Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 340 (2005) 3 – 56

Theoretical Computer Science

www.elsevier.com/locate/tcs

Measuring the confinement of probabilistic systems

Alessandra Di Pierro[a,1,3], Chris Hankin[b,∗,2,3], Herbert Wiklicky[b,3]

[a] *Dipartimento di Informatica, Universitá di Pisa, Italy*
[b] *Department of Computing, Imperial College, London, UK*

Abstract

In this paper we lay the semantic basis for a quantitative security analysis of probabilistic systems by introducing notions of *approximate confinement* based on various process equivalences. We re-cast the operational semantics classically expressed via probabilistic transition systems (PTS) in terms of linear operators and we present a technique for defining approximate semantics as probabilistic abstract interpretations of the PTS semantics. An operator norm is then used to quantify this approximation. This provides a quantitative measure ε of the indistinguishability of two processes and therefore of their confinement. In this security setting a statistical interpretation is then given of the quantity ε which relates it to the number of tests needed to breach the security of the system.

Keywords: Security analysis; Probabilistic bisimulation; Probabilistic weak bisimulation; Static program analysis; Statistical testing

1. Introduction

Since the early 1970s, when it was first raised by Lampson [43], the problem of preventing a program P from leaking private information to unauthorised users (also known as the *confinement problem*) has been extensively studied and various approaches have been proposed

[∗] Corresponding author. Tel.: +44 20 7594 8266; fax: +44 207 581 8024.

E-mail addresses: dipierro@di.unipi.it (A. Di Pierro), clh@imperial.ac.uk (C. Hankin), herbert@doc.ic.ac.uk (H. Wiklicky).

[1] Alessandra Di Pierro is partly funded by Progetto MEFISTO.

[2] Chris Hankin is partly funded by the EU FET project SecSafe.

[3] All three authors are partly funded by the EPSRC project S77066A.

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.03.002

for its solution. After the introduction in the 1980s of the seminal notion of *non-interference* by Goguen and Meseguer [31], and in the 1990s of *probabilistic non-interference* by Gray [32], most of the work on confinement has been based on models which exploit the non-interference-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 [59]. As already noticed in [32] these models aim to achieve perfect security. This is in practice hardly achievable [58]. The definition of the confinement property can be made more usable (i.e. systems are more likely to satisfy the definition) by weakening it so as to allow for a quantifiable amount of interference. This also allows 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, reliability or costs [13].

We have previously studied confinement properties in the setting of a simple probabilistic programming language [52,53,55]. The main contribution of this earlier work has been the development of a notion of *approximate confinement* which allows for the leakage of a certain amount ε 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. Moreover, this definition refers to special kinds of attackers which can be external or internal and are equipped with a specific limited power.

In this paper we present a significant generalisation: we cast our work in the context of probabilistic transition systems [39]. These systems are probabilistic extensions of labelled transition systems which represent a well-established semantics for concurrent and distributed systems. Various models have been proposed in the literature which differ in the way probability is introduced in the underlying non-deterministic model. In the most simple extension probabilistic branching completely replaces non-deterministic branching, although transition probability distributions may depend on the occurrences of actions in different ways. In [29] these different ways are classified in three alternative models called respectively ‘reactive’, ‘generative’ and ‘stratified’. In other probabilistic extensions some form of non-determinism is allowed in order to represent under-specification [36,38,61,65]. Non-determinism can be useful for specifying the behaviour of concurrent processes, i.e. for expressing the different interleavings in the parallel execution of concurrent probabilistic processes; for this reason the models of probabilistic transition systems including non-determinism are often called *concurrent* probabilistic systems [4].

We will adopt the purely probabilistic model concentrating in particular on the reactive and the generative variants. The basic difference between these two variants is that while in the reactive systems, first introduced by Larsen and Skou in [44], each action determines a probability distribution on the states reachable on that action, in the generative systems probability distributions are defined on pairs of actions and states, thus implicitly assigning probabilities also to the occurrences of actions. In this context we will consider *bisimulation* and *weak bisimulation* as the basic process equivalences for defining confinement both in its exact and approximate versions. The notion of probabilistic bisimulation we will adopt is the one introduced by Larsen and Skou in [44] for reactive systems. 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 systems. In fact, two processes which are probabilistic bisimilar are indistinguishable under any attack (or test). These tests are formalised in [44] as processes in a generic language, called a *testing language*, and can be used to represent different kinds of attacks. In particular, for generative systems these tests are *passive* as they do not determine the probabilistic behaviour of the system. Thus, the definition of confinement induced by probabilistic bisimulation generalises the definitions introduced in our previous work where we consider a probabilistic language with a generative semantics and restrict to a particular kind of spies, namely passive and memoryless spies [51,52].

One main result of this paper is the introduction of a characterisation of probabilistic bisimulation equivalence via probabilistic abstract interpretation [56,57]. This translates into the probabilistic setting a result that was already established in the classical setting by Schmidt [60]. Our characterisation is obtained via the representation of a probabilistic transition system by means of a linear operator. The equivalence between two systems is then established by the existence of certain linear transformations (abstractions). In the bisimulation semantics, such abstractions result in a "lumped" process [41]; in fact, as pointed out by [63], Larsen and Skou's notion of probabilistic bisimulation is a recasting of Kemeny and Snell's lumpability condition. 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 operators representing probabilistic bisimulation to define a quantity ε which measures "how much" two processes are not bisimilar. We also show how these same notions can be used to capture weak probabilistic bisimulation and its approximate variant. Our definition of weak bisimulation for probabilistic systems is similar to the one introduced in [4] for generative systems. Because of the presence of τ actions, a straightforward application of the technique we use for probabilistic bisimulation is non-trivial for weak bisimulation. In particular, the linear operator representing the abstracted system must be defined so as to take into account possible looping on τ-transitions.

Computing an ε measure can be computationally expensive, if not infeasible; thus we show how to establish a bound for ε which is easy to compute. Finally, we give a statistical interpretation of ε which, in the setting of security, allows us to relate the level of confinement to the number of tests a spy has to perform in order to breach the system.

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 set S, we call a function $\pi : S \mapsto [0,1]$ a *probability distribution* on S iff $\sum_{s\in S} \pi(s) = 1$. We call the function π a *sub-probability distribution* iff $\sum_{s\in S} \pi(s) \leqslant 1$. We denote by $\mathit{Dist}(S)$ and $\mathrm{SDist}(S)$ the set of all probability and sub-probability distributions on S, respectively. Given an equivalence relation $\sim$ on a finite set S and a distribution π on S, the *lifting* of π to the set of equivalence classes of $\sim$ in S, $S/\sim$, 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

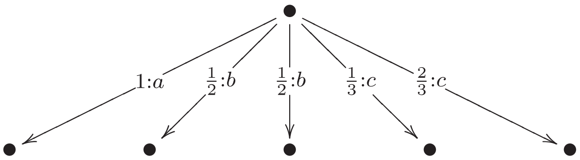


Fig. 1. A reactive probabilistic transition system.

this is indeed a distribution on $S/\sim$ (e.g. [39, Definition 1, Theorem 1]). Analogously, we can show that if π is a sub-probability distribution on S then its lifting is a sub-probability distribution on $S/\sim$.

We recall the definition of a PTS as given in [39, Definition 2].

Definition 1. A *probabilistic transition system* is a tuple $(S, A, \longrightarrow, \pi_0)$, where:

- S is a non-empty, countable set of *states*,
- A is a non-empty, finite set of *actions*,
- $\longrightarrow \subseteq S \times A \times \mathit{Dist}(S)$ is a *transition relation*, and
- $\pi_0 \in \mathit{Dist}(S)$ is an *initial distribution* on S.

For $s \in S$, $\alpha \in A$ and $\pi \in \mathit{Dist}(S)$ we write $s \xrightarrow{\alpha} \pi$ for $(s, \alpha, \pi) \in \longrightarrow$. By $s \xrightarrow{p:\alpha} t$ we denote the transition to individual states t with probability $p = \pi(t)$, on action α.

The above definition of a PTS is very general and allows for purely probabilistic models where each transition is assigned a probability (as in e.g. [14,29,44]) as well as models where both non-deterministic and probabilistic branching are present (as in e.g. [36,38,61,65]). We will consider in this paper two particular variants of this definition which correspond to the *reactive* and *generative* models in [29]. In reactive systems each action determines a probability distribution and for each state s and action α only one distribution is possible, i.e. if $s \xrightarrow{\alpha} \pi_1$ and $s \xrightarrow{\alpha} \pi_2$ then $\pi_1 = \pi_2$. In the generative systems distributions implicitly assign a probability also to the occurrences of actions. Formally we can define the reactive and generative model as a particular case of Definition 1 where the transition relation is a partial function from the set of states into $\mathit{Dist}(A \times S)$, and from the set of the pairs (state, action) into $\mathit{Dist}(S)$, respectively. More formally, we will consider the following definition.

Definition 2. A *reactive system* is a PTS $(S, A, \longrightarrow, \pi_0)$, where the transition relation is a partial function $\longrightarrow : S \times A \hookrightarrow \mathit{Dist}(S)$.

A *generative system* is a PTS $(S, A, \longrightarrow, \pi_0)$, where the transition relation is a partial function $\longrightarrow : S \hookrightarrow \mathit{Dist}(S \times A)$.

An example of a reactive PTS is depicted in Fig. 1. The environment provides three possible actions a, b and c. Once an action has been chosen (or in the terminology of [45], the experiment of pressing the associated button succeeds) the process makes an internal state transition according to the probability distribution associated to that action.

For generative systems, the same probability distribution is used to govern both the choice of the action and the (internal) state transition. This model, also called *fully probabilistic* in

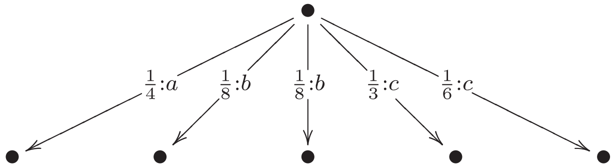


Fig. 2. A generative probabilistic transition system.

[4], is more concrete than the reactive model as all choices are now resolved probabilistically. In Fig. 2 we give an example of a generative PTS. Here it is the process which decides according to a given probability which of the actions *a*, *b* and *c* provided by the environment it will react to.

Note that the fact that in Definition 2 transition relations are partial functions is due to the presence of terminal states, i.e. states where no transitions are possible for any actions. While reactive systems are similar in their structure to Markov Decision Processes where we abstract from the reward function [26], generative systems are effectively discrete Markov chains labelled with actions. This will be made more clear in Section 3.4 where we will introduce linear representations of PTS.

3. Operator representation of quantitative relations

In order to provide an appropriate mathematical framework for the quantitative study of non-interference and confidentiality for probabilistic processes we will recast the common relational presentation of probabilistic transition systems in terms of linear maps and operators. To this purpose we first introduce quantitative relations and present a general way to represent them via linear operators; then we instantiate this method to the particular case of transition relations which are binary, quantitative relations $\longrightarrow \subseteq S \times \mathbb{W} \times S$ on the set *S* of the program states, and with "weights" taken from some appropriate set (ring, field, etc.) $\mathbb{W}$.

In the case where *S* is finite the framework we will consider is essentially algebraic as linear operators on finite dimensional spaces are canonically represented by (finite) matrices. This simple finite setting is sufficient for the treatment of terminating processes and covers also the case of processes with infinite execution paths as long as only finitely many states are involved. In the more general case where *S* is countably infinite we will need to develop additional topological notions which go beyond basic linear algebra and require the use of functional analytical and operator algebraic methods.

3.1. Quantitative relations

Transition relations and probabilistic transition relations are special kinds of *quantitative relations*. As already mentioned, we will consider in this paper at most countable state spaces.

Definition 3. Given a countable set *X* and a set $\mathbb{W}$, a *quantitative relation R* over *X* with weights in $\mathbb{W}$ is a subset $R \subseteq X \times \mathbb{W} \times X$.

For numerical weights—i.e. for $\mathbb{W}$ being a ring, field, etc.—we can interpret $R \subseteq X \times \mathbb{W} \times X$ as a function $R \subseteq X \times X \to \mathbb{W}$ by adding all the weights associated to the same pair $(x, y) \in X \times X$, i.e.

$$R(x, y) = \sum_{(x,w,y)\in R} w.$$

Classical or *qualitative relations* are quantitative relations with $\mathbb{W} = \{0, 1\}$ and $R(x, y) \in \{0, 1\}$. A *probabilistic relation* on a set *X* is a subset $R \subseteq X \times [0, 1] \times X$ such that $p(x) = 1$ for all $x \in X$, where

$$p(x) = \sum\{p \mid (x, p, y) \in R \text{ and } y \in X\} = \sum_{y\in X} R(x, y).$$

In the case of $p(x) \leqslant 1$, *R* will be called a *sub-probabilistic relation*.

3.2. Linear maps and operators

The idea of representing quantitative relations, and in particular transition relations, as matrices is quite straightforward. By a *matrix* we simply mean a (possible countably infinite) rectangular arrangement of weights (see e.g. [33, Definition 3.1.]). We associate to each classical relation $R \subseteq X \times X$ a $0/1$-matrix defined by

$$(\mathbf{M}_R)_{xy} = \begin{cases} 1 & \text{iff } (x, y) \in R, \\ 0 & \text{otherwise,} \end{cases}$$

where $x, y \in X$, and $(\mathbf{M}_R)_{xy}$ denotes the entry in column *x* and row *y* in the matrix $\mathbf{M}_R$. Analogously, the matrix representing a quantitative relation $R \subseteq X \times \mathbb{W} \times X$ is defined by

$$(\mathbf{M}_R)_{xy} = \begin{cases} p & \text{iff } R(x, y) = p, \\ 0 & \text{otherwise.} \end{cases}$$

For probabilistic (sub-probabilistic) relations we obtain a so-called *stochastic* (*sub-stochastic*) matrix, that is a positive matrix where the entries in each row sum up to one (are less than or equal to one).

It is well known from basic linear algebra that matrices are not just schemes for writing down weights but also a way to specify *linear maps* between vector spaces.

Our aim is to investigate the properties of quantitative (transition) relations via their associated linear maps and operators. A similar approach towards analysing the structure of (finite and infinite) graphs is at the center of so-called *algebraic graph theory*, e.g. [7,46,68].

In order to achieve this we introduce an appropriate vector space construction:

Definition 4. The *vector space* $\mathcal{V}(X)$ over a set *X* is the space of formal linear combinations of elements in *X* with coefficients in a field $\mathbb{W}$; we can represent the elements in $\mathcal{V}(X)$ as infinite vectors with coefficients in $\mathbb{W}$ and indexed by *X*:

$$\mathcal{V}(X) = \{(v_x)_{x\in X} \mid v_x \in \mathbb{W}\}.$$

We will assume in our treatment a very general set of weights, namely the complex numbers. This allows us to embed other weight sets of interest in a single general structure, as $\{0, 1\} \subseteq [0, 1] \subseteq \mathbb{R} \subseteq \mathbb{C}$. The use of $\mathbb{C}$ as "the field" is also standard practice in operator theory as it avoids various technical problems (e.g. related to the roots of polynomials).

Clearly, $\mathcal{V}(X)$ has indeed the algebraic structure of a vector space; multiplication with a scalar $c \in \mathbb{C}$ and vector addition can be defined component-wise simply by:

$$c(v_x)_{x\in X} = (cv_x)_{x\in X} \quad \text{and} \quad (v_x)_{x\in X} + (w_x)_{x\in X} = (v_x + w_x)_{x\in X},$$

while the zero vector o is given by $o_x = 0$ for all $x \in X$. Every (sub-probability) distribution corresponds to a vector in $\mathcal{V}(S)$.

For finite sets X of cardinality n, the representation of quantitative relations on X as linear operators on $\mathcal{V}(X)$ is straightforward since $\mathcal{V}(X)$ is isomorphic to the n-dimensional vector space $\mathbb{C}^n$. The matrix representation $\mathbf{M}_R$ of a relation R on X defines a linear operator on $\mathcal{V}(X)$ which, by abuse of notation, we also denote by $\mathbf{M}_R : \mathcal{V}(X) \to \mathcal{V}(X)$ and which is defined via

$$(\mathbf{M}_R\,((v_x)_{x\in X}))_{y\in X} = \left(\sum_{x\in X} v_x (\mathbf{M}_R)_{xy}\right)_{y\in X}.$$

The application of $\mathbf{M}_R$ to a vector $v = (v_x)_{x\in X} \in \mathcal{V}(X)$ is thus simply implemented by vector/matrix multiplication. It is easy to see that this indeed defines a *linear operator* on $\mathcal{V}(X)$, i.e. $\mathbf{M}_R(v + w) = \mathbf{M}_R(v) + \mathbf{M}_R(w)$, and $\mathbf{M}_R(cv) = c\mathbf{M}_R(v)$ for all $c \in \mathbb{C}$ and $w, v \in \mathcal{V}(X)$. We denote the set of all linear maps between two vector spaces $\mathcal{V}$ and $\mathcal{W}$ by $\mathcal{L}(\mathcal{V}, \mathcal{W})$ and the set of linear operators on $\mathcal{V}$ by $\mathcal{L}(\mathcal{V}) = \mathcal{L}(\mathcal{V}, \mathcal{V})$. Note that $\mathcal{L}(\mathcal{V})$ is itself again a vector space with $(c\mathbf{M})(v) = c\mathbf{M}(v)$ and $(\mathbf{M} + \mathbf{N})(v) = \mathbf{M}(v) + \mathbf{N}(v)$. We write $\mathbf{M}_R(v)$ for the application of $\mathbf{M}_R$ to v, but $v\mathbf{M}_R$ when we consider the matrix multiplication which implements this application. Similarly, function composition of linear maps can be implemented by (reverse) matrix multiplication: given two linear maps **M** and **N** their composition $\mathbf{M} \circ \mathbf{N}$ is represented by the matrix we obtain as the product **NM**.

In the case of finite sets X, i.e. for finite dimensional vector spaces $\mathcal{V}(X)$, there is in fact a one-to-one correspondence between matrices and linear maps, e.g. [33, 3.2]. Furthermore, the finite dimensional case also leads to a unique topological structure [33, 1.22] and every linear map/operator is automatically continuous.

For countably infinite sets, however, the situation is more complicated. It is no problem to utilise an infinite countable matrix in order to define a map in the same way as in the finite case. However, for a general infinite matrix we have no guarantee that $\sum_{x\in X} v_x (\mathbf{M}_R)_{xy}$ exists. As an example, for $v_x = 1$ and $(\mathbf{M}_R)_{xy} = 1$ for all natural numbers $x, y \in \mathbb{N}$, then this results in an infinite vector w with $w_x = \infty$ for all $x \in \mathbb{N}$.

Furthermore, even if we restrict ourselves to only those relations for which their matrix representation results in a well-defined linear map we still have the problem that the algebra of infinite matrices which we obtain this way is topologically "unstable". This algebra has no universal topological structure (like in the finite dimensional case) and the notions of linearity and continuity do not coincide. It is therefore difficult to define the limit of a sequence of infinite matrices in a general way.

To overcome these problems, we will restrict our attention to relations which can be represented concretely as so-called bounded linear operators on a Hilbert space or, in other words, correspond to elements of a C*-algebra. From a topological viewpoint C*-algebras are particularly well behaved operator algebras. The algebraic structure of a C*-algebra allows for exactly one (norm) topology [47, Corollary 2.1.2], and thus offers in some sense the same advantages as the linear algebra of finite dimensional matrices.

3.3. Some operator theory

We assume in the following a basic knowledge of concepts in *functional analysis* and *operator theory*, as one can find for example in [15,27,47,69].

To simplify our treatment we consider only complex vector spaces and algebras, i.e. we assume, as before, that the base field is $\mathbb{C}$. We denote by $\overline{\ .\ }$ the complex conjugation in $\mathbb{C}$, i.e. $\overline{x + \mathrm{i}y} = x - \mathrm{i}y$.

3.3.1. Normed vector spaces

The notion of *norm* is essential for our treatment of the countable case and therefore we recall here the basic definition.

Definition 5. A *norm* on a vector space $\mathcal{V}$ is a map $\|.\| : \mathcal{V} \mapsto \mathbb{R}$ such that for all $v, w \in \mathcal{V}$ and $c \in \mathbb{C}$:
(i) $\|v\| \geqslant 0$,
(ii) $\|v\| = 0 \Leftrightarrow v = o$,
(iii) $\|cv\| = |c|\|v\|$,
(iv) $\|v + w\| \leqslant \|v\| + \|w\|$,
with $o \in \mathcal{V}$ the zero vector.

We can always use a norm to define a metric topology on a vector space via the distance function $d(v, w) = \|v - w\|$.

