Probabilistic Bisimulation: Naturally on Distributions*

Holger Hermanns[1], Jan Krčál[1], and Jan Křetínský[2]

[1] Saarland University – Computer Science, Saarbrücken, Germany
{hermanns,krcal}@cs.uni-saarland.de
[2] IST Austria
jan.kretinsky@ist.ac.at

Abstract. In contrast to the usual understanding of probabilistic systems as stochastic processes, recently these systems have also been regarded as transformers of probabilities. In this paper, we give a *natural* definition of strong bisimulation for probabilistic systems corresponding to this view that treats probability *distributions* as first-class citizens. Our definition applies in the same way to discrete systems as well as to systems with uncountable state and action spaces. Several examples demonstrate that our definition refines the understanding of behavioural equivalences of probabilistic systems. In particular, it solves a longstanding open problem concerning the representation of memoryless continuous time by memoryfull continuous time. Finally, we give algorithms for computing this bisimulation not only for finite but also for classes of uncountably infinite systems.

1 Introduction

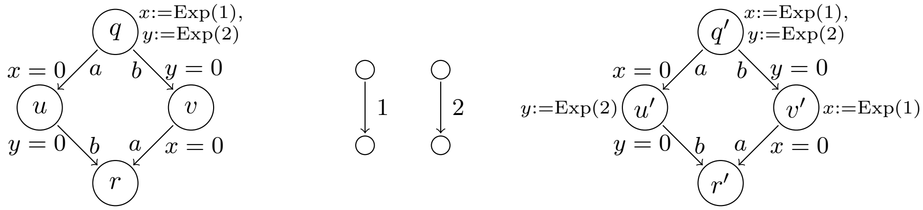
Continuous time concurrency phenomena can be addressed in two principal manners: On the one hand, *timed automata* (TA) extend interleaving concurrency with real-valued clocks [2]. On the other hand, time can be represented by memoryless stochastic time, as in *continuous time Markov chains* (CTMC) and extensions, where time is represented in the form of exponentially distributed random delays [33,30,6,23]. TA and CTMC variations have both been applied to very many intriguing cases, and are supported by powerful real-time, respectively stochastic time model checkers [3,37] with growing user bases. The models are incomparable in expressiveness, but if one extends timed automata with the possibility to sample from exponential distributions [5,10,28], there appears to be

* This work is supported by the EU 7th Framework Programme under grant agreements 295261 (MEALS) and 318490 (SENSATION), European Research Council (ERC) under grant agreement 267989 (QUAREM), Austrian Science Fund (FWF) project S11402-N23 (RiSE), Czech Science Foundation under grant agreement P202/12/G061, the DFG Transregional Collaborative Research Centre SFB/TR 14 AVACS, and by the CAS/SAFEA International Partnership Program for Creative Research Teams. Jan Křetínský is currently on leave from Faculty of Informatics, Masaryk University, Czech Republic.

P. Baldan and D. Gorla (Eds.): CONCUR 2014, LNCS 8704, pp. 249–265, 2014.

a natural bridge from CTMC to TA. This kind of stochastic semantics of timed automata has recently gained considerable popularity by the statistical model checking approach to TA analysis [14,13].

Still there is a disturbing difference, and this difference is the original motivation [12] of the work presented in this paper. The obvious translation of an exponentially distributed delay into a clock expiration sampled from the very same exponential probability distribution fails in the presence of concurrency. This is because the translation is not fully compatible with the natural interleaving concurrency semantics for TA respectively CTMC. This is illustrated by the following example, which in the middle displays two small CTMC, which are supposed to run independently and concurrently.



On the left and right we see two stochastic automata (a variation of timed automata formally defined in Section 3). They have clocks x and y which are initialized by sampling from exponential distributions, and then each run down to 0. The first one reaching 0 triggers a transition and the other clock keeps on running unless resampled, which happens on the right, but not on the left. The left model is obtained by first translating the respective CTMC, and then applying the natural TA interleaving semantics, while the right model is obtained by first applying the equally natural CTMC interleaving semantics prior to translation.

The two models have subtly different semantics in terms of their underlying dense probabilistic timed transition systems. This can superficially be linked to the memoryless property of exponential distributions, yet there is no formal basis for proving equivalence. Our paper closes this gap, which has been open for at least 15 years, by introducing a natural *continuous-space distribution-based* bisimulation. Our result is embedded in several further intriguing application contexts and algorithmic achievements for this novel bisimulation.

The theory of bisimulations is a well-established and elegant framework to describe equivalence between processes based on their behaviour. In the standard semantics of probabilistic systems [38,44], when a probabilistic step from a state to a distribution is taken, the random choice is resolved and we instead continue from one of the successor states. Recently, there has been considerable interest in instead regarding probabilistic systems as deterministic transformers of probability *distributions* [36,1,20], where the choice is not resolved and we continue from the distribution over successors. Thus, instead of the current state the transition changes the current distribution over the states. Although the distribution semantics is very natural in many contexts [29], it has been only partially reflected in the study of bisimulations [29,19,24,23].

Our definition arises as an unusual, but very simple instantiation of the standard coalgebraic framework for bisimulations [42]. (No knowledge of coalgebra is required from the reader though.) Despite its simplicity, the resulting notion is surprisingly fruitful, not only because it indeed solves the longstanding correspondence problem between CTMC and TA with stochastic semantics.

Firstly, it is more adequate than other equivalences when applied to systems with distribution semantics, including large-population models where different parts of the population act differently [39]. Indeed, as argued in [26], some equivalent states are not identified in the standard probabilistic bisimulations and too many are identified in the recent distribution based bisimulations [19,24]. Our approach allows for a bisimulation identifying precisely the desired states [26].

Secondly, our bisimulation over distributions induces an equivalence *on states*, and this relation equates behaviourally indistinguishable states which in many settings are unnecessarily distinguished by standard bisimulations. We shall discuss this phenomenon in the context of several applications. Nevertheless, the key idea to work with distributions instead of single states also bears disadvantages. The main difficulty is that even for finite systems the space of distributions is uncountable, thus bisimulation is difficult to compute. However, we show that it admits a concise representation using methods of linear algebra and we provide an algorithm for computing it. Further, in order to cover e.g. continuous-time systems, we need to handle both uncountably many states (that store the sampled time) and labels (real time durations). Fortunately, there is an elegant way to do so using the standard coalgebra framework. Moreover, it can easily be further generalized, e.g. adding rewards to the generic definition is a trivial task. **Our contribution** is the following:

- We give a natural definition of bisimulation from the distribution perspective for systems with generally uncountable spaces of states and labels.
- We argue by means of several applications that the definition can be considered more useful than the classical notions of probabilistic bisimulation.
- We provide an algorithm to compute this distributional bisimulation on finite non-deterministic probabilistic systems, and present a decision algorithm for uncountable continuous-time systems induced by the stochastic automata mentioned above.

