The Power of (Convex) Algebras

Ana Sokolova UNIVERSITY of SALZBURG

joint work with
Filippo Bonchi ENS de Lyon and Alexandra Silva UCL

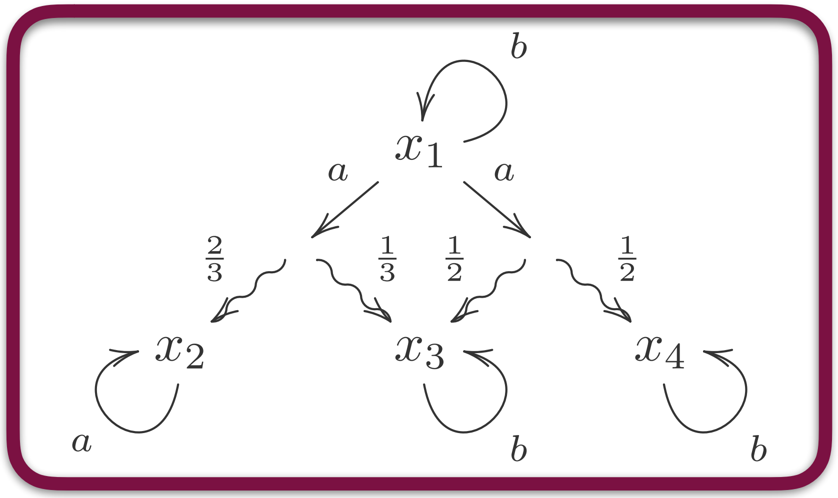
IFIP, WG 1.3. Meeting Binz Rügen, 9.1.17-12.1.17

probabilistic automata

The true nature of PA as transformers of belief states

(co)algebraic

Probabilistic automata I



semantics: probabilistic bisimulation

- **Set** -coalgebras $S \xrightarrow{c} \mathcal{P}(L \times \mathcal{D}S)$ or $S \to (\mathcal{P}\mathcal{D}S)^L$

- Lately also various alternative descriptions of type

belief states

$$\mathcal{D}S \to (\mathcal{P}\mathcal{D}S)^L$$

$$\begin{array}{c} \frac{1}{3}x_1 + \frac{2}{3}x_2 \\ \downarrow a \\ \frac{2}{3}x_2 + \frac{1}{6}x_3 + \frac{1}{6}x_4 \end{array}$$

$$\begin{array}{c} \frac{1}{3}x_1 + \frac{2}{3}x_2 \\ \downarrow a \\ \frac{8}{9}x_2 + \frac{1}{9}x_3 \end{array}$$

Transformers of belief states

$\mathcal{D}S \to (\mathcal{P}\mathcal{D}S)^L$

arXiv:1404.5084v2 [cs.LO] 9 May 2014

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 long-standing open problem concerning the representation of memoryless continuous time by memory-full 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 [37,35,6,26]. 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 [3,42] with growing user bases. The models are incomparable in expressiveness, but if one extends timed automata with the possibility to sample from exponential distributions [5,12,33], there appears to be 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 [16,15].

Still there is a disturbing difference, and this difference is the original motivation [14] 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

DOI 10.1007/s00165-012-0242-7

Formal Aspects of Computing (2012) 24: 749–768

Formal Aspects of Computing

Exploring probabilistic bisimulations, part I

Matthew Hennessy

Department of Computer Science, Trinity College, Dublin 1, Ireland

Abstract. We take a fresh look at strong probabilistic bisimulations for processes which exhibit both non-deterministic and probabilistic behaviour. We suggest that it is natural to interpret such processes as distributions over states in a probabilistic labelled transition system, a pLTS; this enables us to adapt the standard notion of contextual equivalence to this setting. We then prove that a novel form of bisimulation equivalence between distributions are both sound and complete with respect to this contextual equivalence. We also show that a very simple extension to HML, Hennessy–Milner Logic, provides finite explanations for inequivalences between distributions. Finally we show that our bisimulations between distributions in a pLTS are simply an alternative characterisation of a standard notion of probabilistic bisimulation equivalence, defined between states in a pLTS.

Keywords: Probabilistic processes, Contextual equivalence, Bisimulation equivalence, Logical characterisation, Equational theory

1. Introduction

Bisimulations [Mil89] provide a well-established and elegant theory of the behaviour of non-deterministic processes. Let us review the framework.

