Symbolic Bisimulations for Probabilistic Systems

Peng WU
CNRS and LIX, École Polytechnique
91128 Palaiseau, France

Catuscia Palamidessi
INRIA Futurs and LIX, École Polytechnique
91128 Palaiseau, France

Huimin Lin
Lab of Computer Science, Institute of Software, Chinese Academy of Sciences
100080 Beijing, China

Abstract

The paper introduces symbolic bisimulations for a simple probabilistic π-calculus to overcome the infinite branching problem that still exists in checking ground bisimulations between probabilistic systems. Especially the definition of weak (symbolic) bisimulation does not rely on the random capability of adversaries and suggests a solution to the open problem on the axiomatization for weak bisimulation in the case of unguarded recursion. Furthermore, we present an efficient characterization of symbolic bisimulations for the calculus, which allows the "on-the-fly" instantiation of bound names and dynamic construction of equivalence relations for quantitative evaluation. This directly results in a local decision algorithm that can explore just a minimal portion of the state spaces of the probabilistic processes in question.

1 Introduction

Probability has been introduced into process calculi for the specification and verification of systems that present uncertainty aspects amenable to quantitative treatment. Bisimulations were defined correspondingly [23, 22, 11] and have been applied in the context of security analysis. To name a few, [1, 5] employed probabilistic bisimulation notions to specify the information flow security properties; [15] formalized the non-repudiation property through a weak bisimulation between two probabilistic models, one with a well-behaved recipient and the other with a malicious recipient.

In the non-probabilistic case, symbolic bisimulations [16, 4] were proposed to overcome the "infinite branching" problem, namely the definition of ground bisimulation requires infinite many residuals to be further checked after matching input actions, each residual representing one possible instantiation of input names. The same problem exists for probabilistic systems but has got less attention so far.

This paper introduces symbolic bisimulations for a *simple probabilistic π-calculus*, abbreviated as π_{sp}, which is a newly defined extension of π-calculus with a probabilistic blind choice. π_{sp} stems from [14, 3, 7] and supports the co-existence of nondeterministic and probabilistic behaviors in two ways: by allowing probabilistic distribution over nondeterminism and by allowing nondeterminism among probabilistic distributions. Besides, [11] showed that weak bisimulation cannot be defined on a probabilistic *mixed-guarded* choice without using combined transitions, and also left as an open problem how to define an axiomatizable relation of weak bisimulation. This paper presents a revised definition of weak bisimulation that does not rely on probabilistic combinations of transitions and paves a way for axiomatization. We also present probabilistic versions of symbolic bisimulations for completeness.

As in the non-probabilistic case, a symbolic bisimulation for π_{sp} is a family of *equivalence* relations over processes, indexed by equality and(or) inequality conditions over finite sets of names. As far as an input action is concerned, the symbolic bisimulation requires an input name to be instantiated only with fresh names for each element of a finite partition of current indexing condition.

In total four symbolic bisimulations are presented, including strong/weak symbolic bisimulations and strong/weak probabilistic symbolic bisimulations. Moreover, we propose a practical way to characterize symbolic bisimulations, based on which a *local* decision algorithm (framework) is proposed in the sense that only a minimal portion of the state spaces of π_{sp} processes in question will be explored for checking the bisimilarity between them. The characterization relaxes the underlying equivalence relations in the definitions of symbolic bisimulations to be symmetric relations, which make a local algorithm possible. Furthermore, the characterization avoids constructing the finite partitions of indexing conditions, which is expensive, by considering an ordered name space. Our work is adapted from [16, 17, 18, 19] and improves the efficiency of the algorithm in [18] by tabling the intermediate dependencies between pairs of processes. As far as we know, this is the first efficient local algorithm for checking bisimulations for probabilistic systems.

The rest of this paper is organized as follows. Section 2 introduces the syntax of π_{sp} with its concrete and symbolic operational semantics. Section 3 presents the strong and weak symbolic bisimulations and shows the relationships between ground and symbolic bisimulations. The probabilistic versions of ground and symbolic bisimulations for π_{sp} are presented in Section 4. The practical characterization of these bisimulations is proposed in Section 5, followed by the "on-the-fly" decision algorithm in Section 6. Section 7 concludes the paper with some future work outlined.

2 A Simple Probabilistic π-Calculus

We presuppose a countably infinite set $\mathcal{N}$ of names, ranged over by x, y, z. Let $\tilde{x}, \tilde{y}, \tilde{z}$ range over name vectors. A substitution $\sigma \equiv \{\tilde{y}/\tilde{x}\}$ is a mapping from a vector of distinct names $\tilde{x}$ to a name vector $\tilde{y}$, where $\tilde{x}$ and $\tilde{y}$ are of same length.

2.1 Syntax

Let t, u range over processes, A over process identifiers and p, q over real numbers in $(0, 1]$. The syntax of π_{sp} is given by the following BNF grammar:

$$\begin{array}{rcl}\alpha & ::= & \tau \mid x(y) \mid \bar{x}y \\ t & ::= & \alpha.t \mid \sum_{i\in I} t_i \mid \sum_{i\in I} p_i\tau.t_i \mid t \mid t \mid \nu xt \\ & & \mid [x=y]t \mid A(\tilde{y})\end{array}$$

where I is an index set.

The prefixed process $\alpha.t$ can evolve into t with probability 1, after performing action α. There are three types of basic actions: an input action $x(y)$, an output action $\bar{x}y$ and the silent action τ.

Two types of summations are involved: $\sum_{i\in I} t_i$ and $\sum_{i\in I} p_i\tau.t_i$, which indicate a nondeterministic and a probabilistic blind (or internal) choice among t_i's, respectively. Herein, the probabilistic branches are prefixed explicitly with silent actions, each associated with a positive probability p_i such that $\sum_{i\in I} p_i = 1$. We will use 0 (*inactive* process) to stand for the empty summation, $t + u$ for a binary nondeterministic choice and $p\tau.t + (1-p)\tau.u$ for a binary probabilistic blind choice.

Such a *simple* probabilistic choice does not lose the expressiveness of more general probabilistic choices [12, 24]. The notion of a reactive probabilistic choice $\alpha. \sum_{i\in I} p_i t_i$ can be represented in π_{sp} as $\alpha. \sum_{i\in I} p_i\tau.t_i$, while a generative probabilistic choice $\sum_{i\in I} p_i\alpha_i t_i$ can be represented as $\sum_{i\in I} p_i\tau.\alpha_i.t_i$.

In composition $t \mid u$, the component t and u can proceed in parallel and can also interact via shared names. It can be seen that in π_{sp}, probabilistic choices have no extra effect on parallel composition [24], because only probabilistic internal choices are involved.

The restriction νxt, match construction $[x = y]t$ and identifier A take meanings from π-calculus.

An occurrence of name y in process t is *bound* if it is in a subexpression of t of the form $x(y).u$ or νyu; otherwise, it is *free*. The sets of free and bound names of process t are denoted by $fn(t)$ and $bn(t)$, respectively.

2.2 Operational Semantics

We specify the concrete operational semantics of π_{sp} as a probabilistic automaton [23], and its symbolic operational semantics based on a probabilistic extension of symbolic transition graph [16, 4]. The notion of a probability space is used to interpret the probabilistic aspect of probabilistic systems. For notational convenience, index sets I will be omitted, if no distinction is necessary. Especially, the notation $\{\cdot \mid i \in I\}$ will be simplified as $\{\cdot\}_i$.

2.2.1 Probability Spaces

A probability space is a triplet $\mathcal{P} = (\Omega, F, \eta)$ where Ω is a set, F is a collection of subsets of Ω that includes Ω and is closed under complement and countable union, and $\eta : F \rightarrow [0, 1]$ is a probability distribution function such that $\eta(\Omega) = 1$ and for any collection $\{C_i\}_i$ of at most countably many pairwise disjoint elements of F, $\eta(\bigcup_i C_i) = \sum_i \eta(C_i)$.

A probability space (Ω, F, η) is *discrete* if $F = 2^{\Omega}$, and hence abbreviated as (Ω, η). Let $\mathbb{PROB}(X)$ denote the set of all discrete probability spaces (Ω, η) on a set X such that $\Omega \subseteq X$. The *Dirac* distribution over the singleton set $\{x\}$ constitutes the probability space with the unique element x, denoted by $\mathcal{D}(x)$.

