Symbolic protocol verification with dice: process equivalences in the presence of probabilities (extended version)
Vincent Cheval, Raphaëlle Crubillé, Steve Kremer

To cite this version:
Vincent Cheval, Raphaëlle Crubillé, Steve Kremer. Symbolic protocol verification with dice: process equivalences in the presence of probabilities (extended version). 2023. hal-03683907v2

HAL Id: hal-03683907
https://inria.hal.science/hal-03683907v2
Preprint submitted on 30 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Symbolic protocol verification with dice: process equivalences in the presence of probabilities∗
(extended version)

Vincent Cheval  Raphaëlle Crubillé  Steve Kremer
Inria Paris  Aix-Marseille Université  Université de Lorraine
LIS, CNRS  Inria Nancy Grand-Est & LORIA

Abstract

Symbolic protocol verification generally abstracts probabilities away, considering computations that succeed only with negligible probability, such as guessing random numbers or breaking an encryption scheme, as impossible. This abstraction, sometimes referred to as the perfect cryptography assumption, has shown very useful as it simplifies automation of the analysis. However, probabilities may also appear in the control flow where they are generally not negligible. In this paper we consider a framework for symbolic protocol analysis with a probabilistic choice operator: the probabilistic applied pi calculus. We define and explore the relationships between several behavioral equivalences. In particular we show the need to require randomized schedulers – indeed we exhibit a counter-example to one of the main results in a previous work that relied on non-randomized ones. As in other frameworks that mix both non-deterministic and probabilistic choices, schedulers may sometimes be unrealistically powerful. We therefore consider two subclasses of processes that avoid this problem. When considering purely non-deterministic protocols, as is done in classical symbolic verification, we show that a probabilistic adversary has—maybe surprisingly—a strictly superior distinguishing power for may testing, which, when the number of sessions is bounded, we show to coincide with purely possibilistic similarity. Finally, we consider fully probabilistic protocols and show that trace equivalence corresponds to a notion of may testing with purely probabilistic attackers. We also briefly discuss complexity and automation for these subclasses when the number of sessions is bounded.

∗This work has been partly supported by the ANR Research and teaching chair in AI ASAP with support from the region Grand Est and by France 2030 program managed by ANR (ANR-22-PECY-0006).
Contents

1 Introduction 5

1 A Probabilistic Applied Pi Calculus 7

2 Probabilistic Applied $\pi$-calculus 7
   2.1 Message as terms 7
   2.2 Syntax of the process calculus 8
   2.3 Operational semantics 9

3 May Testing Equivalences 11
   3.1 Resolving the internal non-determinism 11
   3.2 Computing the probability of reaching a barbed state 16
   3.3 Defining May Testing Equivalence 17

4 Trace equivalence 17
   4.1 Labelled semantics 18
   4.2 Trace Preorder 19
   4.3 On randomized or non-randomized resolutions 19

5 Simulation and observational preorders 21
   5.1 Simulation preorder 23
   5.2 Observational preorder 25
   5.3 Bisimilarity and observational equivalence do not coincide with non-randomized
       schedulers 26
   5.3.1 Counter-example to coincidence of bisimulation and observational
       congruence in [GLPT07] 30
   5.3.2 The counter-example for a conservative variant of [GLPT07] bisim-
       ulation 35

II Well behaved subclasses of protocols 39

6 Non-Probabilistic Processes 40
   6.1 May-testing with non-probabilistic adversary and trace equivalence coincide 40
   6.2 May-testing and simulation coincide for bounded processes 41
       6.2.1 Hennessy-Milner logical characterization of strong simulation 42
       6.2.2 Complexity 44
   6.3 May-testing and simulation do not coincide for unbounded processes 44
### G Non probabilistic processes

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.1 Hennessy-Milner’s Logical characterisation</td>
<td>129</td>
</tr>
<tr>
<td>G.2 Proposition 6</td>
<td>133</td>
</tr>
</tbody>
</table>

### H Fully probabilistic processes

136
1 Introduction

Automated symbolic protocol verification, based on the seminal work of Dolev and Yao [DY83], has nowadays reached a level of maturity enabling successful use on complex real-world security protocols, including TLS [BBK17, CHH+17], authentication protocols of the 5G standard [BDH+18], or EMV’s secure payment protocols [BST21] to name only a few. In the symbolic model, a non-deterministic, computationally unbounded attacker is assumed to have complete control of the network, being able to intercept any messages, and forge new ones. As a counterpart, cryptography is idealized and the attacker can only use predefined rules to manipulate messages that are represented by terms, e.g. expressed by an equation $\text{dec}(\text{enc}(m, k), k) = m$ stating that a message $m$ encrypted with $k$ can be decrypted with the same key. This treatment of cryptography is in opposition to computational models where we assume a probabilistic polynomial time attacker, messages are represented by bitstrings and assumptions that an arbitrary such attacker has at most negligible probability of breaking a cryptographic primitive. Similarly, in the symbolic model, random values, such as keys or nonces, are chosen freshly from an infinite domain, rather than chosen randomly from a sufficiently large domain. These symbolic abstractions of cryptography and randomness have even been shown sound [CKW10] (under rather strong assumptions) and significantly ease the automation of proofs. Hence, symbolic modeling of messages is arguably useful for formally analyzing cryptographic protocols.

However, the above-described abstractions of randomness only apply to the messages, and not to the control flow. Typical examples which crucially rely on randomized control flow are mechanisms for providing anonymity, such as the dining cryptographers protocol [Cha88], Mix-nets [Cha81] or Crowds [RR98]. In this paper, we will investigate indistinguishability properties, expressed as equivalences in a cryptographic process calculus, the applied pi calculus [ABF17], extended with a probabilistic choice operator. Typically, the testing equivalence expresses that two processes are equivalent if they exhibit the same behaviour when put in parallel with an arbitrary attacker process. Our work presents foundations for a model that (i) extends the scope of symbolic protocol analysis to probabilistic protocols, and (ii) allows to consider non-probabilistic protocols in the presence of a probabilistic attacker. In particular, when we consider purely concurrent processes–without probabilistic behavior–the equivalence we obtain is strictly stronger than the standard testing equivalence on such purely concurrent processes; in other terms, probabilistic adversaries are–for good reasons, as we will argue–more powerful in order to distinguish such processes than the purely concurrent adversaries considered in existing works and tools.

Our contributions. In a first part we introduce a probabilistic applied pi calculus and its semantics, which has similarities to [GLPT07], with two major differences. (i) We express our semantics in terms of general non-deterministic probabilistic transition systems (NPLTS)–also called probabilistic automata in the literature–which allows us to benefit from a large body of existing results on these systems [SL95, PS07, Cas18, BSV19, DVGHM09, Sto02, Eis17]. (ii) More importantly, we differ in the way non-determinism is
resolved: unlike GLPT07 we allow for randomized schedulers—rather than choosing one particular non-deterministic choice, we allow the scheduler to choose an arbitrary distribution on the available non-deterministic choices. Next we introduce testing equivalence, and show that unlike in the purely non-deterministic choice, may-testing is strictly stronger than trace equivalence. Finally, we introduce notions of (bi)simulation and observational preorder and equivalence and show that the simulation and observational pre-order, respectively bisimilarity and observational equivalence, coincide for randomized schedulers. We also show that for non-randomized schedulers, these equivalences do not coincide, which provides a counter-example to one of the main results in GLPT07.

A well-known phenomenon CP10 AAPvR10 in process calculi that are both probabilistic and non-deterministic is the existence of some nonrealistic schedulers that are able to use the internal probabilistic choices done by an agent in order to schedule another agent’s non-deterministic choices, i.e., the scheduler leaks the probabilistic choices. Therefore, we study two important subclasses of processes that avoid this phenomenon. First, we consider the classical class of non-probabilistic processes, as in the original applied pi calculus, but in the presence of probabilistic adversaries. We show that, if we additionally bound the number of sessions, may-testing with probabilistic adversaries coincides with the classical, purely possibilistic notion of similarity. This provides a contextual characterization of the notion of similarity which is reminiscent of DVGHM09 in the setting of CSP. Relying on results from CKR18, this implies that testing equivalence with probabilistic adversaries is co-NEXPTIME complete for a large class of cryptographic primitives. Second, we consider a class of fully probabilistic processes, which is reminiscent of a probabilistic version of simple processes in CCD13 CC08, and a slight generalization of the processes in CSV17. We next show that trace equivalence as considered in CSV17, is weaker than may-testing, but coincides with a version of may-testing where attacker processes are restricted disallowing replication, parallel, and non-deterministic choice, but allowing probabilistic choices. Finally, we briefly discuss how the algorithm for deciding trace equivalence in the DeepSec verifier CKR18 could be adapted to this fully probabilistic case, providing a more general setting than Bauer et al. BCSV18 who additionally bound the size of input messages. A thorough treatment of this adaptation and implementation is left for future work. Most of our results are summarized in Figure 1 that depicts the relationship between the different preorders (with similar relations between corresponding equivalences).
Part I

A Probabilistic Applied Pi Calculus

In this first part we present a probabilistic extension of the applied pi calculus. We first introduce its syntax and internal semantics. Then we introduce different equivalences and preorders, and study their relations. We start with the classical may-testing equivalence, which requires to be precise about how non-determinism is resolved. Then we introduce a labelled semantics and the notion of trace equivalence. Unlike in the purely non-deterministic case, trace equivalence is (strictly) weaker than may testing. Finally, we introduce other equivalences and preorders: similarity, bisimilarity and observational preorder/equivalence. We emphasize the need for randomized schedulers; indeed, the use of non-randomized schedulers results into a non-transitive simulation relation.

2 Probabilistic Applied $\pi$-calculus

In this paper, we will work with probabilistic applied $\pi$-calculus, the probabilistic variant of applied $\pi$-calculus introduced by Goubault-Larrecq et al. [GLPT07].

2.1 Message as terms

We assume an infinite set of names, denoted $\mathcal{N} = \{a, b, \ldots\}$. We partition the set $\mathcal{N}$ into two disjoint infinite sets $\mathcal{N}_{pub}$ and $\mathcal{N}_{priv}$ to respectively represent public and private names. The distinction between public and private names is analogous to the distinction between free and bound names in the original applied pi calculus. We define an infinite set of variables $\mathcal{X}$. Finally we consider a finite set of function symbols with their arity
\( F = \{ f/n, g/m, \ldots \} \), called signature. Terms are defined as names, variables, and function symbols applied to other terms. For any \( F \subseteq F, N \subseteq N \) and \( V \subseteq X \), the set of terms built from \( N \) and \( V \) by applying function symbols in \( F \) is denoted by \( \mathcal{T}(F, N \cup V) \). We call protocol terms the terms from \( \mathcal{T}(F, N \cup X) \).

We also suppose that terms are equipped with a binary relation \( \equiv \) that expresses that two terms evaluate to the same result, and a predicate \( \text{Msg}(\cdot) \) that is intended to hold when evaluation succeeds. We suppose that \( \equiv \) is a symmetric and transitive relation that is closed under substitution of terms for names and variables, as well as application of function symbols and such that for all \( a, b \in N \), \( a = b \) if and only if \( a \equiv b \). \( \text{Msg}(\cdot) \) is supposed to hold on any names, to be closed under renamings and \( t_1 \equiv t_2 \) implies that \( \text{Msg}(t_1) \) and \( \text{Msg}(t_2) \). Finally, we require that \( \text{Msg}(t) \) implies \( t \equiv t \).

For example, the \( \equiv \) relation could capture that \( \text{dec}(\text{enc}(m, k), k) \equiv m \) for any \( m, k \) modelling that decryption cancels out encryption when the same key \( k \) is used and one may define \( \text{Msg}(\text{dec}(n, k)) \) as false to express that decryption could fail if the ciphertext it is applied to is not an encryption with the matching key.

How \( \equiv \) and \( \text{Msg}(\cdot) \) are precisely defined is not relevant for the results of this paper and we wish to capture several formalisms. \( \equiv \) can for instance be defined by an equational theory, as in the applied pi calculus [ABF17] (where \( \text{Msg}(\cdot) \) would evaluate to true on any term), defined by a constructor-destructor rewrite system, allowing evaluation to fail when a destructor application does not reduce, as in the DeepSec tool [CKR18], or a combination of these as in the ProVerif tool [Bla16].

### 2.2 Syntax of the process calculus

The syntax for processes is defined as follows:

\[
P, Q ::= 0 \quad \text{nil}
\]
\[
in(u, x); P \quad \text{output}
\]
\[
out(u, v); P \quad \text{input}
\]
\[
P \parallel Q \quad \text{parallel composition}
\]
\[
!P \quad \text{replication}
\]
\[
\text{new } a; P \quad \text{restriction}
\]
\[
\text{if } u = v \text{ then } P \text{ else } Q \quad \text{conditional}
\]
\[
P + Q \quad \text{non-deterministic choice}
\]
\[
P + _p Q \quad \text{probabilistic choice}
\]

where \( u, v \) are protocol terms, \( x \in X, a \in N \) and \( p \in ]0; 1[ \). A process \( P \) is closed when all variables in \( P \) are bound by an input.

**Notation 1.** We write \( \mathcal{SP} \) for the set of all processes in probabilistic applied \( \pi \)-calculus, and \( \mathcal{MP} \) for the set of all multisets over \( \mathcal{SP} \). We sometimes consider syntactic subclasses of these processes. For finite processes, i.e., without replication, we denote these sets by
\(SP^{<\infty}\) and \(MP^{<\infty}\). For non probabilistic, i.e. purely non-deterministic processes without \(\ast\), we write \(SP^{np}\) and \(MP^{np}\) and when considering both restrictions we write \(SP^{<\infty, np}\) and \(MP^{<\infty, np}\).

2.3 Operational semantics

We will now define the semantics of the probabilistic applied \(\pi\)-calculus. We opt for a different presentation of the semantics than Goubault-Larrecq et al. [GLPT07] relying on existing formalisms for transition systems. Moreover, we allow for a more general class of schedulers.

Notation 2. Let \(S\) be any set. We denote by \(\mathcal{D}(S)\) the set of all finitely supported probability distributions over \(S\). We denote by \(\mathcal{D}^{\leq 1}(S)\) the set of all sub-probability distributions over \(S\) (observe that \(\mathcal{D}(S) \subseteq \mathcal{D}^{\leq 1}(S)\)). For \(p, q \geq 0\), and \(D, E\) two sub-distributions, we write \(p \cdot D + q \cdot E\) for the measure defined as \((p \cdot D + q \cdot E)(x) = p \cdot D(x) + q \cdot E(x)\). In the particular case where \(q = 0\), the resulting sub-distribution does not depend on \(E\), and we simply write \(p \cdot D\) instead of \(p \cdot D + 0 \cdot E\).

If \(D \in \mathcal{D}(S)\), we denote by \(\text{supp}(D)\) the support of \(D\), i.e. the set of all elements \(s \in S\) such that \(D(s) > 0\). If \(S' \subseteq S\), we note \(D(S') = \sum_{s \in S'} D(s)\). Finally, we denote by \(\delta_x\) the Dirac distribution on \(x\).

The operational semantics of processes is defined by a relation between multisets of processes and probability distributions on multisets of processes, denoted \(\mathcal{P} \rightarrow_\tau \mu\). This relation is defined in Figure 2.

Remark 1. One may note that our calculus offers a non-deterministic choice operator that is resolved internally. This differs from the standard \(\pi\)-calculus [MPW92] where the non-deterministic choice operator is resolved externally. Even though the original applied \(\pi\) calculus [ABF17] does not contain non-deterministic choice the process \(P + Q\) can be encoded as

\[
\text{new } c; ((\text{out}(c, \text{tok}); P) | (\text{in}(c, \text{tok}); Q))
\]

where \(c\) and \(x\) do not occur in \(P\) nor in \(Q\). Alternatively, we could have defined \(\ast\) as syntactic sugar.

In the following, we will deal with the operational semantics of our calculus with tools designed for generic probabilistic systems. We choose the notion of non-deterministic probabilistic labelled transition systems (NPLTS) used for instance in [BSV19]. A NPLTS allows to represent states that have both internal and external non-deterministic behavior. It can be noted that it coincides with the notion of simple probabilistic automata of Segala et al. [SL95].

Notation 3. Let \(S\) be any set. We denote \(\mathcal{P}(S)\) for the set of all finite subsets of \(S\). Moreover, we can augment \(S\) with a special element \(\ast\) will be used to talk about deadlocks, i.e. \(S \cup \{\ast\}\).
\( \mathcal{P} \cup \{ \{ 0 \} \} \rightarrow_\tau \delta \mathcal{P} \) (Null)
\( \mathcal{P} \cup \{ \text{if } u = v \text{ then } P \text{ else } Q \} \rightarrow_\tau \delta_{\mathcal{P} \cup \{ P \}} \) if \( u \neq v \) (Then)
\( \mathcal{P} \cup \{ \text{if } u = v \text{ then } P \text{ else } Q \} \rightarrow_\tau \delta_{\mathcal{P} \cup \{ Q \}} \) (Else)
\( \mathcal{P} \cup \{ \text{out}(u, t).P, \text{in}(v, x).Q \} \rightarrow_\tau \delta_{\mathcal{P} \cup \{ P, Q \}} \) if \( \text{Msg}(t) \land u \neq v \) (Comm)
\( \mathcal{P} \cup \{ \text{out}(u, t).P, \text{in}(v, x).Q \} \rightarrow_\tau \delta_{\mathcal{P} \cup \{ Q \}} \) if \( u \neq v \) (Choice-1)
\( \mathcal{P} \cup \{ \text{new } a; P \} \rightarrow_\tau \delta_{\mathcal{P} \cup \{ a \rightarrow t \}} \) where \( a \in \mathcal{N}_{\text{priv}} \) is fresh (New)
\( \mathcal{P} \cup \{ P + Q \} \rightarrow_\tau \delta_{\mathcal{P} \cup \{ Q \}} \) (Choice-2)
\( \mathcal{P} \cup \{ P + p, Q \} \rightarrow_\tau p \cdot \delta_{\mathcal{P} \cup \{ P \}} + (1 - p) \cdot \delta_{\mathcal{P} \cup \{ Q \}} \) (PChoice)

Figure 2: Semantics of the calculus

**Definition 1** (NPLTS). A NPLTS is a triple \((\mathcal{S}, \mathcal{A}, \text{trans})\), where

- \( \mathcal{S} \) is a set of states,
- \( \mathcal{A} = \mathcal{A}_{\text{int}} \sqcup \mathcal{A}_{\text{ext}} \) is a set of labels, and
- \( \text{trans} : \mathcal{S} \rightarrow \mathcal{A} \rightarrow \mathcal{P}(\mathcal{D}(\mathcal{S})) \) is a transition function: for each state in \( \mathcal{S} \), and each label in \( \mathcal{A} \), \( \text{trans}(s)(a) \) is a set of (finitely supported) distributions.

The labels in \( \mathcal{A}_{\text{int}} \), respectively \( \mathcal{A}_{\text{ext}} \) are called internal, respectively external actions. For \( s \in \mathcal{S}, a \in \mathcal{A} \), when \( D \in \text{trans}(s, a) \) we write \( s \overset{a}{\rightarrow} D \) if \( a \in \mathcal{A}_{\text{ext}} \) and \( s \overset{\tau}{\rightarrow} D \) if \( a \in \mathcal{A}_{\text{int}} \).

In the rest of this paper, we may define a NPLTS by only its transition function, i.e., we will say that \( \text{trans} : \mathcal{S} \rightarrow \mathcal{A} \rightarrow \mathcal{P}(\mathcal{D}(\mathcal{S})) \) is the NPLTS \((\mathcal{S}, \mathcal{A}, \text{trans})\). When considering an NPLTS with only one internal action, this action will usually be noted \( \tau \), in such a way that \( s \overset{\tau}{\rightarrow} D \) if and only if \( D \in \text{trans}(s, \tau) \).

We now express our operational semantics as a NPLTS, whose only action is the internal action \( \{ \tau \} \). The need for external actions will become clear in Section 4 when defining a labelled semantics as a NPLTS \( \mathcal{N}^\ell \) with an internal action \( \tau \), and several external actions that represent adversarial transitions. Moreover, in Section ??, we consider a refinement of both labelled and non-labelled semantics, using several internal actions: an internal action will correspond to an internal action \( \tau \) decorated with an annotation visible by the scheduler.

**Definition 2.** The operational semantics is the NPLTS \( \mathcal{N}^\circ = (\mathcal{M}\mathcal{P}, \{ \tau \}, \text{trans}^\circ) \) where for every \( s \in \mathcal{M}\mathcal{P}, \text{trans}^\circ(s)(\tau) = \{ D \mid s \overset{\tau}{\rightarrow} D \} \).

Note that the states of the NPLTS \( \mathcal{N}^\circ \) contain all possible multisets of processes and how they are executed. Obviously, \( \mathcal{N}^\circ \) is thus an infinite transition system. In examples
illustrating transitions of a multiset of processes $\mathcal{P}$, we only show the relevant fragment of $\mathcal{N}$ that contains $\mathcal{P}$.

**Example 1.** The complete execution of the following process $P$ is given in Figure 3.

\[
\text{(out}(c, k) +_{1/3} \text{out}(c, a)) \mid \text{in}(c, x); \text{if } x = k \text{ then out}(c, ok)
\]

\begin{figure}[h]
\centering
\begin{tikzpicture}[node distance=2cm, auto]
  \node (P) {$\{P\}$};
  \node (A_B) [below of=P] {$\{A, B\}$};
  \node (out_c_k) [right of=P] {$\{\text{out}(c, k'), B\}$};
  \node (out_c_a) [below of=out_c_k] {$\{\text{out}(c, a), B\}$};
  \node (0_B_3) [right of=out_c_k] {$\{0, B_3\}$};
  \node (0_B_2) [right of=out_c_k] {$\{0, B_2\}$};
  \node (0_B_1) [below of=out_c_k] {$\{0, B_1\}$};
  \node (0_0) [right of=0_B_1] {$\{0, 0\}$};
  \node (B_1) [above of=0_B_1] {$\{B_1\}$};
  \node (B_2) [right of=0_B_2] {$\{B_2\}$};
  \node (B_3) [right of=0_B_3] {$\{B_3\}$};

  \path[->] (P) edge node {$\tau$} (A_B);
  \path[->] (A_B) edge [in=200, out=160, distance=2cm] node {$\tau$} (out_c_k);
  \path[->] (out_c_k) edge [in=200, out=160, distance=2cm] node {$\tau$} (0_B_3);
  \path[->] (out_c_k) edge node {$1/3$} (0_B_2);
  \path[->] (out_c_k) edge node {$2/3$} (out_c_a);
  \path[->] (out_c_a) edge node {$\tau$} (0_B_1);
  \path[->] (0_B_1) edge node {$\tau$} (0_B_2);
  \path[->] (0_B_2) edge node {$\tau$} (B_2);
  \path[->] (0_B_3) edge node {$\tau$} (B_3);
  \path[->] (B_1) edge node {$\tau$} (B_2);
  \path[->] (B_2) edge node {$\tau$} (B_3);
  \path[->] (B_3) edge node {$\tau$} (0_0);
  \path[->] (B_1) edge node {$\tau$} (0_B_2);
  \path[->] (B_2) edge node {$\tau$} (0_B_3);
  \path[->] (B_3) edge node {$\tau$} (0_0);

\end{tikzpicture}
\end{figure}

$A = \text{out}(c, k) +_{1/3} \text{out}(c, a)$

$B = \text{in}(c, x); \text{if } x = k' \text{ then out}(c, ok)$

$B_1 = \text{if } a = k' \text{ then out}(c, ok)$

$B_2 = \text{if } k' = k \text{ then out}(c, ok)$

$B_3 = \text{out}(c, ok)$

Figure 3: Semantics of the process $P$ from Example 1

# 3 May Testing Equivalences

## 3.1 Resolving the internal non-determinism

We generically express the way to resolve the internal non-determinism of a NPLTS using the formalism of *resolutions*: resolving the non-determinism means choosing from the NPLTS, a particular transition system without internal non-determinism. The resulting transition system is called a *Reactive Probabilistic Labelled Model* and has still external, but no internal, non-determinism. It can be noted that this model is equivalent to Labelled Markov Chains when extended with internal actions.

**Definition 3 (RPLTS).** A Reactive Probabilistic Labelled Transition System (RPLTS) is a triple $(S, \mathcal{A}, \text{trans})$, where

- $S$ is a set of states,

- $\mathcal{A} = \mathcal{A}_{\text{ext}} \sqcup \mathcal{A}_{\text{int}}$ a set of labels, and
\textbf{trans} : \mathcal{S} \rightarrow (\mathcal{A}_{\text{ext}} \rightarrow (\mathcal{D}(\mathcal{S}) \cup \{\star\})) \sqcup (\mathcal{A}_{\text{int}} \times \mathcal{D}(\mathcal{S})) \text{ is a transition function that assigns to each state in } \mathcal{S}:

\begin{itemize}
  \item either a function mapping labels in \mathcal{A}_{\text{ext}} to a failure or a distribution over \mathcal{S} (the non deterministic external actions);
  \item or one unique distribution with its internal label from \mathcal{A}_{\text{int}} (the deterministic internal action).
\end{itemize}

The states \( s \in \mathcal{S} \) such that \textbf{trans}(s) : \mathcal{A}_{\text{ext}} \rightarrow (\mathcal{D}(\mathcal{S}) \cup \{\star\}) \) are called \textit{external states}, while the ones such that \textbf{trans}(s) : (\mathcal{A}_{\text{int}} \times \mathcal{D}(\mathcal{S})) \) are called \textit{internal states}. Given an RPLTS \( R \), we will denote by \( \mathcal{S}_{\text{ext}}(R) \) and \( \mathcal{S}_{\text{int}}(R) \) the sets of external and internal states of \( R \) respectively.

In the particular case of a NPLTS with no external action, resolving the internal non-determinism results in a system with \textit{no} non-determinism at all. This is for instance the case of the operational semantics \( \mathsf{No} \). Such a purely probabilistic system can be modelled as a reactive probabilistic transition system (that is equivalent to the notion of Markov Chain).

\textbf{Definition 4} (RPTS). A Reactive Probabilistic Transition System (RPTS) is a pair \((\mathcal{S}, \textbf{trans})\), where

\begin{itemize}
  \item \( \mathcal{S} \) is a set of states, and
  \item \textbf{trans} : \mathcal{S} \rightarrow (\mathcal{D}(\mathcal{S}) \cup \{\star\}) \text{ is a transition function that assigns to each state in } \mathcal{S} \text{ either}
  \begin{itemize}
    \item a failure token \( \star \), or
    \item a distribution over \( \mathcal{S} \).
  \end{itemize}
\end{itemize}

We now define \textit{resolutions}–or schedulers–for a NPLTS that allow to solve the \textit{internal}, but not external, non-determinism: a resolution describes \textit{one} of the possible ways of turning an NPLTS into a RPLTS. It means that for each state \( s \), a resolution should choose whether \( s \) is an internal state or external state; in the first case, a \textit{unique} pair of an internal action and a post-transition distribution must be chosen; in the second case, for each external action \( a \), the resolution must choose a \textit{unique} distribution \( D \) (if it exists) such that \( s \rightarrow_a D \). It is however slightly more complicated, since the possible existence of cycles in the execution–meaning that from the same state, the scheduler can do successively different choices–lead to the necessity of a \textit{correspondance function}.

\textbf{Example 2}. There are indeed cycles in the NPLTS \( \mathsf{No} \), because of the replication operator \( ! \). For instance, we can consider the following process \( P \):

\[
P := !Q \quad \text{with} \quad Q := \text{in}(u,x) \mid (\text{out}(u,0) + \text{out}(u,1)).
\]

We represent in Figure 4 a fragment of \( \mathsf{No} \) containing the state \( \{\{P\}\} \). (Observe that this fragment is not self-contained, in the sense that there exist \( \mathsf{No} \)-transitions from states in this fragment to states absent from this fragment).
Figure 4: A cycle in \( N^0 \), from Example 2

**Notation 4.** Let \( S \) be a set of states and \( \Delta \subseteq D(S) \) be a finite set of probability distributions over \( S \). The *convex hull* of \( \Delta \) is defined as:

\[
\text{conv}(\Delta) := \{ D \in D(S) \mid \exists \alpha_1, \ldots, \alpha_n \in \mathbb{R} \exists D_1, \ldots, D_n \in \Delta. \sum_{i=1}^n \alpha_i = 1 \land D = \sum_{i=1}^n \alpha_i \cdot D_i \}
\]

**Notation 5.** Let \( S, S' \) be two sets of states and \( f : S \to S' \). We define the function \( \bar{f} : D(S) \to D(S') \) to be the probabilistic lifting of \( f \), where \( \bar{f}(D) = \sum_{s \in S} D(s) \cdot \delta_{f(s)} \).

When obvious from context, we will overload the notation and write \( f \) instead of \( \bar{f} \).

**Definition 5** ([BSV19]). Let \( \text{trans} : S \to \mathcal{A}_{\text{int}} \sqcup \mathcal{A}_{\text{ext}} \to \mathcal{P}(D(S)) \) be a NPLTS. A randomized, respectively non-randomized, resolution for \( \text{trans} \) is a triple \( (S', \text{corr}, \text{trans}') \), where \( S' \) is a set of states, \( \text{corr} : S' \to S \) is the correspondence function, and \( \text{trans}' : S' \to (\mathcal{A}_{\text{ext}} \to (D(S') \cup \{\star\})) \sqcup (\mathcal{A}_{\text{int}} \times D(S')) \) is a RPLTS \( R \) such that:

- if \( s' \in S_{\text{ext}}(R) \) and \( \text{trans}'(s')(a) = D \), or \( s' \in S_{\text{int}}(R) \) and \( \text{trans}'(s') = (a, D) \),

then \( \text{corr}(D) \in \text{conv}(\text{trans}(\text{corr}(s'))(a)) \) (randomized), respectively

\( \text{corr} \) is injective on the support of \( D \) and \( \text{corr}(D) \in \text{trans}(\text{corr}(s'))(a) \) (non-randomized).

Given a NPLTS \( \mathcal{N} \) we denote by \( \mathcal{R}_r(\mathcal{N}) \), respectively \( \mathcal{R}_n(\mathcal{N}) \), the set of randomized, respectively non-randomized resolutions. Observe that any non-randomized resolution is also a randomized resolution, i.e., \( \mathcal{R}_n(\mathcal{N}) \subseteq \mathcal{R}_r(\mathcal{N}) \).

**Remark 2.** In the particular case of an NPLTS \( \mathcal{N} \) that has only one internal action \( \tau \), and no external actions, as for instance \( \mathcal{N}^0 \), a resolution is a triple \( R = (S', \text{corr}, \text{trans}') \), where \( S' \) is a set of states, \( \text{corr} : S' \to S \) is the correspondence function, and \( \text{trans}' : S' \to (\emptyset \to (D(S') \cup \{\star\})) \sqcup (\{\tau\} \times D(S')) \). Recall that for any set \( X \), the cardinality of the set \( (\emptyset \to X) \) is 1: by abuse of notation, we denote by \( \star \) the unique element of the set.
(a) A NPLTS $N$

(b) A non-randomized resolution for $N$
$corr(s_i^1) = s_1 \ (1 \leq i \leq 3), \ corr(s_j^2) = s_j \ (2 \leq j \leq 3)$.

(c) A randomized resolution for $N$
$corr(s_i) = s_i \ (1 \leq i \leq 3)$
$(\emptyset \to (D(S') \cup \{\star\}))$. With this notation—and by indentifying any set $X$ to the isomorphic set $\{\tau\} \times X$—we can simply rewrite $\mathsf{trans}'$ as:

$$\mathsf{trans}' : S' \to \{\star\} \sqcup D(S').$$

To sum up, there are two possibilities when we start from a R-state $s' \in S'$: either the execution stops here (in this case, $\mathsf{trans}(s') = \star$, and thus $s'$ is an external state), or the scheduler chooses the next step (in this case, $\mathsf{trans}(s') \in D(S')$, and thus $s'$ is an internal state). Formally, it means that the \(\mathbb{N}\)-resolutions are actually RPTSs—{}the purely probabilistic structure defined in Definition 4. Moreover, we can rewrite the simplified constraints on resolutions given in Definition 5 as follows:

if $\mathsf{trans}'(s') = D$

then $\mathsf{corr}(D) \in \text{conv}(\mathsf{trans}(\mathsf{corr}(s'))(\tau))$ (randomized), respectively

$\mathsf{corr}$ is injective on the support of $D$ and $\mathsf{corr}(D) \in \mathsf{trans}(\mathsf{corr}(s'))(\tau)$ (non-randomized).

We show in Figure 6 examples of \(\mathbb{N}\)-resolutions for the process $P$ from Example 1 and its semantics in Figure 3.

(a) A non-randomized resolution.

(b) A randomized resolution.

Figure 6: Examples of resolutions for process $P$ from Example 1 where the correspondence function $\mathsf{corr}$ is the identity. Resolutions in \(\mathbb{N}\) correspond to a Markov Chain.
3.2 Computing the probability of reaching a barbed state

We first precise when a state of \( N^o \), i.e., a multiset of processes, exhibits a barb \( c \).

**Definition 6.** For any public name \( c \in N_{pub} \), and a multiset of processes \( P \in MP \), we say that \( P \) exhibits barb \( c \) when there exists a process \( \text{out}(u,t).Q \) in \( P \) where \( c = u \) and \( \text{Msg}(t) \). We denote by \( \downarrow c \) the set of all multisets of processes that exhibit the barb \( c \).

We formalize here the probability of reaching a state in some set of target states, in a transition system, i.e., in a transition system where all non-determinism—internal or external—has already been resolved. We first define the probability of reaching such a state in at most \( n \) steps, and then we take the probability of reaching them eventually as the limit of the \( n \)-step reaching probabilities.

**Definition 7.** Let \( R = (S, \text{trans}) \) be a RPTS, \( T \subseteq S \) a set of states, and \( s \) an initial state. For every \( n \in \mathbb{N} \) we define the probability of reaching \( T \) from \( s \) in at most \( n \) steps as:

\[
\text{RProb}^\leq n(s, T) = \begin{cases} 
1 & \text{if } s \in T \\
0 & \text{otherwise.}
\end{cases}
\]

\[
\text{RProb}^\leq n+1(s, T) = \begin{cases} 
1 & \text{if } s \in T \\
0 & \text{if } s \notin T \land \text{trans}(s) = * \\
\sum_{u \in \text{supp}(D)} D(u) \cdot \text{RProb}^\leq n(u, T) & \text{if } s \notin T \land \text{trans}(s) = D
\end{cases}
\]

We define the probability of reaching \( T \) from \( s \) as:

\[
\text{RProb}_R(s, T) = \lim_{n \to +\infty} \text{RProb}^\leq n(s, T).
\]

Note that, as \( \text{RProb}^\leq n(s, T) \) is an increasing function in \( n \) we can replace the limit by the supremum on \( n \in \mathbb{N} \).

**Notation 6.** Let \( N = (S_N, A, \text{trans}_N) \) be a NPLTS, and \( R = (S_R, \text{corr}_R, \text{trans}_R) \) be a resolution for \( N \). Recall that \( \bar{R} = (S_R, A, \text{trans}_R) \) is a RPLTS. From \( \bar{R} \), we build a RPTS by removing the external actions: we define \( \bar{R} = (S_{\bar{R}}, \text{trans}_{\bar{R}}) \) where

\[
\text{trans}_{\bar{R}}(s) = \begin{cases} 
* & \text{when } s \in S_{\text{ext}}(\bar{R}) \\
D & \text{when } s \in S_{\text{int}}(\bar{R}) \text{ and } \text{trans}_R(s) = (a_{\text{int}}, D)
\end{cases}
\]

For \( T \subseteq S_R \), \( s \in S_R \) and resolution \( R \) we write \( \text{RProb}_R(s, T) \), respectively \( \text{RProb}^\leq_R(n, s, T) \), for the probability \( \text{RProb}_R(s, T) \), respectively \( \text{RProb}_R^\leq_R(n, s, T) \).

For \( T \subseteq S_N \), \( s \in S_N \), and \( R \subseteq R_t(N) \), we note \( \text{RProb}_R(s, T) \), respectively \( \text{RProb}_R^\leq_R(n, s, T) \), for the probability \( \sup\{\text{RProb}_R(s', \text{corr}_R^{-1}(T)) \mid R = (S_R, \text{corr}_R, \text{trans}_R) \in R, \text{corr}_R(s') = s\} \), respectively \( \sup\{\text{RProb}_R^\leq_R(n, s', \text{corr}_R^{-1}(T)) \mid R = (S_R, \text{corr}_R, \text{trans}_R) \in R, \text{corr}_R(s') = s\} \).

In the following we denote by \( R^o_t \) (resp. \( R^o_{nr} \)) the set of all randomized (resp. non-randomized) resolutions for \( N^o \), i.e., \( R^o_t = R_t(N^o) \) and \( R^o_{nr} = R_{nr}(N^o) \).
3.3 Defining May Testing Equivalence

Definition 8 (May testing equivalence). Let $\mathcal{R}$ be a set of resolutions for $N^o$ and $P, Q \in MP$. We say that $P \leq^R_{may} Q$ iff

$$\forall Adv \in MP \text{ s.t. } fn(Adv) \subseteq N_{pub}, \forall c \in N_{pub}, RProb_R(\mathcal{P} \cup Adv, \downarrow c) \leq RProb_R(\mathcal{Q} \cup Adv, \downarrow c)$$

We say that $P, Q$ are may testing equivalent for $\mathcal{R}$, denoted $P \approx^R_{may} Q$ when both $P \leq^R_{may} Q$ and $Q \leq^R_{may} P$.

We first show that for may-testing, it is sufficient to consider non-randomized resolutions.

Proposition 1. The may testing preorder coincides for randomized and non-randomized resolutions:

$$\leq^{R^o}_{may} = \leq^{R^o}_{nr_{may}}$$

The proof of this result is given in Appendix A.

Given that the may testing preorder coincides for randomized and non-randomized resolutions we will simply write $\leq_{may}$ in what follows.

Remark 3. One alternative definition of probabilistic may testing pre-orders studied in the literature is as follows: $P \leq' Q$ iff

for all $Adv \in MP$ s.t. $fn(Adv) \subseteq N_{pub}, c \in N_{pub}$, resolution $R$ and $R$-state $s$ s.t. $corr(s) = P \mid Adv$,

there exists resolution $R'$ and $R'$-state $s'$ such that $corr'(s') = Q \mid Adv$ and

$$RProb_R(s, corr^{-1}(\downarrow c)) = RProb_R(s', corr'^{-1}(\downarrow c))$$

However, the resulting pre-orders on processes are counter-intuitive: for instance, when we consider the processes $P := \text{out}(a, 0)$ and $Q := \text{out}(a, 0) +_{1/2} (\text{out}(a, 0) +_{1/2} \text{out}(a, 0))$, we can show that $(Q, P) \notin \leq'$, for both randomized or non-randomized schedulers. Indeed, when we take $Adv = 0$, and we see that there exists a resolution $R$ such that $RProb_R(Q, \downarrow a) = \frac{1}{2}$, but for every resolution $R'$, it holds that $RProb_{R'}(P, \downarrow a) = 1$.

4 Trace equivalence

As usual in pi calculi, and in the applied \pi-calculus in particular we define a labelled semantics. The intent of the labels is to capture adversarial actions and avoid the universal quantification over processes in equivalence definitions.
4.1 Labelled semantics

A state in this labeled semantics is defined by associating a multiset of processes with a frame, modeling the knowledge accumulated by the adversary. We also consider a new set of variables $\mathcal{AX} = \{a_1x_1, a_2x_2, \ldots\}$ distinct from $\mathcal{X}$ that will act as pointers to output messages.

**Definition 9.** An extended process is an ordered pair $(P, \phi)$, where $P \in \mathcal{MP}$ and $\phi$ is a ground substitution $\{a_1x_1 \mapsto t_1; \ldots; a_nx_n \mapsto t_n\}$ such that $a_i \in \mathcal{AX}$, $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{N})$ and $\text{Msg}(t_i)$ for $1 \leq i \leq n$.

We denote by $\mathcal{SP}_\ell$ the set of all extended processes (and use $\mathcal{SP}_\ell^{<\infty}$, $\mathcal{SP}_\ell^{np}$ and $\mathcal{SP}_\ell^{<\infty, np}$ for subclasses as before).

A *recipe* is a term from $\mathcal{T}(\mathcal{F}, \mathcal{N}_{pub} \cup \mathcal{AX})$ representing how an attacker can deduce a message.

**Notation 7.** If $D$ is a distribution over $\mathcal{MP}$, and $\phi$ a frame, we write $(D, \phi)$ for the distribution over extended processes defined as $(D, \phi) = \sum_{P \in \text{supp}(D)} D(P) \cdot \delta(P, \phi)$.

We now define the NPLTS $\mathcal{N}^\ell$ for the labelled semantics. The only internal action is $\{\tau\}$, and we add *external actions*, that model the interactions with an external environment.

**Definition 10.** The labelled semantics is the NPLTS $\mathcal{N}^\ell = (\mathcal{SP}_\ell, \mathcal{A}_{ext}^\ell \cup \{\tau\}, \text{trans}^\ell)$ where

- $\mathcal{A}_{ext}^\ell = \{\text{in}(\xi, \zeta), \text{out}(\xi, ax), (\xi \equiv \zeta), (\xi \neq \zeta) \mid \xi, \zeta \text{ recipes}, ax \in \mathcal{AX}\}$

- $\text{trans}^\ell((P, \phi))(a) = \{D \mid (P, \phi) \rightarrow_a D\}$ where $\rightarrow_a$ is defined by the following rules:

  - $(P, \phi) \rightarrow_\tau (D, \phi)$ if $P \rightarrow_\tau D$

  - $(\{\text{in}(u, x); P\} \cup P, \phi) \rightarrow_{\text{in}(\xi, \zeta)} \delta(\{P(\xi / \zeta)\} \cup P, \phi)$ if $u \equiv \xi \phi, \text{Msg}(\zeta \phi)$ and $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi)$

  - $(\{\text{out}(u, t); P\} \cup P, \phi) \rightarrow_{\text{out}(\xi, ax_{n+1})} \delta(\{P\} \cup P, \phi(ax_{n+1} \mapsto t))$ if $u \equiv \xi \phi, \text{Msg}(t), \text{vars}(\xi) \subseteq \text{dom}(\phi)$ and $|\phi| = n$

  - $(P, \phi) \rightarrow_{(\xi \sim \zeta)} \delta(P, \phi)$ if $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi)$ and $\xi \phi \sim \zeta \phi$ where $\sim \in \{\equiv, \neq\}$

Note that when we lift $\rightarrow_\tau$ to extended processes we suppose that the freshness requirement of a new name $a'$ in the (NEW) rule of fig. 2 also applies to the frame $\phi$, i.e., $a'$ must not appear in $\phi$. As for the operational semantics, we consider randomized and non-randomized resolutions for $\mathcal{N}^\ell$ and denote the corresponding sets of resolutions by $\mathcal{R}_r^\ell$, respectively $\mathcal{R}_{nr}^\ell$.

**Remark 4.** It should be noted that we deal with static equivalence in a different way as done usually in the applied pi-calculus, or implicitly in the probabilistic applied pi-calculus in [GLPT07]: indeed, static equivalence is encoded into the NPLTS $\mathcal{N}^\ell$ by a countable set of actions— all the tests $(\xi \equiv \zeta)$ and their negations— instead of just one action testing static
equivalence. The motivation behind this choice is to be able to represent every action from the NPLTS by an elementary push-button action of the adversary. As shown later, this choice has no effect whatsoever on the definition of the simulation pre-orders (or on bisimulation)–see Remark 8 for more details, but it leads to a different notion of trace equivalence, that is closer to may testing equivalence–see Remark 5.

4.2 Trace Preorder

We first define the probability of executing a trace in a resolution. As we are interested in a weak trace preorder, traces are sequences of external actions only. In our definition we will use the previously introduced notation $R\text{Prob}_R(s, \{t\})$: recall that this denotes the probability of reaching state $t$ from state $s$ under resolution $R$ using only internal actions.

Definition 11. Let $N = (S_N, A_{int} \sqcup A_{ext}, \text{trans}_N)$ be a NPLTS. Let $R = (S_R, \text{corr}_R, \text{trans}_R)$ be a resolution for $N^\ell$. Let $w \in A_{ext}^*$ be a trace, i.e., a finite word on the alphabet $A_{ext}$. For all states $s \in S_R$, we define the probability of executing $w$ starting from $s$ in $R$ as:

$$
\begin{align*}
\text{Prob}_R(s, \epsilon) &= 1 \\
\text{Prob}_R(s, a.w) &= \sum_{t \in S_R} R\text{Prob}_R(s, \{t\}) \cdot \sum_{s' \in \text{supp}(D)} D(s') \cdot \text{Prob}_R(s', w)
\end{align*}
$$

Given a set of resolutions $\mathcal{R} \subseteq \mathcal{R}_t(N)$, a state $u \in S_N$, we define by $\text{Prob}_\mathcal{R}(s, w)$ the maximal probability of executing $w$ from $s$ in $\mathcal{R}$ as:

$$
\text{Prob}_\mathcal{R}(u, w) = \sup_{R=(S_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}} \text{Prob}_R(s, w)
$$

In this work, we are interested mostly by the may paradigm for non-determinism. Observe that nonetheless, we could define must-trace equivalence where we would compute the infimum of probabilities rather than supremum of probabilities in Definition 11.

Definition 12. Let $\mathcal{R} \subseteq \mathcal{R}_t^\ell$ be a set of resolutions for $N^\ell$ and $(\mathcal{P}, \phi), (\mathcal{P}', \phi') \in \mathcal{SP}_t^\ell$.

$$(\mathcal{P}, \phi) \leq_{\text{tr}} (\mathcal{P}', \phi') \iff \forall w \in A_{ext}^*, \text{Prob}_\mathcal{R}((\mathcal{P}, \phi), w) \leq \text{Prob}_\mathcal{R}((\mathcal{P}', \phi'), w)$$

$(\mathcal{P}, \phi)$ and $(\mathcal{P}', \phi')$ are trace equivalent for $\mathcal{R}$ if $(\mathcal{P}, \phi) \leq_{\text{tr}} (\mathcal{P}', \phi')$ and $(\mathcal{P}', \phi') \leq_{\text{tr}} (\mathcal{P}, \phi)$.

4.3 On randomized or non-randomized resolutions

Observe that Definition 12 depends on the class of admissible resolutions under consideration, i.e., whether we restrict ourselves to non-randomized resolutions, or we consider all randomized resolutions. In the following, we show–as we formally state in proposition 2–that these two notions of trace equivalence actually coincide. This result is not surprising: indeed it has for instance been shown in [Cas18] (in the restricted setting of strong
Proposition 2. Let \((\mathcal{P}, \phi)\) and \((\mathcal{P}', \phi')\) be two extended processes.

\[
(\mathcal{P}, \phi) \leq_{tr}^R (\mathcal{P}', \phi') \quad \text{iff} \quad (\mathcal{P}, \phi) \leq_{tr}^{R_{nr}} (\mathcal{P}', \phi')
\]

The proof of this result is given in Appendix B.1. In the rest of this paper, as \(\leq_{tr}^{R_{nr}} = \leq_{tr}^R\), we simply denote trace inclusion by \(\leq_{tr}\).

Trace preorders are (strictly) weaker than the may testing preorder: it is the sense of Proposition 3 below.

Proposition 3. Let \(P, Q \in \mathcal{MP}\) be two processes.

\[
P \leq_{may} Q \Rightarrow (P, \emptyset) \leq_{tr} (Q, \emptyset)
\]

Moreover, there exist processes \(P, Q \in \mathcal{MP}\) such that

\[
(P, \emptyset) \leq_{tr} (Q, \emptyset) \quad \text{and} \quad P \not\leq_{may} Q
\]

Proof sketch. We show both points separately.

- Our goal is to show that every trace \(w\) can be emulated by an adversary \(Adv\). Observe that any trace \(w\) can be seen as a (linear) formula in \(\mathcal{F}\), where \(\mathcal{F}\) is the Hennessy-Milner logic from Definition 25. It means that for every \(ok \in \mathcal{N}_{pub}\) such that \(ok \notin fn(w)\) and \(n \in \mathbb{N}\), we can consider the adversary \(Adv_{ok}^{w,n}\) from Definition 27. Recall that we cannot apply directly Lemma 12, because here there can be probabilistic choices inside \(P\). Nonetheless, as long as the formulas considered are traces, we are going to show a generalization of Lemma 12 to probabilistic processes. More precisely, we are going to show: for all trace \(w\), for all \(P \in \mathcal{MP}\), \(\phi \in \mathcal{F}\) with \(ok \notin fn(P, \phi, w)\):

\[
\text{Prob}_{\mathcal{R}_{\text{nl}}}(P, \phi, w) = \text{RProb}_{\mathcal{R}_{\text{nl}}}(P \cup \{Adv_{w,n}^{ok} \mid \phi\}, \downarrow \downarrow ok).
\]

We write \(I\) for the set

\[
I := \{(P \cup \{Adv_{w,n}^{ok} \mid \phi\}) \mid (P, \phi) \in SP_\ell, w \in A^*_{\text{ext}}, ok \notin fn(P, \phi, w)\}
\]

Observe that we can write the restriction of \(\mathcal{N}^\circ\) to \(I\) using the NPLTS \(\mathcal{N}^{\ell_{tr}}\) defined in Definition 44 in the sense where: \(\forall s = P \cup \{Adv_{w,n}^{ok} \mid \phi\} \in I: s \xrightarrow{\mathcal{N}^{\ell_{tr}}} D\), if and only if there exists \(E\) such that \((P, \phi, w) \xrightarrow{\mathcal{F}} \mathcal{N}^{\ell_{tr}} E\), and \(D = \mathcal{F}(E)\) where \(f : ((P', \phi'), w') \in \mathcal{S}_{\mathcal{N}^{\ell_{tr}}} \mapsto \mathcal{P}' \cup \{Adv_{w',n}^{ok} \mid \phi'\} \in I\); see Notation 5 for the meaning of \(\mathcal{F}\). Moreover, a
$$\mathcal{P} = \{ \text{new } k; (P(0) \mid P(0) \mid P(1) \mid P_{\text{dec}}) \}$$ and $$P(x) = \text{new } r; \text{out}(c, \text{enc}(x, r, k))$$

$$\mathcal{Q} = \{ \text{new } k; (P(0) \mid P(0) \mid P(1) \mid P_{\text{dec}}) \}$$

Figure 7: Processes $\mathcal{P}, \mathcal{Q}$ such that $(\mathcal{P}, \emptyset) \leq_{\text{tr}} (\mathcal{Q}, \emptyset)$ and $\mathcal{P} \not\leq_{\text{may}} \mathcal{Q}$

The fact that may testing is strictly stronger is for instance witnessed by processes $\mathcal{P}, \mathcal{Q}$ in Figure 7.

Remark 5. Observe that—contrary to usual results in the literature, see for instance [CCD13]—Proposition 3 holds even for processes non image-finite. This discrepancy comes from our choice of labelled actions for static equivalence (see Remark 4): a trace cannot test directly static equivalence, but can only do a finite numbers of recipe tests.

5 Simulation and observational preorders

In this section, we define the simulation and observational preorders on probabilistic pi-calculus processes and corresponding equivalences. Our definition of simulation preorder is taken from the definition of randomized weak simulation preorder introduced by Segala and Lynch for probabilistic automata [SL95]. Both simulation and observational preorders rely on a weak relation for internal actions which we define as follows, one for randomized and one for non-randomized resolutions.

Definition 13. Let $N = (S_N, A_{\text{ext}} \sqcup A_{\text{int}}, \text{trans}_N)$ be a NPLTS, and $D, E \in D^{\leq 1}(S_N)$.

- We write $D \xrightarrow{\tau} R_{\tau}(N) E$ when there exists $R = (S_R, \text{corr}_R, \text{trans}_R) \in R_{\tau}(N)$, and $D', E' \in D^{\leq 1}(S_R)$ such that $\text{corr}_R(D') = D$, $\text{corr}_R(E') = E$, $\text{supp}(E') \subseteq S_{\text{ext}}(R)$ and for all $u \in S_{\text{ext}}(R)$, $E'(u) = \sum_{s' \in S_R} D'(s') \cdot \text{RProb}_R(s', \{u\})$. 
We write $D \xrightarrow{\tau} \mathcal{R}_r(N)$ $E$ when there exists $R = (\mathcal{S}_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}_r(N)$, and $D', E' \in D \subseteq (\mathcal{S}_R)$ such that $\text{corr}_R$ is injective on the support of $D'$, $\text{corr}_R(D') = D$, $\text{corr}(E') = E$, $\text{supp}(E') \subseteq \mathcal{S}_\text{ext}(R)$ and for all $u \in \mathcal{S}_\text{ext}(R), E'(u) = \sum_{s' \in \mathcal{S}_R} D'(s') \cdot \text{RProb}_R(s', \{u\})$.

Moreover, for all $s \in \mathcal{S}_N$, $D \in D \subseteq (\mathcal{S}_N)$, and $\mathcal{R} \in \{\mathcal{R}_r(N), \mathcal{R}_\text{nr}(N)\}$, we write $s \xrightarrow{\tau} \mathcal{R} D$ when $\delta_s \xrightarrow{\tau} \mathcal{R} D$.

Observe that in the NPLTS $N^\ell$, even though $s \xrightarrow{\tau} D$ implies that $D$ has finite support, it is possible to have $s \xrightarrow{\tau} \mathcal{R} E$, where the sub-distribution $E$ has infinite support.

As we will see below, the non-randomized weak relation $\xrightarrow{\tau} \mathcal{R}_r(N)$ does not allow to define a robust simulation preorder. However, before defining the simulation preorder, we give a more direct characterization of the weak relation $\xrightarrow{\tau} \mathcal{R}_r(N)$, without using the notion of resolutions. This characterization is more widely used in the literature on probabilistic bisimulation—see e.g. [DVGHM09]—and eases proofs. First, we lift a relation $\rightarrow \subseteq \mathcal{S} \times \mathcal{D}(\mathcal{S})$ to a relation in $\mathcal{D}(\mathcal{S}) \times \mathcal{D}(\mathcal{S})$.

**Definition 14.** Let $\mathcal{S}$ be a discrete set and $\rightarrow \subseteq \mathcal{S} \times \mathcal{D}(\mathcal{S})$ be a probabilistic reduction relation. We define its lifting $\rightarrow_r \subseteq \mathcal{D} \subseteq (\mathcal{S}) \times \mathcal{D} \subseteq (\mathcal{S})$ as

$$D \rightarrow_r \sum_{i \in I} \alpha_i \cdot E_i$$

when $I$ is any countable set, $D = \sum_{i \in I} \alpha_i \cdot E_i$, and $x_i \rightarrow E_i$ for all $i \in I$ (observe that it can be that $x_i = x_j$ even when $i \neq j$).

Observe that we can apply Definition 14 above to any NPLTS by considering its probabilistic reduction relation $\rightarrow$. We will do this implicitly in the following.

We are now ready to give an alternative characterisation for the weak reduction relation $\xrightarrow{} \mathcal{R}_r$, more in line with the tradition in the field of probabilistic simulation. There are a number of equivalent definitions in the literature: we chose here the one given in [DVGHM09].

**Definition 15.** Let $\rightarrow$ be a binary relation in $\mathcal{S} \times \mathcal{D}(\mathcal{S})$ for a discrete set $\mathcal{S}$. We define the relation $\xrightarrow{} \mathcal{R}_r \subseteq \mathcal{D} \subseteq (\mathcal{S}) \times \mathcal{D} \subseteq (\mathcal{S})$ as

$$D \xrightarrow{} \mathcal{R}_r D'$$

if there is an infinite scheme:

$$D \xrightarrow{} \mathcal{R}_r D_0 \xrightarrow{} \mathcal{R}_r D_1 \xrightarrow{} \mathcal{R}_r D_2 \xrightarrow{} \mathcal{R}_r \cdots$$

where $D_0, D_1, D_2, \ldots$ are sub-distributions over $\mathcal{S}$ and $D' = \sum_{k \in \mathbb{N}} D_k$.

When starting from the transition relation $\xrightarrow{}$ for a NPLTS $N$, we denote the resulting relations by $\xrightarrow{\tau} \mathcal{R}_r N$. When $N$ is clear from the context, we may directly write $\xrightarrow{\tau} \mathcal{R}_r$. 

22
Remark 6. Whenever we have a finite scheme in a similar shape as the infinite scheme asked for in Definition 15, observe that we can always complete it into an infinite scheme, using the fact that $\emptyset \to \emptyset + \emptyset$–where $\emptyset$ stands for the sub-distribution with empty support, i.e. $\emptyset(x) = 0$ for every $x \in S$. In particular, if we start from a state $(P, \phi)$ in $N^\ell$ where $P$ is a process without replication, then every infinite scheme $\delta(P, \phi) \xrightarrow{\tau} D$ is actually a finite scheme. A simple corollary to this remark is that $\xrightarrow{\tau} r$ is reflexive (by taking $D_0^\top = D$ and all other distributions being $\emptyset$).

We now show in the following statement that the two characterizations, i.e. $\xrightarrow{\tau} r, N$ and $\xrightarrow{\tau} r, N$, are equal (proof in Appendix C).

**Lemma 1.** Let $N = (S_N, A_{\text{ext}} \sqcup A_{\text{int}}, \text{trans}_N)$ be a NPLTS and $D, E \in D_{\leq 1}(S_N)$.

$$D \xrightarrow{\tau} r E \text{ if and only if } D \xrightarrow{\tau} r, E$$

**Remark 7.** The weak reduction relation $\xrightarrow{\tau} r, N$ (and so equivalently $\xrightarrow{\tau} r, (N^\ell)$) allows to lose some probabilities in the presence of a replication in our extended processes. Consider the extended process $A = (\{!P\} \cup P, \phi)$. We have that $\delta_A \xrightarrow{\tau} r D$ where $\text{supp}(D) = \{A\}$, $D(A) = p$ and $p$ can be any value smaller than 1. To show this, we can build the following infinite scheme:

$$\delta_A = D_0 + D$$

and for all $k \in \mathbb{N}$, $D_k \xrightarrow{\tau} D_{k+1} + \emptyset$

with for all $k \in \mathbb{N}$, $\text{supp}(D_k) = \{A_k\}$, $A_k = \{\ldots {!P} ; P ; \ldots P} \cup P, \phi$ and $D_k(A_k) = 1 - p$.

On the other hand, as noted in Remark 6 when processes do not contain replication, $\delta((P, \phi) \xrightarrow{\tau} r D$ actually corresponds to a finite scheme. Thus, we cannot lose probabilities, i.e. $D$ is a distribution.

### 5.1 Simulation preorder

We are now going to define the notions of probabilistic simulation preorder, similarity and bisimilarity. However, in these definitions, we will only consider the relation $\xrightarrow{\tau} r$ (or equivalently $\xrightarrow{\tau} r, N$) and not $\xrightarrow{\tau} r, N$. First, as highlighted for instance in [Eis17, Den05], when considering bisimilarity or similarity on NPLTSs in general, we are forced to consider randomized resolutions, otherwise the resulting relations are not transitive. Note that in [GLPT07], they do rely on non-randomized schedulers in their definition of bisimulation. It does not necessarily mean that their relation is not transitive as they focus directly on the semantics of processes. However, we will show in Section 5.3 that bisimulation and observational equivalence do not coincide when using the non-randomized relation $\xrightarrow{\tau} r, N$, hence disproving [GLPT07, Theorem 2]. This reenforces our belief that it is preferable to use randomized schedulers in our definition.

To define the simulation preorder, we need to introduce the lifting of a relation to sub-distributions as follows.
**Definition 16** (Lifting of a relation). Let $R$ be a binary relation on a discrete set $S$. We define the **lifting of $R$ to sub-distributions** as the binary relation on $D^{\leq 1}(S)$, denoted $\tilde{R}$, defined as:

$$D \tilde{R} E \iff \forall S' \subseteq S, D(S') \leq E(R(S'))$$

where $R(S') = \{ s \in S \mid s' \in S' \land s' R s \}$.

We note that this lifting has the following properties:

- if $R$ is reflexive, then $\tilde{R}$ is reflexive as well;
- if $R$ is symmetric then for any $D_1, D_2 \in D(S)$, i.e., $D_1, D_2$ are distributions rather than sub-distributions, if $D_1 \tilde{R} D_2$ then $D_2 \tilde{R} D_1$.

The second property is actually not obvious on the current definition. It is however a direct consequence of an alternative, *coupling-based* characterization of lifting, that can be found under the name of *Strassen’s theorem* for instance in [ABB+18] (it is stated there for proper distributions, instead of sub-distributions, but the generalisation to sub-distributions can be easily done by lifting all sub-distributions $D$ over $X$ to proper distributions $D\perp$ over $X \cup \{\bot\}$ where $\bot$ is an arbitrary new element added to represent divergence).

**Notation 8.** Let $N = (S_N, A_{ext} \sqcup A_{int}, \text{trans}_N)$ be a NPLTS. Let $a \in A_{ext}$ and $D, D' \in D(S_N)$. We write $D \overset{a}{\Rightarrow} D'$ when $D \overset{\tau}{\Rightarrow} E_1, E_1 \overset{a}{\Rightarrow} E_2$ and $E_2 \overset{\tau}{\Rightarrow} D'$ for some $E_1, E_2$.

Similarly, for $a_1, \ldots, a_n \in A_{ext}$ then we write $D \overset{a_1 \ldots a_n}{\Rightarrow} D'$ when $D \overset{a_1}{\Rightarrow} E_1 \overset{a_2}{\Rightarrow} \ldots \overset{a_n-1}{\Rightarrow} E_{n-1} \overset{a_n}{\Rightarrow} D'$.

**Definition 17.** Let $N = (S_N, A_{ext} \sqcup A_{int}, \text{trans}_N)$ be a NPLTS. A relation $R \subseteq (S_N \times S_N)$ is

- a **simulation** if $s_1 R s_2$ implies that for all $a \in A_{ext} \cup \{\tau\}$, $D_1 \in D(S_N)$

  $$\text{if } s_1 \overset{a}{\Rightarrow} D_1 \text{ then } s_2 \overset{a}{\Rightarrow} D_2, D_2 \in D(S_N) \text{ and } D_1 \tilde{R} D_2$$

- a **bisimulation** if $R$ is a simulation and $R$ is symmetric.

The simulation preorder, denoted $\leq_{\text{sim}}^N$, is the largest simulation and bisimilarity, denoted $\approx_{\text{bs}}^N$, is the largest bisimulation. We define similarity, denoted $\approx_{\text{sim}}^N$, as $\leq_{\text{sim}}^N \cap \leq_{\text{sim}}^{-1}^N$.

As usual in the field of (bi)simulation, it can be shown that $\leq_{\text{sim}}^N$, respectively $\approx_{\text{bs}}^N$, exists [Eis17] and that it is a pre-order, i.e. a reflexive and transitive relation, respectively an equivalence, i.e., a reflexive, symmetric and transitive relation [Sto02].

The following proposition from [Sto02] states that, as usual in the non-probabilistic case, the weak arrow $\overset{a}{\Rightarrow}$ can replace the single arrow $\overset{a}{\Rightarrow}$ in the definition of weak simulation. It is important to observe that this is not the case when we choose *non-randomized* schedulers (see Section 5.3).
Proposition 4. Let \( N = (S_N, A_{\text{ext}} \cup A_{\text{int}}, \text{trans}_N) \) be a NPLTS, and \( R \) be the largest binary relation on \( S_N \) such that \( s_1 R s_2 \) implies that for every \( a \in A_{\text{ext}} \cup \{\tau\} \), if \( s_1 \overset{a}{\rightarrow} D_1 \) then \( s_2 \overset{a}{\rightarrow} D_2 \) and \( D_1 \overset{\tau}{\rightarrow} D_2 \). We have \( R = \leq_{N}^{\text{sim}} \).

