# Measuring the Confinement of Concurrent Probabilistic Systems

Alessandra Di Pierro[1], Chris Hankin[2*], and Herbert Wiklicky[2]

[1] Dipartimento di Informatica, Universitá di Pisa, Italy
[2] Department of Computing, Imperial College, London, UK

**Abstract.** In previous work we have studied the notion of *approximate confinement* based on I/O observables. In this paper, we generalise this work to consider probabilistic bisimulation (both exact and approximate) in the context of probabilistic transition systems. We also show the relationship between probabilistic bisimulation and probabilistic abstract interpretation; this result parallels Schmidt's result in the classical (non-probabilistic) setting where he relates bisimulation and Galois connections. We conclude with some examples.

## 1 Introduction

Since the early 1970s, when it was first raised by Lampson [19], the problem of preventing a program P from leaking private information to unauthorised users (aka *confinement problem*) has been extensively studied and various approaches have been proposed for its solution. After the introduction in the 1980s of the seminal notion of *noninterference* by Goguen and Meseguer [14], and in the 1990s of *probabilistic noninterference* by Gray [15], most of the works on confinement have been based on models which exploit the noninterference-based formalisation of the problem: since (probabilistic) interference can be exploited by a Trojan horse to reliably leak high information to unauthorised users, the absence of any illegal information flow will guarantee the perfect confinement of a system. Such models ultimately depend on some notion of process equivalence by identifying the absence of information flow between two processes via the indistinguishability of their behaviours [24]. As already noticed in [15] these models are aimed to achieve perfect security. This is in practice hardly achievable [23]. By weakening the definition of the confinement property so as to allow for a quantifiable amount of interference we can make the definition more usable, i.e. more systems will satisfy the definition. Moreover, we allow system developers to formally quantify the security of the system, e.g. to precisely determine the capacity of (probabilistic) covert channels. They are then able to trade off the security of the system with other design goals such as performance or reliability.

We have recently been studying confinement properties in the setting of a simple probabilistic programming language [10, 9]. The main contribution of this earlier work has been the development of a notion of *approximate confinement*

* Chris Hankin is partly funded by the EU FET open project SecSafe.

which allows for the leakage of a certain amount $\varepsilon$ of information. Such a quantity gives a measure of how hard an attacker has to work in order to breach security. The process equivalence we have based our definition on considers I/O observables. We have also previously considered special kinds of attackers which can be external or internal and are equipped with a specific limited power.

In this current paper we present a significant generalisation: we cast our work in the context of probabilistic transition systems [18]. These systems are probabilistic extensions of labelled transition systems which represent a well-established semantic model for concurrent and distributed systems. In this context we consider *bisimulation* as the basic process equivalence for defining confinement both in its exact and approximate versions.

We will adopt the model introduced by Larsen and Skou [20], where a probabilistic transition system is thought of as a system which reacts to actions of the environment by means of probability distributions on the set of states of the system. In the terminology of [17, 18, 27] these are called reactive systems and do not contain non-determinism or in other words are purely probabilistic. Larsen and Skou adapt the key behavioural equivalence for ordinary transition systems, namely bisimulation, to these systems by introducing the notion of probabilistic bisimulation. This is elegantly characterised by means of a testing language so that two states are probabilistic bisimilar if and only if they react with the same probability distribution to each test.

By interpreting tests as possible attackers, this notion immediately translates into a definition of confinement for probabilistic concurrent systems. In fact, two processes which are probabilistic bisimilar are indistinguishable under any attack (or test). Since the latter is a process in a generic language (called in [20] the testing language), this definition of confinement generalises the definitions introduced in our previous work where we restrict to particular kind of spies (namely passive and memoryless spies [8, 10]).

One main result of this paper is the introduction of a characterisation of both classical and probabilistic bisimulation equivalence via probabilistic abstract interpretation [12]. This translates into the probabilistic setting a result that was already established in the classical setting by Schmidt [25]. This characterisation is obtained via the representation of a probabilistic transition system via a linear operator. The equivalence between two systems is then established by the existence of certain linear transformations (abstractions). The use of linear operators to represent relations provides us with a means to define a notion of distance via an appropriate operator norm. In particular, we will use the operator representing probabilistic bisimulation to define a quantity $\varepsilon$ representing "how much" two processes are not bisimilar, or equivalently how much the system formed by the two processes is not confined.