Definition 6. Given a normed vector space $\mathcal{V}$ the *operator norm* for linear operators $\mathbf{M}$: $\mathcal{V} \to \mathcal{V}$ on $\mathcal{V}$ is defined by

$$\|\mathbf{M}\| = \sup_{v \in \mathcal{V}} \frac{\|\mathbf{M}(v)\|}{\|v\|} = \sup_{\|v\|=1} \|\mathbf{M}(v)\|.$$

The operator norm, if defined, is indeed a norm on $\mathcal{L}(\mathcal{V})$ and depends on the particular vector norm $\|.\|$. Common examples of (vector) norms are:

1-norm or taxi cab-norm: $\|(v_i)_i\|_1 = \sum_i |v_i|$,

2-norm or Euclidian norm: $\|(v_i)_i\|_2 = \sqrt{\sum_i |v_i|^2}$,

∞-norm or supremum-norm: $\|(v_i)_i\|_\infty = \sup_i |v_i|$.

In the case of finite dimensional vector spaces—although in general resulting in numerically different values, all these norms induce equivalent topologies, i.e. convergence in one norm implies convergence in any of the others. However, for infinite dimensional vector spaces this is not any more the case.

3.3.2. Bounded operators on Hilbert spaces

In order to deal with "well-behaved" relations on countable infinite spaces we first define a restricted vector space on X.

Definition 7. The *Hilbert space* $\ell^2(X)$ over a countable set X is the space of formal linear combinations of elements in X with coefficients in $\mathbb{C}$ which we can represent as infinite vectors with complex coefficients and indexed by elements in X such that:

$$\ell^2(X) = \left\{ (v_x)_{x\in X} \mid v_x \in \mathbb{C} \text{ and } \sum_{x\in X} |v_x|^2 < \infty \right\}.$$

Clearly, $\ell^2(X) \subseteq \mathcal{V}(X)$ and scalar multiplication and vector addition can be defined in the same way as for $\mathcal{V}(X)$. In the finite dimensional case we can identify $\mathcal{V}(X) \cong \ell^2(X)$. Furthermore, the standard inner product $\langle ., . \rangle : \ell^2(X) \times \ell^2(X) \to \mathbb{C}$ defined by

$$\langle (v_x)_{x\in X}, (w_x)_{x\in X} \rangle = \sum_{x\in X} v_x \overline{w_x}$$

can be used to define the standard norm on ℓ^2, that is the Euclidian norm, as $\|v\|_2 = \sqrt{\langle v, v \rangle}$. Well-known results show that this is indeed a norm, and that the induced metric topology is complete, i.e. all Cauchy sequences converge. Furthermore, one can show that every separable Hilbert space $\mathcal{H}$ is isomorphic to the "standard" Hilbert space $\ell^2 = \ell^2(\mathbb{N})$, see e.g. [40, Corollary 2.2.13].

The second element of our model of "well-behaved" relations on countable infinite spaces is a restriction to a particular class of linear operators.

Definition 8. A linear operator $\mathbf{M} \in \mathcal{L}(\mathcal{H})$ on a Hilbert space $\mathcal{H}$ is said to be *bounded* if its operator norm is bounded, i.e. if $\|\mathbf{M}\| < \infty$. We denote by $\mathcal{B}(\mathcal{H})$ the set of bounded linear operators on $\mathcal{H}$.

Obviously we have $\mathcal{B}(\mathcal{H}) \subseteq \mathcal{L}(\mathcal{H})$ and that $\mathcal{B}(\mathcal{H})$ is a vector space, i.e. the vector space operations inherited from $\mathcal{L}(\mathcal{H})$ do not lead out of $\mathcal{B}(\mathcal{H})$. Furthermore, we can define an algebra product between elements in $\mathcal{B}(\mathcal{H})$ as the function composition.

It is straightforward to show that a linear operator $\mathbf{M}$ on $\mathcal{H}$ is continuous if and only if it is bounded, e.g. [15, Proposition 1.1]. This means that as in the finite dimensional case linearity and continuity coincide for operators in $\mathcal{B}(\mathcal{H})$.

3.3.3. C-Algebras*

In the following we will utilise operators in $\mathcal{B}(\ell^2(X))$ as our model of "well-behaved" quantitative relations on a countable infinite space X. The domain $\mathcal{B}(\ell^2(X))$ is important as it provides the standard example of a so-called *C*-algebra*. We recall some of the basic definitions and results from the theory of C*-algebras.

An *algebra* is a vector space $\mathcal{A}$ together with a map $\mathcal{A} \times \mathcal{A} \to \mathcal{A}$ denoted by $(a, b) \mapsto a \cdot b = ab$, which is bi-linear—i.e. $a(\alpha b) = \alpha ab$, $(za)b = zab$ for $z \in \mathbb{C}$, and $(a + b)c =$

$ac + ab$, $a(b + c) = ab + ac$—such that $a(bc) = (ab)c$. An algebra with a norm (as a vector space) which is also sub-multiplicative, i.e. $\|ab\| \leqslant \|a\|\|b\|$, is called a *normed algebra*. A normed algebra which is complete is called a *Banach algebra*. An *involutive algebra* or a **-algebra* is an algebra $\mathcal{A}$ together with a conjugate-linear—i.e. $(za)^* = \overline{z}a^*$ for $z \in \mathbb{C}$, and $(a + b)^* = a^* + b^*$—map $\mathcal{A} \mapsto \mathcal{A}$ denoted by $a \mapsto a^*$, such that $a^{**} = a$ and $(ab)^* = b^*a^*$. A *Banach *-algebra* is a complete, normed, involutive algebra such that $\|a^*\| = \|a\|$. A C*-algebra is then defined as follows, e.g. [47, p. 36].

Definition 9. A C**-algebra* is a Banach *-algebra such that:

$$\|a^*a\| = \|a\|^2.$$

A simple example of a C*-algebra is the set $\mathcal{M}(n)$ of complex, finite dimensional $n \times n$ matrices. The scalar multiplication, addition and algebra product are the usual ones for matrices. The unique C*-norm of $a \in \mathcal{M}_n$ is given by the square root of the *spectral radius*—i.e. the largest eigenvalue—of a^*a: $\|a\|^2 = \rho(a^*a)$.

Other examples of C*-algebras include $\mathbb{C}$ the complex numbers, $C(X)$ the algebra of continuous functions on a compact space X with pointwise operations and $\mathcal{B}(\mathcal{H})$ the algebra of bounded linear operators on Hilbert spaces $\mathcal{H}$. In fact C*-algebras are all isomorphic to a sub-algebra of $\mathcal{B}(\mathcal{H})$ (e.g. [27, Theorems 2.2.1, 5.4.1]).

Proposition 10 (*Gelfand–Naimark*). *Any* C**-algebra is isometrically *-isomorphic to a* C**-subalgebra of some* $\mathcal{B}(\mathcal{H})$*, i.e.* C**-algebra of bounded, linear operators on a Hilbert space* $\mathcal{H}$*. If the* C**-algebra is separable then* $\mathcal{H}$ *can be taken to be separable*.

All infinite dimensional, separable C*-algebras can therefore be represented as C*-subalgebras of $\mathcal{B}(\ell^2)$. It is common to distinguish between *abstract C*-algebras* which we denote by $\mathcal{A}$, $\mathcal{B}$, etc. with elements $a, b, \ldots \in \mathcal{A}$ and *concrete C*-algebras*, i.e. C*-algebras which are given as C*-subalgebras of some $\mathcal{B}(\mathcal{H})$ and whose elements are linear bounded operators denoted by $\mathbf{A}, \mathbf{B}, \ldots \in \mathcal{B}(\mathcal{H})$.

The C*-algebraic setting allows the investigation of properties of linear operators independently of their concrete representation. For example, one can use an abstract characterisation to define certain types of operators, such as an (orthogonal) projection operator $\mathbf{P}$ which has to fulfill the conditions $\mathbf{P}^2 = \mathbf{P}$ and $\mathbf{P}^* = \mathbf{P}$.

Although C*-algebras have a unique C*-norm, there are several important (in the infinite dimensional case non-equivalent) topologies on the concrete C*-algebra $\mathcal{B}(\ell^2)$, e.g. [20, Section I.6], in particular:

norm or uniform topology: a sequence $(\mathbf{A}_n)_n$ in $\mathcal{B}(\ell^2)$ converges *uniformly* if there exists an operator $\mathbf{A} \in \mathcal{B}(\ell^2)$ such that $\lim_{n\to\infty} \|\mathbf{A}_n - \mathbf{A}\| = 0$.

strong operator topology: a sequence $(\mathbf{A}_n)_n$ in $\mathcal{B}(\ell^2)$ converges *strongly* if there exists an $\mathbf{A} \in \mathcal{B}(\ell^2)$ such that for all $x \in \ell^2$: $\lim_{n\to\infty} \|\mathbf{A}_n x - \mathbf{A}x\| = 0$.

We write $\lim \mathbf{A}_n$ for the uniform limit and s-lim $\mathbf{A}_n$ for the strong limit. The strong operator topology is weaker than the uniform or norm topology, i.e. convergence in the norm implies convergence in the strong topology but not vice versa.

3.4. Representation of probabilistic transition systems

We now return to the issue of how we will represent probabilistic transition systems on at most countably infinite state spaces.

Our aim is to establish whether transition relations for generative and reactive PTS's are "well-behaved", i.e. if they are represented by bounded linear operators on $\ell^2(S)$. We will not address the general problem of when a transition relation can be represented by a bounded operator but only aim to establish a simple criterion which guarantees that a given transition relation corresponds to an operator in $\mathcal{B}(\ell^2(S))$.

Definition 2 implies that for both generative and reactive PTS if we fix a $\alpha \in A$ the relation $\xrightarrow{\alpha}$ is a partial function $S \hookrightarrow \mathrm{SDist}(S)$. In particular, while for reactive systems this function always results in a distribution whenever is defined, for generative systems it gives in general a sub-probability distribution. We now show that for PTS satisfying a certain condition, relations $\xrightarrow{\alpha}$ can be represented by bounded linear operators.

For a state s in a generative or reactive PTS $(S, A, \longrightarrow, \pi_0)$ we denote by out-deg(s) the number of *successors* of s, i.e. the cardinality of the set:

$$\{t \in S \mid \exists \alpha \in A \text{ and } p \neq 0 : s \xrightarrow{p:\alpha} t\}$$

and by in-deg(s) the number of *predecessors* of s, i.e. the cardinality of the set:

$$\{t \in S \mid \exists \alpha \in A \text{ and } p \neq 0 : t \xrightarrow{p:\alpha} s\}.$$

Proposition 11. *Let S be a countable set and $\longrightarrow: S \hookrightarrow SDist(S)$ such that* $\sup_{s\in S}$ in-deg$(s) < \infty$ *and* $\sup_{s\in S}$ out-deg$(s) < \infty$. *Then the matrix $\mathbf{M}_{\longrightarrow}$ defines a bounded linear operator $\mathbf{M}(\longrightarrow) \in \mathcal{B}(\ell^2(S))$.*

Proof. In the following we will denote $\mathbf{M}_{\longrightarrow}$ by $\mathbf{M}$. We show that for all $v = (v_s)_{s\in S} \in \ell^2(S))$ such that $\|v\|_2 = 1$, we have $\|\mathbf{M}(v)\|_2 < \infty$. We have that

$$\|\mathbf{M}(v)\|_2^2 = \sum_{j=1}^{\infty}\left(\sum_{i=1}^{\infty}\mathbf{M}_{ij}v_i\right)^2.$$

Let $m = \sup_{s\in S}$ in-deg(s) and $n = \sup_{s\in S}$ out-deg(s). This means that in each column i of $\mathbf{M}$ there are at most $m(i) \leqslant m$ non-zero entries $\mathbf{M}_{f_1(i)i}, \mathbf{M}_{f_2(i)i}, \ldots, \mathbf{M}_{f_{m(i)}i}$. The functions $f_1, \ldots, f_m$ are functions picking out the non-zero entries in each column i in decreasing order, i.e. we assume that $v_{f_1(i)} \geqslant v_{f_2(i)} \geqslant \cdots \geqslant v_{f_{m(i)}(i)}$. Since $\mathbf{M}_{ij} \leqslant 1$ for all i, j, we get

$$\|\mathbf{M}(v)\|_2^2 = \sum_{j=1}^{\infty}\left(\sum_{i=1}^{m(j)}\mathbf{M}_{f_i(j)j}v_{f_i(j)}\right)^2 \leqslant \sum_{j=1}^{\infty}\left(\sum_{i=1}^{m(j)}v_{f_i(j)}\right)^2 \leqslant \sum_{j=1}^{\infty}(mv_{f_1(j)})^2.$$

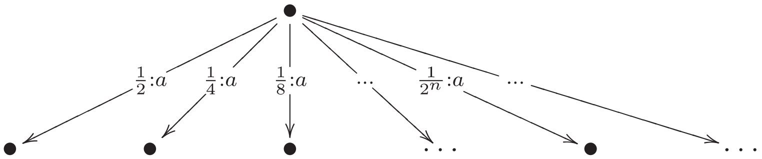
Since $\sup_{s\in S}$ out-deg $= n < \infty$, we have that for every row k the number of i's such that $f_1(i) = k$ cannot be greater than n. We therefore have

$$\|\mathbf{M}(v)\|_2^2 \leqslant m^2 \sum_{j=1}^{\infty} v_{f_1(j)}^2 \leqslant m^2 n \sum_{k=1}^{\infty} v_k^2 = nm^2 \|v\|_2^2$$

Therefore, $\|\mathbf{M}\|_2 = \sup_{\|v\|_2=1} \|\mathbf{M}(v)\|_2 \leqslant nm^2 < \infty$. □

This result is closely related to a well-known theorem regarding the so-called adjacency operator in the algebraic theory of *non-oriented* infinite graphs, e.g. [46, Theorem 3.1] or [68].

Note that the condition in Proposition 11 is indeed only a sufficient condition. There are simple infinitely branching PTS's which also give rise to bounded linear operators on $\ell^2(S)$, for example:



For computational purposes, infinite dimensional matrices, even when they represent a bounded linear operator in $\mathcal{B}(\ell^2(S))$ are anything but easy to handle. However, it is possible to define an *approximating sequence* for an operator $\mathbf{M} \in \mathcal{B}(\ell^2(S))$ as a sequence of finite dimensional approximations.

Given an operator $\mathbf{M} \in \mathcal{B}(\ell^2)$, consider a sequence of (orthogonal) projections $\mathbf{P}_n : \ell^2 \to \ell^2$ onto the first n coordinates of ℓ^2, that is operators such that $\mathbf{P}_n^2 = \mathbf{P}_n = \mathbf{P}_n^*$. We call $\mathbf{M}_n = \mathbf{P}_n\mathbf{M}\mathbf{P}_n$ a *finite section* of $\mathbf{M}$. It corresponds effectively to taking the $n \times n$ sub-matrix in the upper left corner of the matrix representing $\mathbf{M}$. The sequence $(\mathbf{M}_n)_n$ is an *approximating sequence* for $\mathbf{M}$ in the sense that $\mathbf{M}$ is the strong limit of this sequence, i.e. $\mathbf{M} = \text{s-lim}\, \mathbf{M}_n = \text{s-lim}_{n\to\infty} \mathbf{P}_n\mathbf{M}\mathbf{P}_n$ (see e.g. [9, Section 2.1]. This so called *finite section method* plays an important role in the numerical analysis of general operators. We will adopt this method in the case of PTS's with countable infinite state spaces.

Knowing that we can represent the *partial transition relations* $\xrightarrow{\alpha}$ of generative and reactive PTS by bounded linear operators on $\ell^2(S)$ we can now define the representation of a PTS.

Definition 12. Given a (generative or reactive) PTS $p = (S, A, \longrightarrow, \pi_0)$, we define its *matrix* or *operator representation* $(\mathbf{M}(X), \mathbf{M}(\pi_0))$ as the direct sum of the operator representations of the transition relations $\xrightarrow{\alpha}$ for each $\alpha \in A$:

$$\mathbf{M}(p) = \bigoplus_{\alpha\in A} \mathbf{M}(\xrightarrow{\alpha}),$$

and $|A|$ copies of the vector π_0 representing π_0: $\mathbf{M}(\pi_0) = \bigoplus_{\alpha\in A} \pi_0$.

We recall that for a set $\{\mathbf{M}_i\}_{i=1}^k$ of $n_i \times m_i$ matrices, the *direct sum* of these matrices is defined by the $(\sum_{i=1}^k n_i) \times (\sum_{i=1}^k m_i)$ matrix:

$$\mathbf{M} = \bigoplus_i \mathbf{M}_i = \begin{pmatrix} \mathbf{M}_1 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{M}_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{M}_k \end{pmatrix}.$$

This definition extends in the obvious way to countable infinite matrices, and it is the case that if the $\mathbf{M}_i$ represent bounded linear operators on some space $\ell^2(S_i)$ then $\bigoplus_i \mathbf{M}_i$ represents a bounded linear operator on $\ell^2(S_1 \times S_2 \times \cdots S_k)$.

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

3.4.1. Examples

Example 13. Consider the simple finite PTS A in Fig. 3. The matrix representation of this PTS is given by

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

Example 14. We can also represent an infinite PTS as a bounded linear operator. Consider for example the PTS B in Fig. 3. This infinite process requires an infinite dimensional matrix, i.e. an operator, to describe it. Utilising the finite section method we can approximate this

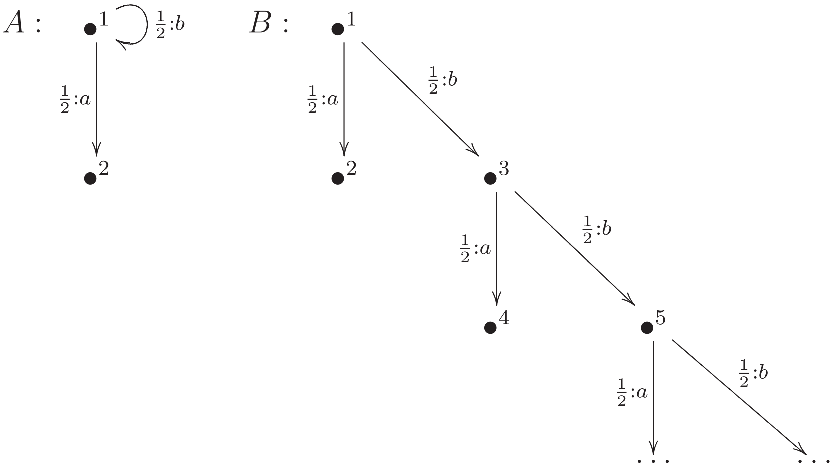


Fig. 3. Two probabilistic transition system.

operator via a sequence of finite dimensional operators, i.e. matrices in $\mathcal{M}(2n)$ of the form:

$$\left(\mathbf{M}_a^{2n}(B)\right)_{ij} = \begin{cases} 1/2 & \text{for } i = 2k-1 \ \wedge \ j = 2k, \\ & \text{with } k = 1 \ldots n, \\ 0 & \text{otherwise,} \end{cases}$$

$$\left(\mathbf{M}_b^{2n}(B)\right)_{ij} = \begin{cases} 1/2 & \text{for } i = 2k-1 \ \wedge \ j = 2k+1, \\ & \text{with } k = 1 \ldots n-1 \text{ and,} \\ & \text{for } i = 2n-1 \ \wedge \ j = 2n-1, \\ 0 & \text{otherwise.} \end{cases}$$

Then we can represent the infinite PTS B by the strong limit of this sequence,

$$\mathbf{M}(B) = \text{s-}\lim_{n\to\infty}(\mathbf{M}_a^{2n}(B) \oplus \mathbf{M}_b^{2n}(B)).$$

3.4.2. Properties of PTS representations

We recall that a matrix is called *stochastic* if the elements of every row sum up to 1; it is called *sub-stochastic* if this sum is less than or equal to 1.

The matrix representation of reactive systems will always lead to a direct sum of a special kind of sub-stochastic matrices. More precisely, for every action α the corresponding factor $\mathbf{M}_\alpha$ in the direct sum is such that the sum on the sth row is 1 if the state s is not terminal and 0 otherwise.

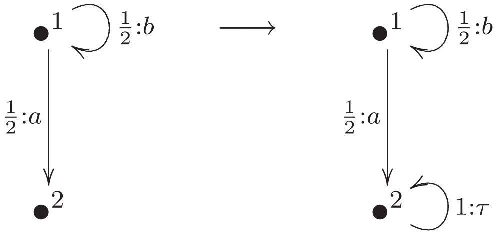
For generative systems the factors in the direct sum $\bigoplus_\alpha \mathbf{M}_\alpha$ are sub-stochastic matrices. However, as one intuitively expects, the sum of all factors always results in a matrix $\sum_\alpha \mathbf{M}_\alpha$ which is stochastic but for the terminal states. This is due to the fact that the combined probabilities for all actions leaving a non-terminal state s define a distribution in $Dist(A \times S)$ which corresponds to the sth row in $\sum_\alpha \mathbf{M}_\alpha$.