Remark 8. Please observe that our choice of labelled actions for static equivalence (see Remark 4) has no impact on the resulting simulation and bisimulation. Indeed, if two \( N' \)-states \( (P, \phi) \) and \( (Q, \psi) \) are in the simulation pre-order or bisimulation, then \( \phi \) is statically equivalent to \( \psi \). Indeed, for all \( a \in \{\xi \overset{?}{=} \zeta, \xi \overset{?}{\neq} \zeta, ax \in, ax \notin\} \), \( \delta(P, \phi) \overset{a}{\rightarrow} \delta_1 \) implies that \( \delta(Q, \psi) \overset{a}{\rightarrow} \delta_2 \) and \( \delta_1 \overset{\tau}{\rightarrow} \delta_2 \). Since neither the \( a \) transition or \( \tau \) transition modifies the frame, the former indicates that \( \delta(Q, \psi) \overset{a}{\rightarrow} E_1 \), \( E_1 \overset{a}{\rightarrow} E_2 \) and \( E_2 \overset{\tau}{\rightarrow} D \), with for all \( (Q', \psi') \in \text{supp}(E_1) \), \( \psi = \psi' \). The former ensures that \( \text{supp}(E_1) \neq \emptyset \). Therefore, for all \( ax, \xi, \zeta \), if \( \xi \overset{?}{=} \zeta, \xi \overset{?}{\neq} \zeta, ax \in \) or \( ax \notin \) holds on \( \phi \) then it also holds on \( \psi \) respectively. It implies that \( \phi \) and \( \psi \) are statically equivalent.

5.2 Observational preorder

Definition 18. The observational preorder \( \leq_{\text{obs}}^{R} \) is the largest relation \( R \) on \( MP \) such that \( P R Q \) implies:

- for all \( c \in N_{\text{pub}} \), \( \text{RProb}_{R}(P, \downarrow c) \leq \text{RProb}_{R}(Q, \downarrow c) \);
- if \( P \overset{\tau}{\rightarrow} D \) then \( Q \overset{\tau}{\rightarrow} E \) and \( D \overset{\tau}{\rightarrow} E \);
- for all closed \( \text{Adv} \in MP \) such that \( \text{fn}(\text{Adv}) \subseteq N_{\text{pub}}, \{\text{Adv}\} \cup P R \{\text{Adv}\} \cup Q \).

The observational equivalence \( \approx_{\text{obs}}^{R} \) is defined by additionally requesting \( R \) to be symmetric and in the second bullet point, by requesting both \( D \overset{\tau}{\rightarrow} E \) and \( E \overset{\tau}{\rightarrow} D \) to hold.

Remark 9. Note that we slightly diverge from the original definition of observational equivalence [ABF17] where an evaluation context \( C[\_] \) is of the form

\[
\text{new } n_1; \ldots; \text{new } n_k; (\_ | A)
\]

In our definition we simply consider a parallel process, and no additional name restriction. However, we prove in Appendix [D] that these two definitions coincide. Intuitively, restricting names whose scope includes the adversarial process \( A \) corresponds to making previously public channels invisible to the attacker at later steps, hence it does not provide additional distinguishing power.

In the following, we show the asymmetric counterpart to the result stated in [GLPT07] on the coincidence between bisimulation and observational equivalence. The proof can be found in Appendix [F].

Proposition 5. Let \( P, Q \) two processes in \( MP \).

\[
P \leq_{\text{obs}}^{R} Q \text{ iff } (P, \emptyset) \leq_{N}^{\text{sim}} (Q, \emptyset) \text{ and } P \approx_{\text{obs}}^{R} Q \text{ iff } (P, \emptyset) \approx_{N}^{\text{bi}} (Q, \emptyset)
\]
5.3 Bisimilarity and observational equivalence do not coincide with non-randomized schedulers

Here, we give more details on the different behaviors of (bi)simulations and observational preorder and equivalence, depending on whether we consider randomized or non-randomized schedulers. In [GLPT07], for instance, non-randomized schedulers are considered in the definition of bisimulation and observational equivalence on the probabilistic applied π-calculus.

As said earlier, for general NPLTSs, simulation is not transitive when considering non-randomized schedulers [Eis17]. We show here that even on our specific NPLTS $N^f$, simulation (resp. resolution) and observational equivalence (resp. preorder) do not coincide (Lemma 3). Furthermore, they do coincide for randomized schedulers (Proposition 5). This discrepancy would also arise when comparing observational preorder with a similar calculus (Lemma 3). However, they do coincide for randomized schedulers (Proposition 5). We show here that even on our specific NPLTS $N^f$, simulation is not transitive (Lemma 2). Furthermore, for non-randomized schedulers, bisimulation (resp. simulation) and observational equivalence (resp. preorder) do not coincide (Lemma 3).

5.3 Bisimilarity and observational equivalence do not coincide with non-randomized schedulers

Here, we give more details on the different behaviors of (bi)simulation and observational preorder and equivalence, depending on whether we consider randomized or non-randomized schedulers. In [GLPT07], for instance, non-randomized schedulers are considered in the definition of bisimulation and observational equivalence on the probabilistic applied π-calculus.

As said earlier, for general NPLTSs, simulation is not transitive when considering non-randomized schedulers [Eis17]. We show here that even on our specific NPLTS $N^f$, simulation (resp. resolution) and observational equivalence (resp. preorder) do not coincide (Lemma 3). However, they do coincide for randomized schedulers (Proposition 5). This discrepancy would also arise when comparing observational preorder with a similar definition where we would only require that $P \to D$ (rather $P \to r, D$) implies $Q \to r, E$ and $D \not\to_r E$. This one-step observational preorder is equivalent to observational preorder for randomized scheduler but not for non-randomized scheduler.

We are going to compare processes with respect to the bisimilarity $\approx_{bi}^{nr}$ as well as the observational equivalence with non-randomized schedulers. The definition of these variants with non-randomized resolutions follows Definition 18 but we use $\approx_{r}^{nr}$ instead of $\Rightarrow_{r,N^f}$. We denote this non-randomized bisimilarity as $\approx_{bi}^{nr}$ (and similar for $\leq_{sim}^{nr}$, $\approx_{obs}^{nr}$ and $\leq_{obs}^{nr}$).

**Lemma 2.** $\leq_{sim}^{nr}$ is not transitive.

**Proof.** Consider processes $P, Q$ and $R$ defined in Figure 8. The corresponding fragment of $N^f$ is displayed in Figure 8a. We will show that

$$((P), \varnothing) \leq_{sim}^{nr} ((R), \varnothing) \text{ and } ((Q), \varnothing) \leq_{sim}^{nr} ((P), \varnothing) \text{ but } ((P), \varnothing) \not\leq_{sim}^{nr} ((R), \varnothing).$$

We first show that $((P), \varnothing) \leq_{sim}^{nr} ((R), \varnothing)$. Notice that any $a$ transition can easily be matched, i.e. $((P), \varnothing) \Rightarrow ((P), \varnothing)$ and $((R), \varnothing) \Rightarrow ((R), \varnothing)$. Furthermore, we have $((P), \varnothing) \Rightarrow 0.5 \cdot \delta_{\{0\}} + 0.5 \cdot \delta_{\{1\}}$ which can be matched in $((R), \varnothing)$ using the scheduler displayed in Figure 8b, i.e. $(R, \varnothing) \Rightarrow R_{\varnothing} 0.5 \cdot \delta_{\{0\}} + 0.5 \cdot \delta_{\{1\}}$. This concludes the proof of $((P), \varnothing) \leq_{sim}^{nr} ((R), \varnothing)$.

It is easy to see that $(Q, \varnothing) \approx_{bi}^{nr} (Q, \varnothing)$ and so $(R, \varnothing) \leq_{sim}^{nr} (Q, \varnothing)$.

Finally, we need to prove that the transition $((P), \varnothing) \Rightarrow 0.5 \cdot \delta_{\{0\}} + 0.5 \cdot \delta_{\{1\}}$ cannot be simulated in $(Q, \varnothing)$. First one can see for all $d \in \{a, b\}$, for all $\mathcal{P} \in \{Q, P_{ab}(0.9), P_{ab}(0.1)\}$, $(out(d, c), \varnothing) \not\leq_{sim}^{nr} (\mathcal{P}, \varnothing)$. Indeed, $(out(d, c), \varnothing) \Rightarrow_{out(ax_1)} \delta_{\{0\}}$ with $\phi = \{ax_1 \mapsto c\}$ but we can have at best $\mathcal{P} \Rightarrow_{out(ax_1)} \delta_{\{0\}}$. Thus, the only relevant extended process from $(\mathcal{Q}, \varnothing)$ reachable from $(Q, \varnothing)$ is $(\mathcal{Q}, \varnothing))$ itself. However, we can only have $(Q, \varnothing) \Rightarrow_{out(ax_1)} \delta_{\{0\}}$. The proof is hence complete.
For the sake of readability, $\alpha$ stands for all labels $\text{ax}_i \notin \emptyset$, for $i \in \mathbb{N}$ and all labels $\xi \sim \xi'$ for all closed recipes $\xi, \xi'$ such that $\xi \sim \xi' \ (\sim \in \{=, \neq\})$. Furthermore, let

\begin{align*}
Q &= (\text{out}(a, c) +_{0.9} \text{out}(b, c)) + (\text{out}(a, c) +_{0.1} \text{out}(b, c)) \\
Q' &= \text{if } c = c \text{ then } Q \text{ else } 0 \\
P_{ab}(p) &= \{ \text{out}(a, c) +_{p} \text{out}(b, c) \} \\
R &= Q +_{0.5} Q' \\
P &= Q +_{0.5} Q' +_{p} \text{out}(b, c) \\
Q' &= \text{if } c = c \text{ then } Q \text{ else } 0 \\
0.5 \cdot \delta(\text{out}(a, c) +_{p} \text{out}(b, c)) \\
0.9 \cdot \delta(\text{out}(a, c)) +_{0.5} \delta(\text{out}(b, c)) \\
0.1 \cdot \delta(\text{out}(a, c)) +_{0.5} \delta(\text{out}(b, c)) +_{0.1} \delta(\text{out}(a, c) +_{0.9} \text{out}(b, c)) \\
0.9 \cdot \delta(\text{out}(a, c) +_{0.9} \text{out}(b, c))
\end{align*}

(a) The fragment of $N^\ell$ corresponding to $\langle \{ Q \}, \emptyset \rangle$ and $\langle \{ R \}, \emptyset \rangle$.

(b) The resolution for $\langle \{ R \}, \emptyset \rangle \Rightarrow_{R_m} 0.5 \cdot \delta(\text{out}(a, c)) + 0.5 \cdot \delta(\text{out}(b, c))$.

Figure 8: Fragments of $N^\ell$ showing that $\langle \{ P \}, \emptyset \rangle \leq_{n r \sim} \langle \{ R \}, \emptyset \rangle$, $\langle \{ R \}, \emptyset \rangle \leq_{n r \sim} \langle \{ Q \}, \emptyset \rangle$ but $\langle \{ P \}, \emptyset \rangle \not\leq_{n r \sim} \langle \{ Q \}, \emptyset \rangle$.

We now show that for non-randomized schedulers $(bi)$simularity and observational equivalence, respectively preorder, do not coincide. The processes witnessing this result are a direct counter-example to [GLPT07, Theorem 2].
For readability, given a closed frame $\phi$, $\alpha(\phi)$ stands for all labels $ax_i \notin \phi$, for $i > |\phi|$, and all labels $\xi \sim \xi'$ for all closed recipes $\xi, \xi'$ such that $\xi \phi \sim \xi' \phi$ ($\sim \in \{=, \neq\}$).

(a) The fragment of $N^\ell$ corresponding to $\{Q\}$ and $\{P\}$:

$P_{da} = \text{out}(d, c); (\text{out}(a, c) + 0.9) 0$

$P_{db} = \text{out}(d, c); (\text{out}(b, c) + 0.9) 0$

$\phi = \{ax_1 \rightarrow c\}$

$\phi' = \{ax_1 \rightarrow c, ax_2 \rightarrow c\}$

(b) The fragment of $N^o$ corresponding to $\{P; \text{in}(d, x).0\}$ and $\{Q; \text{in}(d, x).0\}$:

$m$ Figure 9: Fragments of NPLTS showing $\{Q\} \approx_{nr}^b \{P\}$ and $\{Q\} \leq_{obs}^o \{P\}$. 

}$
Lemma 3. There exist processes \( P, Q \in SP \) such that
\[
(Q, \emptyset) \approx_{bi} (P, \emptyset), \quad (Q, \emptyset) \approx_{bi}^n (P, \emptyset) \quad \text{and} \quad Q \not\leq_{obs}^n P
\]

Proof. We consider the following processes:
\[
P = \text{out}(d, c); (\text{out}(a, c) +_{0, 9} 0) + \text{out}(d, c); (\text{out}(b, c) +_{0, 9} 0)
\]
\[
P' = \text{if } c = c \text{ then } P \text{ else } 0
\]
\[
Q = P +_{1/2} P'
\]

We first show that \((Q, \emptyset) \approx_{bi}^n (P, \emptyset)\). For readability we ignore intermediate states of the form \(P \cup \{0\}\) as, for any \(P\) and \(\phi\), \((P \cup \{0\}), \phi) \xrightarrow{\tau} (P, \phi)\) and \((P \cup \{0\}), \phi) \approx_{bi}^n (P, \phi)\). Following the notations of Figure 9b, define the binary relation \(R\) as the reflexive, symmetric and transitive closure of
\[
\{( (Q, \emptyset), (P, \emptyset) ), \quad ( (P, \emptyset), (P', \emptyset) ) \}
\]

We show that \(R\) is a bisimulation. Let \(s_1, s_2 \in \{(Q, \emptyset), (P, \emptyset), (P', \emptyset)\}\). If \(s_1 \xrightarrow{\alpha} D\) then \(D = \delta_{s_1}\) and \(s_2 \xrightarrow{\alpha} R_{\text{nr}} \delta_{s_2}\). By definition of \(R\), \(\delta_{s_1} \xrightarrow{\tau} R \delta_{s_2}\). If \(s_1 \xrightarrow{\tau} D\) then we are in one of the following cases:

- \(s_1 = (Q, \emptyset)\) and \(D = 0.5 \cdot \delta_{(P, \emptyset)} + 0.5 \cdot \delta_{(P', \emptyset)}\), or \(s_1 = (P', \emptyset)\) and \(D = \delta_{(P, \emptyset)}\). In these cases, \(s_2 \xrightarrow{\tau} R_{\text{nr}} \delta_{s_2}\) and \(D \xrightarrow{\tau} R \delta_{s_2}\).

- \(s_1 = (P, \emptyset)\) and \(D = \delta_{(P_{\text{da}, \emptyset})}\) or \( \delta_{(P_{\text{db}, \emptyset})}\). As \(s_2 \xrightarrow{\tau} R_{\text{nr}} \delta_{s_1}\) we also have that \(s_2 \xrightarrow{\tau} R_{\text{nr}} D\) which allows us to conclude by reflexivity of \(R\).

All remaining cases are trivial by reflexivity of \(R\). The same relation can also be shown to be a witness that \((Q, \emptyset) \approx_{bi} (P, \emptyset)\).

Next, we need to show that \(Q \not\leq_{obs}^n P\). For this we show that
\[
\{Q; \text{in}(d, x).0\} \not\leq_{obs}^n \{P; \text{in}(d, x).0\}
\]
(since \(\leq_{obs}^n\) is closed under application of an attacker process). For readability we tacitly omit intermediate states of the form \(P \cup \{0\}\) as, for any \(P\) and \(\phi\), \((P \cup \{0\}), \phi) \xrightarrow{\tau} P\) and \((P \cup \{0\}), \phi) \approx_{obs}^n P\). Consider Figure 9b. We can build a non-randomized scheduler such that
\[
\{Q; \text{in}(d, x).0\} \xrightarrow{\tau} R_{\text{nr}}(N_{\phi}) D \quad \text{where} \quad D = 0.45 \cdot \delta_{P_a} + 0.45 \cdot \delta_{P_b} + 0.1 \cdot \delta_{\phi}
\]

However, there is no distribution \(E\) such that \(\{P; \text{in}(d, x).0\} \xrightarrow{\tau} R_{\text{nr}}(N_{\phi}) E\), and \(D \not\leq_{obs}^n E\). Indeed, the set \(S\) of all \(E\) such that \(\{P; \text{in}(d, x).0\} \xrightarrow{\tau} R_{\text{nr}}(N_{\phi}) E\) is
\[
S = \{s \mid s \in \{P; \text{in}(d, x).0\}, \{P_{\text{da}}; \text{in}(d, x)\}, \{P_{\text{db}}; \text{in}(d, x)\}, \{P_{a+}\}, \{P_{b+}\}\} \cup \{F_a, F_b\}
\]
where \(F_c = 0.9 \cdot \delta_{P_{c+}} + 0.1 \cdot \delta_{0}\) for \(c \in \{a, b\}\).
Observe—looking at Definition 18—that for any distribution $E$ such that $D \leq_{\text{ns}} E$, it must hold that:

$$\sum_{s \in \text{supp}(E)} E(s) \cdot \text{RProb}_{\nu}(s, \downarrow c) \geq 0.45 \quad \text{for } c \in \{a, b\}. \quad (2)$$

Indeed, we can see that $D \leq_{\text{ns}} E$ implies that:

$$0.45 = D(\{s \mid \text{RProb}_{\nu}(s, \downarrow c) = 1\}) \quad \text{for } c \in \{a, b\}$$

$$\leq E(\{s \mid \text{RProb}_{\nu}(s, \downarrow c) = 1\}) \quad \text{by definition of } \hat{R}$$

$$\leq E(\{s \mid \text{RProb}_{\nu}(s, \downarrow c) = 1\}) \quad \text{by definition of } \leq_{\text{ns}}$$

$$\leq \sum_{s \in \text{supp}(E)} E(s) \cdot \text{RProb}_{\nu}(s, \downarrow c)$$

We now use Equation (2) to show that there is no $E \in S$ such that $D \leq_{\text{ns}} E$.

- we see immediately that Equation (2) does not hold for $F_a, F_b$ as

$$\text{RProb}_{\nu}(\{P_{i+}, \downarrow j\}) = 0.9 \cdot \text{RProb}_{\nu}(\{P_{i}, \downarrow j\}) = 0 \quad \text{for } (i, j) \in \{(a, b), (b, a)\};$$

- by the definition of $\text{RProb}_{\nu}(\cdot)$, we see that

$$\text{RProb}_{\nu}(\{P_{i}, \downarrow j\}) = \sum_{s \in \text{supp}(F_c)} F_c(s) \cdot \text{RProb}_{\nu}(s, \downarrow j)$$

for $i \in \{a, b\}$ and every barb $j$. Hence, using the previous case, we conclude that Equation (2) does not hold for $\delta_{P_{a+}}, \delta_{P_{b+}}$.

- we have that $\text{RProb}_{\nu}(\{P_{i+}; \text{in}(d, x)\}, \downarrow j) = \text{RProb}_{\nu}(\{P_{i+}, \downarrow j\})$ for every barb $j$, and $i \in \{a, b\}$, and from there we can conclude using the previous case;

- finally, we have that

$$\text{RProb}_{\nu}(\{P; \text{in}(d, x), 0\}, \downarrow j) = \max_{i \in \{a, b\}} \text{RProb}_{\nu}(\{P_{i+}; \text{in}(d, x)\}, \downarrow j)$$

for every barb $j$, and we conclude using the previous case.

Therefore, $\|Q\| \not\leq_{\text{obs}} \|P\|$.

### 5.3.1 Counter-example to coincidence of bisimulation and observational congruence in [GLPT07]

Since we defined our operational semantics in a slightly different way as [GLPT07], we present below a proof that Lemma 3 also holds in their framework. Their semantics differs from ours in two distinct ways: the first one is simply a presentation issue, since they define
probabilistic executions and schedulers in a different but equivalent way; while the other one consists of minor difference in the semantics of the reduction. Here, we rewrite their definitions in our framework, while keeping their operational semantics, i.e., their one-step reduction relation. This paragraph is not entirely self-contained, but all definitions and notations used here can be found in [GLPT07].

**Notation 9.** We note \( P_\diamond \) for the set of extended processes (in the sense of [GLPT07]), up to static equivalence (again, in the sense of [GLPT07]). We note \( \alpha \rightarrow_\diamond \subseteq P_\diamond \times D(P_\diamond) \) for the one-step reduction relation there. For any name \( a \), we write \( \downarrow_\diamond a \) for the set of all extended processes (in the sense of [GLPT07]) of the form \( C[\overline{a}(x).P] \), where \( C \) is an evaluation context (in the sense of [GLPT07]) that does not bind \( a \). Whenever, processes are in \( P_\diamond \), we also use the syntax of [GLPT07] for outputs and inputs, i.e., \( \overline{a}(x) \) and \( a(x) \) instead of \( \text{out}(a, x) \) and \( \text{in}(a, x) \).

From there, we define two NPLTS \( N_\circ \) and \( N_\ell \) that translate their operational semantics in our NPLTS based framework.

**Definition 19.** The set of states of both \( N_\circ \) and \( N_\ell \) is \( P_\diamond \). \( N_\circ \)’s set of actions is \( \{ \tau \} \), while \( N_\ell \)’s set of actions is:

\[
\{ \tau \} \cup \{ a \mid a \text{ labelled actions in [GLPT07]} \} \\
\cup \{ \? \in E \mid E \text{ equivalence class for static equivalence } \}.
\]

Observe that there is a direct correspondence between schedulers in the sense of [GLPT07], and non-randomized schedulers on \( N_\ell \). We explicit this correspondence in Lemma 4 below.

**Lemma 4.** Let \( P \in P_\circ \), \( A \) a subset of \( P_\circ \), and \( p \in [0, 1] \).

- there exists a scheduler \( F \) (in the sense of [GLPT07]) such that \( \text{Prob}_P^F(A) = p \) if and only if there exists \( R \in R_{nr}(N_\ell) \), and \( s' \) with \( \text{corr}_R(s') = P \) such that
  \[
  \text{RProb}_R(s', \text{corr}_{R}^{-1}(A)) = p;
  \]

- there exists a scheduler \( F \) (in the sense of [GLPT07]) such that \( \text{Prob}_P^F(\alpha, A) = p \) if and only if there exists \( D \) such that
  \[
  P \xrightarrow{\alpha_{R_{nr}(N_\ell)}} D \quad \text{and} \quad D(A) = p;
  \]

- there exists a scheduler \( F \) (in the sense of [GLPT07]) such that \( \text{Prob}_P^F(p_\star \alpha p_\star, A) = p \) if and only if there exists \( R \in R_{nr}(N_\ell) \), and \( s \) with \( \text{corr}_R(s) = P \) such that
  \[
  \sum_{u \in S_{\text{ext}}(R)} \sum_{\text{trans}_R(u)(a) \neq \star} \text{RProb}_R(s, \{ u \}) \cdot \text{trans}_R(u)(a)(v) \cdot \text{RProb}_R(v, \text{corr}_{R}^{-1}(A)) = p.
  \]
Using the correspondance in Lemma 4, the counter-example proof below can be followed using either original definitions in [GLPT07], or the NPLTS based view of the present paper.

**Notation 10.** Let \( R \) be an equivalence relation on \( \mathcal{P}_o \), and \( P \in \mathcal{P}_o \).

- For any scheduler \( F \), we denote by \( w_{F,P} \) the function that to any equivalence class \( E \) of \( R \) associates \( \text{Prob}^{\text{F}}_E (E) \) and for any \( T \in \mathcal{P}_o \), we note \( \mathcal{H}_T \) for the set of the different possible \( w_{F,T} \) for any scheduler \( F \).

- For any action \( \alpha \), scheduler \( F \), we denote by \( w_{\alpha,F,P} \) the function that to an any equivalence class \( E \) of \( R \) associates \( \text{Prob}^{\text{F}}_E (\tau^\ast \alpha \tau^\ast, E) \). For any \( T \in \mathcal{P}_o \), we note \( \mathcal{H}_T^\alpha \) for the set of the different possible \( w_{\alpha,R,s} \) for \( \text{corr}_R (s) = T \).

Using Notation 10 we can rewrite the definitions of both observational congruence and bisimulation in [GLPT07]:

**Lemma 5.** An observational congruence on \( \mathcal{P}_o \) in the sense of [GLPT07] is a symmetric relation \( R \) on \( \mathcal{P}_o \) such that \( P \ R Q \) implies:

1. \( \text{RProb}_{R_{sb}}(P, \downarrow^\circ a) = \text{RProb}_{R_{sb}}(Q, \downarrow^\circ a) \);
2. \( \mathcal{H}_P = \mathcal{H}_Q \);
3. for all closing evaluation context \( C, C[P] R C[Q] \).

We denote by \( \approx_{o,\text{obs}} \) the largest observational congruence on \( \mathcal{P}_o \).

**Lemma 6.** A bisimulation on \( \mathcal{P}_o \) in the sense of [GLPT07] is a symmetric relation \( R \) on \( \mathcal{P}_o \) such that \( P \ R Q \) implies:

1. \( \mathcal{H}_P = \mathcal{H}_Q \);
2. if \( P \overset{\alpha}{\rightarrow}_{R_{sb}(N_o)} D \) with \( \alpha \neq \tau \), then there exists \( h \in \mathcal{H}_Q^{\circ} \) such that for every equivalence class \( E \) of \( R \), \( h( E ) = D( E ) \)

We denote by \( \approx_{b,\text{bi}} \) the largest bisimulation on \( \mathcal{P}_o \).

**Lemma 7.** There exist extended processes \( Q_1, Q_2 \in \mathcal{P}_o \) such that \( Q_1 \approx_{b,\text{bi}} Q_2 \) and \( Q_1 \not\approx_{o,\text{obs}} Q_2 \).

**Proof.** We slightly change the processes used in the proof of Lemma 3, to account for the change in the one-step reduction relation. We define \( Q_1, Q_2 \in \mathcal{P}_o \) as:

\[
\begin{align*}
P & = \overline{\pi}(c).\overline{(\overline{d}(c).\overline{F}_1(c) + \overline{d}(c).\overline{F}_2(c))} \\
P' & = \text{if } c = c \text{ then } P \text{ else } 0 \\
Q_1 & = \overline{\pi}(c).P \oplus 1/2 \overline{\pi}(c).P' \\
Q_2 & = \overline{\pi}(c).P
\end{align*}
\]
We first show that \( Q_1 \approx_{\delta_i} Q_2 \), using the relevant fragment of \( N_5 \) represented in Figure 10a. We are going to show that the relation \( R \) defined as the reflexive, transitive and symmetric closure of \( \{(Q_1, Q_2), (Q_2, \overline{a}(c).P'), (P, P')\} \) is a bisimulation on \( \mathcal{P}_5 \) in the sense of Lemma 9. What are the equivalence classes of \( R \)? They consist of

\[
C_0 := \{Q_1, Q_2, \overline{a}(c).P'\} \quad C_1 := \{P, P'\} \quad C_T = \{T\} \text{ for all other extended processes } T
\]

We now show that \( R \) is a bisimulation by considering all (non-identical) pairs in \( R \), and we show that the conditions of Lemma 6 hold. Let be \((U_1, U_2) \in R\), with \( U_1 \neq U_2 \) (if \( U_1 = U_2 \), it is obvious that both conditions in Lemma 6 hold).

- We first check Condition 1 in Lemma 6. We can see using Figure 10a that for every \( C \in H \) ordered pair).

- We now check Condition 2 in Lemma 6. First, if \( \alpha \) that \( U \) is statically equivalent, since their frames are empty. (In the semantics of [GLPT07] we see immediately that the condition holds: indeed all the relevant processes are channel names can be output without extending the frame.) Now, let us suppose \( \alpha \) is not an action for static equivalence. As a first step, we compute the function \( h \) and suppose it to be 0 on any class not in \( E_1, \ldots, E_n \).

\[
\mathcal{H}_{Q_1}^\alpha = \{(C_1 \mapsto 1), (C_1 \mapsto \frac{1}{2})\} \text{ when } \alpha = \overline{a}(c), \{0_H\} \text{ otherwise.}
\]

\[
\mathcal{H}_{Q_2}^\alpha = \{(C_1 \mapsto 1), 0_H\} \text{ when } \alpha = \overline{a}(c), \{0_H\} \text{ otherwise.}
\]

\[
\mathcal{H}_{\overline{a}(c).P'}^\alpha = \{(C_1 \mapsto 1), 0_H\} \text{ when } \alpha = \overline{a}(c), \{0_H\} \text{ otherwise.}
\]

\[
\mathcal{H}_{P'} = \{(\{S\} \mapsto 1), 0_H\} \cup \{(\{S\} \mapsto 1, \{P_{di}\} \mapsto 1) \mid i \in \{a, b\}\} \text{ when } \alpha = \overline{a}(c), \{0_H\} \text{ otherwise.}
\]

\[
\mathcal{H}_{P'}^\alpha = \{(\{S\} \mapsto 1), 0_H\} \cup \{(\{S\} \mapsto 1, \{P_{di}\} \mapsto 1) \mid i \in \{a, b\}\} \text{ when } \alpha = \overline{a}(c), \{0_H\} \text{ otherwise.}
\]

We perform a case analysis depending on the first element in \((U_1, U_2)\) (seen as an ordered pair).

- \( U_1 = Q_1 \): Observe that there doesn’t exist \( \alpha \neq \tau \) such that \( Q_1 \xrightarrow{\alpha} R_{\text{re}}(N_5) D \), thus we conclude.

- \( U_1 = Q_2 \): Suppose that \( Q_2 \xrightarrow{\alpha} R_{\text{re}}(N_5) D \) with \( \alpha \neq \tau \) : it means that \( \alpha = \overline{a}(c) \), and \( D = \delta_{P'} \). We need to show that \((C_1 \mapsto 1) \in \mathcal{H}_{\overline{a}(c).P'}^\alpha \) for all \( U_2 \) such that \((U_1, U_2) \in R \). Then we conclude by seeing that indeed \( h : (C_1 \mapsto 1) \in \mathcal{H}_{Q_1}, \mathcal{H}_{\overline{a}(c).P'}^\alpha \) (for \( Q_1 \), for instance, it is simply the scheduling that does

\[
Q_1 \xrightarrow{\tau} R_{\text{re}}(N_5) \frac{1}{2}\delta_{\overline{a}(c).P} + \frac{1}{2}\delta_{\overline{a}(c).P'} \xrightarrow{\alpha} R_{\text{re}}(N_5) \frac{1}{2}\delta_{P'} + \frac{1}{2}\delta_{P'}.
\]

33
Lemma 8. There exist extended processes $P \in \mathcal{P}$ by application of evaluation contexts, thus it is sufficient to show that $C$.

Now, we are going to show that $Q$.

Observe that $Q$.

We consider the following pair of (non-probabilistic) programs:

Proof. We consider the following pair of (non-probabilistic) programs:

$$Q_1 := \begin{cases} 
\text{if } (c = c) \text{ then } (a(x)(\bar{\pi}(0) + \bar{\pi}(1))) \text{ else } 0; \\
Q_1 := Q_1 + 0
\end{cases}$$

Observe that $Q_1 \approx_{bi}^\diamondsuit Q_2$. Now, we are going to show that $Q_1 \not\approx_{bi}^\diamondsuit Q_2$. Let $R$ be a bisimulation on $\mathcal{P}_o$. As previously, for any $P \in \mathcal{P}_o$, we note $cl(P)$ for the equivalence class of $P$ with respect to $R$. We do the proof by contradiction, so we suppose that $cl(Q_1) = cl(Q_2)$. Using the fact that $R$ is a bisimulation, we can show that the following $R$-equivalence classes are distinct:

- $C_0$: the class containing the process 0;
- $C_{Q_1,Q_2}$: the class containing the processes $Q_1$ and $Q_2$;
- $D_0$: the class containing the process $a(x) | \bar{\pi}(0);$
• $D_1$: the class containing the process $a(x) \mid \overline{a}(1)$.

We can see that there exists a (non-randomized) scheduler $F$ such that $w_{F,Q_2}(C_0) = 1$, $w_{F,Q_2}(C_{Q_1,Q_0}) = 1$, and $w_{F,Q_2}(D_i) = 0$ for $i \in \{0,1\}$. ($F$ is the scheduler that, starting from $Q_1 + 0$, does the non-deterministic choice towards 0). But, by looking at all possible scheduling from $Q_1$, we can see that there doesn’t exists $w_{F',Q_1}$ such that $w_{F',Q_1} = w_{F,Q_2}$: as a consequence $H_{Q_1} \neq H_{Q_2}$, thus we have a contradiction.

5.3.2 The counter-example for a conservative variant of [GLPT07] bisimulation

We want to highlight here that the issue of non-conservativity is orthogonal to the issue of coincidence between bisimulation and observational equivalence: our claim is that the later is really based on the choice between randomized or non-randomized schedulers. In this section, we present a variant of the bisimulation defined in [GLPT07] on $P$, which is conservative but where schedulers are non-randomized, and we show that we still do not have the coincidence between bisimulation and observational equivalence.

Definition 20. An observational congruence on $P$ is a symmetric relation $R$ on $P$ such that $P \trianglerighteq R \trianglerighteq Q$ implies:

- $\text{RProb}_{\text{nr}}(N_{\ell} \trianglerighteq D) = \text{RProb}_{\text{nr}}(N_{\ell} \trianglerighteq E)$;
- if $P \xrightarrow{\tau} D$ then there exists $E$ such that $Q \xrightarrow{\tau} E$ and $D \xrightarrow{R} E$;

We note $\approx_{\text{obs}}$ for the greatest observational congruence on $P$.

Definition 21. A bisimulation on $P$ is a symmetric relation $R$ on $P$ such that $P \trianglerighteq R \trianglerighteq Q$ implies:

- if $P \xrightarrow{\tau} D$ then there exists $E$ such that $Q \xrightarrow{\tau} E$ and $D \xrightarrow{R} E$;
- if $P \xrightarrow{a} D$ with $a \neq \tau$, then there exists $E$ such that $Q \xrightarrow{a} E$ and $D \xrightarrow{R} E$;

We note $\approx_{\text{bi}}$ for the greatest bisimulation on $P$. 

35
Observe that bisimilarity in the sense of Definition 21 above does not coincide with bisimilarity— in the sense of Definition 17 on the labelled NPLTS \(N_\delta^c\). The important difference is that, as for observational congruence, in [GLPT07] only non-randomized schedulers are considered. Another technical difference is the use of \(\Rightarrow\) instead of \(\rightarrow\) on the left side in the first clause.

We are now going to show the counterpart of Lemma 3 for the observational congruence and weak bisimilarity in the sense of [GLPT07]. It actually disproves [GLPT07, Theorem 2].

**Lemma 9.** There exist extended processes \(Q_1, Q_2 \in \mathcal{P}_o\) such that \(Q_1 \approx_{bi} Q_2\) and \(Q_1 \not\approx_{o} Q_2\).  

**Proof.** We need to change a little the processes used in the proof of Lemma 3 due to the change in the one-step reduction relation. We define \(Q_1, Q_2 \in \mathcal{P}_o\) as:

\[
P = \overline{a}(c). (\overline{a}(c). \overline{b}(c) + \overline{b}(c). \overline{d}(c))
\]

\[
P' = \text{if } c = c \text{ then } P \text{ else } 0
\]

\[
Q_1 = \overline{a}(c). P \oplus_{1/2} \overline{a}(c). P'
\]

\[
Q_2 = \overline{a}(c). P
\]

We first show that \(Q_1 \approx_{bi} Q_2\), using the relevant fragment of \(N_\delta^c\) represented in Figure 10a.  

We are going to show that the relation \(R\) defined as the reflexive and symmetric closure of \(\{(Q_1, Q_2), (Q_2, \overline{a}(c). P'), (P, P')\}\) is a bisimulation on \(\mathcal{P}_o\) in the sense of Definition 21. We do that by considering all (non-identical) pairs in \(R\), and we show that the conditions of Definition 21 hold.

- \((Q_1, Q_2)\): suppose that \(Q_1 \xrightarrow{\tau} R_{w}(N_\delta^c)\) \(D\), then \(D\) is in \(\{\delta_{Q_1}, (\frac{1}{2}\delta_{Q_2} + \frac{1}{2}\delta_{\overline{a}(c). P'})\}\). From there, we can conclude by observing that \(Q_2 \xrightarrow{\tau} R_{w}(N_\delta^c)\) \(\delta_{Q_2}\), and both \(\delta_{Q_1} \xrightarrow{\tau} \delta_{Q_2}\) and \((\frac{1}{2}\delta_{Q_2} + \frac{1}{2}\delta_{\overline{a}(c). P'})\) \(\delta_{Q_2}\) Now, suppose that \(Q_1 \xrightarrow{\alpha} R_{w}(N_\delta^c)\) \(D\) with \(\alpha \neq \tau\). First, if \(\alpha\) is an action for static equivalence, we see immediately that the condition holds: indeed all the relevant processes are statically equivalent, since their frames are empty. (For this reason, we will not consider explicitly again actions for static equivalence in the following). Looking at \(N_\delta^c\), we see that there is no other non-silent action that \(Q_1\) can do.

- \((Q_2, Q_1)\): suppose that \(Q_2 \xrightarrow{\tau} R_{w}(N_\delta^c)\) \(D\), then \(D = \delta_{Q_2}\), and we can immediately conclude. Suppose now that \(Q_2 \xrightarrow{\alpha} R_{w}(N_\delta^c)\) \(D\) with \(\alpha \neq \tau\): it means that \(\alpha = \overline{a}(c)\), and \(D = \delta_{P}\). Then we can conclude by seeing that \(Q_1 \xrightarrow{\alpha} R_{w}(N_\delta^c)\) \(\delta_{P}\).

- \((Q_2, \overline{a}(c). P')\): suppose that \(Q_2 \xrightarrow{\tau} R_{w}(N_\delta^c)\) \(D\), then as in the previous case \(D = \delta_{Q_2}\), and we can immediately conclude. Suppose now that \(Q_2 \xrightarrow{\alpha} R_{w}(N_\delta^c)\) \(D\) with \(\alpha \neq \tau\): then \(\alpha = \overline{a}(c)\), and \(D = \delta_{P}\). Then we can conclude by seeing that \(\overline{a}(c). P' \xrightarrow{\alpha} R_{w}(N_\delta^c)\) \(\delta_{P}\).
• \((\overline{a}(c).P', Q_2)\). The only \(\tau\) action that starts from \(\overline{a}(c).P'\) is the loop \(\tau\) action, thus we can immediately conclude for the \(\tau\) clause. Whenever \(\overline{a}(c).P' \xrightarrow{\alpha}_{\mathcal{R}_E(N)} D\), it means that \(\alpha = \overline{a}(c)\), and \(D = \delta_{P'}\). From there, we can see that the condition on \(Q_2\) holds by observing that \(Q_2 \xrightarrow{\alpha}_{\mathcal{R}_E(N)} \delta_P\), and \(\delta_{P'} \xrightarrow{\alpha} \delta_P\).

• \((P, P')\): the only \(\tau\) action that starts from \(P\) is the loop action, thus we can pass immediately to the case of non-silent action. Whenever \(P \xrightarrow{\alpha}_{\mathcal{R}_E(N)} D\), it means that \(\alpha = \overline{a}(c)\), and \(D = \delta_S\). Now, observe that \(P' \xrightarrow{\alpha}_{\mathcal{R}_E(N)} \delta_S\), and since \(R\) is reflexive (by definition), \(\delta_S \xrightarrow{\alpha} \delta_S\).

• \((P', P)\): suppose that \(P' \xrightarrow{\tau}_{\mathcal{R}_E(N)} D\), then \(D \in \{\delta_{P'}, \delta_P\}\). Since by the loop action \(P \xrightarrow{\tau}_{\mathcal{R}_E(N)} \delta_P\), and moreover \(P' R P\), we can conclude. We look now at non-silent actions: there are actually none to consider, since there is no non silent actions that \(P'\) can do in one step.

Now, we show that \(Q_1 \not\approx_{\text{obs}} Q_2\). First, observe that by definition \(\approx_{\text{obs}}\) is closed by application of evaluation context, thus it is enough to show that \(C[Q_1] \not\approx_{\text{obs}} C[Q_2]\) for some evaluation context \(C\). We take \(C = [] | a(x).a(x).d(x)\). Looking at the NPLTS \(N_E\), we see that \(C[Q_1] \xrightarrow{\tau}_{\mathcal{R}_E(N)} D := 1/2\delta_{b_1(c)} + 1/2\delta_{b_2(c)}\). Now, we claim that it is not possible to find \(E\) such that \(C[Q_2] \xrightarrow{\tau}_{\mathcal{R}_E(N)} E\), and \(D \approx_{\text{obs}} E\). The argument is as follows: first, observe that all \(E\) such that \(C[Q_2] \xrightarrow{\tau}_{\mathcal{R}_E(N)} E\) are dirac distributions: it is because the schedulers are not randomized, and there is no probabilistic choice in the process \(C[Q_2]\) (this statement can also be checked directly by looking at all cases in the operational semantics). It means that \(D = \delta_A\), for some \(A \in \mathcal{P}_E\). But for having \((D, \delta_P) \in \approx_{\text{obs}}\), \(A\) must be connected by \(\approx_{\text{obs}}\) to all elements in the support of \(D\), i.e. both to \(b_1(c)\) and \(\delta_{b_2(c)}\). Since in particular elements connected by \(\approx_{\text{obs}}\) must have the same (supremum) success probability for each barb, we obtain a contradiction.

\(\square\)

**Remark 10.** The processes \(Q_1, Q_2\) used in the proof of Lemma 9 are not the simplest possible counter-example. However, they have the advantage to be robust to small changes in the definition of bisimulation, in particular in the case where the double arrow \(\xrightarrow{\alpha}_{\mathcal{R}_E(N)}\) is used both left and right in all clauses of the bisimulation definition.
For readability, we omit the actions (? ∈ E) for static equivalence. Observe that since none of these extended processes use active substitutions, in particular they are all in the same equivalence class for static equivalence. We used the following notations: $P_{da} = d\langle c \rangle . b_1 \langle c \rangle$ and $P_{db} = d\langle c \rangle . b_2 \langle c \rangle$.

(a) The fragment of $N_\ell^\delta$ corresponding to $Q_1$ and $Q_2$.

(b) A non-randomized resolution on $N_o^\delta$ witnessing $C[Q_1] \Rightarrow_{R_m(N_\ell^\delta)} \frac{1}{2} \delta_{b_1(c)} + \frac{1}{2} \delta_{b_2(c)}$.

Figure 10: Fragments of NPLTS showing $Q \approx_{bi} P$, and $Q \not\leq_{obs} P$. 

38
Part II  
Well behaved subclasses of protocols

It is a well-known phenomenon that non-determinism and probabilistic choices do not interact well: a particular scheduler may for instance leak a secret probabilistic choice. Such schedulers are generally deemed unrealistic, and several papers aim at restricting schedulers [CP10, AAPvR10]. We illustrate this phenomenon on the following example.

Example 3. Consider the following two processes.

\[ P := (\text{in}(c, x). \text{if } x = 0 \text{ then out}(ok, 1) \text{ else out}(bad, 1)) + \frac{1}{2} (\text{in}(c, x). \text{if } x = 0 \text{ then out}(bad, 1) \text{ else out}(ok, 1)) \]

\[ Q := \text{in}(c, x). (\text{out}(ok, 1) + \frac{1}{2} \text{out}(bad, 1)) \]

One may, intuitively, consider that these two processes exhibit the same behaviour. Q takes an input and then with probability 1/2 decides to either output on ok or on bad. P on the other hand first choses a branch with probability 1/2. Each branch performs an input and according to the input value outputs either on ok or on bad. However, as the two branches make opposite choices on the output depending on the input value, one might expect the probability to output on ok to be 1/2.

However, we now show that P and Q are not may testing equivalent and can be distinguished by the following adversary:

\[ Adv = \text{out}(c, 0) \mid \text{out}(c, 1) \]

Indeed, we can show that:

\[ \exists \text{ resolution } R \text{ s.t. } R\text{Prob}_R(P \mid Adv, \downarrow \text{ok}) = 1 \]

\[ \text{and } \forall \text{ resolution } R', R\text{Prob}_{R'}(P \mid Adv, \downarrow \text{ok}) \leq \frac{1}{2} \]

Intuitively, this results from the fact that the resolution may leak the probabilistic choice through the non-deterministic choice of the attacker to output 0 or 1. The resolution chooses the attacker to output 0 in the first probabilistic branch of P and 1 in the second.

In this part we identify two subclasses of processes that avoid this problem. The first such subclass is that of purely non-deterministic processes, i.e. without the \(+_p\) operator. This is the class of the original applied pi calculus which also enjoys good tool support. Intuitively, the above problem is avoided as there are no secret probabilistic choices to leak. However, we show that even on purely non-deterministic processes probabilistic adversaries have a stronger distinguishing power for the may testing equivalence. We also show that when additionally restricting protocols to a bounded number of sessions, i.e.,
considering processes without replication, may-testing coincides with similarity. We therefore inherit from \cite{ckris18} the fact that may-testing is \text{coNEXP} complete for a large class of cryptographic primitives.

The second subclass considers purely probabilistic processes with (nearly) no non-determinism. We show that trace equivalence in this class (as considered for instance in \cite{csv17}) corresponds to may-testing with a restricted adversary process which again avoids non-determinism. We also sketch how the algorithms of then DeepSec prover could be adapted to check trace equivalence in this probabilistic setting.

6 Non-Probabilistic Processes

In this section, we consider protocols that do not make probabilistic choices, but an \textit{adversary} that can have a probabilistic behaviour. By Proposition \ref{prop:may-testing}, we already know that simulation preorder implies may-testing equivalence. We show in this section that for bounded non-probabilistic processes, may testing and simulation actually coincide. We also show that this is not the case in general for processes with an unbounded number of sessions, i.e., with replication.

6.1 May-testing with non-probabilistic adversary and trace equivalence coincide

We show in this section that our definitions of may testing and trace equivalence coincide with the classical definitions of the original, purely non-deterministic applied pi calculus when both regular and adversary processes are non probabilistic (up to the difference on static equivalence, discussed in Remark \ref{rem:static}).

Indeed, when considering non-probabilistic processes, all distributions in the (labeled) operational semantics are Dirac distributions. Moreover, we already shown in Proposition \ref{prop:may-testing-prob} that trace equivalence with randomized or non-randomized resolutions are equivalent. Thus, when considering non-randomized resolutions, one can notice that the probability of executing a trace \(w\) from \((P, \phi)\) will always be 0 or 1, and in fact is 1 if and only if \((P, \phi) \rightarrow_{a_1} \cdots \rightarrow_{a_n} (P', \phi')\) where \(w = a_1 \cdots a_n\) when having removed \(\tau\) actions and writing \((P, \phi) \rightarrow_a (P', \phi')\) instead of \((P, \phi) \rightarrow_a \delta_{(P', \phi')}\). Similarly, the probability of reaching a barb \(c\) from \(P\) is always 0 or 1, and in fact is 1 if and only if \(P \rightarrow^* P' \in \downarrow c\) when writing \(P \rightarrow Q\) instead of \(P \rightarrow_{\tau} Q\).

We show in the following lemma that may testing and trace equivalence coincide in non-probabilistic settings. In particular, we recover the fact that for the classical definitions in non-probabilistic settings, trace equivalence implies may-testing.

\textbf{Proposition 6.} Let \(P, Q \in \mathcal{MP}^{np}\).

\[
(P, \emptyset) \leq_{tr} (Q, \emptyset) \quad \text{iff} \quad \forall Adv \in \mathcal{MP}^{np} \text{ s.t. fn}(Adv) \subseteq N_{pub}, \forall c \in N_{pub}, \text{ RProb}_R(P \cup Adv, \downarrow c) \leq \text{RProb}_R(Q \cup Adv, \downarrow c)
\]

The proof of this result can be found in Appendix \ref{app:may-testing}.
6.2 May-testing and simulation coincide for bounded processes

To show that may-testing (with probabilistic adversaries) and simulation coincide for bounded non-probabilistic processes, i.e. in $\mathcal{MP}^{<\infty, np}$, we rely on a modal characterization of strong simulation on image finite LTS by a Hennessy-Milner logic. We can rely on simple LTS since we consider non-probabilistic processes (which implies that all distributions on the fragment of $N^f$ with states labeled by non-probabilistic processes are Dirac distributions).

**Definition 22.** A LTS $L = (S, A, \rightarrow)$ is *image-finite* when for every $s \in S, a \in A$, the set $\{ t \mid s \xrightarrow{a} t \}$ is finite.

Note that even the fragment of $N^f$ restricted to states in $\mathcal{SP}^{<\infty, np}$ is technically not image finite. Indeed, the $\tau$-action executing a restriction, i.e. new $a; P$, may lead to infinitely many states (one for each fresh private name $a'$ replacing $a$). However, as all these states (and subsequent states) are equal up to renaming, we can consider a fragment of $N^f$ where we only allow a single transition for the rule (NEW) per occurrence of name restriction in a state.

**Definition 23.** We say that a NPLTS $N$ is a *new-determinization* of $N^f$ when $N = (S_N, A_N, \text{trans})$ with the following constraints on trans:

- either $\text{trans}((P, \phi))(a) = \emptyset$ for $a \in A_N \setminus \{\tau\}$, and $\text{trans}((P, \phi))((\tau))$ is the singleton $\{\delta(Q,\phi)\}$ where $P \xrightarrow{\tau} Q$ by the rule (NEW)
- or $\text{trans}((P, \phi)) = \text{trans}_N((P, \phi))$ and for all $D \in \text{trans}((P, \phi))((\tau)), (P, \phi) \xrightarrow{\tau} D$ by any other rule than (NEW).

Though we use new-determinization primarily in this section for bounded processes to obtain the image-finite property, we can show that new-determinization preserves trace equivalence, simulation and bisimulation in general. The proof of the following lemma can be found in Appendix E.

**Lemma 10.** Let $N$ be a new-determinization of $N^f$.

$$\leq_{tr}^{N} = \leq_{tr}^{N^f} \text{ and } \leq_{sim}^{N} = \leq_{sim}^{N^f} \text{ and } \approx_{bi}^{N} = \approx_{bi}^{N^f}$$

It is a well-known fact ([Ama16]) that simulation for a LTS can be expressed as strong simulation on the corresponding weak LTS. Thus, when we only consider non-probabilistic processes, if we write $(P, \phi) \xrightarrow{a} (P, \phi')$ instead of $(P, \phi) \xrightarrow{a} \delta(P,\phi')$ then we can build from a new-determinization of $N^f$ a LTS where the transitions $(P, \phi) \xrightarrow{\tau} a \xrightarrow{\tau} \tau \rightarrow^* (Q, \psi)$ are merged into a single transition.

**Definition 24.** Given a new-determinization $N$ of $N^f$, we define the LTS $L_N$ as follows: its states and actions set are the same as those of $N$, and its transition function is defined as: for all $a \in A_N, (P, \phi) \xrightarrow{a}_{L_N} (Q, \psi)$ when $(P, \phi) \xrightarrow{\tau} \tau \xrightarrow{a} \tau \rightarrow^* (Q, \psi)$ in $N$ and $P$ is non-probabilistic.
Notice that the fragment of $L_N$ restricted to states in $\mathcal{SP}_\ell^{<\infty, np}$ is image finite.

If we denote the strong simulation on the LTS $L$ by $\leq_{ssim}^L$, we obtain the following result.

**Lemma 11.** Let $N$ be a new-determinization of $N^\ell$. For all $P, Q \in \mathcal{MP}^{np}$, we have:

$$(P, \emptyset) \leq_{ssim}^N (Q, \emptyset) \iff (P, \emptyset) \leq_{ssim}^{LN} (Q, \emptyset).$$

**Proof.** The proof directly follows from the following two observations: for all $a \in A_{N^\ell},$

- $(P, \phi) \xrightarrow{a} D$ implies that for all $(P', \phi') \in \text{supp}(D)$, $(P, \phi) \xrightarrow{\tau} \xrightarrow{a} (P', \phi')$, and
- $(P, \phi) \xrightarrow{\tau} \xrightarrow{\tau} (P', \phi')$ implies $(P, \phi) \xrightarrow{a}, \delta(P', \phi').$ \(\square\)

For the remaining of this section, we fix a new-determinization of $N^\ell$ and denote by $L^\ell$ its corresponding LTS. Moreover, we denote by $L^\ell_{<\infty}$ the fragment of $L^\ell$ restricted to bounded processes.

### 6.2.1 Hennessy-Milner logical characterization of strong simulation

Our Hennessy-Milner logical characterization consists in expressing strong simulation pre-order by the means of satisfaction of *logical formulas* by the LTS.

**Definition 25.** Let $A$ be a countable set of actions. We define the set of *logical formulas* as:

$$F \in \mathcal{F} := \top | a.F | F_1 \land F_2,$$

where $a \in A$.

In our case, the set of a actions will correspond to $A_{N^\ell}$ that is indeed countable. The satisfaction of such formulas by a LTS is defined as follows.

**Definition 26.** Let $L = (S, A, \rightarrow)$ be a LTS. We say that $L$ satisfies a formula $F$, written $s \models F$, if for all $s \in S$,

- $s \models \top$;
- $s \models a.F$ when there exists $t$ such that $s \xrightarrow{a} t$ and $t \models F$;
- $s \models F_1 \land F_2$ when $s \models F_1$ and $s \models F_2$.

The following proposition shows how to relate strong simulation with satisfyability of logical formulas.

**Proposition 7** (HML caracterisation of simulation). For an image-finite LTS $L$,

$$s \leq_{ssim}^L t \iff \forall F \in \mathcal{F}. s \models F \text{ implies } t \models F.$$
Proof. In [Ama16], this characterisation can be found with infinite disjunction. Using the fact that the bisimulation operator on relation is co-continuous for image-finite LTS, (as stated and shown in [Ama16]), we can show that the restriction to finite disjunction is sufficient. A generic proof of the finite-disjunction restriction for any modal characterisation can be found in [GFT12].

In order to prove that simulation coincides with may-testing for bounded non-probabilistic processes, we show that we can emulate any logical formula by a probabilistic adversary: for all formulas $F$, we build a probabilistic adversary $Adv$ such that for all extended processes $(P, \phi)$, $P \cup \{ \{Adv\phi\}\}$ exhibits the adversarial barb $ok$ with probability 1, if and only if the state $(P, \phi)$ satisfies the formula $F$ in $L^t$.

Definition 27. Let $F$ be a formula in $F$, $ok \in \mathcal{N}_{pub}$ such that $ok \not\in fn(P)$ and $n \in \mathbb{N}$. We define $Adv^{ok}_{F,n}$ by induction on the syntax of $F$:

- if $F = \top$ then $Adv^{ok}_{F,n} = out(ok, ok)$.
- if $F = in(\xi, \zeta).F'$ then $Adv^{ok}_{F,n} = out(\xi, \zeta); Adv^{ok}_{F',n}$ when $vars(\xi, \zeta) \subseteq A\mathcal{X}_n$ and $Adv^{ok}_{F',n} = 0$ otherwise.
- if $F = out(\xi, ax).F'$ then $Adv^{ok}_{F,n} = in(\xi, ax); Adv^{ok}_{F',n+1}$ when $ax = ax_{n+1}$, $vars(\xi) \subseteq A\mathcal{X}_n$ and $Adv^{ok}_{F',n} = 0$ otherwise.
- if $F = (\xi = \zeta).F'$ then $Adv^{ok}_{F,n} = 0$ otherwise.
- if $F = (\xi \not= \zeta).F'$ then $Adv^{ok}_{F,n} = 0$ otherwise.
- if $F = F_1 \land F_2$, then $Adv^{ok}_{F,n} = Adv^{ok}_{F_1,n} + 1/2 Adv^{ok}_{F_2,n}$.

In Definition 27, the integer $n$ and the conditions on the variables of $\xi, \zeta$ ensure that the adversarial process $Adv^{ok}_{F,n}$ is closed (no free variables). This is not a restriction as $(P, \emptyset) \models F$ implies that $F$ satisfies these conditions.

Lemma 12. Let $P \in \mathcal{M}P^{np}$ and $ok \in \mathcal{N}_{pub}$ such that $ok \not\in fn(P)$. For all formula $F \in \mathcal{F}$, we have

$$ (P, \emptyset) \models F \iff \text{RProb}_R((P \cup \{Adv^{ok}_{F,0}\}, 1) = 1 $$

The proof of Lemma 12 can be found in Appendix G.1. Notice that the lemma in fact also holds for processes with replication. The main result of this section follows almost directly.

Proposition 8. Let $P, Q \in \mathcal{M}P^{<\infty, np}$.

$$ (P, \emptyset) \leq^N_{sim} (Q, \emptyset) \iff P \leq_{may} Q $$
Proof. Suppose \((P, \emptyset) \leq_{N^f_{\text{sim}}} (Q, \emptyset)\). By Proposition 5, \(P \leq_{R_{\text{obs}}} Q\), which implies \(P \leq_{\text{may}} Q\) by definition.

Let us now assume \(P \leq_{\text{may}} Q\). From Lemmas 10 and 11, we know that \((P, \emptyset) \leq_{N^f_{\text{sim}}} (Q, \emptyset)\) if and only if \((P, \emptyset) \leq_{L^f_{\text{ssim}}} (Q, \emptyset)\). Moreover, since \(P\) and \(Q\) are bounded, we deduce that \((P, \emptyset) \leq_{N^f_{\text{ssim}}} (Q, \emptyset)\) if and only if \((P, \emptyset) \leq_{L^f_{\text{ssim}}} (Q, \emptyset)\). As \(L^f_{\text{ssim}}\) is image finite, we can rely on Proposition 7 to obtain that \((P, \emptyset) \leq_{N^f_{\text{ssim}}} (Q, \emptyset)\) if and only if for all \(F \in \mathcal{F}\), \((P, \emptyset) \models F\) implies \((Q, \emptyset) \models F\). Therefore, let \(F \in \mathcal{F}\) such that \((P, \emptyset) \models F\). We take \(ok \in N_{\text{pub}}\) such that \(ok \not\in \text{fn}(F, P, Q)\). By Lemma 12, we have RProb\(_R\)(\(P \cup \{\{\text{Adv}_{F,0}^{ok}\}\}, \downarrow ok\)) = 1. Since \(P \leq_{\text{may}} Q\), we deduce that RProb\(_R\)(\(Q \cup \{\{\text{Adv}_{F,0}^{ok}\}\}, \downarrow ok\)) = 1 and so \((Q, \emptyset) \models F\) once again by Lemma 12.

6.2.2 Complexity

Cheval et al. have shown [CKR18] that both deciding trace equivalence and bisimilarity is coNEXPTIME complete when cryptographic primitives are modelled by a subterm convergent destructor rewrite system and the number of sessions is bounded. (We refer the reader to [CKR18] for a precise definition of this class of rewrite systems.) The hardness proof reduces SUCCINT 3Sat to both trace equivalence and bisimilarity using a same encoding which also proofs hardness of similarity. The coNEXPTIME decision procedure for bisimilarity can be directly adapted to the case of similarity, hence, by Proposition 8 we have the following result.

**Corollary 1.** Let \(P, Q \in MP^{<\infty, np}\). Deciding \(P \approx_{\text{may}} Q\) is coNEXPTIME complete when \(\approx\) is defined by a subterm convergent destructor rewrite system.

6.3 May-testing and simulation do not coincide for unbounded processes

To prove that may-testing and simulation do not coincide for processes with replication, we consider the following two LTS \(L_1\) and \(L_2\), that are an asymmetric variant of the LTSs used in [vG87] to show that the finitary Hennessy-Milner logic from Definition 26 do not characterise bisimulation.
Though both $L_1$ and $L_2$ may produce an unbounded number of $a$ transitions, the initial transition in $L_2$ decides on an arbitrary, but fixed number of $a$ transitions in the rest of the execution. This in particular shows that $L_2$ does not simulate $L_1$, i.e. $L_1 \not\leq_{\text{sim}} L_2$.

In the applied pi calculus, $L_1$ can be represented by the process $\nu c.a$. Modeling $L_2$ in the applied pi calculus is more complex as it requires to non-deterministically choose an integer $n \geq 0$, before outputting $n + 1$ times $a$. To non-deterministically choose $n$, we rely on the following process that outputs $n$ encoded in unary as $h^n(b)$:

Intuitively, the channel $d$ represents a memory cell initiated with a public name $b$ (encoding 0). When reading on channel $d$ the current value $x$, the process non deterministically chooses to increment $x$ (updating the cell with $h(x)$) or to select the value $x$ by outputting it on channel $e$.

It remains to model the process that given an integer $x$, produces $x + 1$ outputs of $a$:

Similarly to $Q_{\text{count}}(e)$, $Q_{\text{out}}(e)$ relies on a private $d'$ to increment $b$ until reaching the value $x$ read on $e$. It is easy to see that the process outputs $x + 1$ times $a$.

Lemma 13. Let $Q = \nu e.(Q_{\text{count}}(e) \mid Q_{\text{out}}(e))$. We have:

$(!\text{out}(c, a), \emptyset) \not\leq_{\text{sim}}^{N^e} (Q, \emptyset)$ but $!\text{out}(c, a) \leq_{\text{may}} Q$

Proof. To show that $(!\text{out}(c, a), \emptyset) \not\leq_{\text{sim}}^{N^e} (Q, \emptyset)$, we rely on the same idea used to show that $L_1 \not\leq_{\text{sim}} L_2$. Before the first output on $c$, the process $Q$ must produce a communication on $e$, hence fixing the value of $x$ used in $Q_{\text{out}}(e)$ which bounds the number of following outputs on $c$ by $x + 1$. As $!\text{out}(c, a)$ may output on $c$ an unbounded number of times, we conclude.

To prove that $!\text{out}(c, a) \leq_{\text{may}} Q$, we need to show that for all $Adv \in \mathcal{MP}$, for all $ch \in \mathcal{N}_{\text{pub}}$, if $\text{fn}(Adv) \subseteq \mathcal{N}_{\text{pub}}$ then $\text{RProb}_R(\{\text{out}(c, a)\} \cup Adv, \downarrow ch) \leq \text{RProb}_R(\{Q\} \cup$
Let $\mathcal{P} = \{!\text{out}(c, a)\}$. By unfolding the definition of $\text{RProb}_R(\mathcal{P} \cup \text{Adv}, \downarrow \text{ch})$, we know that:

$$\text{RProb}_R(\mathcal{P} \cup \text{Adv}, \downarrow \text{ch}) = \sup_{R = (\mathcal{S}_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}} \text{RProb}_R(s, \text{corr}_R^{-1}(\downarrow \text{ch})) = \sup_{R = (\mathcal{S}_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}} \sup_{n \in \mathbb{N}} \text{RProb}_R^\leq n(s, \text{corr}_R^{-1}(\downarrow \text{ch}))$$

Let $R = (\mathcal{S}_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}$, $s \in \mathcal{S}_R$ such that $\text{corr}_R(s) = \mathcal{P} \cup \text{Adv}$ and let $n \in \mathbb{N}$. By definition, the number of transitions from $s$ in the computation of $\text{RProb}_R^\leq n(s, \text{corr}_R^{-1}(\downarrow \text{ch}))$ is bounded by $n$. Thus, the resolution $R$ may at most unfold $n$ times the process $!\text{out}(c, a)$. By denoting $P^n = \text{out}(c, a)\ldots \text{out}(c, a)$, we can therefore easily build a resolution $R' = (\mathcal{S}_{R'}, \text{corr}_{R'}, \text{trans}_{R'}) \in \mathcal{R}$ and $s' \in \mathcal{S}_{R'}$ such that $\text{corr}_{R'}(s') = \{P^n\} \cup \text{Adv}$ and $\text{RProb}_{R'}^\leq n(s', \text{corr}_{R'}^{-1}(\downarrow \text{ch})) = \text{RProb}_R^\leq n(s', \text{corr}_R^{-1}(\downarrow \text{ch}))$.

However, one can easily show that $(P^n, \emptyset) \leq^N_{\text{sim}} (Q, \emptyset)$ ($Q$ only needs to choose integer $n-1$ to output exactly $n$ times $a$). Since, by Proposition 5, simulation implies may-testing, we have:

$$\sup_{R' = (\mathcal{S}_{R'}, \text{corr}_{R'}, \text{trans}_{R'}) \in \mathcal{R}} \sup_{n' \in \mathbb{N}} \sup_{\text{corr}_{R'}(s') = \{P^n\} \cup \text{Adv}} \text{RProb}_{R'}^\leq n'(s', \text{corr}_{R'}^{-1}(\downarrow \text{ch})) \leq \text{RProb}_R(\{Q\} \cup \text{Adv}, \downarrow \text{ch})$$

By selecting $n' = n$, we deduce that $\text{RProb}_{R'}^\leq n(s', \text{corr}_{R'}^{-1}(\downarrow \text{ch})) \leq \text{RProb}_R(\{Q\} \cup \text{Adv}, \downarrow \text{ch})$ and so $\text{RProb}_R^\leq n(s, \text{corr}_R^{-1}(\downarrow \text{ch})) \leq \text{RProb}_R(\{Q\} \cup \text{Adv}, \downarrow \text{ch})$. It allows us to conclude that $\text{RProb}_R(\mathcal{P} \cup \text{Adv}, \downarrow \text{ch}) \leq \text{RProb}_R(\{Q\} \cup \text{Adv}, \downarrow \text{ch})$. \hfill $\Box$

7 Fully Probabilistic Agents Communicating on Public Channels

We now do a formal comparison with the probabilistic concurrent systems considered in [CSV17, BCSV18]. They consider systems that are built as the parallel compositions of independent agents, called roles, that are not able to communicate directly with each other: all communications are mediated by the adversary. Moreover, there is no non-determinism in the internal behavior of a role; the only non-determinism is controlled by the adversary—thus external—and consists in the adversary’s choice for managing the communications.

