Symbolic protocol verification with dice

Process equivalences in the presence of probabilities

Vincent Cheval [a], Raphaëlle Crubillé [b] and Steve Kremer [c]

[a] *Inria Paris, France*
E-mail: vincent.cheval@inria.fr
[b] *CNRS, LIS, Aix Marseille Univ, Université de Toulon, France*
E-mail: raphaelle.crubille@lis-lab.fr
[c] *Inria Nancy, LORIA,Université de Lorraine, France*
E-mail: steve.kremer@inria.fr

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 π-calculus. We define and explore the relationships between several behavioral equivalences. In particular we show the need for randomized schedulers and exhibit a counter-example to a result 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. In particular, 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.

Keywords: security protocols, symbolic verification, probabilistic process equivalences

1. Introduction

Automated symbolic protocol verification, based on the seminal work of Dolev and Yao [1], has nowadays reached a level of maturity enabling successful use on complex real-world security protocols, including TLS [2, 3], Signal [4], authentication protocols of the 5G standard [5], or EMV's secure payment protocols [6] 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 $dec(enc(m, k), k) = m$ 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 [7] (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 [8], mix-nets [9] or Crowds [10]. In this paper, we will investigate indistinguishability properties, expressed as equivalences in a cryptographic process calculus, the applied π-calculus [11], 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 a probabilistic attacker (even on non-probabilistic protocols). 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.

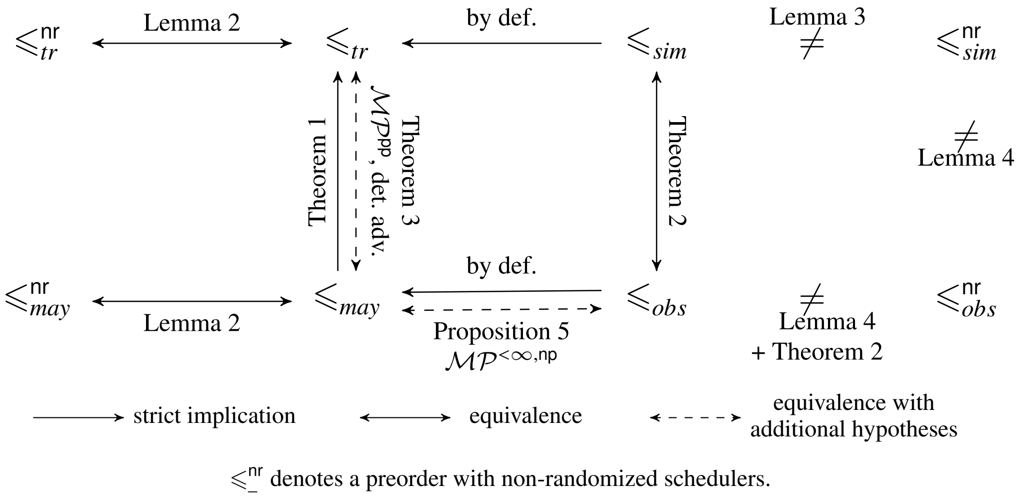

$\leqslant^{\mathsf{nr}}_{tr}$
Lemma 2
$\leqslant_{tr}$
by def.
$\leqslant_{sim}$
Lemma 3
$\neq$
$\leqslant^{\mathsf{nr}}_{sim}$
Theorem 1
Theorem 3
$\mathcal{MP}^{\mathsf{pp}}$, det. adv.
Theorem 2
$\neq$
Lemma 4
by def.
$\leqslant^{\mathsf{nr}}_{may}$
Lemma 2
$\leqslant_{may}$
Proposition 5
$\mathcal{MP}^{<\infty,\mathsf{np}}$
$\leqslant_{obs}$
$\neq$
Lemma 4
+ Theorem 2
$\leqslant^{\mathsf{nr}}_{obs}$
strict implication
equivalence
equivalence with additional hypotheses
$\leqslant^{\mathsf{nr}}_{\_}$ denotes a preorder with non-randomized schedulers.

Figure 1. Summary of the relationship between preorders.

Our contributions. In a first part we introduce a probabilistic applied π-calculus and its semantics, which has similarities to [12], 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 [13–19]. *(ii)* More importantly, we differ in the way non-determinism is resolved: unlike [12] 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.

Second, we define several notions of preorders and equivalences and study their relations. The main results are also summarized in Figure 1, focusing on preorders (with similar relations between corresponding equivalences). We show, in particular, that

- unlike in the purely non-deterministic case, the may-testing preorder ($\leqslant_{may}$) is strictly stronger than the trace equivalence preorder ($\leqslant_{tr}$) (Theorem 1);
- simulation ($\leqslant_{sim}$) and observational pre-order ($\leqslant_{obs}$), respectively bisimilarity and observational equivalence, coincide for randomized schedulers (Theorem 2);
- for non-randomized schedulers, these equivalences ($\leqslant_{sim}^{\mathsf{nr}}$ and $\leqslant_{obs}^{\mathsf{nr}}$) do *not* coincide (Lemma 4), which provides a counter-example to one of the main results in [12].

Third, a well-known phenomenon [20, 21] 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.

We first consider the classical class of *non-probabilistic processes* (denoted $\mathcal{MP}^{\mathsf{np}}$), as in the original applied π-calculus, but in the presence of probabilistic adversaries. We show that, if we additionally bound the number of sessions (denoted $\mathcal{MP}^{<\infty,\mathsf{np}}$),

- may-testing with probabilistic adversaries coincides with the classical, purely possibilistic notion of similarity (Proposition 5 and Theorem 2). This also provides a contextual characterization of the notion of similarity which is reminiscent of [17] in the setting of CSP;
- verification of testing equivalence with probabilistic adversaries is co-NEXPTIME complete for a large class of cryptographic primitives, relying on results from [22].

We next consider a class of *purely probabilistic* processes with a bounded number of sessions (denoted $\mathcal{MP}^{\mathsf{pp}}$), which is reminiscent of a probabilistic version of simple processes in [23, 24], and a slight generalization of the processes in [25]. We show that trace equivalence as considered in [25], is

- weaker than may-testing, but
- coincides with a version of may-testing with *determinate attackers*: attacker processes are restricted by disallowing replication, parallel, and non-deterministic choice, but allowing probabilistic choices (Theorem 3).

Next, we present an algorithm for deciding trace equivalence by extending the procedure of the DeepSec verifier [22]. Our procedure inherits some limitations (bounded number of sessions, the class of admissible rewrite systems) but provides a more general setting than Bauer *et al.* [26] who additionally bound the size of input messages. We have implemented our procedure in the open-source tool DeepSec available at [27].

Finally, we illustrate our framework by studying the Dining Cryptographers protocol. We notably provide a pen and paper proof that the protocol guarantees anonymity (expressed as may-testing equivalence). Using the DeepSec tool we also show that anonymity is not provided when coins are biased.

For readability, we often only provide proof sketches and omit some of the most technical proofs. A full version with more detailed proofs is available at [28].

2. Probabilistic Applied π-calculus

In this section we introduce the probabilistic applied π-calculus, a probabilistic variant of the applied π-calculus introduced by Goubault-Larrecq *et al.* [12].

2.1. Message as terms

Atomic values such as keys and nonces are modelled by *names*. We assume an infinite set of such names $\mathcal{N} = \{a, b, \dots, \}$ and partition it into two disjoint infinite sets $\mathcal{N}_{pub}$ and $\mathcal{N}_{priv}$. The set of *private* names $\mathcal{N}_{priv}$ is a priori unknown to the attacker and models, *e.g.*, honest keys in a protocol. The set of *public* names $\mathcal{N}_{pub}$ models public values, known to the attacker. The distinction between public and private names is analogous to the distinction between free and bound names in the original applied π-calculus. We also define an infinite set of variables $\mathcal{X}$. Finally, we consider a finite set of *function symbols* each equipped with their arity $\mathcal{F} = \{f/n, g/m, \dots\}$. Function symbols model cryptographic operations, *e.g.*, $enc/2$ is a binary symbol that could be used to model encryption. *Terms* are defined as names, variables, and function symbols applied to other terms. For instance, given two names $a, k \in \mathcal{N}$, $enc(a, k)$ represents the encryption of a with the key k. For any $\mathsf{F} \subseteq \mathcal{F}$, $\mathsf{N} \subseteq \mathcal{N}$ and $\mathsf{V} \subseteq \mathcal{X}$, the set of terms built from N and V by applying function symbols in F is denoted by $\mathcal{T}(\mathsf{F}, \mathsf{N} \cup \mathsf{V})$.

We also suppose that terms are equipped with a binary relation $\doteq$ that expresses that two terms evaluate to the same result, and a predicate $Msg(\cdot)$ that is intended to hold when evaluation succeeds. How $\doteq$ and $Msg(\cdot)$ are precisely defined is not relevant for the results of this paper and we wish to capture several formalisms. $\doteq$ can for instance be defined by an equational theory, as in the applied π-calculus [11] (where $Msg(\cdot)$ would evaluate to true on any term), by a constructor-destructor rewrite system, allowing evaluation to fail when a destructor application does not reduce, as in the DEEPSEC tool [22], or a combination of these as in the ProVerif tool [29].

Formally, we require that $\doteq$ is symmetric, transitive, and closed under substitution of terms for names and variables, as well as application of function symbols. Moreover, for all $a, b \in \mathcal{N}$, $a = b$ if and only if $a \doteq b$. $Msg(\cdot)$ is supposed to hold on any names, be closed under renamings and $t_1 \doteq t_2$ implies that $Msg(t_1)$ and $Msg(t_2)$. Finally, we require that $Msg(t)$ implies $t \doteq t$.

For example, the $\doteq$ relation could capture that $dec(enc(m, k), k) \doteq m$ for any m, k modelling that decryption cancels out encryption when the same key k is used; one may also define $Msg(dec(n, k))$ as false to express that decryption fails if the ciphertext argument is not an encryption with the matching key.

2.2. Syntax of the process calculus

The syntax for *processes* is defined as follows:

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

where $u, v \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$, $x \in \mathcal{X}$, $a \in \mathcal{N}$ and $p \in]0;1[$. As usual, in examples we will omit trailing 0 processes and else 0 branches. A process P is closed when all variables in P are bound by an input. We also denote by $fn(P)$ the set of free names in P, i.e., the names not bound by a restriction, and by $n(P)$ the set of all names occurring in P.

Example 1. As an example, consider the process P:

$$(\mathsf{out}(c,k) +_{\frac{1}{3}} \mathsf{out}(c,a)) \mid \mathsf{in}(c,x); \mathsf{if}\ x = k\ \mathsf{then}\ \mathsf{out}(c, ok)$$

P consists of two parallel processes. The left process outputs on a channel c with probability $\frac{1}{3}$ the name k and with probability $\frac{2}{3}$ the name a. The right process inputs a value on channel c and binds this value to x. If x equals k then it outputs the constant ok.

We denote by $\mathcal{SP}$ the set of all processes in the probabilistic applied π-calculus, and by $\mathcal{MP}$ the set of all multisets over $\mathcal{SP}$.

2.3. Operational semantics

We will now define the semantics of the probabilistic applied π-calculus. We opt for a different presentation of the semantics than Goubault-Larrecq *et al.* [12] relying on existing formalisms for transition systems. Moreover, we allow for a more general class of schedulers.

Notation 1. Let $\mathcal{S}$ be an arbitrary set. We denote by $\mathcal{D}(\mathcal{S})$ the set of all finitely supported probability distributions over $\mathcal{S}$ and by $\mathcal{D}^{\leqslant 1}(\mathcal{S})$ the set of all sub-probability distributions over $\mathcal{S}$ (observe that $\mathcal{D}(\mathcal{S}) \subseteq \mathcal{D}^{\leqslant 1}(\mathcal{S})$). For $p, q \geqslant 0$, and sub-distributions D, E, we define the measure

$$(p \cdot D + q \cdot E)(x) = p \cdot D(x) + q \cdot E(x).$$

When $q = 0$, 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 $D \in \mathcal{D}(\mathcal{S})$, we denote by $\mathrm{supp}(D)$ the support of D, *i.e.*, the set of all elements $s \in \mathcal{S}$ such that $D(s) > 0$. If $\mathcal{S}' \subseteq \mathcal{S}$, we define $D(\mathcal{S}') = \sum_{s \in \mathcal{S}'} D(s)$. Finally, we denote by δ_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 $\mathcal{P} \to_\tau \mu$. 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 π-calculus [30] where the non-deterministic choice operator is resolved externally. Note that the original applied π-calculus [11] does not contain non-deterministic choice.

In the following, we define the operational semantics of our calculus using well studied probabilistic systems. We choose the formalism of *non-deterministic probabilistic labelled transition systems* (NPLTS) used for instance in [16]. A NPLTS allows to represent states that allow 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.* [13].

$$\begin{array}{ll}
\mathcal{P} \cup \{\!\{0\}\!\} \to_\tau \delta_{\mathcal{P}} & \\
\mathcal{P} \cup \{\!\{\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{P\}\!\}} & \text{if } u \doteq v \\
\mathcal{P} \cup \{\!\{\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{Q\}\!\}} & \text{if } u \not\doteq v \\
\mathcal{P} \cup \{\!\{\mathsf{out}(u,t).P, \mathsf{in}(v,x).Q\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{P, Q\{x \mapsto t\}\}\!\}} & \text{if } Msg(t) \wedge u \doteq v \\
\mathcal{P} \cup \{\!\{P \mid Q\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{P,Q\}\!\}} & \\
\mathcal{P} \cup \{\!\{!P\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{!P,P\}\!\}} & \\
\mathcal{P} \cup \{\!\{\mathsf{new}\ a; P\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{P\{^{a'}/_{a}\}\}\!\}} & \text{where } a' \in \mathcal{N}_{priv} \text{ is fresh} \\
\mathcal{P} \cup \{\!\{P + Q\}\!\}) \to_\tau \delta_{\mathcal{P} \cup \{\!\{P\}\!\}} & \\
\mathcal{P} \cup \{\!\{P + Q\}\!\} \to_\tau \delta_{\mathcal{P} \cup \{\!\{Q\}\!\}} & \\
\mathcal{P} \cup \{\!\{P +_p Q\}\!\} \to_\tau p \cdot \delta_{\mathcal{P} \cup \{\!\{P\}\!\}} + (1-p) \cdot \delta_{\mathcal{P} \cup \{\!\{Q\}\!\}} &
\end{array}$$

Figure 2. Semantics of the calculus

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

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

The label τ is the *internal action* and the labels in $\mathcal{A}_{ext}$ are the *external* actions. For $s \in \mathcal{S}, a \in \mathcal{A}$, we write $s \xrightarrow{a} D$ when $D \in \mathsf{trans}(s, a)$.

In the remaining of this paper, we may define a NPLTS by only its transition function, *i.e.*, we will say that $\mathsf{trans} : \mathcal{S} \to \mathcal{A} \to \mathcal{P}(\mathcal{D}(\mathcal{S}))$ is the NPLTS $(\mathcal{S}, \mathcal{A}, \mathsf{trans})$.

We now express our operational semantics as a NPLTS without external actions, *i.e.*, $\mathcal{A}_{ext} = \varnothing$. External actions will be used to express our labeled semantics in Section 4.1.

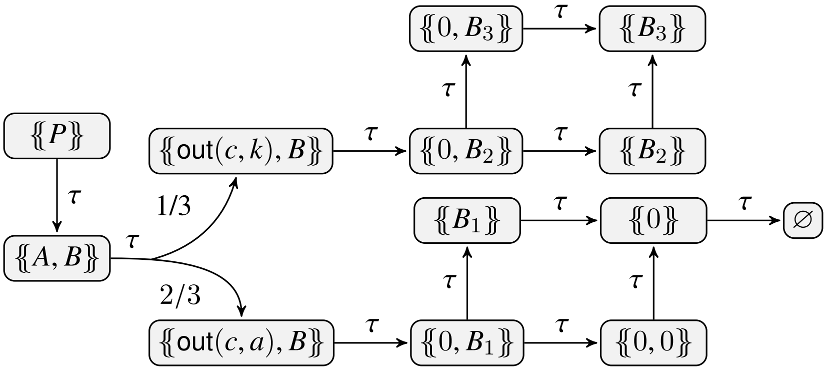
Definition 2. The *operational semantics* is the NPLTS $\mathsf{N}^o = (\mathcal{MP}, \{\tau\}, \mathsf{trans}^o)$ where for every $s \in \mathcal{MP}$, $\mathsf{trans}^o(s)(\tau) = \{D \mid s \to_\tau D\}$.

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

Example 2. The complete execution of the process P, introduced in Example 1 is given in Figure 3.

3. Behavioral equivalences

In this section we define probabilistic versions of two classical equivalences: *may-testing* and the stronger *observational equivalence*. In order to do so we first introduce the notion of *resolution* (also known as scheduler), *i.e.*, how internal non-determinism is resolved, and the notion of *barb*, which models an observational action.



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

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

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

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

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

Figure 3. Semantics of the process P from Example 2

3.1. Resolving the internal non-determinism

Resolutions express how internal non-determinism of a NPLTS is resolved. Intuitively, resolving the non-determinism means, for each state, restricting the transition system either by choosing one particular internal transition, or else by leaving the choice of a non-deterministic external action. The resulting transition system is called a *Reactive Probabilistic Labelled Transition System* (RPLTS) 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. A RPLTS is a triple $(\mathcal{S}, \mathcal{A}, \mathsf{trans})$, where

- $\mathcal{S}$ is a set of states,
- $\mathcal{A} = \{\tau\} \sqcup \mathcal{A}_{ext}$ a set of labels, and
- $\mathsf{trans} : \mathcal{S} \to \mathcal{D}(\mathcal{S}) \sqcup (\mathcal{A}_{ext} \to \mathcal{D}(\mathcal{S}) \cup \{\star\})$ is a transition function that assigns to each state in $\mathcal{S}$
  - $*$ either a unique distribution for the label τ (the deterministic internal action);
  - $*$ or a function mapping labels in $\mathcal{A}_{ext}$ to a failure ($\star$) or a distribution over $\mathcal{S}$ (the non deterministic external actions).

States $s \in \mathcal{S}$ such that $\mathsf{trans}(s) : \mathcal{A}_{ext} \to \mathcal{D}(\mathcal{S}) \cup \{\star\}$ are called *external states*, while the ones such that $\mathsf{trans}(s) : \mathcal{D}(\mathcal{S})$ are called *internal states*. Given a RPLTS R, we denote by $\mathcal{S}_{ext}(R)$ and $\mathcal{S}_{int}(R)$ the sets of external and internal states of R respectively. For a more homogeneous notation, when s is an internal state, we sometimes write $\mathsf{trans}(s)(\tau) = D$ instead of $\mathsf{trans}(s) = D$.

Remark 2. In the particular case of a NPLTS N with no external action, resolving the internal non-determinism results in a RPLTS *without any* non-determinism. This is the case of the operational semantics N^o. Such a purely probabilistic system is typically equivalent to the notion of Markov Chain. By abuse of notation, the transition function

$$\mathsf{trans} : \mathcal{S} \to \mathcal{D}(\mathcal{S}) \sqcup (\varnothing \to \mathcal{D}(\mathcal{S}) \cup \{\star\})$$

of such RPLTS is rewritten as

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

as for any set X, the cardinality of the set $(\varnothing \to X)$ is 1.

