$\begin{aligned} \Delta &= 0.9 \cdot \overline{M_1} + 0.1 \cdot \overline{M_2} \\ M_1 &= \overline{m[c!\langle v \rangle]} \\ M_2 &= \overline{m[\mathbf{0}]} \end{aligned}$$

- (3) The distribution $\Gamma_T \triangleright o_1[\Lambda_1] \mid o_2[\Lambda_2] \mid \Lambda_T$ performs some other internal activity, that is

$$\Gamma_T \triangleright o_1[\Lambda_1] \mid o_2[\Lambda_2] \mid \Lambda_T \xrightarrow{\tau} \Gamma_T \triangleright o_1[\Lambda'_1] \mid o_2[\Lambda'_2] \mid \Lambda'_T$$

- (4) At this point, the distribution $\Gamma_M \triangleright \Delta$ broadcasts the message v with probability 0.9, causing the testing component to evolve in $\Gamma_T \triangleright o_1[\Lambda''_1] \mid o_2[\Lambda''_2] \mid \Lambda'_T$; note that only nodes o_1 and o_2 are affected by the broadcast performed by node m . After performing the broadcast, the tested network becomes deadlocked.

Consider now the network $(\Gamma_N \triangleright N) \sharp \Gamma_T \triangleright T$. For such a network, a matching computation will proceed as follows:

- (1) The testing component $\Gamma_T \triangleright T$ performs the two sequences of internal activities as before, ending up in the distribution $\Gamma_T \triangleright o_1[\Lambda'_1] \mid o_2[\Lambda'_2] \mid \Lambda'_T$.
- (2) At this point, the network $\Gamma_N \triangleright N$ performs a broadcast, $\Gamma_N \triangleright N \xrightarrow{\tau} \Gamma_N \triangleright \Theta$, where

$$\begin{aligned} \Theta &= 0.9 \cdot \overline{N_1} + \overline{N_2} \\ N_1 &= \overline{m[\mathbf{0}] \mid n[c!\langle v \rangle]} \\ N_2 &= \overline{m[\mathbf{0}] \mid n[\mathbf{0}]} \end{aligned}$$

Here note that, since the broadcast of message v fired by the network $\Gamma_M \triangleright M$ can be detected by the sole node o_1 , only the code running at this node is affected in the test. Further, the

- resulting distribution at this node is again Λ_1'' ; the test component after the broadcast of message v to node o_1 is then in the distribution $\Gamma_T \triangleright o_1 \llbracket \Lambda_1'' \rrbracket \mid o_2 \llbracket \Lambda_2'' \rrbracket \mid \Lambda_T'$.
- (3) Before allowing the testing component to perform any activity, we require the distribution $\Gamma_N \triangleright \Theta$ to perform the second broadcast, which will be heard by node o_2 ; this happens with probability 0.9. Further, such a broadcasts affects the probability distribution of processes running at the sole node o_2 . Thus, after the second message has been broadcast by the tested network, the testing component will have the form $\Gamma_T \triangleright o_1 \llbracket \Lambda_1'' \rrbracket \mid o_2 \llbracket \Lambda_2'' \rrbracket \mid \Lambda_T'$. This is exactly the same configuration obtained in the first experiment, after $\Gamma_M \triangleright M$ has broadcast the message to both nodes o_1, o_2 . Further, note that the overall probability $\Gamma_N \triangleright N$ delivering the message to both the external nodes is again 0.9. Finally, after the broadcast has been fired, the tested network reaches a deadlocked configuration.

We have shown that, whenever the broadcast of message v by $\Gamma_M \triangleright M$ affects the testing network $\Gamma_T \triangleright T$ in some way, then $\Gamma_N \triangleright N$ is able to multicast the message to both o_1 and o_2 , causing $\Gamma_T \triangleright T$ to behave in the same way. Note also that $\ln(\Gamma_M \triangleright M) = \ln(\Gamma_N \triangleright N) = \emptyset$, so that the behaviour of the testing component $\Gamma_T \triangleright T$ does not affect that of the tested networks. Now the reader should be convinced that $\Gamma_M \triangleright M \sqsubseteq_{\text{may}} \Gamma_N \triangleright N$.

Next we show that it is the case that $\Gamma_M \triangleright M$ cannot be simulated by $\Gamma_N \triangleright N$. The proof is obtained by contradiction. Suppose that $\Gamma_M \triangleright M \triangleleft_{\text{sim}} \Gamma_N \triangleright N$. Since $\Gamma_M \triangleright M \xrightarrow{\tau} \Gamma_M \triangleright \Delta$, we have that $\Gamma_N \triangleright N \xrightarrow{\tau} \Gamma_N \triangleright \Theta'$ for some distribution Θ' such that $\Delta \triangleleft_{\text{sim}}^e \Theta'$. It is straightforward to note that whenever $\Gamma_N \triangleright N \xrightarrow{\tau} \Gamma_N \triangleright \Theta'$ then $\Theta' = \bar{N}$.

Recall that $\Delta = 0.9 \cdot \bar{M}_1 + 0.1 \cdot \bar{M}_2$. Since $\Gamma_M \triangleright \Delta \triangleleft_{\text{sim}}^e \Gamma_N \triangleright \bar{N}$, the decomposition property of lifted relations, Definition 2.2 ensures that we can rewrite \bar{N} as $0.9 \cdot \bar{N} + 0.1 \cdot \bar{N}$, and $\Gamma_M \triangleright M_1 \triangleleft_{\text{sim}} \Gamma_N \triangleright N$,

Let us focus on the network $\Gamma_M \triangleright M_1$. This network is equipped with the extensional transition $\Gamma_M \triangleright M \xrightarrow{c!v \triangleright \{o_1, o_2\}} \overline{\Gamma_M \triangleright m \llbracket c! \langle v \rangle \rrbracket}$. Since $\Gamma_M \triangleright M_1 \triangleleft_{\text{sim}} \Gamma_N \triangleright N$, it follows that $\Gamma_N \triangleright N \xrightarrow{c!v \triangleright \{o_1, o_2\}} \Gamma_N \triangleright \Theta''$ for some distribution Θ'' . We show that this is not possible.

This is because the only action that can be performed by $\Gamma_N \triangleright N$ is $\Gamma_N \triangleright N \xrightarrow{c!v \triangleright \{o_1\}} \Gamma_N \triangleright \Theta$; in order for $\Gamma_N \triangleright N$ to be able to perform the weak action $\xrightarrow{c!v \triangleright \{o_1, o_2\}}$ we require the distribution $\Gamma_N \triangleright \Theta$ to perform a weak broadcast to node o_2 . However, this is possible if every network in $[\Gamma_N \triangleright \Theta]$ can perform such an action; this is not true, since $N_2 \in [\Theta]$, and the network $\Gamma_N \triangleright N_2$ is deadlocked.

We have shown that $\Gamma_M \triangleright M_1 \not\triangleleft_{\text{sim}} \Gamma_N \triangleright N$, which in turn gives $\Gamma_M \triangleright \Delta \not\triangleleft_{\text{sim}}^e \Gamma_M \triangleright \bar{N}$. Since $\Gamma_M \triangleright M \xrightarrow{\tau} \Gamma_M \triangleright \Delta$, and $\Gamma_N \triangleright \bar{N}$ is the only hyper-derivative of $\Gamma_N \triangleright N$, we conclude that $\Gamma_M \triangleright M \not\triangleleft_{\text{sim}} \Gamma_N \triangleright N$. \square

Note that the example above can be readapted to show that deadlock simulations are incomplete with respect to the must-testing relations. In fact, for the networks $\Gamma_M \triangleright M, \Gamma_N \triangleright N$ of Example 8.1 it is easy to show that $(\Gamma_N \triangleright N) \sqsubseteq_{\text{must}} (\Gamma_M \triangleright M)$, but $(\Gamma_N \triangleright N) \not\sqsubseteq_{\text{DS}} (\Gamma_M \triangleright M)$.

Example 8.1 has more serious consequences than just showing that simulation preorder is not complete with respect to the may testing preorder. One could in fact expect that the notion of simulation can be modified, leading to a less discriminating preorder for networks which characterises the \sqsubseteq_{may} preorder. We show that this is not the case.

Definition 8.2 (τ -Simulations). A relation $\mathcal{R} \subseteq \text{Nets} \times \mathcal{D}(\text{Nets})$ is a τ -simulation if whenever $M \mathcal{R} N$ then $\ln(M) = \ln(N)$, $\text{Out}(M) = \text{Out}(N)$ and whenever $M \xrightarrow{\tau} \Delta$ it follows that $N \xrightarrow{\tau} \Theta$ for some Θ such that $\Delta \mathcal{R} \Theta$. \square

Note that the definition of τ -simulations is very general, since the only constraints that we have placed on them, apart from the standard checks on the input and output nodes in the interface of networks, is that a strong τ -action has to be matched with a weak one. It follows at once that \triangleleft_{sim} is a τ -simulation.

Theorem 8.3. *There exists no τ -simulation $\mathcal{R} \subseteq \text{Nets} \times \mathcal{D}(\text{Nets})$ such that $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$ iff $\mathcal{M}\mathcal{R}\overline{\mathcal{N}}$.*

Proof. The proof is carried out by contradiction. Suppose $\mathcal{R} \subseteq \text{Nets} \times \mathcal{D}(\text{Nets})$ is a τ -simulation such that $\mathcal{M}\mathcal{R}\overline{\mathcal{N}}$ if and only if $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$, and consider the networks $\Gamma_M \triangleright M, \Gamma_N \triangleright N$ from Example 8.1. We have already proved that $\Gamma_M \triangleright M \sqsubseteq_{may} \Gamma_N \triangleright N$ and so by the hypothesis we have $\Gamma_M \triangleright M\mathcal{R}\overline{\Gamma_N \triangleright N}$. Note that $\Gamma_M \triangleright M \xrightarrow{\tau} \Gamma_M \triangleright \Delta$, where $\Delta = 0.9 \cdot \overline{M_1} + 0.1 \cdot \overline{M_2}$, where M_1, M_2 have been already defined in Example 8.1.

Since \mathcal{R} is a τ -simulation, the τ -action performed by $\Gamma_M \triangleright M$ has to be matched by a hyper-derivation in $\Gamma_N \triangleright N$; we have already noted that the only possible hyper-derivation for such a network is given by $\Gamma_N \triangleright N \xrightarrow{\tau} \overline{\Gamma_N \triangleright N}$. Therefore we have that $\Gamma_M \triangleright \Delta \overline{\mathcal{R}} \overline{\Gamma_N \triangleright N}$. The decomposition property of lifted relations, Definition 2.2 ensures that we can rewrite $\overline{\Gamma_N \triangleright N}$ as $0.9 \cdot \Theta_1 + 0.1 \cdot \Theta_2$, and $\Gamma_M \triangleright M_i \mathcal{R} \overline{\Gamma_N \triangleright \Theta_i}$, $i = 1, 2$. It is trivial to note that here the only possibility is that $\Theta_1 = \Theta_2 = \overline{N}$. Therefore $\Gamma_M \triangleright M_1 \mathcal{R} \overline{\Gamma_N \triangleright \overline{N}}$, and by hypothesis this implies that $\Gamma_M \triangleright M_1 \sqsubseteq_{may} \Gamma_N \triangleright N$.