## 2 Probabilistic Transition Systems

In this paper we will consider probabilistic transition systems (PTS), that is labelled transition systems with a probabilistic branching.

Given a finite set $S$, we call a function $\pi : S \mapsto [0,1]$ a *distribution* on $S$ iff $\sum_{s\in S} \pi(s) = 1$. We denote by $Dist(S)$ the set of all distributions on $S$. Every distribution corresponds to a vector in the free vector space $\mathcal{V}(S)$ of formal linear combinations of elements in $S$.

Given an equivalence relation $\sim$ on a finite set $S$ and a distribution $\pi$ on $S$, the *lifting* of $\pi$ to $S/_\sim$, the set of equivalence classes of $\sim$ in $S$, is defined for each equivalence class $[s] \in S/_\sim$ by: $\pi([s]) = \sum_{s'\in[s]} \pi(s')$. It is straightforward to show that this is indeed a distribution on $S/_\sim$ (e.g. [18, Def 1 & Thm 1]).

**Definition 1.** *[18, Def 2] A* probabilistic transition system *(PTS) is a tuple $(S, A, \rightarrow, \pi_0)$, where: $S$ is a non-empty, finite set of* states*, $A$ is a non-empty, finite set of* actions*, $\rightarrow \subseteq S \times A \times Dist(S)$ is a* transition relation*, and $\pi_0 \in Dist(S)$ is an* initial distribution *on $S$.*

For $s \in S$, $\alpha \in A$ and $\pi \in Dist(S)$ we write as usual $s \xrightarrow{\alpha} \pi$ for $(s, \alpha, \pi) \in \rightarrow$. By $s \xrightarrow{\alpha}_{\pi(t)} t$ we denote the transition to individual states $t$ with probability $\pi(t)$.

The above definition of a PTS allows for purely probabilistic as well as non-deterministic transitions as there might be more than one distribution associated to a state $s$ and an action $\alpha$. In this paper we will concentrate on purely probabilistic models, also called *reactive* systems in [17, 18, 27]). In this class of probabilistic transition systems all states $s$ are deterministic in the sense that for each action $\alpha$, if $s \xrightarrow{\alpha} \pi_1$ and $s \xrightarrow{\alpha} \pi_2$ then $\pi_1 = \pi_2$. In a variant of Definition 1 the transition relation is a subset of $S \times Dist(A \times S)$, i.e. the same probability distribution is used to govern both the choice of the action and the (internal) state transition. This model, called *generative* in [17], is more concrete than the reactive model where no information is given about the relative probability of performing an action $\alpha$ versus an action $\beta$ in a given initial state. In our illustrating example in Section 4, we will adopt this kind of transition system as the underlying semantics for our probabilistic process calculus.

### 2.1 Linear Representation of PTS

Relations have a simple representation as linear operators. If the domain is finite then these operators are finite-dimensional, that is matrices. In order to define the matrix associated to a relation on a set $X$, we first have to lift $X$ to a vector space.

**Definition 2.** *The* free vector space $\mathcal{V}(X)$ *over a set $X$ is the space of formal linear combinations of elements in $X$ with coefficients in some field $\mathbb{W}$ (e.g. $\mathbb{W} = \mathbb{R}$), i.e. $\mathcal{V}(X) = \{\sum c_x \boldsymbol{x} \mid c_x \in \mathbb{W},\ x \in X\}$.*