Before defining the notion of *resolution*–or schedulers–, we need to introduce two classical notions in probabilistic models: the *convex hull* of a set of distributions and the *probabilistic lifting* of a function.

Notation 2. Let $\mathcal{S}$ be a set of states. The *convex hull* of $\Delta \subseteq \mathcal{D}(\mathcal{S})$, denoted $\mathrm{conv}(\Delta)$, is the set of distributions $D \in \mathcal{D}(\mathcal{S})$ such that

$$\exists \alpha_1, \dots, \alpha_n \in \mathbb{R}. \quad \exists D_1, \dots, D_n \in \Delta. \quad \sum_{i=1}^{n} \alpha_i = 1 \text{ and } D = \sum_{i=1}^{n} \alpha_i \cdot D_i$$

Intuitively, rather than choosing one distribution in Δ, each element in $\mathrm{conv}(\Delta)$ corresponds to a distribution over the distributions in Δ. This will be useful for defining randomized schedulers.

Next, we lift functions to distributions: applying a function f to a distribution simply defines a new distribution that transfers, according to f, the probability weight of elements in the domain of f to its image, possibly summing these weights when f maps several inputs to a same output.

Notation 3. Let $\mathcal{S}, \mathcal{S}'$ be two sets of states and $f : \mathcal{S} \to \mathcal{S}'$. We define the function $\overline{f} : \mathcal{D}(\mathcal{S}) \to \mathcal{D}(\mathcal{S}')$ to be the *probabilistic lifting of f*, where

$$\overline{f}(D) = \sum_{s \in \mathcal{S}} D(s) \cdot \delta_{f(s)}$$

When obvious from context, we will overload the notation and write f instead of $\overline{f}$.

We now define *resolutions* for a NPLTS that allow to solve the *internal*, but not external, non-determinism: a resolution describes *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 *unique* post-transition distribution must be chosen; in the second case, for each external action a, the resolution must choose to either stop (*i.e.*, $\mathsf{trans}(s) = \star$) or a *unique* distribution D such that $s \xrightarrow{a} D$ (*i.e.*, $\mathsf{trans}(s) = D$). Due to the possible existence of cycles in the NPLTS, a scheduler that visits multiple times a certain state s must be able to choose differently how to resolve the non determinism every time it visits s. This leads to the notion of *correspondence function*.

Definition 4 ([16]). A *randomized resolution* for a NPLTS $\mathsf{N} = (\mathcal{S}, \mathcal{A}, \mathsf{trans})$ is a pair (corr, R) where

- $R = (\mathcal{S}', \mathcal{A}, \mathsf{trans}')$ is a RPLTS, and
- $\mathsf{corr} : \mathcal{S}' \to \mathcal{S}$ is the *correspondence function* such that for all states $s' \in \mathcal{S}'$, $\mathsf{trans}'(s')(a) = D$ implies $\mathsf{corr}(D) \in \mathrm{conv}(\mathsf{trans}(\mathsf{corr}(s'))(a))$.

Given a NPLTS N we denote by $\mathcal{R}_{\mathsf{r}}(\mathsf{N})$ the set of randomized resolutions. Additionally, we denote $\mathcal{R}_{\mathsf{r}}^{o} = \mathcal{R}_{\mathsf{r}}(\mathsf{N}^{o})$. Figure 4 shows an example of a resolution from $\mathcal{R}_{\mathsf{r}}^{o}$ for the process P from Example 2.

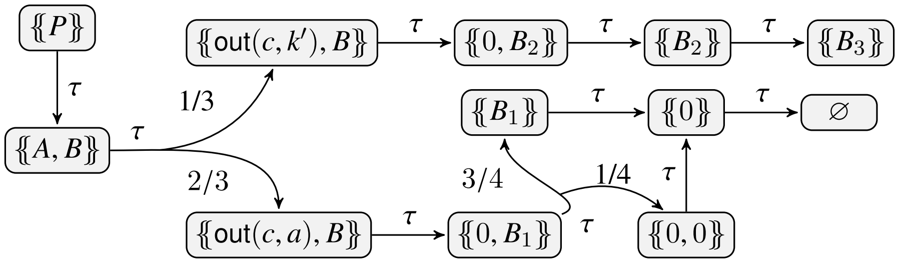


Figure 4. Example of a randomized resolution for process P from Example 2 where the correspondence function corr is the identity.

3.2. Computing the probability to reach a barb

The notion of *barb* is a classical way of expressing observables. Intuitively a state of N^o, *i.e.*, a multiset of processes, exhibits a barb c whenever an output on channel c is possible.

Definition 5. For $c \in \mathcal{N}_{pub}$ and $\mathcal{P} \in \mathcal{MP}$, we say that $\mathcal{P}$ *exhibits barb c* when there exists a process $\mathsf{out}(u,t).Q$ in $\mathcal{P}$ where $c \doteq u$ and $Msg(t)$. We denote by $\downarrow c$ the set of all multisets of processes that exhibit the barb c.

We next define the probability of reaching a state in a set of target states, in a *fully probabilistic* 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 6. Let $R = (\mathcal{S}, \mathcal{A}, \mathsf{trans})$ be a RPLTS, $\mathcal{T} \subseteq \mathcal{S}$ a set of states, and s an initial state. For every $n \in \mathbb{N}$ we define the *probability of reaching $\mathcal{T}$ from s in at most n steps* as:

$$\mathrm{RProb}_R^{\leqslant 0}(s,\mathcal{T}) = \begin{cases} 1 & \text{if } s \in \mathcal{T} \\ 0 & \text{otherwise.} \end{cases} \qquad \mathrm{RProb}_R^{\leqslant n+1}(s,\mathcal{T}) = \begin{cases} 1 & \text{if } s \in \mathcal{T} \\ 0 & \text{if } s \notin \mathcal{T} \wedge s \in \mathcal{S}_{ext}(R) \\ \sum_{u \in \mathrm{supp}(D)} D(u) \cdot \mathrm{RProb}_R^{\leqslant n}(u,\mathcal{T}) & \\ \quad \text{if } s \notin \mathcal{T} \wedge \mathsf{trans}(s)(\tau) = D & \end{cases}$$

We define the *probability of reaching $\mathcal{T}$ from s* as:

$$\mathrm{RProb}_R(s,\mathcal{T}) = \lim_{n \to +\infty} \mathrm{RProb}_R^{\leqslant n}(s,\mathcal{T}).$$

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

Given $\mathsf{N} = (\mathcal{S}_\mathsf{N}, \mathcal{A}, \mathsf{trans}_\mathsf{N})$, we denote by $\mathrm{RProb}_{\mathcal{R}_\mathsf{r}(\mathsf{N})}(s,\mathcal{T})$ the probability:

$$\sup \left\{ \mathrm{RProb}_R(s', \mathsf{corr}^{-1}(\mathcal{T})) \;\middle|\; \begin{array}{l} (\mathsf{corr}, R) \in \mathcal{R}_\mathsf{r}(\mathsf{N}), \\ \mathsf{corr}(s') = s \end{array} \right\}$$

3.3. Defining May Testing Equivalence

Intuitively, two processes are may-testing equivalent if they exhibit the same observations when executed in the presence of any attacker process. This models the inability of an arbitrary process to distinguish them. More formally, two multisets of processes $\mathcal{P}$ and $\mathcal{Q}$ are may testing equivalent when the attacker has the same probability over all schedulers to exhibit the barb c in both $\mathcal{P}$ and $\mathcal{Q}$.

Definition 7 (May testing equivalence)**.** Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}$. We say that $\mathcal{P} \leqslant_{may} \mathcal{Q}$ iff:

$$\begin{array}{c}\forall Adv \in \mathcal{MP} \text{ s.t. } fn(Adv) \subseteq \mathcal{N}_{pub}.\\ \forall c \in \mathcal{N}_{pub}.\end{array} \qquad \mathrm{RProb}_{\mathcal{R}_r^{o}}(\mathcal{P} \cup Adv, \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}_r^{o}}(\mathcal{Q} \cup Adv, \downarrow c)$$

We say that $\mathcal{P}, \mathcal{Q}$ are may testing equivalent, denoted $\mathcal{P} \approx_{may} \mathcal{Q}$, when $\mathcal{P} \leqslant_{may} \mathcal{Q}$ and $\mathcal{Q} \leqslant_{may} \mathcal{P}$.

Remark 3. One could also consider a more fine-grained definition of may testing pre-order that guarantees the equality of probabilities between two schedulers rather than comparing the probabilities over all schedulers. Formally, this pre-order, denoted $\leqslant'_{may}$, requires that for all resolutions $(\mathsf{corr}, R) \in \mathcal{R}_{\mathsf{r}}(\mathsf{N})$ and state s of R such that $\mathsf{corr}(s) = \mathcal{P} \cup Adv$, there exist a resolution (corr', R') and a state s' of R' such that $\mathsf{corr}'(s') = \mathcal{Q} \cup Adv$ and

$$\mathrm{RProb}_{R}(s, \mathsf{corr}^{-1}(\downarrow c)) = \mathrm{RProb}_{R'}(s', \mathsf{corr}'^{-1}(\downarrow c))$$

However, the resulting relation is counter-intuitive and distinguishes processes

$$\mathcal{P} := \{\!\{\mathsf{out}(a, 0)\}\!\} \quad \text{and} \quad \mathcal{Q} := \{\!\{\mathsf{out}(a, 0) +_{\frac{1}{2}} (\mathsf{out}(a, 0) +_{\frac{1}{2}} \mathsf{out}(a, 0))\}\!\}.$$

Indeed, we can show that $\mathcal{Q} \not\leqslant'_{may} \mathcal{P}$: for $Adv = \{\!\{0\}\!\}$, there exists a resolution (corr, R) such that

$$\mathsf{corr}(s) = \mathcal{P} \cup Adv \quad \text{and} \quad \mathrm{RProb}_{R}(\mathcal{Q}, \downarrow a) = \frac{1}{2}$$

but for every resolution (corr', R') such that $\mathsf{corr}'(s') = \mathcal{Q} \cup Adv$,

$$\mathrm{RProb}_{R'}(\mathcal{P}, \downarrow a) = 1.$$

3.4. Defining Observational Equivalence

In this section, we define *observational preorders* and *equivalence* which are stronger than may testing. When studying cryptographic protocols we suppose that internal actions are not observable and therefore only study *weak* equivalences hiding whether such internal actions take place or not. To define the observational preorder we need to introduce a weak relation for internal actions. In a purely non-deterministic system this simply corresponds to the reflexive, transitive closure $\xrightarrow{\tau}^{*}$. However, in our setting we need to compute the corresponding distributions.

Definition 8. Let N be a NPLTS and $D, E \in \mathcal{D}^{\leqslant 1}(\mathcal{S}_{\mathsf{N}})$.
We write $D \xRightarrow{\tau}_{\mathcal{R}_{\mathsf{r}}(\mathsf{N})} E$ when there exists $(\mathsf{corr}, R) \in \mathcal{R}_{\mathsf{r}}(\mathsf{N})$ and $D', E' \in \mathcal{D}^{\leqslant 1}(\mathcal{S}_R)$ such that

- $\mathsf{corr}(D') = D$, $\mathsf{corr}(E') = E$, $\mathrm{supp}(E') \subseteq \mathcal{S}_{ext}(R)$,
- $\forall u \in \mathcal{S}_{ext}(R).\ E'(u) = \sum_{s' \in \mathcal{S}_R} D'(s') \cdot \mathrm{RProb}_R(s', \{u\})$.

To define the observational preorder, we additionally need to lift relations defined on a given set to relations on sub-distributions over this set.

Definition 9 (Lifting of a relation)**.** Let R be a binary relation on a discrete set $\mathcal{S}$. We define the *lifting of R to sub-distributions* as the binary relation on $\mathcal{D}^{\leqslant 1}(\mathcal{S})$, denoted $\widehat{R}$, defined as:

$$D\ \widehat{R}\ E \text{ when } \forall \mathcal{S}' \subseteq \mathcal{S}, D(\mathcal{S}') \leqslant E(R(\mathcal{S}'))$$

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

Using these notions of weak transition and lifting of relations to sub-distributions we can define observational equivalence.

Definition 10. The *observational preorder* $\leqslant_{obs}$ is the largest relation R on $\mathcal{MP}$ such that $\mathcal{P}\ R\ \mathcal{Q}$ implies :

- $\forall c \in \mathcal{N}_{pub}.\ \mathrm{RProb}_{\mathcal{R}_{\mathsf{r}}^{o}}(\mathcal{P}, \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}_{\mathsf{r}}^{o}}(\mathcal{Q}, \downarrow c)$;
- if $\mathcal{P} \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{r}}^{o}} D$ and $D \in \mathcal{D}(\mathcal{S}_{\mathsf{N}^o})$ then $\mathcal{Q} \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{r}}^{o}} E$, $E \in \mathcal{D}(\mathcal{S}_{\mathsf{N}^o})$ and $D\ \widehat{R}\ E$;
- $\forall$ closed $Adv \in \mathcal{MP}$ such that $fn(Adv) \subseteq \mathcal{N}_{pub}$. $\{Adv\} \cup \mathcal{P}\ R\ \{Adv\} \cup \mathcal{Q}$.

The *observational equivalence* $\approx_{obs}$ is defined by additionally requiring R to be symmetric.

Remark 4. Note that we slightly diverge from the original definition of observational equivalence [11] where an evaluation context $C[\_]$ is of the form

$$\mathsf{new}\ n_1; \ldots; \mathsf{new}\ n_k; (\_ \mid A)$$

In our definition we simply consider a parallel process, and no additional name restriction. However, we now show that these two definitions coincide. Intuitively, restricting names whose scope includes the adversarial process A does not provide additional distinguishing power and these names could as well be public. While this result is of independent interest, the reader may safely skip the remainder of this section without hindering their understanding of the rest of the paper.

Our distinction between private and public names enforces that all restricted names $n_1, \ldots, n_k$ are in $\mathcal{N}_{priv}$. Applying an adversarial context with restriction in our formalism corresponds to applying a renaming from public to fresh private names. Hence, in our formalism the original observational equivalence can be defined as follows.

Definition 11. The *original observational preorder* $\leqslant_{ori}^{\mathcal{R}}$ is the largest relation R on $\mathcal{MP}$ such that $\mathcal{P}\ R\ \mathcal{Q}$ implies :

- for all $c \in \mathcal{N}_{pub}$, $\mathrm{RProb}_{\mathcal{R}}(\mathcal{P}, \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}}(\mathcal{Q}, \downarrow c)$;
- if $\mathcal{P} \overset{\tau}{\Longrightarrow}_{\mathsf{r}} D$ then $\mathcal{Q} \overset{\tau}{\Longrightarrow}_{\mathsf{r}} E$ and $D\ \widehat{R}\ E$;
- for all closed $Adv \in \mathcal{MP}$, for all renaming ρ, if $fn(Adv) \subseteq \mathcal{N}_{pub}$, $dom(\rho) \subseteq \mathcal{N}_{pub}$, $img(\rho) \subseteq \mathcal{N}_{priv}$ and $img(\rho) \cap n(\mathcal{P}, \mathcal{Q}, Adv) = \varnothing$ then $\{Adv\rho\} \cup \mathcal{P}\rho\ R\ \{Adv\rho\} \cup \mathcal{Q}\rho$.

The *original observational equivalence* $\approx^{\mathcal{R}}_{ori}$ is defined by additionally requesting R to be symmetric and in the second bullet point, by requesting both $D\ \widehat{R}\ E$ and $E\ \widehat{R}\ D$ to hold.

We now show that the two notions coincide. Intuitively, restricting names whose scope includes the adversarial process Adv corresponds to making previously public channels invisible to the attacker at later steps, which does not provide additional distinguishing power.

Lemma 1. $\leqslant^{\mathcal{R}}_{ori} = \leqslant^{\mathcal{R}}_{obs}$ and $\approx^{\mathcal{R}}_{ori} = \approx^{\mathcal{R}}_{obs}$.

Proof. Taking ρ to be the empty renaming, we have that $\leqslant^{\mathcal{R}}_{ori} \subseteq \leqslant^{\mathcal{R}}_{obs}$ and $\approx^{\mathcal{R}}_{ori} \subseteq \approx^{\mathcal{R}}_{obs}$.

Define the relation $\mathcal{R}$ as $\mathcal{P}\ \mathcal{R}\ \mathcal{Q}$ iff there exists $\mathcal{P}'$, $\mathcal{Q}'$ and a renaming ρ such that $\mathcal{P} = \mathcal{P}'\rho$, $\mathcal{Q} = \mathcal{Q}'\rho$, $\mathcal{P}' \leqslant^{\mathcal{R}}_{obs} \mathcal{Q}'$, $dom(\rho) \subseteq \mathcal{N}_{pub}$, $img(\rho) \subseteq \mathcal{N}_{priv}$ and $img(\rho) \cap n(\mathcal{P}', \mathcal{Q}', Adv) = \varnothing$.