However, this is not possible. We show that there is a test that distinguishes the network $\Gamma_M \triangleright M_1$ from $\Gamma_N \triangleright N$. Consider the test $\Gamma_T \triangleright T$, where Γ_T is the connectivity graph consisting of the sole node o_2 with no connections, while $T = o_2 \llbracket c?(x).\omega \rrbracket$. It is straightforward to note that $1 \in \mathcal{O}((\Gamma_M \triangleright M_1) \# (\Gamma_T \triangleright T))$. However, for any $p \in \mathcal{O}((\Gamma_N \triangleright N) \# (\Gamma_T \triangleright T))$ we have that $p \leq 0.9$. It follows that $\mathcal{O}((\Gamma_M \triangleright M_1) \# (\Gamma_T \triangleright T)) \not\sqsubseteq_H \mathcal{O}((\Gamma_N \triangleright N) \# (\Gamma_T \triangleright T))$. That is, $\Gamma_M \triangleright M_1 \not\sqsubseteq_{may} \Gamma_N \triangleright N$. Contradiction. \square

9. CASE STUDY: PROBABILISTIC ROUTING

While our proof methods for relating probabilistic networks via the testing preorders are not complete, they are still useful for comparing practical examples of wireless networks. Even more, they can be used to perform a model-based verification of network protocols, showing that their behaviour is consistent with respect to some formal specification. In this section we show how this can be done by proving the correct behaviour of a simple probabilistic routing protocol. For the sake of simplicity, we focus on an abstract implementation of a geographic routing protocol, in which much of the details are left unspecified. However, it is worth mentioning that the proposed implementation can be refined, leading to a concrete representation of the *SAMPLE* probabilistic routing protocol [6].

By *formal specification* we mean a network \mathcal{M} , while by *network protocol* we mean a set of networks \mathcal{N} whose elements share the same input and output nodes. Proving that the behaviour of a protocol \mathcal{N} is sound with respect to a formal verification \mathcal{M} consists then in showing that for any network $\mathcal{N} \in \mathcal{N}$ it has to be $\mathcal{M} \approx \mathcal{N}$.

Let us now turn our attention on how this task can be achieved for a probabilistic (connection-less) routing protocol. At least intuitively, the routing policy states that messages broadcast by a location in a network, called source, are eventually delivered to a desired node, called destination. For the sake of simplicity, here we consider a situation in which the source and the destination of a routing policy are two fixed external nodes, i and o respectively.

Designing a specification for the routing policy is easy; however, there are some details that need to be taken into account. First, we need to introduce some mathematical tools that will enable

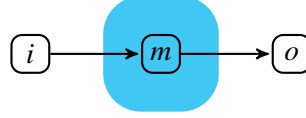


Figure 16: The specification $\Gamma_M \triangleright M$ for the routing policy.

us to equip a node in a network with some sort internal memory; this is necessary, since in a routing protocol nodes have to store the values they have received and which they have not yet forwarded to another node.

This can be done by relying on *multisets*. Roughly speaking, a multiset \mathbb{A} is a set which can contain more than one copy of the same element. Formally, a multiset \mathbb{A} from a set universe U is a function $\mathbb{A} : U \rightarrow \mathbb{N}$ which assigns to each element $u \in U$ the number of copies of u contained in \mathbb{A} . For our purpose the universe U consists of the set of (closed) message values, and we only deal with finite multisets, that is those for which $(\sum_{v \in U} \mathbb{A}(v)) < \infty$.

We denote with \emptyset the empty multiset, that is the multiset such that $\emptyset(v) = 0$ for any value v , and we say that $\mathbb{A} \subseteq \mathbb{B}$ if $\mathbb{A}(v) \leq \mathbb{B}(v)$ for any value v . We say that $v \in \mathbb{A}$ if $\mathbb{A}(v) > 0$. Given a finite collection of multisets $\mathbb{A}_1, \dots, \mathbb{A}_n$, the multiset $(\bigcup_{i=1}^n \mathbb{A}_i)$ is defined by letting $(\bigcup_{i=1}^n \mathbb{A}_i)(v) = \sum_{i=1}^n \mathbb{A}_i(v)$.

Finally, for any multiset \mathbb{A} and a value v , we denote with $\mathbb{A} + v$ the multiset such that $(\mathbb{A} + v)(v) = \mathbb{A}(v) + 1$ and $(\mathbb{A} + v)(w) = \mathbb{A}(w)$ for any $w \neq v$. Similarly, the multiset $\mathbb{A} - v$ is defined by letting $(\mathbb{A} - v)(v) = \mathbb{A}(v) - 1$ if $\mathbb{A}(v) > 0$, $(\mathbb{A} - v)(v) = 0$ if $\mathbb{A}(v) = 0$ and $(\mathbb{A} - v)(w) = \mathbb{A}(w)$ for any $w \neq v$.

The second problem we need to tackle is that of ensuring that the specification we define for the routing policy is a finitary network. This is necessary because our proof techniques are valid only for such networks. As we will see, this can be accomplished by considering a more restricted routing policy, in which only a finite amount of messages will be routed from the source to the destination.

Let $k \geq 0$; the specification we propose for the connection-less routing policy of k values is given by the network $\mathcal{M} = \Gamma_M \triangleright M_{\mathbb{A}}^k$, where Γ_M is the connectivity graph depicted in Figure 16 and $M_{\mathbb{A}}^k$ is a system term (parametrised by a multiset \mathbb{A} and an integer $k \geq 0$) defined as

$$\begin{aligned} M_{\mathbb{A}}^k &= m \llbracket P_{\mathbb{A}}^k \rrbracket \\ P_{\mathbb{A}}^0 &\Leftarrow \sum_{v \in \mathbb{A}} c! \langle v \rangle . P_{\mathbb{A}-v}^0 \\ P_{\mathbb{A}}^{k+1} &\Leftarrow \left(\sum_{v \in \mathbb{A}} c! \langle v \rangle . P_{\mathbb{A}-v}^{k+1} \right) + (c?(x) . P_{\mathbb{A}+x}^k) \end{aligned}$$

Let us discuss the intuitive behaviour of a network of the form $\mathcal{M}_{\mathbb{A}}^k$; at any given point, the internal node m can either receive a message from node i , provided that there are still messages to be routed, or it can forward one of the messages in the multiset \mathbb{A} to the output node o , if any. Note that we require the use of multisets since any value v can be broadcast more than once by the input node i .

Formally, the behaviour of a network $\mathcal{M}_{\mathbb{A}}^k$ can be described as follows.

Proposition 9.1. *For any $k \geq 0$ and finite multiset \mathbb{A}*

- (1) $\mathcal{M}_{\mathbb{A}}^k$ is convergent and finitary,
- (2) $\delta(\mathcal{M}_{\mathbb{A}}^k) = \text{true}$ if and only if $\mathbb{A} = \emptyset$,

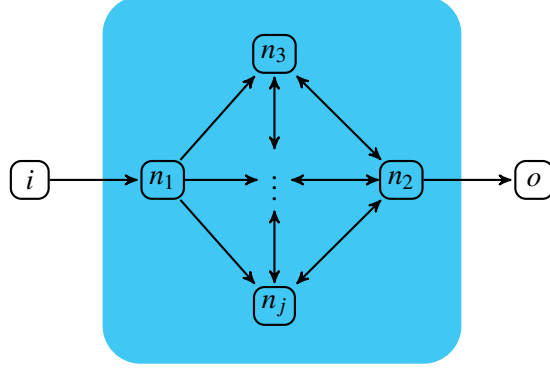


Figure 17: The connectivity graph of the networks in the protocol \mathcal{N} .

(3) if $k > 0$ then $v \mathcal{M}_{\mathbb{A}}^k \xrightarrow{i.c?v} \Delta$ if and only if $\Delta = \overline{\mathcal{M}_{\mathbb{A}+v}^{k-1}}$,

(4) if $k = 0$ then $\mathcal{M}_{\mathbb{A}}^k \xrightarrow{i.c?v} \Delta$ if and only if $\Delta = \overline{\mathcal{M}_{\mathbb{A}}^k}$,

(5) $\mathcal{M}_{\mathbb{A}}^k \xrightarrow{c!v \triangleright \{o\}} \Delta$ if and only if $v \in \mathbb{A}$ and $\Delta = \overline{\mathcal{M}_{\mathbb{A}-v}^k}$. □

Let us now define a protocol which is consistent with the specification $\mathcal{M}_{\mathbb{A}}^k$. As we already mentioned, a protocol is a collection of networks. We consider only the set of networks of the form $\mathcal{N}_{\mathbb{A}}^k = \Gamma_N \triangleright N_{\mathbb{A}}^k$ which satisfy the following conditions.

- (1) $\text{In}(\mathcal{N}_{\mathbb{A}}^k) = \{i\}$, $\text{Out}(\mathcal{N}_{\mathbb{A}}^k) = \{o\}$,
- (2) $\text{nodes}(\mathcal{N}_{\mathbb{A}}^k) = \{n_1, n_2, \dots, n_j\}$ for some $j \geq 2$,
- (3) $\Gamma_N \vdash i \rightarrow m$ if and only if $m = n_1$,
- (4) $\Gamma_N \vdash m \rightarrow o$ if and only if $m = n_2$,
- (5) $\Gamma_N \vdash n_h \rightarrow n_1$ for any $h = 1, \dots, j$,
- (6) for any node $h = 1, \dots, j$ there exists a path from n_h to n_2 in Γ_N ,
- (7) for any node n_h , there exists a probability distribution $\Lambda_h \in \mathcal{D}(\{1, \dots, j\})$ such that $[\Lambda_h] = \{h' \mid \Gamma_N \vdash n_h \rightarrow n_{h'}\}$,
- (8) we assume a set of distinct channels c_1, \dots, c_j such that $c_h \neq c$ for any $h = 1, \dots, j$,
- (9) The system term $N_{\mathbb{A}}^k$ is in the support of a distribution $\Delta_{\mathbb{A}}^k$, defined as

$$\Delta_{\mathbb{A}}^k = \mathbb{P}(n_1 \llbracket Q_{\mathbb{A}_1}^k \rrbracket \mid n_2 \llbracket R_{\mathbb{A}_2} \rrbracket \mid \prod_{h=3}^j n_h \llbracket S_{\mathbb{A}_h}^h \rrbracket)$$

where $(\bigcup_{h=1}^j \mathbb{A}_h) = \mathbb{A}$ and

$$\begin{aligned} \mathcal{Q}_{\mathbb{A}}^0 &\Leftarrow \bigoplus_{h=1}^j \Lambda_1(j) \cdot \left(\sum_{v \in \mathbb{A}} c_h ! \langle v \rangle \cdot \mathcal{Q}_{\mathbb{A}-v}^0 \right) \\ \mathcal{Q}_{\mathbb{A}}^{k+1} &\Leftarrow \bigoplus_{h=1}^j \Lambda_1(h) \cdot \left[c^?(x) \cdot \mathcal{Q}_{\mathbb{A}+x}^k + \left(\sum_{v \in \mathbb{A}} c_h ! \langle v \rangle \cdot \mathcal{Q}_{\mathbb{A}-v}^{k+1} \right) \right] \\ R_{\mathbb{A}} &\Leftarrow c_2^?(x) \cdot R_{\mathbb{A}+x} + \left(\sum_{v \in \mathbb{A}} c ! \langle v \rangle \cdot R_{\mathbb{A}-v} \right) \\ S_{\mathbb{A}}^h &\Leftarrow \bigoplus_{h'=1}^j \Lambda_h(h') \cdot \left[c_h^?(x) \cdot S_{\mathbb{A}+x}^h + \left(\sum_{v \in \mathbb{A}} c_{h'} ! \langle v \rangle \cdot S_{\mathbb{A}-v}^h \right) \right] \end{aligned}$$

Here we use $\bigoplus_{i=1}^n p_i \cdot s_i$ to denote the process such that

$$\mathbb{P}(\bigoplus_{i=1}^n p_i \cdot s_i) = \sum_{i=1}^n p_i \cdot \bar{s}_i$$

We denote with $\mathcal{N}_{\mathbb{A}}^k$ the set of networks $\mathcal{N}_{\mathbb{A}}^k$ described above. The connectivity graph of such networks is depicted in Figure 17

Remark 9.2. Note that we committed an abuse of notation in defining the distribution $\Delta_{\mathbb{A}}^k$, by associating a process definition with a (probabilistic) process, rather than to a state. However, a process definition of the form $A \Leftarrow \bigoplus_{i=1}^n p_i \cdot s_i$ can be seen as the probabilistic process $\bigoplus_{i=1}^n p_i \cdot A_i$, where $A_i \Leftarrow s'_i$ and s'_i is obtained from s_i by replacing each occurrence of A with $\bigoplus_{i=1}^n p_i \cdot A_i$.