We associate to each relation $R \subseteq X \times X$ a 0/1-matrix, i.e. a linear operator $\mathbf{M}_R$ on $\mathcal{V}(X)$ defined by: $(\mathbf{M}_R)_{ij} = 1$ iff $(x, y) \in R$ and $(\mathbf{M}_R)_{ij} = 0$ otherwise, where $i, j \in X$, and $(\mathbf{M}_R)_{ij}$ denotes the entry in column $i$ and row $j$ in the matrix representing $\mathbf{M}_R$. Analogously, the matrix representing a quantitative relation $R \subseteq X \times \mathbb{W} \times X$, that is a relation where a quantity $w \in \mathbb{W}$ is associated to each pair in $X \times X$, is defined by: $(\mathbf{M}_R)_{ij} = w$ iff $(x, w, y) \in R$ and $(\mathbf{M}_R)_{ij} = 0$ otherwise. For probabilistic relations, where $\mathbb{W} = [0, 1]$, we obtain a *stochastic matrix*. Interpreting relations as linear operators allows us to measure them. The standard way to measure the "size" of a linear operator is via an *operator norm* which in turn may have its origins in a vector *norm* [16, Section 7.18].

**Definition 3.** *Given a probabilistic transition system* $X = (S, A, \longrightarrow, \pi_0)$, *we define its* matrix representation $\mathbf{M}(X)$ *as the direct sum of the matrix representations of the transition relations* $\xrightarrow{\alpha}$ *for each* $\alpha \in A$: $\mathbf{M}(X) = \bigoplus_{\alpha \in A} \mathbf{M}(\xrightarrow{\alpha})$.

Distributions must therefore be represented by vectors in the vector space $\mathcal{V}(S) \oplus \ldots \oplus \mathcal{V}(S) = \mathcal{V}(S)^{|A|}$. The matrix $\mathbf{M}(X)$ represents a linear operator on this space. For example, the initial distribution $\pi_0$ will be represented by $\pi_{\mathbf{0}} = \bigoplus_{\alpha \in A} \pi_0$. It is easy to see that starting with $\pi_{\mathbf{0}}$ and applying $\mathbf{M}(X)$ repeatedly for $n$ steps, we get the distributions corresponding to the $n$-step closure of $\longrightarrow$ (by summing up the factors in the direct sum).

For the sake of simplicity we will denote by $\mathbf{X}$ the matrix $\mathbf{M}(X)$, by $\mathbf{X}_\alpha$ the matrices $\mathbf{M}(\xrightarrow{\alpha})$, and by $\mathbf{PX}$ the multiplication of a direct sum $\bigoplus_\alpha \mathbf{P}$ of the same matrix $\mathbf{P}$ with $\mathbf{X}$. By the properties of the direct sum this is the same as $\bigoplus_\alpha (\mathbf{PX}_\alpha)$. Note that for each $\alpha$, $\mathbf{X}_\alpha$ is a stochastic matrix in the reactive model, while it isn't in general for generative systems.

Given a probabilistic transition system $X = (S, A, \longrightarrow, \pi_0)$ and a state $p \in S$, we denote by $S_p \subseteq S$ the set of all states reachable from $p$, by $T(p)$ the transition system induced on the restricted state space $S_p$, and by $\mathbf{M}(p)$ the matrix representation of $T(p)$.

## 3 Probabilistic Bisimulation and Confinement

We briefly recall the classical notion of bisimulation for labelled transition systems:

**Definition 4.** *[27, Def 12] A* bisimulation *is a binary relation* $\sim_b$ *on states of a labelled transition system satisfying for all* $\alpha \in A$:

$$p \sim_b q \text{ and } p \xrightarrow{\alpha} p' \Rightarrow \exists q' : q \xrightarrow{\alpha} q' \text{ and } p' \sim_b q',$$
$$p \sim_b q \text{ and } q \xrightarrow{\alpha} q' \Rightarrow \exists p' : p \xrightarrow{\alpha} p' \text{ and } q' \sim_b p'.$$

Given two processes $p$ and $q$, we say that they are *bisimilar* if there exists a bisimulation relation $\sim_b$ such that $p \sim_b q$. Bisimulations are equivalence relations [27, Prop 8.1].

The standard generalisation of this notion to probabilistic transition systems, i.e. *probabilistic bisimulation*, is due to [20], where the adopted reference model is that of purely probabilistic (or reactive) systems.