Consider the simple generative process in Example 13. The sum

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

is not row-normalised in the second row, corresponding to the terminal state 2. We can nevertheless overcome this technical difficulty and associate to a generative process a stochastic matrix. One way is to introduce a silent τ transition on terminal states.

Example 15. Consider again Example 13 and extend the execution tree as follows:



The extended linear operator representation of A does now correspond to a stochastic matrix:

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

In the following, we will always assume such a *stochastic extension* for terminating processes, even if we omit to mention explicitly the silent moves on the terminal states.

4. Probabilistic abstract interpretation

In Section 5 we will present a technique for defining approximate versions of process semantics and ultimately of security properties which makes use of the framework of *probabilistic abstract interpretation* (PAI). This was introduced in [56,57] as a probabilistic version of the classical *abstract interpretation* (AI) framework by Cousot and Cousot [16,17].

Classical abstract interpretation provides general techniques for the analysis of programs which are based on the construction of *safe* approximations of the concrete semantics of programs via the (order-theoretic) notion of a *Galois connection* [18,49].

Probabilistic abstract interpretation re-casts these techniques in a probabilistic setting where linear spaces replace the classical order-theoretic domains, and the notion of the so-called *Moore–Penrose pseudo-inverse* of a linear operator replaces the classical notion of a Galois connection. The abstractions we get this way are *close* approximations of the concrete semantics. Thus, closeness is a quantitative replacement for classical safety which does not require any approximation ordering.

The definition of a probabilistic abstract interpretation is given in terms of *probabilistic domains*. A probabilistic domain is essentially a space which represents the distributions $\mathit{Dist}(S)$ on the state space S of a PTS, i.e. in our setting the Hilbert space $\ell^2(S)$. For finite state spaces we can identify $\mathcal{V}(S) \simeq \ell^2(S)$.

Definition 16. Let $\mathcal{C}$ and $\mathcal{D}$ be two probabilistic domains. A *probabilistic abstract interpretation* is a pair of bounded linear operators $\mathbf{A} : \mathcal{C} \to \mathcal{D}$ and $\mathbf{G} : \mathcal{D} \to \mathcal{C}$, 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.

A simple method for constructing a probabilistic abstract interpretation which we will use in this paper is as follows: Given a linear operator Φ on some Hilbert 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 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 will now introduce in some more detail the central notion of Moore–Penrose pseudo-inverse.

4.1. Moore–Penrose pseudo-inverse

For an abstract C*-algebra we can define the notion of a Moore–Penrose pseudo-inverse purely algebraically [9, Section 4.7] (see also [12, Definition 1.1.1] and [24, 8.43]). This is sufficient for the finite dimensional setting, while for dealing with the infinite dimensional case we will need some topological considerations which we will use for a more concrete definition.

Definition 17. An element $a \in \mathcal{A}$ in a C*-algebra $\mathcal{A}$ is said to be *Moore–Penrose invertible* if there exists an element $b \in \mathcal{A}$ such that:
(i) $aba = a$,
(ii) $bab = b$,
(iii) $(ab)^* = ab$,
(iv) $(ba)^* = ba$.

If an element $a \in \mathcal{A}$ is Moore–Penrose invertible then there exists a unique element $b = a^\dagger$, the *Moore–Penrose pseudo-inverse* of a, which fulfills the above conditions [9, Proposition 4.20].

An alternative but equivalent definition for concrete C*-algebras is given in [24, 8.43] (see also [12, Definition 1.1.2]).

Definition 18. Let $\mathcal{C}$ and $\mathcal{D}$ be two Hilbert spaces and $\mathbf{A} : \mathcal{C} \mapsto \mathcal{D}$ a bounded linear map between them. A bounded linear map $\mathbf{A}^\dagger = \mathbf{G} : \mathcal{D} \mapsto \mathcal{C}$ is the *Moore–Penrose pseudo-inverse* of $\mathbf{A}$ iff
(i) $\mathbf{A} \circ \mathbf{G} = \mathbf{P}_A$, and
(ii) $\mathbf{G} \circ \mathbf{A} = \mathbf{P}_G$,
where $\mathbf{P}_A$ and $\mathbf{P}_G$ denote orthogonal projections onto the ranges of $\mathbf{A}$ and $\mathbf{G}$.

For finite dimensional C*-algebras—in particular for matrix algebras $\mathcal{M}(n)$—every operator is Moore–Penrose pseudo-invertible [6,9,12,24].

For operator algebras over infinite dimensional Hilbert spaces things are a bit more complicated. In the case of concrete C*-algebras, i.e. of $\mathbf{A} \in \mathcal{B}(\mathcal{H})$, the answer is given by the following result which also states how we can “construct” the Moore–Penrose pseudo-inverse [9, Theorem 4.24].

Proposition 19. *An operator $\mathbf{A} \in \mathcal{B}(\mathcal{H})$ is Moore–Penrose invertible if and only if it is* normally solvable, *i.e. the range $\{\mathbf{A}x | x \in \mathcal{H}\}$ is closed. In this case $\mathbf{A}^*\mathbf{A} + \mathbf{P}$—with $\mathbf{P}$ the orthogonal projection of $\mathcal{H}$ onto the kernel of $\mathbf{A}$, i.e. onto $\{x \in \mathcal{H} | \mathbf{A}x = o\}$—is invertible and*

$$\mathbf{A}^\dagger = (\mathbf{A}^*\mathbf{A} + \mathbf{P})^{-1}\mathbf{A}^*.$$

It is easy to see that if the range of an operator is finite dimensional then it is normally solvable.

For the finite dimensional case, various algorithms are known for the construction of the Moore–Penrose pseudo-inverse [12]. A general technique for computing the Moore–

Penrose pseudo-inverse of infinite operators is via finite sections. For an operator $\mathbf{A}$ with an approximating sequence $(\mathbf{A}_n)_n$ we can construct the Moore–Penrose pseudo-inverse as established by the following proposition [9, Corollary 4.34].

Proposition 20. *Let $\mathcal{H}$ be a separable Hilbert space, $\mathbf{A} \in \mathcal{B}(\mathcal{H})$ and $\mathcal{A}_n$ a sequence of finite dimensional operators $\mathcal{A}_n \in \mathcal{M}(n)$ with $\sup_n \|\mathcal{A}\| < \infty$ and such that $\mathbf{A}_n \to \mathbf{A}$ and $\mathbf{A}_n^* \to \mathbf{A}^*$ strongly. Then $\mathbf{A}$ is normally solvable and $\mathbf{A}_n^\dagger \to \mathbf{A}^\dagger$ strongly.*

In other words, if we can approximate $\mathbf{A}$ by a sequence $(\mathbf{A}_n)_n$ and the sequence $(\mathbf{A}_n^\dagger)_n$ of Moore–Penrose pseudo-inverse converges in the strong operator topology then $\mathbf{A}^\dagger$ exists and is identical to the limit of $(\mathbf{A}_n^\dagger)_n$.

4.2. Special classes of abstraction operators

In this section we introduce the definition and the properties of some particular operators which we will use in Section 5 to define different abstractions of the PTS semantics into various process equivalences. We will also use these special operators to define approximate versions of the process equivalences and their corresponding confinement properties.

4.2.1. Permutation operators

The first class of operators we consider represents very simple abstractions consisting of the permutation of the system's states.

Definition 21. An $n \times n$-matrix $\mathbf{S}$ is called a *permutation matrix* if there exists a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that

$$\mathbf{S}_{ij} = \begin{cases} 1 & \text{if } j = \pi(i) \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $\mathbf{S}$ is the matrix representation of a relation on $\{1, \ldots, n\}$ which is a bijection. This notion can be extended to infinite structures as follows.

Definition 22. A bounded linear operator $\mathbf{S} \in \mathcal{B}(\mathcal{H})$ on a Hilbert space is called a *permutation operator* iff there exists a sequence of permutation matrices $\mathbf{S}_n$ such that $\text{s-lim}\, \mathbf{S}_n = \mathbf{S}$ and $\text{s-lim}\, \mathbf{S}_n^* = \mathbf{S}^*$.

We denote by $\mathcal{S}(n)$ the set of all $n \times n$ permutation matrices and by $\mathcal{S}(\mathcal{H})$ the set of permutation operators on $\mathcal{H}$; obviously we have $\mathcal{S}(n) = \mathcal{S}(\mathbb{C}^n)$.

Proposition 23. *For any permutation matrix $\mathbf{S} \in \mathcal{S}(n)$ the following holds*:

$$\mathbf{S}^{-1} = \mathbf{S}^* = \mathbf{S}^{\mathrm{T}} = \mathbf{S}^\dagger,$$

i.e. inverse, adjoint, transpose, *and* pseudo-inverse *coincide*.

4.2.2. Classification operators

Definition 24. We call an $n \times m$-matrix $\mathbf{K}$ a *classification matrix* iff $\mathbf{K}$ represents a surjective function $\kappa : \{1, \ldots, n\} \to \{1, \ldots, m\}$, i.e.

$$\mathbf{K}_{ij} = \begin{cases} 1 & \text{if } j = \kappa(i) \\ 0 & \text{otherwise.} \end{cases}$$

Again we can generalise this notion to the infinite case.

Definition 25. A bounded linear operator $\mathbf{K} \in \mathcal{B}(\mathcal{H})$ on a Hilbert space is called a *classification operator* iff there exists a sequence of classification matrices $\mathbf{K}_n$ such that s-lim $\mathbf{K}_n = \mathbf{K}$ and s-lim $\mathbf{K}_n^* = \mathbf{K}^*$.

We denote by $\mathcal{C}(n, m)$ the set of all $n \times m$-classification matrices, and by $\mathcal{C}(\mathcal{H}_1, \mathcal{H}_2)$ the set of classification operators; again we have $\mathcal{C}(n, m) = \mathcal{C}(\mathbb{C}^n, \mathbb{C}^m)$.

Classification matrices are stochastic matrices corresponding to a particular type of abstraction which stems from an equivalence relation. For a finite set X we can show that there is a one-to-one correspondence between equivalence relations $\approx$ on X and classification operators on the vector space $\mathcal{V}(X)$.

Proposition 26. *Let X be a finite set. Then for every equivalence relation $\approx$ on X there exists a classification operator $\mathbf{K} \in \mathcal{C}(n, m) \simeq \mathcal{C}(\ell^2(X), \ell^2(X/_\approx))$ and vice versa.*

Proof. The characteristic map $\chi_\approx : X \mapsto X/_\approx$ which associates to each $x \in X$ its equivalence class $[x] \in X/_\approx$ is a surjective function and therefore has a matrix representation (as a relation $\chi_\approx \subseteq X \times X/\approx$) in $\mathcal{C}(n, m)$. Vice versa, by definition a classification matrix $\mathbf{K} \in \mathcal{C}(n, m)$ induces a partition (and therefore an equivalence relation) on the set of its row indices. □

In the infinite case we can show that:

Proposition 27. *Let X be a countable set and $\approx$ an equivalence relation on X such that $X/_\approx$ is finite. Then there exists a classification operator $\mathbf{K} \in \mathcal{C}(\ell^2(X), \ell^2(X/_\approx))$ which represents $\approx$.*

Proof. Firstly, we observe that $\mathbf{K}$ defines a $\infty \times n$ matrix. This maps every $x \in \ell^2(X)$ with $\|x\| = 1$ into a vector $\|\mathbf{K}(x)\| < \infty$. Thus we have $\mathbf{K} \in \mathcal{B}(\ell^2(X), \ell^2(X/_\approx))$.

Secondly, $\mathbf{K}$ is the strong limit of a sequence of finite dimensional classification matrices $\mathbf{K}_n$; to see this simply take an enumeration of X and $\mathbf{K}_n = \pi_n \mathbf{K} \pi_n$ (cf. finite section method [9]). □

Obviously, every permutation matrix is also a classification matrix: $\mathcal{S}(n) \subseteq \mathcal{C}(n, n)$. As a consequence, every permutation operator is a classification operator: $\mathcal{S}(\mathcal{H}) \subseteq \mathcal{C}(\mathcal{H}, \mathcal{H})$.

4.2.3. Moore–Penrose pseudo-inverse of classification operators

Although a classification operator **K** represents a classical function, i.e. corresponds to an (infinite) 0/1-matrix, the pseudo-inverse will in general not be an (infinite) 0/1-matrix. This is because it is *normalised*. The *normalisation* operation $\mathcal{N}$ is defined for a matrix **A** by

$$\mathcal{N}(\mathbf{A})_{ij} = \begin{cases} \dfrac{\mathbf{A}_{ij}}{a_j} & \text{if } a_j = \sum_i \mathbf{A}_{ij} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 28. *The pseudo-inverse of a classification operator* **K** *corresponds to its normalised transpose or adjoint*:

$$\mathbf{K}^{\dagger} = \mathcal{N}(\mathbf{K}^{\mathrm{T}}) = \mathcal{N}(\mathbf{K}^{*}).$$

Proof. Show by computation that $\mathcal{N}(\mathbf{K})$ fulfils the Moore–Penrose conditions of Definition 17 or Definition 18. □

4.2.4. Probabilistic abstract interpretation of stochastic matrices

For a stochastic matrix **M** and any abstraction **A** with Moore–Penrose pseudo-inverse **G** we can in general not guarantee that the abstract operator **GMA** induced by **A** is also a stochastic matrix.

Example 29. Consider the following stochastic matrix:

$$\mathbf{M} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 1 \end{pmatrix},$$

together with abstraction and concretisation maps represented by

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{G} = \mathbf{A}^{\dagger} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

A simple calculation shows that

$$\mathbf{GMA} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

which is *not* a stochastic matrix. Similarly, if we take

$$\mathbf{A} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{G} = \mathbf{A}^{\dagger} = \begin{pmatrix} \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

we get as induced operator the following 1 × 1 matrix:

$$\mathbf{GMA} = \left(\tfrac{8}{9}\right)$$

which again is not a stochastic matrix.

For classification matrices however, we can show the following.

Proposition 30. *For any stochastic matrix* $\mathbf{M}$ *and classification matrix* $\mathbf{K}$ *we have that* $\mathbf{K}^\dagger\mathbf{MK}$ *is again a stochastic matrix.*

Proof. By Proposition 28 we know that $\mathbf{K}^\dagger$ is a stochastic matrix as it is (row) normalised. The same is true for $\mathbf{K}$ (by definition) and $\mathbf{M}$ (by hypothesis). Thus $\mathbf{K}^\dagger\mathbf{MK}$ is stochastic since the product of stochastic matrices is stochastic. □

5. Approximating process equivalences

Several notion of process equivalences have been proposed in the literature on concurrency theory, each one defining a different process semantics. A comparative study of most of these semantics can be found in [28]. The purpose of this section is to present a technique for approximating process equivalences by using probabilities as numerical information for quantifying such an approximation. This provides us with a quantitative measure of the indistinguishability of the process behaviour (according to a given semantics), that is in a security setting a measure of their propensity to leak information. Therefore, for each semantics we are able to measure the confinement of a given system according to the notion of behavioural equivalence established by the given semantics. In order to numerically estimate such a measure we first re-formulate each process equivalence in terms of linear operators using the PAI framework introduced in Section 4. Then we use an appropriate notion of operator norm to calculate the closeness of two processes.

We illustrate this technique for three behavioural equivalences, namely tree equivalence, bisimulation and weak bisimulation, but the method can be extended to deal with all the other semantics in a similar way.

5.1. Graph isomorphism

To illustrate our basic strategy for approximating process equivalences let us first look at the strongest—in some sense too strong [28, Fig. 1]—notion of process equivalence, that is tree equivalence. Following [28, Definition 1.3] the graph associated to a process p of a labelled transition system with actions A is a directed graph rooted in p whose edges are labelled by elements in A. Two processes are *tree equivalent* if their associated graphs are isomorphic. Graph isomorphism is defined as follows (e.g. [28, Definitions 1.3,1.4], [30, p. 2], [25, p. 3]):

Definition 31. An *isomorphism* between directed graphs (V_1, E_1) and (V_2, E_2) is a bijection $\varphi : V_1 \mapsto V_2$ such that $\langle v, w\rangle \in E_1 \Leftrightarrow \langle\varphi(v), \varphi(w)\rangle \in E_2$.

In the usual way, we define the *adjacency operator* $\mathbf{A}(G)$ of a directed graph $G = (V, E)$ as an operator on $\ell^2(V)$ representing the edge-relation E [46]. Then the notion of isomorphism between (finite graphs) can be re-stated in terms of permutation matrices.

We have the following result [30, Lemma 8.8.1]:

Proposition 32. *Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two directed graphs on the same set of nodes V. Then G_1 and G_2 are isomorphic if and only if there is a permutation operator* $\mathbf{S}$ *such that the following holds*: $\mathbf{S}^{\mathrm{T}}\mathbf{A}(G_1)\mathbf{S} = \mathbf{A}(G_2)$.

By using these notions and the operator representation of (probabilistic) transition systems (cf. Definition 12) we can reformulate tree-equivalence of processes as follows.

Proposition 33. *Given the operator representations p and q of two probabilistic transition systems $p = (S, A, \longrightarrow, s_0)$ and $q = (S', A, \longrightarrow', s'_0)$ with $|S| = |S'|$, then p and q are tree-equivalent iff there exists $\mathbf{S} \in \mathcal{P}(\ell^2(S)) = \mathcal{P}(\ell^2(S'))$, such that:*

$$\mathbf{S}^{\mathrm{T}}\mathbf{M}(p)\mathbf{S} = \mathbf{M}(q),$$

i.e. for all $\alpha \in A$ we have $\mathbf{S}^{\mathrm{T}}\mathbf{M}(\overset{\alpha}{\longrightarrow})\mathbf{S} = \mathbf{M}(\overset{\alpha}{\longrightarrow}{}')$.

Therefore, tree equivalence of two systems p and q corresponds to the existence of an abstraction operator (the operator $\mathbf{S}$) which induces a probabilistic abstract interpretation p of q.

5.1.1. Approximate graph isomorphism

In the case where there is no $\mathbf{S}$ which satisfies the property in Proposition 33, i.e. p and q are definitely not isomorphic, we could still ask how close p and q are to being isomorphic. The most direct way to define a kind of "isomorphism defect" would be to look at the difference $\mathbf{M}(p) - \mathbf{M}(q)$ between the operators representing p and q and then measure in some way, e.g. using a norm, this difference.

Obviously, this is not the idea we are looking for: It is easy to see that the same graph—after enumerating its vertices in different ways—has different adjacency operators; it would thus have a non-zero "isomorphism defect" with itself. To remedy this we have to allow first for a reordering of vertices before we measure the difference between the operators representing two probabilistic transition systems. This is the underlying idea behind the following definition.

Definition 34. Let $p = (S, A, \longrightarrow, \pi_0)$ and $q = (S', A, \longrightarrow', \pi'_0)$ be probabilistic transition systems over the same set of actions A, and let $\mathbf{M}(p)$ and $\mathbf{M}(q)$ be their operator representations. We say that p and q are *ε-graph equivalent*, denoted by $p \sim_i^{\varepsilon} q$, iff

$$\inf_{\mathbf{S}\in\mathcal{P}} \|\mathbf{S}^{\mathrm{T}}\mathbf{M}(p)\mathbf{S} - \mathbf{M}(q)\| = \varepsilon$$

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

Note that, in the case of finite probabilistic transition systems, for $\varepsilon = 0$ we recover the original notion of (strict) graph equivalence, i.e. $\sim_i = \sim_i^0$.

Proposition 35. *An ε-isomorphism for $\varepsilon = 0$, i.e. $\sim_i^0$, of finite transition systems is an isomorphism.*

Proof. Observe that there are only finitely many $\mathbf{S} \in \mathcal{P}(n)$, $n < \infty$. Thus the inf can be replaced by min. That means that there exists a permutation operator $\mathbf{S} \in \mathcal{P}(n)$ such that $\|\mathbf{S}^{\mathrm{T}}\mathbf{M}(p)\mathbf{S} - \mathbf{M}(q)\| = 0$. The properties of the norm then imply that $\mathbf{S}^{\mathrm{T}}\mathbf{M}(p)\mathbf{S} - \mathbf{M}(q) = \mathbf{O}$, the null operator, i.e. $\mathbf{S}^{\mathrm{T}}\mathbf{M}(p)\mathbf{S} = \mathbf{M}(q)$. □

5.2. Bisimulation

The finest process equivalence after graph equivalence is bisimulation equivalence [28,45]. Bisimulation is a relation on processes, i.e. states of a labelled transition system. Alternatively, it can be seen as a relation between the *transition graphs* associated to the processes.