Suppose $\mathcal{P}_i = (\Omega_i, \eta_i)$, $i \in \{1, 2\}$ and $Z \subseteq \Omega_1$, $\mathcal{P}_1 +_Z \mathcal{P}_2$ denotes the probability space (Ω, η) such that $\Omega = (\Omega_1 - Z) \cup \Omega_2$ and for each set $Y \subseteq \Omega$,

- if $Y \subseteq \Omega_1 - \Omega_2$, $\eta(Y) = \eta_1(Y)$;
- if $Y \subseteq \Omega_2 - \Omega_1$, $\eta(Y) = \eta_1(Z)\eta_2(Y)$;
- otherwise, $\eta(Y) = \eta_1(Y \cap \Omega_1) + \eta_1(Z)\eta_2(Y \cap \Omega_2)$.

Intuitively, $\mathcal{P}_1 +_Z \mathcal{P}_2$ means to replace the distribution of $\mathcal{P}_1$ over Z by the one of $\mathcal{P}_2$. This operation will be referred to later in Section 3 to define weak transition relations.

Given probability spaces $\{\mathcal{P}_i = (\Omega_i, \eta_i)\}_i$ on X and weights $w_i > 0$ for each i such that $\sum_i w_i = 1$, we define a *convex combination* $\sum_i w_i \mathcal{P}_i$ as the probability space (Ω, η) such that $\Omega = \bigcup_i \Omega_i$ and for each set $Y \subseteq \Omega$, $\eta(Y) = \sum_{Y \cap \Omega_i \neq \emptyset} w_i \eta_i(Y \cap \Omega_i)$

2.2.2 Semantics

Let $\{p_i : t_i\}_i$ denote a probability space $\mathcal{P} = (\{t_i\}_i, \eta)$ such that $\eta(\{t_i\}) = p_i$ for each i. The sets of free and bound names of $\mathcal{P}$ are defined as follows: $fn(\mathcal{P}) = \bigcup_i fn(t_i)$, $bn(\mathcal{P}) = \bigcup_i bn(t_i)$. Especially, $fn(\alpha.\mathcal{P}) = \bigcup_i fn(\alpha.t_i)$. The application of a substitution σ to $\mathcal{P}$, written $\mathcal{P}\sigma$, results in $\{p_i : t_i\sigma\}_i$.

The operational semantics of π_{sp} can be given as a probabilistic automaton (S, Act, T, s_0), where S is a nonempty set of processes (states), Act is a set of actions, $T \subseteq S \times Act \times \mathbb{PROB}(S)$ is a set of probabilistic transitions and $s_0 \in S$ is the initial state. The transition relation T is defined by the rules in Figure 1, where each probabilistic transition takes the form $t \xrightarrow{\alpha} \{p_i : t_i\}_i$. The symmetric rule for PAR is omitted. A *Dirac* probabilistic transition $t \xrightarrow{\alpha} \mathcal{D}(u)$ is abbreviated as $t \xrightarrow{\alpha} u$.

It can be seen that only a transition $t \xrightarrow{\tau} \mathcal{P}$ may introduce a non-*Dirac* probabilistic distribution, which is the main difference from the transitions in π-calculus. Such a lightweight extension inherits most semantical properties from π-calculus, in which, instead, the input and bound output actions are paid highly attention to.

In addition to the actions from the syntax, a bound output action $\bar{x}(y)$ is also introduced in the probabilistic automaton to model *scope extrusion*. The sets of free and bound names of actions are defined as follows: $fn(x(y)) = fn(\bar{x}(y)) = \{x\}$, $fn(\bar{x}y) = \{x, y\}$, $bn(x(y)) = bn(\bar{x}(y)) = \{y\}$ and $fn(\tau) = bn(\tau) = bn(\bar{x}y) = \emptyset$. Let $n(\alpha)$ indicate all the names that occur in action α, that is, $n(\alpha) = fn(\alpha) \cup bn(\alpha)$.

2.2.3 Symbolic Semantics

Symbolic transition graphs were proposed to define the symbolic semantics of value-passing CCS [13]and π-calculus [16, 17, 18, 19]. We recall some notations and properties of conditions below.

Let B, D, M, N denote conditions in forms of $true$, $[x = y]$, $\neg M$ or $M \wedge N$. Especially,‘ $\neg[x = y]$ will be abbreviated as $x \neq y$. A condition M is *consistent* if there is no $x, y \in \mathcal{N}$ such that $M \Rightarrow x = y$ and $M \Rightarrow x \neq y$. M is *maximally consistent* on $V \subset \mathcal{N}$, written $M \in MC_V$, if M is consistent and for any $x, y \in V$, either $M \Rightarrow x = y$ or $M \Rightarrow x \neq y$. N is a *maximally consistent extension* of M on V, written $N \in MCE_V(M)$, if $N \Rightarrow M$, $n(N) - n(M) \subseteq V$ and N is maximally consistent on V. $\sigma \models N$ indicates $N\sigma = true$.

$$\text{PRE} \quad \frac{}{\alpha.t \xrightarrow{\alpha} t}$$

$$\text{PROB} \quad \frac{}{(\sum_i p_i \tau.t_i) \xrightarrow{\tau} \{p_i : t_i\}_i}$$

$$\text{SUM} \quad \frac{t_j \xrightarrow{\alpha} \mathcal{P}}{(\sum_{i \in I} t_i) \xrightarrow{\alpha} \mathcal{P}} \quad j \in I$$

$$\text{PAR} \quad \frac{t \xrightarrow{\alpha} \{p_i : t_i\}_i}{t \mid u \xrightarrow{\alpha} \{p_i : (t_i \mid u)\}_i} \quad bn(\alpha) \cap fn(u) = \emptyset$$

$$\text{RES} \quad \frac{t \xrightarrow{\alpha} \{p_i : t_i\}_i}{\nu x t \xrightarrow{\alpha} \{p_i : \nu x t_i\}_i} \quad x \notin n(\alpha)$$

$$\text{COM} \quad \frac{t \xrightarrow{x(z)} t' \quad u \xrightarrow{\bar{x}y} u'}{t \mid u \xrightarrow{\tau} t'\{y/z\} \mid u'}$$

$$\text{OPEN} \quad \frac{t \xrightarrow{\bar{y}x} u}{\nu x t \xrightarrow{\bar{y}(x)} u} \quad x \neq y$$

$$\text{CLOSE} \quad \frac{t \xrightarrow{x(z)} t' \quad u \xrightarrow{\bar{x}(y)} u'}{t \mid u \xrightarrow{\tau} \nu y(t'\{y/z\} \mid u')}$$

$$\text{MATCH} \quad \frac{t \xrightarrow{\alpha} \mathcal{P}}{[x = x]t \xrightarrow{\alpha} \mathcal{P}} \quad x \notin bn(\alpha)$$

$$\text{IDE} \quad \frac{t\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha} \mathcal{P}}{A(\tilde{y}) \xrightarrow{\alpha} \mathcal{P}} \quad A(\tilde{x}) \triangleq t$$

Figure 1. Concrete Semantics of π_{sp}

The restriction operator νx is defined on conditions as follows (where $x \neq y \neq z$).

$$\begin{array}{rclrcl} \nu x\, true & = & true & \nu x[x = x] & = & true \\ \nu x[x = y] & = & false & \nu x[y = z] & = & [y = z] \\ & & \nu x(M \wedge N) \ = \ \nu x M \wedge \nu x N & & & \end{array}$$

We quote Lemma 2.3 from [19] below.

Lemma 1. Suppose a condition $N \in MC_V$ and M such that $n(M) \subseteq V$. If $\sigma \models N$ and $\sigma \models M$ for some σ, then $N \Rightarrow M$.

The symbolic operational semantics of π_{sp} is given as a *probabilistic symbolic transition graph* (PSTG), which we define as a probabilistic extension of symbolic transition graphs. A PSTG is a rooted directed graph $(S, Act, \mathcal{T}, s_0)$, where $\mathcal{T}$ is a set of probabilistic symbolic transitions. The transition relation $\mathcal{T}$ is defined by the rules in Figure 2. The symmetric rule for PAR is omitted.