As $\doteq$ is closed under substitution of names by terms, we can show that $\mathcal{P}\rho \to_\tau D\rho$ iff $\mathcal{P} \to_\tau D$. This can be propagate to schedulers, i.e. $(\mathsf{corr}, R) \in \mathcal{R}^{o}_{\mathsf{r}}$ if and only if $(\mathsf{corr}\rho, R) \in \mathcal{R}^{o}_{\mathsf{r}}$. Hence, we derive that

- $\mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}, \mathcal{T}) = \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}\rho, \mathcal{T}\rho)$
- $D \overset{\tau}{\Longrightarrow}_{\mathcal{R}^{o}_{\mathsf{r}}} E$ if and only if $D\rho \overset{\tau}{\Longrightarrow}_{\mathcal{R}^{o}_{\mathsf{r}}} E\rho$

We now show that $\mathcal{R}$ satisfies the three items in Definition 11:

- Let $c \in \mathcal{N}_{pub}$. If $c \in dom(\rho)$ then c does neither occur in $\mathcal{P}'\rho$ nor in $\mathcal{Q}'\rho$. Hence, $\mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}', \downarrow c) = 0 = \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{Q}', \downarrow c)$. Otherwise, $\downarrow c = \downarrow c\rho \cup \mathcal{T}$ where for all $\mathcal{P}' \in \mathcal{T}, fn(\mathcal{P}') \cap dom(\rho) \neq \varnothing$. Since, $\mathcal{P}'\rho$ and $\mathcal{Q}'\rho$ do not contain public names from $dom(\rho)$, they can never reach states from $\mathcal{T}$. Hence, $\mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}'\rho, \downarrow c) = \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}'\rho, \downarrow c\rho) = \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}', \downarrow c)$. Since $\mathcal{P}' \leqslant^{\mathcal{R}}_{obs} \mathcal{Q}'$, we deduce that
$$\mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{P}', \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{Q}', \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{Q}'\rho, \downarrow c\rho) \leqslant \mathrm{RProb}_{\mathcal{R}^{o}_{\mathsf{r}}}(\mathcal{Q}', \downarrow c)$$
- If $\mathcal{P}\rho \overset{\tau}{\Longrightarrow}_{\mathcal{R}^{o}_{\mathsf{r}}} D\rho$ then $\mathcal{P}' \overset{\tau}{\Longrightarrow}_{\mathcal{R}^{o}_{\mathsf{r}}} D$. By $\mathcal{P}' \leqslant^{\mathcal{R}}_{obs} \mathcal{Q}'$, we have $\mathcal{Q}' \overset{\tau}{\Longrightarrow}_{\mathcal{R}^{o}_{\mathsf{r}}} E$ and $D\ \widehat{\leqslant^{\mathcal{R}}_{obs}}\ E$. This implies that $D\rho\ \widehat{\mathcal{R}}\ E\rho$ with $\mathcal{Q}'\rho \overset{\tau}{\Longrightarrow}_{\mathcal{R}^{o}_{\mathsf{r}}} E\rho$.
- Let $Adv \in \mathcal{MP}$ and ρ' be a renaming such that $fn(Adv) \subseteq \mathcal{N}_{pub}$, $dom(\rho') \subseteq \mathcal{N}_{pub}$, $img(\rho') \subseteq \mathcal{N}_{priv}$ and $img(\rho') \cap n(\mathcal{P}'\rho, \mathcal{Q}'\rho) = \varnothing$. Note that the domains of ρ and ρ' may not be disjoint. Similarly, the process Adv may contain public names from $dom(\rho)$.
Hence, let ρ_{pub} be a renaming such that $dom(\rho_{pub}) = dom(\rho) \cap (dom(\rho') \cup fn(Adv))$, $img(\rho_{pub}) \subseteq \mathcal{N}_{pub} \setminus fn(\mathcal{P}', \mathcal{Q}', Adv)$. Hence, we define $\rho_1 = \rho\rho_{pub}^{-1}\rho'$.
We have $\mathcal{P}'\rho_1 = (\mathcal{P}'\rho)\rho'$ since $img(\rho_{pub}) \subseteq \mathcal{N}_{pub} \setminus fn(\mathcal{P}', \mathcal{Q}', Adv)$. Moreover, as $dom(\rho_{pub}) = dom(\rho) \cap (dom(\rho') \cup fn(Adv))$, we have that
$$(Adv\rho_{pub})\rho = Adv\rho_{pub}\rho|_{dom(\rho)\setminus dom(\rho_{pub})} = Adv\rho|_{dom(\rho)\setminus dom(\rho_{pub})}\rho_{pub} = Adv\rho_{pub}$$
Therefore, $Adv\rho_{pub}\rho_1 = Adv\rho_{pub}\rho_{pub}^{-1}\rho' = Adv\rho'$. This allows us to deduce that $\{Adv\rho'\} \cup \mathcal{P}'\rho' = (\{Adv\rho_{pub}\} \cup \mathcal{P}')\rho_1$. Similarly, we have $\{Adv\rho'\} \cup \mathcal{Q}'\rho' = (\{Adv\rho_{pub}\} \cup \mathcal{Q}')\rho_1$.
We know that $\mathcal{P}' \leqslant^{\mathcal{R}}_{obs} \mathcal{Q}'$. Hence $\{Adv\rho_{pub}\} \cup \mathcal{P}' \leqslant^{\mathcal{R}}_{obs} \{Adv\rho_{pub}\} \cup \mathcal{Q}'$ which allows us to conclude that $\{Adv\rho_{pub}\rho_1\} \cup \mathcal{P}'\rho_1\, \mathcal{R} \{Adv\rho_{pub}\rho_1\} \cup \mathcal{Q}'\rho_1$ and so $\{Adv\rho'\} \cup \mathcal{P}\rho'\, \mathcal{R} \{Adv\rho'\} \cup \mathcal{Q}\rho'$. □

$$
\begin{array}{ll}
(\mathcal{P},\phi) \rightarrow_{\tau} (D,\phi) & \text{if } \mathcal{P} \rightarrow_{\tau} D \\
(\{\!\!\{\mathsf{in}(u,x);P\}\!\!\} \cup \mathcal{P},\phi) \rightarrow_{in(\xi,\zeta)} \delta_{(\{\!\!\{P\{^{\zeta\phi}/_{x}\}\}\!\!\}\cup\mathcal{P},\phi)} & \text{if } u \doteq \xi\phi, Msg(\zeta\phi) \text{ and } vars(\xi,\zeta) \subseteq dom(\phi) \\
(\{\!\!\{\mathsf{out}(u,t);P\}\!\!\} \cup \mathcal{P},\phi) \rightarrow_{out(\xi,\mathsf{ax}_{n+1})} \delta_{(\{\!\!\{P\}\!\!\}\cup\mathcal{P},\phi\{ax_{n+1}\mapsto t\})} & \text{if } u \doteq \xi\phi, Msg(t), vars(\xi) \subseteq dom(\phi) \text{ and } |\phi| = n \\
(\mathcal{P},\phi) \rightarrow_{(\xi\sim\zeta)} \delta_{(\mathcal{P},\phi)} & \text{if } vars(\xi,\zeta) \subseteq dom(\phi), \xi\phi \sim \zeta\phi \text{ and } \sim \in \{\doteq,\dot{\neq}\}
\end{array}
$$

Figure 5. Labelled semantics: definition of $\rightarrow_a$

4. Labelled semantics and equivalences

As usual in π-calculi, and in the applied π-calculus, we can 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 adversary's *knowledge*. We consider a new set of variables $\mathcal{AX} = \{\mathsf{ax}_1, \mathsf{ax}_2, \ldots\}$ distinct from $\mathcal{X}$ that will act as pointers to messages that were previously output.

Definition 12. An extended process is a pair $(\mathcal{P},\phi)$, where $\mathcal{P} \in \mathcal{MP}$ and ϕ is a ground substitution

$$\{\mathsf{ax}_1 \mapsto t_1;\ \ldots\ ; \mathsf{ax}_n \mapsto t_n\}$$

such that $\mathsf{ax}_i \in \mathcal{AX}$, $t_i \in \mathcal{T}(\mathcal{F},\mathcal{N})$ and $Msg(t_i)$ for $1 \leqslant i \leqslant n$.
We denote by $\mathcal{SP}_\ell$ the set of all extended processes.

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 4. If D is a distribution over $\mathcal{MP}$, and ϕ a frame, we write (D,ϕ) for the distribution over extended processes defined as $(D,\phi) = \sum_{\mathcal{P}\in\mathsf{supp}(D)} D(\mathcal{P}) \cdot \delta_{(\mathcal{P},\phi)}$.

We now define the NPLTS N^ℓ for the labelled semantics. External actions model interactions with the attacker.

Definition 13. The *labelled semantics* is the NPLTS $\mathsf{N}^\ell = (\mathcal{SP}_\ell, \{\tau\} \cup \mathcal{A}^\ell_{ext}, \mathsf{trans}^\ell)$ where

- $\mathcal{A}^\ell_{ext}$ is the set of labels $in(\xi,\zeta)$, $out(\xi,\mathsf{ax})$, $(\xi \overset{?}{=} \zeta)$ and $(\xi \overset{?}{\neq} \zeta)$ with ξ,ζ recipes and $\mathsf{ax} \in \mathcal{AX}$;
- $\mathsf{trans}^\ell((\mathcal{P},\phi))(a) = \{D \mid (\mathcal{P},\phi) \rightarrow_a D\}$ where $\rightarrow_a$ is defined in Figure 5.

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 Figure 2 also applies to the frame ϕ, *i.e.*, a' must not appear in ϕ.

Remark 5. It should be noted that we deal with static equivalence in a different way as done usually in the applied π-calculus, or implicitly in the probabilistic applied π-calculus [12]: we encode static equivalence into the NPLTS N^ℓ by a countable set of actions–all the tests $(\xi \overset{?}{=} \zeta)$ 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 action* of the adversary. As shown later, this choice has no effect on the definition of the simulation pre-orders or on bisimulation, but it leads to a slightly different notion of trace equivalence, that is closer to may testing equivalence.

4.2. Defining Trace Equivalence

We first define the probability of executing a trace for a given resolution. As we are interested in *weak* trace preorder (where internal actions cannot be observed), traces are sequences of external actions only. Our definition uses the previously introduced notation $\mathrm{RProb}_R(s, \{t\})$: recall that this denotes the probability of reaching state t from state s using only internal actions for some resolution (corr, R).

Definition 14. Let $R = (\mathcal{S}, \{\tau\} \sqcup \mathcal{A}_{ext}, \mathsf{trans})$ be a RPLTS. Let $w \in {\mathcal{A}_{ext}}^*$ be a trace, *i.e.*, a finite word on the alphabet $\mathcal{A}_{ext}$. For all states $s \in \mathcal{S}$, we define the *probability of executing w starting from s in R* as:

- $\mathrm{Prob}_R(s, \epsilon) = 1$
- $\mathrm{Prob}_R(s, a.w) = \displaystyle\sum_{\substack{t \in \mathcal{S} \\ \mathsf{trans}(t)(a) = D}} \mathrm{RProb}_R(s, \{t\}) \cdot \sum_{s' \in \mathrm{supp}(D)} D(s') \cdot \mathrm{Prob}_R(s', w)$

Given a NPLTS $\mathsf{N} = (\mathcal{S}, \mathcal{A}, \mathsf{trans})$, we denote by $\mathrm{Prob}_{\mathcal{R}_{\mathsf{r}}(\mathsf{N})}(s, w)$ the probability:

$$\sup\{\mathrm{Prob}_R(s', w) \mid (\mathsf{corr}, R) \in \mathcal{R}_{\mathsf{r}}(\mathsf{N}), \mathsf{corr}(s') = s\}$$

This allows us to define trace equivalence of $(\mathcal{P}, \phi)$ and $(\mathcal{P}', \phi')$: intuitively any trace that can be executed in $(\mathcal{P}, \phi)$ can be executed with at least the same probability in $(\mathcal{P}', \phi')$ and vice-versa.

Definition 15 (trace equivalence)**.** Let $(\mathcal{P}, \phi)$, $(\mathcal{P}', \phi') \in \mathcal{SP}_\ell$. We say that $(\mathcal{P}, \phi) \leqslant_{tr} (\mathcal{P}', \phi')$ iff

$$\text{for all } w \in {\mathcal{A}^{\ell}_{ext}}^*. \quad \mathrm{Prob}_{\mathcal{R}_{\mathsf{r}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), w) \leqslant \mathrm{Prob}_{\mathcal{R}_{\mathsf{r}}(\mathsf{N}^\ell)}((\mathcal{P}', \phi'), w)$$

$(\mathcal{P}, \phi)$ and $(\mathcal{P}', \phi')$ are trace equivalent, denoted $(\mathcal{P}, \phi) \approx_{tr} (\mathcal{P}', \phi')$, iff

$$(\mathcal{P}, \phi) \leqslant_{tr} (\mathcal{P}', \phi') \quad \text{and} \quad (\mathcal{P}', \phi') \leqslant_{tr} (\mathcal{P}, \phi).$$

Unlike, in the purely possibilistic case, in our probabilistic setting trace preorder is (strictly) weaker than the may testing preorder.

Theorem 1. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}$ be two processes.

$$\mathcal{P} \leqslant_{may} \mathcal{Q} \quad \Rightarrow \quad (\mathcal{P}, \varnothing) \leqslant_{tr} (\mathcal{Q}, \varnothing)$$

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

$$(\mathcal{P}, \varnothing) \leqslant_{tr} (\mathcal{Q}, \varnothing) \quad \text{and} \quad \mathcal{P} \not\leqslant_{may} \mathcal{Q}$$

$$
\begin{aligned}
\mathcal{P} &= \{\!\!\{\mathsf{new}\ k;\ (P(0) \mid P(0) \mid P(1) \mid P_{dec})\}\!\!\} \\
\text{and } \mathcal{Q} &= \{\!\!\{\mathsf{new}\ k;\ (P(0) \mid P(1) \mid P(1) \mid P_{dec})\}\!\!\} \\
\text{where } P(x) &= \mathsf{new}\ r; \mathsf{out}(c, enc(x,r,k)) \\
\text{and } P_{dec} &= \mathsf{in}(d,y); \mathsf{out}(d, dec(y,k))
\end{aligned}
$$

Figure 6. $\mathcal{P}, \mathcal{Q}$ such that $(\mathcal{P}, \varnothing) \leqslant_{tr} (\mathcal{Q}, \varnothing)$ and $\mathcal{P} \not\leqslant_{may} \mathcal{Q}$

Figure 6 witnesses that the implication is strict. $\mathcal{P}$ and $\mathcal{Q}$ output each 3 encrypted bits (in a non-deterministic order). $\mathcal{P}$ outputs twice the encryption of 0; $\mathcal{Q}$ twice the encryption of 1. The (randomized) encryption ensures that these three values are indeed indistinguishable. We give the adversary a *single* access to a decryption oracle P_{dec}. Intuitively, trace equivalence holds, as the scheduler can ensure that matching encryptions are sent to P_{dec}. However, may-testing does not hold: the attacker chooses uniformly at random one of the three encryptions to submit. The probability to hit 0 will be $\frac{2}{3}$ in $\mathcal{P}$ and only $\frac{1}{3}$ in $\mathcal{Q}$.

Observe that Theorem 1 holds for any processes and does not require them to be image-finite, contrary to usual results in the literature, *e.g.*, [23]. This discrepancy comes from our choice of labelled actions for static equivalence (see Remark 5): a trace cannot test *directly* static equivalence, but can only do a *finite* numbers of recipe tests. We believe this variant definition of trace equivalence to be of independent interest as it provides an exact characterization of may testing in the purely non-deterministic case.

4.3. Defining (bi)simulations

In this section, we define *simulations* on probabilistic processes and corresponding equivalences. Our definition of simulation preorder is similar to the definition of *randomized weak simulation preorder* introduced by Segala and Lynch for probabilistic automata [13]. We reuse the lifting of a relation and the weak relation for internal actions defined in Section 3.4 but applied to the NPLTS N^ℓ. In particular, given an action $a \in \mathcal{A}^\ell_{ext}$ and two distributions $D, D' \in \mathcal{D}(\mathcal{S})$, we write $D \overset{a}{\Longrightarrow}_{\mathcal{R}^\ell_\mathsf{r}} D'$ when $D \overset{\tau}{\Longrightarrow}_{\mathcal{R}^\ell_\mathsf{r}} E_1$, $E_1 \xrightarrow{a} E_2$ and $E_2 \overset{\tau}{\Longrightarrow}_{\mathcal{R}^\ell_\mathsf{r}} D'$ for some E_1, E_2 and where $\mathcal{R}^\ell_\mathsf{r}$ denotes $\mathcal{R}_\mathsf{r}(\mathsf{N}^\ell)$. Here $E_1 \xrightarrow{a} E_2$ is the natural lifting of the transition function of N^ℓ, *i.e.*, $E_2 = \sum_s E_1(s) \cdot D$ with $s \xrightarrow{a} D$.

Definition 16. A relation $R \subseteq (\mathcal{S}_{\mathsf{N}^\ell} \times \mathcal{S}_{\mathsf{N}^\ell})$ is

- a *simulation* if $s_1\ R\ s_2$ implies that for all $a \in \mathcal{A}^\ell_{ext} \cup \{\tau\}$, $D_1 \in \mathcal{D}(\mathcal{S}_{\mathsf{N}^\ell})$

$$\text{if} \quad s_1 \xrightarrow{a} D_1 \quad \text{then} \quad s_2 \overset{a}{\Longrightarrow}_{\mathcal{R}^\ell_\mathsf{r}} D_2,\ D_2 \in \mathcal{D}(\mathcal{S}_{\mathsf{N}^\ell}) \text{ and } D_1\ \widehat{R}\ D_2$$

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

The *simulation preorder*, denoted $\leqslant_{sim}$, is the largest simulation and *bisimilarity*, denoted $\approx_{bi}$, is the largest bisimulation. We define *similarity*, denoted $\approx_{sim}$, as $\leqslant_{sim} \cap \leqslant_{sim}^{-1}$.

As usual in the field of (bi)simulation, it can be shown that $\leqslant_{sim}$, respectively $\approx_{bi}$, exists [19] 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 [18].

The following proposition from [18] states that, as usual in the non-probabilistic case, the weak arrow $\overset{a}{\Longrightarrow}_{\mathcal{R}_r^\ell}$ can replace the single arrow $\xrightarrow{a}$ in the definition of simulation.

Proposition 1. Let R be the largest binary relation on $\mathcal{S}_{\mathsf{N}^\ell}$ such that $s_1 \; R \; s_2$ implies that for every $a \in \mathcal{A}_{ext} \cup \{\tau\}$,

$$\text{if} \quad s_1 \overset{a}{\Longrightarrow}_{\mathcal{R}_r^\ell} D_1 \quad \text{then} \quad s_2 \overset{a}{\Longrightarrow}_{\mathcal{R}_r^\ell} D_2,\ D_2 \in \mathcal{D}(\mathcal{S}_{\mathsf{N}^\ell}) \text{ and } D_1\ \widehat{R}\ D_2$$

We have $R = \leqslant_{sim}$.

Remark 6. Observe that our choice of labelled actions for static equivalence (see Remark 5) has no impact on the resulting simulation and bisimulation. Indeed, if two N^ℓ-states $(\mathcal{P},\phi)$ and $(\mathcal{Q},\psi)$ are in the simulation pre-order or bisimulation, then ϕ is statically equivalent to ψ. Indeed, for all $a \in \{\xi \overset{?}{=} \zeta, \xi \overset{?}{\neq} \zeta\}$, $\delta_{(\mathcal{P},\phi)} \xrightarrow{a} \delta_{(\mathcal{P},\phi)}$ implies that $\delta_{(\mathcal{Q},\psi)} \overset{a}{\Longrightarrow}_{\mathcal{R}_r^\ell} D$ and $\delta_{(\mathcal{P},\phi)} \widehat{\leqslant_{sim}} D$. Since neither the a transition or τ transition modifies the frame, the former indicates that $\delta_{(\mathcal{Q},\psi)} \overset{\tau}{\Longrightarrow}_{\mathcal{R}_r^\ell} E_1$, $E_1 \xrightarrow{a} E_2$ and $E_2 \overset{\tau}{\Longrightarrow}_{\mathcal{R}_r^\ell} D$, with for all $(\mathcal{Q}',\psi') \in \mathrm{supp}(E_1)$, $\psi = \psi'$. The former ensures that $\mathrm{supp}(E_1) \neq \varnothing$. Therefore, for all ξ,ζ, if $\xi \overset{?}{=} \zeta$ or $\xi \overset{?}{\neq} \zeta$ holds on ϕ then it also holds on ψ respectively. It implies that ϕ and ψ are statically equivalent.

We now show that observational preorder and equivalence are exactly characterized by the simulation preorder and bisimilarity.

Theorem 2. Let $\mathcal{P}$, $\mathcal{Q}$ two processes in $\mathcal{MP}$.

$$\mathcal{P} \leqslant_{obs} \mathcal{Q} \quad \text{iff} \quad (\mathcal{P},\varnothing) \leqslant_{sim} (\mathcal{Q},\varnothing) \qquad \text{and} \qquad \mathcal{P} \approx_{obs} \mathcal{Q} \quad \text{iff} \quad (\mathcal{P},\varnothing) \approx_{bi} (\mathcal{Q},\varnothing)$$

Proof sketch. We here provide the main intuitions of the proof that $\mathcal{P} \leqslant_{obs} \mathcal{Q}$ iff $(\mathcal{P},\varnothing) \leqslant_{sim} (\mathcal{Q},\varnothing)$.