**Definition 28.** We say that a process $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\} \in \mathcal{MP}$ is fully probabilistic (FP) when:

- there is no non-deterministic choice $+$ in $\mathcal{P}_i$, and no parallel composition $|$ and no $!;$
there exists a list $u_1, \ldots, u_n$ of public channels such that all input and output actions in $P_i$ are only done on $u_i$.

**Notation 11.** For a FP process $Q = Q_1 | \ldots | Q_m$, and $j \in \{1, \ldots, m\}$, we write $\text{out}(c_j, t) \cdot Q$ and $\text{in}(c_j, x) \cdot Q$ for $(Q_1 | \ldots | \text{out}(c_j, t) \cdot Q_j | \ldots Q_m)$ and $(Q_1 | \ldots | \text{in}(c_j, x) \cdot Q_j | \ldots Q_m)$ respectively.

**Remark 11.** Observe that when looking at the un-labelled semantics for a component $P_i$ without the syntaxic constructs + and $\mid$, the only source of non-determinism in $\tau \rightarrow$ comes from $\text{new}$-redexes. To overcome this residual non-determinism, we replace as in Section 6 the NPLTS $N^\ell$ by a strict $\text{new}$-determinization of $N^\ell$. In the present section, by abuse of notation, we will write $N^\ell$ for an (arbitrary) $\text{new}$-determinization of $N^\ell$.

In [CSV17, BCSV18], the authors consider the trace equivalence for a restricted fragments of FP processes. More precisely, their processes have a constrained shape, used to suppress the necessity of scheduling decisions for inteleaving of honest $\tau$-actions, and thus ensure that weak trace equivalence and strong trace equivalence coincide.

**Remark 12.** Observe that at this point, fully probabilistic processes have no replication. As a consequence, whenever $s \xrightarrow{\tau} E$, it is actually a finite scheme, and the sub-distribution $E$ is actually a proper distribution.

### 7.1 Removing residual non-determinism

In this section, we are interested in the fragment of $N^\ell$ whose states are of the form $(P, \phi)$, with $P$ a FP processes. Observe that this fragment has still internal non-determinism, since starting from a state $(P_1 | \ldots | P_n, \phi)$, it can be possible to do a $\tau$ action either in a component $P_i$, or in a component $P_j$. However, this internal non-determinism is much more limited that for general processes: indeed, intuitively, that we chose to reduce first $P_i$ or first $P_j$ will not have an effect on the resulting trace equivalence. In this section, we formalize this intuition by transfering known confluence results from (non-probabilistic) abstract rewriting systems to NPLTSs.

The first step consists in building, from any NPLTS $N$ with only one action $\{\tau\}$, an abstract rewriting system $\overline{N}$ by moving all information about probabilities from transition to states.

**Definition 29.** Let $N = (S, \{\tau\}, h_\tau)$ be a NPLTS. We define (non-probabilistic) abstract rewriting system $\overline{N}$ i.e. a LTS with only one action $\tau$–as follows:

- its set of states is $\{\mu = [(\alpha_1, s_1), \ldots, (\alpha_n, s_n)] \in \mathcal{M}_{fin}([0, 1] \times S) \mid \sum \alpha_i = 1\}$.
- its only action is $\tau$;
- the transition function $\xrightarrow{\tau}$ is defined as follows:

$$
s \xrightarrow{N} E \quad (\alpha, s) \in \text{Support}(\mu)$$

$$\mu \xrightarrow{\tau} \overline{N} \cup \{((\alpha \cdot D(t), t)) \cup \mu \setminus \{(\alpha, s)\} \}$$
We formalize in Lemma 14 below how we can recover informations about probabilistic executions in a NPLTS \( \mathcal{N} \) by looking at (non-probabilistic) execution in the abstract rewriting system (Newmann’s Lemma) that a ARS with no infinite sequence.

**Lemma 14.** Let \( \mathcal{N} \) be a NPLTS with only one \( \tau \) action and no infinite sequence, \( s \) be a state in \( \mathcal{N} \), \( A \subseteq \mathcal{S}_{ext}(\mathcal{N}) \) a set of external states, and \( \alpha \in [0,1] \). Then there exists a non-randomized (maximal) resolution \( R \in \mathcal{R}_{nr}(\mathcal{N}) \) such that \( R_{\text{Prob}}(s, \text{corr}^{-1}(A)) = \alpha \) if and only if there is a finite (maximal) sequence \( \sigma := \left\{ (1, s) \right\} \xrightarrow{\mathcal{N}} \mu_1 \xrightarrow{\mathcal{N}} \ldots \mu_n \), and \( \alpha = \sum_{i,t_i \in A} \alpha_i \) with \( \mu_n = \left\{ (\alpha_1, t_1), \ldots, (\alpha_m, t_m) \right\} \).

**Proof.** \( \Rightarrow \): Let \( R \in \mathcal{R}_{nr}(\mathcal{N}) \), \( s_R \in \mathcal{S}_R \) with \( \text{corr}_R(s_R) = s \), and such that \( R_{\text{Prob}}(s_R, \text{corr}^{-1}(A)) = \alpha \). Since \( \mathcal{N} \) has no infinite sequence, there exists \( n \in \mathbb{N} \) such that \( R_{\text{Prob}}^n(s_R, \text{corr}^{-1}(A)) = \alpha \), and when \( R \) is maximal, we can moreover require that \( \{ t \mid R_{\text{Prob}}^n(s_R, t) > 0 \} \subseteq \{ u \mid \text{corr}_R(u) \in \mathcal{S}_{ext}(\mathcal{N}) \} \). We build a reduction sequence \( \mu_0 = \left\{ (1, s) \right\} \xrightarrow{\mathcal{N}} \mu_1 \xrightarrow{\mathcal{N}} \ldots \mu_n \in \overline{\mathcal{N}} \) by first defining a family \( (\rho_j)_{j \in \mathbb{N}} \) as \( \rho_j := \left\{ (D_j(u), \text{corr}_R(u)) \mid u \in \mathcal{S}_R \right\} \), with \( D_j \) the distribution obtained after \( j \) steps from \( s' \) in \( R \). Then we can observe that for every \( j \in \mathbb{N} \), it holds that there exists a \( \mathcal{N} \) such that \( \rho_j \xrightarrow{\mathcal{N}} \rho_{j+1} \), and from there we have the result. Moreover, if \( R \) is maximal, \( \text{Support}(D_n) \subseteq \{ u \mid \text{corr}_R(u) \in \mathcal{S}_{ext}(\mathcal{N}) \} \), thus \( \rho_n \) is a normal from in \( \overline{\mathcal{N}} \).

\( \Leftarrow \): Let \( \sigma \) be a reduction sequence \( \mu_0 = \left\{ (1, s) \right\} \xrightarrow{\mathcal{N}} \mu_1 \xrightarrow{\mathcal{N}} \ldots \mu_n \in \overline{\mathcal{N}} \). We build a resolution \( R \in \mathcal{R}_{nr}(\mathcal{N}) \) by induction on \( n \):

- if \( n = 0 \), the resolution has only one state \( s' \) with \( \text{corr}_R(s') = s \). Moreover, if \( \mu_n = \mu_0 \) is a normal from in \( \overline{\mathcal{N}} \), it means that \( s \) is an external state in \( \mathcal{N} \), thus \( R \) is a maximal resolution.

- Suppose that the result holds for \( n \in \mathbb{N} \). Let \( \sigma := \mu_0 = \left\{ (1, s) \right\} \xrightarrow{\mathcal{N}} \mu_1 \xrightarrow{\mathcal{N}} \ldots \mu_{n+1} \). We write \( \mu_1 = \left\{ (\alpha_1, s_1), \ldots, (\alpha_m, s_m) \right\} \). Looking at the definition of the ARS \( \overline{\mathcal{N}} \), we see that we can split \( \sigma \) into \( m \) sequences \( \left\{ (1, s_i) \right\} \xrightarrow{\mathcal{N}} \rho_{i,1} \xrightarrow{\mathcal{N}} \ldots \rho_{i,n_i} \), with \( n_i \leq n \), and \( \mu_n = \left\{ (\alpha_i \cdot \beta, s) \mid i \in \{1, \ldots, m\}, (\beta, s) \in \rho_{i,n_i} \right\} \). From there, we can apply the induction hypothesis, and we obtain a resolution \( R_i \) for \( i \in \{1, \ldots, m\} \), and a \( s'_i \in \mathcal{S}_{R_i} \) with \( \text{corr}_{R_i}(s'_i) = s_i \). We can suppose that the \( \mathcal{S}_{R_i} \) are all disjoints. From there, we build a resolution \( R \) as the disjoint union of all \( R_i \), plus a fresh state \( s'_0 \) with \( \text{corr}(s'_0) = s \), and \( \text{trans}_{\mathcal{N}}(s'_0) = \sum \alpha_i \cdot \delta_{s'_i} \). Moreover, if \( \mu_n = \) is a normal form in \( \overline{\mathcal{N}} \), then also the \( \rho_{i,n_i} \) are normal from: it means that the induction hypothesis tells us that each of the \( R_i \) is maximal, and from there we can conclude that \( R \) also is maximal.

\( \square \)

In the next step, we show that confluence and termination properties are passed down from \( \mathcal{N} \) to \( \overline{\mathcal{N}} \). We are interested in these properties, because it is a well-known result of the literature on abstract rewriting system (Newmann’s Lemma) that a ARS with no infinite
Moreover, since Rammann’s Lemma, we can see that Proposition 9. Let N be a NPLTS with only one τ action, that has no infinite sequence, and which is locally confluent. Let A ⊆ S_{ext}(N). Then for any maximal resolution R in R_{nr}(N) it holds that:

\[ \text{RProb}_{R}(s_{R}, \text{corr}_{R}^{-1}(A)) = \text{RProb}_{R_{nr}(N)}(\text{corr}_{R}(s_{R}), A). \]

Proof. Let A ⊆ S_{ext}(N). If ρ = \{(α_{1}, t_{1}), \ldots, (α_{m}, t_{m})\} ∈ S_{N}, we write ρ(A) := \sum_{j \text{ s.t. } t_{j} \in A} α_{j}. Let R_{1} in R_{nr}(N) be a maximal resolution, and R_{2} ∈ R_{nr}(N) any resolution. Then we obtain by Lemma 14 two finite sequences \( σ_{i} := \{(1, s_{i})\} \xrightarrow{σ_{i} \in S_{N}} \mu_{i}^{1} \xrightarrow{σ_{i}} \mu_{i}^{2} \ldots \mu_{i}^{n}, \) and \( \text{RProb}_{R_{1}}(s_{i}, \text{corr}_{R_{1}}^{-1}(A)) = \sum_{j \text{ s.t. } t_{j} \in A} α_{j}^{i} \text{ with } μ_{i}^{n} = \{(α_{1}^{i}, t_{1}^{i}), \ldots, (α_{m}^{i}, t_{m}^{i})\} \) for \( i ∈ \{1, 2\}. \)

Moreover, since \( R_{1} \) is maximal, \( μ_{1}^{n} \) is a normal form for \( N_{\mu} \). Using Lemma 15 and Newmann’s Lemma, we can see that \( N_{\mu} \) is strongly normalizing, thus \( μ_{2}^{n} \xrightarrow{N_{\mu}} μ_{1}^{n}. \) In order to conclude, it is enough to see that since \( A ⊆ S_{ext}(N), \) \( ρ \xrightarrow{N_{\mu}} ρ' \) implies that \( ρ(A) ≤ ρ'(A). \)


7.1.1 Determinizing may-testing for FP processes

**Proposition 10.** Let $P$ be an FP process, $\phi$ a frame and $Adv$ a fully determinate adversary. Let $A$ be a target subset of states of the form $P' \cup Adv'$, with $P'$ a FP process, and $Adv'$ a fully deterministic adversary. Then for any maximal non-randomized resolution $R$ on $N^o$:

$$R_{\text{Prob}}(P \cup Adv, A) = R_{\text{Prob}_{(N^o)}}(P \cup Adv, A).$$

**Proof.** We first define a NPLTS $(N^o)_A$ we obtain from $N^o$ as follows:

- We remove all transitions starting from states $a \in A$;
- we add a state $success$, and from each state $a \in A$, we add the transition $a \rightarrow \delta_{success}$

We can see that to each resolution $R$, we can associate a resolution $R_A$ on such that:

$$R_{\text{Prob}}(P \cup Adv, A) = R_{\text{Prob}_{(N^o)}}(P \cup Adv, success)$$

Since neither FP processes nor fully deterministic adversaries have $!$, the (smallest) fragment of $N^o$ containing them has no infinite reduction sequence, and moreover using the fact that we have no meaningful non-determinism, we can show that $N^o$ is locally confluent. As a consequence, it is also the case for $(N^o)_A$. From there--and using the fact that $A$ is a set of external states for $(N^o)_A$, we can conclude as a direct consequence of Proposition 9. 

7.1.2 On trace equivalence for FP processes

It is possible to characterise trace equivalence as the probability of reaching a particular external state $success$ on a NPLTS $(N^\ell)_{tr}$ with only one $\tau$ action, built from $N^\ell$. The construction is done in Definition 44 in Section B.1 and the characterisation is shown in Lemma 33. Since FP processes have no $!$, the FP fragment of $N^\ell$ has no infinite reduction sequences labelled by $\tau$, and moreover is locally confluent. Looking at the way we defined $(N^\ell)_{tr}$, we say that our construction preserves these properties.

As a corollary, and using Lemma 32, we obtain that we can characterise the trace preorder using the ARS $(N^\ell)_{tr}$.

**Proposition 11.** For any non-randomized maximal resolution $R$ on $N^\ell$:

$$\forall s \in S_R, \forall w \in A^*_{ext}, \text{Prob}_R(s, w) = \text{Prob}_{R_{\ell}}(\text{corr}_R(s), w).$$

**Proof.** First, observe that the FP fragment of $N^\ell$ has no infinite $\tau$-reduction sequence, and is locally confluent. Then, we can see that those properties can be passed down to $(N^\ell)_{tr}$. From there, we can conclude as a direct consequence of the characterisation of the success probability of reaching a trace from Lemma 33 and Proposition 9.

**Remark 13.** The proofs in the present section uses heavily the fact that there is no replication into FP-processes. Indeed, suppose that we consider instead processes of the form $\mathcal{P}_1 | \ldots | \mathcal{P}_n$, where the processes $\mathcal{P}_i$ have no non-determinism, but can implement
recursion, then Proposition 11 would fail. It is because maximal resolutions can then lead to non-optimal strategies, where non-essential part of the system can execute a loop infinitely often. Nonetheless, we believe that is possible–for a given trace $w$–to build one optimal resolution $R_w$ that would maximise $\text{Prob}_{R}(s, w)$ for each process $s$, thus allowing us to remove non-determinism in computing the success probability for traces.

7.2 May testing equivalence of FP processes with unrestricted adversary.

As shown by the following example, even on [CSV17, BCSV18]’s fragment, trace equivalence do not coincide with may testing equivalence.

Example 4. Consider again the two processes of Example 3.

$$P := (\text{in}(c, x). \text{if } x = 0 \text{ then } \text{out}(c, ok) \text{ else } \text{out}(c, bad)) + 1/2$$

$$Q := \text{in}(c, x). (\text{out}(c, ok) + 1/2 \text{ out}(c, bad))$$

We have already shown in Example 3 that they are not testing equivalent (using a non-determinate adversary). We can however show easily that $(P, \emptyset)$ and $(Q, \emptyset)$ are trace equivalent. Indeed, the relevant traces are of the form $t_k := \text{in}(c, k).\text{out}(c, x_1). (x_1? = ok)$, for all the integers $k \in \mathbb{N}$. We can see that for every $k$, $\text{Prob}_{R}(((P, \emptyset), t_k) = \text{Prob}_{R}(((Q, \emptyset), t_k) = 1/2$.

7.3 A fully probabilistic operational semantics for FP processes and determinate adversaries

However, if we restrict the set of possible adversary and that we consider only adversaries without non-determinism–i.e. written whitout using + or |–we obtain a (weaker) notion of may testing equivalence for FP processes. The main result of this section is that this weaker may testing coincide with trace equivalence.

Definition 30. We say that a process $Adv$ is fully determinate if it is generated by the following grammar:

$$P, Q := 0 \mid \text{in}(u, x) \cdot P \mid \text{out}(u, v) \cdot P \mid \text{new} \ a \cdot P \mid \text{if} \ u = v \ \text{then} \ P \ \text{else} \ Q \mid P +_p Q$$

Moreover, we ask for a sucess token–formally a channel $ok$–that only the adversary can use. We say that two process $P, Q$ are determinate may testing equivalent whenever they are indistinguishable by fully determinate adversaries.

Our goal is to show that two FP-processes are in the trace pre-ordered if and only if they are in the determinate may-testing preorder. Observe that one direction is immediate: it is
enough to show that any trace can be encoded by a determinate adversary $Adv$, thus two
process that are in the determinate may-testing pre-order are also in the trace pre-order.
In the remaining of this section, we show that the reverse implication also holds.

**Definition 31.** Given a fully determinate process $Adv$ and an integer $n$, we define the set $Tr^{ok}(Adv, n)$ inductively on $Adv$ as:

- $Tr^{ok}(0, n) = \emptyset$;
- $Tr^{ok}(in(u, x); Adv, n) = \{(p_i, (u \neq ok).out(u, ax_{n+1}).w_i)\}^{m}_{i=1}$ when $Tr^{ok}(Adv\{\alpha_{n+1}/x\}, n+1) = \{(p_i, w_i)\}^{m}_{i=1}$
- $Tr^{ok}(out(u, v); Adv, n) = \{(p_i, (u \neq ok).in(u, v).w_i)\}^{m}_{i=1} \cup \{(1, u = ok)\}$ when $Tr^{ok}(Adv, n) = \{(p_i, w_i)\}^{m}_{i=1}$
- $Tr^{ok}(Adv_1 + \nu Adv_2, n) = \{(p \cdot p'_{k}, w'_{k})\}^{k}_{k=1} \cup \{(1 - p) \cdot p^2_{k}, w^2_{k}\}^{k}_{k=1}$ when $Tr^{ok}(Adv_1, n) = \{(p_i, w_i)\}^{m}_{i=1}$ for $i = 1, 2$
- $Tr^{ok}(\text{if } u = v \text{ then } Adv_1 \text{ else } Adv_2, n) = \{(p_1^{1}, (u = v).w_1^{1})\}^{n_1}_{k=1} \cup \{(p_2^{2}, (u \neq v).w_2^{2})\}^{n_2}_{k=1}$
- $Tr^{ok}(\text{new } a; Adv, n) = Tr^{ok}(Adv\{b/a\}, n)$ with $b \in \mathcal{N}_{pub}$ fresh.

The proof of the following lemma can be found in Appendix [H]

**Lemma 16.** Let $ok \in \mathcal{N}_{pub}$. Let $(\mathcal{P}, \phi)$ be a purely probabilistic process. Let $Adv$ be a fully determinate adversarial process such that $fv(Adv) \subseteq dom(\phi)$.

$$RProb_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P} \cup Adv\phi, \downarrow ok) = RProb_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P}, \downarrow ok) + (1 - RProb_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P}, \downarrow ok)) \times \sum_{(\alpha, w) \in Tr^{ok}(Adv, \{dom(\phi)\})} \alpha \cdot Prob_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P}, \phi, w)$$

We can now show the main proposition of this section.

**Proposition 12.** Let $\mathcal{P}, \mathcal{Q}$ be two FP processes. Then $(\mathcal{P}, \mathcal{Q})$ is in the may testing preorder for determinate adversaries if and only if $(\mathcal{P}, \emptyset)$ and $(\mathcal{Q}, \emptyset)$ are in the trace pre-order.

**Proof.** We focus on the non-straightforward implication: the trace preorder implies the may testing preorder for determinate adversaries. Let $Adv$ be a fully determinate adversary process and let $ok$ be a public name.

First, notice that as $\mathcal{P}$ and $\mathcal{Q}$ are both fully probabilistic processes, we have

$$RProb_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P}, \downarrow ok) = Prob_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P}, \emptyset, out(ok, ax_1))$$

$$RProb_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{Q}, \downarrow ok) = Prob_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{Q}, \emptyset, out(ok, ax_1))$$

Let $a_1 = Prob_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{P}, \emptyset, out(ok, ax_1))$ and $a_2 = Prob_{R\mathcal{N}_{\mathcal{N}}}^{\mathcal{N}
}(\mathcal{Q}, \emptyset, out(ok, ax_1))$. Since $(\mathcal{P}, \emptyset)$ and $(\mathcal{Q}, \emptyset)$ are in the trace pre-order, $a_1 \leq a_2$. 

52
Similarly, for all \((\alpha, w) \in Tr^ok(Adv, |\text{dom}(\phi)|)\), as \((\mathcal{P}, \mathcal{Q})\) and \((\mathcal{Q}, \mathcal{Q})\) are in the trace pre-order we have that \(\text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{P}, \mathcal{Q}), w) \leq \text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{Q}, \mathcal{Q}), w)\). Defining
\[
\begin{align*}
b_1 &= \sum_{(\alpha, w) \in Tr^ok(Adv, |\text{dom}(\phi)|)} \alpha \cdot \text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{P}, \phi), w) \\
b_2 &= \sum_{(\alpha, w) \in Tr^ok(Adv, |\text{dom}(\phi)|)} \alpha \cdot \text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{Q}, \mathcal{Q}), w)
\end{align*}
\]
we obtain that \(b_1 \leq b_2\).

Thus, by Lemma 16, \(\text{RProb}_{\mathcal{R}_m(\mathcal{N}_\ell)}(\mathcal{P}, \downarrow ok) = a_1 + (1 - a_1) \times b_1\) and \(\text{RProb}_{\mathcal{R}_m(\mathcal{N}_\ell)}(\mathcal{Q}, \downarrow ok) = a_2 + (1 - a_2) \times b_2\) with \(0 \leq a_1 \leq a_2 \leq 1\) and \(0 \leq b_1 \leq b_2 \leq 1\).

By considering the function \(f(x, y) = x + y - xy\), we notice that showing \(a_1 + (1 - a_1) \times b_1 \leq a_2 + (1 - a_2) \times b_2\) is equivalent to showing that \(f(a_2, b_2) - f(a_1, b_1) \geq 0\). By computing the partial derivations, we can see that \(f(a_2, b_2) \geq f(a_1, b_2) \geq f(a_1, b_1)\) which allows us to conclude.

\[\square\]

8 Deciding trace equivalence and tool support

As previous mentioned, Cheval et al. [CKR13] designed a decision procedure for trace equivalence when cryptographic primitives are modelled by a subterm convergent destructor rewrite system and a bounded number of sessions. This procedure is based on constraint solving techniques that represent the infinite set of all possible concrete executions of the processes and an arbitrary attacker as a finite symbolic tree, called the partition tree. Intuitively, each node of this symbolic tree represents the state of the two processes after executing a trace \(tr\). Due to non-determinism, a node may contain several constraint systems corresponding to every possible interleaving allowing the execution of a given trace \(tr\). Deciding trace equivalence between processes \(A\) and \(B\), in the original, non-probabilistic setting, requires to check that each node of the symbolic tree contains at least one constraint system derived from process \(A\) and one from process \(B\); or the node is empty.

We show how to extend this procedure in order to decide trace equivalence in a general setting where both probabilistic and non-determinism behavior may co-exist in the process. Obviously, we inherit the setting of a bounded number of sessions and cryptographic primitives modeled by a subterm convergent destructor rewrite system.

Following Proposition 2, proving \((\mathcal{P}, \phi) \leq_{tr} (\mathcal{P}', \phi')\) is equivalent to proving \((\mathcal{P}, \phi) \leq_{tr}^{nr} (\mathcal{P}', \phi')\). Thus, by definition, we focus on the computation of \(\text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{P}, \phi), w)\) for all \(w \in \mathcal{A}_{ext}^\ell\). A main difficulty stems from the presence of universal quantification over resolutions. However, as \(\mathcal{P}\) is bounded, we can completely ignore resolutions and focus on the labelled semantics, as shown by the following property.

Lemma 17. Let \((\mathcal{P}, \phi) \in SP_{\ell}^{\leq \infty}\). Let \(a \in \mathcal{A}_{ext}^\ell\). Let \(w \in \mathcal{A}_{ext}^\ell\).

\[
\text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{P}, \phi), \epsilon) = 1 \quad \text{Prob}_{\mathcal{R}_m(\mathcal{N}_\ell)}((\mathcal{P}, \phi), a.w) = \max(p_1, p_2)
\]
where  
\[
p_1 = \max_{(P, \phi) \to_a D} \sum_{(P', \phi') \in \text{supp}(D)} D((P', \phi')) \cdot \text{Prob}_{\mathcal{R}_w(N')}((P, \phi), a.w) \\
p_2 = \max_{(P, \phi) \to_a D} \sum_{(P', \phi') \in \text{supp}(D)} D((P', \phi')) \cdot \text{Prob}_{\mathcal{R}_w(N')}((P, \phi), w)
\]

Note that this inductive definition is well founded as the size of the processes strictly decreases at each semantics step since there is no replication.

### 8.1 History

As mentioned above, partition trees $[\text{CKR18}]$ are a finite symbolic representation of the concrete executions of the two initial processes. Each node contains all reachable states of the two processes after executing some trace $w$. However, to compute the probability $\text{Prob}_{\mathcal{R}_w(N')}((P, \phi), w)$, Lemma $[\text{17}]$ also requires to know the different semantics steps that led to the process states after executing $w$. In other words, it requires the history of each extended processes. Therefore, to simplify the probability computation, we extended the labelled semantics by adding the history of transitions leading to the extended process. To further simplify, we assume that each input, output, probabilistic choice and non-deterministic choice are decorated with a label $\ell \in \mathcal{L}$, denoted $\text{in}^\ell(u, x); P$, $\text{out}^\ell(u, v); P$ and $P +_p^\ell Q$, $P +^\ell Q$ respectively. Additionally, we assume that all labels are distinct in the initial processes.

**Definition 32.** We define history entries as elements from \{h_1(\ell), h_2(\ell, \ell'), h_+(\ell, p, i), h_-(\ell, i) \mid \ell \in \{0, 1\}, p \in \{0, 1\}, i \in \{0, 1\}\}. A history, usually denoted $H$, is a sequence of history entries.

Extended processes and the labelled semantics can naturally be extended to include history. In particular when $(P, \phi) \to_a D$ and $(P', \phi') \in \text{supp}(D)$, we define $(P, \phi, H) \overset{a}{\to} (P', \phi', H')$ where:

- $H' = H \cdot h_+(\ell, p, 0)$ when $P = Q \cup \{P +_p^\ell Q\}$ and $P' = Q \cup \{P\}$;
- $H' = H \cdot h_+(\ell, p, 1)$ when $P = Q \cup \{P +_p^\ell Q\}$ and $P' = Q \cup \{Q\}$;
- $H' = H \cdot h_-(\ell, 0)$ when $P = Q \cup \{P +^\ell Q\}$ and $P' = Q \cup \{P\}$;
- $H' = H \cdot h_-(\ell, 1)$ when $P = Q \cup \{P +^\ell Q\}$ and $P' = Q \cup \{Q\}$;
- $H' = H \cdot h_2(\ell, \ell')$ when $P = Q \cup \{\text{out}^{\ell'}(u, t).P, \text{in}^\ell(v, x).Q\}$, $P' = Q \cup \{P, Q\}$;
- $H' = H \cdot h_1(\ell)$ when $P = Q \cup \{\text{out}^{\ell}(u, t).P\}$ and $a = \text{out}(\xi, ax_n)$ and $P' = Q \cup \{P\}$;
- $H' = H \cdot h_1(\ell)$ when $P = Q \cup \{\text{in}^\ell(u, x).P\}$ and $a = \text{in}(\xi, \zeta)$ and $P' = Q \cup \{P\}$;
- $H' = H$ otherwise.
Given \( w \in \mathcal{A}_{\text{ext}}^{\ell \ast} \), we write \((P, \phi, H) \xrightarrow{w} (P', \phi', H')\) when \((P, \phi, H) \xrightarrow{a_1} \ldots \xrightarrow{a_n} (P', \phi', H')\) and \( w \) is \( a_1 \ldots a_n \) with the \( \tau \) labels removed.

Since labels occur at most once in the initial process, intuitively, two extended processes with a shared prefix executed the same semantics steps. In other words, if

\[ (P, \phi, H) \xrightarrow{w} (P_1, \phi_1, H_0 \cdot H_1) \quad \text{and} \quad (P, \phi, H) \xrightarrow{w} (P_2, \phi_2, H_0 \cdot H_2) \]

then there exist \( w_0, w_1 \) and an extended process \( (P_0, \phi_0, H_0) \) such that \( w = w_0w_1 \) and

\[
\begin{align*}
(P, \phi, H) \xrightarrow{w_0} (P_1, \phi_1, H_0) \\
(P, \phi, H) \xrightarrow{w_1} (P_2, \phi_2, H_0)
\end{align*}
\]

We now describe how we can compute the probability that \( (P, \phi) \) executes a trace \( w \) from the set of histories obtained after executing \( w \), i.e. \( \{ H' \mid (P, \phi, []) \xrightarrow{w} (P', \phi', H') \} \) where [] denotes the empty sequence.

**Definition 33.** Let \( S \) be a set of histories. We denote by \( S|_{h} = \{ H' \mid (h \cdot H') \in S \} \). We define \( \text{compute}(S) \) inductively as follows:

- \( \text{compute}(\emptyset) = 0 \)
- \( \text{compute}(\{[]\}) = 1 \), i.e., if \( S \) is the singleton containing the empty sequence
- otherwise

\[
\text{compute}(S) = \max \left\{ \begin{array}{l}
\max_{\ell} \text{compute}(S|_{h_1(\ell)}) \\
\max_{\ell, \ell'} \text{compute}(S|_{h_2(\ell, \ell')}) \\
\max_{\ell,i} \text{compute}(S|_{h_3(\ell, i)}) \\
\max_{\ell,p} (p \cdot \text{compute}(S|_{h_4(\ell, p, 0)}) + (1 - p) \cdot \text{compute}(S|_{h_4(\ell, p, 1)}))
\end{array} \right\}
\]

The correspondence between \( \text{Prob}_{\mathcal{R}_{\mathcal{N}}(P)}((P, \phi), w) \) and the function \( \text{compute}(\cdot) \) is given in the following lemma.

**Lemma 18.** Let \( (P, \phi) \) be an extended process, \( w \in \mathcal{A}_{\text{ext}}^{\ell \ast} \) and \( S = \{ H' \mid (P, \phi, []) \xrightarrow{w} (P', \phi', H') \} \). We have that \( \text{Prob}_{\mathcal{R}_{\mathcal{N}}(P)}((P, \phi), w) = \text{compute}(S) \).

**Proof.** Let \( A = (P, \phi) \) be an extended process. Let \( w \in \mathcal{A}_{\text{ext}}^{\ell \ast} \). Let us define \( S(A, w) \) the set \( \{ H \mid (P, \phi, []) \xrightarrow{w} (P', \phi', H) \} \). Then, we show by induction on \(|A|, |w|\) that \( \text{compute}(S(A)) = \text{Prob}_{\mathcal{R}_{\mathcal{N}}(P)}(A, w) \).

55
Base case $([A], |w|) = (0, 0)$, i.e., $P = 0$ and $w = \epsilon$. In such a case, by Lemma 17 we deduce that $\text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = 1$. Moreover, by definition of $S(*, \epsilon)$, $S(*, \epsilon) = \{[]\}$ meaning that $\text{compute}(S(*, \epsilon)) = 1$. This allows us to conclude.

Inductive step $(|A|, |w|) > (0, 0)$. We do a case analysis on $w$ and the available semantics steps on $A$ (see Definition 10):

- Case $(\mathcal{P}, \phi) \rightarrow_a \delta_1(\mathcal{P}', \phi)$ with Nil, Par or Conditional rule: In such a case, as the rule are deterministic, two simple inductive proofs allow us to show that $\text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = \text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = \text{compute}(S(*, \epsilon)) = \text{compute}(S(*, \epsilon))$. This typically indicates that the order on which deterministic actions are applied does not matter. Since $|B| < |A|$, we can conclude by applying our inductive hypothesis.

- Case $w = aw'$ with $a = (\xi \sim \zeta)$: In such a case, $A \not\rightarrow_a \delta_A$ iff $(\mathcal{P}, \phi, []) \not\rightarrow (\mathcal{P}', \phi, []).$ Similarly to the previous case, we can easily show by induction that either (i) $\text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = 0$ and $A \not\rightarrow_a \delta_A$, meaning that $(\mathcal{P}, \phi, []) \not\rightarrow (\mathcal{P}', \phi, []).$ or (ii) $\text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = \text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon)$ and $A \rightarrow_a \delta_A$. In the latter case, we deduce by definition that $\text{compute}(S(*, \epsilon)) = \text{compute}(S(*, \epsilon))$. By inductive hypothesis, $\text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = \text{compute}(S(*, \epsilon))$ which allows us to conclude.

- Otherwise, the remaining semantic steps give us that for all $B = (\mathcal{P}', \phi')$, $(\mathcal{P}, \phi) \rightarrow_b D$ and $B \in \text{supp}(D)$ if and only if $(\mathcal{P}, \phi, []) \not\rightarrow (\mathcal{P}', \phi', [h])$. Moreover, due to distinct occurrences on our processes, we also deduce that two distinct transition steps cannot correspond to the same history. Therefore, $(\mathcal{P}, \phi, []) \not\rightarrow (\mathcal{P}', \phi', [h])$ implies that $S(*, \epsilon)|_{[h]} = S(*, \epsilon)$. Applying our inductive hypothesis on $(B, w')$, we deduce that $\text{compute}(S(*, \epsilon)) = \text{compute}(S(*, \epsilon))$. Hence, when $h$ does not correspond to a probabilistic choice, we have by definition that $D = \delta_B$ and so:

$$D(B) \cdot \text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = \text{compute}(S(*, \epsilon))$$

When $h = h_+((\ell, p, i))$ with $i \in \{0, 1\}$, we know that $b = r, \phi' = \phi$ and $(\mathcal{P}, \phi, []) \not\rightarrow (\mathcal{P}', \phi', [h_+((\ell, p, i - 1))])$ such that $\text{supp}(D) = \{B, B'\}$ with $B' = (\mathcal{P}', \phi)$. Therefore,

$$\sum_{s \in \text{supp}(D)} D(s) \cdot \text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) = p_i \cdot \text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon) + p_{1-i} \cdot \text{Prob}_{\mathcal{R}_w(N')}(*, \epsilon)$$

with $p_0 = p$ and $p_1 = 1 - p$.

By definition of $\text{compute}(S(*, \epsilon))$ and Lemma 17, we conclude. $\Box$

This lemma provides the core property that allows us to update the partition trees generated by the procedure in [CKR18] as well as the test performed on each node of this partition tree to conclude trace preorder.
8.2 Partition tree

The algorithm in [CKR18] relies on a symbolic semantics to finitely models the infinitely branching concrete semantics of the calculus. They consider symbolic processes \((P, C)\) where \(P\) is a multiset of (non-necessarily ground) processes and \(C\) is a constraint system containing a frame (non-necessarily ground). The frame included in a constraint system \(C\) is denoted \(\Phi(C)\). We naturally extend symbolic processes with histories: \((P, C, H)\).

In [CKR18], symbolic configurations are accompanied with a symbolic semantics as a labelled transition relation, denoted \(\ell \rightarrow_s\). We naturally extend this semantics to our probabilistic settings. Intuitively, if \((P, C) \ell \rightarrow_s (P', C')\) in [CKR18] semantics then \((P, C, p, H) \ell \rightarrow_s (P', C', p, H')\) where \(H'\) is follows the construction described in Section 8.1. Additionally, we add a semantics rule for the probabilistic choice where we compute the new probability.

\[
\begin{align*}
\left(\{\{P +_r Q\} \cup P, C, H\right) & \xrightarrow{\ell} \left(\{P\} \cup P, C, H \cdot h_+ (\ell, p, 0)\right) \\
\left(\{\{P +_r Q\} \cup P, C, H\right) & \xrightarrow{\ell} \left(\{Q\} \cup P, C, H \cdot h_+ (\ell, p, 1)\right)
\end{align*}
\]

We do not detail how constraint systems are generated here as we do not alter their generation. We however recall that a solution of a constraint system is a pair of substitutions \((\Sigma, \sigma)\) such that \(\Sigma\) is a substitution of recipes, hence representing how the attacker computes messages, and \(\sigma\) is the substitution instantiating the variables of the process. \(\Sigma\) and \(\sigma\) are called second-order and first-order substitutions respectively. Given a constraint system \(C\), the set of solutions is denoted \(Sol(C)\). Note that later on, when talking about partition tree, we will restrict the solutions of constraint systems that satisfy some predicates \(\pi\). These solutions will be denoted \(Sol^\pi(C)\).

The soundness and completeness of the semantics can be adapted from [CKR22] Proposition 3.2] as follows:

**Lemma 19** (soundness and completeness of the symbolic semantics). Let \((P, C, H)\) be a symbolic process.

1. **Soundness**: for all symbolic steps \((P, C, H) \xrightarrow{\ell} (Q, C', H')\) and \((\Sigma, \sigma) \in Sol(C')\), there exists a concrete labelled transition \((P_\Sigma, \Phi(C_\Sigma) \sigma_\downarrow, H) \xrightarrow{a\Sigma} (Q_\Sigma, \Phi(C'_\Sigma) \sigma_\downarrow, H')\).

2. **Completeness**: for all symbolic processes \((P, C, H)\), \((\Sigma, \sigma) \in Sol(C)\) and concrete traces \((P_\Sigma, \Phi(C_\Sigma) \sigma_\downarrow, H) \xrightarrow{\ell} (Q, \Phi, H')\), there exists a symbolic step \((P, C, H) \xrightarrow{\ell} (Q', C', H')\) and \((\Sigma', \sigma') \in Sol(C')\) such that \(\Sigma \subseteq \Sigma'\), \(Q = Q' \sigma'\), \(a = a' \Sigma'\) and \(\Phi = \Phi(C') \sigma'_\downarrow\).

We can now describe the partition trees that can be generated thanks to the procedure described in [CKR22]. Each node of a partition tree is in fact labelled by a configuration defined as follows.

**Definition 34** (configuration). A configuration is a pair \((\Gamma, \pi)\) where \(\Gamma\) is a set of symbolic processes and \(\pi\) a predicate on second-order substitutions. We also require that:
1. the predicate $\pi$ is defined on $\text{vars}^2(\Gamma)$, that is, for all $\Sigma$, $\pi(\Sigma)$ iff $\pi(\Sigma_{\text{vars}^2(\Gamma)})$;

2. for all $(\mathcal{P}_1, C_1, H_1), (\mathcal{P}_2, C_2, H_2) \in \Gamma$, if $(\Sigma, \sigma_1) \in \text{Sol}^\pi(C_1)$ then there exists $\sigma_2$ such that $(\Sigma, \sigma_2) \in \text{Sol}^\pi(C)$ and $\Phi(C_1)\sigma_1 \sim \Phi(C_2)\sigma_2$.

We state the properties satisfied by the partition tree.

**Definition 35** (partition tree). A partition tree of two bounded processes $P$ and $Q$ is a tree $T$ whose nodes are labelled by configurations and edges by visible symbolic actions, and that verifies the following properties. First of all $P, Q \in \Gamma(\text{root}(T))$ and $\pi(\text{root}(T)) = \top$, where $\text{root}(T)$ denotes the root node of the tree. Then for all nodes $n$ of $T$, $(\mathcal{P}, C) \in \Gamma(n)$ and visible symbolic actions $\alpha$:

1. **Closure by $\tau$-transition**: if $(\mathcal{P}, C, H) \xrightarrow{\tau} (\mathcal{P}', C', H')$ and $\text{Sol}^\pi(n)(C') \neq \emptyset$ then $(\mathcal{P}', C', H') \in \Gamma(n)$.

2. for all $(\mathcal{P}, C, p, H) \in \Gamma$, Sol$^\pi(C) \neq \emptyset$;

3. **All symbolic transitions are reflected in the tree**: if $(\mathcal{P}, C, H) \xrightarrow{s} (\mathcal{P}', C', H')$ and $(\Sigma, \sigma) \in \text{Sol}^\pi(n)(C')$ then there exists an edge $n \xrightarrow{\Delta} n'$ in $T$ such that $(\mathcal{P}', C', H') \in \Gamma(n')$ and $(\Sigma', \sigma) \in \text{Sol}^\pi(n')(C')$ for some $\Sigma'$ that coincides with $\Sigma$ on $\text{vars}^2(n)$.

Moreover for all edges $n \xrightarrow{\Delta} n_c$ of $T$ and $(\mathcal{P}_c, C_c, H_c) \in \Gamma(n_c)$:

4. **Predicates are refined along branches**: for all $\Sigma$, if $\Sigma$ verifies $\pi(n_c)$ then it verifies $\pi(n)$.

5. **Nodes are maximal**: if $(\Sigma, \sigma) \in \text{Sol}^\pi(n)(C)$, $(\Sigma_c, \sigma_c) \in \text{Sol}^\pi(n_c)(C_c)$ and $\Sigma \subseteq \Sigma_c$, then $\Gamma(n_c)$ contains all symbolic processes $(\mathcal{P}', C', H')$ such that $(\mathcal{P}, C, H) \xrightarrow{\Delta} (\mathcal{P}', C', H')$ and, for some substitution $\sigma'$, $(\Sigma_c, \sigma') \in \text{Sol}(C')$ and $\Phi(C_c)\sigma_c \sim \Phi(C')\sigma'$.

The set of partition trees of $P$ and $Q$ is written $\text{PTree}(P, Q)$.

The children of a node $n$ represent all the symbolic processes that can be reached by one symbolic transition from the processes of $n$. This corresponds to Item 3 of the definition. Note that these children a grouped by symbolic processes that are statically equivalence following Definition 34 of a configuration. Note that Item 5 indicates that if there were other symbolic processes statically equivalent to one in a configuration, then they should appear in the same configuration. If we denote by $A^\ell_{\text{ext}}$ the set of actions $A^\ell_{\text{ext}}$ minus the one of the form $\xi \sim \zeta$, we can generalize this property to trace in the following lemma.

**Lemma 20** (Adapted from [CKR22, Lemma A.1]). Consider a partition tree $T \in \text{PTree}(P, Q)$. Assume $(\mathcal{P}_1, C_1, H_1), \eta \xrightarrow{\text{ext}^T} (\mathcal{P}_1', C_1', H_1')$, $\eta'$ and $(\mathcal{P}_2, C_2, H_2) \xrightarrow{\text{ext}^T} (\mathcal{P}_2', C_2', H_2')$ with $(\mathcal{P}_2, C_2, H_2) \in \Gamma(\eta)$. For all solutions $(\Sigma', \sigma_1') \in \text{Sol}^\pi(\eta')(C_1')$ and $(\Sigma', \sigma_2') \in \text{Sol}^\pi(\eta')(C_2')$ such that $\Phi(C_1')\sigma_1' \sim \Phi(C_2')\sigma_2'$, we have $(\mathcal{P}_2, C_2, H_2), \eta \xrightarrow{\text{ext}^T} (\mathcal{P}_2', C_2', H_2'), \eta'$. 58
Lemma 21 (Adapted from [CKR22, Lemma A.2]). Consider a partition tree $T \in \mathcal{PTree}(P, Q)$. Let $\eta$ be a node of $T$ and $(P, C, H) \in \Gamma(\eta)$. If $(P, C, H) \xrightarrow{w} (P', C', H')$ and $(\Sigma, \sigma) \in \text{Sol}^{\pi(\eta)}(C')$ then there exist a node $\eta'$ and a substitution $\Sigma'$ such that $(P, C, H), \eta \xrightarrow{w} (P', C', H'), \eta'$ and $(\Sigma', \sigma) \in \text{Sol}^{\pi(\eta')}(C')$.

These two lemmas allow us to prove our main theorem.

Theorem 1. Let $P, Q \in \mathcal{MP}^{\prec \infty}$ and $\preceq$ be defined by a subterm convergent destructor rewrite system. Trace preorder $([P], \emptyset) \preceq_{tr} ([Q], \emptyset)$ is decidable.

Proof. First, we show that we can restrict the set of traces we need to verify. In particular, it suffices to look at traces $w \in \mathcal{A}^*_{\text{ext}}$. In other words, we can push static equivalence tests towards the end of traces. Moreover, for all concrete traces $w_1$ not containing static equivalence tests, it suffices to check a finite number of concrete traces towards the end of traces. Thus, if we denote by $W \setminus \{w\}$, we will show that $P, Q \in \mathcal{MP}^{\prec \infty}$ and $\preceq$ is decidable.

For all $(P', \phi', H') \in S_P(w_1) \cup S_Q(w_1)$, we can define a finite sequence of actions $w_2$ built only on actions of the form $\xi \preceq \zeta$ or $\xi \neq \zeta$ that exactly defines the equivalence class of $(P', \phi', H')$ in $(S_P \cup S_Q) \setminus$, i.e., for all $(P'', \phi'', H'') \in S_P(w_1) \cup S_Q(w_1)$, $\phi' \sim \phi''$ if and only if $(P'', \phi'', H'') \xrightarrow{w_2} (P'', \phi'', H'')$. Thus, if we denote by $W$ this set of traces sufficient for proving trace preorder, we have that $([P], \emptyset) \preceq_{tr} ([Q], \emptyset)$ if and only if for all $w \in W$, $\text{Prob}_{\mathcal{R}_{\text{ext}}(N')}([P], \emptyset), w) \leq \text{Prob}_{\mathcal{R}_{\text{ext}}(N')}([Q], \emptyset), w)$.

Second, thanks to procedure in [CKR18], we know that there exists a partition tree $T \in \mathcal{PTree}(P, Q)$. If we denote by $H(S)$ the set of histories of the (symbolic) extended processes in $S$, we will show that

$$\forall w \in W, \text{Prob}_{\mathcal{R}_{\text{ext}}(N')}([P], \emptyset), w) \leq \text{Prob}_{\mathcal{R}_{\text{ext}}(N')}([Q], \emptyset), w)$$

if and only if

$$\forall \eta \in T, \text{compute}(H(\Gamma_P(\eta))) \leq \text{compute}(H(\Gamma_Q(\eta)))$$

To do so, it suffices to prove the following two properties and applying Lemma [18]

- Soundness: $\forall \eta \in T, \exists w \in W$ s.t. $H(S_P(w)) = H(\Gamma_P(\eta))$ and $H(S_Q(w)) = H(\Gamma_Q(\eta))$
- Completeness: $\forall w \in W, \exists \eta \in T$ s.t. $H(S_P(w)) = H(\Gamma_P(\eta))$ and $H(S_Q(w)) = H(\Gamma_Q(\eta))$

Let us denote $\eta_r$ the root of $T$. To prove the soundness, take $\eta \in T$ and $(P, C, H) \in \Gamma(\eta)$. By definition, there exists $w_1$ such that $([P], C, \emptyset), \eta, \xrightarrow{w_1} (P, C, H), \eta$ or $([Q], C, \emptyset), \eta, \xrightarrow{w_1} (P, C, H), \eta$. W.l.o.g., let us assume that $([P], C, \emptyset), \eta, \xrightarrow{w_1} (P, C, H), \eta$.

1. By Item 2 of Definition 35 there exists $(\Sigma, \sigma) \in \text{Sol}^{\pi(\eta)}(C)$. 59
2. By Lemma 19, \((\{P\}, \emptyset, []) \xrightarrow{w} (P\sigma, \Phi(C)\sigma\downarrow, H)\).

3. By taking \(w \in W\) such that \(w_2\Sigma\) is a prefix of \(w\), i.e. \(w = w_2\Sigma w'\), and \((P\sigma, \Phi(C)\sigma\downarrow, H) \xrightarrow{w'} (P\sigma, \Phi(C)\sigma\downarrow, H)\), i.e. \(w'\) is the witness for the static equivalence, we obtain that \(H \in H(S_P(w))\).

4. By Item 2 of Definition 34, if \((P_2, C_2, H_2) \in \Gamma(\eta)\) and \((\{P\}, C_\emptyset, [])\), \(\eta \xrightarrow{w} (P_2, C_2, H_2)\), \(\eta\) then there exist \(\sigma_2\) such that \((\Sigma, \sigma_2) \in Sol_\pi(\eta)(C_2)\) and \(\Phi(C)\sigma \sim \Phi(C_2)\sigma_2\). Therefore, by applying once again Lemma 19, we obtain that \(H_2 \in H(S_P(w))\). Hence, \(H(\Gamma_P(\eta)) \subseteq H(S_P(w))\).

It remains to show that \(H(S_P(w)) \subseteq H(\Gamma_P(\eta))\): Take \((\{P\}, \emptyset, []) \xrightarrow{w} (Proc_2', \phi_2', H_2)\).

1. By Lemma 19, there exists \((\{P\}, C_\emptyset, []) \xrightarrow{w_3} (P_2, C_2, H_2)\) and \((\Sigma, \sigma_2) \in Sol_\pi(\eta)(C_2)\) such that \(P_2\sigma_2 = P_2'\) and \(\Phi(C_2)\sigma_2\downarrow = \phi_2'\).

2. By definition of \(w\), \(\Phi(C_2)\sigma_2 \sim \Phi(C)\sigma\).

3. By Lemma 20, we deduce that \((Proc_2, C_2, H_2) \in \Gamma(\eta)\). Hence, \(H(S_P(w)) \subseteq H(\Gamma_P(\eta))\).

Let us now prove the completeness property. Let \(w \in W\). Thus, we can split \(w = w_1w_2\), where \(w_1 \in \mathcal{A}_{ext}^\ell\) and \(w_2\) only contain actions of the form \(\xi \sim \zeta\). Thus, \((\{P\}, \emptyset, []) \xrightarrow{w} (P', \phi', H)\).

1. By Lemma 19, there exists \((\{P\}, C_\emptyset, []) \xrightarrow{w_3} (P, C, H)\) and \((\Sigma, \sigma) \in Sol(C)\) such that \(P\sigma = P'\) and \(\Phi(C)\sigma\downarrow = \phi'\).

2. By Lemma 21, there exist a node \(\eta\) and a substitution \(\Sigma'\) such that \((\{P\}, C_\emptyset, []) \xrightarrow{w_3} (P, C, H)\), \(\eta\) and \((\Sigma', \sigma) \in Sol_\pi(\eta)(C)\). Thus, \(H \in H(\Gamma_P(\eta))\). Moreover, we already shown that \(H(S_P(w_1\Sigma'w_2)) = H(\Gamma_P(\eta))\).

3. Consider now \((\{P\}, \emptyset, []) \xrightarrow{w_1w_3} (P_2', \phi_2', H_2)\). Once again by Lemma 19, there exists \((\{P\}, C_\emptyset, []) \xrightarrow{w_3} (P_2, C_2, H_2)\) and \(\sigma_2\) such that \(P_2\sigma_2 = P'\) and \(\Phi(C_2)\sigma_2\downarrow = \phi_2'\). Moreover, by definition of \(w\), \(\phi_2 \sim \phi'\).

4. Notice that since \((\Sigma, \sigma) \in Sol(C)\) and \((\Sigma', \sigma) \in Sol(C)\), we deduce that for all \(X \in dom(\Sigma), X\Sigma\phi' = X\Sigma\phi'.\) Hence \(X\Sigma\phi'_2 = X\Sigma\phi'.\) This allows us to deduce that \((\{P\}, \emptyset, []) \xrightarrow{w_1\Sigma w_3} (P_2', \phi_2', H_2)\) and so \(H(S_P(w_1\Sigma w_2)) \subseteq H(S_P(w_1\Sigma' w_2))\). With similar arguments, we can show that \(H(S_P(w_2)) \subseteq H(S_P(w_1\Sigma w_2))\) and so \(H(S_P(w)) = H(\Gamma_P(\eta))\).

As a direct corollary we obtain that we can decide determinate may testing for purely probabilistic processes as it coincides with trace equivalence.

**Corollary 2.** Let \(P, Q \in \mathcal{M}^\text{Ppp}\) and \(\vdash\) be defined by a subterm convergent destructor rewrite system. Determinate may testing preorder \((\{P\}, \emptyset) \leq_{d-may} (\{Q\}, \emptyset)\) is decidable.
We have implemented the procedure described above in the DeepSec tool. The input language has been extended with a probabilistic choice operator \( +_p \) where \( p \) is a real number in \([0, 1]\). When trace equivalence between 2 processes is queried, the tool uses the above procedure for verification. The development is available on DeepSec’s official github repository at [?].

As we will see in Section 9, we can use DeepSec to show that the dining cryptographers protocol does not provide anonymity when coins are biased.

9 An example: dining cryptographers

In this section, we look at how our formalism behave on a very standard of probabilistic protocol: dining cryptographers.

9.1 Dining cryptographers, in probabilistic pi-calculus.

**Notation 12.** For \( b, b' \in \{0, 1\} \), we note \( b \oplus b' \) for \( b \) xor \( b' \). (And similarly for the syntactic construction xor in the terms grammar).

We split the definition of the protocol in several component. The first step is to model the fact that two adjacent cryptographers can throw a dice, and obtain a probabilistic information bit that will be known only by them. We model this by the oracle process below (the two channels \( c_l \) and \( c_r \) model the pair of agents that throws the dice).

**Definition 36** (Oracle process). We define a process, which is indexed by two communication channels \( c_l \) and \( c_r \), as follows:

\[
O_{c_l, c_r} := (\text{out}(c_l, 0) \mid \text{out}(c_r, 0)) +_{1/2} (\text{out}(c_l, 1) \mid \text{out}(c_r, 1))
\]

We now define the process that correspond to one agent: this process is indexed by three channels—a channel \( c_l \) for communicating with the oracle placed to his right, a channel \( c_r \) for communicating with the oracle placed to his right, and a (public) channel \( c \) for sending its final answer. It also depends on whether the agent has payed or not, which we model by a Boolean \( b \).

**Definition 37** (Agent process). For every Boolean \( b \), we define a process indexed by three communication channels \( c_l, c_r \) and \( c \), as follows:

\[
A_{c_l, c_r, c}(b) := \text{in}(c_r, x_r); \text{in}(c_l, x_l); \text{out}(c, x_l \oplus x_r \oplus b).
\]

We are now ready to define the whole system: we put three agents, and their corresponding oracles, in parallel. Please notice that the channels used to communicate with the oracles are hidden from the adversary, thus binded by new, while the channels used by the agents to communicate their final results are free.

In order to be able to give a compact presentation of this system, we use a three-vertices graph. As represented in the graph below, we need to be able to talk about three families of elements:
• the set of agents (or vertices), that we note $\mathcal{A} = \{a_1, a_2, a_3\}$. (We take those to be public names, this way we can identify an agent with the (public) channel they use for returning their final result);

• the set of oracles (or edges), that we note $\mathcal{E} = \{e_1, e_2, e_3\}$;

• the set of (private) channels that allow the oracle to communicate their random bit to a pair of agent. We note this set $\mathcal{C} = \{c_1, ..., c_n\}$.

**Notation 13.** As represented on the figure below, we can associate at each vertex a pair of channel–the first element being the channel on its left, and the second the channel on its right; and similarly to each edge (i.e to each oracle). Reversely, we can associate to each channel both a vertex and an edge. We formalise this by two functions $c : \mathcal{A} \cup \mathcal{E} \rightarrow \mathcal{C} \times \mathcal{C}$, and $e : \mathcal{C} \rightarrow \mathcal{E}$, $a : \mathcal{C} \rightarrow \mathcal{A}$ (we note $c.l$ and $c.r$ for the first and second projection of $c$ respectively).

Similarly, for any pair of distinct edges $e_1, e_2 \in \mathcal{E}$, we note $a(e_1 \& e_2)$ for their (unique) common vertex, and $a(e_1 \setminus e_2)$, for the (unique) vertex that belongs in $e_1$ but not in $e_2$.

We are now ready to define a process that represents the whole protocol.

**Notation 14.** If $P_1, \ldots, P_n$ is a family of processes, we note $\|_{i \in \{1, \ldots, n\}} P_i$ for the process $P_1 \| \ldots \| P_n$.

**Definition 38.** Let $B : \mathcal{A} \rightarrow \{0, 1\}$ a Boolean-valued function giving the secret bit for each participant $a \in \mathcal{A}$.

$$DC_B := \text{new} \ (c_1, \ldots, c_n) \ (\|_{e \in \mathcal{E}} O_{c_l(e),c_r(e)} \, || \, a \in \mathcal{A} \ A_{c_l(a),c_r(a),a}(B(a)))$$

**Notation 15.** We call 1-weighted those valuation functions $B : \mathcal{A} \rightarrow \{0, 1\}$ such that $\text{card}(B^{-1}(\{1\})) = 1$. We write $\mathcal{W}_1$ for the set of all such functions.

**Notation 16.** In order to analyse the evolution of $DC$, we need a few notation for sub-processes:

1. First, we introduce a notation for the part of DC that has not changed: for $\mathcal{E}' \subseteq \mathcal{E}$, and $\mathcal{A}' \subseteq \mathcal{A}$, we note:

$$P_{\mathcal{E}',\mathcal{A}'}^B = \|_{e \in \mathcal{E}'} O_{c_l(e),c_r(e)} \, || \, a \in \mathcal{A}' \ A_{c_l(a),c_r(a),a}(B(a))$$

62
2. the next notation is for the oracle sub-processes that have already done their prob-
abilistic choice: for $E' \subseteq E$, an assignment function $f : E' \rightarrow \{0, 1\}$, and a set of
already fired pairs $L \subseteq (E' \times \{l, r\})$:

$$O_{E', f, L}^1 = \|_{(e, i) \notin L} (\text{out}(c_i(e), f(e))) .$$

3. We now look at how evolve the participants sub-processes. First, we introduce a
notation for the participants that have done exactly one input: for $A' \subseteq A$, and
$f : A' \rightarrow \{0, 1\}$, we note:

$$C_{A', f}^1 = \|_{a \in A'} \text{in}(c_1(a), x_1); \text{out}(a, f(a) \oplus x_r \oplus B(a)) ;$$

4. finally, we look at the participants that have already done two inputs, but that have
still to communicate their final result. For $A' \subseteq A$, and $f_1, f_2 : A' \rightarrow \{0, 1\}$, we note:

$$C_{A', f_1, f_2}^2 = \|_{a \in A'} \text{out}(a, f_1(a) \oplus f_2(a) \oplus B(a))$$

By abuse of notation, we will note for example $O_0^1$ instead of $O_{E', f}^1$, since it is possible to
recover the domain $E'$ from the function $f$.

Observe that $DC_{B, B'} = \text{new} \ (c_1, \ldots, c_n)P_{E', A}^B$.

9.2 Labelled semantics for the dining cryptographers protocol

Our goal is to show the following security property: for every two functions $B, B' : A \rightarrow
\{0, 1\}$ such that $\text{card}(i \mid C(i) = 1) = 1$ for $C = B, B'$, it holds that $DC_B$ and $DC_{B'}$ are
weakly bisimilar.

Unfortunately, the labelled semantics for $DC$ is quite complicated: in particular, since
we do not enforce a particular order for the communications between the oracles $O_i$ and
the participants $A_i$, we have both probabilistic and non-deterministic behaviour. As a con-
sequence, it is difficult to prove equivalence directly in the LMP for the labelled semantics.
To solve this problem, we define an auxiliary simplified LMP $N_{DC}$, and from there:

- We show that $N_{DC}$ is weakly bisimilar to the LMP for the labelled semantics;

- We show that the states corresponding to $DC_B$ and $DC_{B'}$ in $N_{DC}$ are bisimilar in
  $N_{DC}$.

As a first step, we formalise an intuitive notion of relevant steps in the execution of the
protocol, that we call events. The event $e[\leftarrow b]$ means that the oracle $e \in E$ is sampling
the Boolean $b \in \{0, 1\}$. The event $a[\rightarrow b]$ means that the participant $a \in A$ is outputting
the Boolean $b$. For technical reason, we separate internal events, that corresponds to
probabilistic choice, to external events, that correspond to a participant outputting its
result.
We define a NPLTS $N_{\text{DC}}$. For Notation 18. Proposition 13. There exists an NPLTS bisimulation $R$ on $N_{\text{DC}} \cup N_\ell$ such that for every $B : A \to \{0, 1\}$, it holds that $(\epsilon, B) R (DC_B, \varnothing)$. The next subsection is devoted to the proof of Proposition 13 above. More precisely, Proposition 13 is a direct corollary of Proposition 14 that will be shown at the end of the next sub-section.
9.3 \( N^\ell \) and \( N_{DC} \) are bisimilar

As a first step, we replace \( N^\ell \) by a new -determination of it.

**Notation 19.** We write \( \rightarrow_{\text{det}} \subseteq SP^\ell \times SP^\ell \) for the non-probabilistic \( \tau \)-reduction relation on \( N^\ell \). That is, \( (P, \phi) \rightarrow_{\text{det}} (Q, \psi) \) when \( (P, \phi) \overset{\delta_{(Q,\psi)}}{\rightarrow} \) in \( N^\ell \). Observe that no \( \rightarrow_{\text{det}} \) step can change the frame, thus it also implies that \( \phi = \psi \).

In order to handle more easily all the processes that can be reached during the execution of the protocol, we introduce the technical notions of interaction pair: intuitively it encodes the information about the internal communications between oracles and participants.

**Definition 41.** We call oracle interaction a set of the form \( S \subseteq \{(e, s) \mid e \in \mathcal{E}, s \in \{l, r\}\} \). We write \( I_O \) for the set of all oracle interactions. We write \( \text{dom}(S) \) for all \( e \) such that \( \exists s, (e, s) \in S \).

We call participant interaction an element of the form \( T \subseteq \{(a, s) \mid a \in \mathcal{A}, s \in \{l, r, \rightarrow \}\} \). We write \( \text{dom}(T) \) for all \( a \) such that \( \exists s, (a, s) \in T \). We say that such a participant is valid when for every \( a \in \mathcal{A}, (a, l) \in T \) implies that also \( (a, r) \in T, \) and \( (a, \rightarrow) \in T \) implies that both \( (a, l), (a, r) \in T \). We write \( I_P \) for the sets of all valid participants interactions. We say that a pair \( (S, T) \in I_O \times I_P \) is matching, and we note \( S \triangleleft T \) when: \( \forall (e, s) \in S, (a.s(e), \neg s) \in T, \) and \( \forall (a, s) \in T \) with \( s \in \{l, r\}, (e.s(a), \neg s) \in S \) (where \( \neg l = r, \) and \( \neg r = l \)).

**Notation 20.** To every matching pair \( S \triangleleft T \), and every function \( f : \mathcal{E}' \rightarrow \{0, 1\} \) where \( \text{dom}(S) \subseteq \mathcal{E}' \), and \( B \in \mathcal{W}_1 \), we can define uniquely the corresponding process and frame:

\[
P(B, f, S \triangleleft T) := P_{\mathcal{E}', \mathcal{A}, \text{dom}(T)}^B | O_{f,S}^1 | C_{\{a \in \text{dom}(T) \mid (a, l) \notin T\}, (a \rightarrow f(e.r(a)))}^1 | C_{\{a \in \text{dom}(T) \mid (a, l) \in T, (a, r) \notin T\}, (a \rightarrow f(e.r(a)), f(e.l(a)))}^2
\]

Observe that \( P(B, f, S \triangleleft T) \) is exactly the process obtained by doing all oracle probabilistic choice as specified by \( f \), and then the interaction between oracles and participants specified by \( S \triangleleft T \). We can also observe that it does not matter–i.e. the resulting process is the same–in which order the interactions and the probabilistic choices are done, as long as they happen as specified by \( f \) and \( S \triangleleft T \).