Our aim is to show that for any $\mathcal{N} \in \mathcal{N}_{\emptyset}^k$ we have that $\mathcal{M}_{\emptyset}^k \simeq \mathcal{N}$.

Before supplying the details of the proof of the statement above, let us describe informally the behaviour of a distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$; we also discuss the requirements that we have placed on the structure of the connectivity graph $\Gamma_{\mathcal{N}}$. In a distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$ a network is waiting to receive exactly k messages from node i , and whose multiset of received messages which have been received but have not yet been forwarded to the external node o is \mathbb{A} . Note that we have placed many requirements in the definition of the connectivity graph of such networks; first we require that i is their only input node, while o is their only output node. This requirement is necessary, since to show that such networks are testing equivalent to the specification we have to ensure that they share with the latter the same sets of input and output nodes.

We require the connectivity graph of the networks in Δ to have a path from n_h to n_2 for every $h = 1, \dots, j$. This condition is needed to ensure that messages detected by node n_1 (which in turn have been broadcast by i) can flow through the network until reaching node n_2 , which in turn can broadcast the message to the output node o . As we will see this always happens with probability 1.

The other constraints that we placed on the connectivity graph of $\mathcal{N} \in [\Delta]$ are purely technical; we require that the only node connected to i is n_1 , while the only node connected to o is n_2 . As we will see when discussing the code running at $\mathcal{N}_{\mathbb{A}}^k$, nodes n_1 and n_2 have the role of handling the values broadcast by i , and which have to be forwarded to o , respectively. We also require that $\Gamma_{\mathcal{N}} \vdash n_h \rightarrow n_1$ for any $h = 1, \dots, j$. This constraint ensures that all the messages received by n_1 have been broadcast by the input node i ; note in fact that, in general, a node cannot detect the name of the node that fired a broadcast.

Let us now turn our attention to the code defined for the distribution $\Delta_{\mathbb{A}}^k$. Here we assume a set of channels c_1, \dots, c_j ; each node $n_h, h \neq 1$ can only detect messages broadcast along the channel c_h . Intuitively, when a message is broadcast along channel c_h by a node $n_{h'}$, then it will be delivered to node n_h . In other words, node $n_{h'}$ has selected n_h as the next hop in a routing path.

We also assume a set of probability distributions $\Lambda_1, \dots, \Lambda_j$. When a node $n_h, h \neq 2$ wishes to select the next hop in a routing path, it selects it according to the probability distribution Λ_h . Note that we require that $h' \in [\Lambda_h]$ if and only if $\mathcal{N}_{\mathbb{A}}^k \vdash n_h \rightarrow n_{h'}$, that is a node can be selected as the next hop in a routing path by n_h if and only if it is in the range of transmission of n_h . Further, any neighbour of n_h can be selected as the next hop in a routing path. As we will see, this constraint ensures that, in unbounded time, a message stored in node n_h will reach the node n_2 with probability 1.

Any network distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\emptyset}^k)$ can be seen as a probability distribution of networks running a (connection-less) routing algorithm of k messages. Such an algorithm is designed by letting any node n_h , with the exception of n_2 , to select the next hop in a routing path probabilistically among its neighbours. For node n_2 , the message is broadcast along channel c with probability 1, thus forwarding it to the only output node o . Also, the message to be forwarded to a next-hop in a routing path by node n_h is selected non-deterministically among those stored in such a node, that is the nodes in the multiset \mathbb{A}_h .

Roughly speaking, the behaviour of a network $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$ can be described as follows:

- (1) node n_1 can receive a message v broadcast by node i along channel c , provided $k \geq 0$. Then it stores it in the multiset associated to it,
- (2) At any given point, any node $n_h, h \neq 2$ can select the next hop in a routing path among its neighbours. Then it selects the message to be forwarded non-deterministically among those stored in its internal multiset
- (3) At any given point, node n_2 can broadcast one of the messages stored in its multiset along channel c . This broadcast is detected by the output node o .

The behaviour of a network $\mathcal{N} \in \mathcal{N}_{\emptyset}^k$ is similar, with the only exception that the first time each node receives a message, the next-hop of a routing path it chooses is fixed.

Let us now turn our attention to the extensional transitions performed by a network $\mathcal{N} \in \mathcal{N}_{\mathbb{A}}^k$, and more generally by a distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$. To this end it is useful to introduce some notation. First we define the (state based) processes

$$\begin{aligned} q_{\mathbb{A}}^{0,h} &= \sum_{v \in \mathbb{A}} c_h ! \langle v \rangle . \mathcal{Q}_{\mathbb{A}-v}^0 \\ q_{\mathbb{A}}^{k+1,h} &= c ? (x) . \mathcal{Q}_{\mathbb{A}+x}^k + \left(\sum_{v \in \mathbb{A}} c_j ! \langle v \rangle . \mathcal{Q}_{\mathbb{A}-v}^{k+1} \right) \\ s_{\mathbb{A}}^{h,h'} &= c_h ? (x) . \mathcal{S}_{\mathbb{A}+x}^h + \left(\sum_{v \in \mathbb{A}} c_{h'} ! \langle v \rangle . \mathcal{S}_{\mathbb{A}-v}^h \right) \end{aligned}$$

and we note that any network $\mathcal{N} \in \mathcal{N}_{\mathbb{A}}^k$ has the form

$$\mathcal{N} = \Gamma_N \triangleright \llbracket q_{\mathbb{A}_1}^{k,h} \rrbracket \llbracket R_{\mathbb{A}_2} \rrbracket \prod_{h=3}^j n_h \llbracket s_{\mathbb{A}_h}^{h,h'} \rrbracket$$

where $(\bigcup_{h=1}^j \mathbb{A}_h) = \mathbb{A}$. For such networks, we define $\text{Values}_{\mathcal{N}}(h) = \mathbb{A}_h$. Intuitively, this function returns the multiset of values stored at node n_h in the network \mathcal{N} .

Finally, let $(\mathcal{N}_{\mathbb{A}}^k)^x$ be the unique network such that $\text{Values}_{(\mathcal{N}_{\mathbb{A}}^k)^x}(2) = \mathbb{A}$. This is the network where all the nodes that have to be routed are stored in the node n_2 ; therefore they are ready to be forwarded to the destination o .

We are now ready to characterise the set of strong extensional transitions performed by any network $\mathcal{N} \in \mathcal{N}_{\mathbb{A}}^k$.

Proposition 9.3. For any network $\mathcal{N} \in \mathcal{N}_{\mathbb{A}}^k$,

- (1) (i) $\mathcal{N} \xrightarrow{\tau} \Delta$ iff $\mathcal{N} = (\mathcal{N}_{\mathbb{A}}^k)^x$,
 (ii) otherwise $\mathcal{N} \xrightarrow{\tau} \Delta$ for some $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$,
- (2) conversely, whenever $\mathcal{N} \xrightarrow{\tau} \Delta$ then $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$,
- (3) if $k > 0$ then
 - (i) $\mathcal{N} \xrightarrow{i.c?v} \Delta$ for some $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}+v}^{k-1})$,
 - (ii) conversely, whenever $\mathcal{N} \xrightarrow{i.c?v} \Delta$ then $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}+v}^{k-1})$,
- (4) if $k = 0$ then $\mathcal{N} \xrightarrow{i.c?v} \Delta$ iff $\Delta = \overline{\mathcal{N}}$,
- (5) for any $v \in \text{Values}_{\mathcal{N}}(2)$ then $\mathcal{N} \xrightarrow{c!v \triangleright \{o\}} \Delta$ for some $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}-v}^k)$,
- (6) conversely, whenever $\mathcal{N} \xrightarrow{c!v \triangleright \{o\}} \Delta$ then $v \in \text{Values}_{\mathcal{N}}(2)$ and $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}-v}^k)$. □

However, in order to show that the specification $\mathcal{M}_{\emptyset}^k$ is testing equivalent to any network in $\mathcal{N}_{\emptyset}^k$ we have to characterise also the set of weak extensional actions performed by such networks. To this end, we first analyse the structure of any τ -extensional transition performed by any distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$.

Proposition 9.4. Let $k \geq 0$, \mathbb{A} be a finite multiset and suppose $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$.

- (1) $\Delta \xrightarrow{\tau} \overline{(\mathcal{N}_{\mathbb{A}}^k)^x}$,
- (2) whenever $\Delta \xrightarrow{\tau} \Delta'$, then $\Delta' \xrightarrow{\tau} \overline{(\mathcal{N}_{\mathbb{A}}^k)^x}$.

Outline of the Proof. First note that, for any $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$ Proposition 9.3 ensures that $\Delta \xrightarrow{\tau} \Delta'$ implies $\Delta' \in \mathcal{D}_{\text{sub}}(\mathcal{N}_{\mathbb{A}}^k)$.

Let us focus on the proof of the first statement. Let $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$ for some $k \geq 0$ and finite multiset \mathbb{A} . We actually prove a stronger statement than (1), that is that $\Delta \xrightarrow{\tau} \overline{(\mathcal{N}_{\mathbb{A}}^k)^x}$. First note that Theorem 2.6 (4) ensures that there exists a sub-distribution Θ such that $\Delta \xrightarrow{\tau} \Theta$. Such a distribution Θ has to be an element of the set $\mathcal{D}_{\text{sub}}(\mathcal{N}_{\mathbb{A}}^k)$; further, any state in its support should not be able to perform an extensional τ -action. It follows from Proposition 9.3 that the only possibility is that $\{\Theta\} \supseteq \{(\mathcal{N}_{\mathbb{A}}^k)^x\}$, or equivalently that $\Theta = p \cdot (\mathcal{N}_{\mathbb{A}}^k)^x$ for some $0 \leq p \leq 1$. It remains to prove that $p = 1$.

This follows because the probability distribution used by any node $n_h, h \neq 2$, to select the next-hop in a routing path is defined so that any neighbour of n_h can be chosen with probability strictly greater than 0; in particular, since we are assuming that there exists a path from node n_h to node n_2 , a node $n_{h'}$ whose distance to n_2 is less than the distance between n_h and n_2 can be selected with non-negligible probability. As a consequence, in the long run the average distance between the node where a message $v \in \mathbb{A}$ is stored and the node n_2 decreases to 0; that is, with probability 1 message v is stored in the node n_2 . Since this line of reasoning is independent from the value v , we also have that in the long run any message in \mathbb{A} will be stored in n_2 with probability 1; formally, $\Theta = 1 \cdot (\mathcal{N}_{\mathbb{A}}^k)^x$.

Now statement (2) follows trivially. Whenever $\Delta \xrightarrow{\tau} \Delta'$ we have that $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$, and by (1) above it follows that $\Delta' \xRightarrow{\tau} \overline{\mathcal{N}_{\mathbb{A}}^k}$. \square

Corollary 9.5. Any $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$ is convergent.

Proof. Suppose $\Delta \xrightarrow{\tau} \Delta'$ for some Δ' ; then $\Delta' \xrightarrow{\tau} \overline{(\mathcal{N}_{\mathbb{A}}^k)^x}$; it follows that $|\Delta'| \geq 1$, hence $|\Delta'| = 1$. As a consequence, for no network $\mathcal{N} \in [\Delta]$ we have $\mathcal{N} \xRightarrow{\tau} \varepsilon$. \square

The last step that we need to take is that of characterising the set of (weak) input and output transitions for any distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$. This can be done by using both propositions 9.3 and 9.4.