In Figure 2, each probabilistic symbolic transition takes the form $t \xrightarrow{M,\alpha} \{p_i : t_i\}_i$, which means if condition M holds, process t can perform action α and evolve into process t_i with probability p_i. Similarly, $t \xrightarrow{M,\alpha} \mathcal{D}(u)$ is abbreviated as $t \xrightarrow{M,\alpha} u$. The condition $true$ is omitted without causing any confusion. Herein, only the transition $t \xrightarrow{M,\tau} \mathcal{P}$ may introduce a non-*Dirac* probabilistic symbolic distribution.

$$\textsc{Pre}\qquad \frac{}{\alpha.t \xrightarrow{\alpha} t}$$

$$\textsc{Prob}\qquad \frac{}{(\sum_i p_i\tau.t_i) \xrightarrow{\tau} \{p_i : t_i\}_i}$$

$$\textsc{Sum}\qquad \frac{t_j \xrightarrow{M,\alpha} \mathcal{P}}{(\sum_{i\in I} t_i) \xrightarrow{M,\alpha} \mathcal{P}}\quad j \in I$$

$$\textsc{Par}\qquad \frac{t \xrightarrow{M,\alpha} \{p_i : t_i\}_i}{t \mid u \xrightarrow{M,\alpha} \{p_i : (t_i \mid u)\}_i}\quad bn(\alpha) \cap fn(u) = \emptyset$$

$$\textsc{Res}\qquad \frac{t \xrightarrow{M,\alpha} \{p_i : t_i\}_i}{\nu xt \xrightarrow{\nu xM,\alpha} \{p_i : \nu xt_i\}_i}\quad x \notin n(\alpha)$$

$$\textsc{Com}\qquad \frac{t \xrightarrow{M,y(z)} t' \quad u \xrightarrow{N,\bar{x}v} u'}{t \mid u \xrightarrow{[x=y]\wedge M\wedge N,\tau} t'\{v/z\} \mid u'}$$

$$\textsc{Open}\qquad \frac{t \xrightarrow{M,\bar{y}x} u}{\nu xt \xrightarrow{\nu xM,\bar{y}(x)} u}\quad x \neq y$$

$$\textsc{Close}\qquad \frac{t \xrightarrow{M,y(z)} t' \quad u \xrightarrow{N,\bar{x}(v)} u'}{t \mid u \xrightarrow{[x=y]\wedge M\wedge N,\tau} \nu v(t'\{v/z\} \mid u')}$$

$$\textsc{Match}\qquad \frac{t \xrightarrow{M,\alpha} \mathcal{P}}{[x = y]t \xrightarrow{[x=y]\wedge M,\alpha} \mathcal{P}}\quad \{x, y\} \cap bn(\alpha) = \emptyset$$

$$\textsc{Ide}\qquad \frac{t\{\tilde{y}/\tilde{x}\} \xrightarrow{M,\alpha} \mathcal{P}}{A(\tilde{y}) \xrightarrow{M,\alpha} \mathcal{P}}\quad A(\tilde{x}) \triangleq t$$

Figure 2. Symbolic Semantics of π_{sp}

Lemma 2. If $t \xrightarrow{M,\alpha} \mathcal{P}$, $n(M) \cup fn(\alpha) \subseteq fn(t)$.

Proof. By transition induction. □

The following lemma relates the symbolic and concrete operational semantics of π_{sp}.

Lemma 3. 1. If $t \xrightarrow{M,\alpha} \mathcal{P}$ then for any $\sigma \models M$ with $bn(\alpha) \cap (fn(t) \cup n(\sigma)) = \emptyset$, $t\sigma \xrightarrow{\alpha\sigma} \mathcal{P}\sigma$.

2. If $t\sigma \xrightarrow{\alpha'} \mathcal{P}'$ then there are M, α, and $\mathcal{P}$ such that $\sigma \models M$, $\alpha' \equiv \alpha\sigma$, $\mathcal{P}' \equiv \mathcal{P}\sigma$ and $t \xrightarrow{M,\alpha} \mathcal{P}$.

Proof. By transition induction. Details are similar to Lemma 2.6 in [19]. □

3 Symbolic Bisimulations

This paper focuses on late bisimulations, while the results can be carried over to early bisimulations in a systematic manner. As usual, we need to lift an equivalence relation on S to a relation on $\mathbb{PROB}(S)$ for defining bisimulations for probabilistic systems.

Definition 1. Let $\mathcal{R}$ be an equivalence relation over S. Two discrete probability spaces $\mathcal{P}_1 = (S, \eta_1)$ and $\mathcal{P}_2 = (S, \eta_2)$ are $\mathcal{R}$-equivalent, written $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$, if for each equivalence class C of $S/\mathcal{R}$, $\eta_1(C) = \eta_2(C)$. Especially, $\mathcal{D}(t) \equiv_{\mathcal{R}} \mathcal{D}(u)$ will be abbreviated as $t \equiv_{\mathcal{R}} u$.

3.1 Strong Symbolic Bisimulation

We formalize the notion of strong bisimulation for π_{sp}, following the lines of [23], in order to show the relationship between symbolic bisimulations and ground ones in the probabilistic settings.

Definition 2 (Strong Bisimulation)**.** An equivalence relation $\mathcal{R}$ is a *late bisimulation* if $(t, u) \in \mathcal{R}$ implies

1. whenever $t \xrightarrow{x(y)} t'$ with $x \notin fn(t, u)$ then $u \xrightarrow{x(y')} u'$ for some u' such that for any z, $t'\{z/y\} \equiv_{\mathcal{R}} u'\{z/y'\}$;

2. whenever $t \xrightarrow{\alpha} \mathcal{P}$ for any other action α with $bn(\alpha) \cap fn(t, u) = \emptyset$ then $u \xrightarrow{\alpha} \mathcal{Q}$ for some $\mathcal{Q}$ such that $\mathcal{P} \equiv_{\mathcal{R}} \mathcal{Q}$.

Write $t \sim_l u$ if there exists a late bisimulation $\mathcal{R}$ such that $(t, u) \in \mathcal{R}$.

Let $\alpha =^B \beta$ mean that actions α and β are identical under condition B, as defined in [17, 19]. Now we are set to introduce strong symbolic bisimulation for π_{sp}.

Definition 3 (Strong Symbolic Bisimulation)**.** A condition indexed family of equivalence relations between processes, $\mathcal{S} = \{S^B\}$, is a *late symbolic bisimulation* if $(t, u) \in S^B$ implies whenever $t \xrightarrow{M,\alpha} \mathcal{P}$ with $bn(\alpha) \cap fn(t, u, B) = \emptyset$, for each $D \in MCE_{fn(t,u)}(B \cup M)$ there is a $u \xrightarrow{N,\beta} \mathcal{Q}$ such that $D \Rightarrow N$, $\alpha =^D \beta$ and $\mathcal{P} \equiv_{S^{D'}} \mathcal{Q}$ where

- if $\alpha = \bar{x}(y)$, $D' = D \cup \{y \neq z \mid z \in fn(\alpha.\mathcal{P}, \beta.\mathcal{Q})\}$.

- otherwise, $D' = D$.

Write $t \sim_L^B u$ if $(t, u) \in S^B \in \mathcal{S}$ for some late symbolic bisimulation $\mathcal{S}$.

By the similar proof for bisimulations defined in [12], $\sim_l$ and $\sim_L$ can be shown to be equivalence relations. We relate $\sim_L$ with $\sim_l$ through the following propositions.

Proposition 1. Suppose $B \in MC_{fn(t,u)}$, $t \sim_L^B u$ iff $t\sigma \sim_l u\sigma\ for\ any\ \sigma \vDash B$.

Proof. We outline the proof below and more details are given in Appendix A.

($\Rightarrow$) Define $\mathcal{R} = \{(t\sigma, u\sigma) \mid t \sim_L^B u \text{ for some } B \in MC_{fn(t,u)}, \text{ and } \sigma \vDash B\}$. It can be shown that $\mathcal{R}$ is a late bisimulation.

($\Leftarrow$) Define $S^B = \{(t, u) \mid B \in MC_{fn(t,u)}, t\sigma \sim_l u\sigma \text{ for some } \sigma \vDash B\}$. Let $\mathcal{S} = \{S^B\}$. It can be shown that $\mathcal{S}$ is a late symbolic bisimulation. □

Then Proposition 2 relaxes the requirement on indexing conditions in Proposition 1.

Proposition 2. $t \sim_L^B u$ iff $t \sim_L^D u$ for any $D \in MCE_{fn(t,u)}(B)$.

Proof. Similar to Proposition 2.14 in [19]. □

With the above two propositions, we can show that $\sim_L$ captures $\sim_l$.

Theorem 1. $t \sim_L^B u$ iff $t\sigma \sim_l u\sigma$ for any $\sigma \vDash B$.

Proof. By Lemma 1, Proposition 1 and 2. □

3.2 Weak Symbolic Bisimulation

3.2.1 Weak Bisimulation

Weak bisimulations have been proposed for various probabilistic systems. However, [11] showed that the "natural" weak bisimulation cannot be defined upon probabilistic guarded choices (i.e. $\sum_i p_i\alpha_i.t_i$), even for finite-state probabilistic processes, because it turns out to be non-transitive. Meanwhile, [11] proposed an alternative definition that relies on the random capability of adversaries (explained later in Section 4). Furthermore, [11] left as an open problem how to define weak bisimulation that allows axiomatization with finitary inference rules. Herein, π_{sp} uses a simpler notation of probabilistic choice (i.e. $\sum_i p_i\tau.t_i$), which allows direct definitions of weak (symbolic) bisimulations in a usual way.