A full version of this paper is available [31].

2 Probabilistic Bisimulation on Distributions

A (potentially uncountable) set S is a *measurable space* if it is equipped with a σ-algebra, denoted by $\Sigma(S)$. The elements of $\Sigma(S)$ are called *measurable sets.* For a measurable space S, let $\mathcal{D}(S)$ denote the set of *probability measures* (or *probability distributions*) over S. The following definition is similar to the treatment of [51].

Definition 1. *A* non-deterministic labelled Markov process *(NLMP) is a tuple* $\mathbf{P} = (S, L, \{\tau_a \mid a \in L\})$ *where S is a measurable space of* states*, L is a measurable* *space of* labels*, and* $\tau_a : S \to \Sigma(\mathcal{D}(S))$ *assigns to each state* s *a measurable set of probability measures* $\tau_a(s)$ *available in* s *under* a*.*[(1)]

When in a state $s \in S$, NLMP reads a label $a \in L$ and *non-deterministically* chooses a successor distribution $\mu \in \mathcal{D}(S)$ that is in the set of convex combinations[(2)] over $\tau_a(s)$, denoted by $s \xrightarrow{a} \mu$. If there is no such distribution, the process halts. Otherwise, it moves into a successor state according to μ. Considering convex combinations is necessary as it gives more power than pure resolution of non-determinism [43].

Example 1. If all sets are finite, we obtain *probabilistic automata (PA)* defined [43] as a triple $(S, L, \longrightarrow)$ where $\longrightarrow \subseteq S \times L \times \mathcal{D}(S)$ is a probabilistic transition relation with $(s, a, \mu) \in \longrightarrow$ if $\mu \in \tau_a(s)$.

Example 2. In the continuous setting, consider a random number generator that also remembers the previous number. We set $L = [0, 1]$, $S = [0, 1] \times [0, 1]$ and $\tau_x(\langle \mathit{new}, \mathit{last} \rangle) = \{\mu_x\}$ for $x = \mathit{new}$ and $\emptyset$ otherwise, where μ_x is the uniform distribution on $[0, 1] \times \{x\}$. If we start with a uniform distribution over S, the measure of successors under any $x \in L$ is 0. Thus in order to get any information of the system we have to consider successors under sets of labels, e.g. intervals.

For a measurable set $A \subseteq L$ of labels, we write $s \xrightarrow{A} \mu$ if $s \xrightarrow{a} \mu$ for some $a \in A$, and denote by $S_A := \{s \mid \exists \mu : s \xrightarrow{A} \mu\}$ the set of states having some outgoing label from A. Further, we can lift this to probability distributions by setting $\mu \xrightarrow{A} \nu$ if $\mu(S_A) > 0$ and $\nu = \frac{1}{\mu(S_A)} \int_{s \in S_A} \nu_s \; \mu(d\, s)$ for some measurable function assigning to each state $s \in S_A$ a measure ν_s such that $s \xrightarrow{A} \nu_s$. Intuitively, in μ we restrict to states that do not halt under A and consider all possible combinations of their transitions; we scale up by $\frac{1}{\mu(S_A)}$ to obtain a distribution again.

Example 3. In the previous example, let υ be the uniform distribution. Due to the independence of the random generator on previous values, we get $\upsilon \xrightarrow{[0,1]} \upsilon$. Similarly, $\upsilon \xrightarrow{[0.1,0.2]} \upsilon_{[0.1,0.2]}$ where $\upsilon_{[0.1,0.2]}$ is uniform on $[0, 1]$ in the first component and uniform on $[0.1, 0.2]$ in the second component, with no correlation.

Using this notation, a non-deterministic and probabilistic system such as NLMP can be regarded as a non-probabilistic, thus solely non-deterministic, labelled transition system over the uncountable space of probability distributions. The natural bisimulation from this distribution perspective is as follows.

Definition 2. *Let* $(S, L, \{\tau_a \mid a \in L\})$ *be a NLMP and* $R \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$ *be a symmetric relation. We say that* R *is a (strong)* probabilistic bisimulation *if for each* $\mu \, R \, \nu$ *and measurable* $A \subseteq L$

[(1)] We further require that for each $s \in S$ we have $\{(a, \mu) | \mu \in \tau_a(s)\} \in \Sigma(L) \otimes \Sigma(\mathcal{D}(S))$ and for each $A \in \Sigma(L)$ and $Y \in \Sigma(\mathcal{D}(S))$ we have $\{s \in S \mid \exists a \in A. \tau_a(s) \cap Y \neq \emptyset\} \in \Sigma(S)$. Here $\Sigma(\mathcal{D}(S))$ is the Giry σ-algebra [27] over $\mathcal{D}(X)$.

[(2)] A distribution $\mu \in \mathcal{D}(S)$ is a *convex combination* of a set $M \in \Sigma(\mathcal{D}(S))$ of distributions if there is a measure ν on $\mathcal{D}(S)$ such that $\nu(M) = 1$ and $\mu = \int_{\mu' \in \mathcal{D}(S)} \mu' \nu(d\mu')$.

1. $\mu(S_A) = \nu(S_A)$, and
2. for each $\mu \xrightarrow{A} \mu'$ there is a $\nu \xrightarrow{A} \nu'$ such that $\mu' \, R \, \nu'$.

We set $\mu \sim \nu$ if there is a probabilistic bisimulation R such that $\mu \, R \, \nu$.

Example 4. Considering Example 2, states $\{x\} \times [0,1]$ form a class of $\sim$ for each $x \in [0,1]$ as the old value does not affect the behaviour. More precisely, $\mu \sim \nu$ iff marginals of their first component are the same.

Naturalness. Our definition of bisimulation is not created ad-hoc as it often appears for relational definitions, but is actually an instantiation of the standard bisimulation for a particular *coalgebra*. Although this aspect is not necessary for understanding the paper, it is another argument for naturalness of our definition. For reader's convenience, we present a short introduction to coalgebras and the formal definitions in [31]. Here we only provide an intuitive explanation by example.

Non-deterministic labelled transition systems are essentially given by the transition function $S \to \mathcal{P}(S)^L$; given a state $s \in S$ and a label $a \in L$, we can obtain the set of the successors $\{s' \in S \mid s \xrightarrow{a} s'\}$. The transition function corresponds to a coalgebra, which induces a bisimulation coinciding with the classical one of Park and Milner [40]. Similarly, PA are given by the transition function $S \to \mathcal{P}(\mathcal{D}(S))^L$; instead of successors there are distributions over successors. Again, the corresponding coalgebraic bisimulation coincides with the classical ones of Larsen and Skou [38] and Segala and Lynch [44].