(⇒). To show that observational preorder implies simulation, we need to represent the frame of an extended process $(\mathcal{P},\phi)$ as a process: we output in parallel the terms $\mathsf{ax}_i\phi$, with $\mathsf{ax}_i \in dom(\phi)$, on a public channel c_i, distinct for each i and not occurring anywhere in $(\mathcal{P},\phi)$. Thus, we build the relation $\mathcal{R}$ such that

$$(\mathcal{P},\phi)\ \mathcal{R}\ (\mathcal{Q},\phi') \qquad \text{if} \qquad \mathcal{P} \cup \{\!\{\mathsf{out}(c_i,\mathsf{ax}_i\phi)\}\!\}_{i=1}^n \leqslant_{obs} \mathcal{Q} \cup \{\!\{\mathsf{out}(c_i,\mathsf{ax}_i\phi')\}\!\}_{i=1}^n$$

with $|dom(\phi)| = |dom(\phi')| = n$ and $c_1,\ldots,c_n \in \mathcal{N}_{pub}$ pairwise distinct and not occurring in $\mathcal{P},\mathcal{Q},\phi,\phi'$.

As the public channels c_i do not occur anywhere else, any internal transition on

$$\mathcal{P}_1 = \mathcal{P} \cup \{\!\{\mathsf{out}(c_i,\mathsf{ax}_i\phi)\}\!\}_{i=1}^n$$

must correspond to an internal transition on $\mathcal{P}$; and similarly for $\mathcal{Q}$.

For all visible actions, we rely on $\leqslant_{obs}$ being closed by composition with an adversarial process. For example, when the action is the test $\xi \overset{?}{=} \zeta$, we compose with the adversarial process that *(i)* reads the frame, *(ii)* applies the test, and *(iii)* outputs on a fresh public channel ok if the test succeeds:

$$Adv = \mathsf{in}(c_1,x_1);\ldots;\mathsf{in}(c_n,x_n);\mathsf{if}\ \xi\rho = \zeta\rho\ \mathsf{then}\ \mathsf{out}(ok,ok)$$

where $x_1, \dots, x_n$ are fresh variables and $\rho = \{\mathsf{ax}_i \mapsto x_i\}_{i=1}^n$. We then consider the transition

$$\{\!\{Adv\}\!\} \cup \mathcal{P}_1 \overset{\tau}{\Longrightarrow}_{\mathcal{R}_\mathsf{r}^\varrho} \delta_{\mathcal{P} \cup \mathsf{out}(ok,ok)}$$

and the fact that $\mathrm{RProb}_{\mathcal{R}_\mathsf{r}}(\mathcal{P} \cup \mathsf{out}(ok, ok), \downarrow ok) = 1$ to conclude. Indeed, for

$$\{\!\{Adv\}\!\} \cup \mathcal{Q} \cup \{\!\{\mathsf{out}(c_i, \mathsf{ax}_i\phi').0\}\!\}_{i=1}^n \overset{\tau}{\Longrightarrow}_\mathsf{r} E$$

to exist with $\delta_{\mathcal{P} \cup \mathsf{out}(ok,ok)} \widehat{R}\ E$, $\{\!\{Adv\}\!\} \cup \mathcal{Q} \cup \{\!\{\mathsf{out}(c_i, \mathsf{ax}_i\phi'); 0\}\!\}_{i=1}^n$ must also have passed the test $\xi\rho = \zeta\rho$ in the conditional branching. Hence $\xi\phi' = \zeta\phi'$ and so $(\mathcal{Q}, \phi') \xrightarrow{\xi \overset{?}{=} \zeta} \delta_{(\mathcal{Q},\phi')}$.

When the visible action is an output or an input, the process is more complicated. The adversarial process starts by reading the frame as before and executing the action. The last part of the adversarial process consists in outputting again the frame so that we re-enter the relation $\mathcal{R}$. Assume for instance the action $in(\xi, \zeta)$. By definition, $\mathcal{P} = \mathcal{P}_1 \cup \{\!\{\mathsf{in}(c, x); P\}\!\}$ with $\xi\phi \doteq c$, $(\mathcal{P}, \phi) \xrightarrow{in(\xi,\zeta)} \delta_{\mathcal{P}_1 \cup \{\!\{P\sigma\}\!\}}$ and $\sigma = \{^{\zeta\phi}/_x\}$.

We consider the following adversarial process Adv:

$$Adv = \mathsf{in}(c_1, x_1); \dots; \mathsf{in}(c_n, x_n); \mathsf{out}(\xi\rho, \zeta\rho); (\mathsf{out}(ok, ok) +_{0.5} (\mathsf{out}(c_1', x_1) \mid \cdots \mid \mathsf{out}(c_n', x_n)))$$

where $c_1', \dots, c_n'$ are fresh public names pairwise distinct not occurring anywhere else. We will conclude by considering the transition

$$\{\!\{Adv\}\!\} \cup \mathcal{P} \cup \{\!\{\mathsf{out}(c_i, \mathsf{ax}_i\phi); 0\}\!\}_{i=1}^n \overset{\tau}{\Longrightarrow}_\mathsf{r} D$$

where $D = 0.5 \cdot \delta_{\mathcal{P}_1 \cup \{\!\{P\{^{\zeta\phi}/_x\}, \mathsf{out}(ok,ok)\}\!\}} + 0.5 \cdot \delta_{\mathcal{P}_1 \cup \{\!\{P\{\zeta\phi/x\}\}\!\} \cup \{\!\{\mathsf{out}(c_i', \mathsf{ax}_i\phi); 0\}\!\}_{i=1}^n}$. Indeed, for

$$\{\!\{Adv\}\!\} \cup \mathcal{Q} \cup \{\!\{\mathsf{out}(c_i, \mathsf{ax}_i\phi'); 0\}\!\}_{i=1}^n \overset{\tau}{\Longrightarrow}_\mathsf{r} E$$

to exist with $D\ \widehat{R}\ E$, $\{\!\{Adv\}\!\} \cup \mathcal{Q} \cup \{\!\{\mathsf{out}(c_i, \mathsf{ax}_i\phi'); 0\}\!\}_{i=1}^n$ must also have applied an internal transition executing the construct $\mathsf{out}(\xi\rho, \zeta\rho)$ which allows for the labeled action $in(\xi, \zeta)$ to be executed on $(\mathcal{Q}, \phi')$.

($\Leftarrow$). Showing that simulation implies observational preorder is more straightforward. We build a relation $\mathcal{R}$ such that $\mathcal{P}\ \mathcal{R}\ \mathcal{Q}$ when there exist two extended processes $(\mathcal{P}_1, \phi), (\mathcal{Q}_1, \phi')$ with compatible frames (i.e. $dom(\phi) = dom(\phi')$), a renaming ρ from $\mathcal{N}_{pub}$ to $\mathcal{N}_{priv}$, and a multiset of adversarial processes $\mathcal{P}_{Att}$ such that:

- names in $img(\rho)$ do not occur in $\mathcal{P}_1, \phi, \mathcal{Q}_1$ and ϕ';
- $\mathcal{P} = \mathcal{P}_1\rho \cup \mathcal{P}_{Att}\{^{\mathsf{ax}_i\phi}/_{x_i}\}_{i=1}^n\rho$;
- $\mathcal{Q} = \mathcal{Q}_1\rho \cup \mathcal{P}_{Att}\{^{\mathsf{ax}_i\phi'}/_{x_i}\}_{i=1}^n\rho$;
- $(\mathcal{P}_1, \phi) \leqslant_{sim} (\mathcal{Q}_1, \phi')$.

The renaming ρ replaces the private names that are generated by the processes in $\mathcal{P}_{Att}$ (through the construct $\mathsf{new}\ a; P$) with public names that are chosen *fresh* (i.e. not in $\mathcal{P}_1, \phi, \mathcal{Q}_1$ and ϕ'). □

4.4. Randomized vs non-randomized schedulers

All our equivalence notions are based on randomized schedulers where the non-determinism is solved by picking a distribution from the convex hull of the available distributions. In the literature, more restrictive *non-randomized schedulers* have also been considered when defining observational equivalence and bisimilarity [12]. A non-randomized scheduler solves the non-determinism by choosing directly one of the available distributions. Formally, in Definition 4, instead of requiring that $\mathsf{trans}'(s')(a) = D$ implies $\mathsf{corr}(D) \in \mathrm{conv}(\mathsf{trans}(\mathsf{corr}(s'))(a))$, a non-randomized resolution requires that $\mathsf{trans}'(s')(a) = D$ implies $\mathsf{corr}(D) \in \mathsf{trans}(\mathsf{corr}(s'))(a)$ and corr is injective on the support of D.

Denoting by $\mathcal{R}_{\mathsf{nr}}(\mathsf{N})$ the set of all non-randomized schedulers of N, we can naturally update the notions used to define behavioural equivalences to non-randomized schedulers. For instance, we denote by $\mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N})}(s, \mathcal{T})$ the probability of reaching $\mathcal{T}$ from s over all non randomized schedulers $\mathcal{R}_{\mathsf{nr}}(\mathsf{N})$. Similarly, $\mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N})}(s, w)$ denotes the probability of executing the trace w from s over all schedulers from $\mathcal{R}_{\mathsf{nr}}(\mathsf{N})$. Updating the definitions results into may-testing and trace preorder for non-randomized schedulers, denoted $\leqslant_{may}^{\mathsf{nr}}$ and $\leqslant_{tr}^{\mathsf{nr}}$ respectively. We now show that $\leqslant_{may}$ and $\leqslant_{tr}$ do not depend on the whether schedulers are randomized or not (unlike simulation based notions as we will see below).

Lemma 2. May testing and trace preorders with randomized and non-randomized resolutions coincide:

$$\leqslant_{may} = \leqslant_{may}^{\mathsf{nr}} \quad \text{and} \quad \leqslant_{tr} = \leqslant_{tr}^{\mathsf{nr}}$$

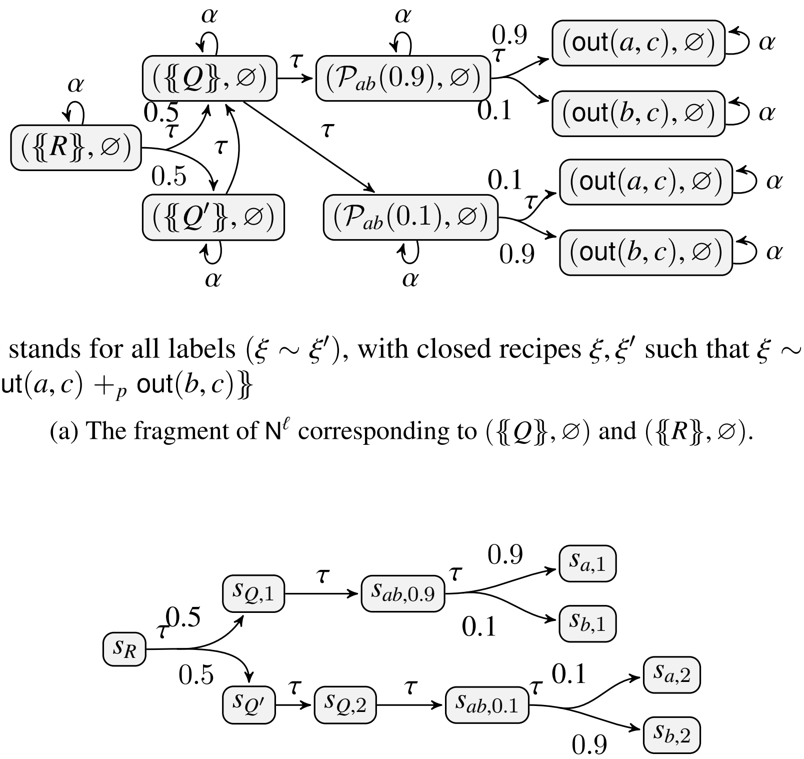
Proof sketch. The core of the proof is the following fact: when we fix a randomized resolution R, an initial state s, and $n \in \mathbb{N}$, it is possible to *decompose* the behaviour of R from state s and during the n first execution steps into a weighted family of non-randomized resolution $(\alpha_i, R_i)_{i \in I}$ (where the weight α_i is a coefficient in $[0, 1]$, in the sense that for every set of processes $\mathcal{P}$, $\mathrm{RProb}_R^{\leqslant n}(s, \mathcal{P}) = \sum_{i \in I} \alpha_i \mathrm{RProb}_{R_i}^{\leqslant n}(s, \mathcal{P})$. The construction of this decomposition is defined inductively on n. □

This result is of interest as it is often easier to manipulate non-randomized schedulers, and we expect automated verification to be more convenient as well.

When considering observational equivalence, simulation and bisimulation, non-randomized schedulers raise a number of issues. First, as highlighted for instance in [19, 31], when considering bisimulation or simulation on general NPLTSs, non-randomized resolutions result into relations that are *not transitive*. We show that even on the specific NPLTS N^{ℓ}, simulation is not transitive.

Simulation for non-randomized scheduler is naturally defined by extending the notation $D \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{r}}(\mathsf{N})} E$ to non-randomized scheduler, denoted $D \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N})} E$: from Definition 8, we require (corr, R) to be in $\mathcal{R}_{\mathsf{nr}}(\mathsf{N})$ and additionally require an injectivity property on the correspondence function, *i.e.*, corr is injective on the support of D'. We denote the resulting simulation with non-randomized schedulers by $\leqslant_{sim}^{\mathsf{nr}}$ (and similarly for $\leqslant_{obs}^{\mathsf{nr}}$, $\approx_{obs}^{\mathsf{nr}}$ and $\approx_{bi}^{\mathsf{nr}}$).

Lemma 3. $\leqslant_{sim}^{\mathsf{nr}}$ is not transitive.



For readability, α stands for all labels $(\xi \sim \xi')$, with closed recipes ξ, ξ' such that $\xi \sim \xi'$ $(\sim \in \{\dot{=}, \dot{\neq}\})$ and $\mathcal{P}_{ab}(p) = \{\!\{\mathsf{out}(a,c) +_p \mathsf{out}(b,c)\}\!\}$

(a) The fragment of N^ℓ corresponding to $(\{\!\{Q\}\!\}, \varnothing)$ and $(\{\!\{R\}\!\}, \varnothing)$.

(b) The resolution for $(\{\!\{R\}\!\}, \varnothing) \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{nr}}} 0.5 \cdot \delta_{\mathsf{out}(a,c)} + 0.5 \cdot \delta_{\mathsf{out}(b,c)}$.

Figure 7. Fragments of N^ℓ showing $(\{\!\{P\}\!\}, \varnothing) \leqslant^{\mathsf{nr}}_{sim} (\{\!\{R\}\!\}, \varnothing) \leqslant^{\mathsf{nr}}_{sim} (\{\!\{Q\}\!\}, \varnothing)$ but $(\{\!\{P\}\!\}, \varnothing) \not\leqslant^{\mathsf{nr}}_{sim} (\{\!\{Q\}\!\}, \varnothing)$

Proof sketch. Consider processes

$$\begin{aligned} P &= \mathsf{out}(a,c) +_{0.5} \mathsf{out}(b,c) \\ Q &= (\mathsf{out}(a,c) +_{0.9} \mathsf{out}(b,c)) + (\mathsf{out}(a,c) +_{0.1} \mathsf{out}(b,c)) \\ Q' &= \mathsf{if}\ c = c\ \mathsf{then}\ Q\ \mathsf{else}\ 0 \\ R &= Q +_{0.5} Q' \end{aligned}$$

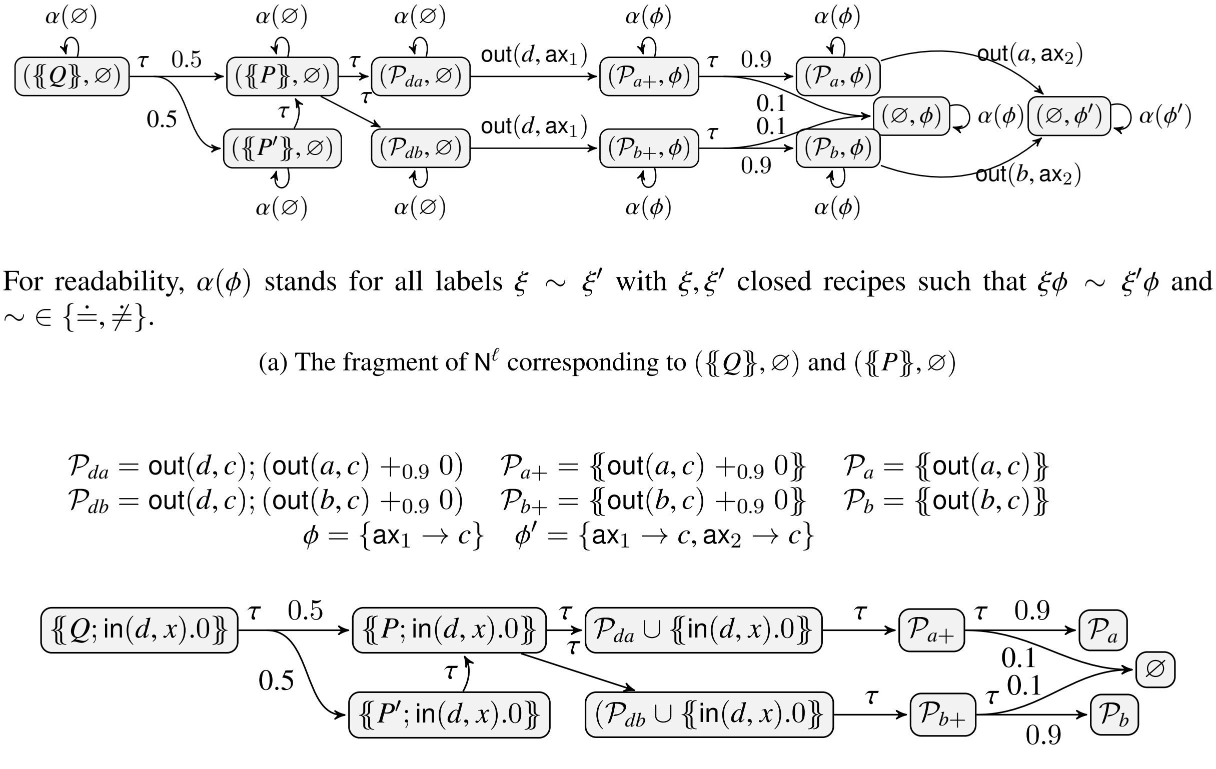
The corresponding fragment of N^ℓ is displayed in Figure 7a. We will show that

$$(\{\!\{P\}\!\}, \varnothing) \leqslant^{\mathsf{nr}}_{sim} (\{\!\{R\}\!\}, \varnothing) \text{ and } (\{\!\{R\}\!\}, \varnothing) \leqslant^{\mathsf{nr}}_{sim} (\{\!\{Q\}\!\}, \varnothing) \text{ but } (\{\!\{P\}\!\}, \varnothing) \not\leqslant^{\mathsf{nr}}_{sim} (\{\!\{Q\}\!\}, \varnothing)$$

It is easy to see that $(\{\!\{Q\}\!\}, \varnothing) \approx^{\mathsf{nr}}_{bi} (\{\!\{Q'\}\!\}, \varnothing)$ and so $(\{\!\{R\}\!\}, \varnothing) \leqslant^{\mathsf{nr}}_{sim} (\{\!\{Q\}\!\}, \varnothing)$. The difficult part of the proof of $(\{\!\{P\}\!\}, \varnothing) \leqslant^{\mathsf{nr}}_{sim} (\{\!\{R\}\!\}, \varnothing)$ is to match

$$(\{\!\{P\}\!\}, \varnothing) \xrightarrow{\tau} 0.5 \cdot \delta_{(\{\!\{\mathsf{out}(a,c)\}\!\}, \varnothing)} + 0.5 \cdot \delta_{(\{\!\{\mathsf{out}(b,c)\}\!\}, \varnothing)}$$

This is achieved by the scheduler displayed in Figure 7b.