Proposition 9.6. Let $k \geq 0$ and \mathbb{A} be a multiset. Then for any distribution $\Delta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$,

- (i) $\Delta \xrightarrow{\tau} \Delta'$ with $\delta(\mathcal{N}) = \text{true}$ for any $\mathcal{N} \in [\Delta']$ if and only if $\mathbb{A} = \emptyset$,
- (ii) if $k > 0$ then
 - (i) $\Delta \xrightarrow{i.c?v} \Delta'$ for some $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}+v}^{k-1})$,
 - (ii) conversely, whenever $\Delta \xrightarrow{i.c?v} \Delta'$ then $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}+v}^{k-1})$,
- (iii) if $k = 0$ then
 - (i) $\Delta \xrightarrow{i.c?v} \Delta'$ for some $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^0)$,
 - (ii) whenever $\Delta \xrightarrow{i.c?v} \Delta'$ then $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^0)$,
- (iv) if $\mathbb{A} \neq \emptyset$,
 - (a) $\Delta \xrightarrow{c!v>\{o\}} \Delta'$ for some $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}-v}^k)$,
 - (b) conversely, whenever $\Delta \xrightarrow{c!v>\{o\}} \Delta'$ it follows that $\Delta' \in \mathcal{D}(\mathcal{N}_{\mathbb{A}-v}^k)$. \square

We are now ready to show that the protocol $\mathcal{N}_{\mathbb{A}}^k$ satisfies the specification $\mathcal{M}_{\mathbb{A}}^k$.

Theorem 9.7. For any $k \geq 0$ and $\mathcal{N} \in \mathcal{N}_{\emptyset}^k$ we have that $\mathcal{M}_{\emptyset}^k \simeq \mathcal{N}_{\emptyset}^k$.

Proof. Let $k \geq 0$ and \mathbb{A} be a finite multiset. We have already noted that the network $\mathcal{M}_{\mathbb{A}}^k$ is finitary. Further, it is easy to show that any network $\mathcal{N} \in \mathcal{N}_{\mathbb{A}}^k$ is finite state, and by Corollary 9.5 it follows that it is also finitary.

Therefore, it suffices to show that for any $\mathcal{N} \in \mathcal{N}_{\emptyset}^k$ we have both $\mathcal{M}_{\emptyset}^k \triangleleft_{ds} \mathcal{N}$ and $\mathcal{N} \triangleleft_{ds} \mathcal{M}_{\emptyset}^k$. Theorem 6.16 gives that $\mathcal{M}_{\emptyset}^k =_{\text{must}} \mathcal{N}$, while Theorem 6.17 ensures that $\mathcal{M}_{\emptyset}^k =_{\text{may}} \mathcal{N}$.

In fact we prove a stronger statement. For any $k \geq 0$, finite multiset \mathbb{A} and network $\mathcal{N} \in \mathcal{N}_{\mathbb{A}}^k$ we have that $\mathcal{M}_{\mathbb{A}}^k \triangleleft_{ds} \mathcal{N}$, and conversely $\mathcal{N} \triangleleft_{ds} \mathcal{M}_{\mathbb{A}}^k$. Theorem 9.7 follows by letting $\mathbb{A} = \emptyset$.

To this end, consider the relation

$$\mathcal{S} = \{(\mathcal{M}_{\mathbb{A}}^k, \mathcal{N}') \mid \mathcal{N}' \in \mathcal{N}_{\mathbb{A}}^k\}$$

We show that this relation satisfies the requirements of Definition 6.15. First suppose that $\delta(\mathcal{M}_{\mathbb{A}}^k) = \text{true}$. Then $\mathbb{A} = \emptyset$ by Proposition 9.1, and Proposition 9.6 ensures that $\overline{\mathcal{N}_{\emptyset}^k} \xrightarrow{\tau} \Theta$ for some Θ such that for any $\mathcal{N}' \in [\Theta]$ we have that $\delta(\mathcal{N}') = \text{true}$.

Now, suppose that $\mathcal{M}_{\mathbb{A}}^k \xrightarrow{i.c?v} \Delta$. By Proposition 9.1 we have two possible cases:

- (1) $k = 0$; in this case $\Delta = \overline{\mathcal{M}_{\mathbb{A}}^k}$; by Proposition 9.6 we have that $\mathcal{N} \xrightarrow{i.c?v} \Theta$ for some $\Theta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}}^k)$, and trivially $\mathcal{M}_{\mathbb{A}}^k \mathcal{S}^e \Theta$.

(2) $k \geq 0$; here $\Delta = \overline{\mathcal{M}_{\mathbb{A}}^{k-1}}$. The action $\mathcal{M}_{\mathbb{A}}^k \xrightarrow{i.c?v} \overline{\mathcal{M}_{\mathbb{A}+v}^{k-1}}$ can be matched by $\mathcal{N} \xrightarrow{i.c?v} \Theta$, where $\Theta \in \mathcal{D}(\mathcal{N}_{\mathbb{A}+v}^{k-1})$, again using Proposition 9.6.

The last case we need to check is $\mathcal{M} \xrightarrow{c!v\triangleright\{o\}} \Delta$. This case is handled in the same way of the previous ones, again using Propositions 9.1 and 9.6.

For the opposite implication, $\mathcal{N} \triangleleft_{ds} \mathcal{M}^k$, it is sufficient to consider the converse relation \mathcal{S}^{-1} , showing that it satisfies the requirements of Definition 6.15. The proof is similar to the one above, this time by using Proposition 9.3 to infer the structure of an extensional action of the form $\mathcal{N} \xrightarrow{\lambda} \Theta$, and by matching it with an action performed by $\mathcal{M}_{\mathbb{A}}^k$ according to Proposition 9.1. Here it is important to note that every action of the form $\mathcal{M}_{\mathbb{A}}^k \xrightarrow{\lambda} \Delta$ is also a weak action, that is $\mathcal{M}_{\mathbb{A}}^k \xRightarrow{\lambda} \Delta$, and that a strong τ -extensional action of the form $\mathcal{N} \xrightarrow{\tau} \Theta$ can be matched by the weak action $\mathcal{M}_{\mathbb{A}}^k \xRightarrow{\tau} \overline{\mathcal{M}_{\mathbb{A}}^k}$. \square

10. CONCLUSIONS

In this paper we have developed a calculus for wireless systems, which enjoys both probabilistic behaviour and local broadcast communication. We have developed a theory based on the probabilistic testing preorders, and provided sound proof methods for finitary networks to prove that they can be related via our behavioural preorders. We have applied our proof techniques to check that a probabilistic routing protocol is consistent with a given specification.

While testing theories have been analysed for process algebras [9] with broadcast communication over a flat topology, we believe that this is the first work that considers testing theories for a calculus which enjoys local broadcast communication.

In the past the development of formal tools for wireless networks has focused either on other forms of behavioural theories (such as variants of weak bisimulation) and the analysis of protocols. Here we give a brief review of the main works which have inspired our calculus.

To the best of our knowledge, the first paper describing a process calculus for broadcasting systems, **CBS**, is [28]. In this paper the author presents a simple process calculus in which a synchronisation between a sender and a receiver is modelled as an output action, rather than an internal activity as in standard process calculi such as CCS. This allows multiple receivers to detect a message sent by a sender, thus implementing broadcast communication. In [17] different notions of barbed congruence for a variant of CBS are introduced; these correspond to strong barbed congruence and weak barbed congruence. For each of them, a characterisation result in terms of strong and weak bisimulation, respectively, is proved.

Another calculus to model broadcast systems, known as the $b\pi$ -calculus and inspired by both CBS and the π -calculus [31], is introduced in [8]; as the author points out, broadcast communication is modelled in the same style of CBS. In this paper the authors define three different behavioural equivalences, corresponding to barbed congruence, step equivalence and labelled bisimilarity. The author proves that such behavioural equivalences coincide.

In [9] the authors define both the may and must testing preorders for processes of the $b\pi$ -calculus, and they prove a characterisation result for each of them. The main contribution here lies in the characterisation of the must-testing preorder; as the authors point out, in fact, broadcast communication leads to a non-standard characterisation of the latter. In particular, the non-blocking nature of broadcast actions does not allow acceptance sets to be used in their characterisation result.

In the last decade, broadcast calculi have been modified in several ways by equipping processes with a topological structure, thus modelling wireless networks; the idea is that of representing a process as a set of locations, running different code for broadcasting and receiving messages; the topology defined for a process establishes how communication is modelled, for example by letting only some locations being able to detect the messages broadcast at another one.

In [26] the authors propose to model the topological structure of a network by using a connectivity graph; a process is viewed as a set of locations running code, while a graph whose vertices are locations is used to determine how communication is carried out. Intuitively, a transmission originated at a given location can only be detected by those vertices which are connected to the former. The transition relation of processes is defined as parametric in a connectivity graph. This framework has been proposed by the authors as a basis for the analysis of security protocols in wireless networks.

In [25] an allocation environment is used to represent the topological structure of a wireless networks. A wireless network is intended as a parallel composition of processes, each of which is associated with a set of locations to which the process belongs and a probability distribution over locations; intuitively, the latter describes the probability with which a message broadcast by the process is detected at a given location.

In [12] the authors propose a *restricted broadcast process theory* to model wireless networks. Here a network consists of a parallel composition of different processes; each process is associated with a location name, and a function between locations to sets of locations is used to represent the network topology. The authors propose the standard notion of weak bisimulation as the behavioural equivalence to be used to relate networks and they show a case study in which they prove the correctness of a routing protocol.

In [13] an extension of the restricted broadcast process theory, the *Computed Network Theory*, is proposed; here the expressive power of a network is augmented through different operators. For the resulting calculus, a variant of strong bisimulation is defined and proved to be a congruence. The main result in the paper is a sound axiomatisation of the strong bisimulation, thus enabling equational reasoning for wireless networks. The authors also show that the proposed axiomatisation is complete in a setting where only non-recursive networks are considered. The *Computed Network Theory* framework is also used in [14] to check properties of mobile networks; the authors show how both the equational theory and model checking can be used to verify the correctness of a routing protocol.

In [32] the authors view a network as a collection of processes, each of which is associated with one or more groups. Processes which belong to the same group are assumed to be neighbours; as a consequence, a broadcast performed by a process can be detected by all the processes which belong to at least one group of the broadcaster. The authors show that in their framework state reachability is a decidable problem; further, they introduce different notions of behavioural equivalences, based on late bisimilarity and its weak variant, and they show that such equivalences are in fact congruences. Finally, they apply their calculus by formalising and analysing the behaviour of a leader election protocol and a routing protocol.

In [21] the authors describe wireless networks by using metric spaces; they assume that a network consists of a set of processes, each of which has an associated location and a radius of transmission; a metric distance over the set of locations is assumed to determine how communication is modelled. The authors describe the behaviour of a wireless network in terms of both a reduction semantics and a labelled transition semantics. These two semantics are proved to be equivalent up-to

a notion of structural congruence. We remark that in their paper the authors assume that a communication between two stations consists of two phases, one for the beginning and one for termination. This allows the authors to model collision-prone communication.

In [36] the authors present an extension of the calculus described above, in which node mobility and timed communication are introduced. The authors give both a reduction semantics and a labelled transition semantics, and they prove that they are congruent up-to structural congruence.

Another calculus for wireless networks in which collision-prone behaviour is taken into account is described in [23]. In their work, the authors describe a network as a set of processes running in parallel, each of which has a location name and a semantic tag associated with it; the latter consists of a set of locations names and it corresponds to the set of locations which can detect messages broadcast by the process. The calculus includes a notion of discrete-time, in the style of [19], and broadcasts of messages start and end at different time slots. The authors develop a notion of barbed congruence for wireless systems and they propose a sound, but not complete, characterisation result in terms of weak bisimulation.

A variant of this calculus which considers only networks with flat topology is presented in [5]. Here the authors develop a notion of reduction barbed congruence for their calculus; they also introduce an extensional semantics whose induced weak bisimulation principle is proved to be sound and complete with respect to the barbed congruence.