(1) *Labelled transitions systems, LTSs*: These provide an intensional semantics for processes, describing their computations or more generally the interactions between processes and their environment.
(2) *Process calculi*: Formal description languages for describing processes and their specifications. These usually consist of a small number of *combinators* with which processes can be described by their structure.
(3) *Behavioural equivalence*: This determines which process descriptions are extensionally equivalent; that is which processes can not be distinguished by their users or more generally their environments. Perhaps the most uncontroversial is the so-called *contextual equivalence*, $\sim_{cxt}$ [MS92, SW01, HY95, Hen07], defined in terms of simple properties one would expect of a behavioural equivalence.
(4) *Bisimulations*: These are relations between processes which satisfy simple properties expressed in terms of the intensional semantics. They provide an elegant proof methodology for demonstrating process equivalence; to show two processes behaviourally equivalent it is sufficient to exhibit a witness bisimulation containing them. In many settings this proof method is not only sound with respect to contextual equivalence but also complete.

Transformers of belief states $\mathcal{D}S \to (\mathcal{P}\mathcal{D}S)^L$

When Equivalence and Bisimulation Join Forces in Probabilistic Automata*

Yuan Feng[1,2] and Lijun Zhang[3]

[1] University of Technology Sydney, Australia
[2] Department of Computer Science and Technology, Tsinghua University, China
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

Abstract. Probabilistic automata were introduced by Rabin in 1963 as language acceptors. Two automata are equivalent if and only if they accept each word with the same probability. On the other side, in the process algebra community, probabilistic automata were re-proposed by Segala in 1995 which are more general than Rabin's automata. Bisimulations have been proposed for Segala's automata to characterize the equivalence between them. So far the two notions of equivalences and their characteristics have been studied most independently. In this paper, we consider Segala's automata, and propose a novel notion of distribution-based bisimulation by joining the existing equivalence and bisimilarities. Our bisimulation bridges the two closely related concepts in the community, and provides a uniform way of studying their characteristics. We demonstrate the utility of our definition by studying distribution-based bisimulation metrics, which gives rise to a robust notion of equivalence for Rabin's automata.

1 Introduction

In 1963, Rabin [29] introduced the model *probabilistic automata* as language acceptors. In a probabilistic automaton, each input symbol determines a stochastic transition matrix over the state space. Starting with the initial distribution, each word (a sequence of symbols) has a corresponding probability of reaching one of the final states, which is referred to the accepting probability. Two automata are equivalent if and only if they accept each word with the same probability. The corresponding decision algorithm has been extensively studied, see [29, 31, 25, 26].

Markov decision processes (MDPs) were known as early as the 1950s [3], and are a popular modeling formalism used for instance in operations research, automated planning, and decision support systems. In MDPs, each state has a set of enabled actions and each enabled action leads to a distribution over successor states. MDPs have been widely used in the formal verification of randomized concurrent systems, and are now supported by probabilistic model checking tools such as PRISM [27], MRMC [24] and IscasMC [20].

* Supported by the National Natural Science Foundation of China (NSFC) under grant No. 61361136002, and Australian Research Council (ARC) under grant Nos. DP130102764 and FT100100218. Y. F. is also supported by the Overseas Team Program of Academy of Mathematics and Systems Science, Chinese Academy of Sciences.

DOI 10.1007/s00165-012-0242-7
BCS © 2012
Formal Aspects of Computing (2012) 24: 749–768

Formal Aspects of Computing

Exploring probabilistic bisimulations, part I

Matthew Hennessy
Department of Computer Science, Trinity College, Dublin 1, Ireland

Abstract. We take a fresh look at strong probabilistic bisimulations for processes which exhibit both non-deterministic and probabilistic behaviour. We suggest that it is natural to interpret such processes as distributions over states in a probabilistic labelled transition system, a pLTS; this enables us to adapt the standard notion of contextual equivalence to this setting. We then prove that a novel form of bisimulation equivalence between distributions are both sound and complete with respect to this contextual equivalence. We also show that a very simple extension to HML, Hennessy–Milner Logic, provides finite explanations for inequivalences between distributions. Finally we show that our bisimulations between distributions in a pLTS are simply an alternative characterisation of a standard notion of probabilistic bisimulation equivalence, defined between states in a pLTS.

Keywords: Probabilistic processes, Contextual equivalence, Bisimulation equivalence, Logical characterisation, Equational theory

1. Introduction

Bisimulations [Mil89] provide a well-established and elegant theory of the behaviour of non-deterministic processes. Let us review the framework.

(1) *Labelled transitions systems, LTSs*: These provide an intensional semantics for processes, describing their computations or more generally the interactions between processes and their environment.
(2) *Process calculi*: Formal description languages for describing processes and their specifications. These usually consist of a small number of *combinators* with which processes can be described by their structure.
(3) *Behavioural equivalence*: This determines which process descriptions are extensionally equivalent; that is which processes can not be distinguished by their users or more generally their environments. Perhaps the most uncontroversial is the so-called *contextual equivalence*, $\sim_{cxt}$ [MS92, SW01, HY95, Hen07], defined in terms of simple properties one would expect of a behavioural equivalence.
(4) *Bisimulations*: These are relations between processes which satisfy simple properties expressed in terms of the intensional semantics. They provide an elegant proof methodology for demonstrating process equivalence; to show two processes behaviourally equivalent it is sufficient to exhibit a witness bisimulation containing them. In many settings this proof method is not only sound with respect to contextual equivalence but also complete.