$\alpha(\varnothing)$
$(\{\!\!\{Q\}\!\!\}, \varnothing)$
τ 0.5
0.5
$(\{\!\!\{P\}\!\!\}, \varnothing)$
$(\{\!\!\{P'\}\!\!\}, \varnothing)$
τ
$(\mathcal{P}_{da}, \varnothing)$
$(\mathcal{P}_{db}, \varnothing)$
$\mathsf{out}(d, \mathsf{ax}_1)$
$\alpha(\phi)$
$(\mathcal{P}_{a+}, \phi)$
$(\mathcal{P}_{b+}, \phi)$
τ 0.9
0.1
$(\mathcal{P}_a, \phi)$
$(\mathcal{P}_b, \phi)$
$(\varnothing, \phi)$
$(\varnothing, \phi')$
$\alpha(\phi')$
$\mathsf{out}(a, \mathsf{ax}_2)$
$\mathsf{out}(b, \mathsf{ax}_2)$

For readability, $\alpha(\phi)$ stands for all labels $\xi \sim \xi'$ with ξ, ξ' closed recipes such that $\xi\phi \sim \xi'\phi$ and $\sim \in \{\doteq, \not\doteq\}$.

(a) The fragment of N^{ℓ} corresponding to $(\{\!\!\{Q\}\!\!\}, \varnothing)$ and $(\{\!\!\{P\}\!\!\}, \varnothing)$

$$\mathcal{P}_{da} = \mathsf{out}(d,c); (\mathsf{out}(a,c) +_{0.9} 0) \quad \mathcal{P}_{a+} = \{\!\!\{\mathsf{out}(a,c) +_{0.9} 0\}\!\!\} \quad \mathcal{P}_a = \{\!\!\{\mathsf{out}(a,c)\}\!\!\}$$
$$\mathcal{P}_{db} = \mathsf{out}(d,c); (\mathsf{out}(b,c) +_{0.9} 0) \quad \mathcal{P}_{b+} = \{\!\!\{\mathsf{out}(b,c) +_{0.9} 0\}\!\!\} \quad \mathcal{P}_b = \{\!\!\{\mathsf{out}(b,c)\}\!\!\}$$
$$\phi = \{\mathsf{ax}_1 \to c\} \quad \phi' = \{\mathsf{ax}_1 \to c, \mathsf{ax}_2 \to c\}$$

$\{\!\!\{Q; \mathsf{in}(d,x).0\}\!\!\}$
τ 0.5
0.5
$\{\!\!\{P; \mathsf{in}(d,x).0\}\!\!\}$
$\{\!\!\{P'; \mathsf{in}(d,x).0\}\!\!\}$
τ
$\mathcal{P}_{da} \cup \{\!\!\{\mathsf{in}(d,x).0\}\!\!\}$
$(\mathcal{P}_{db} \cup \{\!\!\{\mathsf{in}(d,x).0\}\!\!\}$
$\mathcal{P}_{a+}$
$\mathcal{P}_{b+}$
τ 0.9
0.1
$\mathcal{P}_a$
$\mathcal{P}_b$
$\varnothing$

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

Figure 8. Fragments of NPLTS showing $(\{\!\!\{Q\}\!\!\}, \varnothing) \approx^{\mathsf{nr}}_{bi} (\{\!\!\{P\}\!\!\}, \varnothing)$ and $\{\!\!\{Q\}\!\!\} \not\preccurlyeq^{\mathsf{nr}}_{obs} \{\!\!\{P\}\!\!\}$.

Finally, we prove $(\{\!\!\{P\}\!\!\}, \varnothing) \not\preccurlyeq^{\mathsf{nr}}_{sim} (\{\!\!\{R\}\!\!\}, \varnothing)$ by showing that the transition $(\{\!\!\{P\}\!\!\}, \varnothing) \xrightarrow{\tau} 0.5 \cdot \delta_{(\{\!\!\{\mathsf{out}(a,c)\}\!\!\}, \varnothing)} + 0.5 \cdot \delta_{(\{\!\!\{\mathsf{out}(b,c)\}\!\!\}, \varnothing)}$ cannot be simulated in $(\{\!\!\{Q\}\!\!\}, \varnothing)$. □

Note that the definitions of bisimilarity in [12] rely on non-randomized schedulers. Even though this does not necessarily imply that their relation is not transitive (as they focus directly on the semantics of processes) we show in the next lemma that $\approx^{\mathsf{nr}}_{bi}$ and $\approx^{\mathsf{nr}}_{obs}$ do *not* coincide, hence disproving [12, Theorem 2]. This reenforces our belief that it is preferable to use randomized schedulers in our definition.

Lemma 4. There exist processes $P, Q \in \mathcal{SP}$ such that

- $(\{\!\!\{Q\}\!\!\}, \varnothing) \approx_{bi} (\{\!\!\{P\}\!\!\}, \varnothing)$,
- $(\{\!\!\{Q\}\!\!\}, \varnothing) \approx^{\mathsf{nr}}_{bi} (\{\!\!\{P\}\!\!\}, \varnothing)$, and
- $\{\!\!\{Q\}\!\!\} \not\preccurlyeq^{\mathsf{nr}}_{obs} \{\!\!\{P\}\!\!\}$.

Proof sketch. We consider the following processes:

$$\begin{aligned} P &= \mathsf{out}(d,c); (\mathsf{out}(a,c) +_{0.9} 0) + \\ &\quad\ \mathsf{out}(d,c); (\mathsf{out}(b,c) +_{0.9} 0) \\ P' &= \mathsf{if}\ c = c\ \mathsf{then}\ P\ \mathsf{else}\ 0 \\ Q &= P +_{\frac{1}{2}} P' \end{aligned}$$

Both $(\{\!\{Q\}\!\}, \varnothing) \approx^{\mathsf{nr}}_{bi} (\{\!\{P\}\!\}, \varnothing)$ and $(\{\!\{Q\}\!\}, \varnothing) \approx^{\mathsf{nr}}_{bi} (\{\!\{P\}\!\}, \varnothing)$ are proved by showing that the binary relation R, defined as the reflexive, symmetric and transitive closure of $\{((\{\!\{Q\}\!\}, \varnothing), (\{\!\{P\}\!\}, \varnothing)), ((\{\!\{P\}\!\}, \varnothing), (\{\!\{P'\}\!\}, \varnothing))\}$, is a bisimulation (see Figure 8a).

To prove that $\{\!\{Q\}\!\} \not\leqslant^{\mathsf{nr}}_{obs} \{\!\{P\}\!\}$, we show that $\{\!\{Q;\ \mathsf{in}(d,x).0\}\!\} \not\leqslant^{\mathsf{nr}}_{obs} \{\!\{P;\ \mathsf{in}(d,x).0\}\!\}$. In particular (see Figure 8b), we build a non-randomized scheduler such that $\{\!\{Q;\ \mathsf{in}(d,x).0\}\!\} \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)} D$ where $D = 0.45 \cdot \delta_{\mathcal{P}_a} + 0.45 \cdot \delta_{\mathcal{P}_b} + 0.1 \cdot \delta_{\varnothing}$. However, there is no distribution E such that $\{\!\{P;\ \mathsf{in}(d,x).0\}\!\} \overset{\tau}{\Longrightarrow}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)} E$, and $D \widehat{\leqslant^{\mathsf{nr}}_{obs}} E$. □

Remark that we have cast the definitions of [12] in our own framework. In the full version [28] we show that processes P, Q in Lemma 4 can be adapted to obtain the counterpart of Lemma 4 in the exact framework of [12]. The failure of the proof of [12, Theorem 2] can be traced back to the auxilliary lemma [12, Lemma 3] that states that bisimilarity is closed under application of closing evaluation contexts. No proof of this lemma is however provided, and it is actually false: as shown in the proof of (our) Lemma 4, the extended processes $(\{\!\{Q\}\!\}, \varnothing)$ and $(\{\!\{P\}\!\}, \varnothing)$ defined there are bisimilar (with respect to non-randomized schedulers), but it is not the case of the extended processes $(\{\!\{Q \mid \mathsf{in}(d,x).0\}\!\}, \varnothing)$ and $(\{\!\{P \mid \mathsf{in}(d,x).0\}\!\}, \varnothing)$.

5. 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, *e.g.*, [20, 21]. We illustrate this phenomenon on the following example.

Example 3. Consider the processes

$$\begin{aligned} P &:= (\mathsf{in}(c,x).\ \text{if}\ x = 0\ \text{then}\ \mathsf{out}(ok,1)\ \text{else}\ \mathsf{out}(bad,1)) +_{\frac{1}{2}} \\ &\qquad (\mathsf{in}(c,x).\ \text{if}\ x = 0\ \text{then}\ \mathsf{out}(bad,1)\ \text{else}\ \mathsf{out}(ok,1)) \\ Q &:= \mathsf{in}(c,x).(\mathsf{out}(ok,1) +_{\frac{1}{2}} \mathsf{out}(bad,1)) \end{aligned}$$

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

However, P and Q are not may testing equivalent and can be distinguished by the adversary $Adv = \{\!\{\mathsf{out}(c,0) \mid \mathsf{out}(c,1)\}\!\}$. Indeed, we can show that:

$$\mathrm{RProb}_{\mathcal{R}^o_{\mathsf{r}}}(\mathcal{P} \cup Adv, \downarrow ok) = 1 \quad \mathrm{RProb}_{\mathcal{R}^o_{\mathsf{r}}}(\mathcal{Q} \cup Adv, \downarrow ok) = \frac{1}{2}$$

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 section we identify two subclasses of processes that avoid this problem. The first such subclass is that of non-probabilistic processes, *i.e.*, without the $+_p$ operator (we denote by $\mathcal{MP}^{\mathsf{np}}$ all the multisets of such processes). This is the class of the original applied π-calculus which also enjoys good tool support. Figure 6 already illustrated that even on non-probabilistic processes, *probabilistic* adversaries have a stronger distinguishing power for the may testing equivalence. We formally characterize this distinguishing power when restricting protocols to a bounded number of sessions (denoted $\mathcal{MP}^{<\infty,\mathsf{np}}$), *i.e.*, considering processes without replication: for this subclass, may-testing coincides with similarity. We therefore inherit from [22] the fact that deciding may-testing is 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 [25]) corresponds to may-testing with a restricted, *determinate* adversary process. We also sketch how the algorithms of the DEEPSEC prover [22] could be adapted to check trace equivalence in this probabilistic setting.

5.1. Non-probabilistic processes

5.1.1. May-testing with non-probabilistic adversaries and trace equivalence coincide

Our definitions of may testing and trace equivalence coincide with the classical definitions of the original, purely non-deterministic applied π-calculus when all processes are non probabilistic. As a first step, we observe that the weak operational semantics we defined in Section 4.3 is a conservative extension of the weak (non-probabilistic) operational semantics: indeed, when considering non-probabilistic processes, all distributions in the (labeled) operational semantics are Dirac distributions.

Notation 5. We write $\mathcal{SP}^{\mathsf{np}}_{\ell}$ for the set of all non-probabibilistic extended processes We write $\rightarrow_{\mathsf{np}}$, respectively $\xrightarrow{a}_{\mathsf{np}}$, for the one-step reduction relation we obtain when we restrict the NPLTS N^o to $\mathcal{MP}^{\mathsf{np}}$, respectively N^{ℓ} to $\mathcal{SP}^{\mathsf{np}}_{\ell}$. For $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{\mathsf{np}}$, we write $\mathcal{P} \Rightarrow_{\mathsf{np}} \mathcal{Q}$ when there exists a sequence $\mathcal{P} = \mathcal{P}_0 \rightarrow_{\mathsf{np}} \ldots \rightarrow_{\mathsf{np}} \mathcal{P}_n = \mathcal{Q}$.

Lemma 5. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{\mathsf{np}}$.

$$\mathcal{P} \Rightarrow_{\mathsf{np}} \mathcal{Q} \quad \text{iff} \quad \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(\mathcal{P}, \{\mathcal{Q}\}) = 1$$

We look now at *preorder relations* between non-probabilistic processes. We first recall formally how may testing, trace equivalence and bisimulation are defined for non-probabilistic processes (Definition 17 below). Those definitions are coherent with those from the literature, e.g [23], (up to the difference on static equivalence, discussed in Remark 5).

Notation 6. If b is a barb, we write $\mathcal{P} \Downarrow_b$ when there exists $\mathcal{Q}$ such that $\mathcal{P} \Rightarrow_{\mathsf{np}} \mathcal{Q}$, and $\mathcal{Q} \downarrow_b$. For $a \in \mathcal{A}^{\ell}_{ext}$, we write $(\mathcal{P},\phi) \xRightarrow{a}_{\mathsf{np}} (\mathcal{Q},\psi)$ when $(\mathcal{P},\phi) \xrightarrow{\tau}_{\mathsf{np}} \dots \xrightarrow{a}_{\mathsf{np}} \dots \xrightarrow{\tau}_{\mathsf{np}} (\mathcal{Q},\psi)$. If $\alpha = a_1,\dots,a_n$ is a trace, we write $(\mathcal{P},\phi) \xRightarrow{\alpha}_{\mathsf{np}} (\mathcal{Q},\psi)$ when there exists a sequence $(\mathcal{P},\phi) \xRightarrow{a_1}_{\mathsf{np}} \dots \xRightarrow{a_n}_{\mathsf{np}} (\mathcal{Q},\psi)$.

Definition 17. We define the binary relations $\leqslant^{\mathsf{np}}_{may}$, $\leqslant^{\mathsf{np}}_{tr}$, $\leqslant^{\mathsf{np}}_{sim}$ on $\mathcal{MP}^{\mathsf{np}}$ as follows:

- $\mathcal{P} \leqslant^{\mathsf{np}}_{may} \mathcal{Q}$ when $\forall Adv \in \mathcal{MP}^{\mathsf{np}}$ s.t. $fn(Adv) \subseteq \mathcal{N}_{pub}$. $\forall c \in \mathcal{N}_{pub}$. $Adv \cup \mathcal{P} \Downarrow_c$ implies $Adv \cup \mathcal{Q} \Downarrow_c$;
- $(\mathcal{P},\phi) \leqslant^{\mathsf{np}}_{tr} (\mathcal{Q},\psi)$ when for every trace α, $(\mathcal{P},\phi) \xRightarrow{\alpha}_{\mathsf{np}} (\mathcal{P}',\phi')$ implies $(\mathcal{Q},\psi) \xRightarrow{\alpha}_{\mathsf{np}} (\mathcal{Q}',\psi')$;
- $\leqslant^{\mathsf{np}}_{sim}$ is the largest reflexive and transitive relation R such that $(\mathcal{P},\phi)\ R\ (\mathcal{Q},\psi)$ implies that for every $a \in \mathcal{A}^{\ell}_{ext} \cup \{\tau\}$, and $(\mathcal{P},\phi) \xrightarrow{a}_{\mathsf{np}} (\mathcal{P}',\phi')$, there exists $(\mathcal{Q}',\psi')$ such that $(\mathcal{Q},\psi) \xRightarrow{a}_{\mathsf{np}} (\mathcal{Q}',\psi')$ and $(\mathcal{P}',\phi')\ R\ (\mathcal{Q}',\psi')$.

The preorders $\leqslant_{sim}$ and $\leqslant_{tr}$–and the corresponding equivalence relations–are conservative extensions of $\leqslant^{\mathsf{np}}_{sim}$ and $\leqslant^{\mathsf{np}}_{tr}$. As expected, the preorder $\leqslant_{may}$ *is not* a conservative extension of $\leqslant^{\mathsf{np}}_{may}$, because of the additional expressive power of probabilistic adversaries. Nonetheless, we can recover $\leqslant^{\mathsf{np}}_{may}$ when we restrict the adversaries in the definition of $\leqslant_{may}$ to non-probabilistic adversaries.

Proposition 2. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{\mathsf{np}}$.

- $(\mathcal{P},\varnothing) \leqslant^{\mathsf{np}}_{sim} (\mathcal{Q},\varnothing)$ iff $(\mathcal{P},\varnothing) \leqslant_{sim} (\mathcal{Q},\varnothing)$;
- $(\mathcal{P},\varnothing) \leqslant^{\mathsf{np}}_{tr} (\mathcal{Q},\varnothing)$ iff $(\mathcal{P},\varnothing) \leqslant_{tr} (\mathcal{Q},\varnothing)$;
- $(\mathcal{P},\varnothing) \leqslant^{\mathsf{np}}_{may} (\mathcal{Q},\varnothing)$ iff $\begin{array}{l} \forall Adv \in \mathcal{MP}^{\mathsf{np}} \text{ s.t. } fn(Adv) \subseteq \mathcal{N}_{pub}.\ \forall c \in \mathcal{N}_{pub}. \\ \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(\mathcal{P} \cup Adv, \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(\mathcal{Q} \cup Adv, \downarrow c) \end{array}$

Proof sketch. For may-testing and trace preorder, the proof uses crucially the fact that it is enough to consider non-randomized distributions (thanks to Lemma 2), and from there, we conclude using Lemma 5. Since it is not possible to consider only non-randomized schedulers in the definition of simulation, the proof of the first point in Proposition 2 is more subtle, and uses well-structured properties of the lifting of a relation from Definition 9. We need to show (1) that the (non-probabilistic) simulation $\leqslant^{\mathsf{np}}_{sim}$ is a probabilistic simulation in the sense of Definition 16, and (2) that $\leqslant_{sim}$ is also a non-probabilistic simulation in the sense of Definition 17.