In [34] the authors propose a model in which the topological structure of a network is represented as a graph whose vertices are locations; further, they assign to each edge in the graph a (possibly unknown) probability as a likelihood estimate of whether a message broadcast by a location at the starting end-point of an edge will be delivered to the location at the terminal end-point of the same. The proposed model also allows the network topology of a system to evolve according to a probabilistic mobility function. The authors prove that, in the proposed calculus, the logical equivalence defined over a variant of PCTL coincides with weak bisimulation.

In [33] several models for modelling probabilistic ad hoc networks are developed; the author first defines a probabilistic process calculus where connections between nodes are probabilistic. Behavioural theories based on bisimulation and temporal logics are defined for analysing the properties of networks in such a calculus. The presented calculus is then extended in order to model different features of wireless networks, such as exponential time delays and changes in the network topology.

In [22] the authors define a language for wireless networks in which the code running at network locations contains both non-probabilistic and non-deterministic behaviour. The topological structure of a network is defined in the same way of [23]; the authors introduce a notion of simulation, parametrised in a probability value, in order to capture the concept of two networks exhibiting the same behaviour up-to such a probability. The model used to represent wireless networks and define their formal behaviour is that of a pLTS.

In [11] the authors propose a probabilistic, energy-aware process calculus of networks. In this calculus nodes can move probabilistically among locations of a given metric space. Nodes can also choose the transmission radius of a broadcast in an optimal way with respect to energy consumption. The authors propose a notion of probabilistic barbed congruence, parametrised in a set of schedulers, for which they give a characterisation in terms of bisimulations. The authors also introduce a preorder which compare networks which exhibit the same behaviour according to the proposed contextual equivalence, but differ in terms of energy consumption.

In [2] a variant of the calculus above is proposed, where energy consumption is no longer considered and the possibility of interferences in communications is introduced. The authors define a contextual equivalence in terms of probabilistic reduction barbed congruence, for which they develop a sound and complete proof technique based on bisimulations.

In [15] a different approach is made to formalise a wireless network. The authors identify a network as a set of processes associated with a location address and a queue, representing the data at the datalink layer that a station has not yet broadcast. The calculus they use is a probabilistic generalisation of the restricted broadcast process theory of [13]; here the sending primitive consists of a message to be broadcast and a probability rate, representing the likelihood that such a message will be sent. The model used to describe the behaviour of a system is that of Continuous Time Markov Automata.

APPENDIX A. PROPERTIES OF THE OPERATOR \sharp

Proof of Proposition 4.4(1): Let $\mathcal{M} = \Gamma_M \triangleright M$, $\mathcal{N} = \Gamma_N \triangleright N$, and suppose $(\mathcal{M} \sharp \mathcal{N})$ is defined. Then such a network is equal to $(\Gamma_M \cup \Gamma_N) \triangleright (M | N)$, for which we have to verify two statements:

- $(\Gamma_M \cup \Gamma_N) \triangleright (M | N)$ satisfies the constraints we have placed over all, possibly non well-formed, networks. These require $M | N \in \text{sSys}$ (that is, it does not contain replicated node names), $\text{nodes}(M | N) \subseteq \Gamma_V$, and $(\Gamma_M \cup \Gamma_N)_E$ being irreflexive.

(1) $M | N \in \text{sSys}$; note that if there were a node name m which appears more than once in $M | N$, then we should have $m \in \text{nodes}(M), m \in \text{nodes}(N)$. This is because $M \in \text{sSys}, N \in \text{sSys}$, so that m cannot appear more than once in M , nor in N . Thus the statement follows if we can prove that $\text{nodes}(M) \cap \text{nodes}(N) = \emptyset$; since $\mathcal{M} \sharp \mathcal{N}$ is defined, it follows that $\text{nodes}(M) \cap (\Gamma_N)_V = \emptyset$. Since $\text{nodes}(N) \subseteq (\Gamma_N)_V$, we also have $\text{nodes}(M) \cap \text{nodes}(N) = \emptyset$, and there is nothing left to prove.

(2) $\text{nodes}(M | N) \subseteq (\Gamma_M \cup \Gamma_N)_V$; note that $\text{nodes}(M) \subseteq (\Gamma_M)_V$ and $\text{nodes}(N) \subseteq (\Gamma_N)_V$. Therefore we have that

$$\text{nodes}(M | N) = \text{nodes}(M) \cup \text{nodes}(N) \subseteq (\Gamma_M)_V \cup (\Gamma_N)_V = (\Gamma_M \cup \Gamma_N)_V.$$

(3) $(\Gamma_M \cup \Gamma_N)_E$ is irreflexive. Suppose $(\Gamma_M \cup \Gamma_N) \vdash m \rightarrow n$; We need to show that $m \neq n$. Note that we have either $\Gamma_M \vdash m \rightarrow n$ or $\Gamma_N \vdash m \rightarrow n$; without loss of generality, assume $\Gamma_M \vdash m \rightarrow n$. Since $(\Gamma_M)_E$ is irreflexive, it follows that $m \neq n$.

- $(\Gamma_M \cup \Gamma_N) \triangleright (M | N)$ satisfies the constraints of Definition 4.2. This amounts to prove the following:

(1) for any m, n such that $\Gamma_M \cup \Gamma_N \vdash m \rightleftharpoons n$, either $m \in \text{nodes}(M | N)$ or $n \in \text{nodes}(M | N)$. Let m, n be two nodes for which $(\Gamma_M \cup \Gamma_N) \vdash m \rightleftharpoons n$; that is either $(\Gamma_M \cup \Gamma_N) \vdash m \rightarrow n$ or $(\Gamma_M \cup \Gamma_N) \vdash m \leftarrow n$. Without loss of generality, assume that $(\Gamma_M \cup \Gamma_N) \vdash m \rightarrow n$.

In this case either $\Gamma_M \vdash m \rightarrow n$ or $\Gamma_N \vdash m \rightarrow n$. We only give details for the case in which $\Gamma_M \vdash m \rightarrow n$, as the proof for the second case is analogous. Since $\mathcal{M} \in \text{Nets}$, we have that either $m \in \text{nodes}(M)$ or $n \in \text{nodes}(N)$. If $m \in \text{nodes}(M)$ then $m \in \text{nodes}(M | N)$, while if $n \in \text{nodes}(M)$, then $n \in \text{nodes}(M | N)$. Thus, either $m \in \text{nodes}(M | N)$ or $n \in \text{nodes}(M | N)$.

(2) Let $m \in (\Gamma_M \cup \Gamma_N)_V$ be a node such that $(\Gamma_M \cup \Gamma_N) \vdash m \rightleftharpoons n$ for no node $n \in (\Gamma_N)_V$. Then $m \in \text{nodes}(M | N)$.

Since $m \in (\Gamma_M \cup \Gamma_N)_V$ then either $\Gamma_M \vdash m$ or $\Gamma_N \vdash m$. Without loss of generality let $\Gamma_M \vdash m$. Also, since $(\Gamma_M \cup \Gamma_N) \vdash m \rightleftharpoons n$ for no $n \in (\Gamma_N)_V$, $(\Gamma_M)_V \subseteq (\Gamma_M \cup \Gamma_N)_V$ and $(\Gamma_M)_E \subseteq (\Gamma_M \cup \Gamma_N)_E$, then we also have that $\Gamma_M \vdash m \rightleftharpoons n$ for no $n \in (\Gamma_M)_V$.

Thus we have that $\Gamma_M \vdash m$ and $\Gamma_M \vdash m \rightleftharpoons n$ for no $n \in (\Gamma_M)_V$. Since $(\Gamma_M \triangleright M)$ is well-formed by hypothesis, we must have $m \in \text{nodes}(M)$, from which it follows that $m \in \text{nodes}(M | N)$. \square

Proof of Proposition 4.4(2): It is sufficient to check that $\text{nodes}(\mathcal{M}) \cap (\mathcal{N})_V = \emptyset$ and $\text{nodes}(\mathcal{M} \# \mathcal{N}) \cap (\mathcal{L})_V = \emptyset$ if and only if $\text{nodes}(\mathcal{N}) \cap (\mathcal{L})_V = \emptyset$ and $\text{nodes}(\mathcal{M}) \cap (\mathcal{N} \# \mathcal{L})_V = \emptyset$. In fact, from this claim it follows that $(\mathcal{M} \# \mathcal{N}) \# \mathcal{L}$ is defined if and only if $\mathcal{M} \# (\mathcal{N} \# \mathcal{L})$ is defined; the equality of these two networks follows from the associativity of both set union and parallel composition of system terms.

Let $\mathcal{M} = \Gamma_M \triangleright M$, $\mathcal{N} = \Gamma_N \triangleright N$ and $\mathcal{L} = \Gamma_L \triangleright L$. We prove the two implications above separately.

Suppose that

$$(\text{nodes}(\mathcal{M}) \cap (\Gamma_N)_V) = \emptyset \quad (\text{A.1})$$

$$(\text{nodes}(\mathcal{M} \mid N) \cap (\Gamma_L)_V) = \emptyset \quad (\text{A.2})$$

We want to show that $\text{nodes}(\mathcal{N}) \cap (\Gamma_L)_V = \emptyset$, and $\text{nodes}(\mathcal{M}) \cap (\Gamma_N \cup \Gamma_L)_V = \emptyset$. The former statement is a straightforward consequence of Equation (A.2), since $\text{nodes}(\mathcal{N}) \subseteq \text{nodes}(\mathcal{M} \mid N)$. The second statement can be proved as follows: let $m \in \text{nodes}(\mathcal{M})$. By Equation (A.1) we have that $\Gamma_N \not\vdash m$, so that it remains to show $\Gamma_L \not\vdash m$. This is a trivial consequence of Equation (A.2); in fact, since $m \in \text{nodes}(\mathcal{M})$, we also have $m \in \text{nodes}(\mathcal{M} \mid N)$, and therefore $\Gamma_L \not\vdash m$.

Now suppose that

$$(\text{nodes}(\mathcal{N}) \cap (\Gamma_L)_V) = \emptyset \quad (\text{A.3})$$

$$\text{nodes}(\mathcal{M}) \cap (\Gamma_N \cup \Gamma_L)_V = \emptyset \quad (\text{A.4})$$

We need to show that $\text{nodes}(\mathcal{M}) \cap (\Gamma_N)_V = \emptyset$, and $\text{nodes}(\mathcal{M} \mid N) \cap (\Gamma_L)_V = \emptyset$. The first statement is an immediate consequence of Equation (A.4), by noticing that $(\Gamma_N)_V \subseteq (\Gamma_N \cup \Gamma_L)_V$. For the second statement, let m be a node such that $\Gamma_L \vdash m$. By Equation (A.3) we have that $m \notin \text{nodes}(\mathcal{N})$. Also, by Equation (A.4) it holds that $m \notin \text{nodes}(\mathcal{M})$; in fact, since $\Gamma_L \vdash m$, we also have $\Gamma_N \cup \Gamma_L \vdash m$, and therefore $m \notin \text{nodes}(\mathcal{M})$. Since $m \notin \text{nodes}(\mathcal{M})$ and $m \notin \text{nodes}(\mathcal{N})$, it follows that $m \notin \text{nodes}(\mathcal{M} \mid N)$, as we wanted to prove. \square

Proof of Proposition 4.6: . Let $\mathcal{M} = \Gamma_M \triangleright M$ be a well-formed network, and assume that $\text{nodes}(\mathcal{M}) \neq \emptyset$. That is, there exists a node name m , a state s and a system term N such that $M \equiv m[[s]] \mid N$.