Definition 4. The *late weak probabilistic transitions* $t \overset{\alpha}{\Rightarrow}_l \mathcal{P}$ are defined as the least relation satisfying the rules shown in Figure 3(a). Furthermore, we extend $\Rightarrow_l$ onto discrete probability spaces with the rule shown in Figure 3(b). Let $\hat{\tau} = \epsilon$ and $\hat{\alpha}$ be α for $\alpha \neq \tau$. Especially, let $\Rightarrow_l$ be $\overset{\epsilon}{\Rightarrow}_l$.

$$1)\ \frac{}{t \overset{\epsilon}{\Rightarrow}_l t} \qquad 2)\ \frac{t \xrightarrow{\alpha} \mathcal{P}}{t \overset{\alpha}{\Rightarrow}_l \mathcal{P}}$$

$$3)\ \frac{t \xrightarrow{\tau} \{p_i : t_i \mid i \in I\} \quad \forall t_i \neq t, i \in I\ t_i \overset{\alpha}{\Rightarrow}_l \mathcal{P}_i}{t \overset{\alpha}{\Rightarrow}_l \sum_{i \in I, t_i \neq t} \frac{p_i}{1-\lambda}\mathcal{P}_i}$$

where $\lambda = \sum_{i \in I, t_i = t} p_i$

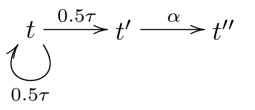
$$4)\ \frac{t \overset{\alpha}{\Rightarrow}_l \{p_i : t_i \mid i \in I\} \quad t_k \xrightarrow{\tau}_l \mathcal{P}}{t \overset{\alpha}{\Rightarrow}_l \{p_i : t_i \mid i \in I\} +_{\{t_k\}} \mathcal{P}}$$

where $k \in I, \alpha \neq x(y)$

(a)

$$\frac{\forall i\ t_i \Rightarrow_l \mathcal{P}_i}{\{p_i : t_i\}_i \Rightarrow_l \sum_i p_i\mathcal{P}_i}$$

(b)



(c)

Figure 3. Weak Concrete Transition Relations

Definition 4 gives rise to a wider notation of *weak* probabilistic choice in the form of $t \overset{\alpha}{\Rightarrow}_l \mathcal{P}$, where in case $\alpha \neq \tau$, $\mathcal{P}$ is no longer restricted to be a probability space with only one element. This certainly complements the expressiveness of π_{sp} from the observational perspective. Particularly, rule 3) in Figure 3(a) can compute out an observable probabilistic choice from a concrete one containing self τ-loops. For example, the case shown in Figure 3(c) will result in a weak probabilistic transition $t \overset{\alpha}{\Rightarrow} t''$. In this way, a "natural" weak variant of Definition 2 would be as follows.

Definition 5. An equivalence relation $\mathcal{R}$ is a *late weak bisimulation* if $(t, u) \in \mathcal{R}$ implies

1. whenever $t \xrightarrow{x(y)} \mathcal{P}$ with $x \notin fn(t, u)$ then $u \overset{x(y')}{\Longrightarrow}_l \mathcal{Q}'$ for some $\mathcal{Q}'$ such that for any z, there is $\mathcal{Q}'\{z/y'\} \Rightarrow_l \mathcal{Q}$ and $\mathcal{P}\{z/y\} \equiv_{\mathcal{R}} \mathcal{Q}$;
2. whenever $t \xrightarrow{\alpha} \mathcal{P}$ for any other action α with $bn(\alpha) \cap fn(t, u) = \emptyset$ then $u \overset{\hat{\alpha}}{\Rightarrow}_l \mathcal{Q}$ for some $\mathcal{Q}$ such that $\mathcal{P} \equiv_{\mathcal{R}} \mathcal{Q}$.

Definition 5 shows a way to define weak bisimulation without referring to the random capability of adversaries. The axiomatization results in [11] can also be applied onto the finite-state fragment of π_{sp}. However, Definition 5 suffers the same problem as mentioned in [11], that is, it is not always finitely computable, even for finite-state processes. This problem not only affects the work on axiomatization, but also rules out some reasonable case of weak bisimilarity, such as the one shown in Figure 4[1].

[1] This example was presented in [22] to illustrate the notion of weak

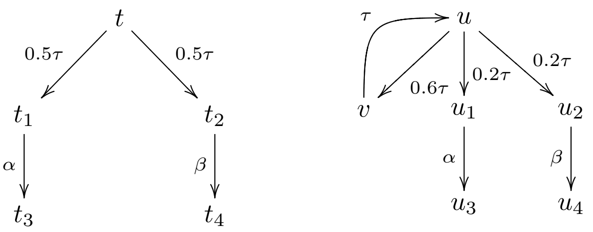


Figure 4. Weak Bisimulation

Both t and u can perform actions α and β equally likely. But u may randomly reach v that is weakly bisimilar to t (because of the transition $v \xrightarrow{\tau} u$). This leads to an observation that u weakly performs actions α and β with equal probability 0.5. Therefore, t and u should be regarded as weakly bisimilar. However such probability distribution can only be derived from u by applying the weak transition rules in Figure 3 infinitely many times. Due to the presence of a τ-loop (like in u), [11] also conjectured that the observational equivalence are not finitely axiomatizable.

This example suggests a finer version of weak bisimulation to capture its intuitive sense [20, 21]. In fact, we only need to revise the definition of lifting equivalences.

Definition 6. Let $\mathcal{R}$ be an equivalence relation over S and $s \in S$. $[s]_\mathcal{R} = \{s' \mid (s, s') \in \mathcal{R}\}$ denotes the equivalence class of s. Two discrete probability spaces $\mathcal{P}_1 = (S, \eta_1)$ and $\mathcal{P}_2 = (S, \eta_2)$ on X are *weak* $\mathcal{R}$-equivalent with respect to s, written $\mathcal{P}_1 \equiv_{\mathcal{R}\upharpoonright_s} \mathcal{P}_2$, if for each equivalence class C of $S/\mathcal{R} - [s]_\mathcal{R}$, $\xi_1(C) = \xi_2(C)$ where

$$\xi_i(C) = \begin{cases} \dfrac{\eta_i(C)}{1 - \eta_i([s]_\mathcal{R})} & \text{if } \eta_i([s]_\mathcal{R}) \neq 1 \\ \eta_i(C) & \text{otherwise} \end{cases}$$

for $i = 1, 2$.

Definition 6 evaluates probability distributions weighted by the probability of a given equivalence class. In this way, τ-loops in weak probabilistic transitions can be ignored, while the extreme probability distributions (e.g. $\{0.5 : u_1, 0.5 : u_2\}$ from u) can be reached without an infinite number of calls to weak transition rules. Therefore, the definition of weak bisimulation for π_{sp} can be given as follows.

Definition 7 (Weak Bisimulation)**.** An equivalence relation $\mathcal{R}$ is a *late weak bisimulation* if $(t, u) \in \mathcal{R}$ implies

1. whenever $t \xrightarrow{x(y)} \mathcal{P}$ with $x \notin fn(t, u)$ then $u \xRightarrow{x(y')}_l \mathcal{Q}'$ for some $\mathcal{Q}'$ such that for any z, there is $\mathcal{Q}'\{z/y'\} \Rightarrow_l \mathcal{Q}$ and $\mathcal{P}\{z/y\} \equiv_\mathcal{R} \mathcal{Q}$;
2. whenever $t \xrightarrow{\tau} \mathcal{P}$ then $u \Rightarrow_l \mathcal{Q}$ for some $\mathcal{Q}$ such that $\mathcal{P} \equiv_{\mathcal{R}\upharpoonright_t} \mathcal{Q}$.
3. whenever $t \xrightarrow{\alpha} \mathcal{P}$ for any other action α with $bn(\alpha) \cap fn(t, u) = \emptyset$ then $u \overset{\hat{\alpha}}{\Rightarrow}_l \mathcal{Q}$ for some $\mathcal{Q}$ such that $\mathcal{P} \equiv_\mathcal{R} \mathcal{Q}$.

Write $t \approx_l u$ if there exists a late weak bisimulation $\mathcal{R}$ such that $(t, u) \in \mathcal{R}$.

Note that only the case $\alpha = \tau$ requires special concerns because it is the unique case where τ-loops may occur. Thus, Definition 7 paves a way for the axiomatization of weak bisimulation with finitary inference rules.