The classical notion of bisimulation equivalence for labelled transition systems can be stated as follows [28, Definition 12]:

Definition 36. A *bisimulation* is a binary relation $\sim_b$ on states of a labelled transition system satisfying for all $\alpha \in A$:

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

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 [28, Proposition 8.1].

The standard generalisation of this notion to probabilistic transition systems, i.e. *probabilistic bisimulation*, is due to [44, Definition 4], where it is defined for reactive systems.

Definition 37. A *probabilistic bisimulation* is an equivalence relation $\sim_b$ on states of a probabilistic transition system satisfying for all $\alpha \in A$:

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

The same definition can be given also for generative systems with the only difference that in this case π and ϱ are sub-probability distributions.

This definition is equivalent to the characterisation of probabilistic bisimulation given in [44] 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). The same button pressing interpretation can be given also in the case of generative systems but for the way experiments are performed: here the observer may attempt to depress more than one button at a time and it is the process which decides which action to react to according to a given probability distribution. In our security setting these 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.

Note that in the case of generative systems tests represent *passive* spies, in the sense that it is not possible for an observer to actively interfere in the process internal behaviour by deciding which action has to be chosen.

Definition 38. Let $T = (S, A, \to, \pi_0)$ be a probabilistic transition system and let $T(p)$ and $T(q)$, with $p, q \in S$, represent two processes in a probabilistic language modelled by *T*. Then we say that *p* and *q* are *probabilistically confined* iff they are probabilistic bisimilar.

It is easy to see that a probabilistic bisimulation equivalence $\sim$ on a PTS $T = (S, A, \to, \pi_0)$ defines a probabilistic abstract interpretation of *T*. In fact, by Proposition 27, there is a classification operator $\mathbf{K} \in \mathcal{C}(\ell^2(S), \ell^2(S/\sim))$, which represents $\sim$. If $\mathbf{M}(T)$ is the operator representation of *T* then $\mathbf{K}^\dagger\mathbf{M}(T)\mathbf{K}$ is the abstract operator induced by $\mathbf{K}$. Intuitively, this is an operator 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 operator representations $\mathbf{M}(p)$ and $\mathbf{M}(q)$. 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*:

Proposition 39. *Given the operator representation* $\mathbf{M}(p)$ *and* $\mathbf{M}(q)$ *of two probabilistic processes p and q, then p and q are probabilistic bisimilar iff there exists a* $\mathbf{K}_p \in \mathcal{C}(\ell^2(R_p), \ell^2(S))$ *and* $\mathbf{K}_q \in \mathcal{C}(\ell^2(R_q), \ell^2(S))$ *for some set S such that*

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

Proof. We assume in the following that there is an enumeration of the processes $\{p_i\}_{i=1}^{n_p} = R_p$ and $\{q_j\}_{j=1}^{n_q} = R_q$. Therefore, $\mathbf{M}_\alpha(p)$ is a $n_p \times n_p$ matrix, and i.e. $\mathbf{M}_\alpha(q)$ is a $n_q \times n_q$ matrix, for each $\alpha \in A$.

(only if) Suppose that there is a probabilistic bisimulation relation $\sim$ between processes *p* and *q*. This relation determines a partition on R_p as well as on R_q such that $|R_p/\sim| = |R_q/\sim|$. Define *S* as the set of all the $\sim$-equivalence classes with a given enumeration $\{[r_k]\}_{k=1}^m = R_p/\sim = R_q/\sim$, with $m = |S|$.

Let us define the two matrices

$$(\mathbf{K}_p)_{ik} = \begin{cases} 1 & \text{if } p_i \in [r_k], \\ 0 & \text{otherwise,} \end{cases}$$

for all $p_i \in R_p$,$[r_k] \in S$, and

$$(\mathbf{K}_q)_{jk} = \begin{cases} 1 & \text{if } q_j \in [r_k], \\ 0 & \text{otherwise,} \end{cases}$$

for all $q_j \in R_q$, $[r_k] \in S$.

We have that $\mathbf{K}_p \in \mathcal{C}(\ell^2(R_p), \ell^2(S))$ and $\mathbf{K}_q \in \mathcal{C}(\ell^2(R_q), \ell^2(S))$. We now show that for every base vector $x_k \in \ell^2(S)$ representing an equivalence class $[r_k]$ the following holds:

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

Then by linearity we can conclude that the above equation holds for all $x \in \ell^2(S)$.

- $x_k \mathbf{K}_p^\dagger$ and $x_k \mathbf{K}_q^\dagger$ are two (row) vectors in $\mathcal{V}(R_p)$ and $\mathcal{V}(R_q)$ respectively which represent uniform distributions on all those processes in R_p and R_q belonging to the equivalence class $[r_k]$:

$$(x_k \mathbf{K}_p^\dagger)_i = \begin{cases} \dfrac{1}{n_p^k} & \text{if } p_i \in [r_k], \\ 0 & \text{otherwise,} \end{cases}$$

$$(x_k \mathbf{K}_p^\dagger)_j = \begin{cases} \dfrac{1}{n_q^k} & \text{if } q_i \in [r_k], \\ 0 & \text{otherwise,} \end{cases}$$

where n_p^k and n_q^k represent the number of processes in R_p and R_q belonging to $[r_k]$.

- The application of $\mathbf{M}_\alpha(p)$ and $\mathbf{M}_\alpha(q)$ (for each $\alpha \in A$) to these vectors gives us a distribution on those processes in R_p and R_q which can be reached from a state belonging to the equivalence class $[r_k]$ in one step:

$$(x_k \mathbf{K}_p^\dagger \mathbf{M}_\alpha(p))_{i'} = \sum_{\substack{p_i \in [r_k] \\ p_i \xrightarrow{\alpha}_\pi p_{i'}}} \frac{\pi(p_{i'})}{n_p^k},$$

$$(x_k \mathbf{K}_q^\dagger \mathbf{M}_\alpha(q))_{j'} = \sum_{\substack{q_j \in [r_k] \\ q_j \xrightarrow{\alpha}_\pi q_{j'}}} \frac{\pi(q_{j'})}{n_q^k}.$$

- The classification of these vectors via $\mathbf{K}_p \in \mathcal{C}(n_p, m)$ and $\mathbf{K}_q \in \mathcal{C}(n_q, m)$ gives us the distributions over equivalence classes:

$$(x_k \mathbf{K}_p^\dagger \mathbf{M}_\alpha(p) \mathbf{K}_p)_k = \sum_{\substack{p_i, p_{i'} \in [r_k] \\ p_i \xrightarrow{\alpha}_\pi p_{i'}}} \frac{\pi(p_{i'})}{n_p^k},$$

and

$$(x_k \mathbf{K}_q^\dagger \mathbf{M}_\alpha(q) \mathbf{K}_q)_k = \sum_{\substack{q_j, q_{j'} \in [r_k] \\ q_j \xrightarrow{\alpha}_\pi q_{j'}}} \frac{\pi(q_{j'})}{n_q^k},$$

which must be the same since by hypothesis $\sim$ is a probabilistic bisimulation.

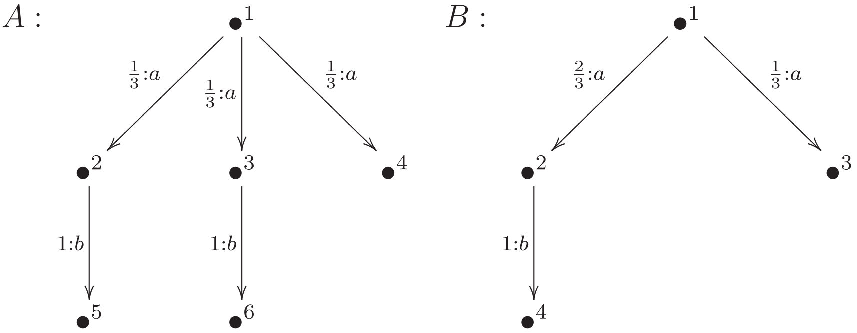


Fig. 4. Two reactive probabilistic transition systems.

(if) Suppose that we have $\mathbf{K}_p \in \mathcal{C}(n_p, m)$ and $\mathbf{K}_q \in \mathcal{C}(n_q, m)$ such that $\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p = \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$. Define a relation $\sim_K$ between processes in R_p and processes in R_q as follows:

$$p_i \sim_K q_j \quad \text{iff } p_i\mathbf{K}_p = q_j\mathbf{K}_q.$$

In order to show that $\sim_K$ is a probabilistic bisimulation we have to show that $\sim_K$ satisfies for all $\alpha \in A$, $p \in R_p$ and $q \in R_q$:

$$p \sim_K q \quad \text{and} \quad p \xrightarrow{\alpha} \pi \Rightarrow q \xrightarrow{\alpha} \varrho \quad \text{and} \quad \pi \sim_K \varrho$$

or, equivalently

$$[p]_K = [q]_K \quad \text{and} \quad p \xrightarrow{\alpha} \pi \Rightarrow q \xrightarrow{\alpha} \varrho \quad \text{and} \quad \pi \sim_K \varrho.$$

We will use the notation $[\pi_p^\alpha]_K = [\varrho_q^\alpha]_K$ to indicate the condition above. For processes $p_i \in R_p$ and $q_j \in R_q$ belonging to the same equivalence class $[r_k] = [p_i]_K = [q_j]_K$ we know that

$$p_i\mathbf{K}_p = q_j\mathbf{K}_q.$$

Since by hypothesis $\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p = \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$, we then conclude that $[\pi_p^\alpha]_K = [\varrho_q^\alpha]_K$. Thus $\sim_K$ is a probabilistic bisimulation. □

Corollary 40. *Let $\mathbf{M}(p)$ and $\mathbf{M}(q)$ be the matrix representations of two processes p and q. If p and q are probabilistic bisimilar then there exists a PTS x which is the probabilistic abstract interpretation of both p and q.*

Proof. Consider the PTS with states in $R_p \cup R_q$ and the classification operator associated to the relation $\sim_K$ constructed in the proof of Proposition 39. □

Example 41. Consider the two reactive processes A and B in Fig. 4 taken from [44, Fig. 4]. The corresponding matrices are:

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

and

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

Consider the classification operators $\mathbf{K}_A$ and $\mathbf{K}_B$, and their pseudo-inverses defined by

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

and $\mathbf{K}_B$ and $\mathbf{K}_B^\dagger$ are simply 4 × 4 identity matrices. We then get

$$\mathbf{K}_A^\dagger \mathbf{M}_a(A) \mathbf{K}_A = \mathbf{M}_a(B)$$
$$\mathbf{K}_A^\dagger \mathbf{M}_b(A) \mathbf{K}_A = \mathbf{M}_b(B)$$

which shows that A and B are probabilistic bisimilar.

By Corollary 40 we can therefore construct a system which abstracts both A and B. Since A and B are probabilistic bisimilar we can define an equivalence relation on the union T of the two PTS's in Fig. 4 which is compatible with $\mathbf{K}_A$ and $\mathbf{K}_A$. This is given by the classification operator

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_A \\ \mathbf{K}_B \end{pmatrix}.$$

We can then see that $\mathbf{K}^\dagger \mathbf{M}(T) \mathbf{K}$ is a system which abstracts both A and B. In fact, given that $\mathbf{M}(T) = \mathbf{M}(A) \oplus \mathbf{M}(B)$, and that

$$\mathbf{K}^\dagger = \begin{pmatrix} \frac{|A|}{|A| + |B|} \mathbf{K}_A^\dagger & \frac{|B|}{|A| + |B|} \mathbf{K}_B^\dagger \end{pmatrix},$$

where $|A|$ ($|B|$) is the cardinality of the set of states in the PTS for A (B), we have that

$$\begin{aligned}
\mathbf{K}^\dagger\mathbf{M}(T)\mathbf{K} &= \mathbf{K}^\dagger(\mathbf{M}(A)\oplus\mathbf{M}(B))\mathbf{K}\\
&= \mathbf{K}^\dagger(\mathbf{M}(A)\oplus\mathbf{O})\mathbf{K}+\mathbf{K}^\dagger(\mathbf{O}\oplus\mathbf{M}(B))\mathbf{K}\\
&= \left(\frac{|A|}{|A|+|B|}\mathbf{K}_A^\dagger\mathbf{M}(A)\mathbf{K}_A+\frac{|B|}{|A|+|B|}\mathbf{K}_B^\dagger\mathbf{O}\mathbf{K}_B\right)\\
&\quad+\left(\frac{|A|}{|A|+|B|}\mathbf{K}_A^\dagger\mathbf{O}\mathbf{K}_A+\frac{|B|}{|A|+|B|}\mathbf{K}_B^\dagger\mathbf{M}(B)\mathbf{K}_B\right)\\
&= \frac{|A|+|B|}{|A|+|B|}\mathbf{K}_B^\dagger\mathbf{M}(B)\mathbf{K}_B=\frac{|A|+|B|}{|A|+|B|}\mathbf{K}_A^\dagger\mathbf{M}(A)\mathbf{K}_A\\
&= \mathbf{K}_B^\dagger\mathbf{M}(B)\mathbf{K}_B=\mathbf{K}_A^\dagger\mathbf{M}(A)\mathbf{K}_A,
\end{aligned}$$

where we denote by $\mathbf{O}$ the null matrix of the appropriate dimensions.

Example 42. It is easy to see that the two generative processes A and B in Example 13 and 14 are probabilistic bisimilar. To show that these processes are bisimilar we construct an operator $\mathbf{K}$ such that:

$$\begin{aligned}
\mathbf{M}_a(A) &= \mathbf{K}^\dagger\mathbf{M}_a(B)\mathbf{K},\\
\mathbf{M}_b(A) &= \mathbf{K}^\dagger\mathbf{M}_b(B)\mathbf{K},
\end{aligned}$$

and then we simply take $\mathbf{K}_A=\mathbf{I}$ and $\mathbf{K}_B=\mathbf{K}$.

We need to construct again an infinite operator $\mathbf{K}$ as a sequence of $2\times 2n$-matrices:

$$\left(\mathbf{K}^{2n}\right)_{ij}=\begin{cases}1 & \text{for } i=2k-1\ \wedge\ j=1 \text{ and},\\ & \text{for } i=2k\ \wedge\ j=2 \text{ with } k=1\ldots n,\\ 0 & \text{otherwise,}\end{cases}$$

with their $2n\times 2$ pseudo-inverses:

$$\left(\mathbf{K}^{2n}\right)^\dagger_{ij}=\begin{cases}\frac{1}{n} & j=2k-1\ \wedge\ i=1 \text{ and},\\ & j=2k\ \wedge\ i=2 \text{ with } k=1\ldots n,\\ 0 & \text{otherwise.}\end{cases}$$

We therefore have that

$$\mathbf{K}=\text{s-}\lim_{n\to\infty}\mathbf{K}^{2n}\quad\text{and}\quad\mathbf{K}^\dagger=\text{s-}\lim_{n\to\infty}(\mathbf{K}^{2n})^\dagger$$

and from Example 14 we know that

$$\mathbf{M}(B)=\text{s-}\lim_{n\to\infty}(\mathbf{M}_a^{2n}(B)\oplus\mathbf{M}_b^{2n}(B)).$$

The operator multiplication is in general not strongly continuous, but it is if one of the factors is restricted to a bounded set, see e.g. [40, 2.5.10] or [20, I.6]. Clearly, all the $\mathbf{K}^{2n}$

and $(\mathbf{K}^{2n})^\dagger$ are from a bounded set, therefore we have

$$\begin{aligned}\mathbf{K}^\dagger \cdot \mathbf{M}(B) \cdot \mathbf{K} &= (\text{s-lim}(\mathbf{K}^{2n})^\dagger) \cdot (\text{s-lim}\,\mathbf{M}^{2n}(B)) \cdot (\text{s-lim}\,\mathbf{K}^{2n}) \\ &= \text{s-lim}((\mathbf{K}^{2n})^\dagger \cdot \mathbf{M}^{2n}(B) \cdot \mathbf{K}^{2n}) \\ &= \mathbf{M}(A).\end{aligned}$$

The matrix formulation of (probabilistic) bisimulation makes it also easy to see how graph and bisimulation equivalence are related. As $\mathcal{P}(n) \subset \mathcal{C}(n, n)$ we have:

Proposition 43. *If $p \sim_i q$ then $p \sim_b q$.*

5.2.1. Approximate bisimulation

In the case in which it is not possible to find a bisimulation equivalence for two states p and q of a PTS T, we can still identify them although only approximately. In order to do so, we introduce an ε-version of probabilistic bisimilarity. The intuitive idea is to find a classification operator $\mathbf{K}$ 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 us an estimate of the non-bisimilarity degree of p and q. By Definition 38, this will also be an estimate of the confinement of the system formed by the two processes p and q, which tells us how much the system is actually secure.

Definition 44. Let $T = (S, A, \longrightarrow, \pi_0)$ be a probabilistic transition system and let p and q be two states in S with operator representations $\mathbf{M}(p)$ and $\mathbf{M}(q)$. We say that p and q are *ε-bisimilar*, denoted by $p \sim_b^\varepsilon q$, iff

$$\inf_{\mathbf{K}_p, \mathbf{K}_q \in \mathcal{C}} \|\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q\| = \varepsilon,$$

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 partially the original notion of strict (probabilistic) bisimulation:

Proposition 45. *An ε-bisimulation with $\varepsilon = 0$, i.e. $\sim_b^0$, is a (probabilistic) bisimulation for finite (probabilistic) transition systems.*

Proof. By hypothesis there are only finitely many $\mathbf{K}_p$ and $\mathbf{K}_q$. Thus inf can be replaced by min. That means that there exist classification operators $\mathbf{K}_p$ and $\mathbf{K}_q$ such that $\|\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q\| = 0$. This implies $\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q = \mathbf{O}$, i.e. $\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_q = \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q$. □

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

Example 46. Let us compare the three, obviously somehow "similar" PTS's in Fig. 5. These processes are not probabilistic bisimilar. However one can try to determine how

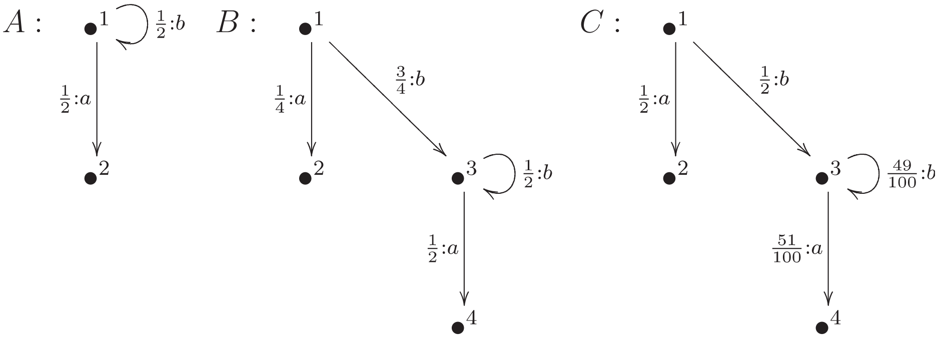


Fig. 5. Three generative probabilistic transition systems.

similar they are. The matrix $\mathbf{A} = \mathbf{M}(A)$ is the same as in Example 13; for the others we get

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

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

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

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

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

Measuring the difference (by means of the operator norm) leads to the following result:

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

In a security setting, this result allows us to conclude that although both the systems $\{A, B\}$ and $\{A, C\}$ are not probabilistic confined, the latter is "more secure" than the former in the sense that the chances of an information leak by observing the system are much smaller.

5.3. Weak bisimulation

Several authors have argued that bisimulation, although weaker than graph isomorphism, is still a too strong requirement for many purposes and suggested a number of variations (see [28] for a detailed account).

Weak bisimulation was introduced in [45] as a bisimulation which abstracts from internal computation by considering transitions of the form $\Rightarrow\xrightarrow{\alpha}\Rightarrow$, where $\Rightarrow$ is the transitive, reflexive closure of $\xrightarrow{\tau}$, and τ is an internal action representing some invisible computation.

Various probabilistic extensions of weak bisimulation have been proposed by several authors in the context of fully probabilistic systems [4], as well as for a generalisation of reactive systems [61], for probabilistic systems which allow for both non-deterministic and probabilistic branching [50], and for generative-reactive models [1]. In a language-based setting, a notion of probabilistic weak bisimulation has been introduced in [63] for a multi-threaded language modelled via discrete Markov chains. Applications to the problem of secure information flow are considered in [1,63].