Let $\mathcal{G} = \Gamma_G \triangleright m[[s]]$, where Γ_G is defined by

$$(\Gamma_G)_V = \{m\} \cup \{n \in (\Gamma_M)_V \mid \Gamma_M \vdash m \leftrightarrow n\} \quad (\text{A.5})$$

$$(\Gamma_G)_E = \{(m, n) \mid \Gamma_M \vdash m \rightarrow n\} \cup \{(n, m) \mid \Gamma_M \vdash n \rightarrow m\}. \quad (\text{A.6})$$

Let also $\mathcal{N} = \Gamma_N \triangleright N$, where Γ_N is defined by letting

$$(\Gamma_N)_V = \text{nodes}(\mathcal{N}) \cup \{n \mid n \neq m, \Gamma_M \vdash m' \leftrightarrow n \text{ for some } m' \in \text{nodes}(\mathcal{N})\} \quad (\text{A.7})$$

$$(\Gamma_N)_E = (\Gamma_M)_E \setminus (\Gamma_G)_E \quad (\text{A.8})$$

We need to show the following facts:

- (1) $\mathcal{G} \in \mathbb{G}$,
- (2) $\mathcal{N} \in \text{Nets}$,
- (3) $\text{nodes}(\mathcal{G}) \cap (\mathcal{N})_V = \emptyset$,
- (4) $(\mathcal{M})_V = (\mathcal{G})_V \cup (\mathcal{N})_V$,
- (5) $(\mathcal{M})_E = (\mathcal{G})_E \cup (\mathcal{N})_E$,
- (6) $M \equiv m[[s]] \mid N$.

Each of the statements above is proved separately. Note that (6) follows by the hypothesis.

Proof of Statement 1: $\mathcal{G} \in \mathbb{G}$.

First note that $|\text{nodes}(m[[P]])| = 1$, so that it suffices to show that \mathcal{G} is well-formed. To this end, we show that \mathcal{G} satisfies both the constraints that we have placed over networks and those given in Definition 4.2.

- (1) $m[[s]] \in \text{sSys}$; this is trivial, since no node name can appear more than once in a system term which contains only one node name
- (2) $\text{nodes}(m[[s]]) \subseteq (\Gamma_G)_V$; this statement follows from the definition of $(\Gamma_G)_V$, Equation (A.5), which gives that $\{m\} \subseteq (\Gamma_G)_V$.
- (3) $(\Gamma_G)_E$ is irreflexive; note that $(\Gamma_G)_E \subseteq (\Gamma_M)_E$, and the latter is irreflexive. Therefore, $(\Gamma_G)_E$ has to be irreflexive as well.
- (4) Whenever $\Gamma_G \vdash l \rightleftharpoons k$ for some nodes l, k , then either $l \in \text{nodes}(m[[s]])$ or $k \in \text{nodes}(m[[s]])$. Equivalently, we prove that whenever $\Gamma_G \vdash l \rightleftharpoons k$ for some nodes l, k , then either $l = m$ or $k = m$.

Suppose $\Gamma_G \vdash l \rightleftharpoons k$; then either $\Gamma_G \vdash l \rightarrow k$ or $\Gamma_G \vdash l \leftarrow k$. Due to the arbitrariness of l, k , it is sufficient to consider only the first case.

By Definition of $(\Gamma_G)_E$, Equation (A.6), either $(l, k) = (m, n)$ for some n such that $\Gamma_M \vdash m \rightarrow n$, or $(l, k) = (n, m)$ for some n such that $\Gamma_M \vdash n \rightarrow m$. In the first case we obtain $l = m$, in the second $k = m$, and there is nothing left to prove.

- (5) If $\Gamma_G \vdash n$ and $\Gamma_G \vdash n \rightleftharpoons l$ for no $l \in (\Gamma_G)_V$, then $n \in \text{nodes}(m[[s]])$, or equivalently $n = m$.

Note that if $\Gamma_G \vdash n$ then by Equation (A.5) either $n = m$, in which case there is nothing to prove, or $\Gamma_M \vdash m \rightleftharpoons n$. By Equation A.6 we also have that $\Gamma_G \vdash m \rightleftharpoons n$, which contradicts the hypothesis.

Proof of Statement 2: $\mathcal{N} \in \text{Nets}$.

We need to show that \mathcal{N} satisfies the standard requirements we placed over all networks, plus the requirements required for a network to be well-formed, that is those listed in Definition 4.2

- (1) $N \in \text{sSys}$. By hypothesis we already know that $M \in \text{sSys}$, and since $M \equiv m[[s]] \mid N$, it follows that no node name appears in N more than once.
- (2) $\text{nodes}(N) \subseteq (\Gamma_N)_V$. This follows immediately from Equation (A.7).
- (3) $(\Gamma_N)_E$ is irreflexive; this follows since, by Equation (A.8), $(\Gamma_N)_E \subseteq (\Gamma_M)_E$; the latter is irreflexive by hypothesis.
- (4) Whenever $\Gamma_N \vdash n \rightleftharpoons l$ for some nodes n and l , then either $n \in \text{nodes}(N)$ or $l \in \text{nodes}(N)$. Due to the arbitrariness of node names n, l , it is sufficient to show that the property holds whenever $\Gamma_N \vdash n \rightarrow l$. Let n, l be two nodes such that $\Gamma_N \vdash n \rightarrow l$. Note that by Equation (A.8) we have that $(\Gamma_N)_E \subseteq (\Gamma_M)_E$; since \mathcal{M} is well-formed, it follows that either $n \in \text{nodes}(M)$ or $l \in \text{nodes}(M)$.

However, since $\Gamma_N \vdash n \rightarrow l$, we also have that $\Gamma_N \vdash n$ and $\Gamma_N \vdash l$, Equation (A.7) also ensures that $n, l \neq m$.

Thus either $n \in \text{nodes}(M) \setminus \{m\}$ or $l \in \text{nodes}(M) \setminus \{m\}$; but since $M \equiv n[[s]] \mid N$, and $M \in \text{sSys}$, $\text{nodes}(M) \setminus \{m\}$ is exactly $\text{nodes}(N)$.

- (5) if $\Gamma_N \vdash n$ and $\Gamma_N \vdash n \rightleftharpoons l$ for no $l \in (\Gamma_N)_V$, then $n \in \text{nodes}(N)$. By Equation A.7 either $n \in \text{nodes}(N)$, in which case there is nothing to prove, or there exists $m' \in \text{nodes}(N)$ such that $\Gamma_N \vdash n \rightleftharpoons m'$. But this last case is not possible, since it contradicts the hypothesis.

Proof of Statement 3: $\text{nodes}(G) \cap (\Gamma_N)_V = \emptyset$.

Let $n \in (\Gamma_N)_V$; we need to show that $n \neq m$. By Equation A.7 there are two possible cases:

- (1) $n \in \text{nodes}(N)$, in which case $m = n$ would contradict the hypothesis that $M \equiv m[[s]]|N \in \text{sSys}$, or
 (2) $n \in \{l \mid l \neq m, \Gamma_M \vdash m' \not\rightleftharpoons l \text{ for some } m' \in \text{nodes}(N)\}$; again $n \neq m$.

Proof of Statement 4: $(\Gamma_M)_V = (\Gamma_G)_V \cup (\Gamma_N)_V$.

Note that equations (A.7) and (A.5) ensure that $(\Gamma_N)_V \subseteq (\Gamma_M)_V$ and $(\Gamma_G)_V \subseteq (\Gamma_M)_V$, respectively. Therefore, it is sufficient to show that $(\Gamma_M)_V \subseteq (\Gamma_G)_V \cup (\Gamma_N)_V$.

Suppose that $\Gamma_M \vdash n$. There are two possible cases:

- (1) $n \in \text{nodes}(M)$; Since $M \equiv m[[s]]|N$, either $n = m$, in which case $m \in (\Gamma_G)_V$ by Equation (A.5), or $n \in \text{nodes}(N)$, in which case $n \in (\Gamma_N)_V$ by Equation (A.7),
 (2) $n \in \text{Intf}(M)$; since M is well-formed, there exists a node $l \in \text{nodes}(M)$ such that $\Gamma_M \vdash l \not\rightleftharpoons n$. Then either $l = m$, in which case Equation (A.5) ensures that $n \in (\Gamma_G)_V$, or $l \in \text{nodes}(N)$, in which case $n \in (\Gamma_N)_V$ by Equation (A.7).

Proof of Statement 5: $(\Gamma_M)_E = (\Gamma_G)_E \cup (\Gamma_N)_E$.

This follows immediately from Equation (A.8) and the fact that $(\Gamma_G)_E \subseteq (\Gamma_N)_E$. \square

APPENDIX B. DECOMPOSITION AND COMPOSITION RESULTS

To prove Propositions 5.6 and 5.7, we first need to prove the following statements for actions which can be derived in the intensional semantics:

Proposition B.1 (Weakening). *Let $\Gamma_1 \triangleright M$ be a network, and let Γ_2 be such that $(\Gamma_2)_V \cap \text{nodes}(M) = \emptyset$. Then*

$$\Gamma_1 \triangleright M \xrightarrow{\mu} \Delta \text{ implies } (\Gamma_1 \cup \Gamma_2) \triangleright M \xrightarrow{\mu} \Delta$$

where μ ranges over the actions $m.\tau, m.c!v, m.c?v$.

Proof. By structural induction on the proof of the derivation $\Gamma_1 \triangleright M \xrightarrow{\mu} \Delta$. \square

Proposition B.2 (Strengthening). *Let $\Gamma_1 \triangleright M$ be a network, and let Γ_2 such that $(\Gamma_2)_V \cap \text{nodes}(M) = \emptyset$. Then*

$$(\Gamma_1 \cup \Gamma_2) \triangleright M \xrightarrow{\mu} \Delta \text{ implies } \Gamma_1 \triangleright M \xrightarrow{\mu} \Delta$$

where μ ranges over the actions $m.\tau, m.c!v, m.c?v$.

Proof. By structural induction on the proof of the transition $(\Gamma_1 \cup \Gamma_2) \xrightarrow{\mu} \Delta$. \square

Proof of Proposition 5.6. Let $\mathcal{M} = \Gamma_M \triangleright M$ be a network and $\mathcal{G} = \Gamma_N \triangleright n[[s]]$ be a generating network such that $(\mathcal{M} \# \mathcal{G})$ is defined. We prove only the first statement of Proposition 5.6. The details for the other statements are similar.

Suppose then that $(\mathcal{M} \# \mathcal{G}) \xrightarrow{\tau} \Lambda$. By definition of extensional actions we have two possible cases.

- (1) $(\mathcal{M} \# \mathcal{G}) \xrightarrow{m.\tau} \Lambda$ for some $m \in \text{nodes}(\mathcal{M} \# \mathcal{G}) = \text{nodes}(M) \cup \{n\}$. We perform a case analysis on whether $m \in \text{nodes}(M)$ or $m = n$.
- If $m \in \text{nodes}(M)$ then by Proposition 3.3 we have that $\mathcal{M} \# \mathcal{G} \equiv (\Gamma_M \cup \Gamma_N) \triangleright m[[\tau.p + t]] \mid M' \mid n[[s]]$ for some t, p, M' such that $M \equiv m[[\tau.p + t]] \mid M'$ and $\Lambda \equiv (\Gamma_M \cup \Gamma_N) \triangleright (\mathbb{P}(m[[p]]) \mid \overline{M'} \mid n[[s]])$. Note that if we let $\Delta = (\Gamma_M \triangleright \mathbb{P}(m[[p]]) \mid \overline{M'})$ we can rewrite Λ as $\Delta \# n[[s]]$; further, Proposition 3.3 gives that $\mathcal{M} \xrightarrow{m.\tau} \Delta$, which by definition of extensional actions gives $\mathcal{M} \xrightarrow{\tau} \Delta$.