In contrast, our definition can be obtained by considering states S' to be distributions in $\mathcal{D}(S)$ over the original state space and defining the transition function to be $S' \to ([0,1] \times \mathcal{P}(S'))^{\Sigma(L)}$. The difference to the standard non-probabilistic case is twofold: firstly, we consider all measurable sets of labels, i.e. all elements of $\Sigma(L)$; secondly, for each label set we consider the mass, i.e. element of $[0,1]$, of the current state distribution that does not deadlock, i.e. can perform some of the labels. These two aspects form the crux of our approach and distinguish it from other approaches.

3 Applications

We now argue by some concrete application domains that the distribution view on bisimulation yields a fruitful notion.

Memoryless vs. Memoryfull Continuous Time. First, we reconsider the motivating discussion from Section 1 revolving around the difference between continuous time represented by real-valued clocks, respectively memoryless stochastic time. For this we introduce a simple model of *stochastic automata* [10].

Definition 3. *A* stochastic automaton (SA) *is a tuple $\mathcal{S} = (\mathcal{Q}, \mathcal{C}, \mathcal{A}, \to, \kappa, F)$ where $\mathcal{Q}$ is a set of locations, $\mathcal{C}$ is a set of clocks, $\mathcal{A}$ is a set of actions, $\to \subseteq \mathcal{Q} \times \mathcal{A} \times 2^{\mathcal{C}} \times \mathcal{Q}$ is a set of edges, $\kappa : \mathcal{Q} \to 2^{\mathcal{C}}$ is a clock setting function, and F assigns to each clock its distribution over $\mathbb{R}_{\geq 0}$.*

Avoiding technical details, $\mathcal{S}$ has the following NLMP semantics $\mathbf{P}_\mathcal{S}$ with state space $S = \mathcal{Q} \times \mathbb{R}^\mathcal{C}$, assuming it is initialized in some location q_0: When a location q is entered, for each clock $c \in \kappa(q)$ a positive *value* is chosen randomly according to the distribution $F(c)$ and stored in the state space. Intuitively, the automaton idles in location q with all clock values decreasing at the same speed until some edge (q, a, X, q') becomes *enabled*, i.e. all clocks from X have value ≤ 0. After this *idling time* t, the action a is taken and the automaton enters the next location q'. If an edge is enabled on entering a location, it is taken immediately, i.e. $t = 0$. If more than one edge become enabled simultaneously, one of them is chosen non-deterministically. Its formal definition is given in [31]. We now are in the position to harvest Definition 2, to arrive at the novel bisimulation for stochastic automata.

Definition 4. *We say that locations q_1, q_2 of an SA $\mathcal{S}$ are* probabilistic bisimilar*, denoted $q_1 \sim q_2$, if $\mu_{q_1} \sim \mu_{q_2}$ in $\mathbf{P}_\mathcal{S}$ where μ_q denotes a distribution over the state space of $\mathbf{P}_\mathcal{S}$ given by the location being q, every $c \notin \kappa(q)$ being* 0*, and every $c \in \kappa(q)$ being independently set to a random value according to $F(c)$.*

This bisimulation identifies q and q' from Section 1 unlike any previous bisimulation on SA [10]. In Section 4 we discuss how to compute this bisimulation, despite being continuous-space. Recall that the model initialized by q is obtained by first translating two simple CTMC, and then applying the natural interleaving semantics, while the model, of q' is obtained by first applying the equally natural CTMC interleaving semantics prior to translation. The bisimilarity of these two models generalizes to the whole universe of CTMC and SA:

Theorem 1. *Let $SA(\mathcal{C})$ denote the stochastic automaton corresponding to a CTMC $\mathcal{C}$. For any CTMC $\mathcal{C}_1, \mathcal{C}_2$, we have*

$$SA(\mathcal{C}_1) \parallel_{SA} SA(\mathcal{C}_1) \quad \sim \quad SA(\mathcal{C}_1 \parallel_{CT} \mathcal{C}_1).$$

Here, $\parallel_{CT}$ and $\parallel_{SA}$ denotes the interleaving parallel composition of SA [11] (echoing TA parallel composition) and CTMC [33,30] (Kronecker sum of their matrix representations), respectively.

Bisimulation for Partial-Observation MDP (POMDP). A POMDP is a quadruple $\mathcal{M} = (S, \mathcal{A}, \delta, \mathcal{O})$ where (as in an MDP) S is a set of states, $\mathcal{A}$ is a set of actions, and $\delta : S \times \mathcal{A} \to \mathcal{D}(S)$ is a transition function. Furthermore, $\mathcal{O} \subseteq 2^S$ partitions the state space. The choice of actions is resolved by a policy yielding a Markov chain. Unlike in an MDP, such choice is not based on the knowledge of the current state, only on knowing that the current state belongs into an *observation* $o \in \mathcal{O}$. POMDPs have a wide range of applications in robotic control, automated planning, dialogue systems, medical diagnosis, and many other areas [45].

In the analysis of POMDP, the distributions over states, called *beliefs*, arise naturally and bisimulations over beliefs have already been considered [7,34]. However, to the best of our knowledge, no algorithms for computing belief bisimilation for POMDP exist. We fill this gap by our algorithm for computing distribution bisimulation for PA in Section 4. Indeed, two beliefs μ, ν in POMDP

$\mathcal{M}$ are belief bisimilar in the spirit of [7] iff μ and ν are distribution bisimilar in the induced PA $D_{\mathcal{M}} = (S, \mathcal{O} \times \mathcal{A}, \longrightarrow)$ where $(s, (o, a), \mu) \in \longrightarrow$ if $s \in o$ and $\delta(s, a) = \mu$.[3]

Further Applications. Probabilistic automata are especially apt for compositional modelling of *distributed systems.* The only information a component in a distributed system has about the current state of another component stems from their mutual communication. Therefore, each component can be also viewed from the outside as a partial-observation system. Thus, also in this context, distribution bisimulation is a natural concept. While $\sim$ is not a congruence w.r.t. standard parallel composition, it is apt for compositional modelling of distributed systems where only *distributed schedulers* are considered. For details, see [31,48].

Furthermore we can understand a PA as a description, in the sense of [25,39], of a representative *agent* in a large homogeneous *population.* The distribution view then naturally represents the ratios of agents being currently in the individual states and labels given to this large population of PAs correspond to global control actions [25]. For more details on applications, see [31].

4 Algorithms

In this section, we discuss computational aspects of deciding our bisimulation. Since $\sim$ is a relation over distributions over the system's state space, it is uncountably infinite even for simple finite systems, which makes it in principle intricate to decide. Fortunately, the bisimulation relation has a linear structure, and this allows us to employ methods of linear algebra to work with it effectively. Moreover, important classes of continuous-space systems can be dealt with, since their structure can be exploited. We exemplify this on a subset of deterministic stochastic automata, for which we are able to provide an algorithm to decide bisimilarity.