- Suppose that $\mathcal{P} \leqslant^{\mathsf{np}}_{sim} \mathcal{Q}$ for $\mathcal{P}, \mathcal{Q} \in \mathcal{SP}^{\mathsf{np}}_{\ell}$. Let $a \in \mathcal{A}^{\ell}_{ext} \cup \{\tau\}$, $D \in \mathcal{D}(\mathcal{S}_{\mathsf{N}^{\ell}})$ such that $\mathcal{P} \xrightarrow{a}_{\mathsf{r}} D$. Looking at the way we defined $\xrightarrow{a}_{\mathsf{r}}$ (and since $\mathcal{P}$ is non-probabilistic), we also see that $\mathcal{P} \xrightarrow{a}_{\mathsf{np}} \mathcal{P}'_i$ for every $\mathcal{P}'_i$ is the support of D. From there, we obtain that for each i, there exists $\mathcal{Q}'_i$ such that $\mathcal{Q} \xRightarrow{a}_{\mathsf{np}} \mathcal{Q}'_i$, and $\mathcal{P}'_i \leqslant^{\mathsf{np}}_{sim} \mathcal{Q}'_i$. At that point, we build $E = \sum_i D(\mathcal{P}'_i) \cdot \delta_{\mathcal{Q}'_i}$, and we can see that $\mathcal{Q} \xRightarrow{a}_{\mathsf{r}} E$. Moreover, the structural properties of the lifting allows us to go from $(\forall i,\ \mathcal{P}'_i \leqslant^{\mathsf{np}}_{sim} \mathcal{Q}'_i)$ to $D \mathrel{\widehat{\leqslant^{\mathsf{np}}_{sim}}} E$. Hence, we have shown that $\leqslant^{\mathsf{np}}_{sim}$ is indeed a probabilistic simulation in the sense of Definition 16.
- Suppose that $\mathcal{P} \leqslant_{sim} \mathcal{Q}$ for $\mathcal{P}, \mathcal{Q} \in \mathcal{SP}^{\mathsf{np}}_{\ell}$. Let $a \in \mathcal{A}^{\ell}_{ext} \cup \{\tau\}$, and $\mathcal{P}' \in \mathcal{SP}^{\mathsf{np}}_{\ell}$ such that $\mathcal{P} \xrightarrow{a}_{\mathsf{np}} \mathcal{P}'$. This transition carries over to N^{ℓ}, i.e., $\mathcal{P} \xrightarrow{a}_{\mathsf{r}} \delta_{\mathcal{P}'}$. We obtain that there exists a distribution E such that $\mathcal{Q} \xRightarrow{a}_{\mathsf{r}} E$, and $\delta_{\mathcal{P}'} \widehat{\leqslant_{sim}} E$. But by structural property of the lifting, we have that $\mathcal{P}' \leqslant_{sim} \mathcal{Q}'$ for every element $\mathcal{Q}'$ in the support of E. Moreover, since $\mathcal{Q}$ is non-probabilistic, it holds that $\mathcal{Q} \xRightarrow{a}_{\mathsf{np}} \mathcal{Q}'$ for every element $\mathcal{Q}'$ in the support of E. Since E is a distribution, there exists at least one such element $\mathcal{Q}'$, thus we can conclude. □

The following result indicates 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, as shown in [23].

Proposition 3. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{\mathsf{np}}$.

$$(\mathcal{P}, \varnothing) \leqslant_{tr} (\mathcal{Q}, \varnothing) \quad \text{iff} \quad \begin{array}{c} \forall Adv \in \mathcal{MP}^{\mathsf{np}} \text{ s.t. } fn(Adv) \subseteq \mathcal{N}_{pub}.\ \forall c \in \mathcal{N}_{pub}. \\ \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(\mathcal{P} \cup Adv, \downarrow c) \leqslant \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(\mathcal{Q} \cup Adv, \downarrow c) \end{array}$$

5.1.2. May-testing and simulation coincide for bounded processes

We rely on a modal characterization of strong simulation on *image finite labeled transition systems* (LTS) by a Hennessy-Milner logic [32] (HML).

We can rely on *strong* simulation as it is a well-known fact ([33]) that simulation for a LTS can be expressed as strong simulation on the corresponding *weak LTS*, that is, in our case all transitions $(\mathcal{P}, \phi) \xrightarrow{\tau}^* \xrightarrow{a} \xrightarrow{\tau}^* (\mathcal{Q}, \psi)$ are merged into a single transition.

A LTS is image finite when the LTS cannot infinitely branch from a state and a label. Therefore, as we consider only bounded processes and by denoting $\leqslant_{ssim}^{\mathsf{L}}$ the strong simulation relation on a LTS L, we can build an image finite LTS L^ℓ such that for all $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{\mathsf{np}}$:

$$(\mathcal{P}, \varnothing) \leqslant_{sim} (\mathcal{Q}, \varnothing) \quad \text{iff} \quad (\mathcal{P}, \varnothing) \leqslant_{ssim}^{\mathsf{L}^\ell} (\mathcal{Q}, \varnothing)$$

Our HML characterization consists in expressing strong simulation preorder by the means of satisfaction of *logical formulas* by the LTS.

Definition 18. Let $\mathcal{A}$ be a countable set of actions. We define the set of *logical formulas* as:

$$F \in \mathcal{F} := \top \mid a.F \mid F_1 \wedge F_2, \qquad \text{where } a \in \mathcal{A}$$

In our case, the set of actions corresponds to $\mathcal{A}_{ext}^\ell$ that is indeed countable. The satisfaction of such formulas by a LTS is defined as follows.

Definition 19. Let $\mathsf{L} = (\mathcal{S}, \mathcal{A}, \rightarrow)$ be a LTS. We say that L satisfies a formula F, written $s \models F$, if for all $s \in \mathcal{S}$,

- $s \models \top$;
- $s \models a.F$ when $s \xrightarrow{a} t$ and $t \models F$;
- $s \models F_1 \wedge F_2$ when $s \models F_1$ and $s \models F_2$.

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

Proposition 4 (HML caracterisation of simulation)**.** For an image finite LTS L,

$$s \leqslant_{ssim}^{\mathsf{L}} t \quad \text{iff} \quad \forall F \in \mathcal{F}.\ s \models F \text{ implies } t \models F$$

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_F^c such that for all bounded non-probabilistic extended processes $(\mathcal{P}, \phi)$,

$$(\mathcal{P}, \varnothing) \models F \quad \text{iff} \quad \mathrm{RProb}_{\mathcal{R}_r}(\mathcal{P} \cup \{\!\!\{Adv_F^c\}\!\!\}, \downarrow c) = 1$$

The construction of Adv_F^c is defined as follows.

Definition 20. Let F be a formula in $\mathcal{F}$, $ok \in \mathcal{N}_{pub}$ such that $ok \notin fn(F)$ and $n \in \mathbb{N}$. We let $\{\!\!\{Adv_F^c\}\!\!\} = Adv_{F,0}^{ok}$ and define $Adv_{F,n}^{ok}$ by induction on the syntax of F:

- if $F = \top$ then $Adv_{F,n}^{ok} = \mathsf{out}(ok, ok)$.
- if $F = in(\xi, \zeta).F'$ then $Adv_{F,n}^{ok} = \mathsf{out}(\xi, \zeta); Adv_{F',n}^{ok}$ when $vars(\xi, \zeta) \subseteq \mathcal{AX}_n$ and $Adv_{F,n}^{ok} = 0$ otherwise.
- if $F = out(\xi, \mathsf{ax}).F'$ then $Adv_{F,n}^{ok} = \mathsf{in}(\xi, \mathsf{ax}); Adv_{F',n+1}^{ok}$ when $\mathsf{ax} = \mathsf{ax}_{n+1}$, $vars(\xi) \subseteq \mathcal{AX}_n$ and $Adv_{F,n}^{ok} = 0$ otherwise.
- if $F = (\xi \overset{?}{=} \zeta).F'$ then $Adv_{F,n}^{ok} = \mathsf{if}\ \xi = \zeta\ \mathsf{then}\ Adv_{F',n}^{ok}$ when $vars(\xi, \zeta) \subseteq \mathcal{AX}_n$ and $Adv_{F,n}^{ok} = 0$ otherwise.
- if $F = (\xi \overset{?}{\neq} \zeta).F'$ then $Adv_{F,n}^{ok} = \mathsf{if}\ \xi = \zeta\ \mathsf{then}\ 0\ \mathsf{else}\ Adv_{F',n}^{ok}$ when $vars(\xi, \zeta) \subseteq \mathcal{AX}_n$ and $Adv_{F,n}^{ok} = 0$ otherwise.
- if $F = F_1 \wedge F_2$, then $Adv_{F,n}^{ok} = Adv_{F_1,n}^{ok} +_{\frac{1}{2}} Adv_{F_2,n}^{ok}$.

In Definition 20, the integer n and the conditions on the variables of ξ, ζ ensure that the adversarial process $Adv_{F,n}^{ok}$ is closed (no free variables). This is not a restriction as $(\mathcal{P}, \varnothing) \models F$ implies that F satisfies these conditions. In particular, we note that conjunction is encoded by probabilistic choice and on formula $\top$, the adversary process exhibits the barb c. The main result of this section follows almost directly.

Proposition 5. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{<\infty,\mathsf{np}}$.

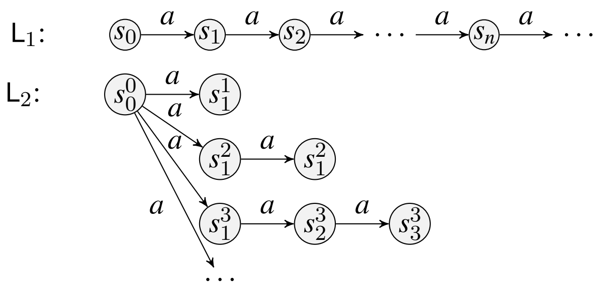
$$(\mathcal{P}, \varnothing) \leqslant_{sim} (\mathcal{Q}, \varnothing) \quad \text{iff} \quad \mathcal{P} \leqslant_{may} \mathcal{Q}$$

Cheval *et al.* have shown [22] 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 [22] 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 proves hardness of similarity. The coNEXPTIME decision procedure for bisimilarity can be directly adapted to the case of similarity, hence, we have the following result.

Corollary 1. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{<\infty,\mathsf{np}}$. Deciding $\mathcal{P} \approx_{may} \mathcal{Q}$ is coNEXPTIME complete when $\doteq$ is defined by a subterm convergent destructor rewrite system.

5.1.3. May-testing and simulation do not coincide for unbounded processes

We consider the following two simple LTS L_1 and L_2, that are an asymetric variant of the LTSs used in [34] to show that the finitary HML does not characterize 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.*, $\mathsf{L}_1 \not\leqslant_{sim} \mathsf{L}_2$.

In the applied π-calculus, L_1 can be represented by the process $!\mathsf{out}(c,a)$. Modeling L_2 is more complex: we non-deterministically choose an integer $n \geqslant 0$, and output $n+1$ times a. The non deterministic choice of n is realized through the following process that outputs the unary encoding $h^n(b)$ of n:

$$Q_{count}(e) = \mathsf{new}\ d.\ (\mathsf{out}(d,b) \mid !\mathsf{in}(d,x).\ (\mathsf{out}(e,x) + \mathsf{out}(d,h(x))))$$

Channel d represents a memory cell initiated with a public name b (encoding 0). When reading on 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:

$$Q_{out}(e) = \mathsf{new}\ d'.\mathsf{in}(e,x).\ \big(\mathsf{out}(d',b) \mid !\mathsf{in}(d',y).\mathsf{out}(c,a).\mathsf{if}\ x = y\ \mathsf{then}\ 0\ \mathsf{else}\ \mathsf{out}(d',h(y))\big)$$

Similarly to $Q_{count}(e)$, $Q_{out}(e)$ 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 6. Let $Q = \mathsf{new}\ e.(Q_{count}(e) \mid Q_{out}(e))$. We have:

$$(!\mathsf{out}(c,a),\varnothing) \not\leqslant_{sim}^{\mathsf{N}^\ell} (Q,\varnothing) \quad \text{but} \quad !\mathsf{out}(c,a) \leqslant_{may} Q$$

Proof sketch. $(!\mathsf{out}(c,a),\varnothing) \not\leqslant_{sim}^{\mathsf{N}^\ell} (Q,\varnothing)$ is shown by relying on the same idea used to show $\mathsf{L}_1 \not\leqslant_{sim} \mathsf{L}_2$. Before the first output on c, Q must produce a communication on e, hence fixing the value of x used in $Q_{out}(e)$ which bounds the number of following outputs on c by $x+1$. As $!\mathsf{out}(c,a)$ may output on c an unbounded number of times, we conclude.

To prove that $!\mathsf{out}(c,a) \leqslant_{may} Q$, we unfold the definition of $\mathrm{RProb}_{\mathcal{R}}(\{\!\{!\mathsf{out}(c,a)\}\!\} \cup Adv, \downarrow ch)$ and focus on $\mathrm{RProb}_R^{\leqslant n}(s, \mathsf{corr}^{-1}(\downarrow ch))$ where (corr, R) is a resolution, $n \in \mathbb{N}$ and $\mathsf{corr}(s) = \{\!\{!\mathsf{out}(c,a)\}\!\} \cup Adv$. One can notice that by definition, the number of transitions from s in the computation of $\mathrm{RProb}_{\mathsf{R}}^{\leqslant n}(s, \mathsf{corr}_{\mathsf{R}}^{-1}(\downarrow ch))$ is bounded by n. Thus, the resolution R may at most unfold n times the process $!\mathsf{out}(c,a)$. By denoting $P^n = \underbrace{\mathsf{out}(c,a);\ldots;\mathsf{out}(c,a)}_{n\text{ times}}$, we can build a resolution whose probability to exhibit the barb ch from $\{\!\{P^n\}\!\} \cup Adv$ is the same as $\mathrm{RProb}_{\mathsf{R}}^{\leqslant n}(s, \mathsf{corr}_{\mathsf{R}}^{-1}(\downarrow ch))$. We then rely on $(P^n,\varnothing) \leqslant_{sim} (Q,\varnothing)$ and Theorem 2 to conclude that $\mathrm{RProb}_{\mathsf{R}}^{\leqslant n}(s, \mathsf{corr}_{\mathsf{R}}^{-1}(\downarrow ch)) \leqslant \mathrm{RProb}_{\mathcal{R}}(\{\!\{Q\}\!\} \cup Adv, \downarrow ch)$ which allows us to conclude. □

5.2. Purely probabilistic processes

At the opposite of the spectrum of purely non-deterministic processes, we study *purely probabilistic* processes with (nearly) no non-determinism. A similar class of processes has been considered in [25, 26] to model various protocols relying on randomization (*e.g.*, Crowds [10], mix-net [9], electronic voting [35]). They consider systems that are built as the parallel composition of independant agents, called *roles*, and where all communications are mediated by the adversary. Moreover, the internal behavior of each role is deterministic; the only non-determinism is controlled by the adversary–thus external–and consists in the adversary's choice for scheduling the communications. We model purely probabilistic processes as follows.

Definition 21. A process is *fully determinate* if it does not contain the operators $+$, $|$, nor $!$.
$\mathcal{P} = \{P_1, \dots, P_n\} \in \mathcal{MP}$ is *purely probabilistic* when:

- each P_i is fully determinate;
- there exist distinct public channels $c_1, \dots, c_n \in \mathcal{N}_{pub}$ such that for all $i \in \{1, \dots, n\}$, all input and output actions in P_i are on c_i.

The class of purely probabilistic multisets of processes is denoted by $\mathcal{MP}^{\mathsf{pp}}$.

$\mathcal{MP}^{\mathsf{pp}}$ can also be seen as a probabilistic extension of the class of *simple processes* introduced in [36] to show that trace equivalence coincides with observational equivalence for such processes.

5.2.1. Removing scheduling of τ-actions

In [25, 26], the authors consider trace equivalence for a slightly restricted fragment of purely probabilistic processes. More precisely, all processes have exactly the same control structure which removes the necessity of scheduling honest τ-actions and allows to directly consider *strong* trace equivalence, as exactly the same τ-actions occur. In this work, we lift this restriction on the shape of the processes and show instead that the non-determinism related to honest τ actions is inconsequential when deciding trace equivalence. Indeed, in a multiset of processes $\{P_1, \dots, P_n\}$, a τ action may be available simultaneously in multiple components. However, all such τ-action are in fact purely deterministic (*e.g.*, conditional branching, probabilistic choice). Moreover, as the P_is do not contain parallel composition and all input and output occur on distinct channels c_i, no internal communication between processes P_i and P_j is possible.

We show that to compute the probability of executing a trace w, we only need to consider a single *maximal* resolution on the NPLTS N^ℓ. Such a resolution always executes an action when at least one is available. Formally, a resolution (corr, R) with $R = (S, \mathcal{A}, \mathsf{trans})$ on purely probabilistic processes is maximal when for all $s \in S$, if there exists $\mathsf{corr}(s) \xrightarrow{a} D$ in N^ℓ for some a, D then $\mathsf{trans}(s)(a) = D'$ for some D'.

Proposition 6. Let $(\mathcal{P}, \phi)$ be an extended purely probabilistic process. For all maximal resolutions (corr, R) on N^ℓ, for all $s \in \mathcal{S}(R)$ with $\mathsf{corr}(s) = (\mathcal{P}, \phi)$,

$$\forall w \in \mathcal{A}_{ext}^{\star}.\ \mathrm{Prob}_R(s, w) = \mathrm{Prob}_{\mathcal{R}_\tau^\ell}((\mathcal{P}, \phi), w).$$

5.2.2. May-testing and trace equivalence coincide for fully determinate adversaries

As illustrated in Example 3, may-testing is strictly stronger than trace equivalence even on purely probabilistic processes due to the non-determinism in the adversarial process. However, by restricting the adversarial process to be fully determinate, we can show that may-testing and trace equivalence coincide. We define the resulting *determinate may testing preorder*, denoted $\leqslant_{d\text{-}may}$, exactly as in Definition 7 but additionally restrict the adversary process Adv to be a singleton $\{\!\{A\}\!\}$ where A is fully determinate.