Transformers of belief states $\mathcal{DS} \to (\mathcal{PDS})^L$

When Equivalence and Bisimulation Join Forces in Probabilistic Automata*

Yuan Feng[1,2] and Lijun Zhang[3]

[1] University of Technology Sydney, Australia
[2] Department of Computer Science and Technology, Tsinghua University, China
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

Abstract. Probabilistic automata were introduced by Rabin in 1963 as language acceptors. Two automata are equivalent if and only if they accept each word with the same probability. On the other side, in the process algebra community, probabilistic automata were re-proposed by Segala in 1995 which are more general than Rabin's automata. Bisimulations have been proposed for Segala's automata to characterize the equivalence between them. So far the two notions of equivalences and their characteristics have been studied most independently. In this paper, we consider Segala's automata, and propose a novel notion of distribution-based bisimulation by joining the existing equivalence and bisimilarities. Our bisimulation bridges the two closely related concepts in the community, and provides a uniform way of studying their characteristics. We demonstrate the utility of our definition by studying distribution-based bisimulation metrics, which gives rise to a robust notion of equivalence for Rabin's automata.

1 Introduction

In 1963, Rabin [29] introduced the model *probabilistic automata* as language acceptors. In a probabilistic automaton, each input symbol determines a stochastic transition matrix over the state space. Starting with the initial distribution, each word (a sequence of symbols) has a corresponding probability of reaching one of the final states, which is referred to the accepting probability. Two automata are equivalent if and only if they accept each word with the same probability. The corresponding decision algorithm has been extensively studied, see [29, 31, 25, 26].

Markov decision processes (MDPs) were known as early as the 1950s [3], and are a popular modeling formalism used for instance in operations research, automated planning, and decision support systems. In MDPs, each state has a set of enabled actions and each enabled action leads to a distribution over successor states. MDPs have been widely used in the formal verification of randomized concurrent systems, and are now supported by probabilistic model checking tools such as PRISM [27], MRMC [24] and IscasMC [20].

* Supported by the National Natural Science Foundation of China (NSFC) under grant No. 61361136002, and Australian Research Council (ARC) under grant Nos. DP130102764 and FT100100218. Y. F. is also supported by the Overseas Team Program of Academy of Mathematics and Systems Science, Chinese Academy of Sciences.

Approximate Verification of the Symbolic Dynamics of Markov Chains

Manindra Agrawal, School of Electrical and Computer Engineering, Indian Institute of Technology, Kanpur, India 208016. Email: manindra@iitk.ac.in

S. Akshay, IRISA, ENS Cachan Bretagne, Campus de Beaulieu, Rennes, France 35042. Email: akshay@irisa.fr

Blaise Genest, CNRS, UMI IPAL joint with NUS and A*STAR/I2R, Singapore 138632. Email: bgenest@irisa.fr

P. S. Thiagarajan, School of Computing, National University of Singapore, Singapore 117417. Email: thiagu@comp.nus.edu.sg

***Abstract*—A finite state Markov chain M is often viewed as a probabilistic transition system. An alternative view - which we follow here - is to regard M as a linear transform operating on the space of probability distributions over its set of nodes. The novel idea here is to discretize the probability value space [0,1] into a finite set of intervals. A concrete probability distribution over the nodes is then symbolically represented as a tuple D of such intervals. The i-th component of the discretized distribution D will be the interval in which the probability of node i falls.**

The set of discretized distributions is a finite set and each trajectory, generated by repeated applications of M to an initial distribution, will induce a unique infinite string over this finite set of letters. Hence, given a set of initial distributions, the symbolic dynamics of M will consist of an infinite language L over the finite alphabet of discretized distributions. We investigate whether L meets a specification given as a linear time temporal logic formula whose atomic propositions will assert that the current probability of a node falls in an interval.

Unfortunately, even for restricted Markov chains (for instance, irreducible and aperiodic chains), we do not know at present if and when L is an (omega)-regular language. To get around this we develop the notion of an epsilon-approximation, based on the transient and long term behaviors of M. Our main results are that, one can effectively check whether (i) for each infinite word in L, at least one of its epsilon-approximations satisfies the specification; (ii) for each infinite word in L all its epsilon-approximations satisfy the specification. These verification results are strong in that they apply to all finite state Markov chains. Further, the study of the symbolic dynamics of Markov chains initiated here is of independent interest and can lead to other applications.