Finite Systems – Greatest Fixpoints. Let us fix a PA $(S, L, \longrightarrow)$. We apply the standard approach by starting with $\mathcal{D}(S) \times \mathcal{D}(S)$ and pruning the relation until we reach the fixpoint $\sim$. In order to represent $\sim$ using linear algebra, we identify a distribution μ with a vector $(\mu(s_1), \ldots, \mu(s_{|S|})) \in \mathbb{R}^{|S|}$.

Although the space of distributions is uncountable, we construct an implicit representation of $\sim$ by a system of equations written as columns in a matrix E.

Definition 5. *A matrix E with $|S|$ rows is a* bisimulation matrix *if for some bisimulation R, for any distributions μ, ν*

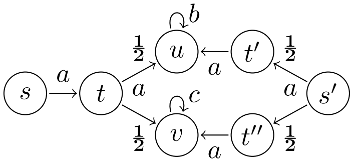
$$\mu \, R \, \nu \quad \textit{iff} \quad (\mu - \nu)E = \mathbf{0}.$$

For a bisimulation matrix E, an equivalence class of μ is then the set $(\mu + \{\rho \mid \rho E = \mathbf{0}\}) \cap \mathcal{D}(S)$, the set of distributions that are equal modulo E.

Example 5. The bisimulation matrix E below encodes that several conditions must hold for two distributions μ, ν to be bisimilar. Among others, if we multiply

[3] Note that [7] also considers rewards that can be easily added to $\sim$ and our algorithm.

$\mu - \nu$ with e.g. the second column, we must get 0. This translates to $(\mu(v) - \nu(v)) \cdot \mathbf{1} = 0$, i.e. $\mu(v) = \nu(v)$. Hence for bisimilar distributions, the measure of v has to be the same. This proves that $u \not\sim v$ (here we identify states and their Dirac distributions). Similarly, we can prove that $t \sim \frac{1}{2}t' + \frac{1}{2}t''$. Indeed, if we multiply the corresponding difference vector $(0, 0, 1, -\frac{1}{2}, -\frac{1}{2}, 0, 0)$ with any column of the matrix, we obtain 0.



$$\begin{array}{c} s: \\ s': \\ t: \\ t': \\ t'': \\ u: \\ v: \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Note that the unit matrix is always a bisimulation matrix, not relating anything with anything but itself. For which bisimulations do there exist bisimulation matrices? We say a relation R over distributions is *convex* if $\mu R \nu$ and $\mu' R \nu'$ imply $\big(p\mu + (1-p)\mu'\big)\ R\ \big(p\nu + (1-p)\nu'\big)$ for any $p \in [0, 1]$.

Lemma 1. *Every convex bisimulation has a corresponding bisimulation matrix.*

Since $\sim$ is convex (see [31]), there is a bisimulation matrix corresponding to $\sim$. It is a least restrictive bisimulation matrix E (note that all bisimulation matrices with the least possible dimension have identical solution space), we call it *minimal bisimulation matrix.* We show that the necessary and sufficient condition for E to be a bisimulation matrix is *stability* with respect to transitions.

Definition 6. *For a $|S| \times |S|$ matrix P, we say that a matrix E with $|S|$ rows is P-*stable *if for every $\rho \in \mathbb{R}^{|S|}$,*

$$\rho E = \mathbf{0} \implies \rho P E = \mathbf{0} \tag{1}$$

We first briefly explain the stability in a simpler setting.

Action-Deterministic Systems. Let us consider PA where in each state, there is at most one transition. For each $a \in L$, we let $P_a = (p_{ij})$ denote the transition matrix such that for all i, j, if there is (unique) transition $s_i \xrightarrow{a} \mu$ we set p_{ij} to $\mu(s_j)$, otherwise to 0. Then μ evolves under a into μP_a. Denote $\mathbf{1} = (1, \ldots, 1)^\top$.

Proposition 1. *In an action-deterministic PA, E containing $\mathbf{1}$ is a bisimulation matrix iff it is P_a-stable for all $a \in L$.*

To get a minimal bisimulation matrix E, we start with a single vector $\mathbf{1}$ which stands for an equation saying that the overall probability mass in bisimilar distributions is the same. Then we repetitively multiply all vectors we have by all the matrices P_a and add each resulting vector to the collection if it is linearly independent of the current collection, until there are no changes. In Example 5, the second column of E is obtained as $P_c\mathbf{1}$, the fourth one as $P_a(P_c\mathbf{1})$ and so on.

The set of all columns of E is thus given by the described iteration

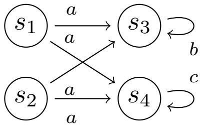
$$\{P_a \mid a \in L\}^* \mathbf{1}$$

modulo linear dependency. Since P_a have $|S|$ rows, the fixpoint is reached within $|S|$ iterations yielding $1 \leq d \leq |S|$ equations. Each class then forms an $(|S| - d)$-dimensional affine subspace intersected with the set of probability distributions $\mathcal{D}(S)$. This is also the principle idea behind the algorithm of [50] and [19].

Non-Deterministic Systems. In general, for transitions under A, we have to consider c_i^A non-deterministic choices in each s_i among all the outgoing transitions under some $a \in A$. We use variables w_i^j denoting the probability that j-th transition, say (s_i, a_i^j, μ_i^j), is taken by the scheduler/player[(4)] in s_i. We sum up the choices into a "non-deterministic" transition matrix P_A^W with parameters W whose ith row equals $\sum_{j=1}^{c_i^A} w_i^j \mu_i^j$. It describes where the probability mass moves from s_i under A depending on the collection W of the probabilities the player gives each choice. By $\mathcal{W}_A$ we denote the set of all such W.

A simple generalization of the approach above would be to consider $\{P_A^W \mid A \subseteq L, W \in \mathcal{W}_A\}^* \mathbf{1}$. However, firstly, the set of these matrices is uncountable whenever there are at least two transitions to choose from. Secondly, not all P_A^W may be used as the following example shows.