We first prove a caracterisation of matchings interaction pairs.

**Lemma 22.** Let \( \eta : \mathcal{E} \times \{l, r\} \mapsto (a.s(e), \neg s) \in \mathcal{A} \times \{l, r, \rightarrow \} \), and \( \xi \) the partial function defined as \( \xi : (a, s) \in \mathcal{A} \times \{l, r, \rightarrow \} \mapsto (e.s(a), \neg s) \in \mathcal{E} \times \{l, r\} \) when \( s \in \{l, r\} \). Then if \( S \) is an oracle interaction, and \( T \) a valid participant interaction, \( S \triangleleft T \) is a valid interaction pair if and only if \( \eta(S) \subseteq T \) and \( \xi(T) = S \).

**Proof.** First, we can see that \( \eta \circ \xi = \text{id}_{\mathcal{E} \times \{l, r\}} \), and moreover we can rewrite immediately the definition of matching pairs as: if \( S \) is an oracle interaction, and \( T \) a valid participant interaction, \( S \triangleleft T \) is a valid interaction pair iff \( \eta(S) \subseteq T \), and \( \xi(T) \subseteq S \). We now prove the intended result.
• Suppose that \( S \triangleleft T \) is an interaction pair. Observe that, since \( \xi \circ \eta = \text{id}_{\mathcal{E} \times \{l,r\}} \), we can deduce from the first inequality: \( S = \xi(\eta(S)) \subseteq \xi(T) \), which is enough to conclude.

• Suppose that \( S \) is an oracle interaction, and \( T \) a valid participant interaction, such that \( \eta(S) \subseteq T \), and \( S = \xi(T) \). Then the result is immediate.

Observe that a causal order can be defined on the notions of matching pairs: we will write \( S \triangleleft T \leq S' \triangleleft T' \) when the interaction defined in \( S' \triangleleft T' \) can be obtained by internal communications (between oracles and participants) from the interaction defined in \( S \triangleleft T \). We formalize this idea in the notation below.

**Notation 21.** We define a partial order \( \leq \) on matching interaction pairs by: \( (S \triangleleft T) \leq (S' \triangleleft T') \) when both \( (S \triangleleft T) \) and \( (S' \triangleleft T') \) are matching pairs in \( I_O \times I_P \), and moreover \( S \subseteq S' \), \( T \subseteq T' \). We define \( \leq^* \) by \( (S \triangleleft T) \leq^* (S' \triangleleft T') \) when \( (S \triangleleft T) \leq (S' \triangleleft T') \) and moreover there is no element of the form \( (a, \rightarrow) \) in \( T' \setminus T \).

**Lemma 23.** Let \( B \in \mathcal{W}_1 \), \( f : \mathcal{E}_0 \rightarrow \{0,1\} \). Let \( S \triangleleft T \) such that \( \text{dom}(S) \subseteq \mathcal{E}_0 \). Then for every \( S' \triangleleft T' \) with \( \text{dom}(S') \subseteq \mathcal{E}_0 \) and \( (S \triangleleft T) \leq^* (S' \triangleleft T') \), it holds that for every frame \( \phi \), \( P(B, f, S \triangleleft T), \phi \rightarrow_{\det} P(B, f, S' \triangleleft T'), \phi \).

**Proof.** Let \( S \triangleleft T \) be a matching interaction pair such that \( \text{dom}(S) \subseteq \mathcal{E}_0 \) and \( (S' \triangleleft T') \leq^* (S \triangleleft T) \). By definition of the preorder \( \leq \) on interaction pairs, it means that \( S' \subseteq S \), \( T' \subseteq T \). So we note \( n_S = \text{card}(S \setminus S') \), and \( n_T = \text{card}(T \setminus T') \) respectively. We’re going to do the proof by induction on \( n_S + n_T \). First, observe that if \( n = 0 \), then \( S = S' \), \( T = T' \), and we obtain immediately the result. We now consider the induction case. Suppose that \( n > 0 \). Observe that, by definition of \( \leq^* \), we know that there is no element of the form \( (a, \leftarrow) \in T \setminus T' \).

• Suppose that there is an element of the form \( (a, l) \in T \setminus T' \). We take \( T'' = T \setminus \{a, l\} \). Since \( (a, \leftarrow) \) is not in \( T \setminus T' \), it cannot be in \( T \) either (that’s because, since \( (a, l) \) is in \( T \setminus T' \), it is not in \( T' \), thus by definition of valid participant interaction, \( (a, \rightarrow) \notin T' \)). From there, we see that \( (a, \leftarrow) \notin T'' \subseteq T \), and thus \( T'' \) is again a valid participant interaction. We note \( S'' = S \setminus \xi(a,l) \)-observe that \( \xi \) is well-defined on this element. Using Lemma 22, we can check that \( S'' \triangleleft T'' \) is again a valid interaction pair: indeed \( \eta(S'') = \eta(S) \setminus \{\eta(\xi(a,l))\} = \eta(S) \setminus \{a,l\} \subseteq T \setminus \{a,l\} = T'' \), and \( \xi(T'') = \xi(T) \setminus \{\xi(a,l)\} = S'' \). So we obtain \( (S' \triangleleft T') \leq^* (S'' \triangleleft T'') \), and we can apply the induction hypothesis to \( S'' \triangleleft T'' \), thus it holds that: \( P(B, f, S' \triangleleft T'), \phi \rightarrow_{\det} P(B, f, S'' \triangleleft T''), \phi \). In order to conclude, it is enough to see that passing from \( S'' \triangleleft T'' \) to \( S \triangleleft T \) correspond to doing a communication on the left channel for the participant \( a \), and thus \( P(B, f, S'' \triangleleft T''), \phi \rightarrow_{\det} P(B, f, S \triangleleft T), \phi \).

• Suppose that there is no element of the form \( (a, l) \in T \setminus T' \), but an element of the form \( (a, r) \in T \setminus T' \). We take \( T'' = T \setminus \{a, r\} \). Again, we can see that \( T'' \) is again a valid participant interaction. We note \( S'' = S \setminus \xi(a,l) \)-observe that \( \xi \) is
we write only part of Looking at how we defined the components $O$.

Proof. Recall that by definition: every frame $\phi$ (\text{\textit{\textbullet}}) \rightarrow_{\text{det}} P(B, f, S' \triangleleft T')$, $\phi$. In order to conclude, it is enough to see that passing from $S'' \triangleleft T''$ to $S \triangleleft T$ correspond to doing a communication on the right channel for the participant $a$, and thus $P(B, f, S'' \triangleleft T'')$, $\phi \rightarrow_{\text{det}} P(B, f, S \triangleleft T)$, $\phi$.

• Actually, there is no other cases: indeed, when we are not in one of the previous case, it means that $T = T'$. But we can also use Lemma 22 that tells us that $\xi(T) = S$, and $\xi(T') = S'$. It means that $T = T'$, and $S = S'$, thus $n = 0$.

\begin{lemma}
Let $B \in W_1$, $f : \mathcal{E}' \rightarrow \{0, 1\}$. Let $S \triangleleft T$ such that $\text{dom}(S) \subseteq \mathcal{E}'$. Then for every frame $\phi$, $(P(B, f, S \triangleleft T), \phi) \xrightarrow{\text{out}(b', a)} D$ if and only if:

1. $\{(a, l), (a, r)\} \subseteq \text{dom}(T)$, and $(a, \rightarrow) \notin \text{dom}(T)$.

2. $b' = B(a) \oplus f(e.l(a)) \oplus f(e.r(a))$ (Observe that item 11 implies that $e.l(a), e.r(a) \in S$ because $T$ is valid, and $S \triangleleft T$ is a matching pair. Thus also $e.l(a), e.r(a) \in \mathcal{E}'$, and $b$ is well-defined.)

3. $D = (P(B, f, S \triangleleft T'), \phi \cup \{ax_{n+1} = b'\}$, where $n$ is the length of the $\phi$, and $T' := T \cup \{(a, \rightarrow)\}$.

\end{lemma}

\begin{proof}
Recall that by definition:

\[ P(B, f, S \triangleleft T) := \begin{array}{l}
P^B_{E \subseteq \mathcal{E}', A \theta \text{dom}(T)} \mid O^1_{f, S} \mid C^1_{\{a \in \text{dom}(T) \mid (a, l) \notin T, (a, \rightarrow) \notin T\}, (a, \rightarrow) \notin f(e.r(a))}\] \[ \mid C^2_{\{a \in \text{dom}(T) \mid (a, l) \notin T, (a, \rightarrow) \notin T, (a, \rightarrow) \notin f(e.r(a))\}}\] \[ \mid C^2_{\{a \in \text{dom}(T) \mid (a, l) \notin T, (a, \rightarrow) \notin T\}}\]

Looking at how we defined the components $O^1, C^1, C^2...$ in Notation 16, we see that the only part of $P(B, f, S \triangleleft T)$ that can do an output action is the $C^2$-component, i.e.:

\[ P := \{\{a \in \text{dom}(T) \mid (a, l) \notin T, (a, \rightarrow) \notin T\} \text{ out}(a, f(e.l(a)) \oplus f(e.r(a)) \oplus B(a))\}

We write $Q$ for the non $C^2$-part of $P(B, f, S \triangleleft T)$, that is for the process such that $P(B, f, S \triangleleft T) = P \mid Q$.

• Suppose that $(P(B, f, S \triangleleft T), \phi) \xrightarrow{\text{out}(b', a)} D$. It means that $D = \delta_{(P' \mid Q, \phi')}$, with $(P, \phi) \xrightarrow{\text{out}(b', a)} \delta_{(P' \mid Q, \phi')}$. Looking at the shape of $P$, we see immediately from there that we must have $a \in \{a \in \text{dom}(T) \mid (a, l) \notin T, (a, \rightarrow) \notin T\}$. Moreover, since $T$ is a valid participant interaction (by hypothesis), we can deduce from there that $a \in \{a \in \text{dom}(T) \mid (a, l) \notin T, (a, \rightarrow) \notin T, (a, \rightarrow) \notin T\}$, which is exactly item 1. Moreover—and again by looking at the shape of $P$, we see that the output $b'$ must
be: \( f(e.l(a)) \oplus f(e.r(a)) \oplus B(a) \): it is exactly item 1. We now look at the resulting process \( P' \): we see that it must be:

\[
P' = C^2_{\{a' \in \text{dom}(T) \mid (a', i) \in T, (a', \to) \in T\} \setminus \{a\}} \text{out}(a', f(e.l(a')) \oplus f(e.r(a')) \oplus B(a'))
\]

Thus we see that \( Q \mid P' = P(B, f, S \triangleleft T') \), with \( T' = T \cup \{(a, \to)\} \) and we can check easily that \( S \triangleleft T' \) is indeed a valid matching pair. Similarly, when we look at the frame \( \psi \), we see that it must be \( \psi = \phi \cup \{ax_n = f(e.l(a)) \oplus f(e.r(a)) \oplus B(a)\} \), and that concludes the proof of item 3.

- Suppose that the conditions item 1, item 2, item 3 hold. Let \( T' := T \cup \{(a, \to)\} \). We’re going to show that \((P, \phi) \xrightarrow{\text{out}(b, a)} D\). First, observe that by item 1, the sub-process \( \text{out}(a', f(e.l(a')) \oplus f(e.r(a')) \oplus B(a')) \) is in \( P \). Then, by item 2, we see that \( b' = \text{out}(a', f(e.l(a')) \oplus f(e.r(a')) \oplus B(a')) \), so we can indeed do the action \( \text{out}(b', a) \) from \( P' \). Finally, by item 3, we see that the target of this action is indeed \( D \).

\[ \square \]

**Notation 22.** An history \( h \) is compatible with an interaction pair \( S \triangleleft T \) when: \( \text{dom}(S) \subseteq \{e \mid e \in h\} \), and \( \{(a \mid (a, \to) \in T) = \{a \mid a \in h\} \). We note \( (S \triangleleft T) \Pi h \) when it is the case.

In order to show our simulation result, we will need the following auxiliary lemma:

**Lemma 25.** Let \( h = (s, l) \in H \), and \( S \triangleleft T \) an interaction matching pair compatible with \( h \). Suppose that \( e \not\in h \). Then \( h_0 := (s \cup [e \leftarrow 0], l), h_1 := (s \cup [e \leftarrow 1], l) \) are both compatible with \( S \triangleleft T \), and moreover:

\[
P(B, v_E(h), S \triangleleft T) \xrightarrow{\frac{1}{2} \cdot \delta_{P(B, v_E(h_0), S \triangleleft T)}} + \frac{1}{2} \cdot \delta_{P(B, v_E(h_1), S \triangleleft T)}.
\]

**Proof.** We first show that for \( i \in \{0, 1\} \), \( h_i \) is compatible with \( S \triangleleft T \). Observe that \( \text{dom}(S) \subseteq \text{dom}(E(h)) \subseteq \text{dom}(E(h_i)) \), and \( \{(a \mid (a, \to) \in T) = \text{dom} \Pi (h) = \text{dom} \Pi (h_i) \), so \( (S \triangleleft T) \Pi h_i \). We now look at the shape of \( P(B, v_E(h), S \triangleleft T) \): recall that

\[
P(B, v_E(h), S \triangleleft T) := P^B_{E \setminus \text{dom}_E(h), A \setminus \text{dom}(T)} \mid O^1_{v_E}, S \mid C^1_{\{a \in \text{dom}(T) \mid (a, l) \not\in T\}, (a \to \tau(e.r(a)))} \mid C^2_{\{a \in \text{dom}(T) \mid (a, l) \in T, (a, \to) \not\in T\}, (a \to \tau(e.r(a)), v_E(e.l(a))))
\]

We look at the part of the system that remain unchanged since the beginning of the execution, i.e. \( P^B_{E \setminus \text{dom}_E(h), A \setminus \text{dom}(T)} \). Recall that:

\[
P^B_{E \setminus \text{dom}_E(h), A \setminus \text{dom}(T)} = \|e \in E \setminus \text{dom}_E(h) \cdot O_{c_l(e), c_r(e)} \| \|a \in A \setminus \text{dom}(T) \cdot A_{c_l(a), c_r(a), a} \cdot B(a)\).
\]

Observe that, for every \( e_0 \in E \), \( O_{c_l(e_0), c_r(e_0)} \rightarrow \frac{1}{2} \cdot \delta_{\text{out}(c_l(e_0), 0)} \text{out}(c_r(e_0), 0) + \frac{1}{2} \cdot \delta_{\text{out}(c_l(e_0), 1)} \text{out}(c_r(e_0), 1) \). From there, we can deduce, for every \( e_0 \in E \setminus \text{dom}_E(h) \): \( P^B_{E \setminus \text{dom}_E(h), A \setminus \text{dom}(T)} \rightarrow \frac{1}{2} \cdot \delta_{Q_1(e_0)} + \frac{1}{2} \cdot \delta_{Q_2(e_0)}, \) where \( Q_1(e_0) := P^B_{E \setminus \text{dom}_E(h \cup \{e_0\}), A \setminus \text{dom}(T)} \cdot \text{out}(c_l(e_0), 0) \cdot \text{out}(c_r(e_0), 0) \). Now,
observe that, for $i \in \{0,1\}$, $O_{v_E(h),S}^1 \mid \text{out}(c_l(e_0),i) \mid \text{out}(c_r(e_0),i) = O_{v_E(h)+(e_0\mapsto i)}^1$. Now, putting everything together, we obtain, for every $e_0 \in \mathcal{E} \setminus \text{dom}_E(h)$:

$$P(B, v_E(h), S \triangleleft T) \rightarrow \frac{1}{2} \cdot \delta_{P(B, v_E(h)+(e_0\mapsto 0),S\triangleleft T)} + \frac{1}{2} \cdot \delta_{P(B, v_E(h)+(e_0\mapsto 1),S\triangleleft T)}.$$  

And we can conclude by observing that $v_E(h) + (e_0 \mapsto i) = v_E(h_i)$ for $i \in \{0,1\}$. 

\textbf{Definition 42.} We define a binary relation $R \subseteq \mathcal{S}_{DC} \times \mathcal{N}$. We are going to define $R$ as $R = \cup_{h,B}\{ (h,B) \times R_{h}^B \times \{ \phi_h \} \}$, Let be $h \in \mathcal{H}$, and $B \in \mathcal{W}_1$. We first look at the case where no result is outputted (i.e. the only non-empty part of the history $h = (s,l)$ is the set of internal actions $s$). Recall that a $h \in \mathcal{H}$ is characterised by the data of $(\mathcal{E}', \mathcal{A}', \leq, f : \mathcal{E}' \rightarrow \{0,1\}, g : \mathcal{A}' \rightarrow \{0,1\})$, where all these objects are as in Notation 18.

To build the $R_{h}^B$, we do a case disjunction on the respective cardinality of $\mathcal{A}'$ and $\mathcal{E}'$:

- if $\text{card}(\mathcal{A}') = \text{card}(\mathcal{E}') = 0$—that is, $h = e_h$—we take $R_{h}^B = \{(DC_B, \phi) \cup \{ (\text{new} (c_p, \ldots, c_n), P_{E,\mathcal{A}}, \phi) \mid 1 \geq p \geq n \}$;

- $R_{h}^B = \{ P(B, v, S \triangleleft T) \mid (S \triangleleft T \Pi h) \land (\forall a \in \mathcal{A}', g(a) = B(a) \oplus f(e.l(a)) \oplus f(e.r(a))) \land v = f \}.$

\textbf{Lemma 26.} Let be $B \in \mathcal{W}_1$, $h \in \mathcal{H}$. Suppose that $P \in S_{h}^B$.

- if $(h, B) \xrightarrow{\tau} D$. Then there exists $h_1, h_2$ with $D = \frac{1}{2} \cdot \delta_{(h_1,B)} + \frac{1}{2} \delta_{(h_2,B)}$, and $(P, \phi_h) \xrightarrow{\tau} \frac{1}{2} \cdot \delta_{(P_1,\phi_{h_1})} + \frac{1}{2} \cdot \delta_{(P_2,\phi_{h_2})}$, with $P_i \in S_{h_i}^B$ for $i \in \{1,2\}$.

- if $(h, B) \xrightarrow{a} (h', B)$. Then $(P, \phi_h) \xrightarrow{a} (P', \phi_{h'})$, and $P' \in S_{h'}^B$.

\textbf{Proof.}  

- Suppose that $(h, B) \xrightarrow{\tau} D$. Looking at the definition of $N_{DC}$, we see that the only possible case for a $\tau$ action when the rule applied is: $(s,l,B) \xrightarrow{\tau} \frac{1}{2} \cdot \delta_{(s \cup \{ e \mid l \} ,l,B)} + \frac{1}{2} \cdot \delta_{(s \cup \{ e \mid l \} ,l,B)}$ when $e \notin s$, and $h = (s,l)$. So we take $h_1 = (s \cup \{ e \mid l \} ,l,B)$, and $h_2 = (s \cup \{ e \mid l \} ,l,B)$. Then we can conclude using Lemma 25 (and the fact that we have again $(S \triangleleft T) \Pi h_i$ for $i \in \{0,1\}$).

- Suppose that $(h, B) \xrightarrow{a} (h', B)$, where $a$ is an action testing frame equivalence. Then we see that $h' = h$. By design of $N_{DC}$, we can see that those actions match the ones in $N'$, and thus $(P, \phi_h) \xrightarrow{a} (P, \phi_{h'})$ (observe that this action depends only on $\phi_h$, thus no guarantee on $P$ are necessary). Since by hypothesis $(h, B)R(P, \phi_h)$, we can conclude;

- Suppose that $(h, B) \xrightarrow{a} (h', B)$, and $a$ is not a testing frame action. By looking at how we defined $N_{DC}$, it means that the rule used was (with $h = (s,l)$): $(h, B) \xrightarrow{\text{out}(b,a)} \delta_{(s,l;\{ a \rightarrow b \} ,l,B)}$ and moreover the following conditions hold: $a \notin h$, $\exists b_l, b_r \in \{0,1\}, e_l(a)[\leftarrow b_l] \in s \land e_r(a)[\leftarrow b_r] \in s$ and $b = b_r \oplus b_l \oplus B(a)$. We note $f : \mathcal{E}' \rightarrow \{0,1\}$ and $g : \mathcal{A}' \rightarrow \{0,1\}$ for the two partial valuations functions associated to $h$. By definition of $R$, we know that $P$ is of the form $P(B, f, S \triangleleft T)$
Lemma 27. Let \((h, B) \in \mathcal{H} \times \mathcal{W}_1\). Suppose that \(P \in S^B_h\).

1. if \((P, \phi_h) \xrightarrow{\tau} \text{det}(P', \phi')\), then \(\phi' = \phi_{h'}\) and \(P' \in S^B_{h'}\);

2. if \((P, \phi_h) \xrightarrow{a} D\), and \(D\) is not a dirac distribution. Then there exists \(h_1, h_2, P_1, P_2\) with \(P_1 \in S^B_{h_1}, P_2 \in S^B_{h_2}\) with \(D = \frac{1}{2} \cdot \delta_{(P_1, \phi_{h_1})} + \frac{1}{2} \cdot \delta_{(P_2, \phi_{h_2})}\), and moreover \((h, B) \xrightarrow{\frac{1}{2} \cdot \delta_{(h_1, B)} + \frac{1}{2} \cdot \delta_{(h_2, B)}}\);

3. \((P, \phi_h) \xrightarrow{a} D\), then there exists \(P', h'\) with \(D = \delta_{(P', \phi_{h'})}\), \((h, B) \xrightarrow{a} (h', B)\), and \(P' \in S^B_{h'}\).

Proof. Let be \(P \in S^B_h\). Recall that a \(h \in \mathcal{H}\) is characterised by the data of \((\text{dom}_E(h), v_E(h), \text{dom}_A(h), \leq_h, v_A(h))\), where all these objects are as in Notation 18. \(P \in S^h_h\) means that there exists a valid interaction pair \(S \prec T\), and such that \((S \prec T, \Pi h)\), and \((\forall a \in \text{dom}_A(h), v_A(h)(a) = B(a) \oplus v_E(h)(e.l(a)) \oplus v_E(h)(e.r(a)))\), and moreover \(P = P(B, v_E(h), S \prec T)\).

1. We look at all the possible deterministic \(\tau\) actions from \((P(B, v_E(h), S \prec T), \phi_h)\): intuitively, those are the \(\tau\)-steps that correspond to a communication between a participant and an oracle. Observe that when an oracle process throws its dice, i.e. does a probabilistic choice, it is not a deterministic \(\tau\)-step. From there, we can see that all these deterministic \(\tau\)-step reach a state of the form \((P(B, v_E(h), S' \prec T'), \phi_h)\), where \(S' \prec T'\) is a valid interaction pair compatible with the history \(h\). From there, we see that \(P(B, v_E(h), S' \prec T') \in S^B_h\).

2. Suppose now that \((P(B, v_E(h), S \prec T), \phi_h) \xrightarrow{\tau} D\), and \(D\) is not a dirac distribution. Looking at the definition of \(P(B, v_E(h), S \prec T)\), we see that the only possible \(\tau\)-step here is when some oracle process \(e \not\in \text{dom}_E(h)\) does a probabilistic choice—i.e. throws its dice. It means that neither the frame, nor the history of internal communication \(S \prec T\) is modified: the only difference is that the partial valuation function is more defined. So we end up in the distribution over processes: \(D = \).
Proof. It is a direct consequence of Lemmas 26 and 27.

Let $N$.

9.4 Security of the dining cryptographers protocol

Proposition 14. The relation $R$ of Definition 12 is a bisimulation on $N_{DC} \sqcup N^f$.

Proof. It is a direct consequence of Lemmas 26 and 27.

9.4 Security of the dining cryptographers protocol

Proposition 13 allows us to study the simple system $N_{DC}$ insted of working directly with $N^f$.

Proposition 15. Let $B, B' \in W_1$. Then the $N_{DC}$-states $(\epsilon_H, B)$ and $(\epsilon_H, B')$ are bisimilar.
Proof. We fix $B, B' \in \mathcal{W}_1$. We build a relation $R$ on $\mathbb{N}_{DC}$-states that contains the pair $((\epsilon_H, B), (\epsilon_H, B'))$, and we show that it is a bisimulation. We use again the history characterisation of Notation 18 which tells us that $h = (s_h, l_h) \in H$ is uniquely defined by giving $l_h$, and a set $\mathcal{F}_h \subseteq \mathcal{E}$ equipped with a function $f_h : \mathcal{F} \to \{0, 1\}$.

Our construction is as follows: the states $(h, B)$ and $(h', B')$ are connected by $R$ when all the three following conditions hold:

1. $l_h = l_{h'}$, $\mathcal{F}_h = \mathcal{F}_{h'}$ (we will simply note them $\mathcal{F}$ below);
2. $\forall e_1, e_2 \in \mathcal{F}_h, e_1 \neq e_2$, it holds that $B(a) \oplus f(e_1) \oplus f(e_2) = B'(a) \oplus f'(e_1) \oplus f'(e_2)$, where $a = a(e_1, e_2)$ as in Notation 13;
3. $\forall e_1, e_2 \in \mathcal{F}_h, e_1 \neq e_2, (f(e_1) \oplus B(a_1)) \oplus (f(e_2) \oplus B(a_2)) = (f'(e_1) \oplus B'(a_1)) \oplus (f'(e_2) \oplus B'(a_2))$, where $a_1 = a(e_1 \setminus e_2)$ and $a_2 = a(e_2 \setminus e_1)$ as defined in Notation 13.

First, observe that $(\epsilon_H, B) R (\epsilon_H, B')$. Thus to show Proposition 15, it is enough to show that $R$ is a bisimulation. Let $(t, t') \in R$, and $\alpha \in \mathcal{A}_{DC}$ such that $t \xrightarrow{\alpha} D$. We need to show that $t' \xrightarrow{\alpha} D'$, and $D(\tilde{R})' D'$. First, observe that since $(t, t') \in R$, we know that there exists $s, s'$ set of internal events such that $t = ((s, l), B)$ and $s' = ((s', l), B')$. Moreover, we can see that $s, s'$ have the same underlying subset of edges, i.e. $s, s'$ are characterised by $f_s, f_{s'} : \mathcal{F} \subseteq \mathcal{E} \to \{0, 1\}$.

We do a case disjunction on the form of $\alpha$; the only relevant cases are the ones mentioned in Definition 10.

- if $\alpha$ is of the form $(\xi \supseteq \zeta)$, or $(\xi \nsubseteq \zeta)$ we have immediately the result, because $\phi_h = \phi_{h'} = \phi_t$;
- if $\alpha = \text{out}(b, a)$, then $D = \delta_{(s,t::(a[\rightarrow b]), B)}$, and $\exists b_l, b_r \in \{0, 1\}, e_l(a)[l \leftarrow b_l] \in s \wedge e_r(a)[l \leftarrow b_r] \in s$ and $b = b_r \oplus b_l \oplus B(a)$. Actually, with our history characterisation from Notation 18, we have $b_l = f_s(e_l(a)), b_r = f_s(e_r(a))$. By condition 1, $\exists b_l', b_r' \in \{0, 1\}, e_l(a)[l \leftarrow b_l'] \in s' \wedge e_r(a)[l \leftarrow b_r'] \in s'$, and actually $b_l' = f_{s'}(e_l(a))$ and $b_r' = e_r(a)$. By condition 2, we also have $b = b_r' \oplus b_l' \oplus B'(a)$. From there, we obtain that $t' \xrightarrow{\alpha} \delta_{(s', t::(a[\rightarrow b]), B')}$. To conclude, we need to show that $(s, l :: (a[\rightarrow b]), B) R (s', l :: (a[\rightarrow b]), B')$, i.e. that the requirements (1), (2) and (3) hold. We can do this easily in that case: on the one hand, the internal part of the history hasn’t been changed by the action $\alpha$. On the other hand, the external parts of the histories coincide for these two states: it is $l :: (a[\rightarrow b])$, so all the requirements hold.

- if $\alpha = \tau$, then $\exists e \notin \mathcal{F}$ such that $D = \frac{1}{2} \cdot \delta_{(s \cup (e[\leftarrow 0], l), B)} + \frac{1}{2} \cdot \delta_{(s \cup (e[\leftarrow 1], l), B)}$. In that case, we see that we can replicate this $\tau$ action on $t'$, in the sense that (since $e \notin \mathcal{F}$) $t' \xrightarrow{\tau} D' := \frac{1}{2} \cdot \delta_{(s' \cup (e[\leftarrow 0], l), B')} + \frac{1}{2} \cdot \delta_{(s' \cup (e[\leftarrow 1], l), B')}$. For $b \in \{0, 1\}$, we note $u(b) = (s \cup (e[\leftarrow b], l), B)$ and $u'(b) = (s' \cup (e[\leftarrow b], l), B')$. Observe that in order to show that $D(\tilde{R})D'$, it is enough to show either $(u(0)Ru'(0) \wedge u(1)Ru'(1))$ or $(u(0)Ru'(0) \wedge u(1)Ru'(1))$. First, we can immediately check that Condition 1 holds for all these pair of states. So it is enough to look at Conditions 2 and 3. Here, we do a case disjunction on the cardinality of $\mathcal{F}$—that here, can be 0, 1 or 2:
– if \( \text{card}(\mathcal{F}) = 0 \) (i.e. \( s = s' = \emptyset \)), then we can show that \((u(0)Ru'(0) \land u(1)Ru'(1))\). We do the proof that \(u(0)Ru'(0)\) (the proof of \(u(1)Ru'(1)\) is similar. Since \( \text{card}(\mathcal{F}) = 0 \), the cardinal of the internal part of the history in the states \(u(0)\) and \(u'(0)\) is 1: that means that the requirements (2) and (3) are empty, and we can conclude.

– if \( \text{card}(\mathcal{F}) = 1 \), we note \( \mathcal{F} = \{e_0\} \). We note \( a \) for the (uniquely defined) participant that belong to both \( e \) and \( e_0 \). We now do a case disjunction on the boolean \( b_0 := (B(a) \oplus f(e_0)) \oplus (B'(a) \oplus f'(e_0)) \). If \( b_0 = 0 \), we are going to show that \((u(0)Ru'(0) \land u(1)Ru'(1))\), and if \( b_0 = 1 \), we will show that \((u(0)Ru'(1) \land u(1)Ru'(0))\). We can deal with these two cases at the same time, by stating our new goal as:

\[
\forall b \in \{0, 1\}, \ u(b) \ R u'(b \oplus b_0).
\]

We fix \( b \in \{0, 1\} \). It means that we need to show Conditions (2) and (3) for \( u(b) \) and \( u'(b \oplus b_0) \):

**Cond (2):** When we rewrite (2) in that case, we see that it boils down to: \( B(a) \oplus f(e_0) \oplus b = B'(a) \oplus f'(e_0) \oplus (b \oplus b_0) \). We can write this equality equivalently as:

\[
(B(a) \oplus f(e_0) \oplus b) \oplus (B'(a) \oplus f'(e_0) \oplus (b \oplus b_0)) = 0.
\]

Now, we are able to conclude using commutativity and associativity of \( \oplus \) and the definition of \( b_0 \).

**Cond (3):** When we rewrite (3) in that case, we see that it boils down to: \( f(e_0) \oplus B(a_{e_0}) \oplus b \oplus B(a_e) = f'(e_0) \oplus B'(a_{e_0}) \oplus (b \oplus b_0) \oplus B'(a_e) \), with \( a_{e_0} = a(e_0 \setminus e) \) and \( a_e = a(e \setminus e_0) \). Recall that both \( B \) and \( B' \) are 1-weighted, so we have: \( B(a) \oplus B(a_{e_0}) \oplus B(a_e) = 1 = B'(a) \oplus B'(a_{e_0}) \oplus B'(a_e) \). From there, we can conclude as above.

– if \( \text{card}(\mathcal{F}) = 2 \), then we note \( \mathcal{F} = \{e_1, e_2\} \), and \( a = a(e_1 \& e_2) \). Observe that since \( tRt' \), we know that \( f(e_1) \oplus B(a_1) \oplus f(e_2) \oplus B(a_2) = f'(e_1) \oplus B'(a_1) \oplus f'(e_2) \oplus B'(a_2) \), with \( a_1 = a(e_1 \setminus e_2) \), and \( a_2 = a(e_2 \setminus e_1) \). Observe that this equality can be rewritten equivalently as: \( f(e_1) \oplus B(a_1) \oplus f(e_2) \oplus B(a_2) \oplus f'(e_1) \oplus B'(a_1) \oplus f'(e_2) \oplus B'(a_2) = 0 \), and thus also as \( f(e_1) \oplus B(a_1) \oplus f'(e_1) \oplus B'(a_1) = f(e_2) \oplus B(a_2) \oplus f'(e_2) \oplus B'(a_2) := b_0 \). We note \( b_0 \) for the resulting boolean. We are going to show:

\[
\forall b \in \{0, 1\}, \ u(b) \ R u'(b \oplus b_0).
\]

Observe that it is enough to check Conditions (2) and (3) on the pair of edges \( \{e_1, e\} \) and \( \{e_2, e\} \). (We already know that it works for \( \{e_1, e_2\} \), since \( tRt' \).) We do the proof for \( \{e_1, e\} \) (it is similar for the other pair).

**Cond (2):** We need to show: \( B(\tilde{a}) \oplus f(e_1) \oplus b = B'(\tilde{a}) \oplus f'(e_1) \oplus (b \oplus b_0) \), where \( \tilde{a} = a(e_1 \& e) = a(e_1 \setminus e_2) = a_1 \). Thus we can rewrite our goal as: \( B(a_1) \oplus f(e_1) \oplus b \oplus B'(a_1) \oplus f'(e_1) \oplus (b \oplus b_0) = 0 \). From there, we can conclude using the fact that \( b_0 = f(e_1) \oplus B(a_1) \oplus f'(e_1) \oplus B'(a_1) \).
two functions such that \( \text{card}(\text{schedulers}) \), overlooked in previous attempts. In addition to solving technical problems, such a framework turns out to rely on subtle technicalities such as the need for randomized modelled as process equivalences, in symbolic models enhanced with probabilities. Defining

In this paper we introduced a framework to reason about indistinguishability properties, we also have \( \text{card}(\text{schedulers}) \) and

\[ f(e_1) \oplus B(a_2) \oplus f'(e_2) \oplus f'(e_2) = 0. \]

We can now simplify the equation above as:

\[ B(a_2) \oplus b_0 \oplus B'(a_2) \oplus f(e_2) \oplus f'(e_2) = 0, \]

and finally we can conclude since by definition of \( b_0 \) we have: \( b_0 = f(e_2) \oplus B(a_2) \oplus f'(e_2) \oplus B'(a_2). \)

\[ \square \]

**Corollary 3** (Security for the dining cryptographers protocol). Let \( B, B' : A \rightarrow \{0, 1\} \) two functions such that \( \text{card}(B^{-1}(\{1\})) = \text{card}(B'^{-1}(\{1\})) = 1 \). Then the states \( s_B := (DC_B, \phi) \) and \( s_{B'} := (DC'_B, \phi) \), with \( \phi_0 := \{a_1 = a_1, a_2 = a_2, a_3 = a_3\} \) are bisimilar in \( N^f \).

**Proof.** Since we know by Proposition \[ 15 \] that \( t_B := ((\emptyset, \epsilon), B) \) and \( t_{B'} := ((\emptyset, \epsilon), B') \) are bisimilar in \( N_{DC} \), we also have \( (t_B, t_{B'}) \in \sim_{N_{DC} \cup N^f} \) (because \( N^f \) and \( N_{DC} \) have disjoint state spaces). By proposition\[ 13 \] we also have \( t_B \sim_{N_{DC} \cup N^f} s_B, \) and \( t_{B'} \sim_{N_{DC} \cup N^f} s_{B'}. \) Since \( \sim_{N_{DC} \cup N^f} \) is an equivalence relation, we can combine all this to obtain that \( s_B \sim_{N_{DC} \cup N^f} s_{B'}. \) From here (again because \( N^f \) and \( N_{DC} \) have disjoint state spaces), we can conclude that \( s_B \sim_{N^f, s_{B'}}. \)

\[ \square \]

10 Conclusion and future work

11 Conclusion and future work

In this paper we introduced a framework to reason about indistinguishability properties, modelled as process equivalences, in symbolic models enhanced with probabilities. Defining such a framework turns out to rely on subtle technicalities such as the need for randomized schedulers, overlooked in previous attempts. In addition to solving technical problems,
we believe that randomized schedulers capture more faithfully the idea that one cannot predict how non-determinism is resolved. Randomized schedulers generalize the idea that a schedulers chooses a particular distribution among the ones available by allowing an arbitrary combination (in the convex hull) of the available distributions.

We define different, classical behavioral and labelled equivalences and show their precise relations. As usual in models mixing non-determinism and probabilities, the resulting equivalences may be considered as too strong; indeed arbitrary schedulers may leak the (private) probabilistic choices of the processes and give the attacker an unrealistically strong distinguishing power. Defining more restricted schedulers that are only allowed partial knowledge of the current state, such as in [CP10], is orthogonal to our work. We however believe that our work provides a convenient framework for defining such more fine-grained notions of schedulers and consider this an interesting direction for future work.

We therefore study two classes of protocols where this problem is avoided. First, we study protocols that do not make probabilistic choices, but allow the adversary to do so. This class of non-probabilistic protocols corresponds to the classical setting and captures all major case studies performed in the context of symbolic models. Our results highlight that the classical notion of may-testing, considered rather intuitive as it models an arbitrary attacker running in parallel, does not take into account attackers that make probabilistic choices. Interestingly, when bounding the number of sessions, (non-probabilistic) similarity exactly captures such probabilistic attackers and offers an attractive target for automated analysis. Second, we study a class of fully probabilistic protocols, also considered in [CSV17], and show that trace equivalence on such protocols coincides with may testing in the presence of a (syntactic) class of determinate attackers. One may indeed argue that determinacy removes artificial non-deterministic choices that the attacker could exploit and that correspond to unrealistic behaviors. When protocols can be expressed in the class of purely probabilistic processes, from a formal analysis point, it seems appealing to do so as it also simplifies the analysis.

We also show how deciding trace equivalence can be automated in the presence of probabilities. We propose a decision procedure that extends previous work by Cheval el al [CKR18] and implement this procedure in the DeepSec tool. Hence, we provide tool support for proving determinate may-testing on the class of purely probabilistic processes when the number of sessions is bounded and cryptographic primitives are modeled as a subterm destructor rewrite system. On more general classes of processes, our tool can be used for attack finding: disproving trace equivalence implies that may testing (and all stronger equivalences) does not hold either, therefore violating security properties stated as an equivalence.

Finally, we illustrate our framework by studying the well-known Dining Cryptographers protocol. We model the protocol and its anonymity property in the probabilistic applied π-calculus. Then, we use our framework to prove that anonymity holds, and demonstrate DeepSec’s attack finding ability on a variant of the protocol that uses a biased coin.

Our work paves the road towards several future works, in addition to exploring restricted schedulers mentioned above. The insight that (purely possibilistic) similarity takes into account probabilistic adversaries (as it coincides with may testing) when the number
of sessions is bounded and protocols are non-deterministic motivates adding support for (bi)similarity in a tool such as DeepSec (which currently only verifies trace equivalence). A different direction going beyond the subclasses considered in this paper is to investigate restrictions of the scheduler (building, e.g., on ideas from [CP10, AAPvR10]) in our framework to limit the adversary’s power without restricting the class of protocols. Finally, a more prospective direction is the use of more quantitative equivalences, i.e., distances between processes, that might be interesting to compare different protocols that try to achieve a same property.

**Acknowledgements.** We thank Alwen Tiu and the anonymous reviewers for their helpful comments and suggestions.

**References**


[BDH+18] David A. Basin, Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, Ralf Sasse, and Vincent Stettler. A formal analysis of 5g authentication. In David Lie,


A Randomized and non-randomized resolutions coincide for may-testing

Lemma 28 (One-step derandomization). Let $N = (S_N, A, \mathsf{trans}_N)$ be a NPLTS, such that $\mathsf{trans}_N(s, \tau) = D$ implies that $D$ has finite support. Let $R = (S_R, \mathsf{corr}_R, \mathsf{trans}_R) \in \mathcal{R}_r(N)$, and $s \in S_R$ an internal state. Let $(\tau, D) \in A_{\text{int}} \times \mathcal{D}(S_R)$ the R-transition step from $s$, i.e. $\mathsf{trans}_R(s) = (\tau, D)$. We define the set of possible de-randomizations of $\beta = (s, \tau, D)$ as.

$$\Delta(\beta) = \{ F \in \mathcal{D}(S_R) \mid (\tau, \mathsf{corr}_R(F)) \in \mathsf{trans}_N(\mathsf{corr}_R(s)), \quad \mathsf{corr}_R \text{ injective on } \supp(F), \supp(F) \subseteq \supp(D) \}$$

Then the randomized transition step $\beta$ can be expressed as the convex sum of de-randomized step, or more formally:

$$\exists \eta \in \mathcal{D}(\Delta(\beta)) \text{ s.t. } \sum_{F \in \Delta(\beta)} \eta(F) \cdot F = D.$$ 

Proof. By hypothesis, $R \in \mathcal{R}_r(N)$, thus there exists a distribution $\gamma$ over $\mathsf{trans}_N(\mathsf{corr}_R(s))(\tau)$ such that $\mathsf{corr}_R(D) = \sum_{E \in \mathsf{trans}_N(\mathsf{corr}_R(s))(\tau)} \gamma(E) \cdot E$. Recall that, for any $F \in \Delta(\beta)$, since $\mathsf{corr}_R$ is injective on the support of $F$, we can define $(\mathsf{corr}_R)^{-1}(t) \in S_R$, for any $t \in S_N$. We define, for $F \in \Delta(\beta) \subseteq \mathcal{D}(S_R)$:

$$\eta(F) = \gamma(\mathsf{corr}_R(F)) \cdot \prod_{s \in \supp(\mathsf{corr}_R(F))} \frac{D((\mathsf{corr}_R)^{-1}(s))}{\mathsf{corr}_R(D)(s)}.$$ 

Notice that $\supp(F) \subseteq \supp(D)$ implies $\supp(\mathsf{corr}_R(F)) \subseteq \supp(\mathsf{corr}_R(D))$, and so for all $s \in \supp(\mathsf{corr}_R(F))$, $\mathsf{corr}_R(D)(s) \neq 0$ which ensures that $\eta(D)$ is correctly defined.

- First, we check that $\eta$ is indeed a distribution.

$$\sum_{F \in \Delta(\beta)} \eta(F) = \sum_{F \in \Delta(\beta)} \gamma(\mathsf{corr}_R(F)) \cdot \prod_{s \in \supp(\mathsf{corr}_R(F))} \frac{D((\mathsf{corr}_R)^{-1}(s))}{\mathsf{corr}_R(D)(s)}$$

$$= \sum_{E \in \mathsf{trans}_N(\mathsf{corr}_R(s))(\tau)} \gamma(E) \cdot \sum_{F \in \Delta(\beta)} \prod_{E \in \mathsf{trans}_N(\mathsf{corr}_R(s))(\tau)} \frac{D((\mathsf{corr}_R)^{-1}(s))}{\mathsf{corr}_R(D)(s)}.$$ 

Recall that $\gamma$ is a distribution over $\mathsf{trans}_N(\mathsf{corr}_R(s))(\tau)$. Hence, for all $E \in \mathsf{trans}_N(\mathsf{corr}_R(s))(\tau)$, $\gamma(E) \neq 0$ if and only if $E \in \supp(\gamma)$. Hence:

$$\sum_{F \in \Delta(\beta)} \eta(F) = \sum_{E \in \mathsf{trans}_N(\mathsf{corr}_R(s))(\tau) \cap \supp(\gamma)} \gamma(E) \cdot \sum_{F \in \Delta(\beta)} \prod_{E \in \mathsf{trans}_N(\mathsf{corr}_R(s))(\tau) \cap \supp(\gamma)} \frac{D((\mathsf{corr}_R)^{-1}(s))}{\mathsf{corr}_R(D)(s)}.$$
Observe that, given some $E \in \text{trans}_N(s)(\tau) \cap \text{supp}(\gamma)$, choosing some $F \in \Delta(\beta)$ such that $\text{corr}_R(F) = E$ is equivalent to choosing, for every $s \in \text{supp}(E)$, a unique $\tilde{s} \in S_R$ such that $\text{corr}_R(\tilde{s}) = s$, and then to define $F = \sum_{s \in \text{supp}(E)} E(s) \cdot \delta_{\tilde{s}}$. Notice that by construction, we have $\text{corr}_R$ injective on $\text{supp}(F)$. Moreover, for all $v \in \text{supp}(F)$, $\text{corr}_R(v) \in \text{supp}(E)$. Since $E \in \text{supp}(\gamma)$ and $\text{corr}_R(D) = \sum_{E \in \text{trans}_N(\text{corr}_R(s))} \gamma(E) \cdot E$, we deduce that $\text{corr}_R(v) \in \text{supp}(\text{corr}_R(D))$ and so $v \in \text{supp}(D)$. This ensures that $F \in \Delta(\beta)$.

As a consequence, for $E \in \text{trans}_N(s)(\tau) \cap \text{supp}(\gamma)$:

$$\sum_{F \in \Delta(\beta)} \prod_{s \in \text{supp}(\text{corr}_R(F))} \frac{D((\text{corr}_R)^{-1}(\text{supp}(F))(s))}{\text{corr}_R(D)(s)} = \sum_{\phi : \text{supp}(E) \rightarrow S_R} \prod_{s \in \text{supp}(E)} \frac{D(\phi(s))}{\text{corr}_R(D)(s)}$$

(3)

Recall that for all $E \in \text{supp}(\gamma)$, for all $s \in \text{supp}(E)$, $\text{corr}_R(D)(s) \neq 0$ and $\text{corr}_R(D)(s) = \sum_{u \in \text{corr}_R^{-1}(\{s\})} D(u)$, and thus $u \in \text{corr}_R^{-1}(\{s\}) \mapsto \frac{D(u)}{\text{corr}_R(D)(s)} \in [0, 1]$ is a distribution. From there, we can consider the product distribution for any (finite) set $S_I = \{s_i \mid i \in I \land \text{corr}_R(D)(s_i) \neq 0\} \subseteq S_N$ given as: $(u_i)_{i \in I} \in \prod_{i \in I} \text{corr}_R^{-1}(\{s_i\}) \mapsto \prod_{i \in I} \frac{D(u_i)}{\text{corr}_R(D)(s_i)}$.

In particular, since this product distribution is indeed a distribution, it holds that:

$$\sum_{\phi : S_I \rightarrow S_R} \prod_{s \in S_I} \frac{D(\phi(s))}{\text{corr}_R(D)(s)} = 1.$$  

(4)

By taking $S_I = \text{supp}(E)$, [1] tells us that: $\sum_{\phi : \text{supp}(E) \rightarrow S_R} \prod_{s \in \text{supp}(E)} \frac{D(\phi(s))}{\text{corr}_R(D)(s)} = 1$, and we can conclude from there.

• we want now to show that $\sum_{F \in \Delta(\beta)} \eta(F) \cdot F = D$. Let $t \in S_R$. By unfolding the definition of $\eta(F)$ and using the same reasoning as in the previous bullet point, we obtain:

$$\sum_{F \in \Delta(\beta)} \eta(F) \cdot F(t) = \sum_{F \in \Delta(\beta)} \gamma(\text{corr}_R(F)) \cdot \prod_{s \in \text{supp}(\text{corr}_R(F))} \frac{D((\text{corr}_R)^{-1}(\text{supp}(F))(s))}{\text{corr}_R(D)(s)} \cdot F(t)$$

$$= \sum_{E \in \text{trans}_N(s)(\tau)} \gamma(E) \cdot \sum_{\phi : \text{supp}(E) \rightarrow S_R} \prod_{s \in \text{supp}(E)} \frac{D(\phi(s))}{\text{corr}_R(D)(s)} \cdot E(\phi^{-1}(t))$$

The last equality is obtained using the same reasoning as in the previous bullet point. Additionally, we restrict the function $\phi$ we consider by only taking the one where $t \in \text{Im}(\phi)$ (as $t \not\in \text{Im}(\phi)$ would imply $F(t) = 0$). Also notice that $F(t) = E(\phi^{-1}(t))$. 81
Finally, since \( \phi^{-1}(t) = \text{corr}_R(t) \), we obtain the following:

\[
\sum_{F \in \Delta(n)} \eta(F) \cdot F(t) = \sum_{E \in \text{trans}_N(s)(\tau)} \gamma(E) \cdot \sum_{\phi: \text{supp}(E) \rightarrow S_R \atop \forall u, \text{corr}_R(\phi(u)) = u, \phi(\text{corr}_R(t)) = t} \prod_{\phi: \text{supp}(E) \setminus \{\text{corr}_R(t)\} \rightarrow S_R \setminus \{\text{corr}_R(t)\}} \frac{D(\phi(s))}{\text{corr}_R(D)(s)} \cdot E(\text{corr}_R(t))
\]

\[
= \sum_{E \in \text{trans}_N(s)(\tau)} \gamma(E) \cdot D(t) \cdot \frac{E(\text{corr}_R(t))}{\text{corr}_R(D)(\text{corr}_R(t))} \text{ by hypothesis on } \gamma
\]

\[
= D(t).
\]

\( \square \)

**Lemma 29** (finite steps derandomization). Let \( N \) be a NPLTS defined by its transition function \( \text{trans}_N : S_N \rightarrow A_{\text{int}} \sqcup A_{\text{ext}} \rightarrow \mathcal{P}(D(S_N)) \). Moreover, we ask \( N \) to be such that \( \text{trans}_N(s, \tau) = D \) implies that \( D \) has finite support. For every set \( T \subseteq S_N \), for every randomized resolution \( R = (R_N, \text{corr}_R, \text{trans}_R) \in \mathcal{R}_r(N) \), for every state \( s \in S_N \), it holds that for every \( n \in \mathbb{N} \):

\[
\text{RProb}^n_R(s, \text{corr}_R^{-1}(T)) \in \text{conv}\{\text{RProb}_R(s', \text{corr}_R^{-1}(T)) \mid R' = (R_N, \text{corr}_{R'}, \text{trans}_{R'}) \in \mathcal{R}_r(N), \text{corr}_R(s) = \text{corr}_R(s')\}
\]

**Proof of Lemma 29**. Let \( T \subseteq S_N \), and \( R = (S_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}_r(N) \). We do the proof by induction on \( n \):

- Base case \( n = 0 \): We define \( R_0 := (S_N, \text{id}, \text{trans}_{R_0}) \), where \( \text{id} \) stands for the identity function, and all states in \( S_N \) are external states such that \( \text{trans}_{R_0}(u)(a) = * \) for every \( u \in S_N, a \in A_{\text{ext}} \). We can see immediately that \( R_0 \in \mathcal{R}_r(N) \). Observe that \( \text{RProb}_{R_0}(u, T) = 1 \) when \( u \in T \), and 0 otherwise. As a consequence, for \( s \in S_R \), \( \text{RProb}_{R_0}^n(s, \text{corr}_R^{-1}(T)) = \text{RProb}_{R_0}(s, T) \).

- Inductive step: We suppose that the result holds for \( n \geq 0 \). Let \( s \in S_R \). We do a case disjunction on the first \( R \)-transition from \( s \):
if \( s \) is an external state or \( s \in \corr_R^{-1}(T) \) then \( \text{RProb}_R^{\leq n}(s, \corr_R^{-1}(T)) = \text{RProb}_R^{\leq 0}(s, \corr_R^{-1}(T)) \) by definition. By applying the same reasoning as in the base case, the result holds;

otherwise there exists a distribution \( E \in \mathcal{D}(\mathcal{S}_R) \) such that \( \text{trans}_R(s) = (\tau, E) \), with \( \tau \) an internal action, and \( \text{corr}_R(E) \in \text{conv}(\text{trans}_R(\text{corr}_R(s))(\tau)) \). For each \( v \in \mathcal{S}_N \), we define the set of resolutions starting from \( v \) as: \( \mathcal{R}_nr(N, v) = \{(R', w) \mid R' = (\mathcal{S}_R', \text{corr}_R', \text{trans}_R') \in \mathcal{R}_nr(N), w \in \mathcal{S}_R', \text{corr}_R'(w) = v\} \). By applying the induction hypothesis to the states in the support of \( E \), we obtain that for all \( t \in \text{supp}(E) \) there exists a distribution \( \Theta_t \in \mathcal{D}(\mathcal{R}_nr(N, \text{corr}_R(t))) \) such that:

\[
\text{RProb}_R^{\leq n}(t, \text{corr}_R^{-1}(T)) = \sum_{(R', u) \in \mathcal{R}_nr(N, \text{corr}_R(t))} \Theta_t(R', u) \cdot \text{RProb}_{R'}(u, \text{corr}_{R'}^{-1}(T)).
\]

(5)

Observe that choosing, for every \( t \in \text{supp}(E) \), one non-randomized resolution starting from \( \text{corr}(t) \), i.e a \( (R_t, u_t) \in \mathcal{R}_nr(N, \text{corr}_R(t)) \), means choosing a function \( f : t \in \text{supp}(E) \mapsto (R_t, u_t) \in \mathcal{R}_nr(N, \text{corr}_R(t)) \). We write \( \mathcal{F} \) for the set of all those functions. Using the distributions \( \Theta_t \), we can built the product distribution, that compute the probability of choosing some \( f \in \mathcal{F} \), when choosing each \( f(t) \) with the probabilities specified by the distribution \( \Theta_t \). This way, we obtain \( \Theta \in \mathcal{D}(\mathcal{F}) \) defined as \( \Theta(f) = \prod_{t \in \text{supp}(E)} \Theta_t(f(t)) \). Observe that, for any \( t \in \text{supp}(E) \), and \( (R', u) \in \mathcal{R}_nr(N, \text{corr}_R(t)) \), it holds that:

\[
\Theta_t(R', u) = \sum_{f : f(t) = (R', u)} \Theta(f).
\]

(6)

Moreover, by defining \( \beta = (s, \tau, E) \) and using Lemma 28, we obtain the existence of a distribution \( \eta \in \mathcal{D}(\Delta(\beta)) \), with \( \Delta \) defined in Lemma 28, such that:

\[
\sum_{D \in \Delta(\beta)} \eta(D) \cdot D = E.
\]

(7)

For every element \( D \in \Delta(\beta) \), and any function \( f : t \in \text{supp}(E) \mapsto (R_t, u_t) \in \mathcal{R}_nr(N, \text{corr}_R(t)) \), we build a non-randomized resolution \( \text{R}_{D,f} \) as:

* \( \mathcal{S}_{R_{D,f}} = \{s\} \cup \{\mathcal{S}_{R_t} \mid t \in \text{supp}(E)\}, f(t) = (R_t, u_t)\};
* the correspondence function \( \text{corr}_{R_{D,f}} \) is defined as: \( \text{corr}_{R_{D,f}}(u) = \text{corr}_{R_t} \) if \( z \in \mathcal{S}_{R_t}, \) and \( \text{corr}_{R_{D,f}}(s) = \text{corr}(s) \);
* the transition function \( \text{trans}_{R_{D,f}} \) is defined as: if \( u \in \mathcal{S}_{R_t} \), then \( \text{trans}_{R_{D,f}}(u) = \text{trans}_{R_t}(u) \), and for all \( u \in \mathcal{S}_{R_{D,f}} \), \( \text{trans}_{R_{D,f}}(s)(u) = D(t) \) when \( \exists t \in \text{supp}(E) \) such that \( f(t) = (R_t, u) \) for some \( R_t \); and 0 otherwise.

We can easily check that \( \text{R}_{D,f} \in \mathcal{R}_nr(N) \): to do so, we need to use the fact that, by induction hypothesis, all the \( R_t \) given by \( f \) are in \( \mathcal{R}_nr(N) \), and the definition
of $\Delta(\beta)$ given in Lemma $28$. In order to conclude, we are going to show:

$$RProb_R^{\leq n+1}(s, \text{corr}_R^{-1}(T)) = \sum_{D \in \Delta(\beta), f \in F} \eta(D) \cdot \Theta(f) \cdot RProb_{R_D,f}(s, \text{corr}^{-1}_{R_D,f}(T)).$$

$$RProb_R^{\leq n+1}(s, \text{corr}_R^{-1}(T))$$

$$= \sum_{t \in \text{supp}(E)} E(t) \cdot RProb_R^{\leq n}(t, \text{corr}_R^{-1}(T)) \text{ by Definition } 7$$

$$= \sum_{t \in \text{supp}(E)} E(t) \cdot \sum_{(R',u) \in \mathcal{R}_{w}(N, \text{corr}(t))} \Theta_i(R',u) \cdot RProb_{R'}(u, \text{corr}^{-1}_{R'}(T)) \text{ by (5)}$$

$$= \sum_{t \in \text{supp}(E)} (\sum_{D \in \Delta(\beta)} \eta(D) \cdot D(t)) \cdot \sum_{(R',u) \in \mathcal{R}_{w}(N, \text{corr}(t))} \Theta_i(R',u) \cdot RProb_{R'}(u, \text{corr}^{-1}_{R'}(T)) \text{ by (7) and (6)}$$

$$= \sum_{D \in \Delta(\beta)} \sum_{f \in F} \eta(D) \cdot \Theta(f) \cdot D(t) \cdot \sum_{(R',u) \in \mathcal{R}_{w}(N, \text{corr}(t))} RProb_{R'}(u, \text{corr}^{-1}_{R'}(T))$$

$$= \sum_{D \in \Delta(\beta)} \sum_{f \in F} \eta(D) \cdot \Theta(f) \cdot D(t) \cdot RProb_{R'}(u, \text{corr}^{-1}_{R'}(T))$$

By definition of $R_{D,f}$, for all $t \in \text{supp}(E)$ with $f(t) = (R',u)$, we know that $\text{trans}_{R_{D,f}}(s)(u) = D(t)$, and for all other states $u \in \mathcal{S}_{R_{D,f}}$, $\text{trans}_{R_{D,f}}(s)(u) = 0$. Hence:

$$RProb_R^{\leq n+1}(s, \text{corr}_R^{-1}(T))$$

$$= \sum_{D \in \Delta(s,t,E), f \in F} \eta(D) \cdot \Theta(f) \cdot \sum_{u \in \mathcal{S}_{R_{D,f}}} \text{trans}_{R_{D,f}}(s)(u) \cdot RProb_{R_{D,f}}(u, \text{corr}^{-1}_{R_{D,f}}(T))$$

Finally, $RProb_{R_{D,f}}(u, \text{corr}^{-1}_{R_{D,f}}(T)) = \lim_{n \to +\infty} RProb_{R_{D,f}}^{\leq n}(u, \text{corr}^{-1}_{R_{D,f}}(T))$. With $\text{supp}(\text{trans}_{R_{D,f}}(s))$ being finite, we deduce that:

$$\sum_{u \in \mathcal{S}_{R_{D,f}}} \text{trans}_{R_{D,f}}(s)(u) \cdot RProb_{R_{D,f}}(u, \text{corr}^{-1}_{R_{D,f}}(T))$$

$$= \sum_{u \in \mathcal{S}_{R_{D,f}}} \text{trans}_{R_{D,f}}(s)(u) \cdot \lim_{n \to +\infty} RProb_{R_{D,f}}^{\leq n}(u, \text{corr}^{-1}_{R_{D,f}}(T))$$

$$= \lim_{n \to +\infty} \sum_{u \in \mathcal{S}_{R_{D,f}}} \text{trans}_{R_{D,f}}(s)(u) \cdot RProb_{R_{D,f}}^{\leq n}(u, \text{corr}^{-1}_{R_{D,f}}(T))$$

$$= RProb_{R_{D,f}}(s, \text{corr}^{-1}_{R_{D,f}}(T))$$

84
We conclude that:

\[
\text{RProb}_{R}^{\leq n+1}(s, \text{corr}_{R}^{-1}(T)) = \sum_{D \in \Delta(\beta)} f \in F \eta(D) \cdot \Theta(f) \cdot \text{RProb}_{R, f}(s, \text{corr}_{R, f}^{-1}(T))
\]

Lemma 30. For all \( P \in MP \), for all \( c \in N_{\text{pub}} \), \( \text{RProb}_{\text{nr}}(P, \downarrow c) = \text{RProb}_{R}(P, \downarrow c) \).

Proof. Since \( \text{R}_{\text{nr}} \subset \text{R} \), we immediately have that \( \text{RProb}_{\text{nr}}(P, \downarrow c) \leq \text{RProb}_{R}(P, \downarrow c) \).

The reverse inequality follows by Lemma 29 (observe that the NPLTS \( N_{\text{no}} \) is indeed such that \( \text{trans}_{N_{\text{no}}}(s, \tau) = D \) implies that \( D \) has finite support): for any resolution \( R \in \text{R} \), it holds that for every \( n \in N \)

\[
\text{RProb}_{R}^{\leq n}(P, \downarrow c) \leq \text{RProb}_{\text{nr}}(P, \downarrow c)
\]

As a consequence:

\[
\text{RProb}_{R}(P, \downarrow c) = \sup_{n \in N} \text{sup}_{\text{R} \in \text{R} \atop \text{R} \in \text{R}_{\text{nr}}} (\text{RProb}_{R}^{\leq n}(P, \downarrow c)) \leq \text{RProb}_{\text{nr}}(P, \downarrow c).
\]

Proposition 1. The may testing preorder coincides for randomized and non-randomized resolutions:

\[
\leq_{\text{may}}^{\text{R}} = \leq_{\text{may}}^{\text{nr}}
\]

Proof. Direct from Lemma 30.

B Appendix on trace equivalence

B.1 Randomized and non-randomized resolutions coincide for trace equivalence

Lemma 31. Let \( N = (S_{N}, A_{\text{ext}} \uplus A_{\text{int}}, \text{trans}_{N}) \) be a NPLTS, and \( R = (S_{R}, \text{corr}_{R}, \text{trans}_{R}) \in \text{R}(N) \). For \( n \in N \), \( w \in A_{\text{ext}}^{*} \) and \( s \in S_{R} \), we define \( \text{Prob}_{R}^{n}(s, w) \) as:

\[
\text{Prob}_{R}^{n}(s, a.w) = 0
\]

\[
\text{Prob}_{R}^{n}(s, \epsilon) = 1
\]

\[
\text{Prob}_{R}^{n}(s, a.w) = \sum_{t \in S_{R}} \text{RProb}_{R}^{\leq n-1}(s, \{t\}) \cdot \sum_{s' \in \text{supp}(D)} D(s') \cdot \text{Prob}_{R}^{n-1}(s', w)
\]

s.t. \( \exists D \in \text{D}(S_{R}), \text{trans}_{R}(t)(a) = D \)

Then, for any \( u \in S_{N} \), it holds that:

\[
\text{Prob}_{R}(u, w) = \sup_{n \in N} \text{sup}_{\text{R} = (S_{R}, \text{corr}_{R}, \text{trans}_{R}) \in \text{R} \atop s \in S_{R} \text{ s.t. corr}(s) = u} \text{Prob}_{R}^{n}(s, w)
\]
Proof. The proof is by induction on the length of $w$. \hfill \square

Here, we develop the tools to compute the trace success probability as reachability probabilities on a new auxiliary NPLTS: the traces NPLTS $N^{tr}$, that we build from any NPLTS $N$. We first define a particular kind of NPLTS: a probabilistic abstract rewriting system (PARS) is a NPLTS with no external actions, and a unique internal action $\tau$.

**Definition 43.** We say that a NPLTS $N$ is a probabilistic abstract rewriting system when $N$ has exactly one internal action $\tau$, and no external action. When it is the case, we say that a $N$ state $s$ is a normal form when $\text{trans}(s, \tau) = \emptyset$, i.e. no transition starts from $s$.

**Definition 44.** Let $N = (S_N, A_{ext} \sqcup A_{int}, \text{trans}_N)$ be a NPLTS. We define the PARS $N^{tr}$ as follows: its states are $\{(s, w) \mid s \in S_N, w \in A_{ext}^*\} \sqcup \{\text{success}\}$. The transition function $\rightarrow$ is defined as:

- $(s, w) \rightarrow_{N^{tr}} (D, w)$ when $s \xrightarrow{N} D$;
- $(s, aw) \rightarrow_{N^{tr}} (D, w)$ when $s \xrightarrow{a} D$;
- $(s, \epsilon) \rightarrow \delta_{\text{success}}$.