***Index Terms*—Model Checking, Probabilistic Computation, Approximation, Markov Processes.**

I. Introduction

Finite state Markov chains are a fundamental model of probabilistic dynamical systems. They are well-understood [13], [20] and their formal verification is well established [3]–[5], [8]–[10], [12], [14], [16], [17], [23]. In a majority of the verification related studies, the Markov chain is viewed a probabilistic transition system. The goal is to reason about the paths of the transition system using probabilistic temporal logics such as $PCTL$ [5], [10], [12].

An alternative view - which we follow here - is to view the state space of the chain to be the set of probability distributions over the nodes of the chain. The Markov chain transforms - in a linear fashion - a given probability distribution into a new one. Starting from a distribution μ one iteratively applies M to generate a trajectory consisting of a sequence of distributions. Given a set of initial distributions, one can study the properties of the set of trajectories generated by these distributions. The novel idea we explore in this setting is the *symbolic dynamics* of a Markov chain. We do so by discretizing the probability value space $[0, 1]$ into a *finite* set of intervals $\mathcal{I} = \{[0, p_1), [p_1, p_2), \ldots, [p_m, 1]\}$. A probability distribution μ of M over its set of nodes $\{1, 2, \ldots, n\}$ is then represented symbolically as a tuple of intervals $(d_1, d_2, \ldots, d_n)$ with d_i being the interval in which $\mu(i)$ falls. Such a tuple of intervals which symbolically represents at least one probability distribution is called a *discretized distribution*. In general a discretized distribution will represent an infinite set of concrete distributions.

A simple but crucial fact is that the set of discretized distributions, denoted $\mathcal{D}$, is a *finite* set. Consequently, each trajectory generated by an initial probability distribution will uniquely induce a sequence over the finite alphabet $\mathcal{D}$. Hence, given a (possibly infinite) set of initial distributions, the symbolic dynamics of M can be studied in terms of a language over the alphabet $\mathcal{D}$. Our focus here will be on infinite behaviors. Consequently, the main object of our study will be L_M, the ω-language over $\mathcal{D}$ induced by the set infinite trajectories generated by the set of initial distributions.

The main motivation for studying Markov chains in this fashion is to avoid the difficulties of numerically tracking sequences of probability distributions exactly. In many applications such as the probabilistic behavior of biochemical networks, queuing systems or sensor networks, exact estimates of the probability distributions (including the initial ones) may neither be feasible nor necessary. Further, not all the nodes may be relevant for the question at hand. In this case we can filter out such nodes by associating the "don't care" discretization $\{[0, 1]\}$ with each of them. This is a novel approach to dimension reduction and it can significantly reduce the practical complexity of analyzing high dimensional Markov chains. In our future work, we plan to apply this idea specifically to study the dynamics of biochemical networks.

To reason about the symbolic dynamics, we formulate a

Transformers of belief states $\mathcal{DS} \to (\mathcal{PDS})^L$

International Journal of Foundations of Computer Science

Equivalence of Labeled Markov Chains*

Laurent Doyen
I&C, École Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland

Thomas A. Henzinger
I&C, École Polytechnique Fédérale de Lausanne (EPFL)
EECS, University of California at Berkeley

Jean-François Raskin
DI, Université Libre de Bruxelles (ULB)
Boulevard du Triomphe CP 212, 1050 Bruxelles, Belgium

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

We consider the equivalence problem for labeled Markov chains (LMCs), where each state is labeled with an observation. Two LMCs are equivalent if every finite sequence of observations has the same probability of occurrence in the two LMCs. We show that equivalence can be decided in polynomial time, using a reduction to the equivalence problem for probabilistic automata, which is known to be solvable in polynomial time. We provide an alternative algorithm to solve the equivalence problem, which is based on a new definition of bisimulation for probabilistic automata. We also extend the technique to decide the equivalence of weighted probabilistic automata.
Then, we consider the equivalence problem for labeled Markov decision processes (LMDPs), which asks given two LMDPs whether for every scheduler (*i.e.* way of resolving the nondeterministic decisions) for each of the processes, there exists a scheduler for the other process such that the resulting LMCs are equivalent. The decidability of this problem remains open. We show that the schedulers can be restricted to be observation-based, but may require infinite memory.

Keywords: Labeled Markov chain, Markov decision process, probabilistic automaton, equivalence, bisimulation.

*This research was supported by the Belgian FNRS grant 2.4530.02 of the FRFC project "Centre Fédéré en Vérification", by the project "MoVES", an Interuniversity Attraction Poles Programme of the Belgian Federal Government, by the Swiss National Science Foundation, and by the European Network of Excellence on Embedded Systems Design (ARTIST 2).