3.2.2 Weak Symbolic Bisimulation

Definition 8. The *late weak probabilistic symbolic transitions* $t \xRightarrow{M,\alpha}_L \mathcal{P}$ are defined as the least relation satisfying the rules shown in Figure 5(a). Furthermore, we extend $\Rightarrow_L$ onto discrete probability spaces with the rule shown in Figure 5(b). Similarly, let $\overset{M}{\Longrightarrow}_L$ be $\xRightarrow{M,\epsilon}_L$.

$$1)\ \frac{}{t \xRightarrow{true,\epsilon}_L t} \qquad 2)\ \frac{t \xrightarrow{M,\alpha} \mathcal{P}}{t \xRightarrow{M,\alpha}_L \mathcal{P}}$$

$$3)\ \frac{t \xrightarrow{M,\tau} \{p_i : t_i \mid i \in I\} \quad \forall t_i \neq t, i \in I\ t_i \xRightarrow{N_i,\alpha}_L \mathcal{P}_i}{t \xRightarrow{M\wedge(\bigwedge_i N_i),\alpha}_L \sum_{i\in I, t_i\neq t} \dfrac{p_i}{1-\lambda}\mathcal{P}_i}$$

where $\lambda = \sum_{i\in I, t_i = t} p_i$

$$4)\ \frac{t \xRightarrow{M,\alpha}_L \{p_i : t_i \mid i \in I\} \quad t_k \xrightarrow{N_k,\tau}_L \mathcal{P}}{t \xRightarrow{M\wedge N_k,\alpha}_L \{p_i : t_i \mid i \in I\} +_{\{t_k\}} \mathcal{P}}$$

where $k \in I, \alpha \neq x(y)$

(a)

$$\frac{\forall i\ t_i \overset{N_i}{\Longrightarrow}_L \mathcal{P}_i}{\{p_i : t_i\}_i \xRightarrow{\bigwedge_i N_i}_L \sum_i p_i \mathcal{P}_i}$$

(b)

Figure 5. Weak Symbolic Transition Relations

Now we are set to introduce weak symbolic bisimulation for π_{sp}.

bisimulation for labeled concurrent Markov chains, which is defined in terms of schedulers and requires computing probabilities of traces, but not of single transitions. Here we reuse this example for π_{sp}, a more complex probabilistic model, to present an algebraic solution for weak bisimulation.

Definition 9 (Weak Symbolic Bisimulation)**.** A condition indexed family of equivalence relations $\mathcal{S} = \{S^B\}$ is a *late weak symbolic bisimulation* if $(t, u) \in S^B$ implies whenever $t \xrightarrow{M,\alpha} \mathcal{P}$ with $bn(\alpha) \cap fn(t, u, B) = \emptyset$, then for each $D \in MCE_{fn(t,u)}(B \cup M)$ there is a $u \overset{N,\hat{\beta}}{\Longrightarrow}_L \mathcal{Q}$ such that $D \Rightarrow N, \alpha =^D \beta$, and

- if $\alpha = x(y)$ then for each $D' \in MCE_{fn(t,u)\cup\{y\}}(D)$ there is $\mathcal{Q} \overset{N'}{\Longrightarrow}_L \mathcal{Q}'$ such that $D' \Rightarrow N'$ and $\mathcal{P} \equiv_{S^{D'}} \mathcal{Q}'$.
- if $\alpha = \bar{x}(y)$ then $\mathcal{P} \equiv_{S^{D\cup\{y\neq u \mid u\in fn(\alpha.\mathcal{P},\beta.\mathcal{Q})\}}} \mathcal{Q}$.
- if $\alpha = \tau$ then $\mathcal{P} \equiv_{S^D\restriction_t} \mathcal{Q}$.
- otherwise $\mathcal{P} \equiv_{S^D} \mathcal{Q}$.

Write $t \approx_L^B u$ if $(t, u) \in S^B \in \mathcal{S}$ for some late weak symbolic bisimulation $\mathcal{S}$.

It can be shown that $\approx_L^B$ captures $\approx_l$.

Theorem 2. $t \approx_L^B u$ iff $t\sigma \approx_l u\sigma$ for any $\sigma \vDash B$.

Proof. Similar to Theorem 1, combining the part of the proof of Theorem 2.16 in [19] for the treatment on input actions. □

4 Probabilistic Symbolic Bisimulations

In the probabilistic settings, adversaries (or schedulers) were introduced to resolve the nondeterminism among probability distributions. Moreover, an adversary can determine the next probabilistic transition probabilistically, that is, by combining several probabilistic transitions of the peer process. This gave rise to the notion of probabilistic bisimulation in the sense that one system can simulate the other by combining its probabilistic transitions and vive versa [23]. We can also formalize this notion for π_{sp} in the symbolic framework.

Definition 10 (Combined Probabilistic Transition (Symbolic Transition))**.** Given $\{t \xrightarrow{\alpha} \mathcal{P}_i\}_i$ ($\{t \xrightarrow{N_i,\alpha} \mathcal{P}_i\}_i$) and weights $w_i \geq 0$ for each i such that $\sum_i w_i = 1$, a *combined probabilistic transition (symbolic transition)* is denoted by $t \xrightarrow{\alpha}_c \mathcal{P}$ ($t \xrightarrow{\wedge_i N_i,\alpha}_c \mathcal{P}$) where $\mathcal{P} = \sum_i w_i \mathcal{P}_i$.

Note that a combined probabilistic transition $t \xrightarrow{\alpha}_c \mathcal{P}$, as well as a combined probabilistic symbolic transition $t \xrightarrow{\wedge_i N_i,\alpha}_c \mathcal{P}$, is a fully reactive probabilistic transition because for any action α, the residual $\mathcal{P}$ is no longer restricted to be a probability space with only one element.

The probabilistic version of Definition 2 is straightforward as follows.

Definition 11 (Strong Probabilistic Bisimulation)**.** An equivalence relation $\mathcal{R}$ is a *late probabilistic bisimulation* if $(t, u) \in \mathcal{R}$ implies

1. whenever $t \xrightarrow{x(y)} \mathcal{P}$ with $x \notin fn(t, u)$ then $u \xrightarrow{x(y')}_c \mathcal{Q}$ for some $\mathcal{Q}$ such that for any z, $\mathcal{P}\{z/y\} \equiv_{\mathcal{R}} \mathcal{Q}\{z/y'\}$;
2. whenever $t \xrightarrow{\alpha} \mathcal{P}$ for any other action α with $bn(\alpha) \cap fn(t, u) = \emptyset$ then $u \xrightarrow{\alpha}_c \mathcal{Q}$ for some $\mathcal{Q}$ such that $\mathcal{P} \equiv_{\mathcal{R}} \mathcal{Q}$.

Write $t \dot{\sim}_l u$ if there exists a late probabilistic bisimulation $\mathcal{R}$ such that $(t, u) \in \mathcal{R}$.

Similarly, replacing $u \xrightarrow{N,\beta} \mathcal{Q}$ in Definition 3 with $u \xrightarrow{N,\beta}_c \mathcal{Q}$ results in a *strong probabilistic symbolic bisimulation*, written $\dot{\sim}_L^B$. It is easy to see that $\sim_L^B$ is a special case of $\dot{\sim}_L^B$ because a probabilistic symbolic transition itself is a special case of combined probabilistic symbolic transitions. Moreover, $\dot{\sim}_L^B$ captures $\dot{\sim}_l$.

Theorem 3. $t \dot{\sim}_L^B u$ iff $t\sigma \dot{\sim}_l u\sigma$ for any $\sigma \vDash B$.

Proof. Similar to Theorem 1. □

In the same way, we can define a *weak probabilistic bisimulation*, written $\dot{\approx}_l$, and a *weak probabilistic symbolic bisimulation*, written $\dot{\approx}_L^B$, while $\approx_L^B$ is also a special case of $\dot{\approx}_L^B$. Moreover, $\dot{\approx}_L^B$ captures $\dot{\approx}_l$.

Theorem 4. $t \dot{\approx}_L^B u$ iff $t\sigma \dot{\approx}_l u\sigma$ for any $\sigma \vDash B$.

Proof. Similar to Theorem 2. □

In the sequel, we will consider mainly probabilistic symbolic bisimulations, while the results are also applicable for symbolic bisimulations.

5 Characterization

Symbolic bisimulations for π_{sp} inherit the feature from those for π-calculus, namely avoiding instantiating input actions with infinitely many names, while instead, allowing the construction of maximal consistent extensions to divide the name space. The advantage of symbolic bisimulations rests in an observation that if a name set V is finite, MC_V is also finite. However, it is generally expensive to compute these structures.

In addition, as far as we know, the existing bisimulation decision algorithms for probabilistic systems are mostly based on the equivalence partition techniques, where the set of states of the processes in question is regarded as the largest equivalence class and then is refined iteratively until the set of bisimulation equivalence classes is reached

[2, 6]. This is partly because the definitions of bisimulations for probabilistic systems requires equivalence relations as premises.