Theorem 3. Let $\mathcal{P}, \mathcal{Q} \in \mathcal{MP}^{\mathsf{pp}}$.

$$\mathcal{P} \leqslant_{d\text{-}may} \mathcal{Q} \quad \text{iff} \quad (\mathcal{P}, \varnothing) \leqslant_{tr} (\mathcal{Q}, \varnothing)$$

Proof sketch. To prove that $(\mathcal{P}, \varnothing) \leqslant_{d\text{-}may} (\mathcal{Q}, \varnothing)$ implies $\mathcal{P} \leqslant_{tr} \mathcal{Q}$ we encode any trace w into a determinate adversary Adv^c_w where c is fresh. Adv^c_w is defined in a similar way as Adv^c_F (Section 5.1), *e.g.*,

$$Adv^c_{in(\xi,\zeta).w'} = out(\xi,\zeta); Adv^c_{w'} \qquad \text{and} \qquad Adv^c_{(\xi \overset{?}{=} \zeta).w'} = \mathsf{if}\ \xi = \zeta\ \mathsf{then}\ Adv^c_{w'}$$

In particular, on the empty trace the adversary process exhibits the barb c: $Adv^c_\varepsilon = \mathsf{out}(c,c)$. We obtain that

$$\mathrm{Prob}_{\mathcal{R}_{\mathsf{r}}(\mathsf{N}^\ell)}((\mathcal{P}, \varnothing), w) = \mathrm{RProb}_{\mathcal{R}^o_{\mathsf{r}}}(\mathcal{P} \cup \mathcal{A}_{dv}, \downarrow c)$$

The other implication is more difficult as the adversarial process Adv is allowed to use probabilistic choices which cannot be directly encoded in a trace. Instead, we show that any adversarial process Adv aiming to exhibit a barb c corresponds to a multiset of weighted traces $\mathsf{Tr}(Adv)$, built inductively on Adv. For instance, when $Adv = Adv_1 +_p Adv_2$ and

$$\mathsf{Tr}(Adv_i) = \{\!\{(p^i_k, w^i_k)\}\!\}^{n_i}_{k=1} \quad \text{for } i = 1, 2$$

then

$$\mathsf{Tr}(Adv) = \{\!\{(p \cdot p^1_k, w^1_k)\}\!\}^{n_1}_{k=1} \cup \{\!\{((1-p) \cdot p^2_k, w^2_k)\}\!\}^{n_2}_{k=1}$$

For other constructs, the set of weighted traces is built as expected, *e.g.*, $\mathsf{Tr}(0) = \varnothing$ and $\mathsf{Tr}(\mathsf{new}\ a; Adv') = \mathsf{Tr}(Adv')$. For outputs or inputs, we additionally test if the channel corresponds to the barb c, *i.e.*, when $Adv = \mathsf{out}(\xi,\zeta); Adv'$ and $\mathsf{Tr}(Adv') = \{\!\{(p_k, w_k)\}\!\}^n_{k=1}$,

$$\mathsf{Tr}(Adv) = \{\!\{(p_k, (\xi \overset{?}{\neq} c).in(\xi,\zeta).w_k)\}\!\}^n_{k=1} \cup \{\!\{(1, \xi \overset{?}{=} c)\}\!\}$$

This construction yields the following property:

$$\begin{aligned}\mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(P \cup Adv, \downarrow c) =& \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(P, \downarrow c)\\ &+ (1 - \mathrm{RProb}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^o)}(P, \downarrow c)) \cdot \sum_{(\alpha,w) \in \mathsf{Tr}(Adv)} \alpha \cdot \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((P, \varnothing), w)\end{aligned}$$

Intuitively, exhibiting the barb on c does not require any interaction with the adversary, or this interaction is correctly encoded in $\mathsf{Tr}(Adv)$. Using this property we easily conclude. □

6. Deciding trace equivalence and tool support

As previous mentioned, Cheval *et al.* [22] 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 Lemma 2, proving $(\mathcal{P}, \phi) \leqslant_{tr} (\mathcal{P}', \phi')$ is equivalent to proving $(\mathcal{P}, \phi) \leqslant_{tr}^{\mathsf{nr}} (\mathcal{P}', \phi')$. Thus, by definition, we focus on the computation of $\mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), w)$ 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 7. Let $(\mathcal{P}, \phi) \in \mathcal{SP}_\ell^{<\infty}$. Let $a \in \mathcal{A}_{ext}^{\ell}$. Let $w \in {\mathcal{A}_{ext}^{\ell}}^*$.

$$\mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), \epsilon) = 1 \qquad \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), a.w) = \max(p_1, p_2)$$

where

$$p_1 = \max_{(\mathcal{P},\phi) \to_\tau D} \sum_{(\mathcal{P}',\phi') \in \mathrm{supp}(D)} D((\mathcal{P}', \phi')) \cdot \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), a.w)$$
$$p_2 = \max_{(\mathcal{P},\phi) \to_a D} \sum_{(\mathcal{P}',\phi') \in \mathrm{supp}(D)} D((\mathcal{P}', \phi')) \cdot \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), w)$$

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.

As mentioned above, partition trees [22] 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 $\mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\mathcal{P}, \phi), w)$, Lemma 7 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 $\mathsf{in}^\ell(u, x); P$, $\mathsf{out}^\ell(u, v); P$ and $P +_p^\ell Q$, $P +^\ell Q$ respectively. Additionally, we assume that all labels are distinct in the initial processes.

Definition 22. We define *history entries* as elements from $\{\mathsf{h}_1(\ell), \mathsf{h}_2(\ell, \ell'), \mathsf{h}_+(\ell, p, i), \mathsf{h}_c(\ell, i) \mid i \in \{0, 1\}, p \in]0, 1[, \ell \text{ label}\}$. 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 $(\mathcal{P},\phi) \to_a D$ and $(\mathcal{P}',\phi') \in \mathrm{supp}(D)$, we define $(\mathcal{P},\phi,\mathsf{H}) \xrightarrow{a} (\mathcal{P}',\phi',\mathsf{H}')$ where:

- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{+}(\ell,p,0)$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{P +_p^\ell Q\}\!\}$ and $\mathcal{P}' = \mathcal{Q} \cup \{\!\{P\}\!\}$;
- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{+}(\ell,p,1)$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{P +_p^\ell Q\}\!\}$ and $\mathcal{P}' = \mathcal{Q} \cup \{\!\{Q\}\!\}$;
- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{c}(\ell,0)$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{P +^\ell Q\}\!\}$ and $\mathcal{P}' = \mathcal{Q} \cup \{\!\{P\}\!\}$;
- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{c}(\ell,1)$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{P +^\ell Q\}\!\}$ and $\mathcal{P}' = \mathcal{Q} \cup \{\!\{Q\}\!\}$;
- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{2}(\ell,\ell')$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{\mathsf{out}^\ell(u,t).P, \mathsf{in}^{\ell'}(v,x).Q\}\!\}$, $\mathcal{P}' = \mathcal{Q} \cup \{\!\{P, Q\{x \mapsto t\}\}\!\}$;
- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{1}(\ell)$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{\mathsf{out}^\ell(u,t).P\}\!\}$ and $a = out(\xi,\mathsf{ax}_n)$ and $\mathcal{P}' = \mathcal{Q} \cup \{\!\{P\}\!\}$;
- $\mathsf{H}' = \mathsf{H} \cdot \mathsf{h}_{1}(\ell)$ when $\mathcal{P} = \mathcal{Q} \cup \{\!\{\mathsf{in}^\ell(u,x).P\}\!\}$ and $a = in(\xi,\zeta)$ and $\mathcal{P}' = \mathcal{Q} \cup \{\!\{P\{x \mapsto \zeta\phi\}\}\!\}$;
- $\mathsf{H}' = \mathsf{H}$ otherwise.

Given $w \in {\mathcal{A}_{ext}^{\ell}}^*$, we write $(\mathcal{P},\phi,\mathsf{H}) \overset{w}{\Rightarrow} (\mathcal{P}',\phi',\mathsf{H}')$ when $(\mathcal{P},\phi,\mathsf{H}) \xrightarrow{a_1} \ldots \xrightarrow{a_n} (\mathcal{P}',\phi',\mathsf{H}')$ and w is $a_1 \ldots a_n$ with the τ 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

$$(\mathcal{P},\phi,\mathsf{H}) \overset{w}{\Rightarrow} (\mathcal{P}_1,\phi_1,\mathsf{H}_0 \cdot \mathsf{H}_1) \quad \text{and} \quad (\mathcal{P},\phi,\mathsf{H}) \overset{w}{\Rightarrow} (\mathcal{P}_2,\phi_2,\mathsf{H}_0 \cdot \mathsf{H}_2)$$

then there exist w_0, w_1 and an extended process $(\mathcal{P}_0,\phi_0,\mathsf{H}_0)$ such that $w = w_0 w_1$ and

$$\begin{tikzcd}
 & & (\mathcal{P}_1,\phi_1,\mathsf{H}_0 \cdot \mathsf{H}_1) \\
(\mathcal{P},\phi,\mathsf{H}) \arrow[r, Rightarrow, "w_0"] & (\mathcal{P}_0,\phi_0,\mathsf{H}_0) \arrow[ur, Rightarrow, "w_1"] \arrow[dr, Rightarrow, "w_1"'] & \\
 & & (\mathcal{P}_2,\phi_2,\mathsf{H}_0 \cdot \mathsf{H}_2)
\end{tikzcd}$$

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

Definition 23. Let S be a set of histories. We denote by $S_{|h} = \{\mathsf{H}' \mid (h \cdot \mathsf{H}') \in S\}$. We define $\mathsf{compute}(S)$ inductively as follows:

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

$$\mathsf{compute}(S) = \max \begin{cases} \max\limits_{\ell} \mathsf{compute}(S_{|\mathsf{h}_1(\ell)}) \\ \max\limits_{\ell,\ell'} \mathsf{compute}(S_{|\mathsf{h}_2(\ell,\ell')}) \\ \max\limits_{\ell,i} \mathsf{compute}(S_{|\mathsf{h}_c(\ell,i)}) \\ \max\limits_{\ell,p} (p \cdot \mathsf{compute}(S_{|\mathsf{h}_+(\ell,p,0)}) + (1-p) \cdot \mathsf{compute}(S_{|\mathsf{h}_+(\ell,p,1)})) \end{cases}$$

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

Lemma 8. Let (P,ϕ) be an extended process, $w \in {\mathcal{A}^{\ell}_{ext}}^*$ and $S = \{\mathsf{H}' \mid (\mathcal{P},\phi,[]) \overset{w}{\Rightarrow} (P',\phi',\mathsf{H}')\}$. We have that $\mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}(\mathsf{N}^\ell)}}((\mathcal{P},\phi),w) = \mathsf{compute}(S)$.

This lemma provides the core property that allows us to update the partition trees generated by the procedure in [22] as well as the test performed on each node of this partition tree to conclude trace preorder.

Theorem 4. Let $P, Q \in \mathcal{MP}^{<\infty}$ and $\doteq$ be defined by a subterm convergent destructor rewrite system. Trace preorder $(\{\!\{P\}\!\},\varnothing) \leqslant_{tr} (\{\!\{Q\}\!\},\varnothing)$ is decidable.

Proof sketch. First, we show that we can restrict the set of traces we need to verify. In particular, it suffices to look at traces $w \in {\mathcal{A}^{\ell}_{ext}}^*$ that can be split in two parts: $w = w_1 \cdot w_2$ where w_2 only consists of actions of the form $\xi \doteq \zeta$ or $\xi \not\doteq \zeta$, and w_1 of any other kind of actions. 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 $w_2^1,\dots,w_2^n$ built only on static equivalence tests. To do this, we use a proof technique similar to the proof in [23] showing that trace equivalence and may testing coincide for processes with bounded number of sessions. Let us define the sets $S_P(w_1) = \{(\mathcal{P}',\phi',\mathsf{H}') \mid (\{\!\{P\}\!\},\varnothing,[]) \overset{w_1}{\Longrightarrow} (\mathcal{P}',\phi',\mathsf{H}')\}$ and $S_Q(w_1) = \{(\mathcal{P}',\phi',\mathsf{H}') \mid (\{\!\{Q\}\!\},\varnothing,[]) \overset{w_1}{\Longrightarrow} (\mathcal{P}',\phi',\mathsf{H}')\}$. Since P and Q are bounded processes, $S_P(w_1)$, $S_Q(w_1)$ and $(S_P(w_1) \cup S_Q(w_1))\backslash_\sim$ are finite (here $\sim$ refers to static equivalence of frames). Therefore, for all $(\mathcal{P}',\phi',\mathsf{H}') \in S_P(w_1) \cup S_Q(w_1)$, we can define a finite sequence of actions w_2 built only on actions of the form $\xi \doteq \zeta$ or $\xi \not\doteq \zeta$ that exactly defines the equivalence class of $(\mathcal{P}',\phi',\mathsf{H}')$ in $(S_P \cup S_Q)\backslash_\sim$, i.e., for all $(\mathcal{P}'',\phi'',\mathsf{H}'') \in S_P(w_1) \cup S_Q(w_1)$, $\phi' \sim \phi''$ if and only if $(\mathcal{P}'',\phi'',\mathsf{H}'') \overset{w_2}{\Longrightarrow} (\mathcal{P}'',\phi'',\mathsf{H}'')$. Thus, if we denote by W this set of traces sufficient for proving trace preorder, we have that $(\{\!\{P\}\!\},\varnothing) \leqslant_{tr} (\{\!\{Q\}\!\},\varnothing)$ if and only if for all $w \in W$, $\mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}(\mathsf{N}^\ell)}}((\{\!\{P\}\!\},\varnothing),w) \leqslant \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}(\mathsf{N}^\ell)}}((\{\!\{Q\}\!\},\varnothing),w)$.

Second, similarly to how we augmented our labelled semantics with histories, we also augment the symbolic semantics used in [22] to generate partition trees with histories. As the addition of histories in symbolic extended processes does not impact in any way the generation of the partition trees, we can deduce from [22] that there exists a partition tree from P and Q, denoted $\mathsf{PTree}(P,Q)$. Thus, each node η of the partition tree will now contain a set $\Gamma(\eta)$ of symbolic processes of the form $(\mathcal{P}_c,\phi_c,\mathcal{C},\mathsf{H})$ where $(\mathcal{P}_c,\phi_c)$ is an open extended processes, $\mathcal{C}$ is a set of *constraints* and H is a history. The set $\Gamma(\eta)$ can furthermore be divided into the sets of symbolic processes coming from P and from Q, respectively $\Gamma_P(\eta)$ and $\Gamma_Q(\eta)$. If we denote by $\mathsf{H}(S)$ the set of histories of the (symbolic) extended processes in S, our decision procedure consists in checking that for all nodes $\eta \in \mathsf{PTree}(P,Q)$, $\mathsf{compute}(\mathsf{H}(\Gamma_P(\eta))) \leqslant \mathsf{compute}(\mathsf{H}(\Gamma_Q(\eta)))$.

To show that it indeed decides trace preorder, we recall that soundness of partition trees ensures that the nodes of the partition tree rooted by $(\{\!\{P\}\!\},\varnothing,[])$ and $(\{\!\{Q\}\!\},\varnothing,[])$ contain the symbolic processes representing all concrete extended processes reached after executing some trace w and that are statically equivalent. Moreover, completeness of partition trees ensures that for all concrete traces $w \in W$, there exists a node in the partition tree such that if $(\{\!\{P\}\!\},\varnothing,[]) \overset{w}{\Rightarrow} (\mathcal{P}',\phi',\mathsf{H}')$ or $(\{\!\{Q\}\!\},\varnothing,[]) \overset{w}{\Rightarrow} (\mathcal{P}',\phi',\mathsf{H}')$ then $\Gamma(\eta)$ contains a symbolic process representing $(\mathcal{P}',\phi',\mathsf{H}')$.

Formally, we show that:

- Soundness: $\forall \eta \in \mathsf{PTree}(P, Q), \exists w \in W$ s.t. $\mathsf{H}(S_P(w)) = \mathsf{H}(\Gamma_P(\eta))$ and $\mathsf{H}(S_Q(w)) = \mathsf{H}(\Gamma_Q(\eta))$
- Completeness: $\forall w \in W, \exists \eta \in \mathsf{PTree}(P, Q)$ s.t. $\mathsf{H}(S_P(w)) = \mathsf{H}(\Gamma_P(\eta))$ and $\mathsf{H}(S_Q(w)) = \mathsf{H}(\Gamma_Q(\eta))$

Applying Lemma 8, we obtain that

$$\begin{array}{c}\forall w \in W, \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\{\!\{P\}\!\}, \varnothing), w) \leqslant \mathrm{Prob}_{\mathcal{R}_{\mathsf{nr}}(\mathsf{N}^\ell)}((\{\!\{Q\}\!\}, \varnothing), w)\\ \text{if and only if}\\ \forall \eta \in \mathsf{PTree}(P, Q), \mathsf{compute}(\mathsf{H}(\Gamma_P(\eta))) \leqslant \mathsf{compute}(\mathsf{H}(\Gamma_Q(\eta)))\end{array}$$

which allows us to conclude the proof. □

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 $P, Q \in \mathcal{MP}^{\mathsf{pp}}$ and $\doteq$ be defined by a subterm convergent destructor rewrite system. Determinate may testing preorder $(\{\!\{P\}\!\}, \varnothing) \leqslant_{d\text{-}may} (\{\!\{Q\}\!\}, \varnothing)$ 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 [27].

As we will see in Section 7, we can use DEEPSEC to show that the dining cryptographers protocol does not provide anonymity when coins are *biased*.

7. An example: dining cryptographers

In this section, we illustrate our framework by proving anonymity of the well-known *dining cryptographers* protocol. This protocol has been designed to solve the following problem: three participants have a secret bit $b_i \in \{0, 1\}$, such that at most one of these bits equals 1, i.e. the sum s of the three bits is either 0 or 1. The goal is to determine whether $s = 1$ while hiding any additional information of each b_i; in particular, if $s = 1$, it must be impossible to identify for which i $b_i = 1$. To do so, we suppose that each pair of participants $\{a_i, a_j\}$ has a private coin, that they can toss privately, i.e., only a_i and a_j can read the result. To solve the above described problem, participants proceed as follows:

(1) each pair (a_i, a_j) of participants tosses its coin, and privately shares the result $c_{i,j}$;
(2) each participant a_i outputs their result r_i, computed as the exclusive or (xor) of their secret bit b_i, and the result of both their left and right coins;
(3) each participant can compute s by doing the xor of all the three r_i.

We suppose that the secret bits b_i of each participant are determined by an assignment $B : \{a_1, a_2, a_3\} \to \{0, 1\}$, and denote by DC_B the process (defined below) modeling the protocol with the given assignment. As we are interested in anonymity we focus on the case where one secret bit is 1 and we denote by $\mathcal{W}_1$ the set of *1-weighted* assignment functions, that is those functions that assign map exactly one participant to 1. Then, the *security property* for the dining cryptographers protocol can be stated as

$$\forall B, B' \in \mathcal{W}_1.\ DC_B \approx_{may} DC_{B'}$$

that is no adversary should be able to distinguish the systems DC_B and $DC_{B'}$ whenever one of the secret bits equals 1.

Our contributions in this section are the following. First, we model the dining cryptographers protocol as a process DC_B in the probabilistic applied π-calculus. Then we give a manual proof that (when coins are unbiased) DC_B and $DC_{B'}$–where B, B' are assignment functions as above–are may-testing equivalent. Finally, we show, using the tool DEEPSEC, that when the coins are biased those systems are not trace equivalent, thus not may-testing equivalent (Theorem 1).

7.1. Dining cryptographers in the probabilistic applied π-calculus

We split the definition of DC_B in several components. First, we model the fact that two adjacent participants can toss a coin, and privately share the result. We model this by the *oracle process* O_{c_l,c_r} (indexed by channels c_l, c_r): it performs a fair probabilistic choice, and sends the result in parallel on channel c_r on its right, and channel c_l on its left.

$$O_{c_l,c_r} := (\mathsf{out}(c_l, 0) \mid \mathsf{out}(c_r, 0)) +_{1/2} (\mathsf{out}(c_l, 1) \mid \mathsf{out}(c_r, 1))$$

We now define the process that models *one* agent $A_{c_l,c_r,c}(b)$: it is indexed by three channels–(private) channels c_l and c_r for communicating with the oracle placed to its left, respectively to its right, and a (public) channel c for sending its final result. It also depends on the agent's secret bit $b \in \{0, 1\}$.

$$A_{c_l,c_r,c}(b) := \mathsf{in}(c_r, x_r); \mathsf{in}(c_l, x_l); \mathsf{out}(c, x_l \oplus x_r \oplus b)$$

where $\oplus$ models bitwise xor.

Notation 7. We fix a set of public names $\mathcal{A} = \{a_1, a_2, a_3\} \subseteq \mathcal{N}_{pub}$ and we will identify each participant to their public channel a_i.

Moreover, we define the set of oracles (or *edges*), that we note $\mathcal{E} := \{\{a_i, a_j\} \mid i \neq j, a_i, a_j \in \mathcal{A}\}$. Observe that $\mathcal{E}$ has three elements, that we call e_1, e_2, e_3 for convenience.