Observe that the normal forms of this PARS are: either the state $\text{success}$, or states of the form $(s, a.w)$, with $s \not\xrightarrow{N} a$ and $s \not\xrightarrow{N} \tau$.

Our goal is to prove that the trace equivalence can be computed directly in the PARS $P^{tr}$, instead of the labelled NPLTS $N'$. As a first step, we build a correspondence between $N$ resolutions and $N^{tr}$ resolutions. This correspondance consists actually of two operators: $C : R_r(N) \rightarrow R_r(N^{tr})$ and $D : R_{nr}(N^{tr}) \rightarrow R_{nr}(N)$. Observe that, for technical reasons, the operator $D$ can only be applied to non-randomized resolutions.

**Definition 45.** We define an operator $C : R_r(N) \rightarrow R_r(N^{tr})$ as follows: let $R = (S_R, \text{corr}_R, \text{trans}_R) \in R_r(N)$. We take $C_R = (S_{CR}, \text{corr}_{CR}, \text{trans}_{CR})$ where:

- $S_{CR} = \{(s, w) \mid s \in S_R, w \in A_{ext}^*\} \sqcup \{\text{success}\}$;
- $\text{corr}_{CR}(s, w) = (\text{corr}_R(s), w)$, and $\text{corr}_{CR}(\text{success}) = \text{success}$;
- $\text{trans}_{CR}(s, w)(\tau) = \begin{cases} \text{trans}_R(s), w) & \text{if } s \text{ is an internal state in } R \\ \delta_{\text{success}} & \text{if } s \text{ external state and } w = \epsilon \\ \text{trans}_R(s)(a), w') & \text{when } w = a.w' \text{ and } \text{trans}_R(s)(a) \text{ is defined} \\ \text{undefined otherwise.} & \end{cases}$

Observe moreover that whenever $R$ is a non-randomized resolution over $N$, $C_R$ is a non-randomized resolution over $N^{tr}$.

**Definition 46.** We define an operator $D : R_{nr}(N^{tr}) \rightarrow R_{nr}(N)$ as follows: let $R = (S_R, \text{corr}_R, \text{trans}_R) \in R_r(N^{tr})$. We take $D_R = (S_R, \text{corr}_R, \text{trans}_{DR})$ where: $\text{trans}_{DR} : S_R \times A_N \rightarrow D(S_R)$ is defined as:
• if \( u \in S_R \) is such that \( \text{trans}_R(u)(\tau) \) is undefined, then \( \text{trans}_R(u)(a) \) is undefined for every \( a \in A_{\text{ext}} \cup \{ \tau \} \);

• suppose now that \( \text{trans}_R(u)(\tau) = D \). Since \( R \) is a non-randomized resolution, we can do the following cases disjunction:

  – either \( u \in S_R \) is such that \( \text{corr}_R(u) = (s, \epsilon) \), and \( \text{corr}_R(D) = \delta_{\text{success}} \). Then we take \( \text{trans}_{DR}(u)(a) \) as undefined for every \( a \in A_{\text{ext}} \cup \{ \tau \} \);

  – or \( u \in S_R \) is such that \( \text{corr}_R(u) = (s, w) \), and \( \text{corr}_R(D) = (E, w) \). Then we take \( \text{trans}_{DR}(u)(\tau) = D \), and \( \text{trans}_{DR}(u)(a) \) as undefined for every \( a \in \{ \tau \} \cup (A_{\text{ext}} \setminus \{ a_0 \}) \).

**Lemma 32.** Let \( N \) a NPLTS, and \( w \in A_{\text{ext}}^* \). Then:

1. For every \( R \in R_r(N) \), and \( s \in S_R \), \( \text{Prob}_R(s, w) \leq R\text{Prob}_{CR}((s, w), \text{corr}^{-1}_R(\text{success})) \).

2. For every \( R \in R_{nr}(N_r) \), and \( s \in S_R \) such that there exists \( u \), \( \text{corr}(s) = (u, w) \), it holds that \( \text{Prob}_{DR}(s, w) \geq R\text{Prob}_{R}(s, \text{corr}^{-1}_R(\text{success})) \).

**Proof of item 1 in Lemma 32.** Let us suppose \( R \in R_{r}(N) \), and \( s \in S_R \). First, we can show easily by induction on \( n \), that for all non-empty trace \( \omega \), for all states \( t \in S_R, z \in S_{CR} \), it holds that:

\[
R\text{Prob}_{CR}^{\leq n}((t, w), \{ z \}) = \begin{cases} 
R\text{Prob}_{R}^{\leq n}(t, u) & \text{if } z = (u, w); \\
0 & \text{if } \nexists u \text{ with } z = (u, w). 
\end{cases}
\] (8)

Now, observe that \( \text{corr}^{-1}_R(\text{success}) = \{ \text{success} \} \). We use the limit-based caracterisation of trace success probability (see Lemma 31), and we obtain a rewriting of our goal—i.e. item 1 in Lemma 32—as the Equation (9) below:

\[
\forall n \in \mathbb{N}, \text{Prob}_{R}^{n}(s, w) \leq R\text{Prob}_{CR}((s, w), \{ \text{success} \});
\] (9)

The proof of Equation (9) is by induction on length(\( w \)). If \( w = \epsilon \), then \( \text{Prob}_{R}^{n}(s, w) = 1 \). Moreover, \( R\text{Prob}_{CR}((s, \epsilon), \{ \text{success} \}) = 1 \), since \( \text{trans}_{CR}(s, \epsilon)(\tau) = \delta_{\text{success}} \), thus the result
holds. Suppose now that \( w = a.w' \), and that Equation 9 holds for \( w' \). Then:

\[
\text{Prob}_R^n(s, a.w') = \sum_{t \in S_R} \text{RProb}_R^n(s, \{t\}) \cdot \sum_{s' \in \text{supp}(D)} D(s') \cdot \text{Prob}_R^n(s', w') \text{ by Lemma 31}
\]

\[
= \sum_{t \in S_R} \text{RProb}_C^n((s, a.w'), \{(t, a.w')\}) \cdot \sum_{s' \in \text{supp}(D)} D(s') \cdot \text{Prob}_C^n(s', w')
\]

\[
\leq \sum_{t \in S_R} \text{RProb}_C^n((s, a.w'), \{(t, a.w')\}) \cdot \sum_{s' \in \text{supp}(D)} D(s') \cdot \text{RProb}_C^n((s', w'), \{success\}) \text{ by HI;}
\]

\[
= \sum_{u=(t, a.w') \in SCR} \text{RProb}_C^n((s, a.w'), \{u\}) \cdot \sum_{v=(s', w') \in \text{supp}(D)} D(v) \cdot \text{Prob}_C(v, \{success\})
\]

\[
\leq \sum_{u=(t, a.w') \in SCR} \text{RProb}_C^n((s, a.w'), \{u\}) \cdot \sum_{v=(s', w') \in \text{supp}(D)} D(v) \cdot \text{Prob}_C(v, \{success\})
\]

\[
= \text{RProb}_C^n((s, a.w'), \{success\})
\]

\[\square\]

**Proof of item 2 in Lemma 32.** Let \( R \in R_{nr}(N^r) \), and \( s \in S_R \) such that there exists \( u, \text{corr}_R(s) = (u, w) \). First, observe that we can restate our goal as:

\[
\text{Prob}_{DR}(s, w) \geq \text{RProb}^{\leq n}_R(s, \text{corr}^{-1}_R(\text{success})).
\]

We do the proof by induction on \( n \):

- \( n = 0 \): since by hypothesis \( \text{corr}_R(s) = (u, w) \neq \text{success} \), it holds that \( \text{RProb}^{\leq 0}_R(s, \text{corr}^{-1}_R(\text{success})) = 0 \), thus the result holds;

- \( n = 1 \): looking at the way we define \( N^r \), we see that \( \text{RProb}^{\leq 1}_R(s, \text{corr}^{-1}_R(\text{success})) > 0 \) implies that \( w = \epsilon \). When it is the case, \( \text{Prob}_{DR}(s, w) = 1 \), thus we can conclude;

- \( n \to n + 1 \) with \( n > 0 \): if \( \text{RProb}^{\leq n+1}_R(s, \text{corr}^{-1}_R(\text{success})) = 0 \), we have immediately...
Lemma 33. Let $N = (S_N, A_{ext} \sqcup A_{int}, \text{trans}_N)$ be a NPLTS. Let be $s \in S_N$, and $w \in A_{ext}^*$.

1. Randomized case:

$$\Pr_{R,N}(s, w) \leq \sup\{\Pr_{R}(s', \text{corr}_R^{-1}(success)) \mid R \in \mathcal{R}_r(N^{tr}), s' \in S_R \text{ corr}_R(s') = (s, w)\}$$

2. Non-randomized case:

$$\Pr_{R_n,N}(s, w) = \sup\{\Pr_{R}(s', \text{corr}_R^{-1}(success)) \mid R \in \mathcal{R}_{n}(N^{tr}), s' \in S_R \text{ corr}_R(s') = (s, w)\}$$

Proof. We do the proof using Lemma 32; let us fix $s \in S_N$.

- Using (the first part of) Lemma 32, we see that for every $R \in \mathcal{R}_r(N)$, and $s' \in S_R$ with $\text{corr}_R(s') = s$, it holds that $\Pr_{R}(s', w) \leq \Pr_{CR}(s', w, \text{corr}_{CR}^{-1}(success))$. From there, we can see that $\Pr_{R}(s', w) \leq \sup_{R \in \mathcal{R}_r(N^{tr}), u \in S_R \text{ corr}_R(u) = (s, w)} \Pr_{R}(u, \text{corr}_R^{-1}(success))$. As a consequence, we obtain the item 1 in Lemma 33. Since moreover $C(\mathcal{R}_{n}(N)) \subseteq \mathcal{R}_{n}(N^{tr})$, we obtain also the \leq inequality for the item 2 in Lemma 33.
• Using (the second part of) Lemma 32, we see that for every \( R \in R_{nr}(N_{tr}) \), and \( s' \in S_R \), such that \( \text{corr}_R(s') = (s, w) \), it holds that \( \text{RProb}_R(s', \text{corr}_R^{-1}(success)) \leq \text{Prob}_{DR}(s', w) \). From there, we obtain that \( \text{RProb}_R(s', \text{corr}_R^{-1}(success)) \leq \text{Prob}_{R_{nr}(N_{tr})}(s, w) \), thus we can conclude.

\[ \]

Proposition 2. Let \((P, \phi)\) and \((P', \phi')\) be two extended processes.

\[ (P, \phi) \leq_{Tr} (P', \phi') \iff (P, \phi) \leq_{Tr} (P', \phi') \]

Proof. First, observe that it is enough to show that for every trace \( w \), for every \( N \)-state \( u \):

\[ \text{Prob}_{R_{tr}(N)}(u, w) = \text{Prob}_{R_{nr}(N)}(u, w). \]

• Since \( R_{nr}(N) \subseteq R_{tr}(N) \), we see immediately that \( \text{Prob}_{R_{tr}(N)}(u, w) \geq \text{Prob}_{R_{nr}(N)}(u, w) \).

• We now show the reverse inequality.

\[ \text{Prob}_{R_{tr}(N)}(u, w) = \sup \{ \text{RProb}_R(s, \text{corr}_R^{-1}(success)) \mid R \in R_{tr}(N_{tr}), s' \in S_R, \text{corr}_R(s') = s \} \text{ by Lemma 33} \]

\[ = \sup \{ \text{RProb}_{\leq n}(s, \text{corr}_R^{-1}(success)) \mid R \in R_{tr}(N_{tr}), s' \in S_R, \text{corr}_R(s') = s, n \in \mathbb{N} \} \]

\[ \leq \sup \{ \text{RProb}_{\leq n}(s, \text{corr}_R^{-1}(success)) \mid R \in R_{nr}(N_{tr}), s' \in S_R, \text{corr}_R(s') = s, n \in \mathbb{N} \} \text{ by Lemma 29} \]

\[ = \text{Prob}_{R_{nr}(N)}(u, w) \text{ by Lemma 33} \]

C Relations \( \rightarrow_{tr} \) and \( \Rightarrow_{R_{tr}} \) coincide

Lemma 34. Let \( N = N = (S_N, A_{ext} \sqcup A_{int}, trans_N) \) be an NPLTS and a resolution \( R = (S_R, corr_R, trans_R) \) of \( N \). For all subset \( T \subseteq S_R \), for all \( s \in S_R \),

\[ \text{RProb}_R(s, T) = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{otherwise.} \end{cases} \]

\[ \text{RProb}_R(s, T) = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \notin T \land \text{trans}_R(s) = * \\ \sum_{u \in \text{supp}(D)} D(u) \cdot \text{RProb}_R(u, T) & \text{if } s \notin T \land \text{trans}_R(s) = (a, D) \end{cases} \]

Proof. Direct from Definition 7 and the fact that \( \text{RProb}_{\leq n}(s, T) \) is increasing on \( n \). \( \square \)

We now state three technical lemmas—Lemmas 35, 36, 37—about basic properties of the reduction relation \( \rightarrow \) on distributions. These lemmas are used later in the proof of Lemma 1. 90
Lemma 35. Let $X$ be a discrete set, $\rightarrow \subset X \times \mathcal{D}(X)$ a binary relation, and $s \in X$.

$$\delta_s \rightarrow_D D \iff D \in \text{conv}\{E \mid s \rightarrow E\}$$

Proof. Direct from Definition 14 and Notation 4. \qed

Lemma 36. Let $N = (S_N, \mathcal{A}_{ext} \sqcup \mathcal{A}_{int}, \text{trans}_N)$ be a NPLTS, $R = (\mathcal{S}_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}_r(N)$ and $D \in \mathcal{D}^\leq_1(\mathcal{S}_R)$ such that $\text{supp}(D) \subseteq \mathcal{S}_{int}(R)$. We have $\text{corr}_R(D) \rightarrow_r \text{corr}_R(\text{trans}_R(D))$.

Proof. By Definition 5 for all $s \in \text{supp}(D)$, $\text{corr}_R(\text{trans}_R(s)) \in \text{conv}(\text{trans}_N(\text{corr}_R(s)))$. Hence $\text{corr}_R(\text{trans}_s(s)) \in \text{conv}(\{E_s \mid \text{corr}_R(s) \rightarrow E_s\})$. Furthermore, $\text{corr}_R(\text{trans}_R(D)) = \sum_{s \in \text{supp}(D)} D(s) \cdot \text{corr}_R(\text{trans}_R(s)) = \sum_{s \in \text{supp}(D)} D(s) \cdot \sum_{i=1}^{n_s} \alpha_i^s \cdot E_i^s = \sum_{s \in \text{supp}(D)} \sum_{i=1}^{n_s} (D(s) \cdot \alpha_i^s) \cdot E_i^s$ with $\text{corr}_R(s) \rightarrow E_i^s$ for all $i \in \{1, \ldots, n_s\}$ and $\sum_{i=1}^{n_s} \alpha_i^s = 1$. Since $D = \sum_{s \in \text{supp}(D)} D(s) \cdot \sum_{i=1}^{n_s} \alpha_i^s \cdot \delta_s$, we conclude that $\text{corr}_R(D) = \sum_{s \in \text{supp}(D)} D(s) \cdot \sum_{i=1}^{n_s} \alpha_i^s \cdot \delta_{\text{corr}_R(s)}$ and so $\text{corr}_R(D) \rightarrow_r \text{corr}_R(\text{trans}_R(D))$. \qed

Lemma 37. Let $X$ be any discrete set, and let $\rightarrow$ be a binary relation in $X \times \mathcal{D}(X)$.

Suppose that $D \rightarrow E$. There exists a family of sub-distribution over $X$: $(E_s)_{s \in \text{supp}(D)}$ such that for all $s \in \text{supp}(D)$, $\delta_s \rightarrow_D E_s$ and $E = \sum_{s \in \text{supp}(D)} D(s) \cdot E_s$.

Proof. Assume that $D \rightarrow E$. By definition, there exists a countable set $I$ and a multiset $\{x_i\}_{i \in I}$ with for all $i \in I$, $x_i \in X$ such that:

- $D = \sum_{i \in I} \alpha_i \cdot \delta_{x_i}$
  - for all $i \in I$, $x_i \rightarrow D_i$
  - $E = \sum_{i \in I} \alpha_i \cdot D_i$

  Note that $E = \sum_{s \in \text{supp}(D)} \sum_{i=1}^{n_s} \alpha_i \cdot D_i$ and for all $s \in X$, $D(s) = \sum_{i=1}^{n_s} \alpha_i$. Hence, $E = \sum_{s \in \text{supp}(D)} D(s) \cdot \sum_{i=1}^{\alpha_i} \frac{\alpha_i}{D(s)} \cdot D_i$. If we define for all $s \in \text{supp}(D)$, $E_s = \sum_{i=1}^{\alpha_i} \frac{\alpha_i}{D(s)} \cdot D_i$, we obtain that $E = \sum_{s \in \text{supp}(D)} D(s) \cdot E_s$ and for all $s \in \text{supp}(D)$, $E_s \in \text{conv}(\{F \mid s \rightarrow F\})$ which implies $\delta_s \rightarrow_D E_s$ by Lemma 35. \qed

Corollary 4. Let $X$ be any discrete set, and let $\rightarrow$ be a binary relation in $X \times \mathcal{D}(X)$.

Suppose that $D \rightarrow E + F$. Then there exists two families of sub-distribution over $X$: $(E_s)_{s \in \text{supp}(D)}$, $(F_s)_{s \in \text{supp}(D)}$ such that for all $s \in \text{supp}(D)$, $\delta_s \rightarrow_D E_s + F_s$, $E = \sum_{s \in \text{supp}(D)} D(s) \cdot E_s$ and $F = \sum_{s \in \text{supp}(D)} D(s) \cdot F_s$.

Proof. By Lemma 37 we know that there exists $(D_s)_{s \in \text{supp}(D)}$ such that for all $s \in \text{supp}(D)$, $\delta_s \rightarrow_D D_s$ and $E + F = \sum_{s \in \text{supp}(D)} D(s) \cdot D_s$.

We define the two families $(E_s)_{s \in \text{supp}(D)}$ and $(F_s)_{s \in \text{supp}(D)}$ as follows: for all $s \in \text{supp}(D)$, for all $x \in X$,

- $E_s(x) = \frac{D_s(x) \cdot E(x)}{E(x) + F(x)}$
\[ F_s(x) = \frac{D_s(x) \cdot F(x)}{E(x) + F(x)} \]

We trivially have \( D_s = E_s + F_s \).

**Lemma 1.** Let \( N = (S_N, \mathcal{A}_{ext} \sqcup \mathcal{A}_{int}, \text{trans}_N) \) be a NPLTS and \( D, E \in D^{\leq 1}(S_N) \).

\[ D \xrightarrow{r} E \text{ if and only if } D \xrightarrow{r} E \]

**Proof.** We prove the two directions of the equivalences separately.

**Left-to-right direction.** Let us suppose that \( D \xrightarrow{r} E \). It means that there exists an infinite scheme of the form:

\[
D = D_0^+ + D_0^T \\
D_0^+ \xrightarrow{r} D_1^+ + D_1^T \\
\vdots \\
D_k^+ \xrightarrow{r} D_{k+1}^+ + D_{k+1}^T \\
\vdots
\]

with \( E = \sum_{k \in \mathbb{N}} D_k^T \). The proof consists of two steps: we first define a resolution \( R \in \mathcal{R}_r(N) \), and in a second step we use \( R \) to show that \( D \xrightarrow{r} E \).

1. We define \( R = (S_R, \text{corr}_R, \text{trans}_R) \), where \( S_R = \{(s, n, \alpha) \mid n \in \mathbb{N}, \alpha \in \{\rightarrow, \top\}, s \in \text{supp}(D_n^+)\} \) and \( \text{corr}(s, n, \alpha) = s \) for every \( n \in \mathbb{N} \) and every \( \alpha \in \{\rightarrow, \top\} \), and

\[
\text{trans}_R(s, n, \alpha) = \begin{cases} \tau_A & \text{if } \alpha = \top; \\
(\tau, \sum_{s' \in S} \Delta(s, n, \top)(s') \cdot \delta(s', n+1, \top) + \Delta(s, n, \rightarrow)(s') \cdot \delta(s', n+1, \rightarrow)) & \text{if } \alpha = \rightarrow.
\end{cases}
\]

where \( \tau_A \) stands for the function \((a \in \mathcal{A}_{ext} \mapsto *)\), \( S = \{s \mid (s, n, \alpha) \in S_R\} \) and for all \( s \in \text{supp}(D_n^+) \), \( n \in \mathbb{N}, \alpha \in \{\rightarrow, \top\} \), the sub-distributions \( \Delta(s, n, \alpha) \) are obtained from Corollary 4 such that:

\[
\delta_s \xrightarrow{r} \Delta(s, n, \rightarrow) + \Delta(s, n, \top), \quad \forall s \in \text{supp}(D_n^+); \quad (13)
\]

\[
D_{n+1}^+ = \sum_{s \in \text{supp}(D_n^+)} D_n^+(s) \cdot \Delta(s, n, \rightarrow); \quad (14)
\]

\[
D_{n+1}^T = \sum_{s \in \text{supp}(D_n^+)} D_n^+(s) \cdot \Delta(s, n, \top). \quad (15)
\]

We first check that \( R \) is indeed a \( \mathbb{N} \)-resolution: first observe that for all external states \( s' \in S_{\text{ext}}(R) \), for all \( a \in \mathcal{A}_{ext}, \text{trans}_R(s') = * \). Secondly, for all internal states \( s' \in S_{\text{int}}(R) \) with \( \text{trans}_R(s') = (\tau, F) \), it means that \( s' = (s, n, \rightarrow) \), \( s \in \text{supp}(D_n^+) \) and

\[
F = \sum_{s' \in S} \Delta(s, n, \top)(s') \cdot \delta(s', n+1, \top) + \sum_{s' \in S} \Delta(s, n, \rightarrow)(s') \cdot \delta(s', n+1, \rightarrow)
\]
2. We define a distribution over $\mathbb{R}$-states: $D' = \sum_{s \in S} D_0^s(s) \cdot \delta_{s,0,\rightarrow} + \sum_{s \in S} D_0^s(s) \cdot \delta_{s,0,\tau}$.

Looking at Definition 13, we see that we need now to compute the sub-distribution $\text{corr}(E')$, with $E'$ defined as $\text{supp}(E') \subseteq S_{\text{ext}}(R)$ and for all $u' \in S_{\text{ext}}(R)$, $E'(u') = \sum_{s' \in S_R} D'(s') \cdot \text{RProb}_R(s', \{u'\})$. First, recall that for all $u' \in S_{\text{ext}}(R)$, $u'$ is of the form $(u, p, \top)$ with $p \in \mathbb{N}$ and $u \in S$. Additionally, note that both $\text{supp}(D_0^s)$ and $\text{supp}(D_0^s)$ are included in $S \subseteq S_N$.

$$E'(u, p, \top) = \sum_{s \in S_N} D_0^s(s) \cdot \text{RProb}_R((s, 0, \rightarrow), \{(u, p, \top)\})$$

$$+ \sum_{s \in S_N} D_0^s(s) \cdot \text{RProb}_R((s, 0, \top), \{(u, p, \top)\})$$

Observe that for every $(s, 0, \top) \in S_{\text{ext}}(R)$, $\text{RProb}_R((s, 0, \top), \{(u, p, \top)\}) = 1$ if $u = s$, $p = 0$ and 0 otherwise.

Furthermore, by definition of $\text{trans}_R$, if $\text{trans}_R(s, n, \rightarrow) = (\tau, F)$ then $(s', n', \alpha') \in \text{supp}(F)$ implies $n' = n + 1$. Thus, we can easily prove by induction that for all $(s, n, \rightarrow), (t, m, \top) \in S_R$, $\text{RProb}_R((s, n, \rightarrow), \{(t, m, \top)\}) > 0$ implies that $m > n$.

Using these two facts, we obtain that:

$$E'(u, 0, \top) = D_0^\top(u) \quad (16)$$

$$E'(u, p, \top) = \sum_{s \in S_N} D_0^s(s) \cdot \text{RProb}_R((s, 0, \rightarrow), \{(u, p, \top)\}) \quad \text{for } p \geq 1. \quad (17)$$

To go forward from here, we are going to show the following auxiliary statement: for all $p \in \mathbb{N}$, for all $m < p \in \mathbb{N}$, for all $u \in S$,

$$\sum_{s \in S_N} D_m^s(s) \cdot \text{RProb}_R((s, m, \rightarrow), \{(u, p, \top)\}) = D_p^\top(u). \quad (18)$$

We do the proof by induction on $p - m$. For all $(s, n, \rightarrow) \in S_{\text{int}}(R)$, let us denote $F_{s,n}$ the distribution such that $\text{trans}_R(s, n, \rightarrow) = (\tau, F_{s,n})$.

$$\sum_{s \in S_N} D_m^s(s) \cdot \text{RProb}_R((s, m, \rightarrow), \{(u, p, \top)\})$$

$$= \sum_{s \in S_N} D_m^s(s) \cdot \sum_{t \in \text{supp}(F_{s,m})} F_{s,m}(t) \cdot \text{RProb}_R(t, \{(u, p, \top)\})$$ by Lemma 34
By definition of $F_{s,n}$, $t \in \text{supp}(F_{s,n})$ implies that $t = (t', m + 1, \alpha)$ with $t' \in S$ and $\alpha \in \{\top, \rightarrow\}$ and $F_{s,n}(t) = \Delta(s, m, \top)(t') + \Delta(s, m, \rightarrow)(t')$. Hence:

\[
\sum_{s \in S_n} D_m^{-}(s) \cdot \text{RProb}_R((s, m, \rightarrow), \{(u, p, \top)\})
\]

\[
= \sum_{s \in S_n} D_m^{-}(s) \cdot \left( \sum_{t \in S_n} \Delta(s, m, \rightarrow)(t) \cdot \text{RProb}_R((t, m + 1, \top), \{(u, p, \top)\}) \right)
\]

\[
+ \sum_{t \in S_n} \Delta(s, m, \rightarrow)(t) \cdot \text{RProb}_R((t, m + 1, \rightarrow), \{(u, p, \top)\})
\]

• if $p = m + 1$ then for all $t \in S_n$, $\text{RProb}_R((t, m + 1, \top), \{(u, m + 1, \top)\}) = 1$ if $t = u$ and 0 otherwise. Moreover, as we previously shown, $\text{RProb}_R((t, m + 1, \rightarrow), \{(u, m + 1, \top)\}) = 0$. Therefore:

\[
\sum_{s \in S_n} D_m^{-}(s) \cdot \text{RProb}_R((s, m, \rightarrow), \{(u, p, \top)\}) = \sum_{s \in S_n} D_m^{-}(s) \cdot \Delta(s, m, \top)(u)
\]

\[
= D_p^T(u) \quad \text{by (15)}.
\]

• if $p > m + 1$ then for all $t \in S_n$, $\text{RProb}_R((t, m + 1, \top), \{(u, p, \top)\}) = 0$. Hence:

\[
\sum_{s \in S_n} D_m^{-}(s) \cdot \text{RProb}_R((s, m, \rightarrow), \{(u, p, \top)\})
\]

\[
= \sum_{s \in S_n} D_m^{-}(s) \cdot \sum_{t \in S_n} \Delta(s, m, \rightarrow)(t) \cdot \text{RProb}_R((t, m + 1, \rightarrow), \{(u, p, \top)\})
\]

\[
= \sum_{t \in S_n} \left( \sum_{s \in S_n} D_m^{-}(s) \cdot \Delta(s, m, \rightarrow)(t) \right) \cdot \text{RProb}_R((t, m + 1, \rightarrow), \{(u, p, \top)\})
\]

\[
= \sum_{t \in S_n} D_{m+1}^{-}(t) \cdot \text{RProb}_R((t, m + 1, \rightarrow), \{(u, p, \top)\}) \quad \text{by (14)}
\]

\[
= D_p^T(u) \quad \text{by induction hypothesis}
\]

This concludes the proof of Equation (18). Combining (16), (17) and (18), we have shown:

\[
E'(u, p, \top) = D_p^T(u) \quad \forall p \in \mathbb{N}.
\]

We conclude the proof by noticing that $\text{corr}(E')(u) = \sum_{p \in \mathbb{N}} D_p^T(u) = E(u)$.

**Right-to-left direction.** We suppose that $D \Rightarrow_R E$. It means that there exists a resolution $R = (S_R, \text{corr}_R, \text{trans}_R) \in \mathcal{R}_r(\mathbb{N})$ and $D', E' \in \mathcal{D}^{\leq 1}(S_R)$ such that $\text{corr}_R(D') = D$, $\text{corr}_R(E') = E$, $\text{supp}(E') \subseteq S_{\text{ext}}(R)$ and for every $u \in S_{\text{ext}}(R)$, $E'(u) = \sum_{s' \in S_R} D'(s') \cdot \text{RProb}_R(s', \{u\})$. From there, we are going to build an infinite scheme, in the spirit of Definition (15). We
define by induction on \( n \) two families of sub-distributions on \( S_R \), \((D^\rightarrow_n)_{n \in \mathbb{N}}\) and \((D^\top_n)_{n \in \mathbb{N}}\), as follows:

\[
D^\rightarrow_0 = D^\leftarrow_{\text{ext}(R)} \\
D^\rightarrow_{n+1} = \text{trans}_R(D^\rightarrow_n) |_{\text{int}(R)} \\
D^\top_0 = D^\leftarrow_{\text{ext}(R)} \\
D^\top_{n+1} = \text{trans}_R(D^\rightarrow_n) |_{\text{ext}(R)}.
\]

We denote for all \( n \in \mathbb{N} \), \( D'_n = D'_n \) and \( D'_n = D'_n \). Hence, \( D' = D'_0 \). Using Lemma 36, we can check that for all \( n \in \mathbb{N} \), \( \text{corr}(D^\rightarrow_n) \rightarrow \text{corr}(D'_n+1) \). Then we define, for every \( n \in \mathbb{N} \), \( \alpha \in \{\rightarrow, \top\}, D'^n = \text{corr}(D'^n) \in D^\alpha_n(S_R) \).

We now show that for all \( n \in \mathbb{N} \), for all \( j \leq n \), for all \( u \in S_{\text{ext}(R)} \),

\[
\sum_{s' \in S_R} D'_j(s') \cdot R\text{Prob}^{\leq n-j}(s', \{u\}) = \sum_{j \leq k \leq n} D^\top_k(u). \tag{19}
\]

We prove this property by induction on \( n - j \):

- if \( n = j \), then we can first see that \( \sum_{s' \in S_R} D'_j(s') \cdot R\text{Prob}^{\leq n-j}(s', \{u\}) = D'_j(u) \). Since \( u \) is an external state, and by definition of \( D'_j \), it holds that \( D'_j(u) = D'^\top_j(u) \), so the result holds.

- case \( n - j > 0 \):

\[
\sum_{s' \in S_R} D'_j(s') \cdot R\text{Prob}^{\leq n-j}(s', \{u\}) \\
= \sum_{s' \in S_R} D'_j(s') \cdot R\text{Prob}^{\leq n-j}(s', \{u\}) + \sum_{s' \in S_R} D'_j(s') \cdot R\text{Prob}^{\leq n-j}(s', \{u\}) \\
= \sum_{s' \in S_R} D'_j(s') \cdot R\text{Prob}^{\leq n-j}(s', \{u\}) + D'^\top_j(u) \quad \text{since } s' \in \text{supp}(D'^\top_j) \Rightarrow s' \in S_{\text{ext}(R)} \\
= \sum_{s' \in S_R, \text{trans}(s') = (\tau, F_s)} D'_j(s') \cdot \sum_{v' \in S_R} F_s(v') \cdot R\text{Prob}^{\leq n-j-1}(v', \{u\}) + D'^\top_j(u) \\
= \sum_{v' \in S_R} D'_{j+1}(v') \cdot R\text{Prob}^{\leq n-(j+1)}(v', \{u\}) + D'^\top_j(u) \quad \text{since } \text{trans}(D'^\rightarrow_j) = D'_{j+1} \\
= \sum_{j+1 \leq k \leq n} D'^\top_k(u) + D'^\top_j(u) \quad \text{by induction hypothesis},
\]

and that concludes the proof of (19).
We are now able to end the proof, by seeing that for all \( w \in S_N \),

\[
E(w) = \sum_{u \in \text{corr}_{R}^{-1}(w)} E'(u)
\]

since by hypothesis \( \text{corr}_{R}(E') = E \)

\[
= \sum_{u \in \text{corr}_{R}^{-1}(w)} \sum_{s' \in S_R} D'_0(s') \cdot R\text{Prob}_R(s', \{u\})
\]

by hypothesis, and since \( D' = D'_0 \)

\[
= \sup_{n \in \mathbb{N}} \sum_{u \in \text{corr}_{R}^{-1}(w)} \sum_{s' \in S_R} D'_0(s') \cdot R\text{Prob}^\leq_n(s', \{u\})
\]

\[
= \sup_{n \in \mathbb{N}} \sum_{0 \leq k \leq n} D_{k}^T(u)
\]

by \ref{19} instantiated with \( j = 0 \)

\[
= \sum_{k \in \mathbb{N}} D_{k}^T(w),
\]

which ends the proof.

\[\square\]

**D Observational equivalence with name restriction in context**

In the original applied pi calculus, observational equivalence is closed by application of an adversarial context, i.e., if \( P \approx Q \) then \( C[P] \approx C[Q] \) where \( C[\_] \) is a context of the form \( \text{\texttt{new}} a_1; \ldots; \text{\texttt{new}} a_n; (\_ | \text{Adv}) \) where \( \text{Adv} \) is a process. Intuitively, previously public names \( a_1, \ldots, a_n \) in \( P \) and \( Q \) may becomes restricted, i.e. private, by the application of the adversarial context.

In our definition of observational equivalence (Definition \[18\]), we only require that the adversarial process \( \text{Adv} \) is put in parellel of \( P \) and \( Q \) without allowing for additional name restrictions. In our formalism, this corresponds to the third bullet point of Definition \[18\], i.e.:

for all closed \( \text{Adv} \in \mathcal{MP} \) such that \( \text{fn}(\text{Adv}) \subseteq \mathcal{N}_{\text{pub}}, \{\text{Adv}\} \cup \mathcal{P} R \{\text{Adv}\} \cup \mathcal{Q} \).

We show in this section that in fact adding these restrictions in the definition of observational equivalence does not add any power to the attacker by showing that the classical definition and our definition coincide. Note that the original formalism of the applied pi calculus is slightly different from ours as they do not separate names in public and private names as we do. Furthermore, they do not have a rule that executes the name restriction as our rule (\texttt{New}). Instead, they consider a structural equivalence \( \equiv \) that manages name restrictions. For instance,

\[
\text{\texttt{new}} a; 0 \equiv 0 \quad \text{\texttt{new}} a; \text{\texttt{new}} b; P \equiv \text{\texttt{new}} b; \text{\texttt{new}} a; P
\]

\[
P \mid \text{\texttt{new}} a; Q \equiv \text{\texttt{new}} a; (P \mid Q) \text{ when } a \not\in \text{fv}(P) \cup \text{fn}(P)
\]
Similarly, by the fact that our calculus relies on multisets of processes, the rules (NULL), (PAR) and (REPL) capture the following equations in the structural equivalence:

\[
P \equiv P' \mid 0 \quad P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3 \quad P \mid Q \equiv Q \mid P \quad !P \equiv !Pan

Thus, as mentioned in [ABF17, Section 2.3], any process \( P \) in the original applied pi calculus is structurally equivalent as a process of the form \texttt{new} \( a_1; \ldots; \texttt{new} \ a_n; (P_1 \mid \ldots \mid P_n) \) where name restrictions have been pushed to the top, which can be expressed by a configuration \( \begin{cases} \{a_1, \ldots, a_n\}, \{P_1, \ldots, P_n\} \end{cases} \). In our calculus, the distinction between private and public names enforces that all names \( a_1, \ldots, a_n \) are in \( \mathcal{N}_{priv} \) which allows us to only keep the multiset \( \{P_1, \ldots, P_n\} \) (see Figure 2).

Therefore, applying an adversarial context with restriction is equivalent in our formalism to applying a renaming from public names of our process to fresh privates names. This leads to the following definition of the original observational equivalence:

**Definition 47.** The original observational preorder \( \leq_{ori}^\mathcal{R} \) is the largest relation \( R \) on \( \mathcal{MP} \) such that \( P \mathrel{R} Q \) implies:

- for all \( c \in \mathcal{N}_{pub} \), \( \text{RProb}^\mathcal{R}(P, \downarrow c) \leq \text{RProb}^\mathcal{R}(Q, \downarrow c) \);
- if \( P \overset{\tau}{\rightarrow}, D \) then \( Q \overset{\tau}{\rightarrow}, E \) and \( D \overset{R}{\rightarrow} E \);
- for all closed \( \text{Adv} \in \mathcal{MP} \), for all renaming \( \rho \), if \( \text{fn(Adv)} \subseteq \mathcal{N}_{pub}, \text{dom(\rho)} \subseteq \mathcal{N}_{pub}, \text{img(\rho)} \subseteq \mathcal{N}_{priv} \) and \( \text{img(\rho) \cap names(P, Q, Adv) = \emptyset} \) then \( \{\text{Adv}\rho\} \cup \mathcal{P}_\rho \mathrel{R} \{\text{Adv}\rho\} \cup \mathcal{Q}_\rho \).

The original observational equivalence \( \approx_{ori}^\mathcal{R} \) is defined by additionally requesting \( R \) to be symmetric and in the second bullet point, by requesting both \( D \overset{R}{\rightarrow} E \) and \( E \overset{R}{\rightarrow} D \) to hold.

Let us now show that the two notions coincide.

**Lemma 38.** \( \leq_{ori}^\mathcal{R} = \leq_{obs}^\mathcal{R} \) and \( \approx_{ori}^\mathcal{R} = \approx_{obs}^\mathcal{R} \).

**Proof.** Trivially, we have that \( \leq_{ori}^\mathcal{R} \subseteq \leq_{obs}^\mathcal{R} \) and \( \approx_{ori}^\mathcal{R} \subseteq \approx_{obs}^\mathcal{R} \) since we can always take \( \rho \) being the empty renaming.

Consider the relation \( \mathcal{R} \) defined as \( \mathcal{P} \mathrel{\mathcal{R}} \mathcal{Q} \) iff there exists \( \mathcal{P}', \mathcal{Q}' \) and a renaming such that \( \rho \) such that \( \mathcal{P} = \mathcal{P}' \rho, \mathcal{Q} = \mathcal{Q}' \rho, \mathcal{P}' \leq_{obs}^\mathcal{R} \mathcal{Q}', \text{dom(\rho)} \subseteq \mathcal{N}_{pub}, \text{img(\rho)} \subseteq \mathcal{N}_{priv} \) and \( \text{img(\rho) \cap names(\mathcal{P}', \mathcal{Q}', \text{Adv}) = \emptyset} \).

Recall that \( \equiv \) is closed under substitution of names by terms. Therefore, for all \( u, v \) such that \( \text{img(\rho) \cap names(u, v) = \emptyset} \), \( u \equiv v \) if and only if \( u \rho = v \rho \). Thus, \( \mathcal{P} \rho \rightarrow_{\tau} D' \) implies \( \mathcal{P} \rightarrow_{\tau} D \) where \( \text{dom(D')} = \text{dom(D)}\rho \) and \( D'(x\rho) = D(x\rho) \). Furthermore, if \( \mathcal{P} \rightarrow_{\tau} D \) then \( \mathcal{P} \rho \rightarrow_{\tau} D \rho \) where \( D \rho \) is the distribution such that \( \text{dom(D\rho)} = \text{dom(D)} \) and \( D\rho(x\rho) = D(x\rho) \). Notice that for the rule (New), we consider w.l.o.g. that \( a' \) is fresh also with respect to the private names in \( \rho \). In other words, we have that \( \mathcal{P} \rho \rightarrow_{\tau} D \rho \) if \( \mathcal{P} \rightarrow_{\tau} D \).

This can be propagate to schedulers, i.e. \( \text{(corr, R)} \in \mathcal{R}_\rho^r \) if and only if \( \text{(corr}\rho, R) \in \mathcal{R}_\rho^r \). Hence, we derive that
We now show that $\mathcal{R}$ satisfies the three items in Definition \[47\]:

- Let $c \in \mathcal{N}_{pub}$. If $c \in \text{dom}(\rho)$ then $c$ does neither occur in $\mathcal{P}' \rho$ nor in $\mathcal{Q}' \rho$. Hence, $\text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{P}', \downarrow c) = 0 = \text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{Q}', \downarrow c)$. Otherwise, $\downarrow c = \downarrow c \rho \cup \mathcal{T}$ where for all $\mathcal{P}' \in \mathcal{T}$, $\text{fn}(\mathcal{P}') \cap \text{dom}(\rho) \neq \emptyset$. Since, $\mathcal{P}' \rho$ and $\mathcal{Q}' \rho$ do not contain public names from $\text{dom}(\rho)$, they can never reach states from $\mathcal{T}$. Hence, $\text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{P}' \rho, \downarrow c) = \text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{P}' \rho, \downarrow c) \leq \text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{P}' \rho, \downarrow c) \leq \text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{Q}' \rho, \downarrow c)$. Since $\mathcal{P}' \rho \leq_{\text{obs}} \mathcal{Q}'$, we deduce that $\text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{P}' \rho, \downarrow c) \leq \text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{Q}' \rho, \downarrow c) \leq \text{RProb}_{\mathcal{R}_{\rho}}(\mathcal{Q}' \rho, \downarrow c)$.

- Let $\mathcal{P} \rho \xrightarrow{\tau} \mathcal{R}_{\rho} \mathcal{P}$. If $\mathcal{P}' \xrightarrow{\tau} \mathcal{R}_{\rho} \mathcal{P}$. By $\mathcal{P}' \leq_{\text{obs}} \mathcal{Q}'$, we have $\mathcal{Q}' \xrightarrow{\tau} \mathcal{R}_{\rho} \mathcal{P}$ and $\mathcal{D} \leq_{\text{obs}} \mathcal{E}$. This implies that $\mathcal{D} \rho \xrightarrow{\mathcal{R}} \mathcal{E} \rho$ with $\mathcal{Q}' \rho \xrightarrow{\tau} \mathcal{R}_{\rho} \mathcal{E}$.

- Let $\text{Adv} \in \mathcal{MP}$ and $\rho'$ be a renaming such that $\text{fn}(\text{Adv}) \subseteq \mathcal{N}_{pub}$, $\text{dom}(\rho') \subseteq \mathcal{N}_{pub}$, $\text{img}(\rho') \subseteq \mathcal{N}_{\text{priv}}$ and $\text{img}(\rho') \cap \text{names}(\mathcal{P}' \rho, \mathcal{Q}' \rho) = \emptyset$. Note that the domains of $\rho$ and $\rho'$ may not be disjoint. Similarly, the process $\text{Adv}$ may contain public names from $\text{dom}(\rho)$.

Hence, let $\rho_{\text{pub}}$ be a renaming such that $\text{dom}(\rho_{\text{pub}}) = \text{dom}(\rho) \cap \text{fn}(\text{Adv})$, $\text{img}(\rho_{\text{pub}}) \subseteq \mathcal{N}_{pub} \setminus \text{fn}(\mathcal{P}', \mathcal{Q}', \text{Adv})$. Hence, we define $\rho_1 = \rho_{\text{pub}}^{-1} \rho'$.

We have $\mathcal{P}' \rho_1 = (\mathcal{P}' \rho) \rho'$ since $\text{img}(\rho_{\text{pub}}) \subseteq \mathcal{N}_{pub} \setminus \text{fn}(\mathcal{P}', \mathcal{Q}', \text{Adv})$. Moreover, as $\text{dom}(\rho_{\text{pub}}) = \text{dom}(\rho) \cap \text{fn}(\text{Adv})$, we have that

$$\text{Adv}_{\rho_{\text{pub}}} \rho = \text{Adv}_{\rho_{\text{pub}}} \rho |_{\text{dom}(\rho) \setminus \text{fn}(\text{Adv})} = \text{Adv}_{\rho} |_{\text{dom}(\rho) \setminus \text{fn}(\text{Adv})} \rho_{\text{pub}} = \text{Adv}_{\rho_{\text{pub}}}$$

Therefore, $\text{Adv}_{\rho_{\text{pub}}} \rho_1 = \text{Adv}_{\rho_{\text{pub}}} \rho_{\text{pub}}^{-1} \rho' = \text{Adv}_{\rho'}$. This allows us to deduce that $\{\text{Adv}_{\rho'}\} \cup \mathcal{P}' \rho' = (\{\text{Adv}_{\rho_{\text{pub}}}\} \cup \mathcal{P}' \rho_1).$ Similarly, we have $\{\text{Adv}_{\rho'}\} \cup \mathcal{Q}' \rho' = (\{\text{Adv}_{\rho_{\text{pub}}}\} \cup \mathcal{Q}' \rho_1).$

We know that $\mathcal{P}' \leq_{\text{obs}} \mathcal{Q}'$. Hence $\{\text{Adv}_{\rho_{\text{pub}}}\} \cup \mathcal{P}' \rho_1 \leq_{\text{obs}} \{\text{Adv}_{\rho_{\text{pub}}}\} \cup \mathcal{Q}' \rho_1$ which allows us to conclude that $\{\text{Adv}_{\rho_{\text{pub}}}\} \cup \mathcal{P}' \rho_1 \mathcal{R} \{\text{Adv}_{\rho_{\text{pub}}}\} \cup \mathcal{Q}' \rho_1$ and so $\{\text{Adv}_{\rho'}\} \cup \mathcal{P}' \rho_1 \mathcal{R} \{\text{Adv}_{\rho'}\} \cup \mathcal{Q}' \rho_1$.

\[\blacksquare\]

E  \(\tau\)-determinisation

**Notation 23.** We says that a NPLTS $\mathcal{N}$ is a **new** -determinization of $\mathcal{N}'$ when $\mathcal{N} = (S_{\mathcal{N}'}, \mathcal{A}_{\mathcal{N}'}, \text{trans})$ with the following constraints on its transition function $\text{trans}$:

- for $a \in \mathcal{A}_{\mathcal{N}'} \setminus \{\tau\}$, $\text{trans}(s)(a) = \text{trans}_{\mathcal{N}'}(s)(a)$, and $\text{trans}(s)(\tau) \subseteq \text{trans}_{\mathcal{N}'}(s)(\tau)$;

- for every $D \in \text{trans}_{\mathcal{N}'}(s)(\tau)$ not obtained by a **new** reduction, $D \in \text{trans}(s)(a)$;
• for \( s = (\{ \text{new } a.Q \} \cup P, \phi) \), there exists at least one \( E \in \text{trans}(s)(\tau) \) with \( E \) of the form \( \delta(\{ Q(a/b) \} \cup P, \phi) \) with \( b \in N_{\text{priv}} \) fresh in \( s \). We say that the new -determinisation is strict when there is exactly one such \( E \in \text{trans}(s)(\tau) \).

Observe that we can easily build a particular strict new -determinisation by ordering the private names, and choosing for each new -reduction the first available fresh private name.

**Notation 24.** We define an equivalence relation on \( SP_\ell \), that we call equivalence with respect to private names substitution, and that we note \( \equiv_{N_{\text{priv}}} : (P, \phi) \equiv_{N_{\text{priv}}} (Q, \psi) \) when there exists private names \( a_1, \ldots, a_n, b_1, \ldots, b_n \) with \( b_1, \ldots, b_n \) fresh in \( (P, \phi) \) such that

\[
(Q, \psi) = \left( P\{a_1/b_1\} \ldots \{a_n/b_n\}, \phi\{a_1/b_1\} \ldots \{a_n/b_n\} \right).
\]

**Lemma 39.** Let \( N \) be a new -determinization of \( N^f \), and \( s, t \in SP_\ell \) such that \( s \equiv_{N_{\text{priv}}} t \). Then \( s \equiv_N^\text{sim} t \).

Observe that in particular \( N^f \) is a (non strict) new -determinization of itself, thus Lemma 39 also holds for \( N = N^f \).

**Lemma 40.** Let \( N \) be a new -determinization of \( N^f \). Let \( s \in SP_\ell \), and \( a \in A_{N^f} \) such that \( s \xrightarrow{a} E \). Then there exists \( E \) such that \( s \xrightarrow{a} E, \) and \( D(\equiv_{N^\text{priv}})E \).

**Lemma 41.** Let \( N \) be a new -determinization of \( N^f \). Let \( s \in SP_\ell \), and \( a \in A_{N^f} \) such that \( s \xrightarrow{a} E \). Then there exists \( E \), such that \( s \xrightarrow{a} E, \) and \( D(\equiv_{N^\text{priv}})E \).

**Notation 25.** Let \( N_1 = (S_{N_1}, A_{\text{ext}} \cup A_{\text{int}}, \text{trans}_{N_1}) \) and \( N_2 = (S_{N_2}, A_{\text{ext}} \cup A_{\text{int}}, \text{trans}_{N_2}) \) be two NPLTSs, with the same set of actions. We define the disjoint union of \( N_1 \) and \( N_2 \) as the NPLTS \( N_1 + N_2 := (S_{N_1} \cup S_{N_2}, A_{\text{ext}} \cup A_{\text{int}}, \text{trans}_{N_1+N_2}) \) where the transition function \( \text{trans}_{N_1+N_2} \) defined as:

\[
\text{trans}_{N_1+N_2}(s) = \begin{cases} 
\text{trans}_{N_1}(s) \text{ when } s \in S_{N_1} \\
\text{trans}_{N_2}(s) \text{ when } s \in S_{N_2}.
\end{cases}
\]

In the following, when considering \( N_1 + N_2 \), where \( N_1 \) and \( N_2 \) have non disjoint states spaces, we will write \( s^{N_1}, s^{N_2} \) to design the copies of \( s \) in \( S_{N_1} \) and \( S_{N_2} \) respectively. Similarly, if \( D \) is a distribution over \( S_{N_1} \cup S_{N_2} \), we will write \( D^{N_1} \) and \( D^{N_2} \) to mean the distributions over \( S_{N_1} \cup S_{N_2} \) where all states \( s \) in the support of the distribution are taken in \( S_{N_1} \) and \( S_{N_2} \) respectively.

**Lemma 42.** Let \( N \) be a new -determinization of \( N^f \). Then the binary relation \( R \subseteq S_{N^f+N} \times S_{N^f+N} \) defined as \( R = \{ (s^{N^f}, t^N) \mid s \equiv_{N^\text{priv}} t \} \cup \{ (s^N, t^{N^f}) \mid s \equiv_{N^\text{priv}} t \} \), is a (strong) bisimulation on the NPLTS \( N_1 + N_2 \).

**Proof.** First, we can easily check that it is an equivalence relation (reflexive, transitive and symmetric).
Let \((s^N, t^N)\) such that \(s \equiv_{N_{\text{priv}}} t\), and let be a such that \(s^{N^\ell} \overset{a}{\rightarrow}_{N_{\text{priv}}^\ell} D^{N^\ell}\). Then by definition of the transition function \(\text{trans}_{N_{\text{priv}}^\ell}\), it means that \(s \overset{a}{\rightarrow}_{N^\ell} D\). Using Lemma 40, we see that there exists \(E\) such that \(s \overset{a}{\rightarrow}_{N} E\), and \(D(\equiv_{N_{\text{priv}}}E)\). Since we know from Lemma 39 that \(\equiv_{N_{\text{priv}}}\) is a bisimulation on \(N\), and since by hypothesis \(s \equiv_{N_{\text{priv}}} t\), we obtain the existence of a third distribution \(F\) such that \(t \overset{a}{\rightarrow}_{N} F\), and \(E \equiv_{N_{\text{priv}}} F\). Recall that by definition of \(N^\ell + N\), \(t \overset{a}{\rightarrow}_{N} F\) implies that \(t^N \overset{a}{\rightarrow}_{N_{\text{priv}}^\ell} F^N\). Moreover, since \(\equiv_{N_{\text{priv}}}\) is transitive, and since \(\sim\) preserves transitivity–from \(\equiv\)–, we can conclude that \(D \equiv_{N_{\text{priv}}} F\).

Let \((s^N, t^N)\) such that \(s \equiv_{N_{\text{priv}}} t\), and let be a such that \(s^N \overset{a}{\rightarrow}_{N_{\text{priv}}^\ell} D^N\). Then it is also true that \(s^{N^\ell} \overset{a}{\rightarrow}_{N_{\text{priv}}^\ell} D^{N^\ell}\). Since by Lemma 39 \(\equiv_{N_{\text{priv}}}\) is a bisimulation on \(N^\ell\), we obtain directly a distribution \(F\) such that \(t \overset{a}{\rightarrow}_{N_{\text{priv}}^\ell} F\), and \(D \equiv_{N_{\text{priv}}^\ell} F\), and we conclude from there.

**Corollary 5.** Let \(N\) be a new -determination of \(N^\ell\). Then \(\leq_{N_{\text{priv}}}^N\) coincides with \(\leq_{N_{\text{priv}}^\ell}^N\).

**Proof.** It is known–and easy to prove–that any pair \((s, t)\) of bisimilar states in a NPLTS \(N\) have also the same supremum probability of doing a trace. Using that fact, and Lemma 42, we see that for any state in \(N^\ell\), and for any \(w \in ((A_{\text{ext}}N^\ell)^*)^*\):

\[
\text{Prob}_{R_{\text{sim}}(N^\ell)}(s, w) = \text{Prob}_{R_{\text{sim}}(N_{\text{priv}}^\ell\cup N)}(s^N, w) \text{ by definition of NPLTSs disjoint union} \\
= \text{Prob}_{R_{\text{sim}}(N_{\text{priv}}^\ell\cup N)}(s^N, w) \text{ since } s^{N^\ell}, s^N \text{ are bisimilar by Lemma 42} \\
= \text{Prob}_{R_{\text{sim}}(N)}(s, w),
\]

and we can conclude from there.

**Corollary 6.** Let \(N\) be a new -determination of \(N^\ell\). Then \(\leq_{N_{\text{sim}}}^N\) coincides with \(\leq_{N_{\text{sim}}^\ell}^N\).

**Proof.** Let us first show that \(\leq_{N_{\text{sim}}}^N\) is a simulation on \(N^\ell\): let \(s, t \in SP_{\ell}\) such that \(s \leq_{N_{\text{sim}}^N} t\). We suppose that for some \(a \in A_{\text{ext}}\), \(s \overset{a}{\rightarrow}_{N^\ell} D\). From there, we can write that \(s^{N^\ell} \overset{a}{\rightarrow}_{N_{\text{sim}}^\ell} D^{N^\ell}\). By Lemma 42, there exists \(E\) such that \(D \equiv_{N_{\text{sim}}^\ell} E\), and \(s^N \overset{a}{\rightarrow}_{N_{\text{sim}}^N} E^N\). From there, we are able to use the fact that \(s \leq_{N_{\text{sim}}^N} t\), to obtain a third distribution \(F\) such that \(t^N \overset{a}{\rightarrow}_{N_{\text{sim}}^N} F^N\), and \(E \leq_{N_{\text{sim}}^\ell} F\). Since a strong bisimulation is also a weak bisimulation for any NPLTS–and Proposition 4, we can again use Lemma 42 to get a distribution \(H\) such that \(F \equiv_{N_{\text{sim}}^\ell} H\) and \(t^{N^\ell} \equiv_{N_{\text{sim}}^\ell} H^{N^\ell}\). To recap, it means that we have \(t \overset{a}{\rightarrow}_{N^\ell} H\), and \(D(\equiv_{N_{\text{sim}}^\ell} \leq_{N_{\text{sim}}^\ell} \equiv_{N_{\text{sim}}^\ell}) H\). By Lemma 39, \(\equiv_{N_{\text{sim}}^\ell} \subseteq \leq_{N_{\text{sim}}^\ell}\), and we can conclude by transitivity of \(\leq_{N_{\text{sim}}^\ell}\) and Lemma 42.

We show similarly that \(\leq_{N_{\text{sim}}^N}\) is a simulation on \(N\).

As a consequence from Corollary 5 and 42, it is enough to consider \(N\), a strict new -determination of \(N^\ell\).
F Simulation and observational preorders coincide

This section is dedicated to the proof of Proposition 5.

F.1 Some preliminary results

Lemma 43. For all \( a \in \mathcal{N} \), for all ground terms \( t \), if \( a \not\in st(t) \) then \( a \neq t \).

Proof. Recall that \( = \) is closed under substitution of names by other terms. Let \( b \in \mathcal{N} \setminus \{a, st(t)\} \) and \( \sigma = \{b/a\} \). Thus, if \( a = t \) then \( a\sigma = t\sigma \) and \( b = t \) which would imply \( a = b \) which contradicts our hypothesis.

Lemma 44. Let \( (\mathcal{P}, \phi) \in SP_t \). For all \( \mathcal{P} \xrightarrow{T} \delta \mathcal{P}' \) by application of the rules (PAR) or (Nil), \( \mathcal{P} \approx_{obs}^R \mathcal{P}' \) and \( (\mathcal{P}, \phi) \approx_{bi}^N (\mathcal{P}', \phi) \).

Proof. Trivial.

Lemma 45. Let \( \mathcal{P}, \mathcal{P}' \in SP, \phi, \phi' \) be two frames and \( \rho \) be a renaming of names.

If one of the two following properties holds

- \( \text{img}(\rho) \cap \text{names}(\mathcal{P}, \mathcal{P}', \phi, \phi') = \emptyset \) and \( \text{dom}(\rho) \cap \text{img}(\rho) = \emptyset \)
- \( \text{fn}(\mathcal{P}, \mathcal{P}', \phi, \phi') \cap \mathcal{N}_{\text{priv}} = \text{dom}(\rho) \) and \( \text{img}(\rho) \subseteq \mathcal{N}_{\text{priv}} \)

then

- For all \( R \in \{\leq_{obs}^R, \approx_{obs}^R\} \), \( \mathcal{P} \rho R \mathcal{P}' \rho \) if and only if \( \mathcal{P} \rho R \mathcal{P}' \rho \)
- For all \( R \in \{\leq_{sim}^N, \approx_{bi}^N\} \), \( (\mathcal{P}, \phi) R (\mathcal{P}', \phi') \rho \) if and only if \( (\mathcal{P}, \phi) \rho R (\mathcal{P}', \phi') \rho \).

Proof. Direct by noticing that \( = \) is closed by under substitution of names by other terms, i.e. for all \( u, v \) such that \( \text{img}(\rho) \cap \text{names}(u, v), u = v \) if and only if \( \mu \rho = \nu \rho \).

Lemma 46. Let \( \mathcal{N} = (S_N, A_{ext} \sqcup A_{int}, \text{trans}_N) \) be a NPLTS, \( D, E \in D^{\leq 1}(S_N) \) and \( R \) be a relation over \( S_N \). If \( D \) has a finite support and \( D \hat{R} E \) then there exists a family of sub-distributions \( \{E_x\}_{x \in \text{supp}(D)} \) such that:

- for all \( y \in S_N \), \( \sum_{x \in \text{supp}(D)} E_x(y) \leq E(y) \)
- for all \( x \in \text{supp}(D) \), \( D(x) \cdot \delta_x \hat{R} E_x \) and \( \text{supp}(E_x) = R(\{x\}) \)

Proof. For this proof, we rely on [LSA14 Lemma 3.9] that states the following: Assume \( p_1, \ldots, p_n \leq 1 \) and a family \( \{r_I \leq 1\}_{I \subseteq \{1, \ldots, n\}} \). If for all \( I \subseteq \{1, \ldots, n\} \), \( \sum_{i \in I} p_i \leq \sum_{J \subseteq \{1, \ldots, n\} \setminus I \neq \emptyset} r_J \leq 1 \) then for all nonempty \( I \subseteq \{1, \ldots, n\} \), for all \( k \in I \), there exists \( s_{k,I} \leq 1 \) such that:

- for all \( I \subseteq \{1, \ldots, n\} \), \( \sum_{k \in I} s_{k,I} \leq 1 \)
Let us show the two desired properties:

For all \(k \in \{1, \ldots, n\}\), \(p_k = \sum_{I \subseteq \{1, \ldots, n\}} s_{k,I} \cdot r_I\).

We know that \(D\) has finite support. Hence, we can consider \(\text{supp}(D) = \{x_1, \ldots, x_n\}\) and so we define for all \(i \in \{1, \ldots, n\}\), \(p_i = D(x_i)\). For all \(I \subseteq \{1, \ldots, n\}\), let us define the sets \(X_I = \{x_i\}_{i \in I}\) and \(S_I\) such that:

\[
S_I = \{y \in S_N \mid \forall i \in \{1, \ldots, n\}, x_i R y \iff i \in I\}
\]

First, notice that \(S_I \cap S_J \neq \emptyset\) implies \(I = J\). Second, let us show that \(R(X_I) = \bigcup_{J \cap I \neq \emptyset} S_J\). Let \(y \in R(X_I)\). There exists \(i \in I\) such that \(x_i R y\). By taking the largest set \(J \subseteq \{1, \ldots, n\}\) such that for all \(j \in J, x_j R y\), we deduce that \(y \in S_J\). Notice that \(i \in J\) and so \(J \cap I \neq \emptyset\). On the other hand, for all \(J \subseteq \{1, \ldots, n\}\), for all \(y \in J\), if \(J \cap I \neq \emptyset\) then there exists \(i \in I\) such that \(x_i R y\) and so \(y \in R(X_I)\).

Thus, for all \(I \subseteq \{1, \ldots, n\}\), we define \(r_I = E(S_I)\). Recall that \(D \hat{R} E\) and \(S_I \cap S_J \neq \emptyset\) implies \(I = J\). Therefore we obtain that for all \(I \subseteq \{1, \ldots, n\}\), \(\sum_{i \in I} p_i = D(X_I) \leq E(R(X_I)) = \sum_{J \subseteq (1, \ldots, n)} E(S_J)\). Hence, for all nonempty \(I \subseteq \{1, \ldots, n\}\), for all \(k \in I\), there exists \(s_{k,I} \leq 1\) such that:

- for all \(I \subseteq \{1, \ldots, n\}\), \(\sum_{k \in I} s_{k,I} \leq 1\)
- for all \(k \in \{1, \ldots, n\}\), \(D(x_k) \leq \sum_{I \subseteq \{1, \ldots, n\}} s_{k,I} \cdot E(S_I)\).

For all \(k \in \{1, \ldots, k\}\), we define the sub-distribution \(E_{x_k} = \sum_{k \in I} s_{k,I} \cdot \sum_{y \in S_I} E(y) \cdot \delta_y\).

Let us show the two desired properties:

- Let \(y \in S_N\). Since \(S_I \cap S_J \neq \emptyset\) implies \(I = J\), there is a most one \(I \subseteq \{1, \ldots, n\}\) such that \(y \in S_I\). Hence \(\sum_{k=1}^n E_{x_k}(y) = \sum_{k=1}^n s_{k,I} \cdot E(y) = E(y) \cdot \sum_{k=1}^n s_{k,I} \leq E(y)\).
- for all \(k \in \{1, \ldots, n\}\), \(D(x_k) \leq \sum_{I \subseteq \{1, \ldots, n\}} s_{k,I} \cdot E(S_I)\). By taking \(X_{\{k\}}\), we already showed that \(R(\{x_k\}) = \bigcup_{I \subseteq \{1, \ldots, n\}} S_I\). Hence \(E_{x_k}(R(\{x_k\})) = \sum_{I \subseteq \{1, \ldots, n\}} E_{x_k}(S_I)\).

Since \(E_{x_k}(S_I) = s_{k,I} \cdot E(S_I)\), we conclude that \(D(x_k) \cdot E_{x_k} \hat{R} E_E\).

The following lemma is similar to the previous one but for equivalence relation.