Approximate Verification of the Symbolic Dynamics of Markov Chains

Manindra Agrawal
School of Electrical and Computer Engineering
Indian Institute of Technology
Kanpur, India 208016.
Email: manindra@iitk.ac.in

S. Akshay
IRISA, ENS Cachan Bretagne
Campus de Beaulieu
Rennes, France 35042.
Email: akshay@irisa.fr

Blaise Genest
CNRS, UMI IPAL
joint with NUS
and A*STAR/I2R
Singapore 138632.
Email: bgenest@irisa.fr

P. S. Thiagarajan
School of Computing
National University of Singapore
Singapore 117417.
Email: thiagu@comp.nus.edu.sg

***Abstract*—A finite state Markov chain M is often viewed as a probabilistic transition system. An alternative view - which we follow here - is to regard M as a linear transform operating on the space of probability distributions over its set of nodes. The novel idea here is to discretize the probability value space [0,1] into a finite set of intervals. A concrete probability distribution over the nodes is then symbolically represented as a tuple D of such intervals. The i-th component of the discretized distribution D will be the interval in which the probability of node i falls.**

The set of discretized distributions is a finite set and each trajectory, generated by repeated applications of M to an initial distribution, will induce a unique infinite string over this finite set of letters. Hence, given a set of initial distributions, the symbolic dynamics of M will consist of an infinite language L over the finite alphabet of discretized distributions. We investigate whether L meets a specification given as a linear time temporal logic formula whose atomic propositions will assert that the current probability of a node falls in an interval.

Unfortunately, even for restricted Markov chains (for instance, irreducible and aperiodic chains), we do not know at present if and when L is an (omega)-regular language. To get around this we develop the notion of an epsilon-approximation, based on the transient and long term behaviors of M. Our main results are that, one can effectively check whether (i) for each infinite word in L, at least one of its epsilon-approximations satisfies the specification; (ii) for each infinite word in L all its epsilon-approximations satisfy the specification. These verification results are strong in that they apply to all finite state Markov chains. Further, the study of the symbolic dynamics of Markov chains initiated here is of independent interest and can lead to other applications.

***Index Terms*—Model Checking, Probabilistic Computation, Approximation, Markov Processes.**

I. Introduction

Finite state Markov chains are a fundamental model of probabilistic dynamical systems. They are well-understood [13], [20] and their formal verification is well established [3]–[5], [8]–[10], [12], [14], [16], [17], [23]. In a majority of the verification related studies, the Markov chain is viewed a probabilistic transition system. The goal is to reason about the paths of the transition system using probabilistic temporal logics such as $PCTL$ [5], [10], [12].

An alternative view - which we follow here - is to view the state space of the chain to be the set of probability distributions over the nodes of the chain. The Markov chain transforms - in a linear fashion - a given probability distribution into a new one. Starting from a distribution μ one iteratively applies M to generate a trajectory consisting of a sequence of distributions. Given a set of initial distributions, one can study the properties of the set of trajectories generated by these distributions. The novel idea we explore in this setting is the *symbolic dynamics* of a Markov chain. We do so by discretizing the probability value space $[0,1]$ into a *finite* set of intervals $\mathcal{I} = \{[0,p_1),[p_1,p_2),\ldots,[p_m,1]\}$. A probability distribution μ of M over its set of nodes $\{1,2,\ldots,n\}$ is then represented symbolically as a tuple of intervals $(d_1,d_2,\ldots,d_n)$ with d_i being the interval in which $\mu(i)$ falls. Such a tuple of intervals which symbolically represents at least one probability distribution is called a *discretized distribution*. In general a discretized distribution will represent an infinite set of concrete distributions.

A simple but crucial fact is that the set of discretized distributions, denoted $\mathcal{D}$, is a *finite* set. Consequently, each trajectory generated by an initial probability distribution will uniquely induce a sequence over the finite alphabet $\mathcal{D}$. Hence, given a (possibly infinite) set of initial distributions, the symbolic dynamics of M can be studied in terms of a language over the alphabet $\mathcal{D}$. Our focus here will be on infinite behaviors. Consequently, the main object of our study will be L_M, the ω-language over $\mathcal{D}$ induced by the set infinite trajectories generated by the set of initial distributions.

The main motivation for studying Markov chains in this fashion is to avoid the difficulties of numerically tracking sequences of probability distributions exactly. In many applications such as the probabilistic behavior of biochemical networks, queuing systems or sensor networks, exact estimates of the probability distributions (including the initial ones) may neither be feasible nor necessary. Further, not all the nodes may be relevant for the question at hand. In this case we can filter out such nodes by associating the "don't care" discretization $\{[0,1]\}$ with each of them. This is a novel approach to dimension reduction and it can significantly reduce the practical complexity of analyzing high dimensional Markov chains. In our future work, we plan to apply this idea specifically to study the dynamics of biochemical networks.