At the base of a weak bisimulation semantics for probabilistic systems is the problem of determining the probability with which a weak transition $\Rightarrow\xrightarrow{\alpha}\Rightarrow$ may take place. In a fully probabilistic model such as the generative one, all the necessary information is available to compute such a probability, as all actions including τ are governed by some internally chosen probability distribution [4]. It is also possible to determine the probability of weak transitions in models which also includes some form of nondeterminism, provided all nondeterminism is first resolved according to criteria which depend on the particular model [50,61]. However, it is hard to imagine how such a probability can be established in a purely probabilistic model such as the reactive model, unless one reserves a different treatment to the internal action τ, thus effectively constructing a mixed reactive-generative model [1].

Based on this argument, we have chosen to exclude the reactive model from our treatment of the weak bisimulation semantics, and to apply linear operator based techniques similar to those we have used for bisimulation to re-cast the probabilistic weak bisimulation notion introduced in [4] for generative systems. As a first step we will show how we can represent the relation $\xrightarrow{\tau}{}^{*}\xrightarrow{\alpha}\xrightarrow{\tau}{}^{*}$ in terms of the transition matrices introduced in Section 3.4.

The probability of reaching a state or a certain class of states by sequences of actions or *traces* is defined in [4] for strings in a generic language $\Lambda \subset A^*$ recursively as follows:

$$\begin{aligned}
&\mathcal{P}(s, \Lambda, C) = 1 \quad \text{if } s \in C \quad \text{and} \quad \varepsilon \in \Lambda,\\
&\mathcal{P}(s, \Lambda, C) = \sum_{(a,t)\in A\times S} P(s, a, t) \cdot \mathcal{P}(t, \Lambda/a, C) \text{ otherwise},
\end{aligned}$$

where Λ/a denotes the set of all strings λ such that $a\lambda \in \Lambda$, and ε denotes the empty string.

By considering the language $\Lambda = \tau^* a \tau^* \cup \varepsilon$, the notion of probabilistic weak bisimulation can be defined as follows.

Definition 47. A weak bisimulation is an equivalence relation $\sim_w$ on S such that for all $s \sim_w s'$ and all $\alpha \in A \setminus \{\tau\} \cup \varepsilon$ and all equivalence classes $C \in S/\sim_w$ we have

$$\mathcal{P}(s, \tau^*\alpha\tau^*, C) = \mathcal{P}(s', \tau^*\alpha\tau^*, C).$$

We observe that the base case in the recursive definition of $\mathcal{P}(s, \Lambda, C)$ ensures the uniqueness of the solution of the second equation in the definition by forcing the consideration of only the minimal trace in Λ leading from s to C; all extensions of this minimal trace the language may contain and which also reach the target class do not contribute to $\mathcal{P}(s, \Lambda, C)$.

We will now show how to define a linear operator $\mathbf{F}$ with entries $(\mathbf{F})_{sC}$ corresponding to the probabilities $\mathcal{P}(s, \tau^*a\tau^*, C)$ for all $s \in S$ and $C \in S/\sim_w$.

The first step towards the definition of a linear operator expressing the probabilistic weak bisimulation relation introduced above is to look at the reachability of a state from another state via a single trace. In particular we are interested in traces of the form $\tau^n\alpha\tau^m$, with $n, m \in \mathbb{N}, n, m \geqslant 0$.

It is well known that iterating a transition matrix n times gives the probability of reaching state s from t in *exactly* n steps. This is sometimes known as the Chapman–Kolmogorov equations, e.g. [34, Theorem 6.1.7]. Generalising this idea slightly leads us to introduce the following operators $\mathbf{E}_\alpha(p)(n, m)$.

Definition 48. Given the operator representation $\mathbf{M}(p)$ of a probabilistic process p with $A = \{\alpha, \beta, \ldots, \tau\}$, then we define, for all $\alpha \in A$,

$$\mathbf{E}_\alpha(p)(n, m) = \mathbf{M}_\tau(p)^n \mathbf{M}_\alpha(p) \mathbf{M}_\tau(p)^m.$$

We denote by $\mathbf{E}(p)(n, m)$ the direct sum $\bigoplus_{\alpha \in A} \mathbf{E}_\alpha(p)(n, m)$.

It is easy to show the following result:

Proposition 49. *Given the operator representation $\mathbf{M}(p)$ of a probabilistic process p, then for all states $s, s' \in S$,*

$$(\mathbf{E}_\alpha(p)(n, m))_{s,s'} = \mathcal{P}(s, \tau^n\alpha\tau^m, s').$$

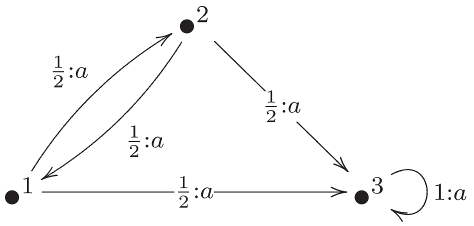
The next step is to look at the probability of reaching a state from any other state by any trace in $\tau^*\alpha\tau^*$. The straightforward idea is to determine this probability by summing up all the probabilities for reaching a state t from s by $\varepsilon, a, \tau\alpha, \alpha\tau, \tau\alpha\tau$, etc., i.e. via the operator

$$\overline{\mathbf{E}}_\alpha(p) = \sum_{n,m=0}^{\infty} \mathbf{E}_\alpha(p)(n, m),$$

for all $\alpha \in A$.

Unfortunately, for essentially the same reasons explained for the recursive definition of $\mathcal{P}$, this simple solution does not work. The problem is that some “reaching probabilities” are counted too often, in particular those associated to traces which are extensions of the minimal trace leading from a given state s to a target state t. The following example illustrates this problem.

Example 50. Consider the following simple PTS with only one action a:



In order to calculate the probabilities $\mathcal{P}(s, a^*, \{t\})$, we construct the operators:

$$\mathbf{M}_a = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix}, \quad \lim_{n\to\infty} \mathbf{M}_a^n = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \sum_{i=0}^{\infty} \mathbf{M}_a^i = \begin{pmatrix} \frac{4}{3} & \frac{2}{3} & \infty \\ \frac{2}{3} & \frac{4}{3} & \infty \\ 0 & 0 & \infty \end{pmatrix}.$$

This result is obviously not reflecting the probabilities we would expect. In fact, the entries in $\sum_{i=0}^{\infty}\mathbf{M}_a^i$ are not probabilities at all.

In order to obtain a correct result we have to compute the probability of reaching a state t the *first* time, i.e. along the minimal trace leading to t. This means we have to "block" out all contributions which come from paths which already passed through t before.

We can achieve this by projecting out all transitions from t in the operator $\mathbf{M}_\alpha(p)$. We define a projection into t as a diagonal matrix which contains a single entry 1 at the position (t, t), and its "negation", i.e.

$$(\mathbf{P}_t)_{ij} = \begin{cases} 1 & \text{for } i = j = t, \\ 0 & \text{otherwise,} \end{cases} \qquad (\mathbf{P}_t^\perp)_{ij} = \begin{cases} 1 & \text{for } i = j \neq t, \\ 0 & \text{otherwise.} \end{cases}$$

If we thus consider the modified transition operator

$$\mathbf{M}_{\alpha,\neg t}(p) = \mathbf{P}_t^\perp \mathbf{M}_\alpha(p),$$

we get the same transitions as in $\mathbf{M}_\alpha(p)$ except that all transitions from t are cancelled out—as the matrix $\mathbf{M}_{\alpha,\neg t}(p)$ is identical to $\mathbf{M}_a(p)$ except for the fact that the row t contains only zeros.

If we consider now the column t in $\mathbf{M}^n_{\alpha,\neg t}(p)$ we obtain for each state s the probability of reaching t in exactly n steps without passing through t, i.e. for the first time. We can extract this t column by multiplying with the projection $\mathbf{P}_t$, i.e.

$$(\mathbf{P}_t^\perp \mathbf{M}_\alpha(p))^n \cdot \mathbf{P}_t = (\mathbf{M}_{\alpha,\neg t}(p))^n \cdot \mathbf{P}_t.$$

The probability of getting from any state s to t via the minimal trace in at most n steps is then given by

$$\sum_{i=0}^{n} (\mathbf{P}_t^\perp \mathbf{M}_\alpha(p))^i \cdot \mathbf{P}_t = \sum_{i=0}^{n} (\mathbf{M}_{\alpha,\neg t}(p))^i \cdot \mathbf{P}_t.$$

This operation avoids the pitfalls of our previous attempt: once we have a trace from a state s reaching state t the first time, all its extensions are ignored as in $\mathbf{M}_{\alpha,\neg t}(p)$ there is no transition which leaves the state t again.

By combining this information for all states t we obtain for all $\alpha \in A$ the matrix

$$\sum_{t\in S}\left(\sum_{i=0}^{n}(\mathbf{P}_t^{\perp}\mathbf{M}_\alpha(p))^i \cdot \mathbf{P}_t\right) = \sum_{t\in S}\left(\sum_{i=0}^{n}(\mathbf{M}_{\alpha,\neg t}(p))^i \cdot \mathbf{P}_t\right).$$

Example 51. Consider again the simple process in Example 50. The projection operators for $t = 2$ are:

$$\mathbf{P}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{P}_2^{\perp} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and the corresponding modified a-transition operator

$$\mathbf{M}_{a,\neg 2} = \mathbf{P}_2^{\perp}\mathbf{M}_a = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The second column of the nth iteration of $\mathbf{M}_{a,\neg 2}$ then gives us the probabilities that we get from any state to the second state in exactly n steps the first time:

$$\mathbf{M}_{a,\neg 2}^0\mathbf{P}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{M}_{a,\neg 2}^1\mathbf{P}_2 = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{M}_{a,\neg 2}^2\mathbf{P}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\ldots$$

The first iteration means that if we start in the second state we “reach” it in zero steps, but there is no other state from which we can reach it in zero steps. The second iteration tells us that we reach the second state in one step only from the first one with probability $1/2$. After that all iterations indicate that there is no path of length larger than one reaching the second state (the first time). Obviously there is, for example, a three step path from state one to two back to one and then again to two: the probability of this path is $1/2 \cdot 1/2 \cdot 1/2 = 1/8$, however it is ignored in this construction as it visits the state twice.

We can combine the information on the probability of reaching all states in i steps in the operator $\sum_{t\in S}\mathbf{M}_{a,\neg t}^i\mathbf{P}_t$, whose iteration results in the following sequence of transition matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \frac{1}{8} \\ 0 & 0 & \frac{1}{8} \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \frac{1}{16} \\ 0 & 0 & \frac{1}{16} \\ 0 & 0 & 0 \end{pmatrix}, \ldots$$

Finally we can compute the probability of reaching a state from any other by any string in the language a^* by

$$\sum_{t\in S}\left(\sum_{i=0}^{\infty}\mathbf{M}^i_{a,\neg t}\mathbf{P}_t\right)=\sum_{i=0}^{\infty}\left(\sum_{t\in S}\mathbf{M}^i_{a,\neg t}\mathbf{P}_t\right)=\begin{pmatrix}1&\frac{1}{2}&1\\ \frac{1}{2}&1&1\\ 0&0&1\end{pmatrix}.$$

Proposition 52. *Given the operator representations* $\mathbf{M}(p)$ *of a probabilistic transition system* $p=(S,A,\longrightarrow,s_0)$ *then for all* $\alpha\in A$:

$$\mathcal{P}(s,\alpha^*,\{t\})=\left(\sum_{i=0}^{\infty}\left(\sum_{t\in S}\mathbf{M}^i_{\alpha,\neg t}(p)\mathbf{P}_t\right)\right)_{st}.$$

The example above suggests that in order to compute the probabilities of reaching a given state with traces in the language $\tau^*\alpha\tau^*$, we first have to appropriately modify the operator $\mathbf{E}_\alpha(p)(n,m)$ in Definition 48.

Definition 53. Given the operator representations $\mathbf{M}(p)$ of a probabilistic transition system $p=(S,A,\longrightarrow,s_0)$ with $A=\{a,b,\ldots,\tau\}$, then we define for all $\alpha\in A$:

$$\mathbf{F}_\alpha(p)(n,m)=\sum_{t\in S}\mathbf{M}_\tau(p)^n\cdot\mathbf{M}_\alpha(p)\cdot(\mathbf{P}_t^\perp\mathbf{M}_\tau(p))^m\cdot\mathbf{P}_t.$$

We denote by $\mathbf{F}(p)(n,m)$ the direct sum $\bigoplus_{\alpha\in A}\mathbf{F}_\alpha(p)(n,m)$ of all $\mathbf{F}_\alpha(p)(n,m)$.

Note that we treat the final τ's in a trace differently from the initial ones (and from the α transition as well). We allow repeated visits in the initial phase, while in the final phase we again "block out" multiple visits to the terminal state. This asymmetry is due to the fact that until the α transition has been performed we cannot terminate our path, only in the second part of a word in $\tau^*\alpha\tau^*$ will we terminate our attempt to find a trace connecting two states as soon as we reach the intended target. This is also reflected in the definition of $\mathcal{P}(s,\tau^*\alpha\tau^*,\{t\})$: we can invoke the rule for the base case only once removing the initial actions from all words in Λ, i.e. when Λ/a, results in a language containing the empty trace ε. No removal of an initial τ can achieve this, only once the α step has happened is this possible.

The operator $\mathbf{F}_\alpha(p)(n,m)$ encodes the probabilities of reaching a state by the trace $\tau^n\alpha\tau^m$, for some fixed $n,m\in\mathbb{N}$. The extension to the language $\tau^*\alpha\tau^*$ can be achieved by the operator

$$\overline{\mathbf{F}}_\alpha(p)=\sum_{n,m=0}^{\infty}\mathbf{F}_\alpha(p)(n,m),$$

which gives us the probabilities for any string in $\tau^*\alpha\tau^*$. More precisely we have:

Proposition 54. *Given the operator representation* $\mathbf{M}(p)$ *of a probabilistic transition system* $p=(S,A,\longrightarrow,s_0)$ *then for all* $\alpha\in A$:

$$\mathcal{P}(s,\tau^*\alpha\tau^*,\{t\})=(\overline{\mathbf{F}}_\alpha(p))_{st}.$$

The last step towards the definition of a linear operator representing the probabilistic weak bisimulation equivalence in Definition 47 is to introduce projection operators on classes of states. Let $C \subseteq S$ be a set of states, then the projection on C and its negation are defined by

$$(\mathbf{P}_C)_{ij} = \begin{cases} 1 & \text{for } i = j \wedge i \in C, \\ 0 & \text{otherwise,} \end{cases} \qquad (\mathbf{P}^{\perp}_{C})_{ij} = \begin{cases} 1 & \text{for } i = j \wedge i \notin C, \\ 0 & \text{otherwise.} \end{cases}$$

As recalled in Section 4.2.2, an equivalence relation $\mathcal{R}$ has a linear representation given by a classification matrix $\mathbf{K}_{\mathcal{R}}$. If $\mathbf{K}$ is the classification matrix associated to a probabilistic weak bisimulation equivalence on a state space S, then we can use it to construct the projection operators $\mathbf{P}_{C_i}$ and $\mathbf{P}^{\perp}_{C_i}$ for all classes C_i in the partition of the state space $S = \bigcup_i C_i$ induced by that relation. We denote by $\mathbf{K}_{\cdot,i}$ the ith column of $\mathbf{K}$, corresponding to class C_i. Then $\mathbf{P}_{C_i}$ can be constructed as the diagonal matrix $diag(\mathbf{K}_{\cdot,i})$ with the ith column of $\mathbf{K}$ as diagonal, and $\mathbf{P}^{\perp}_{C_i}$ as $\mathbf{I} - \mathbf{P}_{C_i} = \mathbf{I} - diag(\mathbf{K}_{\cdot,i})$ with $\mathbf{I}$ the identity matrix.

Definition 55. Given the operator representation $\mathbf{M}(p)$ of a probabilistic transition system $p = (S, A, \longrightarrow, s_0)$ with $A = \{a, b, \ldots, \tau\}$, and a partition $\mathcal{C} = \{C_i\}_i$ of S represented by a classification matrix $\mathbf{K}$ then we define for all $\alpha \in A$:

$$\mathbf{F}_\alpha(p, \mathbf{K})(n, m) = \sum_{C_i \in \mathcal{C}} \mathbf{M}_\tau(p)^n \cdot \mathbf{M}_\alpha(p) \cdot (\mathbf{P}^{\perp}_{C_i}\mathbf{M}_\tau(p))^m \cdot \mathbf{P}_{C_i}.$$

We denote by $\mathbf{F}(p, \mathbf{K})(n, m)$ the direct sum $\bigoplus_{\alpha \in A} \mathbf{F}_\alpha(p, \mathbf{K})(n, m)$, and

$$\overline{\mathbf{F}}_\alpha(p, \mathbf{K}) = \sum_{n,m=0}^{\infty} \mathbf{F}_\alpha(p, \mathbf{K})(n, m).$$

This operators "blocks" out all repeated visits to the same class in essentially the same way as discussed in Section 5.3. We therefore have, as expected, the following result:

Proposition 56. *Given the operator representations $\mathbf{M}(p)$ of a probabilistic transition systems $p = (S, A, \longrightarrow, s_0)$ and a partition $\mathcal{C} = \{C_i\}_i$ of S represented by a classification matrix $\mathbf{K}$ then for all $\alpha \in A$:*

$$\mathcal{P}(s, \tau^*\alpha\tau^*, C) = (\overline{\mathbf{F}}_\alpha(p, \mathbf{K}) \cdot \mathbf{K})_{sC}.$$

The following proposition gives a necessary and sufficient condition for two processes being probabilistic weak bisimilar.

Proposition 57. *Given the operator representations $\mathbf{M}(p)$ and $\mathbf{M}(q)$ of two probabilistic transition systems $p = (S, A, \longrightarrow, s_0)$ and $q = (S', A, \longrightarrow', s'_0)$ then p and q are probabilistic weak bisimilar iff there exist classification matrices $\mathbf{K}_p \in \mathcal{C}(|S|, n)$ and $\mathbf{K}_q \in \mathcal{C}(|S'|, n)$ for some $n \geqslant 1$ such that*

$$\mathbf{K}_p^{\dagger} \cdot \overline{\mathbf{F}}(p, \mathbf{K}_p) \cdot \mathbf{K}_p = \mathbf{K}_q^{\dagger} \cdot \overline{\mathbf{F}}(q, \mathbf{K}_q) \cdot \mathbf{K}_q,$$

i.e. for all $\alpha \in A$ we have $\mathbf{K}_p^{\dagger} \cdot \overline{\mathbf{F}}_\alpha(p, \mathbf{K}_p) \cdot \mathbf{K}_p = \mathbf{K}_q^{\dagger} \cdot \overline{\mathbf{F}}_\alpha(q, \mathbf{K}_q) \cdot \mathbf{K}_q$.

When there are no terminal τ loops in a transition graph, we obviously have $(\mathbf{P}_{C_i}^{\perp}\mathbf{M}_\tau(p))^m \cdot \mathbf{P}_{C_i} = \mathbf{M}_\tau(p)^m$. Thus, in this case we can use $\overline{\mathbf{E}}$ in place of $\overline{\mathbf{F}}$ in order to decide whether two processes are probabilistic weak bisimilar.

5.3.1. Approximate weak bisimulation

An approximative version of this notion allows us to capture how close two processes are to being weakly bisimilar.