Example 6. In each bisimulation class in the following example, the probabilities of $s_1 + s_2$, s_3, and s_4 are constant, as can also be seen from the bisimulation matrix E, similarly to Example 5. Further, E can be obtained as $(\mathbf{1} \;\, P_c\mathbf{1} \;\, P_b\mathbf{1})$. Observe that E is $P_{\{a\}}^W$-stable for W that maximizes the probability of going into the "class" s_3 (both s_1 and s_2 go to s_3, i.e. $w_1^1 = w_2^1 = 1$); similarly for the "class" s_4.



$$P_{\{a\}}^W = \begin{pmatrix} 0 & 0 & w_1^1 & w_2^2 \\ 0 & 0 & w_2^1 & w_2^2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad E = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

However, for $\overline{W}$ with $w_1^1 \neq w_2^1$, e.g. s_1 goes to s_3 and s_2 goes with equal probability to s_3 and s_4 $(w_1^1 = 1, w_2^1 = w_2^2 = \frac{1}{2})$, we obtain from $P_{\{a\}}^{\overline{W}} E$ a new independent vector $(0, 0.5, 0, 0)^\top$ enforcing a partition finer than $\sim$. This does not mean that Spoiler wins the game when choosing such mixed $\overline{W}$ in some μ, it only means that Duplicator needs to choose a *different* W in a bisimilar ν in order to have $\mu P_A^{\overline{W}} \sim \nu P_A^W$ for the successors.

A fundamental observation is that we get the correct bisimulation when Spoiler is restricted to finitely many "extremal" choices and Duplicator is restricted for such extremal W to respond only with the very same W. $(*)$

To this end, consider $M_A^W = P_A^W E$ where E is the current matrix with each of e columns representing an equation. Intuitively, the ith row of M_A^W describes how

[(4)] We use the standard notion of Spoiler-Duplicator bisimulation game (see e.g. [42]) where in $\{\mu_0, \mu_1\}$ Spoiler chooses $i \in \{0, 1\}$, $A \subseteq L$, and $\mu_i \xrightarrow{A} \mu_i'$, Duplicator has to reply with $\mu_{1-i} \xrightarrow{A} \mu_{1-i}'$ such that $\mu_i(S_A) = \mu_{i-1}(S_A)$, and the game continues in $\{\mu_0', \mu_1'\}$. Spoiler wins iff at some point Duplicator cannot reply.

much of s_i is moved to various classes when a step is taken. Denote the linear forms in M_A^W over W by m_{ij}. Since the players can randomize and mix choices which transition to take, the set of vectors $\{(m_{i1}(w_i^1,\ldots,w_i^{c_i}),\ldots,m_{ib}(w_i^1,\ldots,w_i^{c_i})) \mid w_i^1,\ldots,w_i^{c_i} \geq 0, \sum_{j=1}^{c_i} w_i^j = 1\}$ forms a convex polytope denoted by C_i. Each vector in C_i is thus the ith row of the matrix M_A^W where some concrete weights w_i^j are "plugged in". This way C_i describes all the possible choices in s_i and their effect on where the probability mass is moved.

Denote vertices (extremal points) of a convex polytope P by $\mathcal{E}(P)$. Then $\mathcal{E}(C_i)$ correspond to pure (non-randomizing) choices that are "extremal" w.r.t. E. Note that now if $s_j \sim s_k$ then $C_j = C_k$, or equivalently $\mathcal{E}(C_j) = \mathcal{E}(C_k)$. Indeed, for every choice in s_j there needs to be a matching choice in s_k and vice versa. However, since we consider bisimulation between generally non-Dirac distributions, we need to combine these extremal choices. For an arbitrary distribution $\mu \in \mathcal{D}(S)$, we say that a tuple $c \in \prod_{i=1}^{|S|} \mathcal{E}(C_i)$ is *extremal in* μ if $\mu \cdot c^\top$ is a vertex of the polytope $\{\mu \cdot c'^\top \mid c' \in \prod_{i=1}^{|S|} C_i\}$. Note that each extremal c corresponds to particular pure choices, denoted by $W(c)$. Unfortunately, for choices $W(c)$ of Spoiler extremal in *some* distribution, Duplicator may in *another* distribution need to make different choices. Indeed, in Example 6 the tuple corresponding to $\overline{W}$ is extremal in the Dirac distribution of state s_1. Therefore, we define $\mathcal{E}(C)$ to be the set of tuples c extremal in the uniform distribution. Interestingly, tuples extremal in the uniform distribution are (1) extremal in all distributions and (2) reflect all extremal choices, i.e. for every c extremal in some μ, there is a c' extremal in the uniform distribution such that c' is also extremal in μ and $\mu \cdot c = \mu \cdot c'$. As a result, the fundamental property $(*)$ is guaranteed.

Proposition 2. *Let E be a matrix containing* $\mathbf{1}$. *It is a bisimulation matrix iff it is $P_A^{W(c)}$-stable for all $A \subseteq L$ and $c \in \mathcal{E}(C)$.*

```
Input   : Probabilistic automaton $(S, L, \rightarrow)$
Output  : A minimal bisimulation matrix $E$

foreach $A \subseteq L$ do
    compute $P_A^W$                          // non-deterministic transition matrix
$E \leftarrow (\mathbf{1})$
repeat
    foreach $A \subseteq L$ do
        $M_A^W \leftarrow P_A^W E$                 // polytope of all choices
        compute $\mathcal{E}(C)$ from $M_A^W$        // vertices, i.e. extremal choices
        foreach $c \in \mathcal{E}(C)$ do
            $M_A^{W(c)} \leftarrow M_A^W$ with values $W(c)$ plugged in
            $E_{new} \leftarrow$ columns of $M_A^{W(c)}$ linearly independent of columns of $E$
            $E \leftarrow (E\ E_{new})$
until $E$ does not change
```

Algorithm 1: Bisimulation on probabilistic automata

Theorem 2. *Algorithm 1 computes a minimal bisimulation matrix.*

The running time is exponential. We leave the question whether linear programming or other methods [32] can yield E in polynomial time open. The algorithm can easily be turned into one computing other bisimulation notions from the literature, for which there were no algorithms so far, see Section 5.

Continuous-Time Systems - Least Fixpoints. Turning our attention to continuous systems, we finally sketch an algorithm for deciding bisimulation $\sim$ over a subclass of stochastic automata, this constitutes the first algorithm to compute a bisimulation on the uncountably large semantical object.

We need to adopt two restrictions. First, we consider only *deterministic* SA, where the probability that two edges become enabled at the same time is zero (when initiated in any location). Second, to simplify the exposition, we restrict all distributions occurring to exponential distributions. Notably, even for this class, our bisimulation is strictly coarser than the one induced by standard bisimulations [33,30,6] for continuous-time Markov chains. At the end of the section we discuss possibilities for extending the class of supported distributions. Both the restrictions can be effectively checked on SA.