**Definition 5.** *[18, Def 4] A* probabilistic bisimulation *is an equivalence relation* $\sim_b$ *on states of a probabilistic transition system satisfying for all* $\alpha \in A$:

$$p \sim_b q \text{ and } p \xrightarrow{\alpha} \pi \Rightarrow q \xrightarrow{\alpha} \varrho \text{ and } \pi \sim_b \varrho.$$

This definition is equivalent to the characterisation of probabilistic bisimulation given in [20] in terms of "button pressing" tests. Such tests are formally defined by means of a language which specifies the syntactical structure of algorithms for experimenting on a process (i.e. which button to press when). In our security setting such tests represent possible interferences by a spy, and observing the probabilistic result of an experiment corresponds to establishing whether a system is confined (the spy is not able to distinguish the processes in the system) or not. The first case corresponds to a system whose processes are probabilistic bisimilar. This is intuitively the idea behind the following definition of probabilistic confinement for processes specified by a PTS:

**Definition 6.** *Let $\mathcal{T} = (S, A, \rightarrow, \pi_0)$ be a probabilistic transition system and let $T(p)$ and $T(q)$, $p, q \in S$, represent two processes in a probabilistic language modelled by $\mathcal{T}$. Then we say that $p$ and $q$ are probabilistic confined iff they are probabilistic bisimilar.*

### 3.1 Probabilistic Confinement as Probabilistic Abstract Interpretation

Probabilistic Abstract Interpretation has been introduced in [11, 12] as a probabilistic version of the classical abstract interpretation framework introduced by Cousot & Cousot [4, 5]. Due to lack of space we omit a detailed description of this framework, for which we refer to the cited papers. We will just recall here the basic definitions and the technique we will use in this paper. Given two probabilistic domains, $\mathcal{C}$ and $\mathcal{D}$, a *probabilistic abstract interpretation* is a pair of linear maps, $\mathbf{A} : \mathcal{C} \mapsto \mathcal{D}$ and $\mathbf{G} : \mathcal{D} \mapsto \mathcal{C}$, between the concrete domain $\mathcal{C}$ and the abstract domain $\mathcal{D}$, such that $\mathbf{G}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$, and vice versa. Recall the definition of the Moore-Penrose pseudo-inverse:

**Definition 7.** *[3, Def 1.1.2] Let $\mathcal{C}$ and $\mathcal{D}$ be two finite-dimensional vector spaces and $\mathbf{A} : \mathcal{C} \mapsto \mathcal{D}$ a linear map between them. A linear map $\mathbf{A}^\dagger = \mathbf{G} : \mathcal{D} \mapsto \mathcal{C}$ is the* Moore-Penrose pseudo-inverse *of $\mathbf{A}$ iff*

$$\mathbf{A} \circ \mathbf{G} = \mathbf{P}_A \quad \textit{and} \quad \mathbf{G} \circ \mathbf{A} = \mathbf{P}_G$$

*where $\mathbf{P}_A$ and $\mathbf{P}_G$ denote orthogonal projections onto the ranges of $\mathbf{A}$ and $\mathbf{G}$.*

A simple method to construct a probabilistic abstract interpretation which we will use in this paper is as follows. Given a linear operator $\Phi$ on some vector space $\mathcal{V}$ expressing the probabilistic semantics of a concrete system, and a linear abstraction function $\mathbf{A} : \mathcal{V} \mapsto \mathcal{W}$ from the concrete domain into an abstract domain $\mathcal{W}$, we compute the (unique) Moore-Penrose pseudo-inverse $\mathbf{G} = \mathbf{A}^\dagger$ of $\mathbf{A}$. The abstract semantics can then be defined as the linear operator on the abstract domain $\mathcal{W}$: $\Psi = \mathbf{A} \circ \Phi \circ \mathbf{G}$.

We call an $n \times m$-matrix $\mathbf{K}$ a *classification* matrix, if it is a 0/1-matrix, where every row has exactly one non-zero entry and columns have at least one non-zero entry. Classification matrices are thus particular kinds of stochastic matrices. We denote by $\mathcal{K}(n, m)$ the set of all $n \times m$-classification matrices.