We also fix a set of 6 private names $\mathcal{C} = \{c_1, ..., c_6\} \subseteq \mathcal{N}_{priv}$ that will serve as the private channels between oracles and participants; each of the three participants has access to two oracles. We formalize the association between channels, oracles and participants by three functions

$$c : \mathcal{A} \cup \mathcal{E} \to \mathcal{C} \times \mathcal{C} \qquad\qquad e : \mathcal{C} \to E \qquad\qquad a : \mathcal{C} \to A$$

Function c associates to an agent or an edge the pair of channels (we denote by $c.l$ and $c.r$ the first and second projection of c respectively). Functions e and a associate to a channel, the corresponding agent, respectively edge.

The overall system using these notations is depicted in Figure 9. Given $B \in \mathcal{W}_1$ we can now define the process DC_B that puts in parallel three participants, and their three coin toss oracles. The agent channels used by the participants to communicate their final results are public. By contrast, the channels used to communicate with the oracles are hidden from the adversary, thus private and bound by new .

$$DC_B := \mathsf{new}\ c_1, \ldots, c_6 \left(||_{e \in \mathcal{E}}\ O_{c_l(e),c_r(e)} \ \mid ||_{a \in \mathcal{A}}\ A_{c_l(a),c_r(a),a}(B(a)) \right)$$

We can now formally state that the dining cryptographers protocol does ensure anonymity.

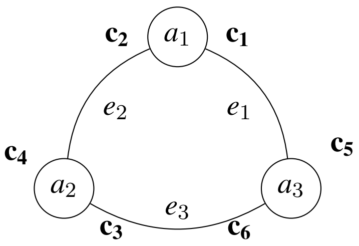


Figure 9. Dining cryptographers: graph representation

Theorem 5. $\forall B, B' \in \mathcal{W}_1. \quad DC_B \approx_{may} DC_{B'}$

In order to show Theorem 5, we are going to show use the labelled semantics and show a stronger result, i.e. that DC_B and $DC_{B'}$ are bisimilar.

7.2. Labelled semantics for the dining cryptographers protocol

The labelled semantics for our model of the dining cryptographers protocol is however complex: one of the reasons is that we do not enforce a particular order for the communications between the oracles O_i and the participants A_i. Thus, we have both probabilistic and non-deterministic concurrent behaviors. As a consequence, it is difficult to prove our equivalence result directly in the NPLTS N^ℓ. To solve this problem, we define an auxiliary *simplified* NPLTS N_{DC}, and show that:

(1) N_{DC} is bisimilar to N^ℓ;
(2) for every $B, B' \in \mathcal{W}_1$ the N_{DC}-states corresponding to DC_B and $DC_{B'}$ are bisimilar in N_{DC}.

The main idea for the construction of the auxiliary NPLTS N_{DC} is that only *relevant steps* should appear. More precisely, we note that a run of the protocol should be entirely characterized by the oracles' probabilistic choices, and the order in which participants output their results. By contrast, the order of internal communications between participants and oracles occur–on the private channels c_i–is irrelevant.

Definition 24. We define sets of *internal events* Λ and *external events* Λ^{ext} as :

$$\Lambda := \{e[\leftarrow b] \mid e \in \mathcal{E}, b \in \{0,1\}\} \qquad \Lambda^{ext} := \{a[\rightarrow b] \mid a \in \mathcal{A}, b \in \{0,1\}\}$$

The event $e[\leftarrow b]$ means that oracle $e \in \mathcal{E}$ is *sampling* $b \in \{0,1\}$. The event $a[\rightarrow b]$ means that participant $a \in A$ is *outputting* b on their public channel. We call *history* a pair $h = (s, l)$, where s is a set of internal events in Λ and l is a list of external events in Λ^{ext} such that no participant $a \in \mathcal{A}$ occurs twice in l. We denote by $\mathcal{H}$ the set of all such h.

Notation 8. To each history $h = (s, l)$, we associate a frame $\phi_h := \{\mathsf{ax}_1 = b_1, \dots \mathsf{ax}_n = b_n\}$ when $l = (a_1[\rightarrow b_1], \dots a_n[\rightarrow b_n])$. Moreover, for $e \in \mathcal{E}$, and history $h = (s, l)$, we write $e \in s$ when $e[\leftarrow b] \in s$ for some boolean b, and $e \notin h$ otherwise. We define similarly $a \in l$ and $a \notin l$ for $a \in \mathcal{A}$. We denote the empty history by $\epsilon_{\mathcal{H}} := (\varnothing, \epsilon)$.

We are now ready to define our simplified NPLTS N_{DC}.N_{DC} has exactly the same set of actions as N^ℓ, the NPLTS built from the labelled semantics. A state of N_{DC} encapsulates information about both the (relevant) history, and the assignment function of secret bits $\mathcal{A} \to \{0,1\}$. These information are enough to decide how a history is modified by an action. More precisely three kinds of actions can be enabled in a given state.

The first kind are actions of the form $(\xi \overset{?}{=} \zeta)$, $(\xi \overset{?}{\neq} \zeta)$ that in N^ℓ perform an equality test on the frame. In N_{DC} the equality test is performed on ϕ_h, the frame built unambiguously from the history as defined in Notation 8. The second kind of action is the internal action τ: in N_{DC} τ-actions correspond to a step where an oracle, that *does not already appear in the history*, randomly samples a boolean b. After such a τ-action, which is inherently probabilistic, the information that this oracle has sampled b is added to the history. The last kind of actions are the (external) output actions: some participant $a \in \mathcal{A}$ can output a result b–if no output by a is already recorded in the history–when the history contains the information of the boolean chosen by both oracles adjacent to a, and that moreover b is equal to the xor of these two booleans with the secret bit of a.

Definition 25. We define a NPLTS $\mathsf{N}_{DC} = (\mathcal{S}_{DC}, \mathcal{A}_{DC}, \mathsf{trans}_{DC})$ where $\mathcal{S}_{DC} := \{(h, B) \mid h \in \mathcal{H}, B \in \mathcal{W}_1\}$, $\mathcal{A}_{DC} := \{\tau\} \cup \mathcal{A}^\ell_{ext}$, and the transition function trans_{DC} is defined as:

$$((s,l),B) \xrightarrow{\tau} \frac{1}{2} \cdot \delta_{(s\cup(e[\leftarrow 0],l),B)} + \frac{1}{2} \cdot \delta_{(s\cup(e[\leftarrow 1],l),B)} \text{ when } e \notin s$$

$$((s,l),B) \xrightarrow{\mathsf{out}(a,b)} \delta_{(s,l::(a[\to b]),B)} \text{ when } a \notin l,\ \exists b_l, b_r \in \{0,1\}, e_l(a)[\leftarrow b_l] \in s \wedge e_r(a)[\leftarrow b_r] \in s$$
$$\wedge\, b = b_r \oplus b_l \oplus B(a)$$

$$(h,B) \xrightarrow{(\xi \overset{?}{=} \zeta)} \delta_{(h,B)} \text{ when } vars(\xi,\zeta) \subseteq dom(\phi_h)\ \wedge\ \xi\phi_h \doteq \zeta\phi_h$$

$$(h,B) \xrightarrow{(\xi \overset{?}{\neq} \zeta)} \delta_{(h,B)} \text{ when } vars(\xi,\zeta) \subseteq dom(\phi_h)\ \wedge\ \xi\phi_h \dot{\neq} \zeta\phi_h.$$

We now follow the road-map that we outlined in the beginning of the present section: first we show that we can replace the study of the labelled semantics for DC_B by the study of N_{DC}, by exhibiting a bisimilarity between those two NPLTSs (Proposition 7). Then, we show that the N_{DC} states (corresponding to) DC_B and $DC_{B'}$ are bisimilar as soon as $B, B' \in \mathcal{W}_1$ (Proposition 8).

Proposition 7. There exists an NPLTS bisimulation R on $\mathsf{N}_{DC} \cup \mathsf{N}^\ell$ such that for every $B \in \mathcal{W}_1$, it holds that $(\epsilon_{\mathcal{H}}, B)\, R\, (DC_B, \varnothing)$.

Proposition 8. If $B, B' \in \mathcal{W}_1$ then the N_{DC}-states $(\epsilon_{\mathcal{H}}, B)$ and $(\epsilon_{\mathcal{H}}, B')$ are bisimilar.

From Propositions 7 and 8, we can show that the process DC_B and $DC_{B'}$ are equivalent for the labelled bisimulation from Definition 16 in Section 4. We state this result formally below.

Corollary 3 (Security for the dining cryptographers protocol)**.** If$B, B' \in \mathcal{W}_1$ then $DC_B \approx_{bi} DC_{B'}$.

Proof. Recall that by definition of $\approx_{bi}$, we need to show that the states $s_B := (DC_B, \varnothing)$ and $s_{B'} := (DC'_B, \varnothing)$ are bisimilar in N^ℓ. Since we know by Proposition 8 that $t_B := (\epsilon_{\mathcal{H}}, B)$ and $t_{B'} = (\epsilon_{\mathcal{H}}, B')$ are bisimilar in N_{DC}, we also have $(t_B, t_{B'}) \in (\sim_{\mathsf{N}_{DC} \cup \mathsf{N}^\ell})$ (because N^ℓ and N_{DC} have disjoint state spaces). By Proposition 7, we also have that $t_B \sim_{\mathsf{N}_{DC} \cup \mathsf{N}^\ell} s_B$, and $t_{B'} \sim_{\mathsf{N}_{DC} \cup \mathsf{N}^\ell} s_{B'}$. Since $\sim_{\mathsf{N}_{DC} \cup \mathsf{N}^\ell}$ is an equivalence relation, we can combine all this to obtain that $s_B \sim_{\mathsf{N}_{DC} \cup \mathsf{N}^\ell} s_{B'}$. From here (again because N^ℓ and N_{DC} have disjoint state spaces), we conclude that $s_B \sim_{\mathsf{N}^\ell}, s_{B'}$. □

As an immediate consequence of Corollary 3, and since labelled bisimulation is stronger than may-testing equivalence, we obtain a proof of Theorem 5, i.e. that $\forall B, B' \in \mathcal{W}_1$, $DC_B \approx_{may} DC_{B'}$.

7.3. *Analysis with* DEEPSEC

We have analyzed the Dining Cryptographers protocol using our probabilistic extension of the DEEPSEC tool [27]. The encoding of processes DC_B for $B \in \mathcal{W}_1$ is straightforward and follows directly the modeling above. Although DEEPSEC does not support xor in general, we can easily encode the ternary version using 8 rewrite rules $xor(b_1, b_2, b_3) \to b$ for $b_1, b_2, b_3 \in \{0, 1\}$ and $b = b1 \oplus b_2 \oplus b_3$. Restricting to a ternary xor is not a loss of generality as the protocol does not take any input from the adversary and only honest users construct xor-terms.

As a first result we can show that

$$\forall B, B' \in \mathcal{W}_1. \quad DC_B \approx_{tr} DC_{B'}$$

This result is actually not satisfactory, as trace equivalence is strictly weaker than may testing (which also motivated the pen and paper proof of the previous section). More interesting is the fact that we can show that anonymity is broken when we use a *biased* coin, i.e., we replace the probability 0.5 in the coin tossing oracle by a probability $p \neq 0.5$. If we denote the modified process by DC_B^p we can for instance show, using DEEPSEC, that, as expected,

$$DC_{B_1}^{0.4} \not\approx_{tr} DC_{B_2}^{0.4}$$

where B_i assigns b_i to 1 and remaining bits to 0. Given that trace equivalence is strictly weaker than may-testing this implies that may testing does not hold either.

Given the rather small size of the processes DEEPSEC performs these verifications in about 1 second each.

8. 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 [20], 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 [25], 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* [22] 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 [20, 21]) 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.

Acknowledgments. We thank Alwen Tiu and the anonymous reviewers for their helpful comments and suggestions. 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).

References

[1] D. Dolev and A.C. Yao, On the security of public key protocols, *IEEE Trans. Inf. Theory* **29**(2) (1983), 198–207. doi:10.1109/TIT.1983.1056650.

[2] K. Bhargavan, B. Blanchet and N. Kobeissi, Verified models and reference implementations for the TLS 1.3 standard candidate, in: *IEEE Symposium on Security and Privacy (S&P'17)*, IEEE, 2017, pp. 483–502.

[3] C. Cremers, M. Horvat, J. Hoyland, S. Scott and T. van der Merwe, A Comprehensive Symbolic Analysis of TLS 1.3, in: *ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, ACM, 2017, pp. 1773–1788.

[4] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, A formal security analysis of the signal messaging protocol, *Journal of Cryptology* **33**(4) (2020), 1914–1983.

[5] D.A. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse and V. Stettler, A Formal Analysis of 5G Authentication, in: *ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*, D. Lie, M. Mannan, M. Backes and X. Wang, eds, ACM, 2018, pp. 1383–1396. doi:10.1145/3243734.3243846.

[6] D.A. Basin, R. Sasse and J. Toro-Pozo, The EMV Standard: Break, Fix, Verify, in: *42nd IEEE Symposium on Security and Privacy (S&P'21)*, IEEE, 2021, pp. 1766–1781. doi:10.1109/SP40001.2021.00037.

[7] V. Cortier, S. Kremer and B. Warinschi, A Survey of Symbolic Methods in Computational Analysis of Cryptographic Systems, *Journal of Automated Reasoning* **46**(3–4) (2010), 225–259. doi:10.1007/s10817-010-9187-9.

[8] D. Chaum, The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability, *J. Cryptol.* **1**(1) (1988), 65–75. doi:10.1007/BF00206326.

[9] D. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Commun. ACM* **24**(2) (1981), 84–88. doi:10.1145/358549.358563.

[10] M.K. Reiter and A.D. Rubin, Crowds: Anonymity for Web Transactions, *ACM Trans. Inf. Syst. Secur.* **1**(1) (1998), 66–92. doi:10.1145/290163.290168.

[11] M. Abadi, B. Blanchet and C. Fournet, The applied pi calculus: Mobile values, new names, and secure communication, *Journal of the ACM (JACM)* (2017).

[12] J. Goubault-Larrecq, C. Palamidessi and A. Troina, A probabilistic applied pi–calculus, in: *Asian Symposium on Programming Languages and Systems*, Springer, 2007, pp. 175–190.

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

[14] A. Parma and R. Segala, Logical characterizations of bisimulations for discrete probabilistic systems, in: *International Conference on Foundations of Software Science and Computational Structures*, Springer, 2007, pp. 287–301.

[15] V. Castiglioni, Trace and Testing Metrics on Nondeterministic Probabilistic Processes, in: *Combined 25th International Workshop on Expressiveness in Concurrency and 15th Workshop on Structural Operational Semantics and 15th Workshop on Structural Operational Semantics,(EXPRESS/SOS) 2018*, Vol. 276, 2018, pp. 19–36.

[16] F. Bonchi, A. Sokolova and V. Vignudelli, The theory of traces for systems with nondeterminism and probability, in: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, 2019, pp. 1–14.

[17] Y. Deng, R. Van Glabbeek, M. Hennessy and C. Morgan, Testing finitary probabilistic processes, in: *International Conference on Concurrency Theory*, Springer, 2009, pp. 274–288.

[18] M. Stoelinga, Alea jacta est: verification of probabilistic, real-time and parametric systems, PhD thesis, University of Nijmegen, the Netherlands, 2002.

[19] C.G. Eisentraut, Principles of Markov automata, PhD thesis, Saarländische Universitäts-und Landesbibliothek, 2017.

[20] K. Chatzikokolakis and C. Palamidessi, Making random choices invisible to the scheduler, *Inf. Comput.* **208**(6) (2010), 694–715. doi:10.1016/j.ic.2009.06.006.

[21] M.S. Alvim, M.E. Andrés, C. Palamidessi and P. van Rossum, Safe Equivalences for Security Properties, in: *Theoretical Computer Science - 6th IFIP TC 1/WG 2.2 International Conference, TCS 2010, Held as Part of WCC 2010*, C.S. Calude and V. Sassone, eds, IFIP Advances in Information and Communication Technology, Vol. 323, Springer, 2010, pp. 55–70. doi:10.1007/978-3-642-15240-5_5.

[22] V. Cheval, S. Kremer and I. Rakotonirina, DEEPSEC: Deciding Equivalence Properties in Security Protocols - Theory and Practice, in: *Proceedings of the 39th IEEE Symposium on Security and Privacy (S&P'18)*, IEEE Computer Society Press, San Francisco, CA, USA, 2018, pp. 525–542. doi:10.1109/SP.2018.00033.

[23] V. Cheval, V. Cortier and S. Delaune, Deciding equivalence-based properties using constraint solving, *Theor. Comput. Sci.* **492** (2013), 1–39. doi:10.1016/j.tcs.2013.04.016.

[24] H. Comon-Lundh and V. Cortier, Computational soundness of observational equivalence, in: *ACM Conference on Computer and Communications Security (CCS'08)*, P. Ning, P.F. Syverson and S. Jha, eds, ACM, 2008, pp. 109–118. doi:10.1145/1455770.1455786.

[25] R. Chadha, A.P. Sistla and M. Viswanathan, Verification of randomized security protocols, in: *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*, IEEE, 2017, pp. 1–12.

[26] M.S. Bauer, R. Chadha, A.P. Sistla and M. Viswanathan, Model checking indistinguishability of randomized security protocols, in: *International Conference on Computer Aided Verification (CAV'18)*, Springer, 2018, pp. 117–135.

[27] The DEEPSEC tool, 2023, https://github.com/DeepSec-prover/deepsec/tree/probability.

[28] V. Cheval, R. Crubillé and S. Kremer, Symbolic protocol verification with dice: process equivalences in the presence of probabilities (extended version), 2023, https://chevalvi.gitlabpages.inria.fr/chevalvi/files/CCK-dice-long.pdf.

[29] B. Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, *Foundations and Trends in Privacy and Security* (2016).

[30] R. Milner, J. Parrow and D. Walker, A Calculus of Mobile Processes, I, *Inf. Comput.* **100**(1) (1992), 1–40. doi:10.1016/0890-5401(92)90008-4.

[31] Y. Deng, Axiomatisations and types for probabilistic and mobile processes, PhD thesis, École des Mines de Paris, 2005.

[32] M. Hennessy and R. Milner, On observing nondeterminism and concurrency, in: *Automata, Languages and Programming (ICALP'80)*, J. de Bakker and J. van Leeuwen, eds, Springer, 1980, pp. 299–309.

[33] R. Amadio, *Operational methods in semantics*, 2016.

[34] R.J. van Glabbeek, Bounded nondeterminism and the approximation induction principle in process algebra, in: *Annual Symposium on Theoretical Aspects of Computer Science*, Springer, 1987, pp. 336–347.

[35] D. Chaum, P.Y.A. Ryan and S.A. Schneider, A Practical Voter-Verifiable Election Scheme, in: *10th European Symposium on Research in Computer Security (ESORICS'05)*, S.D.C. di Vimercati, P.F. Syverson and D. Gollmann, eds, Lecture Notes in Computer Science, Vol. 3679, Springer, 2005, pp. 118–139. doi:10.1007/11555827_8.

[36] V. Cortier and S. Delaune, A Method for Proving Observational Equivalence, in: *22nd IEEE Computer Security Foundations Symposium, (CSF'09)*, IEEE Computer Society, 2009, pp. 266–276. doi:10.1109/CSF.2009.9.