- If $m = n$ then by Proposition 3.3 we have that $s \equiv \tau.p + t$ for some p, t , while $\Lambda \equiv (\Gamma_M \cup \Gamma_N) \triangleright (\overline{M} \mid n[\mathbb{P}(p)])$. If we let $\Theta = \mathbb{P}(p)$ we can rewrite $\Lambda = (\Gamma_M \triangleright \overline{M}) \sharp (\Gamma_N \triangleright n[\Theta])$. Further, by the definition of the rules in the intensional semantics we have that $s \xrightarrow{\tau} \Theta$.
- (2) $(\mathcal{M} \sharp \mathcal{G}) \xrightarrow{m.c!v} \Lambda$ for some $m \in \text{nodes}(M) \cup \{n\}$ such that $\{l \mid (\Gamma_M \cup \Gamma_N \vdash m \rightarrow l)\} \cap \text{Out}(\mathcal{M} \sharp \mathcal{G}) = \emptyset$. Again we have to consider two different cases.
- $m \in \text{nodes}(M)$. By Proposition 3.5 we have that $(\mathcal{M} \sharp \mathcal{G}) \equiv (\Gamma_M \cup \Gamma_N) \triangleright (m[c!\langle e \rangle . p + t] \mid M' \mid n[s])$ for some e, p, t, M' such that $\llbracket e \rrbracket = v$ and $M \equiv m[c!\langle e \rangle . p + t] \mid M'$, while $\Lambda = (\Gamma_M \cup \Gamma_N) \triangleright \mathbb{P}(m[p]) \mid \Lambda'$ for some Λ' such that $(\Gamma_M \cup \Gamma_N) \triangleright (M' \mid n[s]) \xrightarrow{m.c?v} \Lambda'$. It follows from Proposition 3.4 that $\Lambda' \equiv (\Gamma_M \cup \Gamma_N) \triangleright (\Delta' \mid n[\Theta])$, where Δ' and Θ are such that $(\Gamma_M \cup \Gamma_N) \triangleright M \xrightarrow{m.c?v} \Delta'$ and $(\Gamma_M \cup \Gamma_N) \triangleright n[s] \xrightarrow{m.c?v} n[\Theta]$. Now we can apply Proposition B.2 to the transitions $(\Gamma_M \cup \Gamma_N) \triangleright m[c!\langle e \rangle . p + t] \xrightarrow{m.\tau} \mathbb{P}(m[p])$ and $(\Gamma_M \cup \Gamma_N) \triangleright M' \xrightarrow{m.c?v} \Delta'$ to obtain $\Gamma_M \triangleright m[c!\langle e \rangle . p + t] \xrightarrow{m.c!v} \mathbb{P}(m[p])$ and $\Gamma_M \triangleright M' \xrightarrow{m.c?v} \Delta'$, respectively. These two transitions induce, via an application of rule (B-SYNC), the transition $\Gamma_M \triangleright M \xrightarrow{m.c!v} \Gamma_M \triangleright \mathbb{P}(m[p]) \mid \Delta'$; let $\Delta = \Gamma_M \triangleright \mathbb{P}(m[p]) \mid \Delta'$. The extensional transition induced by the broadcast derived for $\Gamma_M \triangleright M$ can be either an internal action or an extensional broadcast, depending on the topology of Γ_M . First note that, since $\text{Out}(\mathcal{M} \sharp \mathcal{G}) \cap \{l \mid (\Gamma_M \cup \Gamma_N) \vdash m \rightarrow l\} = \emptyset$, we have that $\text{Out}(M) \cap \{l \mid \Gamma_M \vdash m \rightarrow l\} \subseteq \text{nodes}(\mathcal{G}) = \{n\}$. Therefore we have two possible cases
 - If $\Gamma_M \vdash m \rightarrow n$ then we have the transition $\Gamma_M \triangleright M \xrightarrow{\tau} \Delta$. Now note that, since $m \notin (\Gamma_N)_V$, we also have that $(\Gamma_M \cup \Gamma_N) \vdash m \rightarrow n$. Then the transition $(\Gamma_M \cup \Gamma_N) \triangleright n[s] \xrightarrow{m.c?v} n[\Theta]$ could have been derived only via an application of either Rule (B-DEAF) or Rule (B-DISC). In both cases we have $\Theta = \bar{s}$.
 - If $\Gamma_M \vdash m \rightarrow n$ then we have that $\Gamma_M \triangleright M \xrightarrow{c!v \triangleright \{n\}} \Delta$. In this case the transition $(\Gamma_M \cup \Gamma_N) \triangleright n[s] \xrightarrow{m.c?v} n[\Theta]$ could have been derived only via an application of either Rule (B-REC) or Rule (B-REC). In the first case we have that $s \xrightarrow{c?v} \Theta$, while in the second case we obtain that $s \xrightarrow{c?v} \Theta$ and $\Theta = \bar{s}$.
 Finally, it is now easy to show that $\Lambda = (\Gamma_M \cup \Gamma_N) \triangleright (\mathbb{P}(m[p]) \mid \Delta' \mid n[\Theta]) = (\Gamma_M \triangleright \mathbb{P}(m[p]) \mid \Delta') \sharp (\Gamma_N \triangleright n[\Theta]) = \Delta \sharp (\Gamma_N \triangleright n[\Theta])$.
 - $m = n$. In this case we have that $\{l \mid (\Gamma_M \cup \Gamma_N) \vdash n \rightarrow l\} \cap \text{Out}(\mathcal{M} \sharp \mathcal{G}) = \text{Out}(\mathcal{G})$, that is $\text{Out}(\mathcal{G}) = \emptyset$. By Proposition 3.5 we have that $s \equiv c!\langle e \rangle . p + t$ for some e, p, t such that $\llbracket e \rrbracket = v$. This ensures that $s \xrightarrow{c!v} \Theta$, where $\Theta = \mathbb{P}(p)$. Further, $(\Gamma_M \cup \Gamma_N) \triangleright M \xrightarrow{n.c?v} \Delta_M$ for some Δ_M such that $\Lambda \equiv (\Gamma_M \cup \Gamma_N) \triangleright (\Delta_M \mid n[\Theta])$. By applying Proposition B.2 to the last transition we obtain $\Gamma_M \triangleright M \xrightarrow{n.c?v} \Delta$. Whether this intensional transition induces an extensional one depends on the topology Γ_M .
 - if $n \in \text{In}(M)$ then we have the extensional transition $\Gamma_M \triangleright M \xrightarrow{n.c?v} \Delta$,
 - otherwise the transition above does not induce an extensional input. However, in this case it is easy to show, using Proposition 3.4 that $\Delta = \overline{M}$.
 Finally, let $\Delta = \Gamma_M \triangleright \Delta_M$. Note that $\Lambda \equiv (\Gamma_M \cup \Gamma_N) \triangleright (\Delta_M \mid n[\Theta]) = \Delta \sharp (\Gamma_N \triangleright n[\Theta])$.

Lemma B.3 (Strong Composition of tau-actions). *Let \mathcal{M} be a network, and $\mathcal{G} = (\Gamma_N \triangleright n[s])$ be a generating network such that $\mathcal{M} \sharp \mathcal{G}$ is well-defined. If $\mathcal{M} \xrightarrow{\tau} \Delta$ then $\mathcal{M} \sharp (\Gamma_N \triangleright n[s]) \xrightarrow{\tau} \Delta \sharp (\Gamma_N \triangleright n[\bar{s}])$.*

Proof. Let $\mathcal{M} = \Gamma_M \triangleright M$. If $\mathcal{M} \xrightarrow{\tau} \Delta$; then $\Delta = \Gamma_M \triangleright \Delta_M$ for some Δ_M . By definition of extensional tau actions, there are two possibilities:

- (1) $\mathcal{M} \xrightarrow{m,\tau} \Delta_M$ for some $m \in \text{nodes}(M)$. By Proposition B.1 we obtain that $(\Gamma_M \cup \Gamma_N) \triangleright M \xrightarrow{m,\tau} \Delta$, and finally $(\Gamma_M \cup \Gamma_N) \triangleright M \mid n[[s]] \xrightarrow{m,\tau} (\Delta_M \triangleright) \llbracket \bar{s}n \rrbracket$ by Rule (B-TAU – PROP). Note that $(\Gamma_M \cup \Gamma_N) \triangleright (\Delta_M \triangleright) \llbracket \bar{s}n \rrbracket = \Delta \# (\Gamma_N \triangleright n[[\bar{s}]])$;
- (2) $\mathcal{M} \xrightarrow{m,c!v} \Delta_M$, and $\Gamma_M \vdash m \rightarrow l$ for no $l \in \text{Out}(\mathcal{M})$; in particular $\Gamma_M \vdash m \nrightarrow n$, which also gives $(\Gamma_M \cup \Gamma_N) \vdash m \nrightarrow n$. Therefore we can infer the transition $(\Gamma_M \cup \Gamma_N) \triangleright n[[s]] \xrightarrow{m,c!v} n[[\bar{s}]]$. By Proposition B.1 we have that $\Gamma_M \triangleright M \xrightarrow{m,c!v} \Delta_M$ implies $(\Gamma_M \cup \Gamma_N) \triangleright M \xrightarrow{m,c!v} \Delta_M$. Now we can apply Rule (B-SYNC) to obtain the transition $(\Gamma_M \cup \Gamma_N) \triangleright (M \mid n[[s]]) \xrightarrow{m,c!v} (\Gamma_M \cup \Gamma_N) \triangleright (\Delta_M \mid n[[\bar{s}]])$. Note that the last network can be rewritten as $\Delta \# (\Gamma_N \triangleright n[[\bar{s}]])$. Finally, since $\Gamma_M \vdash m \rightarrow l$ for no $l \in \text{Out}(\mathcal{M})$ and $\Gamma_N \not\vdash m$, it follows that $(\Gamma_M \cup \Gamma_N) \vdash n \rightarrow l$ for any $l \in \text{Out}(\mathcal{M} \# (\Gamma_N \triangleright n[[s]]))$. Hence we have the extensional transition $\mathcal{M} \# (\Gamma_N \triangleright n[[s]]) \xrightarrow{\tau} \Delta \# (\Gamma_N \triangleright n[[\bar{s}]])$. \square

Proof of Proposition 5.7. We only prove statements (1)(i), (2)(i) and (2)(ii); details for the other statements are similar.

- (1) Suppose that $\Delta \xrightarrow{\tau} \Delta'$ and let Γ_N, n, s be such that $\Delta \# (\Gamma_N \triangleright n[[\bar{s}]])$ is well-defined. We have to show that $\Delta \# (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta' \# (\Gamma_N \triangleright n[[\bar{s}]])$.