**Proposition 1.** *Let $X = \{x_1, \ldots, x_n\}$ be a finite set. Then for each equivalence relation $\approx$ on $X$ with $|X/_\approx| = m$, there exists a classification matrix $\mathbf{K} \in \mathcal{K}(n,m)$ and vice versa.*

As a result, the equivalence class $[x]$ of an element $x \in X$ can be obtained as the coordinates of the vector $\boldsymbol{x} \cdot \mathbf{K}$, where $\boldsymbol{x} \in \mathcal{V}(X)$. The pseudo-inverse of a classification matrix $\mathbf{K} \in \mathcal{K}(n,m)$ corresponds to its normalised transpose or adjoint (these coincide for real $\mathbf{K}$).

It is easy to see that a probabilistic bisimulation equivalence $\sim$ on a PTS $T = (S, A, \rightarrow, \pi_0)$ is nothing else than a probabilistic abstract interpretation of $T$. In fact, by Proposition 1, there is a classification matrix $\mathbf{K} \in \mathcal{K}(|S|, m)$, for some $m$ which represents $\sim$. If $\mathbf{M}(T)$ is the matrix representation of $T$ then $\mathbf{K}^\dagger\mathbf{M}(T)\mathbf{K}$ is the abstract operator induced by $\mathbf{K}$. Intuitively, this is an $m \times m$ matrix which abstracts the original system $T$ by encoding only the transitions between equivalence classes instead of the ones between single states.

Consider now two processes $p, q \in S$ and their matrix representations $\mathbf{M}(p)$ and $\mathbf{M}(q)$. These are the factors of $\mathbf{M}(T)$ associated to the portions of the graph $T$ rooted on $p$ and $q$ respectively, and containing only those nodes which correspond to states reachable by $p$ and $q$ respectively. The restrictions of $\mathbf{K}$ to these two sets of nodes, which we call $\mathbf{K}_p$ and $\mathbf{K}_q$ are the abstraction operators for the two processes $p$ and $q$ and allow us to express exactly the condition for the probabilistic bisimilarity of $p$ and $q$ as stated by the following proposition.

**Proposition 2.** *Given the matrix representation $\mathbf{M}(p)$ and $\mathbf{M}(q)$ of two probabilistic processes $p$ and $q$, then $p$ and $q$ are bisimilar iff there exists a $\mathbf{K}_p \in \mathcal{K}(|S_p|, m)$ and $\mathbf{K}_q \in \mathcal{K}(|S_q|, m)$ such that*

$$\mathbf{K}_p^\dagger\mathbf{M}(p)\mathbf{K}_p = \mathbf{K}_q^\dagger\mathbf{M}(q)\mathbf{K}_q.$$

**Corollary 1.** *Given the matrix representation $\mathbf{M}(p)$ and $\mathbf{M}(q)$ of two processes $p$ and $q$, then $p$ and $q$ are bisimilar, i.e. $p \sim_b q$, iff there exists a PTS $x$ which is a probabilistic abstract interpretation of both $p$ and $q$.*

### 3.2 Approximate Bisimulation and Confinement

When two processes $p$ and $q$ are not probabilistic bisimilar, we can still claim that they are confined although only approximately. In order to do so, we introduce an $\varepsilon$-version of probabilistic bisimilarity. The intuitive idea is to find a classification matrix $\mathbf{K}$ which is the closest one to a bisimulation relation in which $p$ and $q$ are equivalent. The difference between the abstract operators induced by $\mathbf{K}$ for the two processes will give an estimate of the non-bisimilarity degree of $p$ and $q$, and therefore of their confinement.