To reason about the symbolic dynamics, we formulate a

Transformers of belief states $\mathcal{D}S \to (\mathcal{P}\mathcal{D}S)^L$

where do they come from?

Probabilistic Bisimulation:
Naturally on Distributions

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

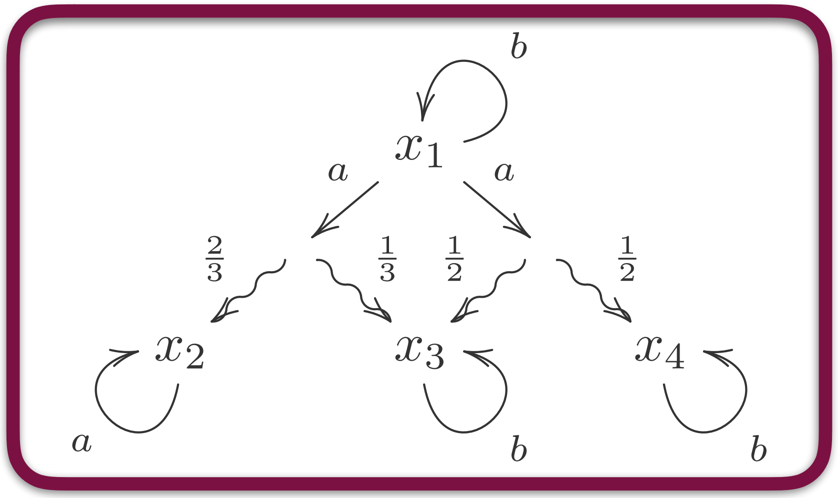
[1] Saarland University – Computer Science, Saarbrücken, Germany
{hermanns,krcal}@cs.uni-saarland.de
[2] IST Austria jan.kretinsky@ist.ac.at

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 long-standing open problem concerning the representation of memoryless continuous time by memory-full 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 memo-

Probabilistic automata II



semantics: combined probabilistic bisimulation [Segala&Lynch]

- **Set** -coalgebras $\quad S \to (\mathcal{C}S)^L$

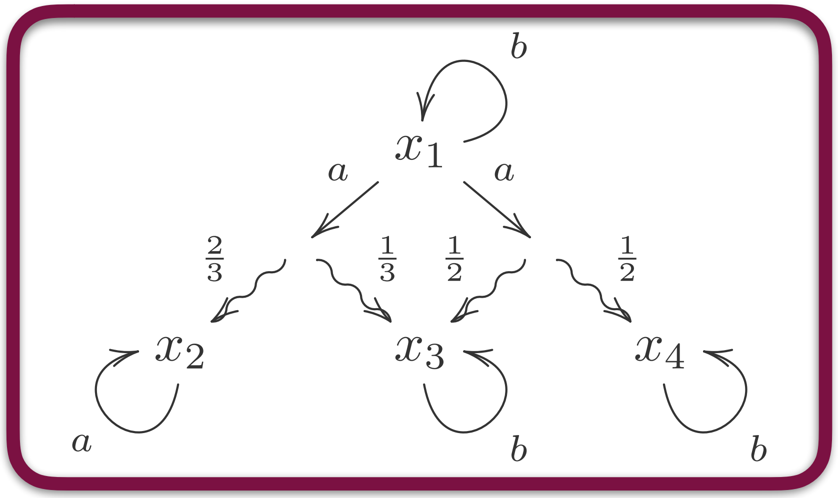
convex-subsets-of-distributions monad

$$\mathcal{C} = \mathcal{P}_c\mathcal{D} = \mathcal{CM}$$

[Varacca&Winskel'06] [Jacobs'08] [Mio'13-'14]

$$x_1 \xrightarrow{a} \tfrac{1}{3}x_2 + \tfrac{5}{12}x_3 + \tfrac{1}{4}x_4$$

PA I —> PA II



preserves semantics: probabilistic-bisimilar states are combined-probabilistic-bisimilar

- **Translation** $\quad \boxed{S \to (\mathcal{PD}S)^L} \xrightarrow{T} \boxed{S \to (\mathcal{C}S)^L}$

- via a natural transformation… $T(c)(a) = \overline{c(a)}$

convex closure

also the other way around…

(co)algebraic

the true nature

For PA III we will now move to $\mathcal{EM}(\mathcal{D})$

the Eilenberg-Moore algebras for the distribution monad

Eilenberg-Moore algebras

$\mathcal{EM}(\mathcal{D})$

- Objects

$$\begin{array}{c} \mathcal{D}A \\ \downarrow a \\ A \end{array}$$

satisfying

$$\begin{tikzcd} A \arrow[r, "\eta"] \arrow[dr, equal, "a"'] & \mathcal{D}A \arrow[d, "a"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} \mathcal{D}\mathcal{D}A \arrow[r, "\mu"] \arrow[d, "\mathcal{D}a"'] & \mathcal{D}A \arrow[d, "a"] \\ \mathcal{D}A \arrow[r, "a"] & A \end{tikzcd}$$