We present a practical characterization of symbolic bisimulations, which can release the efforts for computing maximal consistent extensions and relax the premises of equivalence relations. Following the line of [18], we introduce probabilistic schematic bisimulations for π_{sp} as follows.

Definition 12. Let $\mathcal{R}^+$ be the transitive closure of $\mathcal{R}$. A symmetric relation $\mathcal{R}$ between processes is a *probabilistic schematic bisimulation* if $(t, u) \in \mathcal{R}$ implies

- if $t \xrightarrow{x(y)} \mathcal{P}$ then $u \xrightarrow{x(y')}_c \mathcal{Q}$ for some $\mathcal{Q}$ and
  - for each $v \in fn(t, u)$, $\mathcal{P}\{v/y\} \equiv_{\mathcal{R}^+} \mathcal{Q}\{v/y'\}$;
  - for a fresh $z \notin fn(t, u)$, $\mathcal{P}\{z/y\} \equiv_{\mathcal{R}^+} \mathcal{Q}\{z/y'\}$;
- if $t \xrightarrow{\bar{x}(y)} \mathcal{P}$ then $u \xrightarrow{\bar{x}(y')}_c \mathcal{Q}$ for some $\mathcal{Q}$ and for a fresh $z \notin fn(t, u)$, $\mathcal{P}\{z/y\} \equiv_{\mathcal{R}^+} \mathcal{Q}\{z/y'\}$;
- if $t \xrightarrow{\alpha} \mathcal{P}$ for any other action α then $u \xrightarrow{\alpha}_c \mathcal{Q}$ for some $\mathcal{Q}$ and $\mathcal{P} \equiv_{\mathcal{R}^+} \mathcal{Q}$.

Write $t \dot{\asymp}_l u$ if there exists a probabilistic schematic bisimulation $\mathcal{R}$ such that $(t, u) \in \mathcal{R}$.

Theorem 5. For any $B \in MC_{fn(t,u)}$, $t \dot{\sim}_L^B u$ iff $t\sigma \dot{\asymp}_l u\sigma$ for any $\sigma \models B$.

Proof. We outline the proof below. Details are similar to Theorem 1, combining the part of the proof of Theorem 3.2 in [18] for the treatments on bound names.

($\Rightarrow$) Define $\mathcal{R} = \{(t\sigma, u\sigma) \mid t \sim_L^B u$ for some $B \in MC_{fn(t,u)}$, and $\sigma \models B\}$. It can be shown that $\mathcal{R}$ is a probabilistic schematic bisimulation.

($\Leftarrow$) Define $S^B = \{(t, u) \mid B \in MC_{fn(t,u)}, t\sigma \dot{\sim}_l u\sigma$ for some $\sigma \models B\}$. Let $\mathcal{S} = \{S^{B^+}\}$. It is obvious that S^B is symmetric. Hence S^{B^+} is an equivalence relation. Then it can be shown that $\mathcal{S}$ is a probabilistic symbolic bisimulation. □

The characterization for weak probabilistic symbolic bisimulation can be defined similarly, except that for action τ, the revised lifting equivalence should be applied, that is,

> if $t \xrightarrow{\tau} \mathcal{P}$, then $u \Rightarrow_c \mathcal{Q}$ for some $\mathcal{Q}$ and $\mathcal{P} \equiv_{\mathcal{R}^+ \lceil t} \mathcal{Q}$.

6 A Computing Algorithm

The definition of probabilistic schematic bisimulation triggers an efficient decision algorithm, as shown in Figure 6. The algorithm is adapted from the "on-the-fly" algorithm for π-calculus [18]. It assumes a totally ordered subset of names $\mathcal{SN} \subset \mathcal{N}$. The function $nextSN(t, u)$ returns a fresh name, namely the smallest name in $\mathcal{SN}$ that does not appear in $fn(t, u)$.

The function $bisim(t, u)$ is the main function of the algorithm, which attempts to find the smallest bisimulation containing the pair (t, u). The function $match(t, u)$ invokes a depth-first traversal to match outgoing probabilistic transitions of the pair (t, u). The algorithm differs from the one in [18] in the following aspects.

Numerical Computation The function $check$ tests the numerical equalities among probability distributions. For each probability distribution of one process, say $\{p_i : t_i\}_i$, the function $solve(\{p_i : t_i\}_i, \{\{q_{j_k} : u_{j_k}\}_k\}_j, E)$ checks if it can be simulated by a convex combination of probability distributions $\{\{q_{j_k} : u_{j_k}\}_k\}_j$ of the other process, with respect to the equivalence relation E over processes involved in these distributions. The implementation of the function $solve$ depends on the type of bisimulation in question.

Let $S = \{t_i\}_i$ and $T_j = \{u_{j_k}\}_k$. For checking probabilistic bisimulation, the function is to solve a linear programming problem whether there is $w_j \geq 0$ for each j such that $\sum_j w_j = 1$ and for each equivalence class $C \in (S \cup \bigcup_j T_j)/E$,

$$\sum_{t_i \in C \cap S} p_i = \sum_j w_j \sum_{u_{j_k} \in C \cap T_j} q_{j_k}$$

For checking bisimulation, the function is to solve a special case of the problem above, namely whether there is a j such that for each equivalence class $C \in (S \cup T_j)/E$,

$$\sum_{t_i \in C \cap S} p_i = \sum_{u_{j_k} \in C \cap T_j} q_{j_k}$$

The function $equiv(B)$ returns the smallest equivalence relation generated by the symmetric relation B.

Tabling An exception $WrongAssumption$ was used in [18] to force a rerun of bisimulation checking from the root level. This would invoke redundant executions on pairs of processes that are not related to the pair raising the exception.

Our algorithm can avoid such redundancy by associating each pair (t, u) with a table $Assumed_{t,u}$. $Assumed_{t,u}$ stores the pairs of processes, of which the bisimilarities depend on the bisimilarity of (t, u). When a loop is detected on a pair (t, u) (Line 44), the pair being checked is to be inserted into $Assumed_{t,u}$ (Line 45). If the pair (t, u) is found not to be bisimilar after finishing searching the loop (Line 12), we add it into $NotBisim$ (Line 13) and then recursively clear all its dependent pairs from $Visited$ (Line 14).

The algorithm will always terminate because each time a pair of processes is rematched, the size of $NonBisim$ has been increased by at least one. If $bisim(t, u)$ terminates with $true$, the set $Visited - NonBisim$ constitutes a probabilistic schematic bisimulation containing (t, u); otherwise, $(t, u) \in NonBism$, which means they are not bisimilar.

In addition, the algorithm can be well extended for checking weak probabilistic schematic bisimulation by enumerating weak transitions in $match$. Weak transitions can be derived by calling the rules in Figure 3 recursively. The following rule is needed to cut off τ-loops, according to Definition 6.

$$\frac{t \overset{N}{\Longrightarrow}_L p_k t + \sum_{i \neq k} p_i t_i}{t \overset{N}{\Longrightarrow}_L \sum_{i \neq k} \frac{p_i}{1 - p_k} t_i} \quad k \in I, \sum_{i \in I} p_i = 1$$

7 Conclusion

This paper has presented symbolic bisimulations for π_{sp}, which is a probabilistic extension of π-calculus with a probabilistic blind choice. Such a light-weight extension contributes to the definition of weak (symbolic) bisimulation that does not rely on the random capability of adversaries. The open problem on the axiomatization for weak bisimulation has been addressed by considering weighted probabilistic distributions in the definition of weak bisimulation. Furthermore, the paper has presented a practical characterization for symbolic bisimulations, which can avoid computing maximal consistent extensions of indexing conditions and relax the premises of equivalence relations. Based on the characterization, a decision algorithm (framework) has been proposed that can explore just a minimal portion of the state spaces of the probabilistic processes in question.

As future work, we would like to further investigate the axiomatizations and inference systems for symbolic bisimulations presented in this paper, and move on to probabilistic systems with metrics [10]. We are also interested in the applications of bisimulations. To name a few, the framework for anonymity checking [9] can be reinforced with symbolic bisimulations for probabilistic systems.

To take the Dining Cryptographers Problem [8] as an example, the property of strong anonymity on the cryptographers can be proved through checking if the system is weakly bisimilar to

$$\frac{1}{4}\tau.Obs_{daa} + \frac{1}{4}\tau.Obs_{ada} + \frac{1}{4}\tau.Obs_{aad} + \frac{1}{4}\tau.Obs_{ddd}$$

where each Obs_{xxx} ($x = a$ or d) represents a possible observation.