**Definition 8.** *Let $T = (S, A, \longrightarrow, \pi_0)$ be a probabilistic transition systems and let $p$ and $q$ be two states in $S$ with matrix representations $\mathbf{X}$ and $\mathbf{Y}$. We say that $p$ and $q$ are $\varepsilon$-*bisimilar*, denoted by $p \sim_b^\varepsilon q$, iff*

$$\min_{\mathbf{K}_p, \mathbf{K}_q \in \mathcal{K}} \|\mathbf{K}_p^\dagger\mathbf{X}\mathbf{K}_p - \mathbf{K}_q^\dagger\mathbf{Y}\mathbf{K}_q\| = \varepsilon$$

*where $\|.\|$ denotes an appropriate norm.*

Note that it is possible to use this definition also to introduce an approximate version of the classical notion of bisimulation. Furthermore, for $\varepsilon = 0$ we recover the original notion of strict (probabilistic) bisimulation:

**Proposition 3.** *For finite transition systems, an $\varepsilon$-bisimulation with $\varepsilon = 0$, i.e. $\sim_b^0$, is a (probabilistic) bisimulation.*

However, for infinite PTS the concepts 0-bisimulation and (probabilistic) bisimulation will differ in general.

Notice that this approximate bisimilarity is no longer an equivalence relation. However, we conjecture the validity of a triangle inequality; i.e. $p \sim_b^{\varepsilon_1} q$ and $q \sim_b^{\varepsilon_2} r$ imply that $p \sim_b^{\varepsilon} r$ with $\varepsilon \leq \varepsilon_1 + \varepsilon_2$.

## 4 An Example

We will illustrate our approach by an example of processes in a simple probabilistic process calculus, namely PCCS [13, 18]. PCCS is based on Milner's SCCS [21, 22] and processes communicate via pure synchronisation. We have deviated from the published version of the language slightly, in that we use hiding rather than restriction. The syntax of PCCS is as follows:

$$E ::= \mathbf{0} \;\Big|\; X_i \;\Big|\; \alpha : E \;\Big|\; \textstyle\sum_{i \in I} [p_i] E_i \;\Big|\; E \times F \;\Big|\; E \backslash F \;\Big|\; E[f] \;\Big|\; \mathit{fix}_i \tilde{X} \tilde{E}$$

The small step semantics for PCCS is shown in Table 1.

$$\alpha : E \xrightarrow{\alpha}_1 E \qquad\qquad \frac{E_j \xrightarrow{\alpha}_q E'}{\Sigma_{i \in I}[p_i]E_i \xrightarrow{\alpha}_{p_j q} E'} \quad (j \in I)$$

$$\frac{E \xrightarrow{\alpha}_p E' \quad F \xrightarrow{\beta}_q F'}{E \times F \xrightarrow{\alpha\beta}_{pq} E' \times F'} \qquad\qquad \frac{E \xrightarrow{\alpha}_p E'}{E \backslash A \xrightarrow{\alpha}_{\tilde{p}} E' \backslash A} \quad (\alpha \notin A)$$

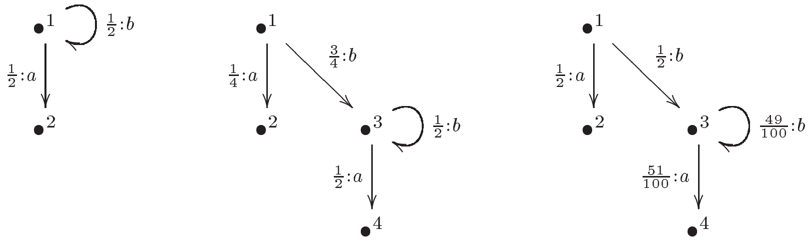
$$\frac{E \xrightarrow{\alpha}_p E'}{E[f] \xrightarrow{f(\alpha)}_p E'[f]} \qquad\qquad \frac{E_i\{\mathit{fix}\tilde{X}\tilde{E}/\tilde{X}\} \xrightarrow{\alpha}_p E'}{\mathit{fix}_i \tilde{X}\tilde{E} \xrightarrow{\alpha}_p E'}$$

**Table 1.** Small Step Semantics of PCCS

*Example 1.* Let us compare the following PCCS processes (we use $+$ for binary summation and the suffix indicates the probability of choosing the right summand):

$$\begin{aligned} A &\equiv \mathit{fix}\ X.b : X +_{\frac{1}{2}} a : \mathbf{0} \\ B &\equiv a : \mathbf{0} +_{\frac{3}{4}} (\mathit{fix}\ X.b : X +_{\frac{1}{2}} a : \mathbf{0}) \\ C &\equiv a : \mathbf{0} +_{\frac{1}{2}} (\mathit{fix}\ X.b : X +_{\frac{51}{100}} a : \mathbf{0}) \end{aligned}$$