Theorem 3. *Let $\mathcal{S} = (\mathcal{Q}, \mathcal{C}, \mathcal{A}, \dashrightarrow, \kappa, F)$ be a deterministic SA over exponential distributions. There is an algorithm to decide in time polynomial in $|\mathcal{S}|$ and exponential in $|\mathcal{C}|$ whether $q_1 \sim q_2$ for any locations q_1, q_2.*

The rest of the section deals with the proof. We fix $\mathcal{S} = (\mathcal{Q}, \mathcal{C}, \mathcal{A}, \dashrightarrow, \kappa, F)$ and $q_1, q_2 \in \mathcal{Q}$. First, we straightforwardly abstract the NLMP semantics $\mathbf{P}_{\mathcal{S}}$ over state space $S = \mathcal{Q} \times \mathbb{R}^{\mathcal{C}}$ by a NLMP $\hat{\mathbf{P}}$ over state space $\hat{S} = \mathcal{Q} \times (\mathbb{R}_{\geq 0} \cup \{-\})^{\mathcal{C}}$ where all negative values of clocks are expressed by $-$. Let ξ denote the obvious mapping of distributions $\mathcal{D}(S)$ onto $\mathcal{D}(\hat{S})$. Then ξ preserves bisimulation since two states s_1, s_2 that differ only in negative values satisfy $\xi(\tau_a(s_1)) = \xi(\tau_a(s_2))$ for all $a \in L$.

Lemma 2. *For any distributions μ, ν on S we have $\mu \sim \nu$ iff $\xi(\mu) \sim \xi(\nu)$.*

Second, similarly to an embedded Markov chain of a CTMC, we further abstract the NLMP $\hat{\mathbf{P}}$ by a *finite* deterministic PA $\bar{D} = (\bar{S}, \mathcal{A}, \longrightarrow)$ such that each state of $\bar{D}$ is a distribution over the uncountable state space $\hat{S}$.

- The set $\bar{S}$ is the set of states reachable via the transitions relation defined below from the distributions μ_{q_1}, μ_{q_2} corresponding to q_1, q_2 (see Definition 4).
- Let us fix a state $\mu \in \bar{S}$ (note that $\mu \in \mathcal{D}(\hat{S})$) and an action $a \in \mathcal{A}$ such that in the NLMP $\hat{\mathbf{P}}$ an a-transition occurs with positive probability, i.e. $\mu \xrightarrow{A_a} \nu$ for some ν and for $A_a = \{a\} \times \mathbb{R}_{\geq 0}$. Thanks to restricting to deterministic SA, $\hat{\mathbf{P}}$ is also deterministic and such a distribution ν is uniquely defined. We set $(\mu, a, M) \in \longrightarrow$ where M is the discrete distribution that assigns probability $p_{q,f}$ to state $\nu_{q,f}$ for each $q \in \mathcal{Q}$ and $f : \mathcal{C} \to \{-, +\}$ where $p_{q,f} = \nu(\hat{S}_{q,f})$, $\nu_{q,f}$ is the conditional distribution $\nu_q(X) := \nu(X \cap \hat{S}_{q,f})/\nu(\hat{S}_{q,f})$ for any measurable $X \subseteq \hat{S}$, and $\hat{S}_{q,f} = \{(q', v) \in \hat{S} \mid q' = q, v(c) \geq 0 \text{ iff } f(c) = + \text{ for each } c \in \mathcal{C}\}$ the set of states with location q and where the sign of clock values matches f.

For exponential distributions all the reachable states $\nu \in \bar{S}$ correspond to some location q where the subset $X \subseteq \mathcal{C}$ is newly sampled, hence we obtain:

Lemma 3. *For a deterministic SA over exponential distributions,* $|\bar{S}| \leq |\mathcal{Q}| \cdot 2^{|\mathcal{C}|}$.

Instead of a greatest fixpoint computation as employed for the discrete algorithm, we take a complementary approach and prove or disprove bisimilarity by a least fixpoint procedure. We start with the initial pair of distributions (states in $\bar{D}$) which generates further requirements that we impose on the relation and try to satisfy them. We work with a *tableau*, a rooted tree where each node is either an *inner node* with a pair of discrete probability distributions over states of $\bar{D}$ as a label, a *repeated node* with a label that already appears somewhere between the node and the root, or a *failure node* denoted by $\square$, and the children of each inner node are obtained by one *rule* from $\{\mathbf{Step}, \mathbf{Lin}\}$. A tableau not containing $\square$ is *successful.*

Step. For a node $\mu \sim \nu$ where μ and ν have *compatible timing*, we add for each label $a \in L$ one child node $\mu_a \sim \nu_a$ where μ_a and ν_a are the unique distributions such that $\mu \xrightarrow{a} \mu_a$ and $\nu \xrightarrow{a} \nu_a$. Otherwise, we add one failure node. We say that μ and ν have compatible timing if for all actions $a \in \mathcal{A}$ we have that $T_a[\mu]$ is equivalent to $T_a[\nu]$. Here $T_a[\rho]$ is a measure over $\mathbb{R}_{\geq 0}$ such that $T_a[\rho](I) := \rho(\hat{S}_{\{a\}\times I})$, i.e. the measure of states moving after time in I with action a.

Lin. For a node $\mu \sim \nu$ linearly dependent on the set of remaining nodes in the tableau, we add one child (repeat) node $\mu \sim \nu$. Here, we understand each node $\mu \sim \nu$ as a vector $\mu - \nu$ in the $|S_{\mathcal{S}}|$-dimensional vector space.

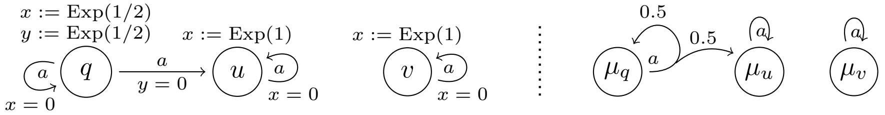
Note that compatibility of timing is easy to check. Furthermore, the set of rules is correct and complete w.r.t. bisimulation in $\hat{\mathbf{P}}$.

Lemma 4. *There is a successful tableau from* $\mu \sim \nu$ *iff* $\mu \sim \nu$ *in* $\hat{\mathbf{P}}$*. Moreover, the set of nodes of a successful tableau is a subset of a bisimulation.*

We get Theorem 3 since $q_1 \sim q_2$ iff $\xi(\mu_{q_1}) \sim \xi(\mu_{q_2})$ in $\hat{\mathbf{P}}$ and since, thanks to **Lin**:

Lemma 5. *There is a successful tableau from* $\mu \sim \nu$ *iff there is a finite successful tableau from* $\mu \sim \nu$ *of size polynomial in* $|\bar{S}|$.

Example 7. Let us demonstrate the rules by a simple example. Consider the following stochastic automaton $\mathcal{S}$ on the left.