On the other hand, the numerical computation involved in the decision algorithm can be reused to compute the probability of reachability. The above specification can be parameterized as

$$x_0\tau.Obs_{daa} + x_1\tau.Obs_{ada} + x_2\tau.Obs_{aad} + x_3\tau.Obs_{ddd}$$

with the constraint $\sum_{i=0}^{4} x_i = 1$. The probability $x_i (i = 0, ..., 3)$ can be resolved through the bisimulation decision algorithm.

Acknowledgements The work of Peng Wu and Catuscia Palamidessi was partially supported by the INRIA/ARC project ProNoBis. Special thanks go to Yuxin Deng for pointing out an error in the earlier version. Valuable comments and suggestions from anonymous referees are also appreciated.

References

[1] A. Aldini. Probabilistic information flow in a process algebra. In *the Proceedings of the 12th International Conference on Concurrency Theory (CONCUR'01)*, pages 152–168, Aalborg, Denmark, August 20-25 2001.

[2] C. Baier, B. Engelen, and M. E. Majster-Cederbaum. Deciding bisimilarity and similarity for probabilistic processes. *Journal of Computer and System Sciences*, 60(1):187–231, 2000.

[3] M. Bhargava and C. Palamidessi. Probabilistic anonymity. In *the Proceedings of 16th International Conference on Concurrency Theory (CONCUR'05)*, pages 171–185, San Francisco, CA, USA, August 23-26 2005.

[4] M. Boreale and R. D. Nicola. A symbolic semantics for the π-calculus. *Information and Computation*, 126(1):34–52, 1996.

[5] A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Bisimulation and unwinding for verifying possibilistic security properties. In *the Proceedings of the 4th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'03)*, pages 223–237, New York, NY, USA, January 9-11 2003.

[6] S. Cattani and R. Segala. Decision algorithms for probabilistic bisimulation. In *the Proceedings of the 13th International Conference on Concurrency Theory (CONCUR'02)*, pages 371–385, Brno, Czech Republic, August 20-23 2002.

[7] K. Chatzikokolakis and C. Palamidessi. A framework to analyze probabilistic protocols and its application to the partial secrets exchange. In *the Proceedings of International Symposium on Trustworthy Global Computing (TGC'05)*, Edinburgh, UK, April 7-9 2005.

[8] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

```
$bisim(t,u) =$
    $NotBisim = \emptyset, root = (t,u)$
    fun bis(t,u) = {
        $Visited = \emptyset$
        match(t,u)
    } handle $WrongAssumption \Rightarrow bis(t,u)$
    return $match(t,u)$

$match(t,u) =$
    $Visited = Visited \cup \{(t,u)\}, Assumed_{t,u} = \emptyset$
    $b = \bigwedge_{\alpha \in \{\tau, \bar{x}y, \bar{x}, x\}} match_{\alpha}(t,u)$
    if (not $b$) {
        $NotBisim = NotBisim \cup \{(t,u)\}$
        $clear\_dep(t,u)$
    }
    return $b$

$match_{\{\alpha \mid \alpha = \tau, \bar{x}y\}}(t,u) =$
    for each $t \xrightarrow{\alpha} \mathcal{P}_i = \{p_{i_m} : t_{i_m}\}_{i_m}, u \xrightarrow{\alpha} \mathcal{Q}_j = \{q_{j_n} : u_{j_n}\}_{j_n}$
        for each $m, n$
            $b_{i_m j_n} = close(t_{i_m}, u_{j_n}, t, u)$
    return $check(\{\mathcal{P}_i\}_i, \{\mathcal{Q}_j\}_j, \{b_{i_m j_n}\}_{i,m,j,n})$

$match_{x(y)}(t,u) =$
    for each $t \xrightarrow{x(y)} \mathcal{P}_i = \{p_{i_m} : t_{i_m}\}_{i_m}, u \xrightarrow{x(y)} \mathcal{Q}_j = \{q_{j_n} : u_{j_n}\}_{j_n}$
        for each $m, n$ {
            $b_{i_m j_n} = true$
            for each $v \in fn(t,u) \cup \{nextSN(t,u)\}$
                $b_{i_m j_n} = b_{i_m j_n} \wedge close(t_{i_m}\{v/x\}, u_{j_n}\{v/y\}, t, u)$
        }
    return $check(\{\mathcal{P}_i\}_i, \{\mathcal{Q}_j\}_j, \{b_{i_m j_n}\}_{i,m,j,n})$

$match_{\bar{x}}(t,u) =$
    for each $t \xrightarrow{\bar{x}(y)} \mathcal{P}_i = \{p_{i_m} : t_{i_m}\}_{i_m}, u \xrightarrow{\bar{x}(y)} \mathcal{Q}_j = \{q_{j_n} : u_{j_n}\}_{j_n}$
        for each $m, n$ {
            $z = nextSN(t,u)$
            $b_{i_m j_n} = close(t_{i_m}\{z/x\}, u_{j_n}\{z/y\}, t, u)$
        }
    return $check(\{\mathcal{P}_i\}_i, \{\mathcal{Q}_j\}_j, \{b_{i_m j_n}\}_{i,m,j,n})$

$close(t, u, td, ud) =$
    if $(t,u) \in NotBisim$
        return false
    else if $(t,u) \in Visited$ {
        $Assumed_{t,u} = Assumed_{t,u} \cup \{(td, ud)\}$
        return true
    }
    else return $match(t,u)$

$check(\{\mathcal{P}_i\}_i, \{\mathcal{Q}_j\}_j, \{b_{i_m j_n}\}_{i,m,j,n}) =$
    for each $i$
        $b_i = solve(\mathcal{P}_i, \bigcup_j \{\mathcal{Q}_j\}, equiv(\{b_{i_m j_n}\}_{m,j,n}))$
    for each $j$
        $b'_j = solve(\mathcal{Q}_j, \bigcup_i \{\mathcal{P}_i\}, equiv(\{b_{i_m j_n}\}_{n,i,m}))$
    return $\bigwedge_i b_i \wedge \bigwedge_j b'_j$

$clear\_dep(t,u) =$
    for each $(t', u') \in Assumed_{t,u}$
        if $((t', u') \neq (t,u))$ {
            $clear\_dep(t', u')$
            $Visited = Visted - \{(t', u')\}$
        }
    $Assumed_{t,u} = \emptyset$
```

Figure 6. Bisimulation Decision Algorithm for π_{sp}

[9] T. Chothia, S. Orzan, J. Pang, and M. T. Dashti. A framework for automatically checking anonymity with mcrl. In *the Proceedings of the 2nd Symposium on Trustworthy Global Computing (TGC'06)*, Lucca, Italy, November 7-9 2006.

[10] Y. Deng, T. Chothia, C. Palamidessi, and J. Pang. Metrics for action-labelled quantitative transition systems. *Electronic Notes in Theoretical Computer Science*, 153(2):79–96, 2006.

[11] Y. Deng and C. Palamidessi. Axiomatizations for probabilistic finite-state behaviors. In *the Proceedings of the 8th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'05)*, pages 110–124, Edinburgh, UK, April 4-8 2005.

[12] R. J. V. Glabbeek, S. A. Smolka, and B. Steffen. Reactive, generative, and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.

[13] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.

[14] O. M. Herescu and C. Palamidessi. Probabilistic asynchronous π-calculus. In *the Proceedings of 3rd International Conference on Foundations of Software Science and Computation Structures (FOSSACS'00)*, pages 146–160, Berlin, Germany, March 25-April 2 2000.

[15] R. Lanotte, A. Maggiolo-Schettini, and A. Troina. Weak bisimulation for probabilistic timed automata and applications to security. In *the Proceedings of the First International Conference on Software Engineering and Formal Methods (SEFM'03)*, pages 34–43, Brisbane, Australia, September 22-27 2003.

[16] H. Lin. Symbolic bisimulation and proof systems for the π-calculus. Technical report, School of Cognitive and Computer Science, University of Sussex, Sussex, UK, 1994.

[17] H. Lin. Unique fixpoint induction for mobile processes. In *the Proceedings of 6th International Conference on Concurrency Theory (CONCUR'95)*, pages 88–102, Philadelphia, Pennsylvania, USA, August 21-24 1995.

[18] H. Lin. Computing bisimulations for finite-control π-calculus. *Journal of Computer Science and Technology*, 15(1):1–9, 2000.

[19] H. Lin. Complete inference systems for weak bisimulation equivalences in the π-calculus. *Information and Computation*, 180(1):1–29, 2003.

[20] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[21] R. Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999.