Their transition graphs are given by:




These processes are not probabilistic bisimilar. We can however calculate an estimate of their non-bisimilarity. The matrices representing the three processes are:

$$\mathbf{A} = \mathbf{M}(A) = \mathbf{M}_a(A) \oplus \mathbf{M}_b(A) = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \mathbf{M}(B) = \mathbf{M}_a(B) \oplus \mathbf{M}_b(B) = \begin{pmatrix} 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{C} = \mathbf{M}(C) = \mathbf{M}_a(C) \oplus \mathbf{M}_b(C) = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{51}{100} \\ 0 & 0 & 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{49}{100} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The problem is to find classification matrices $\mathbf{K}_A, \mathbf{K}_B$, and $\mathbf{K}_C$ such that the norm of the difference between $\mathbf{K}_A\mathbf{A}\mathbf{K}_A^\dagger$ and $\mathbf{K}_B\mathbf{B}\mathbf{K}_B^\dagger$ or $\mathbf{K}_C\mathbf{C}\mathbf{K}_C^\dagger$ is minimal. There is only a finite (though exponentially growing) number of possible classification operators $\mathbf{K} \in \mathcal{K}$. A brute force approach looking at all possible $\mathbf{K}$ allows us to determine in this case the $\varepsilon$-bisimilarity of $A$ and $B$, and of $A$ and $C$. Interestingly, the optimal $\mathbf{K} = \mathbf{K}_B = \mathbf{K}_C$ is coincidentally the same in both cases:

$$\mathbf{K} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{K}^\dagger = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

while for $\mathbf{K}_A$ we can take the identity.

Measuring the difference based on the supremum norm leads to the following result:

$$\inf_{\mathbf{K}\in\mathcal{K}} \|\mathbf{A} - \mathbf{K}\mathbf{B}\mathbf{K}^\dagger\|_\infty = \frac{1}{8} \quad \text{and} \quad \inf_{\mathbf{K}\in\mathcal{K}} \|\mathbf{A} - \mathbf{K}\mathbf{C}\mathbf{K}^\dagger\|_\infty = \frac{1}{200},$$

from which we can for example see that process $A$ is much more bisimilar to process $B$ than it is to process $C$.

## 5 Concluding Remarks

Various quantitative versions of probabilistic bisimulation have been proposed in the literature, based on metrics [13] or pseudo-metrics [7, 26], but without a direct application to security.

The recent work of Aldini *et al* [1] is closely related to the work reported here. They develop their ideas on approximate bisimulation for a CSP-like calculus. Their notion of $\varepsilon$-bisimilarity is based on the supremum norm of a vector of transitions. The main difference bewteen our work and theirs is our characterisation of bisimulation as a probabilistic abstract interpretation. This is an important difference for us because we intend to use this work as a basis for "calculating" static security analyses.

The relation between abstract interpretation and (bi)simulation has been recognised before in the classical Galois connection based setting ([6], [25]), but this appears to be the first investigation of such a relation in a probabilistic setting.