**Lemma 47.** Let \(N = (S_N, \mathcal{A}_{\text{ext}} \cup \mathcal{A}_{\text{int, trans}})\) be a NPLTS, \(D, E \in D^{\leq 1}(S_N)\) and \(R\) be an equivalence relation over \(S_N\). If \(D \hat{R} E\) and \(E \hat{R} D\) then there exists a family of sub-distributions \(\{E_x\}_{x \in \text{supp}(D)}\) such that:

- \(\sum_{x \in \text{supp}(D)} E_x = E\)
- for all \(x \in \text{supp}(D), D(x) \cdot \delta_x \hat{R} E_x, E_x \hat{R} D(x) \cdot \delta_x\) and \(\text{supp}(E_x) = R(\{x\})\)
Lemma 48. Let \( E \) support but infinite scheme of \( D \). We have the following properties:

\[ \sum_{x \in \text{supp}(D)} E_x(y) = \sum_{x \in \text{supp}(D) \cap R(y))} \frac{E(y) \cdot D(x)}{E(R(\{x\}))} = E(y) \]

Moreover, for all \( x \in \text{supp}(D) \), \( E_x(R(\{x\})) = \sum_{y \in R(\{x\})} \frac{E(y) \cdot D(x)}{E(R(\{x\}))} = D(x) \) which allows us to conclude. \( \square \)

In the following, we say that \( D \xrightarrow{\tau} E \) has finite support if both \( D \) and \( E \) have finite support. Similarly, \( D \xrightarrow{\tau} E \) has finite support if all \( D_k \xrightarrow{\tau} D_{k+1} \) and \( D \) composing the infinite scheme of \( D \xrightarrow{\tau} E \) have also finite support. Note that it implies that \( D \) has finite support but \( E \) may still have infinite support.

Lemma 48. Let \( N \in \{N^1, N^2\} \). Let \( D_1, D_2, E_1, E_2 \in \mathcal{D}^{\leq 1}(S_N) \). Let \( R \) be a relation on \( S_N \). We have the following properties:

1. \( D_1 \xleftarrow{\tau} D_2, E_1 \xleftarrow{\tau} E_2 \) and \( (D_2 + E_2) \in \mathcal{D}^{\leq 1}(S_N) \) imply \( D_1 + E_1 \xleftarrow{\tau} D_2 + E_2 \).

2. \( \forall \alpha, D_1 \xleftarrow{\tau} D_2 \) and \( \alpha \cdot D_2 \in \mathcal{D}^{\leq 1}(S_N) \) imply \( \alpha \cdot D_1 \xleftarrow{\tau} \alpha \cdot D_2 \).

3. if \( D_1 + D_2 \xleftarrow{\tau} E \) and \( D_1, D_2 \) have finite support (resp. \( D_1 + D_2 \xleftarrow{\tau} E \), \( E \xleftarrow{\tau} D_1 + D_2 \) and \( R \) is an equivalence relation) then there exist \( E_1, E_2 \) such that \( E_1 + E_2 = E \) and \( D_i \xleftarrow{\tau} E_i \) (resp. \( D_i \xleftarrow{\tau} E_i \) and \( E_i \xleftarrow{\tau} D_i \)) for \( i = 1, 2 \).

4. if \( D_1 \xleftarrow{\tau} E_1 \) and \( \forall x \in \text{supp}(D_1) \), \( \forall y \in R(x), x \rightarrow F_1 \) implies \( \delta_y \xrightleftharpoons{\tau} F_2 \) and \( F_1 \xleftarrow{\tau} F_2 \) then
   
   (a) \( D_1 \xrightarrow{\tau} D_2 \) with finite support implies \( E_1 \xrightarrow{\tau} E_2 \) and \( D_2 \xleftarrow{\tau} E_2 \)

   (b) \( D_1 \xrightarrow{\tau} D_2 \) with finite support implies \( E_1 \xrightarrow{\tau} E_2 \) and \( D_2 \xleftarrow{\tau} E_2 \)

5. if \( R \) is an equivalence relation, \( D_1 \xleftarrow{\tau} E_1, E_1 \xleftarrow{\tau} D_1 \) and \( \forall x \in \text{supp}(D_1), \forall y \in R(x), x \rightarrow F_1 \) implies \( \delta_y \xrightleftharpoons{\tau} F_2 \), \( F_1 \xleftarrow{\tau} F_2 \) and \( F_2 \xleftarrow{\tau} F_1 \) then

   (a) \( D_1 \xrightarrow{\tau} D_2 \) implies \( E_1 \xrightarrow{\tau} E_2 \), \( D_2 \xleftarrow{\tau} E_2 \) and \( E_2 \xleftarrow{\tau} D_2 \)

   (b) \( D_1 \xrightarrow{\tau} D_2 \) and \( D_1 \in \mathcal{D}(S_N) \) imply \( E_1 \xrightarrow{\tau} E_2 \), \( D_2 \xleftarrow{\tau} E_2 \) and \( E_2 \xleftarrow{\tau} D_2 \)

Proof. Properties 1 and 2 are immediate, by definition of \( \hat{R} \).

For Property 3, let \( D = D_1 + D_2 \). If \( D \) has finite support, we can apply Lemma 46 to obtain the family sub-distributions \( \{E_x\}_{x \in \text{supp}(D)} \) such that

- for all \( y \in S_N \), \( \sum_{x \in \text{supp}(D)} E_x(y) \leq E(y) \)
- for all \( x \in \text{supp}(D) \), \( D(x) \cdot \delta_x \xrightarrow{\tau} E_x \) and \( \text{supp}(E_x) = R(\{x\}) \)
Let us define
\[
E'_1 = \sum_{x \in \text{supp}(D)} \frac{D_1(x)}{D(x)} \cdot E_x \quad \text{and} \quad E'_2 = \sum_{x \in \text{supp}(D)} \frac{D_2(x)}{D(x)} \cdot E_x
\]

We first prove that \(D_i \overset{\delta}{\sim} R \overset{\delta}{\sim} E'_i\) for \(i \in \{1, 2\}\). We know that for all \(x \in \text{supp}(D)\), \(D(x) \cdot \delta_x \overset{\delta}{\sim} R \overset{\delta}{\sim} E_x\). Hence, \(D_1(x) \cdot \delta_x \overset{\delta}{\sim} R \overset{\delta}{\sim} \frac{D_1(x) \cdot E_x}{D(x)}\) and so \(D_i \overset{\delta}{\sim} R \overset{\delta}{\sim} E'_i\). Finally, since for all \(y \in S_N\), \(\sum_{x \in \text{supp}(D)} E_x(y) \leq E(y)\) and \(\sum_{x \in \text{supp}(D)} E_x = E'_1 + E'_2\), there exists \(H\) such that \(E'_1 + E'_2 + H = E\), meaning that we can arbitrarily define \(E_1 = E'_1 + H_1\) and \(E_2 = E'_2 + H_2\) with \(H_1 + H_2 = H\) in order to conclude.

If \(R\) is an equivalence relation, \(D_1 + D_2 \overset{\delta}{\sim} R \overset{\delta}{\sim} E\) and \(E \overset{\delta}{\sim} D_1 + D_2\), then we can instead apply Lemma 47. Hence, for \(i \in \{1, 2\}\), for all \(x \in \text{supp}(D)\), \(D_i(x) \cdot \delta_x \overset{\delta}{\sim} R \overset{\delta}{\sim} \frac{D_i(x) \cdot E_x}{D(x)}\) and \(\frac{D_i(x) \cdot E_x}{D(x)} \overset{\delta}{\sim} R \overset{\delta}{\sim} D_i(x) \cdot \delta_x\). Defining \(E'_1\) and \(E'_2\) as above we obtain that \(D_i \overset{\delta}{\sim} R \overset{\delta}{\sim} E'_1\) and \(E'_i \overset{\delta}{\sim} R \overset{\delta}{\sim} D_i\). Moreover, by Lemma 47 for all \(y \in S_N\), \(\sum_{x \in \text{supp}(D)} E_x(y) = E(y)\) and so \(E'_1 + E'_2 = E\) which allows us to conclude.

We now prove Property 4a. Assume that \(D_1\) has finite support, \(D_1 \overset{\delta}{\sim} R \overset{\delta}{\sim} E_1\) and \(\forall x \in \text{supp}(D_1), \forall y \in R(x), x \rightarrow F_1\) implies \(\delta_y \overset{\tau}{\rightarrow} r \overset{\delta}{\sim} F_2\) and \(F_1 \overset{\delta}{\sim} R F_2\).

Let us assume that \(D_1 \overset{\tau}{\rightarrow} r \overset{\delta}{\sim} D_2\) and \(D_2\) has finite support. Thus there exists a countable set \(I\) and a multiset \(\{x_i\}_{i \in I}\) with for all \(i \in I, x_i \in S_N\) such that \(D_1 = \sum_{i \in I} \alpha_i \cdot \delta_{x_i}\), \(D_2 = \sum_{i \in I} \alpha_i \cdot F_i\) and for all \(i \in I, x_i \rightarrow F_i\). Note that by hypothesis, \(x_i \rightarrow F_i\) implies that \(F_i\) has finite support. Thus, \(D_2\) having finite support allows us to consider w.l.o.g that \(I\) is finite.

By applying Lemma 46, \(D_1 \overset{\delta}{\sim} R \overset{\delta}{\sim} E_1\) implies that there exists \(\{E_x\}_{x \in D_1}\) such that

- for all \(y \in S_N\), \(\sum_{x \in \text{supp}(D_1)} E_x(y) \leq E_1(y)\)
- for all \(x \in \text{supp}(D_1)\), \(D_1(x) \cdot \delta_x \overset{\delta}{\sim} R \overset{\delta}{\sim} E_x\) and \(\text{supp}(E_x) = R(\{x\})\)

Let \(i \in I\). Let \(y \in \text{supp}(E_{x_i})\). As \(y \in R(\{x_i\})\), by hypothesis, \(\delta_y \overset{\tau}{\rightarrow} r \overset{\delta}{\sim} F'_{i,y}\) and \(F_1 \overset{\delta}{\sim} R F'_{i,y}\). Thus by properties 1 and 2 \(\sum_{y \in \text{supp}(E_{x_i})} E_{x_i}(y) \cdot F_i \overset{\delta}{\sim} R \sum_{y \in \text{supp}(E_{x_i})} E_{x_i}(y) \cdot F'_{i,y}\) and hence \(F_i \overset{\delta}{\sim} R \sum_{y \in \text{supp}(E_{x_i})} \frac{E_{x_i}(y)}{E_{x_i}(R(\{x_i\}))} \cdot F'_{i,y}\). Since \(E_2 = \sum_{i \in I} \alpha_i \cdot F_i\), if we denote \(H_2 = \sum_{i \in I} \sum_{y \in \text{supp}(E_{x_i})} \frac{\alpha_i \cdot E_{x_i}(y)}{E_{x_i}(R(\{x_i\}))} \cdot F'_{i,y}\) then \(D_2 \overset{\delta}{\sim} R H_2\). Notice that:

\[
H_2 = \sum_{x \in \text{supp}(D_1)} \sum_{i \in I} \sum_{y \in \text{supp}(E_{x_i})} \frac{\alpha_i \cdot E_{x_i}(y)}{E_{x_i}(R(\{x_i\}))} \cdot F'_{i,y}
\]

\[
= \sum_{x \in \text{supp}(D_1)} \sum_{i \in I} \sum_{y \in \text{supp}(E_{x_i})} \frac{\alpha_i \cdot E_{x_i}(y)}{E_{x}(R(\{x\}))} \cdot F'_{i,y}
\]

\[
= \sum_{x \in \text{supp}(D_1)} \sum_{i \in I} E_{x_i}(y) \cdot F'_{i,y}
\]

104
Furthermore, we know that for all $i \in I$, for all $y \in R(\{x_i\})$, $\delta_y \Rightarrow \tau F'_{i,y}$, which implies that:

$$\sum_{x \in \text{supp}(D_1)} \sum_{i \in I} \frac{\alpha_i}{E_x(R(\{x\}))} \cdot E_x \Rightarrow \tau H_2$$

Note that the following holds:

$$\sum_{x \in \text{supp}(D_1)} \sum_{i \in I} \frac{\alpha_i}{E_x(R(\{x\}))} \cdot E_x = \sum_{x \in \text{supp}(D_1)} \frac{\sum_{i \in I} \alpha_i}{E_x(R(\{x\}))} \cdot E_x$$

Hence, if we denote $H_1 = \sum_{x \in \text{supp}(D_1)} \frac{D_1(x)}{E_x(R(\{x\}))} \cdot E_x$, we obtain $H_1 \Rightarrow \tau H_2$. Finally, we know that for all $x \in \text{supp}(D_1)$, $D_1(x) \leq E_x(R(\{x\}))$ and for all $y \in S_N$, $\sum_{x \in \text{supp}(D_1)} E_x(y) \leq E_1(y)$. Thus, there exists $H'_1$ such that $H_1 = H_1 + H'_1$. This allows us to deduce that $E_1 \Rightarrow \tau H_2 + H'_1$, and $D_2 \Rightarrow R(H_2 + H'_1)$. This completes the proof of property 4a.

The proof for property 5a follows very closely. Instead of using Lemma 46, we rely on Lemma 47. Note in this case, $D_1(x) = E_x(R(\{x\}))$ implying that $H_1 = \sum_{x \in \text{supp}(D_1)} E_x = E_1$. This allows us to conclude the proof of property 5a.

Let us now prove property 4b. Assume $D_1 \Rightarrow \tau D_2$ with internal finite support. By definition, there exists the following infinite scheme:

$$D_1 = D_0^\tau + D_0^\top$$
$$D_0^\tau \Rightarrow \tau, D_1^\tau + D_1^\top$$
$$\ldots$$
$$D_k^\tau \Rightarrow \tau, D_{k+1}^\tau + D_{k+1}^\top$$
$$\ldots$$

such that $D_2 = \sum_{k \in \mathbb{N}} D_k^\top$ and for all $k \in \mathbb{N}$, $D_k^\tau$ and $D_k^\top$ have finite support. We can easily show by induction that for all $k \in \mathbb{N}$,

- $E_1 \Rightarrow \tau E_k^\top + \sum_{i=0}^{k} E_i^\top$
- $D_k^\tau \Rightarrow R E_k^\tau$
- $\sum_{i=0}^{k} D_i^\top \Rightarrow R \sum_{i=0}^{k} E_i^\top$

105
In the base case $k = 0$, we can apply Property 3 to deduce that there exists $E_0^→$ and $E_0^\top$ such that $E = E_0^→ + E_0^\top$ and for all $\alpha \in \{\to, \top\}$, $D_0^α R E_0^α$.

In the inductive step $k > 0$, we know that $D_{k-1}^→ \overrightarrow{r} D_k^→ + D_k^\top$. By induction hypothesis, we know that there exist $E_1 \overrightarrow{r} E_{k-1}^→ + \sum_{i=0}^{k-1} E_i^\top$ such that $E_{k-1}^→ \hat{R} E_{k-1}^→$ and $\sum_{i=0}^{k-1} D_i^→ \hat{R} \sum_{i=0}^{k-1} E_i^\top$. By applying Property 4a, we obtain that $E_{k-1}^→ \overrightarrow{r}, E'$ such that $D_k^→ + D_k^\top \hat{R} E'$. By applying Property 3, we deduce that there exists $E_k^→, E''$ such that $E' = E_k^→ + E_k^\top, D_k^→ \hat{R} E_k^→$ and $D_k^\top \hat{R} E_k^\top$. Therefore, $\sum_{i=0}^{k} D_i^→ \hat{R} \sum_{i=0}^{k} E_i^\top$. Note that $E_{k-1}^→ \overrightarrow{r}, E_k^→ + E_k^\top$ implies $E_{k-1}^→ + \sum_{i=0}^{k-1} E_i^\top \overrightarrow{r} E_k^→ + \sum_{i=0}^{k} E_i^\top$. This concludes the inductive proof.

In the case where $\lim_{k→∞} E_k^→$ exists, we obtain that $E_1 \overrightarrow{r} \lim_{k→∞} E_k^→ + \sum_{k∈N} E_k^\top$ with $\sum_{k∈N} D_i^→ \hat{R} \sum_{k∈N} E_k^\top$. Hence, we conclude by defining $E_2 = \lim_{k→∞} E_k^→ + \sum_{k∈N} E_k^\top$ and so $D_2 \hat{R} E_2$. In the case $\lim_{k→∞} E_k^→$ does not exist, the semantics in Figures 2 and 10 imply that there exists $k_1 ∈ N$ and a set $S$ such that for all $k > k_1$, $\text{supp}(E_k^→) \cap \{(1) \cup \mathcal{P} \in \mathcal{SP}\} = S$ (resp. $\{(1), \phi\} \cup \mathcal{P} \in \mathcal{SP}\}$ when $N = N^\top$). Note that if we define $E_k^S = \sum_{x∈S} E_k^→(x) \cdot δ_x$ then $\lim_{k→∞} E_k^S$ exists. Moreover, thanks to Remark 7, we have that $E_k^→ \overrightarrow{r} E_k^S$. Thus, $E_1 \overrightarrow{r}, E_k^S + \sum_{i=0}^{k} E_i^\top$ and we can therefore conclude by defining $E_2 = \lim_{k→∞} E_k^S + \sum_{k∈N} E_k^\top$.

Finally, the proof of property 5b follows the same reasoning as property 4b. We prove by induction that for all $k ∈ N$,

- $E_1 \overrightarrow{r}, E_k^→ + \sum_{i=0}^{k} E_i^\top$
- $D_k^→ \hat{R} E_k^→$ and $E_k^→ \hat{R} D_k^→$
- $\sum_{i=0}^{k} E_i^\top \hat{R} \sum_{i=0}^{k} D_i^→$

The base case once again uses property 3 but the inductive step relies on property 5a rather than property 4a. To conclude the final proof, we know that $E_2 ∈ \mathcal{D}(\mathcal{S}_N)$. Thus, $\lim_{k→∞} E_k^→ = ∅$. This allows us to conclude by defining $E = \sum_{k∈N} E_k^→$. 

F.1.1 Properties on $\text{RProb}_{\mathcal{R}_Γ}(\mathcal{P}, ↓c)$

We define $\mathcal{R}_Γ^→ = \{(S, \text{corr, trans}) ∈ \mathcal{R}_Γ^\top | ∀s ∈ S, \text{trans}(s) = D \text{ implies } D \text{ has finite support}\}$. Let us define $\mathcal{P} \equiv \mathcal{P}'$ when there exists a renaming of names $\rho$ such that $f_n(\mathcal{P}) \cap N_{\text{priv}} = \text{dom}(\rho), \text{img}(\rho) ⊆ N_{\text{priv}}$ and $\mathcal{P} \rho = \mathcal{P}'$. Note that $\mathcal{P} \equiv \mathcal{P}'$ implies $\mathcal{P} ≈_{\text{obs}} \mathcal{P}'$ thanks to Lemma 45. Given a distribution $F$, we denote $F/ = \sum_{x\in\text{supp}(F)/} \sum_{y\in x} F(y) \cdot δ_x$.

Given two distributions $F$ and $F'$, we say that $F ≡ F'$ when $F/ = F'/ =$. 

**Lemma 49.** For all $F, H ∈ \mathcal{D}^{≤1}(N^\circ), F \overrightarrow{r}, H$ with $F$ having finite support implies that $F \overleftarrow{r}, H'$ with $H ≡ H'$ and $H'$ having finite support.

**Proof.** $F \overrightarrow{r}, H$ implies that there exists $F = \sum_{i∈I} α_i \cdot δ_x$, and $H = \sum_{i∈I} α_i \cdot D_i$ with
Notice that by Figure 2, given \( x \) the set \( \{ D \in D(N^o) / \equiv \} \) is finite. Indeed we only consider finite multisets of processes and the only rule that can generate an infinite number of successor is the rule (New) but they are all equivalent under \( \equiv \). Thus, by defining \( H' = \sum_{x \in \text{supp}(F)} \sum_{D \in D(N^o) / \equiv} (\sum_{i \in I} \alpha_i \cdot \delta_{x_i} \cdot D) \), the result holds.

Proof. We prove this property by induction on \( n \).

**Base case** \( n = 0 \): In such a case, \( \text{RProb}^\leq n(s, \text{corr}^{-1}(\downarrow c)) = 1 \) if \( P \in \downarrow c \) and 0 otherwise. Thus, we can define \( D = \delta_P \) to directly obtain that \( P \xrightarrow{\tau_r} D \) and \( \text{RProb}^\leq n(P, \downarrow c) \leq D(\downarrow c) \).

**Inductive step** \( n > 0 \): If \( s \in \text{corr}^{-1}(\downarrow c) \) or if \( \text{trans}(s) = \star \) then we can again define \( D = \delta_P \) to conclude. Let us now assume that \( s \not\in \text{corr}^{-1}(\downarrow c) \) and \( \text{trans}(s) = E \). In such a case,

\[
\text{RProb}^\leq n(s, \text{corr}^{-1}(\downarrow c)) = \sum_{u \in \text{supp}(E)} E(u) \cdot \text{RProb}^\leq n-1(u, \text{corr}^{-1}(\downarrow c))
\]

By definition of a resolution, \( \text{corr}(E) \in \text{conv}(\text{trans}_{N^o}(\text{corr}(s))) \) and so \( \delta_P \xrightarrow{\tau_r} \text{corr}(E) \). Note that when \( R \in \mathcal{R}^f_t \), we know that \( \text{corr}(E) \) has finite support.

By inductive hypothesis, we know that for all \( u \in \text{supp}(E) \), there exists \( D_u \in D(S^N) \) such that:

- \( \text{corr}(u) \xrightarrow{\tau_r} D_u \) (with finite support when \( R \in \mathcal{R}^f_t \))
- \( \text{RProb}^\leq n-1(\text{corr}(u), \text{corr}^{-1}(\downarrow c)) \leq D_u(\downarrow c) \)
Let us define $D = \sum_{u \in \text{supp}(E)} E(u) \cdot D_u$. Furthermore,

$$\sum_{Q \in \mathcal{V}} D(Q) = \sum_{Q \in \mathcal{V}} \left( \sum_{u \in \text{supp}(E)} E(u) \cdot D_u(Q) \right) = \sum_{u \in \text{supp}(E)} E(u) \cdot \sum_{Q \in \mathcal{V}} D_u(Q).$$

Thus, $\text{RProb}^{\mathbb{E}}_{\mathcal{R}}(s, \text{corr}^{-1}(\downarrow c)) \leq D(\downarrow c)$. Note that since for all $u \in \text{supp}(E)$, $\text{corr}(u) \xrightarrow{r} D_u$, we conclude that $\delta(\mathcal{P}) \xrightarrow{r} D$. Furthermore when $R \in \mathcal{R}_r^{\mathcal{I}}$, we know that $\text{corr}(E)$ has finite support and $\text{corr}(u) \xrightarrow{r} D_u$ has finite support. Hence so does $\delta(\mathcal{P}) \xrightarrow{r} D$. □

**Lemma 51.** For all $\mathcal{P} \in \mathcal{M}_{\mathcal{P}}$, for all $c \in \mathcal{N}_{\text{pub}}$,

- $\delta_{\mathcal{P}} \xrightarrow{r} E$ implies $E(\downarrow c) \leq \text{RProb}_{\mathcal{R}_{\mathcal{P}}}(\mathcal{P}, \downarrow c)$
- $\delta_{\mathcal{P}} \xrightarrow{r} E$ with finite support implies $E(\downarrow c) \leq \text{RProb}_{\mathcal{R}_{\mathcal{P}'}}(\mathcal{P}, \downarrow c)$

**Proof.** Thanks to Lemma 1, we know that $\delta_{\mathcal{P}} \xrightarrow{r} E$ implies that there exists $R = (S, \text{corr}, \text{trans}) \in \mathcal{R}_r^{\mathcal{I}}$, and $D', E' \in D^\leq(\mathcal{S})$ such that $\text{corr}(D') = \delta_{\mathcal{P}}$, $\text{corr}(E') = E$, $\text{supp}(E') \subseteq S_{\text{ext}}(R)$ and for all $u \in S_{\text{ext}}(R)$, $E'(u) = \sum_{s' \in S} D'(s') \cdot \text{RProb}_R(s', \{u\})$. Therefore,

$$E(\downarrow c) = \sum_{u \in S_{\text{ext}}(R)} E'(u) = \sum_{u \in S_{\text{ext}}(R)} \sum_{s' \in S} D'(s') \cdot \text{RProb}_R(s', \{u\}) \leq \sum_{s' \in S} D'(s') \cdot \text{RProb}_R(s', \text{corr}^{-1}(\downarrow c)) \leq \sup_{\text{corr}(s')=\mathcal{P}} \text{RProb}_R(s', \text{corr}^{-1}(\downarrow c)) \leq \text{RProb}_{\mathcal{R}_{\mathcal{P}'}}(\mathcal{P}, \downarrow c)$$

Note that one can easily check from the proof of Lemma 1 that the results carries over to $\delta_{\mathcal{P}} \xrightarrow{r} D$ with finite support, i.e. in that case, the resolution $R$ is in $\mathcal{R}_r^{\mathcal{I}}$, and so the rest of the proof follows. □

**Lemma 52.** For all $\mathcal{P} \in \mathcal{M}_{\mathcal{P}}$, for all $c \in \mathcal{N}_{\text{pub}}$,

$$\text{RProb}_{\mathcal{R}_{\mathcal{P}'}}(\mathcal{P}, \downarrow c) = \sup\{ D(\downarrow c) \mid \delta_{\mathcal{P}} \xrightarrow{r} D \} = \sup\{ D(\downarrow c) \mid \delta_{\mathcal{P}} \xrightarrow{r} D \text{ with finite support} \} = \text{RProb}_{\mathcal{R}_{\mathcal{P}'}}(\mathcal{P}, \downarrow c)$$
Proof. Consider the set \( S = \{ D(\downarrow c) \mid \delta_P \xRightarrow{\tau} D \} \). By Lemma 50, we know that \( \text{RProb}_{\mathcal{P}}(\mathcal{P}, \downarrow c) \leq \sup S \) and by Lemma 51, we have that \( \sup S \leq \text{RProb}_{\mathcal{P}}(\mathcal{P}, \downarrow c) \).

Similarly, if we denote \( S_f = \{ D(\downarrow c) \mid \delta_P \xRightarrow{\tau} D \text{ with finite support} \} \), we have \( \sup S_f = \text{RProb}_{\mathcal{P}_f}(\mathcal{P}, \downarrow c) \).

However, by Lemma 49, a simple induction on the infinite scheme building \( \delta_P \xRightarrow{\tau} D \) allows us to deduce that there exists \( D' \) such that \( \delta_P \xRightarrow{\tau} D' \) having finite support and such that \( D \equiv D' \). Since the equivalence \( \equiv \) only affects private names, we conclude that \( D(\downarrow c) = D'(\downarrow c) \) and so \( S = S_f \). \( \square \)

F.2 Restricted characterization of observational relations

Let us define the following equivalences:

**Definition 48.** The restricted observational preorders \( \leq^{r_0}_{\text{obs}}, \leq^{r_1}_{\text{obs}}, \leq^{r_2}_{\text{obs}} \) are the largest relation \( R \) on \( \mathcal{MP} \) such that \( \mathcal{P} R \mathcal{Q} \) implies:

- for all \( c \in \mathcal{N}_{pub} \), \( \text{RProb}_{\mathcal{P}}(\mathcal{P}, \downarrow c) \leq \text{RProb}_{\mathcal{Q}}(\mathcal{Q}, \downarrow c) \);
- Case \( \leq^{r_0}_{\text{obs}} \): if \( \mathcal{P} \xRightarrow{\tau} D \) then \( \mathcal{Q} \xRightarrow{\tau} E \) and \( D \xRightarrow{} \hat{R} \hat{E} \);
  - Case \( \leq^{r_1}_{\text{obs}} \): if \( \mathcal{P} \xRightarrow{\tau} D \) then \( \mathcal{Q} \xRightarrow{\tau} E \) and \( D \xRightarrow{} \hat{R} \hat{E} \);
  - Case \( \leq^{r_2}_{\text{obs}} \): if \( \mathcal{P} \xRightarrow{\tau} D \) with finite support then \( \mathcal{Q} \xRightarrow{} E \) and \( D \xRightarrow{} \hat{R} \hat{E} \);
- for all closed \( \text{Adv} \in \mathcal{MP} \) such that \( \text{fn}(\text{Adv}) \subseteq \mathcal{N}_{pub}, \{\text{Adv}\} \cup \mathcal{P} R \{\text{Adv}\} \cup \mathcal{Q} \)

The restricted observational equivalences \( \approx^{r_0}_{\text{obs}}, \approx^{r_1}_{\text{obs}}, \approx^{r_2}_{\text{obs}} \) are defined by additionally requesting \( R \) to be symmetric respectively in the cases of \( \leq^{r_0}_{\text{obs}}, \leq^{r_1}_{\text{obs}}, \leq^{r_2}_{\text{obs}} \). The restricted observational equivalence \( \approx^{r_0}_{\text{obs}}, \approx^{r_1}_{\text{obs}}, \approx^{r_2}_{\text{obs}} \) are defined respectively in the cases \( \leq^{r_0}_{\text{obs}}, \leq^{r_1}_{\text{obs}}, \leq^{r_2}_{\text{obs}} \) by additionally requesting that \( R \) to be symmetric and in the second bullet point, by requesting \( D \) to be a distribution and both \( D \xRightarrow{} \hat{R} \hat{E} \) and \( E \xRightarrow{} \hat{R} \hat{D} \) to hold.

**Lemma 53.** \( \leq_{\text{obs}}, \leq^{r_0}_{\text{obs}}, \leq^{r_1}_{\text{obs}} \) and \( \leq^{r_2}_{\text{obs}} \) coincide. \( \approx^{r_0}_{\text{obs}}, \approx^{r_1}_{\text{obs}}, \approx^{r_2}_{\text{obs}} \) coincide.

Proof. Let us start with the simple results: Notice that \( \leq^{r_0}_{\text{obs}} \) is in fact the same definition as \( \leq_{\text{obs}} \). In the definition of \( \approx^{r_1}_{\text{obs}} \), if \( \mathcal{P} \xRightarrow{\tau} D \) then \( \mathcal{Q} \xRightarrow{\tau} E \) and \( D \approx^{r_1}_{\text{obs}} E \). However, \( D \) is a distribution by Figure 2. Hence, \( D \approx^{r_1}_{\text{obs}} E \) also implies that \( E \) is a distribution. Since \( \approx^{r_1}_{\text{obs}} \) is symmetric, we deduce that \( E \approx^{r_1}_{\text{obs}} D \) and so \( \approx^{r_1}_{\text{obs}} \) and \( \approx^{r_1}_{\text{obs}} \) coincide.

We show that \( \leq^{r_0}_{\text{obs}} \) coincide with \( \leq^{r_2}_{\text{obs}} \). We trivially have \( \leq^{r_0}_{\text{obs}} \) implies \( \leq^{r_2}_{\text{obs}} \). Let us thus focus on the other implication. Assume that \( \mathcal{P} \leq^{r_2}_{\text{obs}} \mathcal{Q} \). Take \( \mathcal{P} \xRightarrow{\tau} D \). By definition,
there exists the following infinite scheme:

\[
\delta_P = D_0^+ + D_0^\top \\
D_0^+ \vdash \tau, D_1^+ + D_1^\top \\
\ldots \\
D_k^+ \vdash \tau, D_{k+1}^+ + D_{k+1}^\top \\
\ldots
\]

such that \( D = \sum_{k \in \mathbb{N}} D_k^\top. \)

Thanks to Lemma 49, a simple induction allows us to deduce that there exists \( \mathcal{P} \xrightarrow{\tau} \mathcal{P}, D' \) with finite support (\( D' \) may have infinite support) such that \( D \equiv D' \). By \( \mathcal{P} \preceq_{\text{obs}} \mathcal{Q} \), we deduce that there exists \( \mathcal{Q} \xrightarrow{\tau} \mathcal{Q} \) such that \( D' \preceq_{\text{obs}} \mathcal{Q} \). But \( D \equiv D' \) implies \( D \preceq_{\text{obs}} \mathcal{Q} \) which allows us to conclude.

The same proof allows us to show that \( \tau_0 \) and \( \tau_2 \) coincide. A minor modification of the proof also allows us to show that \( \tau_0 \) and \( \tau_2 \). Indeed, only the last sentence would change as we would obtain that there exists \( \mathcal{Q} \xrightarrow{\tau}, \mathcal{E} \) such that \( D' \preceq_{\text{obs}} \mathcal{E} \) and \( \mathcal{E} \preceq_{\text{obs}} D' \). But once again \( D \equiv D' \) and \( E \preceq_{\text{obs}} D' \) also imply \( E \approx_{\text{obs}} D \).

Let us now show that \( \approx_{\text{obs}} \) and \( \approx_{\text{obs}} \) coincide. We trivially have \( \approx_{\text{obs}} \) implies \( \approx_{\text{obs}} \) thus let us focus on the other implication. Assume that \( \mathcal{P} \preceq_{\text{obs}} \mathcal{Q} \). Hence \( \delta_P \succeq_{\text{obs}} \delta_Q \). Let \( \mathcal{P} \xrightarrow{\tau}, \mathcal{D} \) with finite support. By Property 4b of Lemma 48, we have that there exists \( \mathcal{E} \mathcal{E} \xrightarrow{\tau}, \mathcal{E} \) such that \( D \preceq_{\text{obs}} \mathcal{E} \). Hence we conclude the proof. Note that the same proof allows us to prove that \( \approx_{\text{obs}} \) and \( \approx_{\text{obs}} \) coincide.

The same proof where we apply Property 5b of Lemma 48 allows us to prove that \( \approx_{\text{obs}} \) and \( \approx_{\text{obs}} \) coincide.

\[\square\]

### F.3 Observational preorder implies simulation

For this direction of the equivalence, we need to represent as a process the frame of an extended process \((\mathcal{P}, \phi)\). Intuitively, for all \( \alpha \in \text{dom}(\phi) \), we will output on a public name \( c_{\alpha_k} \) the term \( \alpha_k \phi \). Formally, consider a frame \( \phi = \{ \alpha_0 \rightarrow t_1, \ldots, \alpha_n \rightarrow t_n \} \) and a sequence of public names \( \mathcal{N}_{ch} = [c_{\alpha_1} \in \mathcal{N}_{pub}]_{i=1}^n \), we define \( \mathcal{F}(\phi, \mathcal{N}_{ch}) = \{ \text{out}(c_{\alpha_i}, t_i), 0 \}_{i=1}^n \).

Given a frame \( \phi \) and a sequence \( \mathcal{N}_{ch} \), we define the following sets and functions:

- \( \mathcal{SP}_L(\phi, \mathcal{N}_{ch}) = \{ (\mathcal{P}, \phi) \in \mathcal{SP}_L \mid \text{fn}(\phi, \mathcal{P}) \cap \mathcal{N}_{ch} = \emptyset \wedge |\phi| = |\mathcal{N}_{ch}| \} \)
- \( \mathcal{SP}(\phi, \mathcal{N}_{ch}) = \{ \mathcal{P} \cup \mathcal{F}(\phi, \mathcal{N}_{ch}) \in \mathcal{SP} \mid \text{fn}(\phi, \mathcal{P}) \cap \mathcal{N}_{ch} = \emptyset \wedge |\phi| = |\mathcal{N}_{ch}| \} \)
- \( \pi_{\phi, \mathcal{N}_{ch}}^{e \rightarrow p} : \mathcal{SP}_L(\phi, \mathcal{N}_{ch}) \rightarrow \mathcal{SP}(\phi, \mathcal{N}_{ch}) \) such that \( \pi_{\phi, \mathcal{N}_{ch}}^{e \rightarrow p}((\mathcal{P}, \phi)) = \mathcal{P} \cup \mathcal{F}(\phi, \mathcal{N}_{ch}) \)
- \( \pi_{\phi, \mathcal{N}_{ch}}^{x \rightarrow e} : \mathcal{SP}(\phi, \mathcal{N}_{ch}) \rightarrow \mathcal{SP}_L(\phi, \mathcal{N}_{ch}) \) such that \( \pi_{\phi, \mathcal{N}_{ch}}^{x \rightarrow e}(\mathcal{P} \cup \mathcal{F}(\phi, \mathcal{N}_{ch})) = (\mathcal{P}, \phi) \)
To prove that observational preorder and equivalence imply simulation and bisimulation respectively, we define the relation $R_{op}$ with $op \in \{\leq, \approx\}$ on extended processes as follows: For all $(P, \phi), (P', \phi') \in SP_\ell,$ if and only if $(P, \phi) \in SP_\ell(\phi, N_{ch}), (P', \phi') \in SP_\ell(\phi', N_{ch})$ and $\pi_{\phi,N_{ch}}^{e\rightarrow p}((P, \phi)) \oplus \overrightarrow{R_{obs}} \pi_{\phi',N_{ch}}^{e\rightarrow p}((P', \phi'))$ for some $N_{ch}.$

Our definition of $R_{op}$ parametrized by $op$ will allow us to prove intermediate results that will hold both for the proof of observational preorder implying simulation and the proof of observational equivalence implying bisimulation.

**Lemma 54.** Let $N_{ch}$ be a sequence of public names and $\phi, \phi'$ two frames. Let $D, E \in D^{\leq_1}(\bigcup_\phi SP(\phi, N_{ch})).$ For all $op \in \{\leq, \approx\}, D \overrightarrow{op_{obs}} E$ implies $\sum_\phi \sum_{Q \in SP(\phi, N_{ch})} D(Q) \cdot \delta_{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q)} \overrightarrow{R_{op}} \sum_\phi \sum_{Q \in SP(\phi, N_{ch})} E(Q) \cdot \delta_{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q)}.$

**Proof.** Let $S \subseteq SP_\ell.$ Let $[S] = \{(P''', \phi'') \in SP_\ell \mid (P, \phi) \in S \land (P, \phi) \overrightarrow{R_{op}} (P'', \phi'')\}.$ Let $S_1 = \bigcup_\phi \pi_{\phi,N_{ch}}^{e\rightarrow p}((SP_\ell(\phi, N_{ch}) \cap S) \land [S] = \{Q' \in SP \mid Q \in S_1 \land Q \overrightarrow{op_{obs}} Q'\}.

Since $D \overrightarrow{op_{obs}} E,$ we know that $D(S_1) \leq E([S]).$ By definition of $S_1$ and since $D$ is a sub-distribution of $\bigcup_\phi SP(\phi, N_{ch}),$ we have $D(S_1) = \sum_\phi \sum_{Q, \phi \in SP(\phi, N_{ch})} D(Q) \cdot \delta_{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q)}.$

Moreover, we know by hypothesis that $E = \bigcup_\phi D^{\leq_1}(SP(\phi, N_{ch})).$ Hence, if we define $[S_2] = \{Q' \in SP(\phi, N_{ch}) \mid Q \in S_1 \land Q \overrightarrow{op_{obs}} Q' \land \text{frame } \phi \} \text{ then } E([S_1]) = E([S_2]).$ Therefore, by definition of $R_{op}$ and $[S_2],$ we deduce that $S_3 = \{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q) \mid Q \in [S_2]\} \subseteq [S].$ This allows us to deduce that $\sum_\phi \sum_{Q \in SP(\phi, N_{ch})} D(Q) \cdot \delta_{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q)}(S_1) = D(S_1) \leq E([S_1]) = \sum_\phi \sum_{Q \in SP(\phi, N_{ch})} E(Q) \cdot \delta_{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q)}(S_3) \leq \sum_\phi \sum_{Q \in SP(\phi, N_{ch})} E(Q) \cdot \delta_{\pi_{\phi,N_{ch}}^{e\rightarrow p}(Q)}([S]).$ \hfill \Box

**Corollary 7.** Let $N_{ch}$ be a sequence of public names and $\phi, \phi'$ two frames. Let $D \in D^{\leq_1}(SP(\phi, N_{ch})), E \in D^{\leq_1}(SP(\phi', N_{ch})).$ For all $op \in \{\leq, \approx\}, D \overrightarrow{op_{obs}} E$ implies $\pi_{\phi,N_{ch}}^{e\rightarrow p}(D) \overrightarrow{R_{op}} \pi_{\phi',N_{ch}}^{e\rightarrow p}(E).$

### F.3.1 $\tau$ transitions

**Lemma 55.** For all $op \in \{\leq, \approx\},$ for all $(P, \phi), (P', \phi') \in SP_\ell,$ if $(P, \phi) \overrightarrow{R_{op}} (P', \phi')$ and $(P, \phi) \overrightarrow{N^\tau} D$ then $(P', \phi') \overrightarrow{r_{N^\tau}} E$ and $D \overrightarrow{R_{op}} E$ (and $E \overrightarrow{R_{\approx}} D$ when $op = \approx$).

**Proof.** By Definition $\overrightarrow{10} D = (D', \phi)$ with $P \overrightarrow{\tau_{N^\tau}} D'.$ Since $(P, \phi) \overrightarrow{R_{op}} (P', \phi'),$ we know that $(P, \phi) \in SP_\ell(\phi, N_{ch}), (P', \phi') \in SP_\ell(\phi', N_{ch})$ and $P \cup F(\phi, N_{ch}) \overrightarrow{op_{obs}} P' \cup F(\phi', N_{ch})$ for some $N_{ch}.$

By Figure $2,$ we deduce that $P \cup F(\phi, N_{ch}) \overrightarrow{\tau_{N^\tau}} D_1 = \sum_{Q \in supp(D')} D'(Q) \cdot \delta_{Q \cup F(\phi, N_{ch})}$ and $P \cup F(\phi', N_{ch}) \overrightarrow{r_{N^\tau}} D_1.$ By Definition $\overrightarrow{18} P' \cup F(\phi', N_{ch}) \overrightarrow{r_{N^\tau}} E_1$ for some $E_1$ such that $D_1 \overrightarrow{op_{obs}} E_1$ and $E_1 \overrightarrow{R_{\approx}} D_1$ when $op = \approx$.

Let us analyse the support of $E_1.$ For all $Q \cup F(\phi', N_{ch}) \overrightarrow{\tau_{N^\tau}} F$ with $N_{ch} \cap fh(Q) = \phi,$ the only transition from Figure $2$ that may reduce an output from $F(\phi', N_{ch})$ is the rule (COMM). However that would imply that $Q = \{\text{tag}(u, x).Q\} \cup Q'$ with $c_{ax_i} = u$ for some $i \in \{\text{tag}(u, x).Q\} \cup Q'$ with $c_{ax_i} = u$ for some $i \in.$
Lemma 35 and a simple induction on the infinite scheme defining Lemma 56.

F.3.2 Static equivalence transitions

Lemma 56. For all \( op \in \{ \leq, \approx \} \), for all \( (P, \phi), (P', \phi') \in SP_{\ell} \), if \( (P, \phi) \xrightarrow{op} (P', \phi') \) and \( (P, \phi) \xrightarrow{\alpha} D \) with \( a \in \{ \xi \doteq \xi, \xi \neq \xi, \text{ax} \in, \text{ax} \notin \} \) then \( (P', \phi') \xrightarrow{\alpha} E \) and \( D \xrightarrow{op} E \) (and \( E \xrightarrow{\ell} D \) when \( op = \approx \)).

Proof. By definition of \( R_{op} \), we know that \( (P, \phi) \xrightarrow{op} (P', \phi') \) implies \( (P, \phi) \in SP_{\ell}(\phi, N_{ch}) \), \( (P', \phi') \in SP_{\ell}(\phi', N_{ch}) \) and \( \pi_{\phi, N_{ch}}^{\Rightarrow} P_{op}(\phi, \phi') \) for some \( N_{ch} \). Hence, \( |\phi| = |\phi'| \). Hence, for all \( a \in \{ \text{ax} \in, \text{ax} \notin \} \), \( (P, \phi) \xrightarrow{op} (P', \phi') \) implies \( (P', \phi') \xrightarrow{\alpha} \) for some \( N_{ch} \). This result holds.

Let us now focus on the case where \( a = \xi o \xi \) with \( o \in \{ \doteq, \neq \} \), \( n = |\phi| \) and assume that \( N_{ch} = [c_{ax_i}]_{i=1}^{n} \). We consider a public name \( ok \) such that \( ok \notin \text{fn}(P, P', \phi, \phi') \), some variables \( x_1, \ldots, x_n \) and the renaming \( \rho = \{ ax_i \}_{i=1}^{n} \). We build the following processes \( Test^\xi \) and \( Test^{\neq} \).

\[
\begin{align*}
Test^\xi &= \text{in}(c_{ax_1}, x_1). \ldots. \text{in}(c_{ax_n}, x_n). Test_{ij}^\xi \\
Test_{ij}^\xi &= \text{if } \xi \rho = \xi \rho \text{ then out}(ok, ok).0 \text{ else } 0 \\
Test^{\neq} &= \text{in}(c_{ax_1}, x_1). \ldots. \text{in}(c_{ax_n}, x_n). Test_{ij}^{\neq} \\
Test_{ij}^{\neq} &= \text{if } \xi \rho = \xi \rho \text{ then out}(ok, ok).0 
\end{align*}
\]

Since \( \xi, \xi' \) are recipes we know that \( \text{fn}(\xi, \xi') \subseteq N_{pub} \) and so \( \text{fn}(\text{Test}^\xi, \text{Test}^{\neq}) \subseteq N_{pub} \). Hence, by Definition 18, for all \( o \in \{ \doteq, \neq \} \), \( \{ \text{Test}^o \} \cup \pi_{\phi, N_{ch}}^{\Rightarrow} P_{op}(\phi, \phi') \) and \( \{ \text{Test}^o \} \cup \pi_{\phi, N_{ch}}^{\Rightarrow} P_{op}(\phi, \phi') \).

Let us denote \( P_T = \{ \text{Test}^o \} \cup \pi_{\phi, N_{ch}}^{\Rightarrow} P_{op}(\phi, \phi') \) and \( P'_T = \{ \text{Test}^o \} \cup \pi_{\phi', N_{ch}}^{\Rightarrow} P_{op}(\phi, \phi') \).

Notice that \( (P, \phi) \xrightarrow{op} (P', \phi') \) implies that \( \xi \phi o \xi' \phi \). Furthermore, \( \pi_{\phi, N_{ch}}^{\Rightarrow} P_{op}(\phi, \phi') = P \cup \{ \text{out}(c_{ax_1}, \text{ax}_i, 0.0) \}_{i=1}^{n} \). Hence, by successively applying the rule (COMM) on \( c_{ax_1}, \ldots, c_{ax_n} \), we deduce that \( P_T \xrightarrow{r, N_{\ell}} \delta_{P \cup \{ \text{Test}^o \}}(\sigma, \rho^{\neq} \phi) \) for some \( \sigma = \rho^{\neq} \phi \). Since \( \xi \phi o \xi \phi \) implies \( (\xi \rho)\rho^{-1} \phi \) we can apply the rule (THEN) when \( o = \doteq \) (ELSE) otherwise to obtain that \( P_T \xrightarrow{r, N_{\ell}} \delta_{P \cup \{ \text{Test}^o \}}(\sigma, \rho^{\neq} \phi) \). Thus, \( R_{Prob}(P_T, \doteq, ok) = 1 \).

Since \( P_T \xrightarrow{op} P_T' \), we deduce that \( R_{Prob}(P_T', \doteq, ok) = 1 \). Hence, there exists a resolution \( R = (S_r, corr_R, trans_r) \in R_r \) and \( s \in S_r \) such that \( corr_R(s) = P_T' \) and \( R_{Prob}(s, corr_R^{-1}(\doteq, ok)) > 0 \), meaning that there exists \( p \in N \) such that \( R_{Prob}(s, corr_R^{-1}(\doteq, ok)) > 0 \).
Let us define $A = \{ Q \cup \{ \text{out}(c_{ax}, ax \phi'), 0 \} \}_{i=1}^n \cup \{ \text{in}(c_{ax}, x_j) \ldots \text{in}(c_{ax}, x_n), \text{Test}_{ij}^{p} \sigma_j \} \mid j = 1 \ldots n + 1, (Q, \phi') \in \mathcal{SP}_{\ell}(\phi', \mathcal{N}_{ch}) \}$ where $\sigma_j = \{ ax_{\phi'/x_j} \}_{i=1}^{j-1}$. We first notice that $\mathcal{P}_T \in A$. Moreover, thanks to the fact that $(Q, \phi') \in \mathcal{SP}_{\ell}(\phi', \mathcal{N}_{ch})$ implies $c_{ax_i} \notin \text{fn}(Q)$ and thanks to Lemma 34, we can show that for all $Q' \in A$, $Q' \xrightarrow{N} F$ implies $\text{supp}(F) \subseteq A \cup \downarrow \text{ok}$.

This allows us to prove by induction on $p \in \mathbb{N}$ that for all $s \in \mathcal{S}_{R}$, $\text{RProb}_{R}^{x}(s, \text{corr}_{R}(\downarrow \text{ok})) > 0$ and $\text{corr}_{R}(s) \in A$ imply there exist $s_1, s_2 \in \mathcal{S}_{R}$ such that $\text{corr}_{R}(s_1) = Q' \cup \{ \text{Test}_{ij}^{p} \sigma_n \} \in A$, $\text{corr}_{R}(s_2) = Q' \cup \{ \text{out}(ok, ok), 0 \}$ and $s_2 \in \text{supp}(\text{trans}_{R}(s_1))$. For $s_2$ to be in $\text{supp}(\text{trans}_{R}(s_1))$, it implies that $Q' \cup \{ \text{out}(ok, ok), 0 \} \xrightarrow{N} \delta_{Q' \cup \{ \text{out}(ok, ok), 0 \}}$ by application of the rule (THEN) when $o$ is $\text{=}$. and the rule (ELSE) otherwise. Thus, $\xi \rho \sigma_n \alpha \rho \sigma_n$ holds and so $\xi \phi' \alpha \zeta \phi'$ holds.

We can therefore conclude that $(P', \phi') \xrightarrow{\text{R}_{op}} \delta_{(P', \phi')}$ and once again noticing that $(P, \phi) \xrightarrow{\text{R}_{op}} (P', \phi')$ implies $\delta_{(P, \phi)} \xrightarrow{\text{R}_{op}} \delta_{(P', \phi')}$; and that when $\text{op} = \approx$, $R_{\approx}$ is symmetric hence $(P', \phi') \approx R_{\approx} (P, \phi)$ also implying $\delta_{(P', \phi')} \xrightarrow{\text{R}_{\approx}} \delta_{(P, \phi)}$.

F.3.3 Input and output transitions

For input and output transitions, we will need to consider predicates on the relation $\to$ of a NPLTS. Given a predicate $\pi$, we will write $\pi(x \rightarrow D)$ when $\pi$ holds on $x \rightarrow D$. Given a logic formula $\phi$ with predicates as atoms, we denote by $(x \rightarrow D) \models \phi$ the natural satisfiability of $\phi$ by $x \rightarrow D$. Given $D \xrightarrow{\tau} E$ with $D = \sum_{i \in I} \alpha_i \cdot \delta_{x_i}$, $E = \sum_{i \in I} \alpha_i \cdot D_i$ and $x_i \rightarrow D_i$, we will write $(D \xrightarrow{\tau} E) \models \phi$ when for all $i \in I$, $\pi(x_i \rightarrow D_i) \models \phi$. Finally, given $D \xrightarrow{\tau} E$ with the infinite scheme $D = D_0 \rightarrow + D_0$ and $D_k \rightarrow + D_{k+1} + D_{k+1}$ for $k \in \mathbb{N}$, we write $(D \xrightarrow{\tau} E) \models \phi$ when for all $k \in \mathbb{N}$, $(D_k \rightarrow + D_{k+1} + D_{k+1}) \models \phi$.

For some of our proof, we will consider a measure function on states of our NPLTS associated with a total order. Formally, a measure for a NPLTS $N = (S_N, A_{\text{ext}} \sqcup A_{\text{int}}, \text{trans}_N)$ is a function $m : S_N \mapsto I$ for some set $I$ associated with a total order $<_m$ on $I$ that includes a minimal element for $<_m$. Given a sub-distribution $D \in \mathcal{D}^{\leq 1}(S_N)$, we denote $m(D) = \max \{ m(x) \mid x \in \text{supp}(D) \}$.

Lemma 57. Let $N = (S_N, A_{\text{ext}} \sqcup A_{\text{int}}, \text{trans}_N)$ be a NPLTS with $A_{\text{int}} = \{ \tau \}$. Let $\pi_1, \ldots, \pi_n$ be predicates on $\to$. For all $(\sum_{i=1}^{m_A} A_i \xrightarrow{\tau} \sum_{i=1}^{n_B} B_i) \models \pi_1 \lor \ldots \lor \pi_n$, there exist $\{ A_k^p \mid p = 1 \ldots n, k = 1 \ldots m_A \}$ and $\{ B_{k,i}^p \mid p = 1 \ldots n, k = 1 \ldots m_A, i = 1 \ldots m_B \}$ such that:

- for all $p \in \{ 1 \ldots n \}$, for all $k \in \{ 1 \ldots m_A \}$, $(A_k^p \xrightarrow{\tau} \sum_{i=1}^{m_B} B_{k,i}^p) \models \pi_p$
- for all $k \in \{ 1 \ldots m_A \}$, $A_k = \sum_{p=1}^{n} A_k^p$
- for all $i \in \{ 1 \ldots m_B \}$, $B_i = \sum_{p=1}^{n} B_{k,i}^p$

Proof. By definition of $(\sum_{i=1}^{m_A} A_i \xrightarrow{\tau} \sum_{i=1}^{n_B} B_i) \models \pi_1 \lor \ldots \lor \pi_n$, we know there exists a countable set $I$ and a multiset $\{ x_i \}_{i \in I}$ with for all $i \in I$, $x_i \in S_N$ such that $\sum_{i=1}^{m_A} A_i = \sum_{i \in I} \alpha_i \cdot \delta_{x_i}$, $\sum_{i=1}^{n_B} B_i = \sum_{i \in I} \alpha_i \cdot D_i$ and for all $i \in I$, $x_i \rightarrow D_i$ with $\pi_p(x_i \rightarrow D_i)$ for some $p \in \{ 1 \ldots n \}$. 113
With partition $I$ into $I_1 \cup \ldots \cup I_n$ such that for all $p \in \{1, \ldots, n\}$, for all $i \in I_p$, $\pi_p(x_i \to D_i)$. We define for all $k \in \{1, \ldots, m_A\}$, $p \in \{1, \ldots, n\}$, the sub-distribution $A_k^p = \sum_{i \in I_p} \frac{\alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i$. Hence, $(A_k^p \xrightarrow{\cdot} \sum_{i \in I_p} \frac{\alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i) \models \pi_p$. Therefore, for all $\ell \in \{1, \ldots, m_B\}$, we define $B_{k,\ell}^p$ such that for all $x \in \text{supp}(B_{\ell})$, $B_{k,\ell}^p(x) = \frac{B_{\ell}(x) \cdot \sum_{i=1}^{m_B} B_{i}(x)}{\sum_{i=1}^{m_B} B_{i}(x)} \cdot D_i(x)$. We directly have $\sum_{\ell=1}^{m_B} B_{k,\ell}^p = \sum_{i \in I_p} \frac{\alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i$ and so $(A_k^p \xrightarrow{\cdot} \sum_{\ell=1}^{m_B} B_{k,\ell}^p) \models \pi_p$.

Let us show that for all $k \in \{1, \ldots, m_A\}$, $A_k = \sum_{p=1}^{n} A_k^p$. Let $k \in \{1, \ldots, m_A\}$ and $x \in S_N$. We have $\sum_{p=1}^{n} A_k^p(x) = \sum_{p=1}^{n} A_k^p(x) = \sum_{p=1}^{n} \sum_{i \in I_p} \frac{\alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i(x) = \sum_{p=1}^{n} \sum_{i \in I_p} \alpha_i \cdot A_k(x_i) = \sum_{i \in I_p} \alpha_i \cdot D_i(x) = A_k(x)$.

Finally, let us show that for all $\ell \in \{1, \ldots, m_B\}$, $B_\ell = \sum_{p=1}^{n} B_{k,\ell}^p$. By definition, for all $x \in S_N$,

$$\sum_{p=1}^{n} \sum_{k=1}^{m_A} B_{k,\ell}^p(x) = \sum_{p=1}^{n} \sum_{k=1}^{m_A} \sum_{i \in I_p} \frac{\alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i(x)$$

\[= \frac{B_\ell(x) \cdot \sum_{i \in I_p} \alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i(x) \]

\[= \frac{B_\ell(x) \cdot \sum_{i \in I_p} \alpha_i \cdot A_k(x_i)}{\sum_{j=1}^{m_A} A_k(x_j)} \cdot D_i(x) \]

\[= \frac{B_\ell(x) \cdot \sum_{i \in I_p} \alpha_i \cdot D_i(x)}{\sum_{j=1}^{m_A} A_k(x_j)} \]

\[= B_\ell(x) \]

\[\square\]

**Lemma 58.** Let $N = (S_N, A_{\text{ext}} \sqcup A_{\text{int}}, \text{trans}_N)$ be a NPLTS with $A_{\text{int}} = \{\tau\}$. Let $\pi^c$, $\pi^s$, $\pi_{\text{all}}$ be three predicates on $\rightarrow$. Let $\pi^S$ be a predicate on $S_N$ such that for all $x \in S_N$, for all $D \in D^{\leq 1}(S_N)$,

- if $\pi^S(x)$ then $\pi^c(x \to D)$ or $\pi^s(x \to D)$
- if $\pi^c(x \to D)$ then $\pi^S(D)$

For all $D, E \in D^{\leq 1}(S_N)$, if $D \xrightarrow{\tau} E$ with the following infinite scheme

$D = D_0^c + D_0^T \quad \forall k \in \mathbb{N}, D_k^c \to_r D_{k+1}^c + D_{k+1}^T$

and $\pi^S(D_0^c)$ then there exists $A, A^T, B, B^T, C \in D^{\leq 1}(S_N)$ such that

$(D \xrightarrow{\tau} A + A^T) \models \pi^c \quad (A \xrightarrow{\tau} B + B^T) \models \pi^s \quad B \xrightarrow{\tau} C \quad E = C + B^T + A^T$
Moreover, if we assume the existence of measure function \( m \) on \( N \) such that for all \( x \in S_N \), for all \( D' \in D^{<1}(S_N) \), \( \pi_s^\rightarrow(x \rightarrow D') \) implies \( m(x) > m(D') \) and if \( m(A) \leq m(B + B^\top) \) then \((D \xrightarrow{\tau} r, E) \models \pi_e^\rightarrow\).

Finally, if \((D \xrightarrow{\tau} r, E) \models \pi_{alt} \) then \((D \xrightarrow{\tau} r, A + A^\top) \models \pi_{alt} \), \((A \xrightarrow{\tau} r, B + B^\top) \models \pi_{alt} \) and \((B \xrightarrow{\tau} r, C) \models \pi_{alt} \).

**Proof.** Consider the following property \( P(n) \) defined as: there exist \((\text{Save}_{i,k}^\rightarrow)^{n-1}_{k=0} \), \((\text{Save}_{i,k}^\top)^{n-1}_{k=0} \), \( k \in \{0, \ldots, n - i \} \), \((F_k^\rightarrow)^{n}_{k=0} \), \((F_k^\top)^{n}_{k=0} \), and \((F_k^\leftarrow)^{n-1}_{k=0} \) such that

1. for all \( k \in \{0, \ldots, n\} \), \( \pi^S(F_k^\rightarrow) \)
2. for all \( k \in \{0, \ldots, n\} \), \( D_k^\rightarrow = F_k^\rightarrow + \sum_{i=0}^{k-1} \text{Save}_{i,k-i}^\rightarrow \)
3. for all \( k \in \{0, \ldots, n\} \), \( D_k^\top = F_k^\top + \sum_{i=0}^{k-1} \text{Save}_{i,k-i}^\top \)
4. for all \( i \in \{0, \ldots, n-1\} \), for all \( k \in \{0, \ldots, n-i-1\} \), \( \text{Save}_{i,k}^\rightarrow \xrightarrow{r} \text{Save}_{i,k+1}^\rightarrow + \text{Save}_{i,k+1}^\top \)
5. for all \( i \in \{0, \ldots, n-1\} \), \( (\text{Save}_{i,0}^\rightarrow \xrightarrow{r} \text{Save}_{i,1}^\rightarrow + \text{Save}_{i,1}^\top) \models \pi_s^\rightarrow \)
6. for all \( k \in \{0, \ldots, n-1\} \), \( (F_k^\rightarrow \xrightarrow{r} F_{k+1}^\rightarrow + F_{k+1}^\top) \models \pi_e^\rightarrow \)
7. for all \( k \in \{0, \ldots, n-1\} \), \( F_k^\rightarrow = F_k^\rightarrow + \text{Save}_{k,0} \)

We show that for all \( n \in N \), \( P(n) \) holds by induction on \( n \).

**Base case** \( n = 0 \): By choosing \( F_0^\rightarrow = D_0^\rightarrow \) and \( F_0^\top = D_0^\top \), the result directly holds since by hypothesis \( \pi^S(D_0^\rightarrow) \) holds.

**Inductive step** \( n > 0 \): We have that by hypothesis that \( D_n^\rightarrow \xrightarrow{r} D_{n+1}^\rightarrow + D_{n+1}^\top \). Furthermore by inductive hypothesis, \( D_n^\rightarrow = F_n^\rightarrow + \sum_{i=0}^{n-1} \text{Save}_{i,n-i}^\rightarrow \).

By Lemma 57, we know that there exist some sub-distributions \( \text{Save}_{i,n-i+1}^\rightarrow \) and \( \text{Save}_{i,n-i+1}^\top \) for all \( i \in \{0, \ldots, n-1\} \) and \( H^\rightarrow \) and \( H^\top \) such that:

- \( F_n^\rightarrow \xrightarrow{r} H^\rightarrow + H^\top \)
- for all \( i \in \{0, \ldots, n-1\} \), \( \text{Save}_{i,n-i}^\rightarrow \xrightarrow{r} \text{Save}_{i,n-i+1}^\rightarrow + \text{Save}_{i,n-i+1}^\top \)
- for all \( \alpha \in \{\rightarrow, \top\} \), \( D_{n+1}^\alpha = H^\alpha + \sum_{i=0}^{n-1} \text{Save}^\alpha_{i,n-i+1} \)

By hypothesis, \( \pi^S(F_n^\rightarrow) \) hence \((F_n^\rightarrow \xrightarrow{r} H^\rightarrow + H^\top) \models \pi_e^\rightarrow \lor \pi_s^\rightarrow \). By Lemma 57 we deduce that there exists \( F_n^c, F_n^s, H^c\rightarrow, H^s\rightarrow, H^c\top, H^s\top \) such that:

- \( F_n^\rightarrow = F_n^c + F_n^s \)
- for all \( a \in \{s,c\} \), \((F_n^a \xrightarrow{r} H^a_\rightarrow + H^{a,\top}) \models \pi_a \)
- for all \( \alpha \in \{\rightarrow, \top\} \), \( H^\alpha = H^{c,\alpha} + H^{s,\alpha} \)

115
We define the sub-distributions \( F_{n}^c = F_{n}^{c^*}, F_{n+1}^{c^*} = H^{c^*}, F_{n+1}^T = H^{c^* T} \), \( \text{Save}_{n,0}^* = F_{n}^{c^*}, \text{Save}_{n,1}^* = H^{c^* T} \) and \( \text{Save}_{n,1}^T = H^{s^* T} \).

We prove the different properties of \( P(n+1) \) by applying the inductive hypothesis on \( P(n) \) and the following reasoning: Since \( (F_{n}^c, H^{c^* T} + H^{c^* T}) \models \pi_{c^*}^\top \), we know from our hypotheses on \( \pi_{c^*}^\top \) and \( \pi_{S}^{\top} \) that \( \pi_{S}(F_{n+1}^{c^*}) \) holds, hence property 1 holds. Moreover, \( D_{n+1} = H^{c^* T} \bigcup_{i=0}^{n-1} \text{Save}_{i,n-i+1}^{c^*} \), and so \( D_{n+1} = F_{n+1}^{c^*} + \text{Save}_{n,1}^{c^*} + \bigcup_{i=0}^{n-1} \text{Save}_{i,n-i+1}^{c^*} = F_{n+1}^{c^*} + \bigcup_{i=0}^{n} \text{Save}_{i,n-i+1}^{c^*} \), hence hence property 2 holds. Similarly, \( D_{n+1} = H^{c^* T} \bigcup_{i=0}^{n-1} \text{Save}_{i,n-i+1}^{c^*} \) and so \( D_{n+1} = F_{n+1}^{c^*} + \text{Save}_{n,1}^{c^*} + \bigcup_{i=0}^{n} \text{Save}_{i,n-i+1}^{c^*} = F_{n+1}^{c^*} + \bigcup_{i=0}^{n} \text{Save}_{i,n-i+1}^{c^*} \), hence hence property 3 holds. Notice that we showed \( (F_{n}^c, H^{c^* T} + H^{c^* T}) \models \pi_{c^*}^\top \). Thus, \( (\text{Save}_{n,0}^* \rightarrow \text{Save}_{n,1}^* + \text{Save}_{n,1}^{T}) \models \pi_{s}^\top \) and so property 5 holds. We already proved that for all \( i \in \{0, \ldots, n-1\} \), \( \text{Save}_{i,n-i}^* \rightarrow \text{Save}_{i,n-i+1}^* + \text{Save}_{i,n-i+1}^{T} \). Since \( (\text{Save}_{n,0}^* \rightarrow \text{Save}_{n,1}^* + \text{Save}_{n,1}^{T}) \models \pi_{s}^\top \), we directly deduce that for all \( i \in \{0, \ldots, n\} \), \( \text{Save}_{i,n-i}^* \rightarrow \text{Save}_{i,n-i+1}^* + \text{Save}_{i,n-i+1}^{T} \) and so property 4 holds. Similarly, we proved that \( (F_{n}^c \rightarrow H^{c^* T} + H^{c^* T}) \models \pi_{c^*}^\top \) meaning that \( (F_{n}^c \rightarrow F_{n+1}^{c^*} + F_{n+1}^{c^* T}) \models \pi_{c^*}^\top \), thus property 6 holds. Finally, we know that \( F_{n}^c = F_{n}^{c^*} + F_{n}^{s^*} \) implying \( F_{n} = F_{n}^{c^*} + \text{Save}_{n,0}^* \). Therefore, property 7 holds. This conclude the proof of property 7 for all \( n \in \mathbb{N} \).