Thanks to the exponential distributions, $\bar{D}$ on the right has also only three states where $\mu_q = q \otimes \mathit{Exp}(1/2) \otimes \mathit{Exp}(1/2)$ is the product of two exponential distributions with rate $1/2$, $\mu_u = u \otimes \mathit{Exp}(1)$, and $\mu_v = v \otimes \mathit{Exp}(1)$. Note that for both clocks x and y, the probability of getting to zero first is 0.5.

$$\dfrac{1\cdot\mu_u \;\sim\; 1\cdot\mu_v}{1\cdot\mu_u \;\sim\; 1\cdot\mu_v}\,\textbf{Step} \qquad\qquad \begin{array}{c} 1\cdot\mu_q + 0\cdot\mu_u \;\sim\; 1\cdot\mu_v \\ \hline \frac{1}{2}\cdot\mu_q + \frac{1}{2}\cdot\mu_u \;\sim\; 1\cdot\mu_v \\ \hline \frac{1}{4}\cdot\mu_q + \frac{3}{4}\cdot\mu_u \;\sim\; 1\cdot\mu_v \\ \hline \dots \end{array}\begin{array}{l} \\ \textbf{Step} \\ \textbf{Step} \\ \textbf{Step} \end{array}$$

The finite tableau on the left is successful since it ends in a repeated node, thus it proves $u \sim v$. The infinite tableau on the right is also successful and proves $q \sim v$. When using only the rule **Step**, it is necessarily infinite as no node ever repeats. The rule **Lin** provides the means to truncate such infinite sequences. Observe that the third node in the tableau on the right above is linearly dependent on its ancestors.

Remark 1. Our approach can be turned into a complete proof system for bisimulation on models with *expolynomial* distributions [(5)]. For them, the states of the discrete transition system $\bar{D}$ can be expressed symbolically. In fact, we conjecture that the resulting semi-algorithm can be twisted to a decision algorithm for this expressive class of models. This is however out of the scope of this paper.

5 Related Work and Discussion

For an overview of coalgebraic work on probabilistic bisimulations we refer to a survey [46]. A considerable effort has been spent to extend this work to continuous-space systems: the solution of [15] (unfortunately not applicable to $\mathbb{R}$), the construction of [21] (described by [42] as "ingenious and intricate"), sophisticated measurable selection techniques in [18], and further approaches of [17] or [51]. In contrast to this standard setting where relations between states and their successor distributions must be handled, our work uses directly relations on distributions which simplifies the setting. The coalgebraic approach has also been applied to trace semantics of uncountable systems [35]. The topic is still very lively, e.g. in the recent [41] a different coalgebraic description of the classical probabilistic bisimulation is given.

Recently, distribution-based bisimulations have been studied. In [19], a bisimulation is defined in the context of language equivalence of Rabin's deterministic probabilistic automata and also an algorithm to compute the bisimulation on them. However, only finite systems with no non-determinism are considered. The most related to our notion are the very recent independently developed [24] and [48]. However, none of them is applicable in the continuous setting and for neither of the two any algorithm has previously been given. Nevertheless, since they are close to our definition, our algorithm with only small changes can actually compute them. Although the bisimulation of [24] in a rather complex way extends [19] to the non-deterministic case reusing their notions, it can be equivalently rephrased as our Definition 2 only considering singleton sets $A \subseteq L$.

[(5)] With density that is positive on an interval $[\ell, u)$ for $\ell \in \mathbb{N}_0$, $u \in \mathbb{N} \cup \{\infty\}$ given piecewise by expressions of the form $\sum_{i=0}^{I} \sum_{j=0}^{J} a_{ij} x^i e^{-\lambda_{ij} x}$ for $a_{ij}, \lambda_{ij} \in \mathbb{R} \cup \{\infty\}$. This class contains many important distributions such as exponential, or uniform, and enables efficient approximation of others.

Therefore, it is sufficient to only consider matrices P_A^W for singletons A in our algorithm. Apart from being a weak relation, the bisimulation of [48] differs in the definition of $\mu \xrightarrow{A} \nu$: instead of restricting to the states of the support that can perform *some* action of A, it considers those states that can perform *exactly* actions of A. Here each ith row of each transition matrix P_A^W needs to be set to zero if the set of labels from s_i is different from A.

There are also bisimulation relations over distributions defined over finite [9,29] or uncountable [8] state spaces. They, however, coincide with the classical [38] on Dirac distributions and are only directly lifted to non-Dirac distributions. Thus they fail to address the motivating correspondence problem from Section 1. Moreover, no algorithms were given. Further, weak bisimulations [23,22,16] (coarser than usual state based analogues) applied to models without internal transitions also coincide with lifting [29] of the classical bisimulation [38] while our bisimulation is coarser.

There are other bisimulations that identify more states than the classical [38] such as [47] and [4] designed to match a specific logic. Another approach to obtain coarser equivalences on probabilistic automata is via testing scenarios [49].

6 Conclusion

We have introduced a general and natural notion of a distribution-based probabilistic bisimulation, have shown its applications in different settings and have provide algorithms to compute it for finite and some classes of infinite systems. As to future work, the precise complexity of the finite case is certainly of interest. Further, the tableaux decision method opens the arena for investigating wider classes of continuous-time systems where the new bisimulation is decidable.

Acknowledgement. We would like to thank Pedro D'Argenio, Filippo Bonchi, Daniel Gebler, and Matteo Mio for valuable feedback and discussions.

References

1. Agrawal, M., Akshay, S., Genest, B., Thiagarajan, P.: Approximate verification of the symbolic dynamics of Markov chains. In: LICS (2012)
2. Alur, R., Dill, D.: A theory of timed automata. Theor. Comput. Sci. 126(2), 183–235 (1994)
3. Behrmann, G., David, A., Larsen, K.G., Pettersson, P., Yi, W.: Developing UPPAAL over 15 years. Softw., Pract. Exper. 41(2), 133–142 (2011)
4. Bernardo, M., Nicola, R.D., Loreti, M.: Revisiting bisimilarity and its modal logic for nondeterministic and probabilistic processes. Technical Report 06, IMT Lucca (2013)
5. Bravetti, M., D'Argenio, P.R.: Tutte le algebre insieme: Concepts, discussions and relations of stochastic process algebras with general distributions. In: Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.-P., Siegle, M. (eds.) Validation of Stochastic Systems. LNCS, vol. 2925, pp. 44–88. Springer, Heidelberg (2004)
6. Bravetti, M., Hermanns, H., Katoen, J.-P.: YMCA: Why Markov Chain Algebra? Electr. Notes Theor. Comput. Sci. 162, 107–112 (2006)