- Morphisms - algebra homomorphisms

$$\begin{tikzcd} \begin{array}{c} \mathcal{D}A \\ \downarrow a \\ A \end{array} \arrow[r, "h"] & \begin{array}{c} \mathcal{D}B \\ \downarrow b \\ B \end{array} \end{tikzcd}$$

$$h \circ a = b \circ \mathcal{D}h$$

Eilenberg-Moore algebras

infinitely many finitary operations

convex combinations

of $\mathcal{D}$ concretely

- convex algebras

$$\boxed{(A, \sum_{i=1}^{n} p_i(-)_i)} \qquad p_i \in [0,1], \sum_{i=1}^{n} p_i = 1$$

- affine (convex) maps

$$\boxed{h\left(\sum_{i=1}^{n} p_i a_i\right) = \sum_{i=1}^{n} p_i h(a_i)}$$

satisfying

Projection $\quad \sum_{i=1}^{n} p_i a_i = a_k, \quad p_k = 1$

Barycentre $\quad \sum_{i=1}^{n} p_i \left(\sum_{j=1}^{m} p_{ij} a_j\right) = \sum_{j=1}^{m} \left(\sum_{i=1}^{n} p_i p_{ij}\right) a_j$

The convex powerset

$\mathcal{P}_c$ on $\mathcal{EM}(\mathcal{D})$

The convex powerset

nonempty!

$\mathcal{P}_c$ on $\mathcal{EM}(\mathcal{D})$

- On objects

$$\mathcal{P}_c \boxed{\begin{array}{c}\mathcal{D}A \\ \downarrow a \\ A\end{array}} = \boxed{\begin{array}{c}\mathcal{D}A_c \\ \downarrow a_c \\ A_c\end{array}}$$

pointwise !

nonempty convex subsets of A

- On morphisms

$$\boxed{\mathcal{P}_c \text{``}=\text{''} \mathcal{P}}$$

Theorem $\mathcal{P}_c$ is a functor on $\mathcal{EM}(\mathcal{D})$

The constant exponent

$(-)^L$ on $\mathcal{EM}(\mathcal{D})$

The constant exponent

$(-)^L$ on $\mathcal{EM}(\mathcal{D})$

- On objects

$$\boxed{\boxed{\begin{array}{c}\mathcal{D}A\\ \downarrow a\\ A\end{array}}^{L}} = \boxed{\begin{array}{c}\mathcal{D}A^L\\ \downarrow a^L\\ A^L\end{array}}$$

pointwise

constant exponent on **Set**

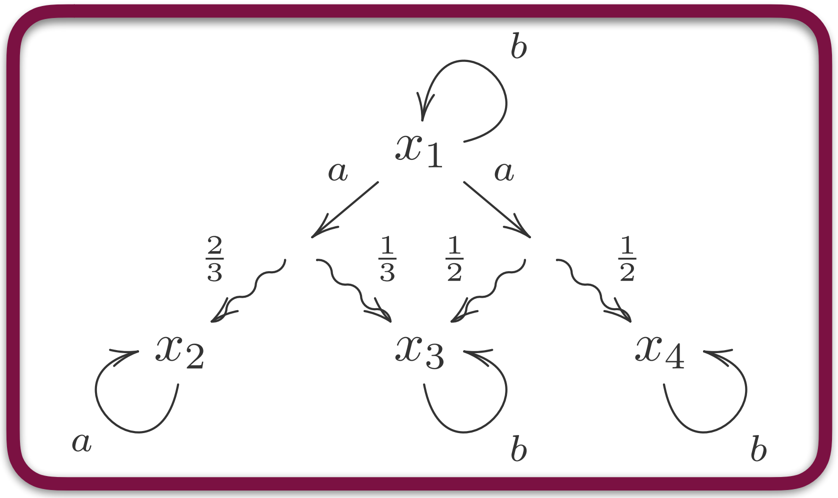
- On morphisms

$$\boxed{(-)^L \text{``}=\text{''} (-)^L}$$

Proposition $(-)^L$ is a functor on $\mathcal{EM}(\mathcal{D})$

Probabilistic automata III

input-enabled



semantics:
bisimilarity on distributions

- $\mathcal{EM}(\mathcal{D})$ -coalgebras with carriers free algebras

$$\begin{array}{c}\mathcal{DDS}\\ \downarrow \mu \\ \mathcal{DS}\end{array} \longrightarrow \left(\mathcal{P}_c \begin{array}{c}\mathcal{DDS}\\ \downarrow \mu \\ \mathcal{DS}\end{array}\right)^L$$

Recall PA II

$$S \to (\mathcal{C}S)^L$$

connection?