Definition 58. Given the operator representations $\mathbf{M}(p)$ and $\mathbf{M}(q)$ of two probabilistic transition systems $p = (S, A, \longrightarrow, s_0)$ and $q = (S', A, \longrightarrow', s_0')$, we say that p and q are *probabilistic* ε*-weak bisimilar*, denoted by $p \sim_w^\varepsilon q$, if

$$\inf_{\mathbf{K}_p, \mathbf{K}_q \in \mathcal{C}} \|\mathbf{K}_p^\dagger \cdot \overline{\mathbf{F}}(p, \mathbf{K}_p) \cdot \mathbf{K}_p - \mathbf{K}_q^\dagger \cdot \overline{\mathbf{F}}(q, \mathbf{K}_q) \cdot \mathbf{K}_q\| = \varepsilon,$$

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

For $\varepsilon = 0$ we recover the original notion of strict probabilistic weak bisimulation:

Proposition 59. *For finite probabilistic transition systems, a probabilistic* ε*-weak bisimulation for* $\varepsilon = 0$*, i.e.* $\sim_w^0$*, is a probabilistic weak bisimulation.*

Example 60. We consider here a slightly modified version of an example taken from [63] where the setting is a multi-threaded language with a Markov chain semantics. The processes P, Q and R are described by the transition graphs in Fig. 6. Their matrix representations are given by

$$\mathbf{M}_a(P) = \begin{pmatrix} 0 & 0 & \frac{1}{3} & \frac{1}{6} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_\tau(P) = \begin{pmatrix} \frac{1}{3} & \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{M}_a(Q) = \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

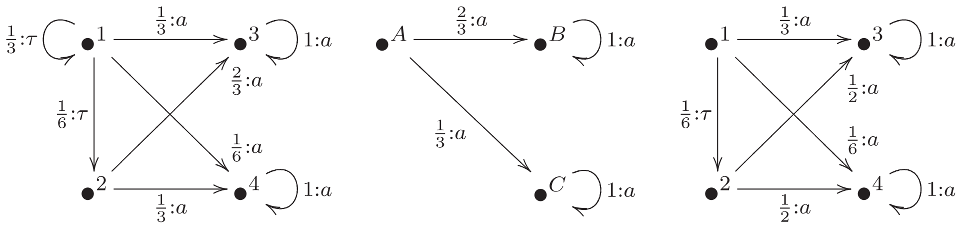


Fig. 6. Three generative probabilistic transition systems: *P*, *Q* and *R*.

$$\mathbf{M}_a(R) = \begin{pmatrix} 0 & 0 & \frac{1}{3} & \frac{1}{6} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_\tau(R) = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Partitioning the states in three classes and using the classification matrix $\mathbf{K}_P$ and its Moore–Penrose pseudo-inverse $\mathbf{K}_P^\dagger$,

$$\begin{aligned} C_1 &= \{s_1, s_2\}, \\ C_2 &= \{s_3\}, \\ C_3 &= \{s_4\}, \end{aligned} \qquad \mathbf{K}_P = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{K}_P^\dagger = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

allows us to compute the transition probabilities $\mathcal{P}(s_i, \tau^* a \tau^*, C_j)$ and the abstracted system

$$\overline{\mathbf{F}}_a(P, \mathbf{K}_P) \cdot \mathbf{K}_P = \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{K}_P^\dagger \cdot \overline{\mathbf{F}}_a(P, \mathbf{K}_P) \cdot \mathbf{K}_P = \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Processes P and Q are thus probabilistic weak bisimilar as we have $\mathbf{K}_P^\dagger \cdot \overline{\mathbf{F}}_a(P) \cdot \mathbf{K} = \mathbf{M}_a(Q)$, (we can use the trivial abstraction $\mathbf{K}_Q = \mathbf{K}_Q^\dagger = \mathbf{I}$ for the process Q). It is interesting to note that in this example the "naive" approach based on the operator $\overline{\mathbf{E}}$ gives the same result, i.e. $\overline{\mathbf{E}}_a(P) \cdot \mathbf{K}_P = \overline{\mathbf{F}}_a(P, \mathbf{K}_P) \cdot \mathbf{K}_P$ and thus:

$$\mathbf{K}_P^\dagger \cdot \overline{\mathbf{E}}_a(P) \cdot \mathbf{K}_P = \mathbf{K}_P^\dagger \cdot \overline{\mathbf{F}}_a(P, \mathbf{K}_P) \cdot \mathbf{K}_P.$$

This is due to the fact that in this example there are no τ loops or cycles possible after *a* has happened.

When we compare processes *P* and *R* we see that they are not weakly bisimilar. However, we can look for abstractions which make the difference between them minimal. Coincidentally, these are given by exactly the same classification matrices as before. For $\mathbf{K}_Q = \mathbf{K}_R$ and $\mathbf{K}_Q = \mathbf{I}$ we obtain a minimal distance between Q and R which we calculate by using the supremum norm as

$$\|\mathbf{K}_R^\dagger \cdot \overline{\mathbf{F}}_a(R, \mathbf{K}_R) \cdot \mathbf{K}_R - \mathbf{M}_a(Q)\| = \left\| \begin{pmatrix} 0 & \frac{13}{24} & \frac{11}{24} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\| = \frac{1}{4}.$$

Example 61. Consider the probabilistic transition system P in Fig. 7 taken from [4]. Its matrix representation is given by $\mathbf{M}_a(P) \oplus \mathbf{M}_b(P) \oplus \mathbf{M}_\tau(P)$:

$$\begin{pmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&.1&0&0\\ 0&0&0&0&0&0&.1&0\\ 0&0&0&0&0&0&0&.2\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{pmatrix} \oplus \begin{pmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&.5&0&.4&0&0\\ 0&0&0&0&0&0&.4&0\\ 0&0&0&0&0&0&0&.8\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{pmatrix} \oplus \begin{pmatrix} 0&.5&.5&0&0&0&0&0\\ 0&0&0&.6&.4&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&.5&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&.5&.5&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0 \end{pmatrix}.$$

In [4] a probabilistic weak bisimulation relation on the states of p is computed which consists of four classes. The classification operator $\mathbf{K}_P$ corresponding to this relation and its Moore–Penrose pseudo-inverse $\mathbf{K}_P^\dagger$ are as follows:

$$\begin{aligned} C_1 &= \{s_1\},\\ C_2 &= \{s_3\},\\ C_3 &= \{s_2, s_4, s_5\},\\ C_4 &= \{s_6, s_7, s_8\}, \end{aligned} \qquad \mathbf{K}_P = \begin{pmatrix} 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&1&0\\ 0&0&0&1\\ 0&0&0&1\\ 0&0&0&1 \end{pmatrix}, \qquad \mathbf{K}_P^\dagger = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&\frac{1}{3}&0&\frac{1}{3}&\frac{1}{3}&0&0&0\\ 0&0&0&0&0&\frac{1}{3}&\frac{1}{3}&\frac{1}{3} \end{pmatrix}.$$

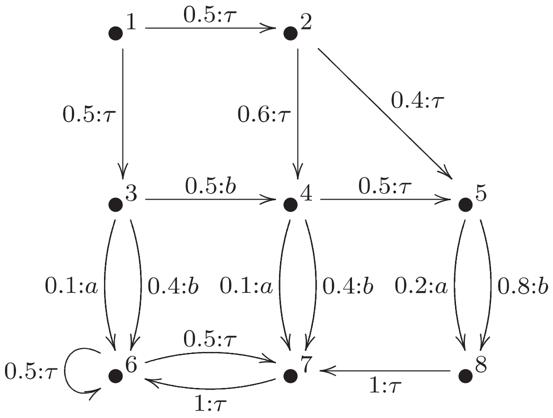


Fig. 7. A generative probabilistic transition system.

The resulting abstract system is given by

$$\mathbf{K}_P^\dagger \overline{\mathbf{F}}_a(P)\mathbf{K}_P = \begin{pmatrix} 0 & 0 & 0 & 0.15 \\ 0 & 0 & 0 & 0.10 \\ 0 & 0 & 0 & 0.20 \\ 0 & 0 & 0 & 0.00 \end{pmatrix}, \quad \mathbf{K}_P^\dagger \overline{\mathbf{F}}_b(P)\mathbf{K}_P = \begin{pmatrix} 0 & 0 & 0.25 & 0.60 \\ 0 & 0 & 0.50 & 0.40 \\ 0 & 0 & 0.00 & 0.80 \\ 0 & 0 & 0.00 & 0.00 \end{pmatrix},$$

is obviously probabilistic weak bisimilar to P.

Note that the use of the operators $\overline{\mathbf{E}}_a(P)$ and $\overline{\mathbf{E}}_b(P)$ would give in this case an incorrect result:

$$\mathbf{K}_P^\dagger \overline{\mathbf{E}}_a(P)\mathbf{K}_P = \begin{pmatrix} 0 & 0 & 0 & \infty \\ 0 & 0 & 0 & \infty \\ 0 & 0 & 0 & \infty \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_P^\dagger \overline{\mathbf{E}}_b(P)\mathbf{K}_P = \begin{pmatrix} 0 & 0 & 0.38 & \infty \\ 0 & 0 & 0.75 & \infty \\ 0 & 0 & 0.00 & \infty \\ 0 & 0 & 0.00 & 0 \end{pmatrix}.$$

We can now measure how much states s_2 and s_3 are not equivalent with respect to $\mathbf{K}_P$ by comparing the associated reduced abstract systems

$$\mathbf{K}_{s_2}^\dagger \overline{\mathbf{F}}_a(s_2, \mathbf{K}_{s_2})\mathbf{K}_{s_2} = \begin{pmatrix} 0 & 0 & 0 & 0.0 \\ 0 & 0 & 0 & 0.0 \\ 0 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0.0 \end{pmatrix}, \quad \mathbf{K}_{s_2}^\dagger \overline{\mathbf{F}}_b(s_2, \mathbf{K}_{s_2})\mathbf{K}_{s_2} = \begin{pmatrix} 0 & 0 & 0 & 0.0 \\ 0 & 0 & 0 & 0.0 \\ 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0.0 \end{pmatrix},$$

$$\mathbf{K}_{s_3}^\dagger \overline{\mathbf{F}}_a(s_3, \mathbf{K}_{s_3})\mathbf{K}_{s_3} = \begin{pmatrix} 0 & 0 & 0 & 0.0 \\ 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0.0 \end{pmatrix}, \quad \mathbf{K}_{s_3}^\dagger \overline{\mathbf{F}}_b(s_3, \mathbf{K}_{s_3})\mathbf{K}_{s_3} = \begin{pmatrix} 0 & 0 & 0.0 & 0.0 \\ 0 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0.0 & 0.8 \\ 0 & 0 & 0.0 & 0.0 \end{pmatrix}.$$

By using again the supremum norm we get

$$\|\mathbf{K}_{s_2}^\dagger \cdot \overline{\mathbf{F}}(s_2, \mathbf{K}_{s_2}) \cdot \mathbf{K}_{s_2} - \mathbf{K}_{s_3}^\dagger \cdot \overline{\mathbf{F}}(s_3, \mathbf{K}_{s_3}) \cdot \mathbf{K}_{s_3}\| = 0.9,$$

which gives us an upper bound to the measure ε in Definition 58.

6. Bounds for ε

It is in general not an easy task to determine the infimum over all possible classification matrices in order to calculate ε. For finite processes we at least know that we have only finitely many classification matrices, but their number is increasing exponentially with the number of states. A brute force approach is therefore not computationally feasible. The complexity for deciding if two processes are probabilistically bisimilar (i.e. the case $\varepsilon = 0$) gives a lower bound for the complexity of the more general problem of determining a possibly non-zero ε.

One of the main advantages of an approximative approach towards the various security notions based on process equivalences is that in practical circumstances it might be sufficient to determine an upper bound for ε. This means that instead of trying to prove the perfect similarity of two processes, e.g. (weak) bisimilarity, our aim is to determine a bound for their dis-similarity as this gives us a bound for the possible or expected chances of a security breach. Such a conservative approximation is closely related to the approach taken in static program analysis.

6.1. One dimensional abstractions

A very crude but computationally cheap way to obtain a rough estimate—or more precisely an upper bound—for ε is to compare the one dimensional abstractions of two processes. That is, we can consider a classification matrix which maps all states into one single abstract state. If process p has n states and process p has m states then $\mathbf{K}_p$ is a $n \times 1$ matrix and $\mathbf{K}_q$ is a $m \times 1$ matrix, both of which contain 1's for each entry. The corresponding Moore–Penrose pseudo-inverses are given by $\mathbf{K}_p^\dagger$ an $1 \times n$ matrix containing $1/n$ for each entry and $\mathbf{K}_q^\dagger$ an $1 \times m$ matrix containing $1/m$ for each entry.

$$\mathbf{K}_p = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \mathbf{K}_q = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \begin{array}{l} \mathbf{K}_p^\dagger = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix}, \\ \\ \mathbf{K}_q^\dagger = \begin{pmatrix} \frac{1}{m} & \frac{1}{m} & \dots & \frac{1}{m} \end{pmatrix}. \end{array}$$

This allows us to construct one dimensional abstractions of both processes which we can compare in oder to obtain a "most general approximation" which we call $\varepsilon_\top$.

Definition 62. Given two probabilistic processes p and q, let $\mathbf{K}_p$ and $\mathbf{K}_q$ be their one dimensional abstraction operators. Then we define

$$\varepsilon_\top(p, q) = \| \mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q \|.$$

As ε is defined to be the infimum over all possible differences between abstractions of p and q we have that if $p \sim_b^\varepsilon q$, then $\varepsilon \leqslant \varepsilon_\top(p, q)$. In other words, $\varepsilon_\top$ gives us a safe upper bound for the approximation ε.

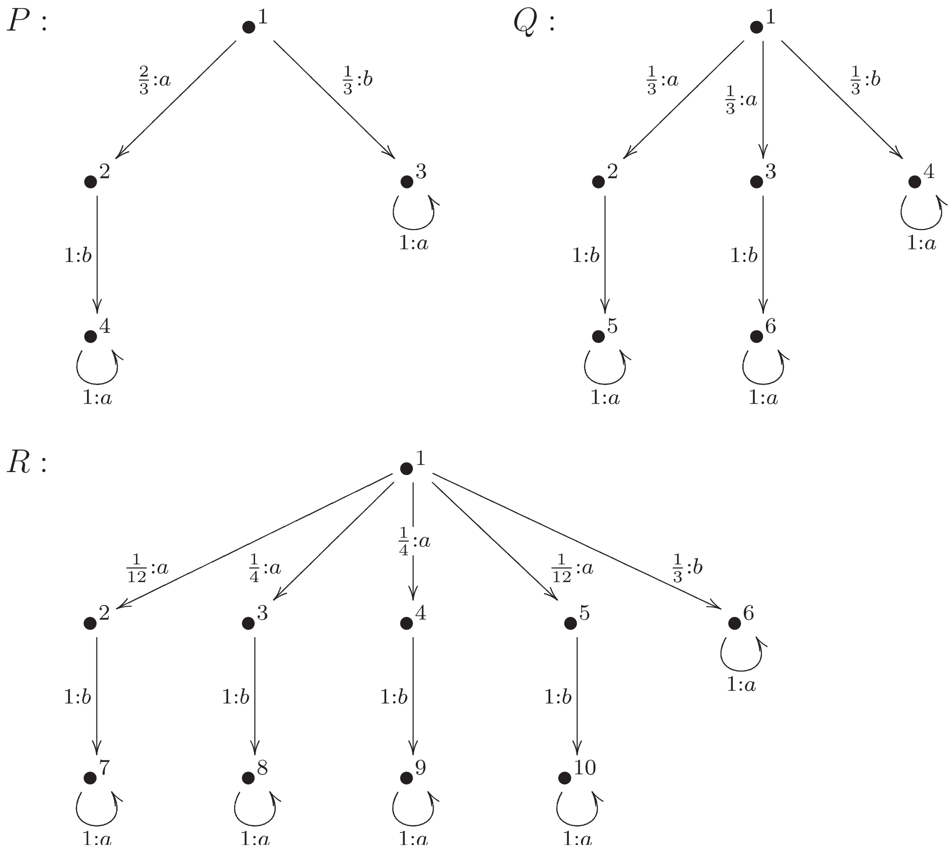


Fig. 8. Three generative probabilistic transition systems: *P*, *Q* and *R*.

Example 63. Consider the processes *P*, *Q* and *R* in Fig. 8 which are variations of the example [44, Fig. 4]. These processes are represented by the following matrices:

$$\mathbf{M}(P) = \begin{pmatrix} 0 & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{M}(Q) = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{M}(R) = \begin{pmatrix} 0 & \frac{1}{12} & \frac{1}{4} & \frac{1}{4} & \frac{1}{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

It is easy to see that these three processes are probabilistically bisimilar, the common abstraction for all three being the process P. However, it is not always so easy to determine the optimal abstractions. In this case one can determine upper bounds for ε by constructing the one dimensional abstractions. If we denote by $\mathbf{K}_P$, $\mathbf{K}_Q$ and $\mathbf{K}_R$ the classification matrices we get

$$\begin{aligned} \mathbf{K}_P^\dagger \mathbf{M}(P)\mathbf{K}_P &= \left(\tfrac{2}{3}\right) \oplus \left(\tfrac{1}{3}\right), \\ \mathbf{K}_Q^\dagger \mathbf{M}(Q)\mathbf{K}_Q &= \left(\tfrac{11}{18}\right) \oplus \left(\tfrac{7}{18}\right), \\ \mathbf{K}_R^\dagger \mathbf{M}(R)\mathbf{K}_R &= \left(\tfrac{17}{30}\right) \oplus \left(\tfrac{13}{30}\right), \end{aligned}$$

from which we can calculate (using the supremum norm) $\varepsilon_\top(P, Q) = \frac{1}{18}$, $\varepsilon_\top(P, R) = \frac{1}{10}$ and $\varepsilon_\top(Q, R) = \frac{2}{45}$, and conclude that

$$\begin{aligned} P &\sim_b^\varepsilon Q \text{ with } \varepsilon \leqslant \tfrac{1}{18}, \\ P &\sim_b^\varepsilon R \text{ with } \varepsilon \leqslant \tfrac{1}{10}, \\ Q &\sim_b^\varepsilon R \text{ with } \varepsilon \leqslant \tfrac{2}{45}. \end{aligned}$$

In other words, we have calculated correct over-approximations for the optimal ε; as we know that all processes are bisimilar, $\varepsilon = 0$ is optimal for all three processes.

From Proposition 30 we know that the abstraction of any stochastic matrix using classification matrices gives us again a stochastic matrix. As there is only a single one dimensional stochastic matrix, namely $\mathbf{M}_1 = (1)$, one might expect that all the one dimensional abstractions $\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p$ result in $\mathbf{M}_1$ and that therefore $\varepsilon_\top(p, q) = 0$ for all processes p and q.

However, unless we have only a single action a, the linear representations $\mathbf{M}(p)$ of generative processes are in general not stochastic (only the sum of their factors gives a stochastic matrix). It thus makes sense to compare the one dimensional abstractions of processes in order to obtain $\varepsilon_\top$, as in the above example where we have:

$$\mathbf{K}_P^\dagger \mathbf{M}(P)\mathbf{K}_P \neq \mathbf{K}_Q^\dagger \mathbf{M}(Q)\mathbf{K}_Q \neq \mathbf{K}_R^\dagger \mathbf{M}(R)\mathbf{K}_R.$$

6.2. On transitivity

Consider the situation in which we have three processes p, q and r, and we know that $p \sim_b^{\varepsilon_1} q$ and $q \sim_b^{\varepsilon_2} r$. If $\varepsilon_1 = \varepsilon_2 = 0$, then we can conclude that $p \sim_b^{\varepsilon} r$ with $\varepsilon = 0$ (since probabilistic bisimulation is an equivalence relation). What can we say about the number ε (without concretely computing it) in the general case where $\varepsilon_1 \neq 0$ or $\varepsilon_2 \neq 0$?

Let us first assume that the abstractions of p and r which we use to determine the values for ε_1 and ε_2 are of the same dimension.

Proposition 64. *Consider the processes p, q and r such that $p \sim_b^{\varepsilon_1} q$ and $q \sim_b^{\varepsilon_2} r$, i.e. there exist classification matrices $\mathbf{K}_p, \mathbf{K}_q, \overline{\mathbf{K}}_q$ and $\mathbf{K}_r$ such that*

$$\|\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q\| = \varepsilon_1 \text{ and } \|\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q - \mathbf{K}_r^\dagger \mathbf{M}(r)\mathbf{K}_r\| = \varepsilon_2.$$

Assume also that the dimensions of $\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$ and $\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q$ are the same.

Define $\delta = \|\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q\|$. Then

$$p \sim_b^{\varepsilon} r \quad \text{with } \varepsilon \leqslant \varepsilon_1 + \varepsilon_2 + \delta.$$

Proof.

$$\begin{aligned}
\varepsilon &= \inf_{\mathbf{K}_1,\mathbf{K}_2} \|\mathbf{K}_1^\dagger \mathbf{M}(p)\mathbf{K}_1 - \mathbf{K}_2^\dagger \mathbf{M}(r)\mathbf{K}_2\| \\
&\leqslant \|\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - \mathbf{K}_r^\dagger \mathbf{M}(r)\mathbf{K}_r\| \\
&= \|\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q + \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q) \\
&\quad -\mathbf{K}_r^\dagger \mathbf{M}(r)\mathbf{K}_r + \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q)\| \\
&= \|(\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q) + (\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q - \mathbf{K}_r^\dagger \mathbf{M}(r)\mathbf{K}_r) \\
&\quad +(\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q)\| \\
&\leqslant \|\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q\| + \|\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q - \mathbf{K}_r^\dagger \mathbf{M}(r)\mathbf{K}_r\| \\
&\quad +\|\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q) - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q)\overline{\mathbf{K}}_q\| \\
&= \varepsilon_1 + \varepsilon_2 + \delta. \qquad \square
\end{aligned}$$

Note that in the case where $\mathbf{K}_q = \overline{\mathbf{K}}_q$ we have: $\varepsilon \leqslant \varepsilon_1 + \varepsilon_2$, i.e. the triangular inequality applies in this case.

Suppose now that the dimension of the abstractions of p and r and the one for q and r which we use to determine the minimal ε_1 and ε_2 are different. The intuitive idea is to “fill up” the smaller one in order to obtain a similar upper bound for ε as in the previous case.