7. Castro, P., Panangaden, P., Precup, D.: Equivalence relations in fully and partially observable Markov decision processes. In: IJCAI (2009)
8. Cattani, S.: Trace-based Process Algebras for Real-Time Probabilistic Systems. PhD thesis, University of Birmingham (2005)
9. Crafa, S., Ranzato, F.: A spectrum of behavioral relations over lTSs on probability distributions. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 124–139. Springer, Heidelberg (2011)
10. D'Argenio, P., Katoen, J.-P.: A theory of stochastic systems, part I: Stochastic automata. Inf. Comput. 203(1), 1–38 (2005)
11. D'Argenio, P., Katoen, J.-P.: A theory of stochastic systems, part II: Process algebra. Inf. Comput. 203(1), 39–74 (2005)
12. D'Argenio, P.R., Baier, C.: What is the relation between CTMC and TA? Personal Communication (1999)
13. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., van Vliet, J., Wang, Z.: Statistical model checking for networks of priced timed automata. In: Fahrenberg, U., Tripakis, S. (eds.) FORMATS 2011. LNCS, vol. 6919, pp. 80–96. Springer, Heidelberg (2011)
14. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Wang, Z.: Time for statistical model checking of real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 349–355. Springer, Heidelberg (2011)
15. de Vink, E.P., Rutten, J.J.M.M.: Bisimulation for probabilistic transition systems: A coalgebraic approach. In: Degano, P., Gorrieri, R., Marchetti-Spaccamela, A. (eds.) ICALP 1997. LNCS, vol. 1256, pp. 460–470. Springer, Heidelberg (1997)
16. Deng, Y., Hennessy, M.: On the semantics of Markov automata. Inf. Comput. 222, 139–168 (2013)
17. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Approximating labeled Markov processes. In: LICS (2000)
18. Doberkat, E.-E.: Semi-pullbacks and bisimulations in categories of stochastic relations. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 996–1007. Springer, Heidelberg (2003)
19. Doyen, L., Henzinger, T., Raskin, J.-F.: Equivalence of labeled Markov chains. Int. J. Found. Comput. Sci. 19(3), 549–563 (2008)
20. Doyen, L., Massart, T., Shirmohammadi, M.: Limit synchronization in Markov decision processes. CoRR, abs/1310.2935 (2013)
21. Edalat, A.: Semi-pullbacks and bisimulation in categories of Markov processes. Mathematical Structures in Computer Science 9(5), 523–543 (1999)
22. Eisentraut, C., Hermanns, H., Krämer, J., Turrini, A., Zhang, L.: Deciding bisimilarities on distributions. In: QEST (2013)
23. Eisentraut, C., Hermanns, H., Zhang, L.: On probabilistic automata in continuous time. In: LICS (2010)
24. Feng, Y., Zhang, L.: When equivalence and bisimulation join forces in probabilistic automata. In: Jones, C., Pihlajasaari, P., Sun, J. (eds.) FM 2014. LNCS, vol. 8442, pp. 247–262. Springer, Heidelberg (2014)
25. Gast, N., Gaujal, B.: A mean field approach for optimization in discrete time. Discrete Event Dynamic Systems 21(1), 63–101 (2011)
26. Georgievska, S., Andova, S.: Probabilistic may/must testing: Retaining probabilities by restricted schedulers. Formal Asp. Comput. 24(4-6), 727–748 (2012)
27. Giry, M.: A categorical approach to probability theory. In: Banaschewski, B. (ed.) Categorical Aspects of Topology and Analysis. LNCS, vol. 915, pp. 68–85. Springer, Heidelberg (1982)

28. Harrison, P.G., Strulo, B.: Spades - a process algebra for discrete event simulation. J. Log. Comput. 10(1), 3–42 (2000)
29. Hennessy, M.: Exploring probabilistic bisimulations, part I. Formal Asp. Comput. (2012)
30. Hermanns, H., Herzog, U., Mertsiotakis, V.: Stochastic process algebras - between LOTOS and Markov chains. Computer Networks 30(9-10), 901–924 (1998)
31. Hermanns, H., Krčál, J., Křetínský, J.: Probabilistic bisimulation: Naturally on distributions. CoRR, abs/1404.5084 (2014)
32. Hermanns, H., Turrini, A.: Deciding probabilistic automata weak bisimulation in polynomial time. In: FSTTCS (2012)
33. Hillston, J.: A Compositional Approach to Performance Modelling. Cambridge University Press, New York (1996)
34. Jansen, D.N., Nielson, F., Zhang, L.: Belief bisimulation for hidden Markov models - logical characterisation and decision algorithm. In: Goodloe, A.E., Person, S. (eds.) NFM 2012. LNCS, vol. 7226, pp. 326–340. Springer, Heidelberg (2012)
35. Kerstan, H., König, B.: Coalgebraic trace semantics for probabilistic transition systems based on measure theory. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 410–424. Springer, Heidelberg (2012)
36. Korthikanti, V., Viswanathan, M., Agha, G., Kwon, Y.: Reasoning about MDPs as transformers of probability distributions. In: QEST (2010)
37. Kwiatkowska, M., Norman, G., Parker, D.: Prism 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011)
38. Larsen, K., Skou, A.: Bisimulation through probabilistic testing. In: POPL (1989)
39. May, R., et al.: Biological populations with nonoverlapping generations: stable points, stable cycles, and chaos. Science 186(4164), 645–647 (1974)
40. Milner, R.: Communication and concurrency. PHI Series in computer science. Prentice Hall (1989)
41. Mio, M.: Upper-expectation bisimilarity and Lukasiewicz μ-calculus. In: Muscholl, A. (ed.) FOSSACS 2014. LNCS, vol. 8412, pp. 335–350. Springer, Heidelberg (2014)
42. Sangiorgi, D., Rutten, J.: Advanced Topics in Bisimulation and Coinduction, 1st edn. Cambridge University Press, New York (2011)
43. Segala, R.: Modeling and Verification of Randomized Distributed Real-time Systems. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1995)
44. Segala, R., Lynch, N.: Probabilistic simulations for probabilistic processes. In: Jonsson, B., Parrow, J. (eds.) CONCUR 1994. LNCS, vol. 836, pp. 481–496. Springer, Heidelberg (1994)
45. Shani, G., Pineau, J., Kaplow, R.: A survey of point-based pomdp solvers. AAMAS 27(1), 1–51 (2013)
46. Sokolova, A.: Probabilistic systems coalgebraically: A survey. Theor. Comput. Sci. 412(38), 5095–5110 (2011)
47. Song, L., Zhang, L., Godskesen, J.C.: Bisimulations meet PCTL equivalences for probabilistic automata. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 108–123. Springer, Heidelberg (2011)
48. Song, L., Zhang, L., Godskesen, J.C.: Late weak bisimulation for Markov automata. CoRR, abs/1202.4116 (2012)
49. Stoelinga, M., Vaandrager, F.: A testing scenario for probabilistic automata. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 464–477. Springer, Heidelberg (2003)

50. Tzeng, W.: A polynomial-time algorithm for the equivalence of probabilistic automata. SIAM J. Comput. 21(2), 216–227 (1992)
51. Wolovick, N.: Continuous probability and nondeterminism in labeled transaction systems. PhD thesis, Universidad Nacional de Córdoba (2012)