Quasi-liftings

$\mathcal{D} = U \circ F$

$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "F"] & \mathcal{EM}(\mathcal{D}) \arrow[l, bend left, "U"] \end{tikzcd} \quad \text{with} \quad F \dashv U$$

$\mathcal{C} = U \circ \mathcal{P}_c \circ F$

$U \circ (-)^L = (-)^L \circ U$

quasi-lifting

lifting

Recall PA II

$S \to (\mathcal{C}S)^L$

connection?

$$\begin{array}{c} \mathcal{D}\mathcal{D}S \\ \downarrow \mu \\ \mathcal{D}S \end{array} \longrightarrow \left(\mathcal{P}_c \left(\begin{array}{c} \mathcal{D}\mathcal{D}S \\ \downarrow \mu \\ \mathcal{D}S \end{array} \right) \right)^L$$

PA II —> PA III

input-enabled

embedding = faithful & injective on objects
preserves semantics

- **Translation**

$$S \to (\mathcal{C}S)^L \ \text{in } \mathbf{Set} \xrightarrow{T} \left(\begin{array}{c} \mathcal{D}\mathcal{D}S \\ \downarrow \mu \\ \mathcal{D}S \end{array} \to \mathcal{P}_c \begin{array}{c} \mathcal{D}\mathcal{D}S \\ \downarrow \mu \\ \mathcal{D}S \end{array} \right)^L \ \text{in } \mathcal{EM}(\mathcal{D})$$

$$\left(\mathcal{P}_c \begin{array}{c} \mathcal{D}\mathcal{D}S \\ \downarrow \mu \\ \mathcal{D}S \end{array} \right)^L = \begin{array}{c} \mathcal{D}((\mathcal{C}S)^L) \\ \downarrow \alpha \\ (\mathcal{C}S)^L \end{array}$$

$$T(c) = c^{\#} = \alpha \circ \mathcal{D}c$$

$$T \text{ “}=\text{” } \mathcal{D}$$

on morphisms

The true nature of PA

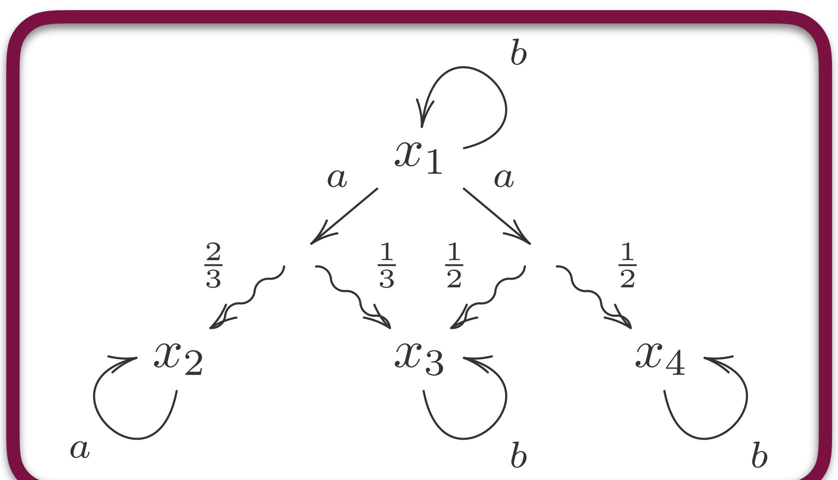
Thank You!

input-enabled

$S \to (\mathcal{P}\mathcal{D}S)^L$ **Set** $\xrightarrow{T}$ $S \to (\mathcal{C}S)^L$ **Set** $\xrightarrow{T}$

$$\left(\begin{array}{c}\mathcal{D}\mathcal{D}S\\ \downarrow \mu\\ \mathcal{D}S\end{array} \to \mathcal{P}_c\left(\begin{array}{c}\mathcal{D}\mathcal{D}S\\ \downarrow \mu\\ \mathcal{D}S\end{array}\right)\right)^L$$

$\mathcal{EM}(\mathcal{D})$



$$\begin{array}{c}\frac{1}{3}x_1 + \frac{2}{3}x_2\\ \downarrow a\\ \frac{2}{3}x_2 + \frac{1}{6}x_3 + \frac{1}{6}x_4\end{array}$$

$$\begin{array}{c}\frac{1}{3}x_1 + \frac{2}{3}x_2\\ \downarrow a\\ \frac{8}{9}x_2 + \frac{1}{9}x_3\end{array}$$