[22] A. Philippou, I. Lee, and O. Sokolsky. Weak bisimulation for probabilistic systems. In *the Proceedings of the 11th International Conference on Concurrency Theory (CONCUR'00)*, pages 334–349, University Park, PA, USA, August 22-25 2000.

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

[24] A. Sokolova and E. P. de Vink. Probabilistic automata: System types, parallel composition and comparison. In *Validation of Stochastic Systems*, pages 1–43, 2004.

A Proof of Proposition 1

Proposition 1 Suppose $B \in MC_{fn(t,u)}$, $t \sim_L^B u$ iff $t\sigma \sim_l u\sigma\ for\ any\ \sigma \vDash B.$

Proof. ($\Rightarrow$) Define $\mathcal{R} = \{(t\sigma, u\sigma) \mid t \sim_L^B u$ for some $B \in MC_{fn(t,u)}$, and $\sigma \vDash B\}$. We show $\mathcal{R}$ is a late bisimulation. It is easy to see that $\mathcal{R}$ is an equivalence relation.

Suppose $(t\sigma, u\sigma) \in \mathcal{R}$. Let $(t, u) \in S^B \in \mathcal{S}$ for some late symbolic bisimulation $\mathcal{S}$ and $t\sigma \xrightarrow{\alpha} \mathcal{P}'$ with $bn(\alpha) \cap fn(t\sigma, u\sigma) = \emptyset$. Four cases are to be considered on the types of α. Herein we mainly examine the cases $\alpha = \tau, \bar{x}(y)$.

- $\alpha = \tau$. By Lemma 3(2), there exists M, $\mathcal{P}$ such that $\sigma \vDash M$, $\mathcal{P}' \equiv \mathcal{P}\sigma$ and $t \xrightarrow{M,\tau} \mathcal{P}$. By Lemma 2, $n(M) \subseteq fn(t)$. Since $B \in MC_{fn(t,u)}$, by Lemma 1, $B \Rightarrow M$. Hence up to logical equivalence, B is the only element in $MCE_{fn(t,u)}(B \cup M)$. By Definition 3, there exists $u \xrightarrow{N,\tau} \mathcal{Q}$ such that $B \Rightarrow N$ and $\mathcal{P} \equiv_{S^B} \mathcal{Q}$. Hence $\sigma \vDash N$ and by Lemma 3(1), $u\sigma \xrightarrow{\tau} \mathcal{Q}\sigma$.

  The next step is to show $\mathcal{P}\sigma \equiv_{\mathcal{R}} \mathcal{Q}\sigma$, which is not as obvious as the one for π-calculus in [18, 19]. Suppose $\mathcal{P} = \{p_i : t_i \mid i \in I\}$ and $\mathcal{Q} = \{q_j : u_j \mid i \in J\}$. By Lemma 2, for any $i \in I$, $j \in J$, $fn(t_i, u_j) \subseteq fn(t, u)$. Since $B \in MC_{fn(t,u)}$, $B \in MC_{fn(t_i,u_j)}$ for any $i \in I$, $j \in J$. Recall that $\sigma \vDash B$, it follows that up to logical equivalence, for any $i \in I$, $j \in J$, $(t_i\sigma, u_j\sigma) \in R$ iff $(t_i, u_j) \in S^B$. Therefore, from $\mathcal{P} \equiv_{S^B} \mathcal{Q}$ we get $\mathcal{P}\sigma \equiv_{\mathcal{R}} \mathcal{Q}\sigma$.

- $\alpha = \bar{x}(y)$. Suppose $t\sigma \xrightarrow{\bar{x}(y)} t'\sigma$. For sake of simplicity, we assume that y is a fresh name, that is, $y \notin fn(t, u) \cup n(\sigma) \cup n(C)$. In the same way, it follows there exists x', z' and $u\sigma \xrightarrow{\bar{x}(y)} u'\sigma$ such that $x \equiv x'\sigma \equiv z'\sigma$, $x' =^B z'$, $t' \equiv_{S^D} u'$ and $D = B \cup \{y \neq z \mid z \in fn(\bar{x'}(y).t', \bar{z'}(y).u')\}$. Then from $B \in MC_{fn(t,\underline{u})}$, $fn(t'\underline{,} u') \subseteq fn(\bar{x'}(y).t', \bar{z'}(y).u') \cup \{y\}$ and $fn(\bar{x'}(y).t', \bar{z'}(y).u') \subseteq fn(t, u)$, we have $D \in MC_{fn(t',u')}$. Furthermore, since $\sigma \vDash B$ and $y \notin fn(t, u) \cup n(\sigma)$, $\sigma \vDash D$. Hence, $(t'\sigma, u'\sigma) \in \mathcal{R}$. By Definition 1, it follows that $t'\sigma \equiv_{\mathcal{R}} u'\sigma$.

- The other cases are similar.

($\Leftarrow$) Define $S^B = \{(t, u) \mid B \in MC_{fn(t,u)}, t\sigma \sim_l u\sigma$ for some $\sigma \vDash B\}$. Let $\mathcal{S} = \{S^B\}$. We show $\mathcal{S}$ is a late symbolic bisimulation. It is easy to see that S^B is an equivalence relation.

Suppose $(t, u) \in S^B$. Let $(t\sigma, u\sigma) \in \mathcal{R}$ for some late strong bisimulation $\mathcal{R}$ and $t \xrightarrow{M,\alpha} \mathcal{P}$ with $bn(\alpha) \cap fn(t, u, B) = \emptyset$. It is reliable to assume $bn(\alpha) \cap n(\sigma) = \emptyset$. By Lemma 2, $n(M) \subseteq fn(t, u)$. If M is inconsistent with B then the conclusion follows from $MCE_{fn(t,u)}(B \cup M) = \emptyset$; otherwise, for the same reason mentioned in the first part, B is the only element of in $MCE_{fn(t,u)}(B \cup M)$ and $\sigma \vDash M$. The proof proceeds by finding a matching symbolic transition from u.

- $\alpha = \tau$. By Lemma 3(1), $t\sigma \xrightarrow{\tau} \mathcal{P}' \equiv \mathcal{P}\sigma$. Thus, by Definition 2, there exists $u\sigma \xrightarrow{\tau} \mathcal{Q}'$ such that $\mathcal{P}' \equiv_{\mathcal{R}} \mathcal{Q}'$. Then by Lemma 3(2), there exists $N, \mathcal{Q}$ such that $\sigma \vDash N$, $\mathcal{Q}' \equiv \mathcal{Q}\sigma$ and $u \xrightarrow{N,\tau} \mathcal{Q}$. Since $\sigma \vDash B$, $B \Rightarrow N$ by Lemma 1.

  The last step is to show $\mathcal{P} \equiv_{S^B} \mathcal{Q}$. Suppose $\mathcal{P} = \{p_i : t_i \mid i \in I\}$ and $\mathcal{Q} = \{q_j : u_j \mid i \in J\}$. It follows that $B \in MC_{fn(t_i,u_i)}$ for any $i \in I$, $j \in J$, by Lemma 2 and $B \in MC_{fn(t,u)}$. Furthermore, since $\sigma \vDash B$, $(t_i, u_j) \in S^B$ iff $(t_i\sigma, u_j\sigma) \in \mathcal{R}$ for any $i \in I$, $j \in J$. Therefore, from $\mathcal{P}\sigma \equiv_R \mathcal{Q}\sigma$ we get $\mathcal{P} \equiv_{S^B} \mathcal{Q}$.

- $\alpha = \bar{x}(y)$. Suppose $t \xrightarrow{M,\bar{x}(y)} t'$ and y is a fresh name. In the same way, it follows that there exists N, x', u' such that $\sigma \vDash N$, $x =^B x'$, $t'\sigma \equiv_R u'\sigma$ and $u \xrightarrow{N,\bar{x'}(y)} u'$. By Lemma 2, $n(N) \in fn(u)$. Since $B \in MC_{fn(t,u)}$ and $\sigma \vDash B$, $B \Rightarrow N$ by Lemma 1. Furthermore, let $D = B \cup \{y \neq z \mid z \in fn(\bar{x}(y).t', \bar{x'}(y).u')\}$. Since $fn(t', u') \subseteq fn(\bar{x}(y).t', \bar{x'}(y).u') \cup \{y\}$ and $fn(\bar{x}(y).t', \bar{x'}(y).u') \subseteq fn(t, u)$, $D \in MC_{fn(t',u')}$. From $\sigma \vDash B$ and $y \notin fn(t, u) \cup n(\sigma)$, we get $\sigma \vDash D$. Hence, $(t', u') \in \mathcal{S}^{\mathcal{D}}$. By Definition 1, it follows that $t' \equiv_{\mathcal{S}^{\mathcal{D}}} u'$.

- The other cases are similar.

□