To complete the main proof, notice that \( F_{n}^c + \text{Save}_{n,0}^* = D_{n}^c \), \( F_{0}^c = D_{0}^c \) and for all \( k \in \mathbb{N} \), \( F_{k}^c \rightarrow F_{k+1}^c \) imply that \( D \rightarrow \sum_{i \in \mathbb{N}} \text{Save}_{i,0}^* + F_{i}^c \). Notice that for all \( k \in \mathbb{N} \), we already proved that \( (F_{k}^c \rightarrow F_{k+1}^c + (\text{Save}_{k+1}^*, \text{Save}_{k+1}^c + F_{k+1}^c)) \models \pi_{c^*}^\top \). By denoting \( A = \sum_{k \in \mathbb{N}} \text{Save}_{k,0}^* \) and \( A^c = \sum_{k \in \mathbb{N}} F_{k}^c \), we deduce that \( (D \rightarrow \sum_{i \in \mathbb{N}} \text{Save}_{i,1}^* + \text{Save}_{i,1}^{T}) \models \pi_{s}^\top \). By denoting \( B = \sum_{i \in \mathbb{N}} \text{Save}_{i,1}^* \) and \( B^c = \sum_{i \in \mathbb{N}} \text{Save}_{i,1}^{T} \), we deduce that \( (A \rightarrow B + B^c) \models \pi_{s}^\top \).

Finally let us build \( B_{0}^c = \sum_{i \in \mathbb{N}} \text{Save}_{i,1}^* \) and \( B_{0}^c = \emptyset \). We know from the property \( \forall n \in \mathbb{N}.P(n) \) that for all \( i \in \mathbb{N} \), for all \( k > 0 \), \( \text{Save}_{i,k}^* \rightarrow \text{Save}_{i,k+1}^* + \text{Save}_{i,k+1}^{T} \). Hence for all \( k > 0 \), \( \sum_{i \in \mathbb{N}} \text{Save}_{i,k}^* \rightarrow \sum_{i \in \mathbb{N}} \text{Save}_{i,k+1}^* + \sum_{i \in \mathbb{N}} \text{Save}_{i,k+1}^{T} \). Hence, if we define for all \( k > 0 \), for all \( \alpha \in \{\rightarrow, \top\} \), \( B_{k}^{\alpha} = \sum_{i \in \mathbb{N}} \text{Save}_{i,k+1}^{\alpha} \) we deduce that for all \( n \in \mathbb{N} \), \( B_{k}^{\alpha} \rightarrow (B_{k+1}^{\alpha} + B_{k+1}^{T}) \). Therefore, by denoting \( C = \sum_{k>0} \sum_{i \in \mathbb{N}} \text{Save}_{i,k+1}^{T} \), we obtain \( B \rightarrow C \). It remains to prove that \( E = A^c + B^c + C : A^c + B^c + C = \sum_{k \in \mathbb{N}} F_{k}^c + \sum_{k \in \mathbb{N}} \text{Save}_{i,1}^{c} + \sum_{k>0} \sum_{i \in \mathbb{N}} \text{Save}_{i,1}^{c} = \sum_{k \in \mathbb{N}} D_{k}^{c} = E \) (see property 3 of the property \( P \)). This allows us to conclude that \( E = A^c + B^c + C \).

Let us now assume that \( m(A) \leq m(B + B^c) \). We proved that \( (A \rightarrow B + B^c) \models \pi_{s}^\top \). Thus, \( m(A) > m(B + B^c) \) and so \( A = B = B^c = \emptyset \). Since \( B \rightarrow C \), we deduce that \( C = \emptyset \). With \( E = C + B^c + A^c \) and \( (D \rightarrow \sum_{i \in \mathbb{N}} \text{Save}_{i,1}^* + \text{Save}_{i,1}^{T}) \models \pi_{c^*}^\top \), we conclude that \( (D \rightarrow E) \models \pi_{c^*}^\top \). \( \square \)

**Lemma 59.** Let \( N = (S_N, A_{ext} \cup A_{int}, \text{trans}_N) \) be a NPLTS with \( A_{int} = \{\tau\} \). Let \( \pi_1, \pi_2 \) be two predicates on \( \rightarrow \) and \( \pi_1^S, \pi_2^S \) be two predicates on \( S_N \) such that for all \( x \in S_N \), for all \( D \in D^{\leq 1}(S_N) \)
• $\pi^S(x) \land \pi^S_2(x)$ does not hold;
• for all $i \in \{1, 2\}$, $\pi^S_i(x)$ if $\pi_i(x \to D)$
• for all $i \in \{1, 2\}, \pi_i(x \to D)$ implies $\pi^S_i(D)$.

For all $A_1, A_2, B \in D^\leq(S_N)$, if

$$A_1 + A_2 \xrightarrow{\tau} B \quad \pi^S_1(A_1) \quad \pi^S_2(A_2)$$

then there exist $B_1, B_2 \in D^\leq(S_N)$ such that $B = B_1 + B_2$ and for all $i \in \{1, 2\}, \pi^S(B_i)$ and $(A_i \xrightarrow{\tau} B_i) \models \pi_i$.

Proof. By definition of $A_1 + A_2 \xrightarrow{\tau} B$, we know there exists an infinite scheme $A_1 + A_2 = C^\tau_0 + C^\tau_0$ and for all $k \in \mathbb{N}$, $C^\tau_k \xrightarrow{\tau} C^\tau_{k+1} + C^\tau_{k+1}$ and $B = \sum_{k \in \mathbb{N}} C^\tau_k$.

Let us consider the property $P(n)$ defined as: there exist $\{C^\alpha_{k,i} \mid \alpha \in \{\neg, \top\}, i \in \{1, 2\}, k \in \{0, \ldots, n\}\}$ such that

1. for all $\alpha \in \{\neg, \top\}$, for all $k \in \{0, \ldots, n\}$, $C^\alpha_{k,1} + C^\alpha_{k,2} = C^\alpha_k$
2. for all $\alpha \in \{\neg, \top\}$, for all $k \in \{0, \ldots, n\}$, for all $i \in \{1, 2\}$, $\pi^S_i(C^\alpha_{k,i})$ holds.
3. for all $k \in \{0, \ldots, n-1\}$, for all $i \in \{1, 2\}$, $(C^\tau_{k,i} \xrightarrow{\tau} C^\tau_{k+1,i} + C^\tau_{k+1,i}) \models \pi_i$.

We prove $P(n)$ for all $n \in \mathbb{N}$ by induction on $n$.

Base case $n = 0$: Since for all $x \in S_N$, $\pi^S(x) \land \pi^S_2(x)$ and $\pi_i(A_i)$ for $i = 1, 2$, we deduce that there exists $C^\tau_{0,1}, C^\tau_{0,2}, C^\top_{0,1}, C^\top_{0,2}$ such that for all $\alpha \in \{\neg, \top\}$, $C^\alpha_{0,1} + C^\alpha_{0,2} = C^\alpha_0$, and $\pi^S_i(C^\alpha_{0,i})$ for $i = 1, 2$.

Inductive step $n > 0$: By our inductive hypothesis, we know that $C^\tau_n = C^\tau_{n,1} + C^\tau_{n,2}$ with $\pi^S_1(C^\alpha_{n,1})$ and $\pi^S_2(C^\alpha_{n,2})$. Furthermore, $C^\tau_k \xrightarrow{\tau} C^\tau_{k+1} + C^\tau_{k+1}$.

By hypothesis on $\pi_1$ and $\pi_2$ and the fact that both $\pi^S_1(C^\alpha_{n,1})$ and $\pi^S_2(C^\alpha_{n,2})$ hold, we deduce that $(C^\tau_{n-1,i} \xrightarrow{\tau} C^\tau_{n+1,i} + C^\tau_{n+1,i}) \models \pi_1 \lor \pi_2$. By Lemma [57] we deduce that there exists $\{C^\tau_{n,k} \mid p = 1, 2 \land k = 1, 2\}$ and $\{C^\alpha_{n,p} \mid \alpha \in \{\neg, \top\}, k \in \{1, 2\}, p \in \{1, 2\}\}$ such that:

• for all $p = 1, 2$, for all $k = 1, 2$, $(C^\tau_{n,k} \xrightarrow{\tau} C^\tau_{n+1,k} + C^\tau_{n+1,k}) \models \pi_p$
• for all $k = 1, 2$, $C^\tau_{n,k} = C^\tau_{n,k} + C^\tau_{n+1,k}$
• for all $\alpha \in \{\neg, \top\}$, $C^\alpha_{n+1} = \sum_{p=1}^2 \sum_{k=1}^2 C^\alpha_{n+1,k}$

By hypothesis on $\pi_1$ and $\pi_2$, for all $p = 1, 2$, for all $k = 1, 2$, $(C^\tau_{n,k} \xrightarrow{\tau} C^\tau_{n+1,k} + C^\tau_{n+1,k}) \models \pi_p$ implies $\pi^S_p(C^\tau_{n,k})$ holds. However, $\pi^S_p(C^\tau_{n,1})$ implies that for all $x \in \text{supp}(C^\tau_{n,1})$, $\neg \pi^S_2(x)$. Hence $\neg \pi^S_2(C^\tau_{n,1})$ and so $C^\tau_{n,1} = \emptyset$. Similarly, we deduce that $C^\tau_{n,2} = \emptyset$. Both of them imply that $C^\tau_{n+1,2} = C^\tau_{n+1,2} = C^\tau_{n+1,1} = C^\tau_{n+1,1} = \emptyset$. To summarize, we have that for all $p \in \{1, 2\}$,
\begin{itemize}
  \item \((C_{n,p}^{r} \rightarrow r C_{n+1,p}^{r} + C_{n+1,p}^{\tau}) \models \pi_p\)
  \item \(C_{n,p} \rightarrow r C_{n,p}^{\tau}\)
  \item for all \(\alpha \in \{\perp, \rightarrow\}, C_{n+1}^{\alpha} = C_{n+1,1}^{\alpha} + C_{n+1,2}^{\alpha}.
\end{itemize}

We conclude by defining \(C_{n+1,i}^{\alpha} = C_{n+1,i}^{\alpha} + C_{n+1,i}^{\tau}\) for \(i \in \{1,2\}\) and \(\alpha \in \{\rightarrow, \top\}\).

Since \(P(n)\) holds for all \(n \in \mathbb{N}\), we directly obtain that for all \(i \in \{1,2\}\), \((C_{0,i}^{r} + C_{0,i}^{\tau} \rightarrow r \sum_{k \in \mathbb{N}} C_{T,i}^{T,k}) \models \pi_i\) with \(\pi_i^\tau(\sum_{k \in \mathbb{N}} C_{T,i}^{T,k})\). With \(B = \sum_{k \in \mathbb{N}} C_{k}^{T} = \sum_{k \in \mathbb{N}} C_{k,1}^{T} + \sum_{k \in \mathbb{N}} C_{k,2}^{T}\), we conclude by taking \(B_i = \sum_{k \in \mathbb{N}} C_{k,i}^{T}\) for \(i \in \{1,2\}\).

\section*{Lemma 60.}

Let \(N = (S_N, A_{ext} \cup A_{int}, \text{trans}_N)\) and \(N' = (S_{N'}, A'_{ext} \cup A'_{int}, \text{trans}_{N'})\) be two NPLTS with \(A_{int} = A'_{int} = \{\tau\}\). Let \(f : S_N \rightarrow S'_{N'}\). Let \(\pi_{all}\) and \(\pi_c\) two predicates on \(\rightarrow_N\). Let \(\pi^S\) be a predicate.

If for all \(A, B \in D^{\leq 1}(S_N)\), for all \(D \in D(S_N)\), for all \(x \in S_N\),
\begin{itemize}
  \item if \(\pi_c^\tau(x \rightarrow_N D)\) then \(\pi^S(D)\)
  \item \((A \rightarrow r_N B) \models \neg \pi_c \land \pi_{all}\) implies \(f(A) = f(B)\) and either \(A = \varnothing\) or \(m(A) > m(B)\)
  \item \((A \rightarrow r_N B) \models \pi_c \land \pi_{all}\) implies \(f(A) \rightarrow r_{N'} f(B)\) and \(m(A) = m(B)\)
\end{itemize}

then for all \((D \rightarrow r_N E) \models \pi_{all}\), we have \(f(D) \rightarrow r_{N'} f(E)\).

\textbf{Proof.} Consider three sub-distributions \(A, A_1^\top, A_2^\top\) such that \((D \rightarrow r_N A + A_1^\top) \models \pi_{all}
E = A_2^\top + A_2^\top\), \((A \rightarrow r_N A_2^\top) \models \pi_{all}\) and \((A \rightarrow r_N A_1^\top) \models \pi_{all}\). Such sub-distributions exist since we can take \(A = D, A_1^\top = \varnothing\) and \(A_2^\top = E\). Let us take \(A, A_1^\top, A_2^\top\) a maximal for the following order \((A, A_1^\top, A_2^\top) < (B, B_1^\top, B_2^\top)\) when \(A_1^\top \prec_D B_1^\top\) or \(A_1^\top = B_1^\top\) and \(m(A) \succ m(B)\) (where \(\prec_D\) stands for the inclusion of distribution, i.e. \(D \prec_D D'\) if for all \(x, D(x) < D'(x)\).

By applying Lemma 58 on \((A \rightarrow r_N A_1^\top) \models \pi_{all}\) (with \(\pi_c^\tau = \pi_c\) and \(\pi_a^\tau = \neg \pi_c\), we deduce that there exist \(\bar{A}^\top, \bar{A}_1^\top, \bar{B}^\top, \bar{B}_1^\top, \bar{C}\) such that:

\[(A \rightarrow r_N \bar{A}^\top + \bar{A}_1^\top) \models \pi_c \land \pi_{all} \quad (\bar{A}^\top \rightarrow r_N \bar{B}^\top + \bar{B}_1^\top) \models \neg \pi_c \land \pi_{all} \quad (\bar{B}^\top \rightarrow r_N \bar{C}\) \models \pi_{all}\]

with \(A_2^\top = \bar{C} + \bar{B}_1^\top + \bar{A}_1^\top\). Let us define \(\bar{A} = \bar{B}^\top, \bar{A}_1 = \bar{A}^\top + \bar{B}^\top + A_1^\top\) and \(\bar{A}_2 = \bar{C}\). We show the following properties:

\begin{itemize}
  \item \(f(D) \rightarrow r_{N'} f(\bar{A}) + f(\bar{A}_1^\top)\): We know that \((A \rightarrow r_N \bar{A}^\top + \bar{A}_1^\top) \models \pi_c \land \pi_{all}\). By hypothesis on \(\pi_c \land \pi_{all}\), we deduce that \(f(A) \rightarrow r_{N'} f(\bar{A}^\top) + f(\bar{A}_1^\top)\). Hence \(f(A) + f(A_1^\top) \rightarrow r_{N'} f(\bar{A}^\top) + f(\bar{A}_1^\top)\). By hypothesis on \(\neg \pi_c \land \pi_{all}\), we deduce from \((\bar{A}^\top \rightarrow r_N \bar{B}^\top + \bar{B}_1^\top) \models \neg \pi_c \land \pi_{all}\) that \(f(\bar{A}^\top) = f(\bar{B}^\top) + f(\bar{B}_1^\top)\). Hence \(f(A) + f(A_1^\top) \rightarrow r_{N'} f(\bar{A}) + f(\bar{A}_1^\top)\) and so \(f(D) \rightarrow r_{N'} f(\bar{A}) + f(\bar{A}_1^\top)\).
\end{itemize}
\( (D \xrightarrow{\tau,r,N} \bar{A} + \bar{A}_1^\top) \models \pi_{all}: (A \xrightarrow{\tau,r,N} \bar{A} + \bar{A}_1^\top) \models \pi_c \land \pi_{all} \) implies that \( (A + A_1^\top \xrightarrow{\tau,r,N} \bar{A} + \bar{A}_1^\top) \models \pi_c \land \pi_{all} \). Furthermore, \( (\bar{A} + \bar{A}_1^\top \xrightarrow{\tau,r,N} \bar{B} + \bar{B}_1^\top) \models \pi_{all} \). With \( \bar{B} + \bar{B}_1^\top + \bar{A} + A_1^\top = \bar{A} + \bar{A}_1^\top \) and the fact that \( (D \xrightarrow{\tau,r,N} A + A_1^\top) \models \pi_{all} \), we conclude that \( (D \xrightarrow{\tau,r,N} \bar{A} + \bar{A}_1^\top) \models \pi_{all} \).

- \( (A \xrightarrow{\tau,r,N} \bar{A}_2^\top) \models \pi_c \lor \pi_s \): Notice that \( \bar{A}_2^\top = \bar{C} \) and \( \bar{A} = \bar{B}^\top \) hence the result holds.

- \( \bar{A}_1^\top + \bar{A}_2^\top = \bar{A} + \bar{B}^\top + A_1^\top + \bar{C} = A_2^\top + A_1^\top = E. \)

Since we selected \( A, A_1^\top, A_2^\top \) as a maximal, then \( (A, A_1^\top, A_2^\top) \not\subseteq (\bar{A}, \bar{A}_1^\top, \bar{A}_2^\top) \). Since \( \bar{A}_1^\top = \bar{A} + \bar{B}^\top + A_1^\top \geq A_1^\top \), we deduce that \( \bar{A} = \bar{B}^\top = \emptyset \). Therefore, \( (A, A_1^\top, A_2^\top) \not\subseteq (\bar{A}, \bar{A}_1^\top, \bar{A}_2^\top) \) implies \( m(A) \leq m(\bar{A}) \).

Note that by hypothesis on \( \pi_c \land \pi_{all} \), \( (A \xrightarrow{\tau,r,N} \bar{A} + \bar{A}_1^\top) \models \pi_c \land \pi_{all} \) implies that \( m(A) = m(\bar{A}) \) and so \( m(A) \leq m(A) = m(\bar{A}) \). By Lemma 58 we obtain that \( (A \xrightarrow{\tau,r,N} A_1^\top) \models \pi_c \land \pi_{all} \) and so \( (A + A_1^\top \xrightarrow{\tau,r,N} E) \models \pi_c \land \pi_{all} \). By hypothesis on \( \pi_c \land \pi_{all} \), we conclude that \( f(A) + f(A_1^\top) \xrightarrow{\tau,r,N} f(E) \) and so \( f(D) \xrightarrow{r,N} f(E) \).

**Lemma 61.** For all \( op \in \{\leq, \approx\} \), for all \( a \in \{in(\xi, \zeta), out(\xi, \phi) \mid \xi, \zeta \text{ recipes}, \phi \in \mathcal{AX}\} \), for all \( (P, \phi), (P', \phi') \in \mathcal{SP}_t \), if \( (P, \phi) \xrightarrow{R_{op}} (P', \phi') \) and \( (P, \phi) \xrightarrow{\delta_{\phi, N_{ch}}} D \) then \( (P', \phi') \xrightarrow{\delta_{\phi', N_{ch}}} E \) and \( D \xrightarrow{R_{op}} E \) (and \( E \xrightarrow{R_{op}} D \) when \( op = \approx \)).

**Proof.** The proof for both input actions and output actions follows the same structure.

We will start by considering the case when \( a = in(\xi, \zeta) \) and only illustrate the part that changes for the case \( a = out(\xi, \phi) \).

By definition, \( (P, \phi) \xrightarrow{\delta_{in(\xi, \zeta)}} D \) implies that \( P = P_1 \cup \{in(u, x).P\} \), \( u = \xi \phi \), \( Msg(\xi \phi) \), \( vars(\xi, \zeta) \subseteq dom(\phi) \) and \( D = \delta_{\{P_1 \cup \{\{\{in(u, x).P\} \} \} \}} \). By definition of \( R_{op} \), we know that \( (P, \phi) \xrightarrow{R_{op}} (P', \phi') \) implies \( (P, \phi) \in \mathcal{SP}_t(\phi, N_{ch}) \), \( (P', \phi') \in \mathcal{SP}_t(\phi', N_{ch}) \) and \( \pi_{\phi, N_{ch}}(\{P, \phi\}) \). By hypothesis on \( \pi_c \land \pi_{all} \), we can assume that \( N_{ch} \cap fn(\xi, \zeta) = \emptyset \).

Let us consider a fresh sequence \( N_{ch}' = \{c_{ax_i}\}_{i=1}^{n} \) of public name, i.e. \( fn(N_{ch}') \cap fn(P, \phi, P', \phi', N_{ch}) = \emptyset \). Let us define \( \rho = \{\xi_i / ax_i\}_{i=1}^{n} \). Finally, consider a fresh public name \( ok \in N_{pub} \).

We build the following process \( Out \):

\[
\text{Out} := \text{in}(c_{ax_1}, x_1) \ldots \text{in}(c_{ax_n}, x_n).\text{out}(\xi \rho, \zeta \rho).\text{(Frame + 0.5 \text{out}(ok, ok).0)}
\]

\[
\text{Frame} := \text{out}(c'_{ax_1}, x_1)_i \ldots \text{out}(c'_{ax_n}, x_n)_0
\]

First notice that \( Out \) is closed and \( fn(Out) \subseteq N_{pub} \) since \( \xi \) and \( \zeta \) are recipes such that \( \text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi) \). If we denote \( P_O = \{\text{Out}\} \cup \pi_{\phi, N_{ch}}(\{P, \phi\}) \) and \( P'_O = \{\text{Out}\} \cup \pi_{\phi', N_{ch}}(\{P', \phi'\}) \), we have \( P_O \xrightarrow{R_{op}} P'_O \).

Since \( \pi_{\phi, N_{ch}}(\{P, \phi\}) = P \cup \{\text{out}(c_{ax_i}, ax_i \phi)_0\}_{i=1}^{n} \), we can successively apply the rule (COMM) on \( c_{ax_1}, \ldots, c_{ax_n} \) to obtain that \( P_O \xrightarrow{R_{o, \pi}} P \cup \{\text{out}(\xi \rho, \zeta \rho).\text{(Frame + 0.5 \text{out}(ok, ok).0)}\} \) with \( \sigma = \rho^{-1} \phi \). By hypothesis, \( \xi \phi = u \) and \( Msg(\xi \phi) \). Hence we can apply the rule (COMM) on \( u \) to obtain
By denoting $D_1 = 0.5 \cdot \delta_{P^0 \cup \{P(\xi_{/x})\} \cup \{\text{Frame} + 0, 0\}} + 0.5 \cdot \delta_{P^1 \cup \{P(\xi_{/x})\} \cup \{\text{Frame} + 0, 0\}}$, we deduce from $\mathcal{P}_o \Rightarrow^r \mathcal{P}_o'$ that there exists $E_1$ such that $\mathcal{P}_o' \Rightarrow^r E_1$ and $D_1 = \mathcal{P}_o \Rightarrow^{r_{\text{obs}}} \mathcal{P}_o$.

Thanks to Lemma [41] we can assume w.l.o.g. that no rule (PAR) or (NIL) is applicable on $E_1$.

Let us analyse $E_1$. We consider the following processes:

- $A_{j,k}^i(Q') = \{\text{out}(c_{axj}, x_j; \ldots; c_{axn}, x_n).\text{out}(\xi\rho_{\sigma_j}, \xi\rho_{\sigma_j})(Frame_{\sigma_{j-1}} + 0.5 \text{out}(ok, ok))\} \cup Q' \cup \{\text{out}(c_{axn}, ax_{n\rho\sigma}').0\}_{j=1}^m \cup \{0\}_{i=1}^n$

- $A_{j,\text{ch}}^i(Q') = \{\text{out}(\xi\rho_{\sigma_j}).(Frame_{\sigma_{j-1}} + 0.5 \text{out}(ok, ok))\} \cup Q' \cup \{0\}_{i=1}^n$

- $A_{j,\text{ok}}^i(Q') = \{\text{out}(\xi\rho_{\sigma_j})(Frame_{\sigma_{j-1}} + 0.5 \text{out}(ok, ok))\} \cup Q' \cup \{0\}_{i=1}^n$

where $\sigma_j = \{ax_{n\rho\sigma}/x_j\}_{j=1}^m$.

Let us denote $A_{j,k}^{i;\text{ch}} = \{A_{j,k}^i(Q) \mid Q \in \mathcal{SP}_\ell(\phi', N_{ch}) \land fn(Q) \cap N_{ch}' = \emptyset\}$. Similarly, we define $A_{j,\text{out}}^i$, $A_{j,\text{ch}}^i$ and $A_{j,\text{ok}}^i$. We define a measure function $m$ on processes from $A_{j,k}^i$, $A_{j,\text{out}}^i$, $A_{j,\text{ch}}^i$ and $A_{j,\text{ok}}^i$ as follows: For all $Q$,

- $Q \in A_{j,k}^i \Rightarrow m(Q) = (5, j, k)$
- $Q \in A_{j,\text{out}}^i \Rightarrow m(Q) = (4, k, 0)$
- $Q \in A_{j,\text{ch}}^i \Rightarrow m(Q) = (2, n - j, k)$
- $Q \in A_{j,\text{ok}}^i \Rightarrow m(Q) = (1, k, 0)$

Moreover, we consider a function $\text{corr} : \mathcal{SP} \mapsto \mathcal{SP}$ such that for all $(Q, \phi') \in \mathcal{SP}_\ell(\phi', N_{ch})$ with $fn(Q) \cap N_{ch}' = \emptyset$, $\text{corr}(A_{j,k}^i(Q)) = \text{corr}(A_{j,\text{out}}^i(Q)) = \text{corr}(A_{j,\text{ch}}^i(Q)) = \text{corr}(A_{j,\text{ok}}^i(Q)) = (Q, \phi')$.

Since $\mathcal{P}_o \Rightarrow^r E_1$, we know that there exists an infinite scheme

$$\delta_{P^0} = E_0^+ + E_0^-$$

$$E_0^+ \Rightarrow^r E_1^+ + E_1^-$$

$$E_k^- \Rightarrow^r E_{k+1}^- + E_{k+1}^+$$

such that $E_1 = \sum_{k \in \mathbb{N}} E_k^\top$. 

First notice that $\mathcal{P}_o = A_{j,\text{frame}}^i(\mathcal{P}')$ and $\text{corr}(\mathcal{P}_o) = (\mathcal{P}', \phi')$. Second, by a simple case analysis, we deduce the following statements for all $Q \rightarrow D$.
\[ Q \in \mathcal{A}^{j,k}_f \] implies either \( \text{supp}(D) \subseteq \mathcal{A}^{j-1,k+1} \cup \mathcal{A}^{j,k-1} \cup \mathcal{A}^{k+1}_f \) and \( \text{corr}(D) = \text{corr}(\delta_Q) \); or \( \text{supp}(D) \subseteq \mathcal{A}^{j,k}_f \) and \( \text{corr}(Q) \overset{\tau}{\rightarrow} \text{corr}(D) \).

\[ Q \in \mathcal{A}^{k}_\text{out} \] implies either \( \text{supp}(D) \subseteq \mathcal{A}^{k-1}_+ \) and \( \text{corr}(D) = \text{corr}(\delta_Q) \); or \( \text{supp}(D) \subseteq \mathcal{A}^{k}_+ \) and \( \text{corr}(Q) \overset{\text{in}(\xi,Q)}{\rightarrow} \text{corr}(D) \); or \( \text{supp}(D) \subseteq \mathcal{A}^{k}_\text{out} \) and \( \text{corr}(Q) \overset{\tau}{\rightarrow} \text{corr}(D) \).

\[ Q \in \mathcal{A}^{k}_+ \] implies either \( \text{supp}(D) \subseteq \mathcal{A}^{j+1,k}_\text{ch} \cup \mathcal{A}^{j,k-1}_\text{ch} \) and \( \text{corr}(D) = \text{corr}(\delta_Q) \); or \( \text{supp}(D) \subseteq \mathcal{A}^{j,k}_\text{ch} \) and \( \text{corr}(Q) \overset{\tau}{\rightarrow} \text{corr}(D) \).

\[ Q \in \mathcal{A}^{k}_\text{ch} \] implies either \( \text{supp}(D) \subseteq \mathcal{A}^{j+1,k}_\text{ch} \cup \mathcal{A}^{j,k-1}_\text{ch} \) and \( \text{corr}(D) = \text{corr}(\delta_Q) \); or \( \text{supp}(D) \subseteq \mathcal{A}^{j,k}_\text{ch} \) and \( \text{corr}(Q) \overset{\tau}{\rightarrow} \text{corr}(D) \).

\[ Q \in \mathcal{A}^{k}_\text{ok} \] implies either \( \text{supp}(D) \subseteq \mathcal{A}^{k-1}_\text{ch} \) and \( \text{corr}(D) = \text{corr}(\delta_Q) \); or \( \text{supp}(D) \subseteq \mathcal{A}^{k}_\text{ch} \) and \( \text{corr}(Q) \overset{\tau}{\rightarrow} \text{corr}(D) \).

One of the direct properties we deduce from these is that for all \( k \in \mathbb{N} \), for all \( \alpha \in \{ \rightarrow, \top \} \), \( \text{supp}(E^n_k) \subseteq \bigcup_{k=0}^n \bigcup_{j=0}^n \mathcal{A}^{j,k}_f \cup \mathcal{A}^{j-1,k} \cup \mathcal{A}^{j,k} \cup \mathcal{A}^{j,k}_+ \cup \mathcal{A}^{j,k}_\text{ch} \cup \mathcal{A}^{j,k}_\text{ok} \).

We proved that \( D_1 \overset{\text{op}_{\text{obs}}^{\mathcal{R}_c}}{\rightarrow} E_1 \). Consider \( E_{\text{ok}} \) the largest sub-distribution of \( E_1 \) such that for all \( \mathcal{P}'' \in \text{supp}(E_{\text{ok}}) \), \( \mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \}, \text{out}(ok, ok).0 \} \} \overset{\text{op}_{\text{obs}}^{\mathcal{R}'}}{\rightarrow} \mathcal{P}'' \). Notice that \( \text{RProb}_{\mathcal{R}_c}(\mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \}, \text{out}(ok, ok).0 \} \), \( \text{out} \) = 1. Hence, \( \text{supp}(E_{\text{ok}}) \subseteq \{ \mathcal{A}^{n,0}_{\text{ch}}(Q'') \} | (Q'', \phi') \in \mathcal{S}\mathcal{P}_{\ell}(\phi', \mathcal{N}_{\text{ch}}) \}. 

Similarly, if we denote \( E_{\text{ch}}' \) the largest sub-distribution of \( E_1 \) such that for all \( \mathcal{P}'' \in \text{supp}(E_{\text{ch}}') \), \( \mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \} \} \cup \mathcal{F}(\phi, \mathcal{N}_{\text{ch}}) \} \overset{\text{op}_{\text{obs}}^{\mathcal{R}'}}{\rightarrow} \mathcal{P}'' \), we obtain that \( \text{supp}(E_{\text{ch}}') \subseteq \{ \mathcal{A}^{n,0}_{\text{ch}}(Q'') \} | (Q'', \phi') \in \mathcal{S}\mathcal{P}_{\ell}(\phi', \mathcal{N}_{\text{ch}}) \}. \)

However, \( D_1 \overset{\text{op}_{\text{obs}}^{\mathcal{R}_c}}{\rightarrow} E_1 \) implies that \( D_1(\mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \} \} \cup \mathcal{F}(\phi, \mathcal{N}_{\text{ch}}) \} \) = 0.5 ≤ \( \sum_{\mathcal{P}''} E_{\text{ch}}'(\mathcal{P}'') \). Since \( \text{supp}(E_{\text{ch}}') \subseteq \mathcal{S}\mathcal{P}(\phi', \mathcal{N}_{\text{ch}}) \}, \) \( \text{E}_1 = E_{\text{ch}}' + E_{\text{ok}} \) with \( \sum_{\mathcal{P}''} E_{\text{ch}}'(\mathcal{P}'') \) = \( \sum_{\mathcal{P}''} E_{\text{ok}}(\mathcal{P}'') \)= 0.5. Finally, since no rule (NIL) or (PAR) is applicable on \( E_1 \), we deduce that \( \text{supp}(E_{\text{ch}}') \subseteq \mathcal{A}^{n,0}_{\text{ch}} \} \) and so \( \text{supp}(E_{\text{ch}}') \subseteq \mathcal{S}\mathcal{P}(\phi', \mathcal{N}_{\text{ch}}) \). 

Note that by construction of \( E_{\text{ch}}', \), \( \delta_{\mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \} \} \cup \mathcal{F}(\phi, \mathcal{N}_{\text{ch}}) \} \overset{\text{op}_{\text{obs}}^{\mathcal{R}'}}{\rightarrow} 2 \cdot E \). Thus, by denoting \( E = 2 \cdot E_{\text{ch}}' \) and by Lemma 54, we obtain that \( D = \pi_{\phi', \mathcal{N}_{\text{ch}}}(\delta_{\mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \} \} \cup \mathcal{F}(\phi, \mathcal{N}_{\text{ch}}) \}) \overset{\text{op}}{\rightarrow} \pi_{\phi', \mathcal{N}_{\text{ch}}}(E) \).

It remains to show that \( (\mathcal{P}', \phi') \overset{\text{in}(\xi,Q)}{\rightarrow} \mathcal{P}_1 \cup \{ P\{\zeta/\phi \}/x \} \} \cup \mathcal{F}(\phi, \mathcal{N}_{\text{ch}}) \} \overset{\text{op}}{\rightarrow} (E) \): Consider the predicate \( \pi^S \) such that \( \pi^S(Q) \) holds iff \( Q \subseteq \bigcup_{k=0}^n \mathcal{A}^{k}_\text{out} \cup \bigcup_{j=1}^n \mathcal{A}^{j,k}_f \). We also consider the predicates \( \pi^+ \) and \( \pi^- \) on \( \mathcal{N} \) such that \( \pi_+^S(Q \rightarrow D) \) holds iff \( Q \in \mathcal{A}_\text{out}^k \) for some \( k \) and \( \text{supp}(D) \subseteq \mathcal{A}^k_\text{ch} \); and \( \pi^-_S(Q \rightarrow D) \) holds iff \( \pi^S(Q) \) and either \( \text{corr}(\delta_Q) = \text{corr}(D) \) or \( \text{corr}(Q) \overset{\tau}{\rightarrow} \text{corr}(D) \). Notice that \( \pi^S, \pi_+^S \) and \( \pi^-_S \) satisfy the conditions of Lemma 57. Therefore, \( A, A^\top, B, B^\top, C \in D^{\leq 1}(\mathcal{N}_S) \) such that

\[ (\delta_{\mathcal{P}_0} \overset{\tau}{\rightarrow} A + A^\top) = \pi_+^S \quad (A \overset{\tau}{\rightarrow} B + B^\top) = \pi_-^S \quad B \overset{\tau}{\rightarrow} C \]

with \( E_1 = C + B^\top + A^\top \). Note that we already proved that \( E_1 = E_{\text{ch}}' \overset{\tau}{\rightarrow} E_{\text{ok}} \) and so we deduce that \( A^\top = B^\top = \emptyset \).
Notice that by definition of $\pi^s$, $A \xrightarrow{\tau} B$ implies $A \xrightarrow{\tau} B$. Moreover, $\pi^s(\tau \rightarrow D)$ also implies $\text{corr}(Q) \xrightarrow{\text{in}(\xi, \zeta)} \text{corr}(D)$. Therefore, $\text{corr}(A) \xrightarrow{\text{in}(\xi, \zeta)} \text{corr}(B)$.

Let us focus on $(\delta_{P_0} \xrightarrow{\tau} A) \models \pi^r$. We define the predicate $\pi^r$ such that $\pi^r(\tau \rightarrow D)$ holds iff $\text{corr}(Q) \xrightarrow{\tau \rightarrow N} \text{corr}(D)$. By construction, notice that $(A \xrightarrow{\tau \rightarrow N} B) \models \neg \pi^r \land \pi^r$ implies $\text{corr}(A) = \text{corr}(B)$ and either $A = \emptyset$ or $m(A) > m(B)$. Similarly, by construction, $(A \xrightarrow{\tau \rightarrow N} B) \models \pi^r \land \pi^r$ implies $\text{corr}(A) \xrightarrow{\tau \rightarrow N'} \text{corr}(B)$ and $m(A) = m(B)$. This allows us to apply Lemma 60 on $(\delta_{P_0} \xrightarrow{\tau} A) \models \pi^r$. Therefore, $\text{corr}(\delta_{P_0}) \xrightarrow{\tau} \text{corr}(A)$.

Let us summarize what we have proved so far: $\text{corr}(\delta_{P_0}) \xrightarrow{\tau} \text{corr}(A) \xrightarrow{\text{in}(\xi, \zeta)} \text{corr}(B)$ with $B \xrightarrow{\tau} E_1$ with $\text{supp}(B) \subseteq \bigcup_{k=1}^n A^k_x$. Let us redefine the predicate $\pi^S$ such that $\pi^S(x)$ holds iff $x \in \bigcup_{k=1}^n A^k_x$. We redefine $\pi_c$ and $\pi_s$ such that $\pi_c(x \rightarrow D) \models \pi^S(D)$ and $\text{corr}(x) \xrightarrow{\tau} \text{corr}(D)$; and $\pi_s(x \rightarrow D)$ holds when $\text{corr}(D) = \text{corr}(\delta_s)$ and $m(x) > m(D)$. By applying Lemma 58 on $B \xrightarrow{\tau} E_1$, we deduce that there exists $(B \xrightarrow{\tau} B_1 + B_1^t) \models \pi_c$, $(B_1 \xrightarrow{\tau} B_2 + B_2^t) \models \pi_s$, $B_1 \xrightarrow{\tau} B_3$ with $E_1 = B_3 + B_1^t + B_2^t$. Once again we know that $E_1 = E_{ch'} + E_{ok}$ and so we deduce that $B_1^t = \emptyset$.

By definition of $\pi_c$, we deduce that $\text{corr}(B) \xrightarrow{\tau} \text{corr}(B_1)$. Moreover, by definition of $\pi_s$, the transition $B_1 \xrightarrow{\tau} B_2 + B_2^t$ correspond to the execution of the probabilistic choice in the process $\text{Frame} + \tau \cdot \text{out}(ok, ok).0$. Thus, there exist two sub distributions $B_{ok}, B_{ch'}$ such that $B_2 + B_2^t = 0.5 \cdot B_{ok} + 0.5 \cdot B_{ch'}$, $\text{supp}(B_{ok}) \subseteq \bigcup_{k=0}^n A^k_{ch'}$, $\text{supp}(B_{ch'}) \subseteq \bigcup_{k=0}^n A^k_{ch'}$ and $\text{corr}(B_1) = \text{corr}(B_{ok}) = \text{corr}(B_{ch'})$. By applying Lemma 59, we deduce that $0.5 \cdot B_{ok} \xrightarrow{\tau} E_{ok}$ and $0.5 \cdot B_{ch'} \xrightarrow{\tau} E_{ch'}$, implying that $B_{ok} \xrightarrow{\tau} 2 \cdot E_{ok}$ and $B_{ch'} \xrightarrow{\tau} 2 \cdot E_{ch'}$.

We will finally show that $\text{corr}(B_{ch'}) \xrightarrow{\tau} \text{corr}(2 \cdot E_{ch'})$ by applying Lemma 60 with predicate $\pi^S$ on $\mathcal{N}_{ch}$ and predicates $\pi_{all}$ on $\rightarrow_{\mathcal{N}_{ch}}$ defined as:

- $\pi^S(x)$ holds iff $x \in \bigcup_{k=0}^n \bigcup_{j=0}^n A^{i,j}_{ch}$
- $\pi_c(x \rightarrow_{\mathcal{N}_{ch}} D)$ holds iff $m(x) = m(D)$ and $\text{corr}(x) \rightarrow_{\mathcal{N}_{ch}} \text{corr}(D)$.
- $\pi_{all}$ is always true.

This complete the proof of $\delta(P', \phi') \xrightarrow{\text{in}(\xi, \zeta)} \text{corr}(2 \cdot E_{ch'}) = \pi_{p \rightarrow c_{ch}}(E)$.

Let us now consider the case of the output action $a = \text{out}(\xi, ax)$. By definition, $(P, \phi) \xrightarrow{\text{out}(\xi, ax)} \text{corr}(D)$ implies that $P = P_1 \cup \{\text{out}(u, t).P\}_{u \equiv \xi, ax = ax_{n+1}, \text{Msg}(t)}$, $\text{vars}(\xi) \subseteq \text{dom}(\phi)$ and $D = \delta(\{P\} \cup P_1, \phi(\text{ax} \rightarrow \text{it}))$. Compare to the input case, we consider a sequence of public name $\mathcal{N}_{ch}$ with one additional channel, i.e. $\mathcal{N}_{ch} = [c_{ax_{n+1}}]'[n+1]$. Moreover, we consider the following process $In$ instead of the process $Out$:

$$In := \text{in}(c_{ax_1}, x_1) \ldots \text{in}(c_{ax_n}, x_n) \cdot \text{in}(\xi, x_{n+1}) \cdot (\text{Frame} + \tau \cdot \text{out}(ok, ok).0)$$

$$\text{Frame} := \text{out}(c'_{ax_1}, x_1).0 \mid \ldots \mid \text{out}(c'_{ax_{n+1}}, x_{n+1})$$

Thus the initial processes $P_O$ and $P'_O$ used in the proof are defined as $P_O = \{In\} \cup \pi_{p \rightarrow c_{ch}}((P, \phi))$ and $P'_O = \{In\} \cup \pi_{p \rightarrow c_{ch}}'(\phi')$. Executing the process $P_O$ allows us to
obtain:
\[ \mathcal{P}_O \xrightarrow{r} 0.5 \cdot \delta_{\mathcal{P}_1 \cup \{ P_{\text{out}(ok, ok), 0}\}} + 0.5 \cdot \delta_{\mathcal{P}_1 \cup \{ P \}_{\mathcal{F}(\{ x_{n+1} \} \cdot \mathcal{N}_{\mathcal{P}})}} \]

Then when we analyse the support of \( E_1 \), we consider the following processes:

- \( \mathcal{A}_{f, k}^j(Q', \phi') = \{ \text{in}(c_{ax_j}, x_j), \ldots, \text{in}(c_{ax_n}, x_n), \text{out}(\xi \rho \sigma_j, x_{n+1}).(\text{Frame} \sigma_j \cdot 0.5 \text{out}(ok, ok)) \} \cup \mathcal{Q} \cup \{ \text{out}(c_{ax_i}, ax_i \phi') \}_{i=1}^{n+k} \]
- \( \mathcal{A}_{in}^k(Q', \phi') = \{ \text{in}(\xi \rho, x_{n+1}).(\text{Frame} \sigma_n \cdot 0.5 \text{out}(ok, ok), 0) \} \cup \mathcal{Q} \cup \{ 0 \}_{i=1}^{k} \]
- \( \mathcal{A}_{ch}^k(Q', \phi') = \{ \text{out}(c_{ax_j}, ax_j \phi') \}_{j=1}^{j} \cup \{ \text{out}(c_{ax_{n+1}}, ax_{n+1} \phi') \}_{i=1}^{n+k} \} \cup \mathcal{Q} \cup \{ 0 \}_{i=1}^{k} \]

where \( \sigma_j = \{ ax_i \phi' / x_i \}_{i=1}^{j} \). Notice here that a frame is given as argument since it will be augmented once the process \( (\xi \rho, x_{n+1}) \) is executed. As such, the set \( \mathcal{A}_{f, k}^j \) is defined as \( \{ A_{f, k}^j(Q', \phi') \mid (Q', \phi') \in \mathcal{SP}_\ell(\phi', \mathcal{N}_{\text{ch}}) \land fn(Q) \cap \mathcal{N}_{\text{ch}} = \emptyset \} \). Similarly, the correspondence function \( \text{corr} : \mathcal{SP} \Rightarrow \mathcal{SP}_\ell \) is defined as for all \( (Q', \phi') \in \mathcal{SP}_\ell(\phi', \mathcal{N}_{\text{ch}}) \) with \( fn(Q) \cap \mathcal{N}_{\text{ch}} = \emptyset \), \( \text{corr}(A_{f, k}^j(Q', \phi')) = \text{corr}(A_{out}^k(Q', \phi')) = \text{corr}(A_{in}^k(Q', \phi')) = \text{corr}(A_{ch}^k(Q', \phi')) = \text{corr}(A_{ok}^k(Q', \phi')) = (Q', \phi') \). The measure function \( m \) only differ on \( \mathcal{A}_{ch}^j \) as follows: if \( Q \in \mathcal{A}_{ch}^j \), then \( m(Q) = (2, n+1-j, k) \). The rest of the proof follows by using the same reasoning.

### F.4 Simulation implies observational preorder

For this direction of the equivalence, we define the relation \( R_{op} \) with \( op \in \{ \leq, = \} \) on multisets of processes as follows: For all \( \mathcal{P}, \mathcal{P}' \in \mathcal{SP}, \mathcal{P} \xrightarrow{op} \mathcal{P}' \) if and only if there exist a multiset of processes \( \mathcal{P}_{\text{Att}} \), a renaming of variables \( \rho \), two extended processes \( (Q, \phi), (Q', \phi') \in \mathcal{SP}_\ell \) such that:

- \( |\phi| = |\phi'| = n \)
- \( \text{dom}(\rho) \subseteq \mathcal{N}_{\text{pub}}, \text{img}(\rho) \subseteq \mathcal{N}_{\text{priv}}, \text{img}(\rho) \cap \text{names}(Q, Q', \phi, \phi') = \emptyset \)
- \( \text{fv}((\mathcal{P}_{\text{Att}}) \in \{ x_1, \ldots, x_n \} \) and \( \text{fn}(\mathcal{P}_{\text{Att}}) \subseteq \mathcal{N}_{\text{pub}} \)
- \( \mathcal{P} = Q_\rho \cup \mathcal{P}_{\text{Att}} \{ ax_i \phi / x_i \}_{i=1}^{n} \rho \)
- \( \mathcal{P}' = Q'_\rho \cup \mathcal{P}_{\text{Att}} \{ ax_i \phi' / x_i \}_{i=1}^{n} \rho \)
- \( (Q, \phi) \xrightarrow{op}^{N_{\ell}} (Q', \phi') \)

**Lemma 62.** For all \( op \in \{ \leq, = \} \), for all \( \mathcal{P}, \mathcal{P}' \in \mathcal{SP}, \mathcal{P} \xrightarrow{op} \mathcal{P}' \) then for all closed \( \mathcal{P}_{\text{Att}} \in \mathcal{SP} \) such that \( \text{fn}(\mathcal{P}_{\text{Att}}) \subseteq \mathcal{N}_{\text{pub}} \) then \( \mathcal{P}_{\text{Att}} \cup \mathcal{P} \xrightarrow{op} \mathcal{P}_{\text{Att}} \cup \mathcal{P}' \).
Proof. Trivial by definition of $R_{op}$.

**Lemma 63.** Let $op \in \{\leq, \approx\}$. Let $D, D' \in D^{\leq}(S_{\mathcal{N}^f})$. Let $n \in \mathbb{N}$. Let $\rho$ be a renaming such that $\text{dom}(\rho) \subseteq N_{\text{pub}}$, $\text{img}(\rho) \subseteq N_{\text{priv}}$ and for all $(\mathcal{P}, \phi) \in \text{supp}(D) \cup \text{supp}(D')$, $\text{img}(\rho) \cap \text{names}(\mathcal{P}, \phi) = \emptyset$ and $|\phi| = n$. Let $\mathcal{P}_{\text{Att}} \in S\mathcal{P}$ such that $fv(\mathcal{P}_{\text{Att}}) \subseteq \{x_1, \ldots, x_n\}$ and $\text{fn}(\mathcal{P}_{\text{Att}}) \subseteq N_{\text{pub}}$.

If $D \overset{op^{N^f}}{\rightarrow} D'$ then

$$
\sum_{(Q, \phi) \in \text{supp}(D)} D((Q, \phi)) \cdot \delta_{Q_0U\mathcal{P}_{\text{Att}}(x_i/x_i)}^{\mathcal{P}} \cdot \sigma_{\rho} \sum_{(Q, \phi) \in \text{supp}(D')} D'((Q, \phi)) \cdot \delta_{Q_0U\mathcal{P}_{\text{Att}}(x_i/x_i)}^{\mathcal{P}} \cdot \rho_{\sigma}
$$

Proof. Given a frame $\phi$ of size $n$, we denote by $\sigma_{\phi}$ the substitution $\{\frac{x_i}{x_i}\}_{i=1}^n$. We denote by $E$ and $E'$ the sub-distributions $\sum_{(Q, \phi) \in \text{supp}(D)} D((Q, \phi)) \cdot \delta_{Q_0U\mathcal{P}_{\text{Att}}(x_i/x_i)}^{\mathcal{P}} \cdot \sigma_{\rho}$ and $\sum_{(Q, \phi) \in \text{supp}(D')} D'((Q, \phi)) \cdot \delta_{Q_0U\mathcal{P}_{\text{Att}}(x_i/x_i)}^{\mathcal{P}} \cdot \sigma_{\rho}$ respectively. Let $S \subseteq S\mathcal{P}$. Let us denote $S_1 = \{(Q, \phi) \in \text{supp}(D) \mid (Q_0 \cup \mathcal{P}_{\text{Att}}(x_i/x_i)) \in S\}$. Let $S' = \{(Q', \phi') \in \text{supp}(D') \mid (Q_0 \cup \mathcal{P}_{\text{Att}}(x_i/x_i)) \in S \cap \text{supp}(D) \land (Q, \phi) \overset{op^{N^f}}{\rightarrow} (Q', \phi')\}$. By definition of $E$, we have:

$$E(S) = \sum_{(Q, \phi) \in \text{supp}(D)} D((Q, \phi))$$

$$= D(S_1) \quad \text{by definition of } S_1$$

$$\leq D'(op^{N^f}(S_1)) \quad \text{by } D \overset{op^{N^f}}{\rightarrow} D'$$

$$\leq \sum_{(Q', \phi') \in \text{supp}(D')} D'((Q', \phi'))$$

$$\leq \sum_{(Q', \phi') \in \text{supp}(D')} D'((Q', \phi')) \quad \text{by definition of } R_{op}$$

This allows us to conclude that $E \overset{R_{op}}{\rightarrow} E'$.

**Lemma 64.** For all $op \in \{\leq, \approx\}$, for all $\mathcal{P}, \mathcal{P}' \in S\mathcal{P}$, if $\mathcal{P} \overset{R_{op}}{\rightarrow} \mathcal{P}'$ and $\mathcal{P} \overset{\tau_{N^f}}{\rightarrow} D$ then $\mathcal{P} \overset{\tau_{R_{op}N^f}}{\rightarrow} D'$ and $D \overset{R_{op}}{\rightarrow} D'$ (and $D' \overset{R_{op}}{\rightarrow} D$ when $op = \approx$).

Proof. By definition of $\mathcal{P} \overset{R_{op}}{\rightarrow} \mathcal{P}'$, we know that there exist $(Q, \phi), (Q', \phi') \in S\mathcal{P}_t$, $\mathcal{P}_{\text{Att}} \in S\mathcal{P}$, a renaming of variables $\rho$ and two substitutions $\sigma, \sigma'$ such that $(Q, \phi) \overset{op^{N^f}}{\rightarrow} (Q', \phi')$, $\mathcal{P} = Q_0 \cup \mathcal{P}_{\text{Att}}(x_i/x_i), \mathcal{P}' = Q_0 \cup \mathcal{P}_{\text{Att}}(x_i/x_i), |\phi| = |\phi'|, fv(\mathcal{P}_{\text{Att}}) \subseteq \{x_1, \ldots, x_n\}, \sigma = \{\frac{x_i}{x_i}\}_{i=1}^n, \sigma' = \{\frac{x_i}{x_i}\}_{i=1}^n, \text{dom}(\rho) \subseteq N_{\text{pub}}, \text{img}(\rho) \subseteq N_{\text{priv}}$ and $\text{img}(\rho) \cap \text{names}(Q, Q') = \emptyset$.

Let us do a case analysis on the rule applied in $\mathcal{P} \overset{\tau_{R_{op}N^f}}{\rightarrow} D$.  

124
Case of a rule solely applied on $Q, \rho$: In such a case, $Q, \rho \xrightarrow{\tau} \pi_{N^0} D_1$ such that $D = \sum_{Q'' \in SP} D_1(Q'') \cdot \delta_{Q'' \cdot P_{Att}}$. Notice that $Q, \rho \xrightarrow{\tau} \pi_{N^0} D_1$ implies that $(Q, \phi) \xrightarrow{\tau} \pi_{N^0} \sum_{Q'' \in SP} D_1(Q'' \cdot \rho) \cdot \delta_{Q'' \cdot \phi} = D_2$ and for all $(Q'', \phi) \in \text{supp}(D_2)$, $\text{img}(\rho) \cap \text{names}(Q'') = \emptyset$. By hypothesis, $(Q, \phi)$ $op_{N^0}$ $(Q', \phi')$. Hence there exists $(Q', \phi') \xrightarrow{\tau} r_{N^0} D'_1$ and $D_2$ $op_{N^0}$ $D'_2$ (and $D'_2$ $op_{N^0}$ $D_2$ when $op = \approx$). By the semantics of $\tau$ in Figure 10 and by Lemma 45, we obtain that for all $(Q'', \phi'') \in \text{supp}(D'_2)$, $\phi'' = \phi''$ and we can assume w.l.o.g. that $\text{img}(\rho) \cap \text{names}(Q'') = \emptyset$ (i.e. when the rule (NEW) generates a new private names, we can always pick one that is not in $\text{img}(\rho)$).

Hence, we also deduce that $Q' \rho \xrightarrow{\tau} r_{N^0} \sum_{(Q'', \phi'') \in \text{supp}(D'_2)} D'_2(Q'', \phi') \cdot \delta_{Q'' \cdot \rho}$. Therefore, $Q' \rho \cup P_{Att} \cdot \tau \rightarrow r_{N^0} \sum_{(Q'', \phi'') \in \text{supp}(D'_2)} D'_2(Q'', \phi') \cdot \delta_{Q'' \cdot \rho \cdot P_{Att} \cdot \tau}$. Let us denote $D' = \sum_{(Q'', \phi'') \in \text{supp}(D'_2)} D'_2(Q'', \phi') \cdot \delta_{Q'' \cdot \rho \cdot P_{Att} \cdot \tau}$. We conclude by applying Lemma 63 that $D \xrightarrow{\tau} D'$ (and $D' \xrightarrow{\tau} D$ when $op = \approx$).

Case of a rule solely applied on $P_{Att} \sigma$: In such a case $P_{Att} \sigma \xrightarrow{\tau} \pi_{N^0} D_1$ such that $D = \sum_{P_{Att} \in SP} D_1(P_{Att}) \cdot \delta_{P_{Att} \cdot \tau}$. We first show that there exists a renaming $\rho'$ with $\text{dom}(\rho') \subseteq \mathcal{N}_{pub}$, $\text{img}(\rho') \subseteq \mathcal{N}_{priv}$ and $\text{img}(\rho') \cap \text{names}(Q, Q') = \emptyset$ such that:

- for all $P_{Att} \in \text{supp}(D_1)$, there exists a multiset of process $\overline{P_{Att}}$ such that $P_{Att} = \overline{P_{Att}} \{ x_i \mapsto \rho' \}$, $\text{fn}(P_{Att}) \subseteq \mathcal{N}_{pub}$ and $\text{fn}(P_{Att}) \subseteq \{ x_1, \ldots, x_n \}$
- $P_{Att} \cdot \tau \rightarrow \pi_{N^0} \sum_{P_{Att} \in \text{supp}(D_1)} D_1(P_{Att}) \cdot \delta_{\overline{P_{Att}} \{ x_i \mapsto \rho' \}}$.

We do a case analysis on the rule applied in $P_{Att} \sigma \xrightarrow{\tau} \pi_{N^0} D_1$. The cases of the rules (NULL), (PAR), (REPL), (CHOICE-1), (CHOICE-2) and (PCchoice) are trivial by taking $\rho' = \rho$.

Note that in all these cases but (PCchoice), $D_1$ is in fact a dirac.

Let us now look at the rule (COMM): In such a case, $P_{Att} = \{ \text{out}(u, t). P, \text{in}(v, x). Q \} \cup P_{Att, 1}$, $P_{Att} = \{ P. \sigma, Q. \sigma \{ t/x \} \}$ $\cup$ $P_{Att, 1}$, $\sigma = \delta_{P_{Att}}$, $\text{msg}(t) \sigma$ and $u \sigma = v \sigma$. Note that $t \sigma = t \rho \sigma$, $u \sigma = u \rho \sigma$ and $v \sigma = v \rho \sigma$. Since both $\approx$ and $\text{msg}(\cdot)$ are closed under renaming, we deduce that $u \phi \approx v \phi$ and $\text{msg}(t \phi)$. Furthermore, $\text{fn}(t, u, v) \subseteq \mathcal{N}_{pub}$ meaning that $t, u, v$ are recipes. Since we know by hypothesis that $(Q, \phi)$ $op_{N^0}$ $(Q', \phi')$, we deduce that $u \phi \approx v \phi$ implies $u \phi' \approx v \phi'$, and $\text{msg}(t \phi)$ implies $\text{msg}(t \phi')$. Once again, by the closure under renaming argument, we obtain that $u \phi' \rho = v \phi' \rho$ and $\text{msg}(t \phi' \rho)$. This allows us to deduce that $P_{Att} \sigma \cdot \tau \rightarrow \pi_{N^0} \delta_{\{ P, Q \{ t/x \} \} \cup P_{Att, 1} \sigma}$, and so we conclude by taking $\rho' = \rho$ and $\overline{P_{Att}} = \{ P, Q \{ t/x \} \} \cup P_{Att, 1}$.

The cases of the rules (THEN) and (ELSE) are done similarly since the condition to apply the rule only depend on the success or not of an equality $\approx$.

Finally, let us consider the rule (NEW): In such a case, $P_{Att} = \{ \text{new} a. P \} \cup P_{Att, 1}$, $P_{Att} = \{ P. \sigma \{ t/a \} \} \cup P_{Att, 1}$ and $\sigma = \delta_{P_{Att}}$, where $a$ is fresh. W.l.o.g., we can take $a$ such that $a \notin \text{names}(Q, Q', \phi, \phi')$. We take a new fresh name $a_{pub} \in \mathcal{N}_{pub}$ and we conclude by defining $\rho' = \rho(a_{pub} \mapsto a')$ and $\overline{P_{Att}} = \{ P \{ a_{pub}/a \} \} \cup P_{Att, 1}$.

We have therefore proved that:

- $P_{Att} \sigma \xrightarrow{\tau} \pi_{N^0} \sum_{P_{Att} \in \text{supp}(D_1)} D_1(P_{Att}) \cdot \delta_{\overline{P_{Att}} \{ x_i \mapsto \rho' \}}$.
• $\mathcal{P}_{Att'} \xrightarrow{\tau} N^o \sum_{\mathcal{P}'_{Att} \in \text{supp}(D_1)} D_1(\mathcal{P}'_{Att}) \cdot \delta_{\overline{\mathcal{P}'_{Att}}}^{\{x_i \phi/x_i\}_{i=1}^n} \rho'\to$

Since $Q\rho = Q'\rho$ and $Q'\rho = Q'\rho'$, we obtain:

• $\mathcal{P}_{Att} \cup Q\rho \xrightarrow{\tau} N^o \sum_{\mathcal{P}'_{Att} \in \text{supp}(D_1)} D_1(\mathcal{P}'_{Att}) \cdot \delta_{\overline{Q\rho \cup \mathcal{P}'_{Att}}}^{\{x_i \phi/x_i\}_{i=1}^n} \rho' = D$

• $\mathcal{P}_{Att} \cup Q'\rho \xrightarrow{\tau} N^o \sum_{\mathcal{P}'_{Att} \in \text{supp}(D_1)} D_1(\mathcal{P}'_{Att}) \cdot \delta_{\overline{Q'\rho \cup \mathcal{P}'_{Att}}}^{\{x_i \phi/x_i\}_{i=1}^n} \rho' = D$

Note that for all $\mathcal{P}'_{Att} \in \text{supp}(D_1)$, by $(Q, \phi)$ $op^N_e (Q', \phi')$ and by construction of $\rho'$ and $\overline{\mathcal{P}'_{Att}}$, we have that:

$$Q\rho' \cup \overline{\mathcal{P}'_{Att}}^{\{x_i \phi/x_i\}_{i=1}^n} R_{op} Q\rho' \cup \overline{\mathcal{P}'_{Att}}^{\{x_i \phi/x_i\}_{i=1}^n} \rho'$$

Note that when $op = \approx$, $(Q, \phi)$ $op^N_e (Q', \phi')$ implies $(Q', \phi')$ $op^N_e (Q, \phi)$ and so we also have:

$$Q'\rho' \cup \overline{\mathcal{P}'_{Att}}^{\{x_i \phi/x_i\}_{i=1}^n} R_{op} Q'\rho' \cup \overline{\mathcal{P}'_{Att}}^{\{x_i \phi/x_i\}_{i=1}^n} \rho'$$

By denoting $D' = \sum_{\mathcal{P}'_{Att} \in \text{supp}(D_1)} D_1(\mathcal{P}'_{Att}) \cdot \delta_{\overline{Q'\rho \cup \mathcal{P}'_{Att}}}^{\{x_i \phi/x_i\}_{i=1}^n} \rho'$, we conclude that $D \xrightarrow{R_{op}} D'$ and $\mathcal{P}_{Att} \cup Q'\rho \xrightarrow{\tau} N^o D'$ (and $D' \xrightarrow{R_{op}} D$ when $op = \approx$).

Case of rule (COMM) between an output on $Q\rho$ and an input on $\mathcal{P}_{Att} \phi$:

In such a case $Q = \{\text{out}(u, t), P\} \cup Q_1$, $\mathcal{P}_{Att} = \{\text{in}(v, x), Q\} \cup \mathcal{P}_{Att,1}$, $D = \delta_{\{P, Q, \phi\}}^{\{P, Q, \phi\}} \cup \mathcal{P}_{Att,1} \cup \mathcal{Q}_\rho$, $Msg(t)$ and $\nu \phi = u \phi$. Note that $f(v) \subseteq N_{pub}$ and $f(u) \subseteq \{x_1, \ldots, x_n\}$. If we denote $\rho_{\text{AX}} = \{x_i \mapsto \text{ax}i\}_{i=1}^n$, we obtain that $u_{\rho_{\text{AX}}}$ is a recipe with $f(v_{\rho_{\text{AX}}}) \subseteq \text{dom}(\phi)$. Note that since $Msg(\cdot)$ and $\approx$ are closed by renaming, we deduce from $Msg(t)$ and $\nu \phi = u \phi$ that $Msg(t)$ and $u_{\rho_{\text{AX}}} \phi = u$. Thus, $(Q, \phi)$ $op^N_e (Q', \phi')$. Thus, there exists $D_1$ such that $(Q', \phi') \xrightarrow{\text{out}(v_{\rho_{\text{AX}}, \text{ax}i_{n+1}})} N^o D_1$ and $D_1 \xrightarrow{op^N_e} D_1'$ (and $D_1' \xrightarrow{op^N_e} D_1$ when $op = \approx$). By definition, there exists $E$ such that $(Q', \phi') \xrightarrow{\tau} N^o \mathcal{P}^{\text{in}(v, x), Q'} \xrightarrow{\text{out}(v_{\rho_{\text{AX}}, \text{ax}i_{n+1}})} N^o F$ and $F \xrightarrow{\tau} N^o D_1'$.