## References


1. A. Aldini, M. Bravetti, and R. Gorrieri. A process algebraic approach for the analysis of probabilistic non-interference. *Journal of Computer Security*, 2003. to appear.
2. J.A. Bergstra, A. Ponse, and S.A. Smolka, editors. *Handbook of Process Algebra*. Elsevier Science, Amsterdam, 2001.
3. S.L. Campbell and D. Meyer. *Generalized Inverse of Linear Transformations*. Constable and Company, London, 1979.
4. P. Cousot and R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Symposium on Principles of Programming Languages (POPL)*, pages 238–252, Los Angeles, 1977.
5. P. Cousot and R. Cousot. Systematic Design of Program Analysis Frameworks. In *Symposium on Principles of Programming Languages (POPL)*, pages 269–282, San Antonio, Texas, 1979.
6. D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Transactions on Programming Languages and Systems*, 19(2):253–291, 1997.
7. J. Desharnais, R. Jagadeesan, V. Gupta, and P.Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 413–422, Copenhagen, Denmark, 22-25 July 2002. IEEE Computer Society.
8. A. Di Pierro, C. Hankin, and H. Wiklicky. Probabilistic confinement in a declarative framework. In *Declarative Programming – Selected Papers from AGP 2000 – La Havana, Cuba*, volume 48 of *Electronic Notes in Theoretical Computer Science*, pages 1–23. Elsevier, 2001.
9. A. Di Pierro, C. Hankin, and H. Wiklicky. Approximate confinement under uniform attacks. In *SAS'02: 9th International Static Analysis Symposium*, volume 2477 of *Lecture Notes in Computer Science*. Springer Verlag, 2002.

10. A. Di Pierro, C. Hankin, and H. Wiklicky. Approximate non-interference. In Iliano Cervesato, editor, *CSFW'02 — 15th IEEE Computer Security Foundation Workshop*, pages 3–17, Cape Breton, Canada, 24–26 June 2002. IEEE Computer Society Press.
11. A. Di Pierro and H. Wiklicky. Concurrent Constraint Programming: Towards Probabilistic Abstract Interpretation. In M. Gabbrielli and F. Pfenning, editors, *Proceedings of PPDP'00 – Priciples and Practice of Declarative Programming*, pages 127–138, Montréal, Canada, September 2000. ACM SIGPLAN, Association of Computing Machinery.
12. A. Di Pierro and H. Wiklicky. Measuring the precision of abstract interpretations. In *Proceedings of LOPSTR'00 – 10th International Workshop on Logic-Based Program Synthesis and Transformation, London, UK*, volume 2042 of *Lecture Notes in Computer Science*, pages 147–164, Berlin – New York, 2001. Springer Verlag.
13. A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of the IFIP WG 2.2/2.3 Working Conference on Programming Concepts and Methods*, pages 443–458, Sea of Gallilee, April 1990. North-Holland.
14. J. Goguen and J. Meseguer. Security Policies and Security Models. In *IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.
15. J.W. Gray, III. Towards a mathematical foundation for information flow security. In *Proceedings of the 1991 Symposium on Research in Security and Privacy*, pages 21–34, Oakland, California, May 1991. IEEE Computer Society.
16. W.H. Greub. *Linear Algebra*, volume 97 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag, New York, third edition, 1967.
17. R.J. van Glabbeek, S.A. Smolka and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121:59–80, 1995.
18. B. Jonsson, W. Yi, and K.G. Larsen. *Probabilistic Extentions of Process Algebras*, chapter 11, pages 685–710. Elsevier Science, Amsterdam, 2001. see [2].
19. B.W. Lampson. A Note on the Confinement Problem. *Communications of the ACM*, 16(10):613–615, 1973.
20. K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991.
21. R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin – New York, 1980.
22. R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.
23. P.Y.A. Ryan, J. McLean, J. Millen, and V. Gilgor. Non-interference, who needs it? In *Proceedings of the 14th IEEE Computer Security Foundations Workshop*, pages 237–238, Cape Breton, Nova Scotia, Canada, June 2001. IEEE.
24. P.Y.A. Ryan and S.A. Schneider. Process algebra and non-interference. *Journal of Computer Security*, 9(1/2):75–103, 2001. Special Issue on CSFW-12.
25. D.A. Schmidt. Binary relations for abstraction and refinement. In *Workshop on Refinement and Abstraction*, Amagasaki, Japan, November 1999.
26. F. van Breugel and J. Worrell. Towards quantitative verification of probabilistic transition systems. In *ICALP: 28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 421–432. Springer Verlag, 2001.
27. R.J. van Glabbeek. *The Linear Time - Branching Time Spectrum I. The Semantics of Concrete, Sequential Processes*, chapter 1, pages 3–99. Elsevier Science, Amsterdam, 2001. see [2].