We first prove a weaker result: if $\Delta \xrightarrow{\tau} \Delta'$ then $\Delta \# (\Gamma_N \triangleright n[[\bar{s}]]) \xrightarrow{\tau} \Delta' \# (\Gamma_N \triangleright n[[\bar{s}]])$. To see why this is true, rewrite Δ as $\sum_{i \in I} p_i \cdot \overline{\mathcal{M}_i}$, where $\sum_{i \in I} p_i \leq 1$. Then there exists a collection of distributions $\{\Delta'_i\}_{i \in I}$ such that $\mathcal{M}_i \xrightarrow{\tau} \Delta'_i$ and $\Delta' = \sum_{i \in I} p_i \cdot \Delta'_i$. We can apply Lemma B.3 to each of such transitions to obtain $\mathcal{M}_i \# (\Gamma_N \triangleright n[[s]]) \xrightarrow{\tau} \Delta'_i \# (\Gamma_N \triangleright n[[\bar{s}]])$. It follows that

$$\begin{aligned} \Delta \# (\Gamma_N \triangleright n[[\bar{s}]]) &= \sum_{i \in I} p_i \cdot \overline{\mathcal{M}_i \# (\Gamma_N \triangleright n[[s]])} \\ &\xrightarrow{\tau} \sum_{i \in I} p_i \cdot \overline{\Delta'_i \# (\Gamma_N \triangleright n[[\bar{s}]])} \\ &= \Delta' \# (\Gamma_N \triangleright n[[\bar{s}]]) \end{aligned}$$

Now suppose that $\Delta \xrightarrow{\tau} \Delta'$. Then there exist two collections of sub-distributions $\{\Delta_k^{\rightarrow}\}_{k \geq 0}$ and $\{\Delta_k^{\times}\}_{k \geq 0}$ such that $\Delta' = \sum_{k=0}^{\infty} \Delta_k^{\times}$ and

$$\begin{array}{rcccl} \Delta & = & \Delta_0^{\rightarrow} & + & \Delta_0^{\times} \\ \Delta_0^{\rightarrow} & \xrightarrow{\tau} & \Delta_1^{\rightarrow} & + & \Delta_1^{\times} \\ \vdots & & \vdots & & \vdots \\ \Delta_k^{\rightarrow} & \xrightarrow{\tau} & \Delta_{k+1}^{\rightarrow} & + & \Delta_{k+1}^{\times} \\ \vdots & & \vdots & & \vdots \end{array}$$

For any $k \geq 0$, let $\Theta_k^{\rightarrow} = \Delta_k^{\rightarrow} \# (\Gamma_N \triangleright n[[\bar{s}]])$, and define Θ_k^{\times} analogously. Note that $\Theta_0^{\rightarrow} + \Theta_0^{\times} = \Delta \# (\Gamma_N \triangleright n[[\bar{s}]])$. Also, from the previous statement we can infer that $\Theta_k^{\rightarrow} \xrightarrow{\tau}$

$(\Delta_{k+1}^{\rightarrow} + \Delta_{k+1}^{\times}) \# (\Gamma_N \triangleright n[\bar{s}])$. This last sub-distribution is exactly $\Theta_{k+1}^{\rightarrow} + \Theta_{k+1}^{\times}$. Therefore we have that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{\tau} \sum_{k=0}^{\infty} \Theta_k^{\times}$. It remains to note that

$$\begin{aligned} \sum_{k=0}^{\infty} \Theta_k^{\times} &= \Theta_k^{\times} = \Delta_k^{\times} \# (\Gamma_N \triangleright n[\bar{s}]) \\ &= \left(\sum_{k=0}^{\infty} \Delta_k^{\times} \right) \# (\Gamma_N \triangleright n[\bar{s}]) \\ &= \Delta' \# (\Gamma_N \triangleright n[\bar{s}]) \end{aligned}$$

- (2) Suppose now that $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$ for some η with $n \notin \eta$; we have to show that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$.

First note that, whenever $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta_M$, with $n \notin \eta$ then $\mathcal{M} \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta} \Delta \# (\Gamma_N \triangleright n[\bar{s}])$. We leave the proof of this result to the reader. An immediate consequence of the result above is that whenever $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$ and $s \xrightarrow{c?v} \Theta$, then $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$.

Finally, suppose that $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$, where $n \notin \eta$. We proceed by induction on the definition of weak extensional outputs.

- The base case is $\Delta \xrightarrow{\tau} \Delta_1 \xrightarrow{c!v \triangleright \eta} \Delta_2 \xrightarrow{\tau} \Delta'$; in this case we have that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{\tau} \Delta_1 \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta} \Delta_2 \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{\tau} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$, as we wanted to prove.
- Suppose now that $\Delta \xrightarrow{c!v \triangleright \eta_1} \Delta_1 \xrightarrow{c!v \triangleright \eta_2} \Delta'$, where $\eta_1 \cap \eta_2 = \emptyset$ and $\eta_1 \cup \eta_2 = \eta$. Note that in this case $n \notin \eta_1$ and $n \notin \eta_2$. By inductive hypothesis we have that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta_1} \Delta_1 \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta_2} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$, and there is nothing left to prove.
- Suppose that $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$ for some η such that $\{n\} \subset \eta$. Also, suppose that $s \xrightarrow{c?v} \Theta$. In this case we want to prove that $\Delta \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta'} \Delta' \# (\Gamma_N \triangleright n[\bar{s}])$, where $\eta' = \eta \setminus \{n\}$. The proof of these statements relies on the following technical result, whose proof is left to the reader: if $\mathcal{M} \xrightarrow{c!v \triangleright \eta} \Delta'$ and $s \xrightarrow{c?v} \Theta$, then $\mathcal{M} \# (\Gamma_N \triangleright n[\bar{s}]) \xrightarrow{c!v \triangleright \eta'} \Delta' \# (\Gamma_N \triangleright n[\Theta])$, where $\eta' = \eta \setminus \{n\}$. Then the proof of the main result can be performed as in the previous case, by noting that if the transition $\Delta \xrightarrow{c!v \triangleright \eta} \Delta'$ is induced by $\Delta \xrightarrow{c!v \triangleright \eta_1} \Delta_1 \xrightarrow{c!v \triangleright \eta_2} \Delta'$, where $\eta_1 \cup \eta_2 = \eta$ and $\eta_1 \cap \eta_2 = \emptyset$, then it cannot be $n \in \eta_1$ and $n \in \eta_2$. In this case it is necessary to rely on Proposition 5.7(2)(i), which has already been proved.

REFERENCES

- [1] Bracha and Toueg. Asynchronous consensus and broadcast protocols. *JACM: Journal of the ACM*, 32, 1985.

- [2] Michele Bugliesi, Lucia Gallina, Andrea Marin, Sabina Rossi, and Sardaouna Hamadou. Interference-sensitive preorders for manets. In *Quantitative Evaluation of Systems (QEST), 2012 Ninth International Conference on*, pages 189–198. IEEE, 2012.
- [3] Andrea Cerone. *Foundations of Ad Hoc Wireless Networks*. PhD thesis, School of Computer Science and Statistics, Trinity College Dublin, July 2012. <http://www.scss.tcd.ie/~acerone/works/thesis.pdf>.
- [4] Andrea Cerone and Matthew Hennessy. Modelling probabilistic wireless networks - (extended abstract). In Holger Giese and Grigore Rosu, editors, *FMOODS/FORTE*, volume 7273 of *Lecture Notes in Computer Science*, pages 135–151. Springer, 2012.
- [5] Andrea Cerone, Matthew Hennessy, and Massimo Merro. Modelling mac-layer communications in wireless systems (extended abstract). In Rocco de Nicola and Christen Julien, editors, *Coordination 2013*. Springer.
- [6] Eoin Curran and Jim Dowling. Sample: Statistical network link modelling in an on-demand probabilistic routing protocol for ad hoc networks. In *WONS*, pages 200–205. IEEE Computer Society, 2005.
- [7] Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, and Carroll Morgan. Testing finitary probabilistic processes. In *Proceedings of the 20th International Conference on Concurrency Theory*, volume 5710 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2009. Full-version available from <http://www.scss.tcd.ie/Matthew.Hennessy/onlinepubs.html>.
- [8] Cristian Ene and Traian Muntean. A broadcast-based calculus for communicating systems. In *IPDPS*, page 149. IEEE Computer Society, 2001.
- [9] Cristian Ene and Traian Muntean. Testing theories for broadcasting processes. *Sci. Ann. Cuza Univ*, 11:214–230, 2002.
- [10] Adrian Francalanza and Matthew Hennessy. A theory of system behaviour in the presence of node and link failure. *Information and Computation*, 206(6):711–759, 2008.
- [11] Lucia Gallina, Sardaouna Hamadou, Andrea Marin, and Sabina Rossi. A probabilistic energy-aware model for mobile ad-hoc networks. In *Analytical and Stochastic Modeling Techniques and Applications*, pages 316–330. Springer, 2011.
- [12] Fatemeh Ghassemi, Wan Fokkink, and Ali Movaghar. Restricted broadcast process theory. In Antonio Cerone and Stefan Gruner, editors, *SEFM*, pages 345–354. IEEE Computer Society, 2008.
- [13] Fatemeh Ghassemi, Wan Fokkink, and Ali Movaghar. Equational reasoning on mobile ad hoc networks. *Fundam. Inform.*, 105(4):375–415, 2010.
- [14] Fatemeh Ghassemi, Wan Fokkink, and Ali Movaghar. Verification of mobile ad hoc networks: An algebraic approach. *Theor. Comput. Sci*, 412(28):3262–3282, 2011.
- [15] Fatemeh Ghassemi, Mahmoud Talebi, Ali Movaghar, and Wan Fokkink. Stochastic restricted broadcast process theory. In Nigel Thomas, editor, *EPEW*, volume 6977 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2011.
- [16] Jens Chr. Godskesen. A calculus for mobile ad hoc networks. In Amy L. Murphy and Jan Vitek, editors, *COORDINATION*, volume 4467 of *Lecture Notes in Computer Science*, pages 132–150. Springer, 2007.
- [17] Hennessy and Rathke. Bisimulations for a calculus of broadcasting systems. *TCS: Theoretical Computer Science*, 200, 1998.
- [18] Matthew Hennessy. *A distributed Pi-calculus*. Cambridge University Press, 2007.
- [19] Matthew Hennessy and Tim Regan. A process algebra for timed systems. *Information and Computation*, 117(2):221–239, March 1995.
- [20] Raja Jurdak, Cristina Videira Lopes, and Pierre Baldi. A survey, classification and comparative analysis of medium access control protocols for ad hoc networks. *IEEE Communications Surveys and Tutorials*, 6(1-4):2–16, 2004.
- [21] Ivan Lanese and Davide Sangiorgi. An operational semantics for a calculus for wireless systems. *Theor. Comput. Sci*, 411(19):1928–1948, 2010.
- [22] Ruggero Lanotte and Massimo Merro. Semantic analysis of gossip protocols for wireless sensor networks. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR*, volume 6901 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2011.
- [23] Massimo Merro and Eleonora Sibilio. A timed calculus for wireless systems. In Farhad Arbab and Marjan Sirjani, editors, *FSEN*, volume 5961 of *Lecture Notes in Computer Science*, pages 228–243. Springer, 2009.
- [24] R. Milner. A calculus of communicating systems. *LNCS*, 92, 1980.
- [25] Sebastian Nanz and Chris Hankin. Static analysis of routing protocols for ad-hoc networks, March 25 2004.
- [26] Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. *TCS: Theoretical Computer Science*, 367, 2006.
- [27] Rocco De Nicola and Matthew Hennessy. Testing equivalences for processes. *Theor. Comput. Sci*, 34:83–133, 1984.

- [28] Prasad. A calculus of broadcasting systems. *SCIPROG: Science of Computer Programming*, 25, 1995.
- [29] S. Rahamatkar, A. Agarwal, and N. Kumar. Analysis and comparative study of clock synchronization schemes in wireless sensor networks. *Analysis*, 2(3):536–541, 2010.
- [30] Julian Rathke and Pawel Sobocinski. Deconstructing behavioural theories of mobility. In *Proc. Fifth IFIP International Conference On Theoretical Computer Science (TCS)*, volume 273 of *IFIP*, pages 507–520. Springer, 2008.
- [31] D. Sangiorgi and D. Walker. *The π -calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.
- [32] Anu Singh, C. R. Ramakrishnan, and Scott A. Smolka. A process calculus for mobile ad hoc networks. *Sci. Comput. Program*, 75(6):440–469, 2010.
- [33] Lei Song. *Probabilistic Models and Process Calculi for Mobile Ad Hoc Networks*. PhD thesis, IT University of Copenhagen, Programming, Logic and Semantics, 2012.
- [34] Lei Song and Jens Godskesen. Probabilistic mobility models for mobile and wireless networks. In Cristian Calude and Vladimiro Sassone, editors, *Theoretical Computer Science*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 86–100. Springer Boston, 2010.
- [35] Andrew S. Tanenbaum. *Computer networks*. P T R Prentice-Hall, pub-PHPTR:adr, fourth edition, 2003.
- [36] Mengying Wang and Yang Lu. A timed calculus for mobile ad hoc networks. *arXiv preprint arXiv:1301.0045*, 2013.