We first state a number of properties of the direct sum of operators [12, Theorem 3.4.1]. Let $\mathbf{A}$ be a $m_1 \times n_1$ matrix, $\mathbf{B}$ be a $m_2 \times n_2$ matrix, $\mathbf{C}$ be a $m_3 \times n_3$ matrix, $\mathbf{D}$ be a $m_4 \times n_4$ matrix, then the following holds:

(i) $(\mathbf{A} \oplus \mathbf{B})^\dagger = \mathbf{A}^\dagger \oplus \mathbf{B}^\dagger$,

(ii) $(\mathbf{A} \oplus \mathbf{B}) + (\mathbf{C} \oplus \mathbf{D}) = (\mathbf{A} + \mathbf{C}) \oplus (\mathbf{B} + \mathbf{D})$ if $m_1 = m_3$, $n_1 = n_3$, $m_2 = m_4$, and $n_2 = n_4$,

(iii) $(\mathbf{A} \oplus \mathbf{B}) \cdot (\mathbf{C} \oplus \mathbf{D}) = (\mathbf{A} \cdot \mathbf{C}) \oplus (\mathbf{B} \cdot \mathbf{D})$ if $n_1 = m_3$, and $n_2 = m_4$,
(iv) $\|\mathbf{A} \oplus \mathbf{O}\| = \|\mathbf{A}\|$ for any null matrix $\mathbf{O}$.

Suppose that $\mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q$ is an $n \times n$ matrix while $\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q$ is an $m \times m$ matrix and $n > m$. In order to allow for a comparison between $\mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q$ and $\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q$ we "fill up" the smaller one with zero entries by constructing the matrix $\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q \oplus \mathbf{O}_{n-m}$, where $\mathbf{O}_k$ indicates the k-dimensional null matrix, that is the $k \times k$ matrix with only zero entries.

Operationally this means that we consider $\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q$ as a process which operates on the same number of abstract states (classes) as $\mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q$ but without any transitions between the "extra" states.

Proposition 65. *Suppose we have three processes p, q and r such that $p \sim_b^{\varepsilon_1} q$ and $q \sim_b^{\varepsilon_2} r$ for some classification matrices $\mathbf{K}_p, \mathbf{K}_q, \overline{\mathbf{K}}_q$ and $\mathbf{K}_r$. Assume that the dimension of $\mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q$ is m and that the dimension of $\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q$ is n with $n > m$. Then we have that*

$$p \sim_b^{\varepsilon} r \quad \textit{with } \varepsilon \leqslant \varepsilon_1 + \varepsilon_2 + \delta',$$

where $\delta' = \|\mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q \oplus \mathbf{O}_{n-m}\|$.

Proof. The proof is essentially the same as in the previous case, making use of the properties listed above. For a $k \times l_1$ matrix $\mathbf{A}$ with $k \geqslant l_1$ and $l_2 \geqslant l_1$ we denote by $(\mathbf{A}|\mathbf{O}_{l_2-l_1})$ a $k \times l_2$ matrix where the first $1, \ldots, l_1$ columns are identical with $\mathbf{A}$ and the columns $l_1 + 1, l_1 + 2 \ldots, l_2$ are filled with zeros.

$$\begin{aligned}
\varepsilon &= \inf_{\mathbf{K}_1, \mathbf{K}_1} \|\mathbf{K}_1^\dagger \mathbf{M}(p) \mathbf{K}_1 - \mathbf{K}_2^\dagger \mathbf{M}(r) \mathbf{K}_2\| \\
&\leqslant \|\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - (\mathbf{K}_r|\mathbf{O}_{n-m})^\dagger \mathbf{M}(r) (\mathbf{K}_r|\mathbf{O}_{n-m})\| \\
&= \|\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_r^\dagger \mathbf{M}(r) \mathbf{K}_r \oplus \mathbf{O}_{n-m}\| \\
&= \|\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q + \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q \\
&\quad - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q \oplus \mathbf{O}_{n-m} + \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q \oplus \mathbf{O}_{n-m} - \mathbf{K}_r^\dagger \mathbf{M}(r) \mathbf{K}_r \oplus \mathbf{O}_{n-m}\| \\
&\leqslant \|\mathbf{K}_p^\dagger \mathbf{M}(p) \mathbf{K}_p - \mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q\| \\
&\quad + \|\mathbf{K}_q^\dagger \mathbf{M}(q) \mathbf{K}_q - \overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q \oplus \mathbf{O}_{n-m}\| \\
&\quad + \|\overline{\mathbf{K}}_q^\dagger \mathbf{M}(q) \overline{\mathbf{K}}_q \oplus \mathbf{O}_{n-m} - \mathbf{K}_r^\dagger \mathbf{M}(r) \mathbf{K}_r \oplus \mathbf{O}_{n-m}\| \\
&= \varepsilon_1 + \varepsilon_2 + \delta'. \qquad \square
\end{aligned}$$

The construction of safe bounds for ε is consistent with our overall conceptual approach in which we aim in a quantification of the behavioural "similarity" of processes as a means for establishing a "confidentiality level". As we will see in the next section, the value of ε corresponds to the distinguishability of processes via certain statistical tests. This is proportional to the information leakage and can be interpreted as a measure which is indirectly proportional to the minimal effort (number of tests/attacks) an attacker needs in order to obtain some confidential information (e.g. the identity of processes). A safe upper bound for ε thus gives us a safe lower bound for this effort, and thus a minimal guaranteed "confidentiality level".

7. The meaning of ε

Given two processes p and q of which we know that $p \sim_b^\varepsilon q$: What property of the two processes, or their difference does ε actually describe in a security context? We will investigate this question in the case of ε-bisimilarity and finite systems; for ε-weak bisimilarity and other approximate similarity notions as well as for infinite systems corresponding arguments can easily be developed along similar lines.

7.1. Process similarity and operator norm

We have already seen that ε in some way describes how (bi)similar the two processes are: In the case that $\varepsilon = 0$ we know that they are indistinguishable in the sense of a bisimulation semantics. Otherwise, we know that two "optimal" abstractions of p and q exists such that

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

In general the norm of a matrix defined as $\|\mathbf{A}\| = \sup_{\|x\|=1} \|x\mathbf{A}\|$ describes the maximal "stretching factor" of normalised vectors. The exact numerical value depends, of course, on the vector norm used (e.g. Euclidean or supremum norm).

The ε value which determines the similarity of p and q thus describes how much the effect of applying $\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p$ and $\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$ differ in the worst case. In other words, ε is a measure for how much the abstractions of p and q differ in a single step. If p and q are bisimilar, i.e. for $\varepsilon = 0$, there is a single, common abstraction of both processes p and q and we thus obtain the same "trace of distributions".

If we utilise the 1-norm, then the value of ε has a direct interpretation as the (positive) unitary vectors are exactly distributions (over abstract equivalence classes). As $\mathbf{K}_p^\dagger \mathbf{M}(q)\mathbf{K}_p$ and $\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$ are positive matrices the norm of their difference describes exactly the maximal difference between the (abstract) distributions we obtain in one step (executing $\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p$ or $\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$).

Restricting to the case of finite dimensional matrices, i.e. considering probabilistic transition systems with finitely many states, allows us to replace the supremum in the definition of an operator norm by the maximum, i.e. there is always a vector (distribution) x with $\|x\| = 1$ for which the norm difference between $x\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p$ and $x\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$ is maximal.

Proposition 66. *Given two stochastic $n \times n$-matrices* $\mathbf{S}$ *and* $\mathbf{T}$ *then*

$$\max_{\|x\|_1=1} \|x\mathbf{S} - x\mathbf{T}\|_1$$

is obtained for an extremal vector $x = (0, \ldots, 0, 1, 0, \ldots, 0)$, i.e. $x_i = 1$ for exactly one $i = 1, \ldots, n$, and $x_j = 0$ for $j \neq i$.

Proof. Consider the case $n = 2$. The general case can be shown analogously (by induction).

Suppose that $x = (x_1, x_2)$ is the maximal vector with $\|x\|_1 = 1$, i.e. $\|x\mathbf{S} - x\mathbf{T}\|_1$ is maximal. Without loss of generality assume that $1 \geqslant x_i \geqslant 0$. We therefore have $\|x\|_1 = |x_1| + |x_2| = x_1 + x_2 = 1$. The 1-norm of $x(\mathbf{S} - \mathbf{T})$ is given by

$$\begin{aligned}
&\|x(\mathbf{S} - \mathbf{T})\|_1 \\
&\quad = \|((\mathbf{S}_{11} - \mathbf{T}_{11})x_1 + (\mathbf{S}_{12} - \mathbf{T}_{12})x_2, (\mathbf{S}_{21} - \mathbf{T}_{21})x_1 + (\mathbf{S}_{22} - \mathbf{T}_{22})x_2)\|_1 \\
&\quad = |(\mathbf{S}_{11} - \mathbf{T}_{11})x_1| + |(\mathbf{S}_{12} - \mathbf{T}_{12})x_2| + |(\mathbf{S}_{21} - \mathbf{T}_{21})x_1| + |(\mathbf{S}_{22} - \mathbf{T}_{22})x_2| \\
&\quad = |\mathbf{S}_{11} - \mathbf{T}_{11}|x_1 + |\mathbf{S}_{12} - \mathbf{T}_{12}|x_2 + |\mathbf{S}_{21} - \mathbf{T}_{21}|x_1 + |\mathbf{S}_{22} - \mathbf{T}_{22}|x_2 \\
&\quad = (|\mathbf{S}_{11} - \mathbf{T}_{11}| + |\mathbf{S}_{21} - \mathbf{T}_{21}|)x_1 + (|\mathbf{S}_{12} - \mathbf{T}_{12}| + |\mathbf{S}_{22} - \mathbf{T}_{22}|)x_2.
\end{aligned}$$

We know that $\mathbf{S}$ and $\mathbf{T}$ are stochastic matrices, i.e. all $1 \geqslant \mathbf{S}_{ij} \geqslant 0$ and $1 \geqslant \mathbf{T}_{ij} \geqslant 0$, as well as $\mathbf{S}_{11} + \mathbf{S}_{12} = 1$, etc. We therefore know that for all absolute values in this expression we have: $0 \leqslant |\mathbf{S}_{11} - \mathbf{T}_{11}| \leqslant 1$, etc. We also know that in each row of $\mathbf{S} - \mathbf{T}$ one entry is positive and that the other is negative, and that the sum of the entries in the first row is the negative of the entries in the second row. There are now the following possible cases:

(i) $\mathbf{S} = \mathbf{T}$ in which case we get $\|\mathbf{S} - \mathbf{T}\|_1 = 0$ and thus any vector, in particular extremal ones, are maximal.

(ii) One row of $\mathbf{S} - \mathbf{T}$ is zero, e.g. the first one. Then either $|\mathbf{S}_{21} - \mathbf{T}_{21}| > |\mathbf{S}_{22} - \mathbf{T}_{22}|$ or vice versa. In the first case, any increase of x_1 (up to the maximal value $x_1 = 1$ results in a larger 1-norm of $\|x(\mathbf{S} - \mathbf{T})\|_1$, i.e. the maximum is achieved for an extremal vector.

(iii) None of the absolute values in the above expression vanishes. If we increase either x_1 or x_2 the above expression increases too, except when $|\mathbf{S}_{11} - \mathbf{T}_{11}| + |\mathbf{S}_{21} - \mathbf{T}_{21}| = |\mathbf{S}_{12} - \mathbf{T}_{12}| + |\mathbf{S}_{22} - \mathbf{T}_{22}|$ which can never happen (except in the two cases above). □

This means that it is sufficient to check how much $x\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p$ and $x\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$ differ by looking at all the extremal (basis) vectors e_i.

7.2. A statistical interpretation

Our basic approach towards confidentiality and non-interference is based on the concept of *identity confinement* [51]. According to this notion, the problem for an attacker or spy is to distinguish between several processes; the "secret" which should be protected in this setting is therefore the "identity" of the processes running. It is easy to translate the traditional notion of *confidentiality* (where the value of some "high level variable" constitutes the relevant "secret") into this essentially behavioural framework and vice versa.

Given now the role of ε for distinguishing two processes p and q—namely as single-step divergence factor—the question arise how one can make use of this information in order to describe how vulnerable some processes are against an attack. To simplify the arguments we only consider the problem of two processes p and q with $p \sim_b^\varepsilon q$.

Using standard statistical methods we can analyse the question of how many tests are needed to distinguish two processes which are ε-bisimilar with a certain confidence α. The framework of so-called *hypothesis testing* (see e.g. [62]) provides a simple way to estimate these parameters α and n.

7.2.1. Identification by testing

Let us consider the situation where we have two processes *p* and *q* which we assume to be ε-bisimilar, for some $\varepsilon \geqslant 0$. In order to simplify the situation, we assume that there is only a single label *a*. We can identify some abstract state *s*, i.e. equivalence class of states $[s]$ and a point distribution (extremal vector) x_s representing *s* such that:

$$\max_{\|x\|_1=1} \|x\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - x\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q\|_1 = \|x_s\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - x_s\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q\|_1.$$

Following the standard interpretation of probabilities as “long-run” relative frequencies [35], we can expect that the number of times a certain class of states $[t]$ is reached (via a transition labelled by *a*) from *s* is given exactly by the corresponding coordinates in $x_s\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p$ and $x_s\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q$. This means that if we execute *p* or *q* “infinitely” often we can determine $p_{s,t} = (x_s\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p)_t$ and $q_{s,t} = (x_s\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q)_t$ as the limit of the frequencies with which we obtain a successor state in $[t]$.

In fact, for any unknown process *x* we can attempt do determine $x_{s,t}$ experimentally by executing *x* over and over again in state *s*. Assuming that *x* is actually the same as either *p* or *q* we know that the $x_{s,t}$ we obtain must be either $p_{s,t}$ or $q_{s,t}$. We thus can easily determine this way if $x = p$ or q, i.e. reveal the identity of *x* (if $\varepsilon \neq 0$), simply by testing *x* in state *s*.

The above described experimental setup is unfortunately only of theoretical value; we have no way to repeat this experiment—as required—infinitely often. For practical purposes we need a way to distinguish *p* and *q* by finite executions of *p* and *q*. If we execute *p* and *q* only a finite number of—say *n*—times, we can observe a certain experimental frequency $p_{s,t}^n$ and $q_{s,t}^n$. Each time we repeat a finite sequence of *n* tests we may get different values for $p_{s,t}^n$ and $q_{s,t}^n$ (only the infinite experiments will eventually converge to the same constant values $p_{s,t}$ and $q_{s,t}$).

Analogously, we can determine the frequency $x_{s,t}^n$ for an unknown process *x* by testing, i.e. by looking at *n* executions of *x*. We can then try to compare $x_{s,t}^n$ with $p_{s,t}^n$ and $q_{s,t}^n$ or with $p_{s,t}$ and $q_{s,t}$ in order to find out if $x = p$ or $x = q$. Unfortunately, there is neither a single value for either $x_{s,t}^n$, $p_{s,t}^n$ or $q_{s,t}^n$ (each experiment may give us different values) nor can we test if $x_{s,t}^n = p_{s,t}^n$ or $x_{s,t}^n = q_{s,t}^n$ nor if $x_{s,t}^n = p_{s,t}$ or $x_{s,t}^n = q_{s,t}$.

For finite experiments we can only make a guess about the true identity of *x*, but never definitely reveal its identity. The *confidence* we can have in our guess or *hypothesis* about the identity of an unknown agent *x*—i.e. the probability that we make a correct guess—depends obviously on two factors: The number of tests *n* and the difference $\varepsilon = \|x_s\mathbf{K}_p^\dagger \mathbf{M}(p)\mathbf{K}_p - x_s\mathbf{K}_q^\dagger \mathbf{M}(q)\mathbf{K}_q\|_1$.

7.2.2. Hypothesis testing

The problem we are faced with is to determine experimentally if an unknown process *x* is one of two known processes *p* and *q*. The only way we can obtain information about *x* is by executing it in state *s*. In this way we can get an experimental estimate for the $x_{s,t}$. We then can compare this estimate with $p_{s,t}$ and $q_{s,t}$.

In other words, based on the outcome of some finite experiments (involving an unknown process *x*) we formulate a hypothesis H about the identity of *x*, namely either that “x is p”

or that “x is q”. Our hypothesis about the identity of x will be formulated according to a simple rule: depending if the experimental estimate for $x_{s,t}$ is closer to $p_{s,t}$ or to $q_{s,t}$ we will identify x with p or q, respectively.

More precisely, the method to formulate the hypothesis H about the identity of the unknown process x consists of the two following steps:

1. We execute x in s exactly n times in order to obtain an experimental approximation, i.e. an average, $x^n_{s,t}$.
2. Depending if $x^n_{s,t}$ is closer to the observables $p_{s,t}$ or $q_{s,t}$ we formulate the hypothesis

$$\mathrm{H}: \begin{cases} x = p & \text{if } \|(x^n_{s,t})_t - (p_{s,t})_t\| \leqslant \|(x^n_{s,t})_t - (q_{s,t})_t\| \text{ or} \\ & \text{if } \|(x^n_{s,t})_t - x_s\mathbf{K}_p^\dagger\mathbf{M}(p)\mathbf{K}_p\| \leqslant \|(x^n_{s,t})_t - x_s\mathbf{K}_q^\dagger\mathbf{M}(q)\mathbf{K}_q\|, \\ x = q & \text{otherwise.} \end{cases}$$

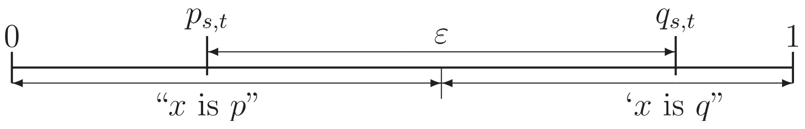
The question is now whether the guess expressed by the hypothesis H about the true identity of the black box x, which we formulate according to the above procedure, is correct; or more precisely: What is the probability that the hypothesis H holds? To do this we have to distinguish two cases or scenarios:

x is actually p: what is the probability (in this case) that we formulate the *correct* hypothesis H =“x is p” and what is the probability that we formulate the *incorrect* hypothesis H =“x is q”?

x is actually q: what is the probability (in this case) that we formulate the *correct* hypothesis H = “x is q” and what is the probability that we formulate the *incorrect* hypothesis H =“x is p”?

Clearly, in each case the probability to formulate a correct hypothesis and the probability to formulate an incorrect hypothesis add up to one. Furthermore, it is obvious that both scenarios “x is actually p” and “x is actually q” are symmetric (just exchange the “names” of the processes p and q). We will therefore investigate only one particular problem: Suppose that x is actually process p, what is the probability that—according to the above procedure—we formulate the—in this case—correct hypothesis H =“x is p”.

In the following we use the notation $x_{s,t}$ and $x^n_{s,t}$ to denote the probability assigned to t in the distribution representing the transitions from s according to the theoretical behaviour of x and in the experimental average, respectively. Furthermore, we look at a simplified situation where we are considering only a single state t. Let us assume without loss of generality that $p_{s,t} < q_{s,t}$ as in the diagram below:



If the experimental value $x^n_{s,t} = p^n_{s,t}$ we obtained in our test is anywhere to the left of $p_{s,t} + \varepsilon/2$ then the hypothesis H we formulate (based on $p^n_{s,t}$) will be the correct one: “x is p”; if the experimental value is to the right of $p_{s,t} + \varepsilon/2$ we will formulate the incorrect hypothesis: “x is q”.

Under the assumption that “x is actually p” the probability $\mathbf{P}(\mathrm{H})$ that we will formulate the correct hypothesis “x is p” is therefore:

$$\mathbf{P}\left(p_{s,t}^{n} < p_{s,t} + \frac{\varepsilon}{2}\right) = 1 - \mathbf{P}\left(p_{s,t} + \frac{\varepsilon}{2} < p_{s,t}^{n}\right).$$

To estimate $\mathbf{P}(\mathrm{H})$ we have just to estimate the probability $\mathbf{P}(p_{s,t}^{n} < p_{s,t} + \varepsilon/2)$, i.e. that the experimental value $p_{s,t}^{n}$ will be left of $p_{s,t} + \varepsilon/2$.

7.2.3. *Confidence estimation*

The confidence we can have in the hypothesis H we formulate is true can be determined by various statistical methods. These methods allow us to estimate the probability that an experimental average X_n—in our case $p_{s,t}^{n}$—is within a certain distance from the corresponding expectation value $\mathbf{E}(X)$—here $p_{s,t}$—i.e. the probability $\mathbf{P}\left(|X_n - \mathbf{E}(X)| \leqslant \varepsilon\right)$ for some $\varepsilon \geqslant 0$. These statistical methods are essentially all based on the *central limit theorem*, e.g. [8,35,62].