By the semantics of $\tau$ in Figure 10 and by Lemma 45, we obtain that for all $(Q'', \phi'') \in \text{supp}(E)$, $\phi' = \phi''$ and we can assume w.l.o.g. that $\text{img}(\rho) \cap \text{names}(Q'') = \emptyset$ (i.e. when the rule (NEW) generates a new private name, we can always pick one that is not in $\text{img}(\rho)$). Hence, we also deduce that $Q\rho \xrightarrow{\tau} N^o \sum_{(Q'', \phi'') \in \text{supp}(E)} E((Q'', \phi')) \cdot \delta_{Q''\rho}$. Therefore $Q'\rho \cup \mathcal{P}_{Att} \phi' \xrightarrow{\tau} N^o \sum_{(Q'', \phi'') \in \text{supp}(E)} E(Q'', \phi') \cdot \delta_{Q''\rho} \cup \mathcal{P}_{Att} \phi'$.

Note that $D_1 \xrightarrow{op^N_e} D_1'$ and $D_1$ being a distribution imply that $E, F$ and $D_1'$ are also distributions. Thus for all $(Q'', \phi'') \in \text{supp}(E)$, there exist $u', t', P'$, $Q''$ such that $Q'' = \{\text{out}(u', t'), P'\} \cup Q''$ such that $u' \approx v_{\rho_{\text{AX}}} \phi'$, $Msg(t')$ and $(Q'', \phi') \xrightarrow{\text{out}(v_{\rho_{\text{AX}}, \text{ax}i_{n+1}})} N^o \delta_{\{P', Q'' \cup \text{ax}i_{n+1} \mapsto t'\}}$ with $(\{P', Q'' \cup \text{ax}i_{n+1} \mapsto t'\}) \in \text{supp}(F)$. Since $u' \approx v_{\rho_{\text{AX}}} \phi'$ and $Msg(t')$ imply $u' \phi' = v \phi'$ and $Msg(t' \rho)$, we deduce that $Q''\rho \cup \{\text{in}(v \phi', x), Q\phi'\} \cup \mathcal{P}_{Att,1} \phi' \xrightarrow{\tau} N^o$.
If we denote \( \phi_1' = \phi'\{x_{n+1} \mapsto t'\} \) and \( \sigma_1' = \{(x_i' \mapsto x_i)\}_{i=1}^{n+1} \rho \) then we obtain \( Q'' \rho \cup \{[\text{in}(v'\sigma', x)].Q\rho\} \cup P_{\text{Att},1}\sigma' \xrightarrow{\tau_n} \delta_{\{P''\cup Q''\rho\cup P_{\text{Att},2}\sigma'\}} \).

We have therefore proved that:

\[
\sum_{(Q'', \phi'') \in \text{supp}(E)} E((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},1}\sigma'} \xrightarrow{\tau_n} \sum_{(Q'', \phi'') \in \text{supp}(F)} F((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},2}\{x_{n+1}' / x_i\}_{i=1}^{n+1} \rho} \]

Note that \( F \xrightarrow{\tau_{n'}} D' \). Once again by the semantics of \( \tau_{n'} \) in Figure 10 and by Lemma 45, we can easily show that:

\[
\sum_{(Q'', \phi'') \in \text{supp}(F)} F((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},2}\{x_{n+1}' / x_i\}_{i=1}^{n+1} \rho} \xrightarrow{\tau_{n'}} \sum_{(Q'', \phi'') \in \text{supp}(D')} D'_1((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},2}\{x_{n+1}' / x_i\}_{i=1}^{n+1} \rho} \]

Let us denote \( D' = \sum_{(Q'', \phi'') \in \text{supp}(D')} D'_1((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},2}\{x_{n+1}' / x_i\}_{i=1}^{n+1} \rho} \). We conclude by applying Lemma 63 that \( D \xrightarrow{\tau_{n'}} D' \) (and \( D' \xrightarrow{\tau_{n'}} D \) when \( op = \infty \)).

Case of rule (Comm) between an input on \( Q \rho \) and an output on \( P_{\text{Att}} : \) In such a case \( Q = \{[\text{in}(v, x)].Q\} \cup Q_1 \), \( P_{\text{Att}} = \{[\text{out}(u, t)].P\} \cup P_{\text{Att},1} \). \( D = \delta_{\{P\rho, Q\rho^{[\sigma'/x]}\}} \cup P_{\text{Att},1} \cup Q_1 \rho \), \( \text{Msg}(t\sigma) \) and \( v\rho = u\sigma \). Note that \( fn(u, t) \subseteq N_{pub} \) and \( fv(u, t) \subseteq \{x_1, \ldots, x_n\} \). If we denote \( \rho_{\text{AX}} = \{(x_i \mapsto \text{ax}_i)\}_{i=1}^{n+1} \), we obtain that \( u_p\rho_{\text{AX}} \) and \( t\rho_{\text{AX}} \) are recipes with \( \text{fu}(u_p\rho_{\text{AX}}, t\rho_{\text{AX}}) \subseteq \text{dom}(\phi) \). Note that since \( \text{Msg}(\cdot) \) and \( \equiv \) are closed by renaming, we deduce from \( \text{Msg}(t\sigma) \) and \( v\rho = u\sigma \) that \( \text{Msg}(t\rho_{\text{AX}}\phi) \) and \( v = u_p\rho_{\text{AX}}\phi \). Thus, \( (Q, \phi) \xrightarrow{\text{in}(u_p\rho_{\text{AX}}, t\rho_{\text{AX}})} N_{\text{pub}} \delta_{\{Q^{[\rho_{\text{AX}}'\phi'/\rho']}\} \cup Q_1, \phi} = D_1 \).

If we define \( P_{\text{Att},2} = \{P\} \cup P_{\text{Att},1} \) then we obtain that \( D = \delta_{\{Q^{[\rho_{\text{AX}}'\phi'/\rho']}\} \cup Q_1, \rho \cup P_{\text{Att},2}\sigma' \} \).

Notice that \( \text{fu}(P_{\text{Att},2}) \subseteq \{x_1, \ldots, x_n\} \) and \( fn(P_{\text{Att},2}) \subseteq N_{pub} \).

By hypothesis, we have \( (Q, \phi) \xrightarrow{\text{op}N_{\text{pub}}} (Q', \phi') \). Thus, there exists \( D'_1 \) such that \( (Q', \phi') \xrightarrow{\tau_{n'}} D'_1 \) and \( \tau_{n'} \xrightarrow{\text{op}N_{\text{pub}}} \tau_{n'} \) (and \( \tau_{n'} \xrightarrow{\text{op}N_{\text{pub}}} \tau_{n'} \) when \( op = \infty \)). By definition, there exists \( E \) and \( F \) such that \( (Q', \phi') \xrightarrow{\tau_{n'}} E \), \( E \xrightarrow{\text{in}(u_p\rho_{\text{AX}}, t\rho_{\text{AX}})} N_{\text{pub}} \xrightarrow{\tau_{n'}} \tau_{n'} \).

By the semantics of \( \tau_{n'} \) in Figure 10 and by Lemma 45, we obtain that for all \( (Q'', \phi'') \in \text{supp}(E) \), \( \phi'' = \phi'' \) and we can assume w.l.o.g. that \( \text{img}(\rho) \cap \text{names}(Q'') = \emptyset \) (i.e. when the rule (NEW) generates a new private name, we can always pick one that is not in \( \text{img}(\rho) \)). Hence, we also deduce that \( (Q', \phi') \xrightarrow{\tau_{n'}} \sum_{(Q'', \phi'') \in \text{supp}(E)} E((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},1}\sigma'} \).

Note that \( D_1 \xrightarrow{\text{op}N_{\text{pub}}} D'_1 \) and \( D_1 \) being a distribution imply that \( E, F \) and \( D'_1 \) are also distributions. Thus for all \( (Q'', \phi'') \in \text{supp}(E) \), there exist \( v', x'', Q'' \) such that \( Q'' = \{[\text{in}(v', x'').Q''].Q''\} \cup Q'' \) and \( v'' = u_p\rho_{\text{AX}}\phi' \), \( \text{Msg}(t\rho_{\text{AX}}\phi') \) and \( (Q'', \phi') \xrightarrow{\text{in}(u_p\rho_{\text{AX}}, t\rho_{\text{AX}})} N_{\text{pub}} \delta_{\{Q'^{[\rho_{\text{AX}}'\phi'/\rho']}\} \cup Q''}, \phi' \in \text{supp}(F) \). Since \( v'' = u_p\rho_{\text{AX}}\phi' \) and \( \text{Msg}(t\rho_{\text{AX}}\phi') \) imply \( v'' = u\sigma' \) and \( \text{Msg}(t\rho_{\text{AX}}\phi') \), we deduce that \( Q'' \rho \cup \{[\text{out}(u\sigma', t\sigma')].P\sigma'\} \cup P_{\text{Att},1}\sigma' \xrightarrow{\tau_{n'}} \delta_{\{P\rho, Q\rho^{[\sigma'/x]}\} \cup Q''}, \rho \cup P_{\text{Att},2}\sigma' \).

Therefore \( Q'' \rho \cup \{[\text{out}(u\sigma', t\sigma')].P\sigma'\} \cup P_{\text{Att},1}\sigma' \xrightarrow{\tau_{n'}} \delta_{\{Q'^{[\rho_{\text{AX}}'\phi'/\rho']}\} \cup Q''\rho \cup P_{\text{Att},2}\sigma'} \).

We have therefore proved that:

\[
\sum_{(Q'', \phi'') \in \text{supp}(E)} E((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},1}\sigma'} \xrightarrow{\tau_{n'}} \sum_{(Q'', \phi'') \in \text{supp}(F)} F((Q'', \phi'')) \cdot \delta_{Q'' \rho \cup P_{\text{Att},2}\sigma'}
\]
Note that $F \xrightarrow{\tau} \tau_{N'} D'_1$. Once again by the semantics of $\tau_{N'}$ in Figure 10 and by Lemma 45, we can easily show that:

$$\sum_{(Q'', \phi') \in \text{supp}(F')} F((Q'', \phi')) \cdot \delta_{Q''} \cdot \mathcal{P}_\text{Att},2\sigma' \xrightarrow{\tau} \tau_{N'} \sum_{(Q'', \phi') \in \text{supp}(D'_1)} D'_1((Q'', \phi')) \cdot \delta_{Q''} \cdot \mathcal{P}_\text{Att},2\sigma'. $$

Let us denote $D' = \sum_{(Q'', \phi') \in \text{supp}(D'_1)} D'_1((Q'', \phi')) \cdot \delta_{Q''} \cdot \mathcal{P}_\text{Att},2\sigma'$. We conclude by applying Lemma 63 that $D \xrightarrow{R_{op}} D'$ (and $D' \xrightarrow{R_{op}} D$ when $op = \approx$).

**Lemma 65.** For all $op \in \{\leq, \approx\}$, for all $\mathcal{P}, \mathcal{P}' \in \mathcal{SP}$, if $\mathcal{P} \xrightarrow{R_{op}} \mathcal{P}'$ then for all $c \in N_{pub}$, $\text{RProb}_{\mathcal{P}}(\mathcal{P}, \downarrow c) \leq \text{RProb}_{\mathcal{P}'}(\mathcal{P}', \downarrow c)$.

**Proof.** By Lemma 52, $\text{RProb}_{\mathcal{P}}(\mathcal{P}, \downarrow c) = \sup\{ D(\downarrow c) \mid \delta_\mathcal{P} \xrightarrow{\tau} D \text{ with finite support}\}$.

Let $\mathcal{P} \xrightarrow{\tau} D$ with finite support. By Lemma 64 and Property 4a of Lemma 48, we deduce that there exists $\mathcal{P}' \xrightarrow{\tau} D'$ such that $D \xrightarrow{R_{op}} D'$.

By definition of $D \xrightarrow{R_{op}} D'$, we know that for all $\mathcal{P}_1 \in \text{supp}(D)$, $\mathcal{P}'_1 \in \text{supp}(D')$, there exist $\mathcal{P}_\text{Att}$, a renaming of variables $\rho$ and two extended processes $(Q, \phi)$ and $(Q', \phi')$ such that:

- $|\phi| = |\phi'| = n$
- $(Q, \phi) \ x_1 \dashrightarrow_n (Q', \phi')$
- $\mathcal{P}_1 = Q \rho \cup \mathcal{P}_\text{Att} \{x_i \mapsto \rho_i\}_{i=1}^n$
- $\mathcal{P}'_1 = Q' \rho \cup \mathcal{P}_\text{Att} \{x_i \mapsto \rho_i\}_{i=1}^n$

By construction, if $\mathcal{P}_\text{Att} \subseteq \downarrow c$ then $\mathcal{P}_1, \mathcal{P}'_1 \subseteq \downarrow c$. Otherwise, if $(Q, \phi) \xrightarrow{out(c, a \times_i+1)} E$ then by Figure 10, we know that $E$ is a dirac. But $(Q, \phi) \ x_1 \dashrightarrow_n (Q', \phi')$ implies that there exist $(Q', \phi') \xrightarrow{\tau}, F_1$ and $F_1 \xrightarrow{out(c, a \times_i+1)} F_2$ and $F_2 \xrightarrow{\tau}, E'$ such that $E \dashrightarrow_n E'$. By definition of $\ x_1 \dashrightarrow_n$ and since $E$ is a dirac, we deduce that $E'$ is a distribution, which implies that both $F_1$ and $F_2$ are distributions. In particular, $F_1 \xrightarrow{out(c, a \times_i+1)} F_2$ tells us that $\sum_{Q'' \in c} F_1(Q'') = 1$. Note that the $\tau$ on $(Q', \phi')$ carries over $\mathcal{P}'_1$. Thus, there exists $H$ such that $\mathcal{P}'_1 \xrightarrow{\tau} H$ and $H(\downarrow c) = 1$.

Finally, let $S = \text{supp}(D) \cap \downarrow c$. Since, $D \xrightarrow{R_{op}} D'$, we deduce that $D(S) \leq D'(R_{op} (S))$. As we have just show that for all $\mathcal{P}'_1 \in R_{op} (S)$, there exists $H$ such that $\mathcal{P}'_1 \xrightarrow{\tau} H$ with $H(\downarrow c) = 1$, we deduce that there exists $D''$ such that $D' \xrightarrow{\tau} D''$ such that $D(S) \leq D'(R_{op} (S)) \leq D''(\downarrow c)$.

To summarize, we have show that for all $\mathcal{P} \xrightarrow{\tau} D$ with finite support, there exists $\mathcal{P}' \xrightarrow{\tau} D''$ such that $D(\downarrow c) \leq D''(\downarrow c)$. Hence $\sup\{ D(\downarrow c) \mid \delta_\mathcal{P} \xrightarrow{\tau} D \text{ with finite support}\} \leq \sup\{ D(\downarrow c) \mid \delta_\mathcal{P'} \xrightarrow{\tau} D \}$. We conclude by applying again Lemma 52.

\[\square\]
Lemma 66. Let \(\mathcal{P}, \mathcal{Q}\) two processes in \(\mathcal{MP}\).

\[
\mathcal{P} \leq_{\text{obs}}^{\mathcal{R}_s} \mathcal{Q} \text{ iff } (\mathcal{P}, \emptyset) \preceq_{\text{sim}}^N (\mathcal{Q}, \emptyset) \quad \text{and} \quad \mathcal{P} \approx_{\text{obs}}^{\mathcal{R}_s} \mathcal{Q} \text{ iff } (\mathcal{P}, \emptyset) \approx_{\text{bisim}}^N (\mathcal{Q}, \emptyset)
\]

Proof. The proof of observational preorder (resp. equivalence) implying simulation (resp. bisimulation) is directly given by application of Lemmas 55, 56 and 61. Lemmas 62, 64 and 65 directly give that simulation (resp. bisimulation) implies \(\leq_{\text{obs}}^{\mathcal{R}_s}\) (resp. \(\approx_{\text{obs}}^{\mathcal{R}_s}\)). We conclude by Lemma 53.

\[\square\]

G Non probabilistic processes

G.1 Hennessy-Milner’s Logical caracterisation

We split the proof of Lemma 12 in two lemmas: Lemmas 66 and 68 below.

Lemma 66. Let \(\mathcal{P}\) a non-probabilistic process, \(\phi\) a frame. Let \(F \in \mathcal{F}\). Let \(ok \in \mathcal{N}_{\text{pub}}\) such that \(ok \notin \text{fn}(\mathcal{P}, \phi, F)\). If \((\mathcal{P}, \phi) \models F\) then there exists a resolution \(R = (Y, \text{corr}, r)\), and \(s \in Y\) such that \(\text{corr}(s) = (\mathcal{P} \cup \{\text{Adv}_{F,|\phi|}^{\phi}\})\) and \(\text{RProb}_R(s, \text{corr}^{-1}(\downarrow ok)) = 1\).

Proof. Let \(n = |\phi|\). We do the proof by induction on the structure of the logical formula \(F\).

Case \(F = \top\): \(\text{Adv}_{F,1,n}^{\phi} \in \downarrow ok\) hence the result directly holds.

Case \(F = a.F’\): we suppose that \((\mathcal{P}, \phi) \models F\). Then by definition of formula validity, we know that there exists a state \((\mathcal{Q}, \psi)\) such that \((\mathcal{P}, \phi) \xrightarrow{\tau^* a} \xrightarrow{\tau^*} (\mathcal{Q}, \psi)\), and \((\mathcal{Q}, \psi) \models F’\). Using the induction hypothesis, we can deduce from \((\mathcal{Q}, \psi) \models F’\) that there exists a pair \((R_1, s_1)\), where \(R_1 = (Y_1, \text{corr}_1, \tau_1)\) is a resolution on the NPLTS \(\mathcal{N}^o\), \(s_1 \in Y_1\) such that \(\text{corr}(s_1) = Q \mid \text{Adv}_{F’,n'}^{F,n'}\psi\) and \(\text{RProb}_{R_1}(s_1, \text{corr}_1^{-1}(\downarrow ok)) = 1\) (where \(n' = n + 1\) when \(a\) is an output and \(n' = n\) otherwise).

Moreover, we can deduce from \((\mathcal{P}, \phi) \xrightarrow{\tau^* a} \xrightarrow{\tau^*} (\mathcal{Q}, \psi)\) that \(\text{Adv}_{F,n}^{\phi} \neq 0\) and there exists a sequence of steps:

\[
(\mathcal{P}, \phi) \xrightarrow{\tau^* a} \xrightarrow{\tau^*} (\mathcal{Q}, \psi)
\]

Looking at the way we define the adversary \(\text{Adv}_{F,n}^{\phi}\) in Definition 27, we see that we can deduce:

\[
(\mathcal{P} \cup \{\text{Adv}_{F,n}^{\phi}\}) \xrightarrow{\tau^* a} \xrightarrow{\tau^*} (\mathcal{Q} \cup \{\text{Adv}_{F,n}^{\phi}\})
\]

We build a resolution \(R = (Y, \text{corr}, r)\) on the NPLTS \(\mathcal{N}^o\) as follows (our construction is graphically represented in Figure 11):
Lemma 67. Let \((\mathcal{P}, \phi)\) be a non-probabilistic extended process. Let \(F \in \mathcal{F}\). If \((\mathcal{P}, \phi) \xrightarrow{*} (\mathcal{P}', \phi)\) and \((\mathcal{P}', \phi) \models F\) then \((\mathcal{P}, \phi) \models F'\).
Figure 11: The resolution $R$ for the formula $a.F'$.

Figure 12: The resolution $R$ for the formula $F'_1 \land F'_2$. 
Proof. By induction on the structure of $F$ and by noticing that if $(\mathcal{P}', \phi) \xrightarrow{\alpha} (\mathcal{Q}, \phi')$ in the LTS $\mathcal{L}^f$ then we also have $(\mathcal{P}, \phi) \xrightarrow{\alpha} (\mathcal{Q}, \phi')$ in the LTS $\mathcal{L}^f$. \hfill \Box

**Lemma 68.** Let $\mathcal{P}$ a non-probabilistic process, $\phi$ a frame. Let $F \in \mathcal{F}$. Let $ok \in \mathcal{N}_{pub}$ such that $ok \not\in fn(\mathcal{P}, \phi, F)$. If there exists a pair $(R, s)$ with $R = (Y, corr, r) \in \mathcal{R}_{nr}^o$, such that $corr(s) = (\mathcal{P} \cup \{Adv_{F,n}^{ok}\phi\})$ and $RProb_R(s, corr^{-1}(\downarrow ok)) = 1$ then $(\mathcal{P}, \phi) \models F$.

**Proof.** Let $n = |\phi|$. We do the proof by induction on the structure of $F$.

*Case $F = \top$:* we can see immediately that the result holds as the formula $\top$ holds for all extended processes.

*Case $F = a.F'$: Since $RProb_R(s, corr^{-1}(\downarrow ok)) = 1$ and $ok \not\in fn(\mathcal{P}, \phi, F)$, we deduce that $Adv_{F,n}^{ok} \neq 0$ and $Adv_{F,n}^{ok} \not\in \downarrow ok$. Thus, using Lemma 34 and the fact that $\mathcal{P}$ is a non-probabilistic process and the fact that $R$ is a non-randomized resolution, we deduce that there exist $\mathcal{P}_1, \ldots, \mathcal{P}_k$, $\mathcal{Q}$, $\phi'$ and the following sequence

$$s \xrightarrow{\tau} \delta_{s_1} \quad s_1 \xrightarrow{\tau} \delta_{s_2} \quad \ldots s_{k-1} \xrightarrow{\tau} \delta_{s_k} \quad s_k \xrightarrow{\tau} \delta_{s_a}$$

such that

- for all $i \in \{1, \ldots k\}$, $corr(s_i) = \mathcal{P}_i \cup \{Adv_{F,n}^{ok}\phi\}$
- $\mathcal{P} \xrightarrow{\tau} \delta_{\mathcal{P}_1}$
- for all $i \in \{1, \ldots, k-1\}$, $\mathcal{P}_i \xrightarrow{\tau} \delta_{\mathcal{P}_{i+1}}$
- $corr(s_a) = \mathcal{Q} \cup \{Adv_{F,n}^{ok}\phi'\}$
- $(\mathcal{P}_k, \phi) \xrightarrow{\alpha} \delta_{(\mathcal{Q}, \phi')}$
- $RProb_R(s_a, corr^{-1}(\downarrow ok)) = 1$

Note that here, $k$ can be equal to 0 and in that case $s \xrightarrow{\tau} \delta_{s_a}$. We deduce that $(\mathcal{P}, \phi) \xrightarrow{\alpha} (\mathcal{Q}, \phi')$ in the LTS $\mathcal{L}^f$. Using our inductive hypothesis on $F'$ and the pair $(R, s_a)$, we deduce obtain that $(\mathcal{Q}, \phi') \models F'$. By definition of the validity of a formula, we conclude that $(\mathcal{P}, \phi) \models a.F'$.

*Case $F = F_1 \land F_2$: Since $Adv_{F,n}^{ok} = Adv_{F_1,n}^{ok} + \frac{1}{2} Adv_{F_2,n}^{ok}$, $Adv_{F,n}^{ok} \not\in \downarrow ok$. Hence, using Lemma 34 and the fact that $\mathcal{P}$ is a non-probabilistic process and the fact that $R$ is a non-randomized resolution, we deduce from $RProb_R(s, corr^{-1}(\downarrow ok)) = 1$ that there exist $\mathcal{P}_1, \ldots, \mathcal{P}_k$, $\mathcal{Q}$, $\phi'$ and the following sequence

$$s \xrightarrow{\tau} \delta_{s_1} \quad s_1 \xrightarrow{\tau} \delta_{s_2} \quad \ldots s_{k-1} \xrightarrow{\tau} \delta_{s_k} \quad s_k \xrightarrow{\tau} \frac{1}{2}\delta_{s_{a_1}} + \frac{1}{2}\delta_{s_{a_2}}$$

such that

- for all $i \in \{1, \ldots k\}$, $corr(s_i) = \mathcal{P}_i \cup \{Adv_{F,n}^{ok}\phi\}$

132
\[ \mathcal{P} \xrightarrow{\tau} \delta \mathcal{P} \]

- for all \( i \in \{1, \ldots, k - 1\} \), \( \mathcal{P}_i \xrightarrow{\tau} \delta \mathcal{P}_{i+1} \)

- \( \text{corr}(s_{a_1}) = \mathcal{P}_k \cup \{\text{Adv}_{F_{1,n}}^\phi\} \)

- \( \text{corr}(s_{a_2}) = \mathcal{P}_k \cup \{\text{Adv}_{F_{2,n}}^\phi\} \)

- \( \text{RProb}_R(s_{a_1}, \text{corr}^{-1}(\downarrow \text{ok})) = 1 \)

- \( \text{RProb}_R(s_{a_2}, \text{corr}^{-1}(\downarrow \text{ok})) = 1 \)

By applying our inductive hypothesis on \( F_1 \) and the pair \((R, s_{a_1})\), we obtain \((\mathcal{P}_k, \phi) \models F_1\). Similarly, we also obtain \((\mathcal{P}_k, \phi) \models F_2\). As \((\mathcal{P}, \phi) \xrightarrow{\tau}^* (\mathcal{P}_k, \phi)\), we conclude by applying Lemma 67 that \((\mathcal{P}, \phi) \models F_1 \land F_2\).

**Lemma 12.** Let \( \mathcal{P} \in \mathcal{M}\mathcal{P}^{en} \) and \( \text{ok} \in \mathcal{N}_{pub} \) such that \( \text{ok} \not\in \text{fn}(\mathcal{P}) \). For all formula \( F \in \mathcal{F}\), we have

\[(\mathcal{P}, \emptyset) \models F \iff \text{RProb}_{R, s}(\mathcal{P} \cup \{\text{Adv}_{F_{1,0}}^\phi\}, \downarrow \text{ok}) = 1\]

**Proof.** Direct from Lemmas 30, 66 and 68.

**G.2 Proposition 6**

**Lemma 69.** Let \((\mathcal{P}, \phi)\) and \((\mathcal{P}', \phi')\) be two extended processes such that \(\text{dom}(\phi) = \text{dom}(\phi')\). Let \( A \in \mathcal{M}\mathcal{P} \) such that \(\text{fn}(A) \subseteq \mathcal{N}_{pub} \) and \(\text{fv}(A) \subseteq \text{dom}(\phi)\). If \( \mathcal{P} \cup A \phi \xrightarrow{\tau}^* \mathcal{Q} \) then there exists \( (\mathcal{P}_1, \phi_1) \in \mathcal{SP}_\ell \), an adversarial process \( A_1 \) for \(\text{dom}(\phi_1)\) and a trace \( w \) such that \( \mathcal{Q} = \mathcal{P}_1 \cup A_1 \phi_1, (\mathcal{P}, \phi) \xrightarrow{w} (\mathcal{P}_1, \phi_1) \) and for all recipes \( \xi, \zeta \), if \(\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi_1)\) and \(\xi \phi_1 \models \zeta \phi_1\) then

\[ (\mathcal{P}', \phi') \xrightarrow{w} (\mathcal{P}', \phi_1) \rightarrow_{\xi \models \zeta} (\mathcal{P}', \phi_1') \text{ implies } \mathcal{P}' \cup A \phi' \xrightarrow{\tau}^* \mathcal{P}'_1 \cup A_1 \phi_1' \]

**Proof.** We prove this lemma by induction on the length of the derivation \( \mathcal{P} \cup A \phi \xrightarrow{\tau}^* \mathcal{Q} \).

**Base case** \( \ell = 0 \): In such a case, \( \mathcal{Q} = \mathcal{P} \cup A \phi \). We take \( w = \varepsilon \) and \( A_1 = A \). Let \( \xi, \zeta \) be recipes such that \(\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi')\) and \(\xi \phi_1 \models \zeta \phi_1\). We conclude by noticing that

\[ (\mathcal{P}', \phi') \xrightarrow{w} (\mathcal{P}_1', \phi_1') \rightarrow_{\xi \models \zeta} (\mathcal{P}_1', \phi_1') \text{ implies } \phi' = \phi_1' \text{ and } \mathcal{P}' \xrightarrow{\tau}^* \mathcal{P}_1' \]

Therefore, \( \mathcal{P}' \cup A \phi \xrightarrow{\tau}^* \mathcal{P}_1' \cup A_1 \phi_1' \).

**Inductive step** \( \ell > 0 \): In such a case, we have \( \mathcal{P} \cup A \phi \xrightarrow{\tau}^* \mathcal{Q}_2 \rightarrow \mathcal{Q} \) where the length of the derivation \( \mathcal{P} \cup A \phi \xrightarrow{\tau}^* \mathcal{Q}_2 \) is strictly smaller than \( \ell \). We can apply our inductive hypothesis to obtain that there exist \( (\mathcal{P}_2, \phi_2) \in \mathcal{SP}_\ell \), an adversarial process \( A_2 \) for \(\text{dom}(\phi_2)\) and a trace \( w_2 \) such that \( \mathcal{Q}_2 = \mathcal{P}_2 \cup A_2 \phi_2 \), \( (\mathcal{P}, \phi) \xrightarrow{w_2} (\mathcal{P}_2, \phi_2) \) and for all recipes \( \xi', \zeta' \), if \(\text{vars}(\xi', \zeta') \subseteq \text{dom}(\phi_2)\) and \(\xi' \phi_2 \models \zeta' \phi_2\) then

\[ (\mathcal{P}', \phi') \xrightarrow{w_2} (\mathcal{P}', \phi_2) \rightarrow_{\xi' \models \zeta'} (\mathcal{P}', \phi_2') \text{ implies } \mathcal{P}' \cup A \phi' \xrightarrow{\tau}^* \mathcal{P}_2' \cup A_2 \phi_2' \]
By taking $\xi' = \zeta' \in N_{\text{pub}}$, we have that $(P', \phi') \xrightarrow{w_3} (P'_2, \phi'_2)$ implies $P' \cup A \phi' \rightarrow^* P'_2 \cup A_2 \phi'_2$.

We do a case analysis on the rule applied in $P_2 \cup A_2 \phi_2 \rightarrow Q$.

**Case of an internal reduction on $P_2$:** In such a case $P_2 \rightarrow P_1$ and $P_Q = P_1 \cup A_2 \phi_2$. Let $A_1 = A_2$, $\phi_1 = \phi_2$ and $w = w_2$. Note that $P_2 \rightarrow P_1$ implies $(P_2, \phi_2) \rightarrow_{\tau} (P_1, \phi_1)$ hence $(P, \phi) \xrightarrow{w} (P_1, \phi_1)$.

Let $\xi, \zeta$ be recipes such that $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi_1)$. If $(P', \phi') \xrightarrow{w} (P'_1, \phi'_1) \rightarrow \xi \zeta (P'_1, \phi'_1)$ then we can directly rely on our inductive hypothesis to conclude.

**Case of an internal reduction on $A_2 \phi_2$:** In such a case, there exists an adversarial process $A_1$ such that $A_2 \phi_2 \rightarrow A_1 \phi_2$ and $P_Q = P_1 \cup A_1 \phi_2$. To define the trace we need to consider, let us look at the rule $A_2 \phi_2 \rightarrow A_1 \phi_2$.

- **Rule Then:** In such a case, $A_2 = A' \cup \{u = v \text{ then } P \text{ else } Q\}$ and $A_1 = A' \cup \{P\}$.
  
  We define $w = w_2.(u = v)$.

- **Rule Else:** In such a case, $A_1 = A' \cup \{u = v \text{ then } P \text{ else } Q\}$ and $A_1 = A' \cup \{Q\}$.
  
  We define $w = w_2.(u \neq v)$.

- Otherwise, we define $w = w_2$

Let $\phi_1 = \phi_2$ and $P_1 = P_2$. In the case of the rule Then (resp. Else), $A_2 \phi_2 \rightarrow A_1 \phi_2$ implies that $u \phi_2 = v \phi_2$ (resp. $u \phi_2 \neq v \phi_2$). Hence $(P_2, \phi_2) \rightarrow_{(u = v)} (P_1, \phi_1)$ (resp. $(P_2, \phi_2) \rightarrow_{(u \neq v)} (P_1, \phi_1)$) and so $(P, \phi) \xrightarrow{w} (P_1, \phi_1)$.

Let us now take $\xi, \zeta$ recipes such that $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi_1)$ and $(P', \phi') \xrightarrow{w_2} (P'_2, \phi'_2) \rightarrow a (P'_1, \phi'_1)$ with $a$ being either $(u = v)$ or $(u \neq v)$ or $\tau$ following the rule applied on $A_2 \phi_2 \rightarrow A_1 \phi_2$. If $a = \tau$ then we trivially have that $(P'_2, \phi'_2) = (P'_1, \phi'_1)$ and $A_2 \phi_2 \rightarrow A_1 \phi_1$. Hence, $P'_2 \cup A_2 \phi_2 \rightarrow P'_1 \cup A_1 \phi'_1$ and so the result holds.

If $a = (u = v)$ (resp. $(u \neq v)$) then $(P'_2, \phi'_2) \rightarrow a (P'_1, \phi'_1)$ implies that $(P'_2, \phi'_2) = (P'_1, \phi'_1)$ and $u \phi'_1 = v \phi'_1$ (resp. $u \phi'_1 \neq v \phi'_1$). Hence, $A_2 \phi_2 \rightarrow A_1 \phi'_1$ which allows us to conclude.

**Case of the rule (COMM) between $P_2$ (input) and $A_2 \phi_2$ (output):** In such a case, we have $P_2 = P_3 \cup \{\text{in}(c, x).P\}$ and $A_2 = A_3 \cup \{\text{out}(u, v).Q\}$ with $\text{Msg}(v \phi_2), u \phi_2 \rightarrow c$, $P_1 = P_3 \cup \{P\{x \rightarrow v \phi_2\}\}$ and $A_1 = A_3 \cup \{Q\}$. Therefore by denoting $\phi_1 = \phi_2$, we have $(P_2, \phi_2) \rightarrow_{\text{in}(u, v)} (P_1, \phi_1)$. We define $w = w_2.\text{in}(u, v)$.

Let us now take $\xi, \zeta$ recipes such that $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi_1)$ and $(P', \phi') \xrightarrow{w_2} (P'_2, \phi'_2) \rightarrow_{\text{in}(u, v)} (P'_1, \phi'_1)$ with $\xi \zeta (P'_1, \phi'_1)$. By definition, we deduce that $\phi'_1 = \phi'_2$, $P'_2 = P'_3 \cup \{\text{in}(c', x).P\}$, $P'_1 = P'_3 \cup \{P\{x \rightarrow v \phi'_1\}\}$, $c' = u \phi'_1$ and $\text{Msg}(v \phi'_1)$. This allows us to conclude that $P'_2 \cup A_2 \phi'_2 \rightarrow P'_1 \cup A_1 \phi'_1$.

**Case of the rule (COMM) between $P_2$ (output) and $A_2 \phi_2$ (input):** In such a case, we have $P_2 = P_3 \cup \{\text{out}(c, t).P\}$ and $A_2 = A_3 \cup \{\text{in}(u, ax_{n+1}).Q\}$ (w.l.o.g. we rename the variable to $ax_{n+1}$ if $|\text{dom}(\phi_1)| = n$) with $\text{Msg}(t), c = u \phi_1$, $P_1 = P_3 \cup \{P\}$ and $A_1 = A_3 \cup \{Q\{x \rightarrow t\}\}$. 134
As $\text{Msg}(t)$ and $c \doteq u\phi_1$, we can notice that $(P_2, \phi_2) \rightarrow_{\text{out}(u,ax_{n+1})} (P_3 \cup \{P\}, \phi_2 \{ax_{n+1} \rightarrow t\})$. By defining $w = w_2 \cdot \text{out}(u,ax_{n+1})$, $\phi_1 = \phi_2 \{ax_{n+1} \rightarrow t\}$, we have $(P, \phi) \xrightarrow{w} (P_1, \phi_1)$.

Let us now take $\xi, \zeta$ recipes such that $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi_1)$ and $(P', \phi') \xrightarrow{w_2} (P_2', \phi_2') \rightarrow_{\text{out}(u,ax_{n+1})} (P'_1, \phi'_1) \rightarrow_{\xi \circ \zeta} (P'_1, \phi'_1)$. By definition, $P' = P'_3 \cup \{\text{out}(c, t'), P''\}$, $\text{Msg}(t')$, $c' = u\phi'_1$, $P'_1 = P'_3 \cup \{P''\}$ and $\phi'_1 = \phi'_2 \{ax_{n+1} \rightarrow t\}$. This allows us to obtain $P'_2 \cup A_2 \phi'_2 \rightarrow P'_1 \cup A_1 \phi'_1$.

**Proposition 6.** Let $P, Q \in \mathcal{MP}^{\text{np}}$.

$$(P, \emptyset) \leq_{\text{tr}} (Q, \emptyset) \iff \forall \text{Adv} \in \mathcal{MP}^{\text{np}} \text{ s.t. } \text{fn}(\text{Adv}) \subseteq \mathcal{N}_{\text{pub}} \text{.} \forall c \in \mathcal{N}_{\text{pub}} \text{.} \text{RProb}_{\text{R}}(P \cup \text{Adv}, \downarrow c) \leq \text{RProb}_{\text{R}}(Q \cup \text{Adv}, \downarrow c)$$

**Proof.** To show that may-testing implies trace equivalence, we rely on Lemma 12. Indeed, a trace $w = a_1 \ldots a_n$ can be seen as a formula $F = a_1.a_2.\ldots.a_n.\top$. In particular, $(P, \emptyset) \xrightarrow{w} (P_1, \phi_1)$ if and only if $(P, \emptyset) \models F$. Hence, by taking $ok \in \mathcal{N}_{\text{pub}}$ such that $ok \notin \text{fn}(w, P, Q)$, we can apply Lemma 12 to obtain that $\text{RProb}_{\text{R}, \text{ok}}(P \cup \text{Adv}_{F,0}, \downarrow ok) = 1$. By hypothesis, we deduce that $\text{RProb}_{\text{R}, \text{ok}}(Q \cup \text{Adv}_{F,0}, \downarrow ok) = 1$ and so $(Q, \emptyset) \models F$ once again by Lemma 12. This allows us to conclude that $(P, \emptyset) \xrightarrow{w} (Q, \phi_1')$.

To show that trace equivalence implies may-testing, let us consider a non-probabilistic $A \in \mathcal{MP}$ such that $\text{fn}(A) \subseteq \mathcal{N}_{\text{pub}}$. Let $c \in \mathcal{N}_{\text{pub}}$.

Assume that $\text{RProb}_{\text{R}_{\text{nu}}}(P \cup A, \downarrow c) \neq 0$ (otherwise the result trivially holds). Thus $\text{RProb}_{\text{R}_{\text{nu}}}(P \cup A, \downarrow c) = 1$ and so $P \cup \text{Adv} \rightarrow P' \in \downarrow c$. By Lemma 69 there exists $(P_1, \phi_1) \in \mathcal{SP}_\ell$, and adversarial process $\text{Adv}_1$ for $\text{dom}(\phi_1)$ and a trace $w$ such that $P' = P_1 \cup \text{Adv}_1 \phi_1$, $(P, \emptyset) \xrightarrow{w} (P_1, \phi_1)$ and for all recipes $\xi, \zeta$, if $\text{vars}(\xi, \zeta) \subseteq \text{dom}(\phi_1)$ and $\xi \phi_1 \doteq \zeta \phi_1$ then

$$(Q, \emptyset) \xrightarrow{w} (Q_1, \phi'_1) \rightarrow_{\xi \circ \zeta} (Q_1, \phi'_1) \text{ implies } Q \cup \text{Adv} \rightarrow^* Q_1 \cup \text{Adv}_1 \phi'_1 \quad (20)$$

If $\text{Adv}_1 \phi_1 \in \downarrow c$ then $\text{Adv}_1 = \text{Adv}_2 \cup \{\text{out}(u, t).P\}$ with $u \phi_1 \doteq c$. Consider the trace $w' = w.(u \doteq c)$. We thus have $(P, \emptyset) \xrightarrow{w'} (P_1, \phi_1)$. Since $(P, \emptyset) \leq_{\text{tr}} (Q, \emptyset)$ then there exists $(Q, \emptyset) \xrightarrow{w.(u \doteq c)} (Q_2, \phi_2)$. Therefore, we have $(Q, \emptyset) \xrightarrow{w} (Q_1, \phi'_1) \rightarrow_{\xi \circ \zeta} (Q_1, \phi'_1) \rightarrow (Q_2, \phi'_2)$. We deduce that $Q \cup A \rightarrow^* Q_1 \cup A_1 \phi'_1$ and $u \phi_1 \doteq c$. Therefore, $A_1 \phi'_1 \in \downarrow c$ and so $Q_1 \cup A_1 \phi'_1 \in \downarrow c$. This allows us to conclude that $\text{RProb}_{\text{R}_{\text{nu}}}(Q \cup A, \downarrow c) = 1$.

If $P_1 \in \downarrow c$ then $P_1 = P_2 \cup \{\text{out}(u, t).P\}$ with $u \doteq c$ and $\text{Msg}(t)$. Hence $(P_1, \phi_1) \rightarrow_{\text{out}(c,ax_{n+1})} (P_2 \cup \{P\}, \phi_1 \{ax_{n+1} \rightarrow t\})$. Since $(P, \emptyset) \leq_{\text{tr}} (Q, \emptyset)$ then there exists $(Q, \emptyset) \xrightarrow{w.(u \doteq c)} (Q_3, \phi'_3)$. Therefore, $(Q, \emptyset) \xrightarrow{w} (Q_1, \phi'_1) \rightarrow_{\text{out}(c,ax_{n+1})} (Q_2, \phi'_2) \rightarrow (Q_3, \phi'_3)$. But $(Q, \emptyset) \xrightarrow{w.(u \doteq c)} (Q_1, \phi'_1) \rightarrow_{\text{out}(c,ax_{n+1})} (Q_2, \phi'_2) \rightarrow (Q_3, \phi'_3)$. Hence $Q_1 \in \downarrow c$. By applying Equation (20) with $\xi = \zeta = c$, we deduce that $Q \cup \text{Adv} \rightarrow^* Q_1 \cup \text{Adv}_1 \phi'_1$ which allows us to conclude that $\text{RProb}_{\text{R}_{\text{nu}}}(Q \cup A, \downarrow c) = 1$. 

\[135\]
H Fully probabilistic processes

Lemma 16. Let $ok \in N_{pub}$. Let $(P, \phi)$ be a purely probabilistic process. Let $Adv$ be a fully determinate adversarial process such that $fv(Adv) \subseteq dom(\phi)$.

$$RProb_{R_{\omega}(N^*)}(P \cup Adv\phi, \downarrow ok) = RProb_{R_{\omega}(N^*)}(P, \downarrow ok) + (1 - RProb_{R_{\omega}(N^*)}(P, \downarrow ok)) \times \sum_{(\alpha, w) \in T_{\omega}(Adv, \downarrow dom(\phi)))} \alpha \cdot Prob_{R_{\omega}(N^*)}((P, \phi), w)$$

Proof. Using the property of maximal schedulers of Proposition 10, we know that there exists a maximal resolution where

Moreover we can suppose that $corr_R$ is simply the identity. It is because, we can see that there always exists a maximal resolution where $corr_R$ is the identity. Hence, we obtain:

$$RProb_{R_{\omega}(N^*)}(P \cup Adv\phi, \downarrow ok) = RProb_R(P \cup Adv\phi, \downarrow ok)$$

We will show in fact that from $R$, we can build a maximal resolution in $R_{\omega}(N^*)$, that we denote $B_\omega(R)$ such that:

Finally, as we only consider finite processes without replication, if we restrict the sets of $R$ to the processes reachable from $P \cup Adv\phi$, i.e. $P \cup Adv\phi \rightarrow^* P'$ then $R$ is finite. Thus:

$$RProb_R(s, T) = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \notin T \land trans(s) = \ast \\ \sum_{u \in supp(D)} D(u) \cdot RProb_R(u, T) & \text{if } s \notin T \land trans(s) = D \end{cases}$$

We do a proof by induction on the sub-scheduler starting from $P \cup Adv\phi$.

Case $P \cup Adv\phi \in \downarrow ok$: In such a case, either there exists $out(u, v); P \in P$ such that $u \not= ok$ or $Adv = out(u', v')$; $Adv'$ with $u' \not= ok$. In the former case, we deduce that for any maximal scheduler $R'$ starting from $P$, $RProb_R(P, \downarrow ok) = 1$. By taking $B_0(R) = R'$ and $B_1(R)$ any maximal scheduler starting from $(P, \phi)$, we conclude. In the latter case, we have by construction that $Tr^ok(Adv\phi, \downarrow dom(\phi)) = \{(p_i, (u', (\alpha_0, x_{n+1})).w_i)\}_{i=1}^m \cup \{(1, u' \not= ok)\}$. We take all $i \in \{1, \ldots, m\}$, $Prob_{R_{\omega}(N^*)}((P, \phi), (u' \not= ok)$. Hence, by taking any maximal scheduler $B_1(R)$, we have:

By taking any maximal scheduler $B_0(R)$ and by denoting $\beta = RProb_{B_0}(P, \downarrow ok)$, we have:

$$RProb_R(P \cup Adv\phi, \downarrow ok) = \beta + (1 - \beta) \times \sum_{(\alpha, w) \in T_{\omega}(Adv, \downarrow dom(\phi)))} \alpha \cdot Prob_{B_1(R)}((P, \phi), (u' \not= ok)) = 1$$

136
Case $\mathcal{P} \cup \text{Adv} \not\downarrow \text{ok}$ and trans($\mathcal{P} \cup \text{Adv}$) = $\ast$: As $R$ is maximal, we deduce that there is no possible transition on $\mathcal{P} \cup \text{Adv}$. Hence, all processes in $\mathcal{P}$ starts by an input or an output (no $\tau$ transition) and either $\text{Adv} = 0$ or $\text{Adv} = \text{in}(c, x)$; $\text{Adv}'$ or $\text{Adv} = \text{out}(c, u)$; $\text{Adv}'$ with $c \phi \neq c_i$ for all $i \in \{1, \ldots, n\}$. Moreover, when $\text{Adv} = \text{out}(c, u)$; $\text{Adv}'$, $c \phi \neq \text{ok}$. By construction, we deduce that for all $(\alpha, w) \in Tr^\text{ok}(\text{Adv}, |\text{dom}(\phi)|)$, $\text{Prob}_{\mathcal{R}_{\text{w}}(N^2)}((\mathcal{P}, \phi), w) = 0$. Hence by taking any maximal scheduler as $B_\ell(R)$, we deduce that:

$$\sum_{(\alpha, w)\in Tr^\text{ok}(\text{Adv}, |\text{dom}(\phi)|)} \alpha \cdot \text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), w) = 0$$

Moreover, since all processes in $\mathcal{P}$ starts by an input or an output (no $\tau$ transition) and they have distinct channels, we deduce that for any maximal scheduler $B_\ell(R)$ from $\mathcal{P}$ with transition function $\text{trans}'$, we have $\text{trans}'(\mathcal{P}) = \ast$. Hence, $\text{RProb}_{B_\ell(R)}(\mathcal{P}, \downarrow \text{ok}) = 0$ which allows us to conclude.

Case $\mathcal{P} \cup \{\text{Adv}\phi\} \not\downarrow \text{ok}$ and trans($\mathcal{P} \cup \{\text{Adv}\phi\}$) = $D$: In such a case, we have four possibilities with respect to the transition $\text{trans}(\mathcal{P} \cup \{\text{Adv}\phi\}) = D$:

- $\tau$ transition on $\mathcal{P}$: In such a case, we have either $D = \delta_{\mathcal{Q}\cup \{\text{Adv}\phi\}}$ where $\mathcal{P} \xrightarrow{\tau} \mathcal{Q}$ is either conditional or a name restriction; or $\mathcal{P} = \{P_1 + P_2\} \cup \mathcal{P}'$ and $D = p \cdot \delta_{\mathcal{Q}_1 \cup \{\text{Adv}\phi\}} + (1 - p) \cdot \delta_{\mathcal{Q}_2 \cup \{\text{Adv}\phi\}}$ with $\mathcal{Q}_1 = \{P_1\} \cup \mathcal{P}'$ and $\mathcal{Q}_2 = \{P_2\} \cup \mathcal{P}'$.

In the first case, by definition, $\text{RProb}_R(\mathcal{Q} \cup \{\text{Adv}\phi\}, \downarrow \text{ok}) = \text{RProb}_R(\mathcal{P} \cup \{\text{Adv}\phi\}, \downarrow \text{ok})$. Moreover, by applying our inductive hypothesis, we deduce that there exist a scheduler $R_\ell$ such that

$$\text{RProb}_R(\mathcal{Q} \cup \{\text{Adv}\phi\}, \downarrow \text{ok}) = \text{RProb}_{\mathcal{R}_{\text{w}}(N^2)}(\mathcal{Q}, \downarrow \text{ok}) + (1 - \text{RProb}_{\mathcal{R}_{\text{w}}(N^2)}(\mathcal{Q}, \downarrow \text{ok})) \times \sum_{(\alpha, w)\in Tr^\text{ok}(\text{Adv}, |\text{dom}(\phi)|)} \alpha \cdot \text{Prob}_{R_\ell}(((\mathcal{Q}, \phi), w))$$

By taking the maximal scheduler $B_\ell(R)$ that connects $(\mathcal{P}, \phi)$ to $(\mathcal{Q}, \phi)$ by the transition $(\mathcal{P}, \phi) \xrightarrow{\tau} \delta_{\mathcal{Q}, \phi}$ (and behaves as $R_\ell$ everywhere else), we deduce that for all $(\alpha, w) \in Tr^\text{ok}(\text{Adv}, |\text{dom}(\phi)|)$, $\text{Prob}_{R_\ell}((\mathcal{Q}, \phi), w) = \text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), w)$. Similarly, by Proposition $\text{RProb}_{\mathcal{R}_{\text{w}}(N^2)}(\mathcal{Q}, \downarrow \text{ok}) = \text{RProb}_{R_\ell}(\mathcal{Q}, \downarrow \text{ok})$ for some $R_\ell$ maximal. By taking the maximal scheduler $R''$ that connects $\mathcal{P}$ to $\mathcal{Q}$ by the transition $\mathcal{P} \xrightarrow{\tau} \delta_{\mathcal{Q}}$ (and behaves as $R'$ everywhere else), we deduce that $\text{RProb}_{R''}(\mathcal{Q}, \downarrow \text{ok}) = \text{RProb}_{R''}(\mathcal{P}, \downarrow \text{ok}) = \text{RProb}_{\mathcal{R}_{\text{w}}(N^2)}(\mathcal{P}, \downarrow \text{ok})$. This allows us to conclude that.

Consider the second case i.e., $\mathcal{P} = \{P_1 + P_2\} \cup \mathcal{P}'$ and $D = p \cdot \delta_{\mathcal{Q}_1 \cup \{\text{Adv}\phi\}} + (1 - p) \cdot \delta_{\mathcal{Q}_2 \cup \{\text{Adv}\phi\}}$. In that case, $\text{RProb}_R(\mathcal{P} \cup \{\text{Adv}\phi\}, \downarrow \text{ok}) = p \cdot \text{RProb}_R(\mathcal{Q}_1 \cup \{\text{Adv}\phi\}, \downarrow \text{ok}) + (1 - p) \cdot \text{RProb}_R(\mathcal{Q}_2 \cup \{\text{Adv}\phi\}, \downarrow \text{ok})$. Moreover, $(\mathcal{P}, \phi) \xrightarrow{\tau} p \cdot \delta_{\mathcal{Q}_1, \phi} + (1 - p) \cdot \delta_{\mathcal{Q}_2, \phi}$.

By inductive hypothesis, for all $i \in \{1, 2\}$, there exist schedulers $R_1, R_2$ such that:

$$\text{RProb}_R(\mathcal{Q}_i \cup \{\text{Adv}\phi\}, \downarrow \text{ok}) = \text{RProb}_{\mathcal{R}_{\text{w}}(N^2)}(\mathcal{Q}_i, \downarrow \text{ok}) + (1 - \text{RProb}_{\mathcal{R}_{\text{w}}(N^2)}(\mathcal{Q}_i, \downarrow \text{ok})) \times \sum_{(\alpha, w)\in Tr^\text{ok}(\text{Adv}, |\text{dom}(\phi)|)} \alpha \cdot \text{Prob}_{R_\ell}(((\mathcal{Q}_i, \phi), w))$$
Actually, since by Proposition 11, all maximal resolutions on \( N' \) lead to the same probability of reaching a trace, we can suppose \( R_1 = R_2 \). By taking the scheduler \( B_\ell(R) \) that connects \((\mathcal{P}, \phi)\) to \((\mathcal{Q}_1, \phi)\) with probability \( p \) and \((\mathcal{P}, \phi)\) to \((\mathcal{Q}_2, \phi)\) with probability \( 1 - p \), (and behaves as \( R_1 = R_2 \) elsewhere) we deduce that for all \((\alpha, w) \in T_{rok}(Adv, \text{dom}(\phi))\), 

\[
\text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), w) = p \cdot \text{Prob}_{B_\ell(R)}((\mathcal{Q}_1, \phi), w) + (1 - p) \cdot \text{Prob}_{B_\ell(R)}((\mathcal{Q}_2, \phi), w).
\]

Similarly, we have that \( R_{Prob_{\phi}}(\mathcal{P}, \downarrow ok) = p \cdot R_{Prob_{\phi}}(\mathcal{Q}_1, \downarrow ok) + (1 - p) \cdot R_{Prob_{\phi}}(\mathcal{Q}_2, \downarrow ok) \).

To summarize, we have

\[
- \sum_{(\alpha, w) \in T_{rok}(Adv, \text{dom}(\phi))} \alpha \cdot \text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), w) = p \cdot \beta_1 + (1 - p) \cdot \beta_2
\]

\[
- \beta_i = \sum_{(\alpha, w) \in T_{rok}(Adv, \text{dom}(\phi))} \alpha \cdot \text{Prob}_{R_1}((\mathcal{Q}_i, \phi), w) \text{ for } i = 1, 2
\]

\[
- R_{Prob_{\phi}}(\mathcal{P}, \downarrow ok) = p \cdot \gamma_1 + (1 - p) \cdot \gamma_2
\]

\[
- \gamma_i = R_{Prob_{\phi}}(\mathcal{Q}_i, \downarrow ok) \text{ for } i = 1, 2
\]

As \( R_{Prob_R}(\mathcal{P} \cup \{Adv\phi\}, \downarrow ok) = p \cdot R_{Prob_R}(\mathcal{Q}_1 \cup \{Adv\phi\}, \downarrow ok) + (1 - p) \cdot R_{Prob_R}(\mathcal{Q}_2 \cup \{Adv\phi\}, \downarrow ok) \).

By inductive hypothesis, for all \( i \in \{1, 2\} \), there exist two maximal schedulers \( R_1, R_2 \) such that:

\[
R_{Prob_R}(\mathcal{P} \cup \{Adv\phi\}, \downarrow ok) = R_{Prob_{\phi}}(\mathcal{P}, \downarrow ok) + (1 - R_{Prob_{\phi}}(\mathcal{P}, \downarrow ok)) \times \sum_{k=1}^{n_1} p_k^1 \cdot \text{Prob}_{R_1}((\mathcal{P}, \phi), w_k^1)
\]

Note that thanks to our lemma on maximal resolutions, we can in fact suppose that \( R_1 = R_2 \) since \( \text{Prob}_{R_1}((\mathcal{P}, \phi), w') = \text{Prob}_{R_2}((\mathcal{P}, \phi), w') \) for all \( w' \). Let us take \( B_\ell(R) = R_1 \). Thus, we obtain that:

\[
R_{Prob_R}(\mathcal{P} \cup \{Adv\phi\}, \downarrow ok) = R_{Prob_{\phi}}(\mathcal{P}, \downarrow ok) + (1 - R_{Prob_{\phi}}(\mathcal{P}, \downarrow ok)) \times (p \cdot \sum_{k=1}^{n_1} p_k^1 \cdot \text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), w_k^1) + (1 - p) \cdot \sum_{k=1}^{n_2} p_k^2 \cdot \text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), w_k^2))
\]

Since \( T_{rok}(Adv_1 + p\ Adv_2, n) = \{((p \cdot p_k^1, w_k^1)\}_{k=1}^{n_1} \cup \{((1 - p) \cdot p_k^2, w_k^2)\}_{k=1}^{n_2} \) we conclude.
• Name restriction on \(Adv\phi\): In such a case, \(Adv = \text{new } a; Adv'\) with \(Tr_{\alpha{k}}(Adv, n) = \text{Tr}_{\alpha{k}}(Adv', \{b/a\}, n)\) and \(b \in N_{\text{pub}}\) fresh. Moreover, by definition:

\[
\text{RProb}_{\mathcal{R}}(\mathcal{P} \cup \{Adv\phi\}, \downarrow ok) = \text{RProb}_{\mathcal{R}}(\mathcal{P} \cup \{Adv'\{b/a\}\phi\}, \downarrow ok)
\]

By applying the inductive hypothesis on \(\mathcal{P} \cup \{Adv\{b/a\}\phi\}\), we directly conclude.

• Conditional branching on \(Adv\phi\). We focus on the case where the transition corresponds to the rule \text{THEN} (the case of the rule \text{ELSE} is similar). Thus \(Adv = \text{if } u = v \text{ then } Adv_1 \text{ else } Adv_2\) and \(Tr_{\alpha{k}}(if u = v \text{ then } Adv_1 \text{ else } Adv_2, n) = \{(p^1_k, (u \neq v).w^1_k)\}_{k=1}^{n_1} \cup \{(p^2_k, (u \neq v).w^2_k)\}_{k=1}^{n_2}\). Since the rule \text{THEN} was applied, we have that \(u\phi \neq v\phi\).

Hence, for all \(k \in \{1, \ldots, n_2\}\), \(\text{Prob}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}((\mathcal{P}, \phi), (u \neq v).w^2_k) = 0\) and for all \(k \in \{1, \ldots, n_1\}\), \(\text{Prob}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}((\mathcal{P}, \phi), w^1_k) = \text{Prob}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}((\mathcal{P}, \phi), (u \neq v).w^1_k)\). By applying our inductive hypothesis on \(\mathcal{P} \cup \{Adv_1\phi\}\), we directly conclude.

• Internal communication on \(Adv\phi\) and input on \(\mathcal{P}\): In such a case, \(Adv = \text{out}(u, v); Adv'\) and \(Tr_{\alpha{k}}(Adv, n) = \{(p_i, (u \neq ok).in(u, v).w_i)\}_{i=1}^{m} \cup \{(1, u = ok)\}\) when \(Tr_{\alpha{k}}(Adv', n) = \{(p_i, w_i)\}_{i=1}^{m}\). Moreover, \(\mathcal{P} = \mathcal{P'} \cup \{\text{in}(c, x); \mathcal{P}\}\) with \(u\phi = c\).

Finally, we deduce that \(D = \delta_{\mathcal{Q} \cup \{Adv\phi\}'}\) where \(\mathcal{Q} = \mathcal{P'} \cup \{P^\phi_{\phi/x}\}\). Hence, by definition, we deduce that:

\[
\text{RProb}_{\mathcal{R}}(\mathcal{P} \cup \{Adv\phi\}, \downarrow ok) = \text{RProb}_{\mathcal{R}}(\mathcal{Q} \cup \{Adv'\phi\}, \downarrow ok)
\]

Note that since \(\mathcal{P} \cup Adv\phi \not\subseteq \downarrow ok\), we know that \(ok \neq c\) and so \(ok \neq c\). Thus, by definition of the labeled semantics, we have \((\mathcal{P}, \phi) \xrightarrow{(\text{ok } \neq u).\text{in}(u, v)} (\mathcal{Q}, \phi)\). By applying our inductive hypothesis, we deduce that there exist a scheduler \(R_\ell\) such that:

\[
\text{RProb}_{\mathcal{R}}(\mathcal{Q} \cup \{Adv'\phi\}, \downarrow ok) = \text{RProb}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}(\mathcal{Q}, \downarrow ok) + (1 - \text{RProb}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}(\mathcal{Q}, \downarrow ok)) \times \sum_{(\alpha, w) \in \text{dom}(\phi)} \alpha \cdot \text{Prob}_{\mathcal{R}_{\alpha}}((\mathcal{Q}, \phi), w)
\]

Recall that \(ok \neq c\). Hence, by definition of purely probabilisitic processes, \(\text{RProb}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}(\mathcal{Q}, \downarrow ok) = \text{RProb}_{\mathcal{R}_{\alpha{m}'}(\mathcal{N}_n)}(\mathcal{P}, \downarrow ok)\).

By taking the scheduler \(B_\ell(R)\) that connect \((\mathcal{P}, \phi)\) to \((\mathcal{Q}, \phi)\) by the transitions \((\mathcal{P}, \phi) \xrightarrow{\text{in}(u, v)} (\mathcal{Q}, \phi)\) and then executes \(R_\ell\), we deduce that for all \(i \in \{1, \ldots, m\}\), \(\text{Prob}_{\mathcal{R}_{\alpha}}((\mathcal{Q}, \phi), w_i) = \text{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), (ok \neq u).\text{in}(u, v).w)\) which allows us to conclude.

• Internal communication with input on \(Adv\phi\) and output on \(\mathcal{P}\): In such a case, \(Adv = \text{in}(u, x); Adv'\) and \(Tr_{\alpha{k}}(Adv, n) = \{(p_i, (u \neq ok).\text{out}(u, ax_{i+1}.w_i)\}_{i=1}^{m}\) when \(Tr_{\alpha{k}}(Adv', n+1) = \{(p_i, w_i)\}_{i=1}^{m}\). Moreover, \(\mathcal{P} = \mathcal{P} \cup \{\text{out}(c, v); \mathcal{P}\}\) with \(u\phi = c\). Finally, we deduce that \(D = \delta_{\mathcal{Q} \cup \{Adv\phi\}'}\) where \(\mathcal{Q} = \mathcal{P} \cup \{P\}\) and \(\phi' = \phi\{ax_{i+1} \mapsto v\}\). Hence, by definition, we deduce that:

\[
\text{RProb}_{\mathcal{R}}(\mathcal{P} \cup \{Adv\phi\}, \downarrow ok) = \text{RProb}_{\mathcal{R}}(\mathcal{Q} \cup \{Adv'\phi'\}, \downarrow ok)
\]
Note that since $\mathcal{P} \cup Adv\phi \not\downarrow \text{ok}$, we know that $\text{ok} \neq c$ and so $\text{ok} \neq c$. Thus, by definition of the labeled semantics, we have $(\mathcal{P}, \phi) \xrightarrow{(\text{ok} \neq u).\text{out}(u, ax_{n+1})} \delta(Q, \phi')$. By applying our inductive hypothesis, we deduce that there exists a scheduler $R_\ell$ such that

$$R\text{Prob}_{R}(Q \cup \{Adv'\phi\}, \downarrow \text{ok}) = R\text{Prob}_{R(N^\circ)}(Q, \downarrow \text{ok}) + (1 - R\text{Prob}_{R(N^\circ)}(Q, \downarrow \text{ok})) \times \sum_{(\alpha, w) \in T_{\text{ok}}(Adv', \text{dom}(\phi'))} \alpha \cdot \operatorname{Prob}_{R_\ell}((Q, \phi'), w)$$

Recall that $\text{ok} \neq c$. Hence, by definition of purely probabilistic processes, $R\text{Prob}_{R(N^\circ)}(Q, \downarrow \text{ok}) = R\text{Prob}_{R(N^\circ)}(P, \downarrow \text{ok})$.

By taking the scheduler $B_\ell(R)$ that connect $(\mathcal{P}, \phi)$ to $(Q, \phi')$ by the transition $(\mathcal{P}, \phi) \xrightarrow{\text{out}(u, ax_{n+1})} (Q, \phi')$ and then executes $R_\ell$, we deduce that for all $i \in \{1, \ldots, m\}$,

$$\operatorname{Prob}_{R_i}((Q, \phi'), w_i) = \operatorname{Prob}_{B_\ell(R)}((\mathcal{P}, \phi), (\text{ok} \neq u).\text{out}(u, ax_{n+1}).w_i)$$

which allows us to conclude.

$\square$