The type of tests we consider here to formulate a hypothesis about the identity of the unknown agent X are described in statistical terms by so called *Bernoulli Trials* which are parametric with respect to two probabilities p and $q = 1 - p$. The central limit theorem for this type of tests [35, Theorem 9.2] gives us an estimate for the probability that the experimental value $S_n = n \cdot X_n$ after n repetitions of the test will be in a certain interval $[a, b]$:

$$\lim_{n\to\infty} \mathbf{P}(a \leqslant S_n \leqslant b) = \frac{1}{\sqrt{2\pi}} \int_{a^*}^{b^*} \exp\left(\frac{-x^2}{2}\right) \mathrm{d}x$$

where $a^* = a - np/\sqrt{npq}$ and $b^* = b - np/\sqrt{npq}$.

Unfortunately, the integral of the so called *standard normal density* on the right hand side of the above expression is not easy to obtain. In practical situations one has to resort to numerical methods or statistical tables, but it allows us—at least in principle—to say something about $\mathbf{P}(\mathrm{H})$.

Identifying S_n with $n \cdot p_{s,t}^{n}$ we can utilise the above expression to estimate the probability $\mathbf{P}(p_{s,t} + \varepsilon/2 \leqslant p_{s,t}^{n})$ which determines $\mathbf{P}(\mathrm{H})$. In order to do this we have to take: $a = p_{s,t} + \frac{\varepsilon}{2}$, $b = \infty$, $p = p_{s,t}$ and $q = 1 - p_{s,t}$. This allows us—in principle—to compute the probability:

$$\lim_{n\to\infty} \mathbf{P}\left(p_{s,t} + \frac{\varepsilon}{2} \leqslant p_{s,t}^{n} \leqslant \infty\right).$$

Approximating—as it is common in statistics—$\mathbf{P}(p_{s,t} + \varepsilon/2 \leqslant p_{s,t}^{n})$ by $\lim \mathbf{P}(p_{s,t} + \varepsilon/2 \leqslant p_{s,t}^{n})$ we get

$$\begin{aligned}\mathbf{P}(\mathrm{H}) &= 1 - \mathbf{P}\left(p_{s,t} + \frac{\varepsilon}{2} \leqslant p_{s,t}^{n}\right)\\ &\approx 1 - \lim_{n\to\infty} \mathbf{P}\left(p_{s,t} + \frac{\varepsilon}{2} \leqslant p_{s,t}^{n}\right)\\ &= 1 - \int_{a_0}^{\infty} \exp\left(\frac{-x^2}{2}\right) \mathrm{d}x\end{aligned}$$

with $a_0 = n\varepsilon/2\,1/\sqrt{npq} = \varepsilon\sqrt{n}/2\sqrt{pq} = \varepsilon\sqrt{n}/2\sqrt{p_{s,t}(1 - p_{s,t})}$.

We see that the only way to increase the probability $\mathbf{P}(\mathrm{H})$, i.e. the confidence that we formulate the right hypothesis about the identity of x, is by minimising the integral. In order to do this we have to increase the lower bound a_0 of the integral. This can be achieved—as one would expect—by increasing the number n of experiments.

We can also see that for a smaller ε we have to perform more tests n to reach the same level of confidence, $\mathbf{P}(\mathrm{H})$: The smaller the n the harder it is to distinguish p and q experimentally. Note that for $\varepsilon = 0$, the probability of correctly guessing which of the agents p and q is in the black box is $\frac{1}{2}$, which is the best blind guess we can make anyway. In other words, for $\varepsilon = 0$ we cannot distinguish between p and q.

8. Conclusion and related work

We have investigated probabilistic transition systems (PTS) in a quantitative setting based on linear spaces and linear operators. We have argued that Hilbert spaces are suitable domains for representing countable infinite state spaces, and we have defined a linear operator semantics for probabilistic processes which encode their operational meaning via bounded linear operators on the Hilbert space of the set of processes.

Based on the framework of *probabilistic abstract interpretation*, previously introduced in [56,57] in a finite dimensional setting and then extended in [54] to the infinite case, we also presented a formulation of various (probabilistic) process equivalences in terms of linear operators. This formulation has a very strong resemblance to notions of similarity in mathematical control theory, e.g. [64, Definition 4.1.1]. The relation between abstract interpretation and (bi)simulation has been recognised before in the classical Galois Connection based framework [19,60], but this appears to be the first investigation of such a relation in a probabilistic setting.

More precisely, we have shown how to represent process equivalences via special linear operators corresponding to some probabilistic abstract interpretation of the PTS semantics. For example, the abstraction resulting in the probabilistic bisimulation of Larsen and Skou is a linear operator satisfying Kemeny and Snell’s lumpability condition for Markov chains, while the probabilistic weak bisimulation of Baier and Hermanns can be obtained by an essentially similar technique extended so as to take into account possible looping on τ-transitions.

This formulation made it possible to weaken strict process equivalences to approximate ones which identify two processes up to a quantity ε. This quantity is defined via the norm of an appropriate operator representing the behavioural difference of the two processes according to the given semantics. This norm defines a distance on the set of processes. Other approaches to the definition of such a distance have been proposed in the literature, starting from the work by Giacalone et al. who first suggested the use of a metric to weaken the notion of probabilistic bisimulation. In fact, as far as we are aware, all the approaches which have been proposed since then rely on constructions involving metric spaces. Among them we mention the metrics for probabilistic processes introduced in [66], although it is mainly inspired by semantical considerations and is not meant for approximation purposes. The approach in [10,11] is more similar in its motivation to our work; their technique uses coalgebraic constructions on the category of metric spaces and non-expansive maps and is applied to probabilistic bisimulation only. The pseudometric defined in [22] is also motivated by a weakening of the notion of bisimulation and is based on the logical characterisation of bisimulation for labelled Markov processes in [21]. More recently, this pseudometric has been extended to consider internal non-determinism and weak bisimulation in [23].

Although the use of different mathematical structures makes it difficult a direct comparison with these works, the measure for the distance resulting in our approach seems to be substantially the same: it is zero exactly when processes are (weak) bisimilar. A more important difference is in the methodology used to define process equivalences. In our approach these result in probabilistic abstract interpretations of the underlying Markov chain: two processes are equivalent if there exists a probabilistic abstraction of both. Moreover, as shown in [55], we are able to give a meaning to the quantity measuring the distance between two processes in terms of the number of tests an external observer needs to perform in order to distinguish them. This statistical interpretation comes from a straightforward application of standard methods in mathematical statistics. This interpretation also makes our approach closer to the *extensional* trend in traditional testing theory [37,48], where systems can be distinguished on the base of their interaction with external observers (i.e. tests).

We argued that our notions of approximate similarity have a natural application in security where they can be fruitfully employed for the definition of non-interference based properties. We have shown this use via the notion of *approximate confinement*, which was previously investigated in [52] in a programming language setting. The quantity ε which defines the approximation represents a quantitative measure of the confinement of a system which from a practical viewpoint offers a more meaningful parameter for evaluating the security of a system. Aldini et al.[1] have adopted a similar approach to study probabilistic non-interference in a CSP-like calculus modelled via a generative-reactive transition system. In their work a notion of probabilistic weak bisimulation with ε-precision is introduced, which allows to identify processes with a small difference in their probabilistic behaviour. This difference is defined in terms of the probabilities on the transitions on each action. As shown in [2], computing this difference corresponds in our approach to taking the supremum norm of a vector encoding the difference between the transition probabilities of two processes for each action.

The statistical interpretation of the number ε mentioned above corresponds in the security context to the number of tests needed to a spy to disclose hidden information. In a previous paper [55] we used very similar arguments for the approximation of probabilistic input–output observables. The difference to the current setting is in the nature of tests we allow for. In [55] we were observing the final results in a certain computational context (i.e. a spy); in the current setting we test in each computational step the chances of reaching a certain (equivalence) class of states, depending on the initial state. Other options can be investigated in order to quantify the difference between two processes on the basis of some (observable) probability distributions—be it the final results as in [55] or the single-step distributions as in the current setting—are to consider their *mutual information* [3] or their *Kullback–Leiber information divergence* [67].

We expect that our linear operator approach towards process equivalences may lead also to efficient implementations. A brute force approach (e.g. checking for all possible classification matrices) is prohibitively expensive. Given that the matrix representations of PTS's are typically very sparse, it seems nevertheless possible to combine efficient numerical algorithms—in particular in the area of linear optimisation—and graph based algorithms in order to develop fast algorithms for checking, for example, if two processes are weakly bisimilar. A similar hybrid approach appears to have been successfully applied to probabilistic model checking [42].

References

[1] A. Aldini, M. Bravetti, R. Gorrieri, A process algebraic approach for the analysis of probabilistic noninterference, J. Comput. Security, 2003, to appear.

[2] A. Aldini, A. Di Pierro, A quantitative approach to noninterference for probabilistic systems, in: M. Bravetti, G. Gorrieri (Eds.), Electronic Notes in Theoretical Computer Science, Vol. 99. Elsevier Science, Amsterdam, 2004 (Proc. MEFISTO Project 2003, Formal Methods for Security and Time).

[3] D. Applebaum, Probability and Information—An Integrated Approach, Cambridge University Press, Cambridge, 1996.

[4] C. Baier, H. Hermanns, Weak bisimulation for fully probabilistic processes, in: Proc. 9th Int. Conf. Computer Aided Verification, Lecture Notes in Computer Science, Vol. 1254, Springer, Berlin, 1997, pp. 119–130.

[5] J.A. Bergstra, A. Ponse, S.A. Smolka (Eds.), Handbook of Process Algebra, Elsevier Science, Amsterdam, 2001.

[6] F.J. Beutler, The operator theory of the pseudo-inverse, J. Math. Anal. Appl. 10 (1965) 451–470, 471–493.

[7] N. Biggs, Algebraic Graph Theory, second ed., Cambridge Mathematical Library, Cambridge University Press, Cambridge, 1993.

[8] P. Billingsley, Probability and Measure, second ed., Wiley, New York, 1986.

[9] A. Böttcher, B. Silbermann, Introduction to Large Truncated Toeplitz Matrices, Springer, New York, 1999.

[10] F. van Breugel, J. Worrell, An algorithm for quantitative verification of probabilistic transition systems, in: Proc. CONCUR'01, Lecture Notes in Computer Science, Vol. 2154, Springer, Berlin, 2001.

[11] F. van Breugel, J. Worrell, Towards quantitative verification of probabilistic transition systems, in: Proc. ICALP'01, Lecture Notes in Computer Science, Vol. 2076, Springer, Berlin, 2001, pp. 421–432.

[12] S.L. Campbell, D. Meyer, Generalized Inverse of Linear Transformations, Constable and Company, London, 1979.

[13] H. Cavusoglu, B. Mishra, S. Raghunathan, A model for evaluating IT security investments, Comm. ACM 47 (7) (2004) 87–92.

[14] A.Z.R. Cleaveland, S. Smolka, Testing preorders for probabilistic processes, in: Proc. ICALP 92, Lecture Notes in Computer Science, Vol. 623, Springer, Berlin, 1992, pp. 708–719.

[15] J.B. Conway, A Course in Functional Analysis, second ed., Graduate Texts in Mathematics, Vol. 96, Springer, New York, 1990.

[16] P. Cousot, R. Cousot, Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: Proc. POPL'77, Los Angeles, 1977, pp. 238–252.

[17] P. Cousot, R. Cousot, Systematic design of program analysis frameworks, in: Proc. POPL'79, San Antonio, TX, 1979, pp. 269–282.

[18] P. Cousot, R. Cousot, Abstract interpretation and applications to logic programs, J. Logic Program. 13 (2–3) (1992) 103–180.

[19] D. Dams, R. Gerth, O. Grumberg, Abstract interpretation of reactive systems, ACM Trans. Program. Languages Systems 19 (2) (1997) 253–291.

[20] K.R. Davidson, C*-Algebras by Example, Fields Institute Monographs, Vol. 6, American Mathematical Society, Providence, RI, 1996.

[21] J. Desharnais, A. Edalat, P. Panangaden, Bisimulation for labelled Markov processes, Inform. Comput. 179 (2002) 163–193.

[22] J. Desharnais, R. Jagadeesan, V. Gupta, P. Panangaden, Metrics for labeled Markov systems, in: Proc. 10th Int. Conf. on Concurrency Theory, Lecture Notes in Computer Science, Vol. 1664, Springer, Berlin, 1999, pp. 258–273.

[23] J. Desharnais, R. Jagadeesan, V. Gupta, P. Panangaden, The metric analogue of weak bisimulation for probabilistic processes, in: Proc. LICS'02, , IEEE, Copenhagen, Denmark, 22–25 July 2002, pp. 413–422.

[24] F. Deutsch, Bet approximation in inner product spaces, CMS Books in Mathematics, Vol. 7, Springer, New York, Berlin, 2001.

[25] R. Diestel, Graph theory, Graduate Texts in Mathematics, Vol. 173, Springer, New York, Heidelberg, Berlin, 1997.

[26] E.A. Feinberg, A. Shwartz (Eds.), Handbook of Markov Decision Processes, Kluwer, Dordrecht, 2002.

[27] P.A. Fillmore, A User's Guide to Operator Algebras, Wiley, New York, Chichester, 1996.

[28] R.J. van Glabbeek, The Linear Time—Branching Time Spectrum I. The Semantics of Concrete, Sequential Processes, Elsevier Science, Amsterdam, 2001, pp. 3–99 (Chapter 1).

[29] R.J. van Glabbeek, S.A. Smolka, B. Steffen, Reactive, generative and stratified models of probabilistic processes, Inform. Comput. 121 (1995) 59–80.

[30] C. Godsil, G. Royle, Algebraic graph theory, Graduate Texts in Mathematics, Vol. 207, Springer, New York, Heidelberg, Berlin, 2001.

[31] J. Goguen, J. Meseguer, Security policies and security models, in: IEEE Symp. on Security and Privacy, IEEE Computer Society Press, Rockville, MD, 1982, pp. 11–20.

[32] J.W. Gray III, Towards a mathematical foundation for information flow security, in: Proc.1991 Symp. on Research in Security and Privacy, IEEE, Oakland, CA, May 1991, pp. 21–34.

[33] W.H. Greub, Linear Algebra, third ed., Grundlehren der mathematischen Wissenschaften, Vol. 97, Springer, Berlin, New York, 1967.

[34] G.R. Grimmett, D.R. Stirzaker, Probability and Random Processes, second ed., Clarendon Press, Oxford, 1992.

[35] C.M. Grinstead, J.L. Snell, Introduction to Probability, second revised ed., American Mathematical Society, Providence, RI, 1997.

[36] H. Hansson, Time and probability in formal design of distributed systems, Ph.D. Thesis, Uppsala University, 1994.

[37] M.C.B. Hennessy, Algebraic Theory of Processes, MIT Press, Cambridge, MA, 1988.

[38] B. Jonsson, W. Yi, Compositional testing preorders for probabilistic processes, in: Proc. LICS'95, 1995, pp. 431–443.

[39] B. Jonsson, W. Yi, K.G. Larsen, Probabilistic Extensions of Process Algebras, Elsevier Science, Amsterdam, 2001, pp. 685–710 (Chapter 11).

[40] R.V. Kadison, J.R. Ringrose, Fundamentals of the theory of operator algebras: Vol. I—elementary theory, Graduate Studies in Mathematics, Vol. 15, American Mathematical Society, Providence, RI, 1997 (reprint from Academic Press edition 1983).

[41] J.G. Kemeny, J.L. Snell, Finite Markov Chains, D. Van Nostrand Company, Princeton, NJ, 1960.

[42] M. Kwiatkowska, G. Norman, D. Parker, PRISM: Probabilistic Symbolic Model Checker, in: TOOLS 2002, Lecture Notes in Computer Science, Vol. 2324, Springer, Berlin, 2002, pp. 200–204.

[43] B.W. Lampson, A note on the confinement problem, Comm. ACM 16 (10) (1973) 613–615.

[44] K.G. Larsen, A. Skou, Bisimulation through probabilistic testing, Inform. Comput. 94 (1991) 1–28.

[45] R. Milner, A Calculus of Communicating Systems, Lecture Notes in Computer Science, Vol. 92, Springer, Berlin, New York, 1980.

[46] B. Mohar, W. Woess, A survey on spectra of infinite graphs, Bull. London Math. Soc. 21 (1988) 209–234.

[47] G.J. Murphy, C^*-Algebras and Operator Theory, Academic Press, San Diego, 1990.

[48] R. De Nicola, M.C.B. Hennessy, Testing equivalences for processes, Theoret. Comput. Sci. 34 (1983) 83–133.

[49] F. Nielson, H. Riis Nielson, C. Hankin, Principles of Program Analysis, Springer, Berlin, Heidelberg, 1999.

[50] A. Philippou, I. Lee, O. Sokolsky, Weak bisimulation for probabilistic processes, in: Proc. CONCUR 2000, Lecture Notes in Computer Science, Vol. 1887, Springer, Berlin, 2000, pp. 334–349.

[51] A. Di Pierro, C. Hankin, H. Wiklicky, Probabilistic confinement in a declarative framework, in: Declarative Programming—Selected Papers from AGP 2000, La Havana, Cuba, Electronic Notes in Theoretical Computer Science, Vol. 48, Elsevier, Amsterdam, 2001, pp. 1–23.

[52] A. Di Pierro, C. Hankin, H. Wiklicky, Approximate non-interference, in: Proc. CSFW'02, IEEE, Cape Breton, Canada, 24–26 June 2002, pp. 3–17.

[53] A. Di Pierro, C. Hankin, H. Wiklicky, Approximate confinement under uniform attacks, in: Proc. SAS'02, Lecture Notes in Computer Science, Vol. 2477, Springer, Berlin, 2002, pp. 310–325.

[54] A. Di Pierro, C. Hankin, H. Wiklicky, Quantitative relations and approximate process equivalences, in: R. Amadio, D. Lugiez (Eds.), Proc. CONCUR 2003, 14th Int. Conf. on Concurrency Theory, Lecture Notes in Computer Science, Vol. 2761, Springer, Berlin, 2003, pp. 508–522.

[55] A. Di Pierro, C. Hankin, H. Wiklicky, Approximate non-interference, J. Comput. Security 12 (1) (2004) 37–81.

[56] A. Di Pierro, H. Wiklicky, Concurrent constraint programming: towards probabilistic abstract interpretation, in: Proc. PPDP'00, ACM, Montréal, Canada, 2000, pp. 127–138.

[57] A. Di Pierro, H. Wiklicky, Measuring the precision of abstract interpretations, in: Proc. LOPSTR'00, Lecture Notes in Computer Science, Vol. 2042, Springer, Berlin, 2001, pp. 147–164.

[58] P.Y.A. Ryan, J. McLean, J. Millen, V. Gilgor, Non-interference who needs it? in: Proc. 14th IEEE Computer Security Foundations Workshop, IEEE, Cape Breton, Nova Scotia, Canada, June 2001, pp. 237–238.

[59] P.Y.A. Ryan, S.A. Schneider, Process algebra and non-interference, J. Comput. Security 9 (1,2) (2001) 75–103 (special issue on CSFW-12).

[60] D.A. Schmidt, Binary relations for abstraction and refinement, in: Workshop on Refinement and Abstraction, Amagasaki, Japan, November 1999.

[61] R. Segala, N. Lynch, Probabilistic simulations for probabilistic processes, in: Proc. CONCUR 94, Lecture Notes in Computer Science, Vol. 836, Springer, Berlin, 1994, pp. 481–496.

[62] J. Shao, Mathematical Statistics, Springer Texts in Statistics, Springer, New York, Berlin, Heidelberg, 1999.

[63] G. Smith, Probabilistic noninterference through weak probabilistic bi-simulation, in: Proc. 16th Computer Security Foundations Workshop (CSFW'03), IEEE, 2003, pp. 3–13.

[64] E.D. Sontag, Mathematical Control Theory: Deterministic Finite Dimensional Systems, Texts in Applied Mathematics, Vol. 6, Springer, New York, Heidelberg, Berlin, 1990.

[65] M. Vardi, Automatic verification of probabilistic concurrent finite-state programs, in: Proc. FOCS'85, 1985, pp. 332–344.

[66] E.P. de Vink, J.J.M.M. Rutten, Bisimulation for probabilistic transition systems: a coalgebraic approach, Theoret. Comput. Sci. 221 (1999) 271–293.

[67] J. Whittaker, Graphical Models in Applied Multivariate Statistics, Wiley, Chicester, New York, 1990.

[68] W. Woess, Random walks on infinite graphs and groups, Cambridge Tracts in Mathematics, Vol. 138, Cambridge University Press, Cambridge, 2000.

[69] K. Yosida, Functional Analysis, Springer, Berlin, Heidelberg, New York, 1980.