Probabilistic Systems Coalgebraically: A Survey

Ana Sokolova[1,*]

Department of Computer Sciences
University of Salzburg
Austria

Abstract

We survey the work on both discrete and continuous space probabilistic systems as coalgebras, starting with how probabilistic systems are modeled as coalgebras and followed by a discussion of their bisimilarity and behavioral equivalence, mentioning results that follow from the coalgebraic treatment of probabilistic systems. It is interesting to note that, for different reasons, for both discrete and continuous probabilistic systems it may be more convenient to work with behavioral equivalence than with bisimilarity.

Keywords: Probabilistic systems, Coalgebra, Markov chains, Markov processes

1. Introduction

Probabilistic systems are models of systems that involve quantitative information about uncertainty. They have been extensively studied in the last two decades in the area of probabilistic verification and concurrency theory. The models originate in the rich theory of Markov chains and Markov processes (see e.g. [49]) and in the early work on probabilistic automata [63, 61].

Discrete probabilistic systems, see e.g. [49, 77, 30, 55, 62, 67, 33, 22] and [70] for an overview, are transition systems on discrete state spaces and come in different flavors: fully probabilistic (Markov chains), labeled (with reactive or generative labels), or combining non-determinism and probability.

*Corresponding author
Email address: `anas@cs.uni-salzburg.at` (Ana Sokolova)
URL: `http://cs.uni-salzburg/~anas` (Ana Sokolova)

[1]Work supported by the Austrian Science Funds (FWF) Project V00125.

Probabilities in discrete probabilistic systems appear as labels on transitions between states. For example, in a Markov chain a transition from one state to another is taken with a given probability.

Continuous probabilistic systems, see e.g. [7, 23, 26, 11, 21, 45] as well as the recent books [59, 27, 28] that contain most of the research on continuous probabilistic systems, are transition systems modeling probabilistic behavior on continuous state spaces. The basic model is that of a Markov process. Central to continuous probabilistic systems is the notion of a probability measure on a measurable space. Therefore, the state space of a continuous probabilistic system is equipped with a σ-algebra and forms a measurable space. It is no longer the case that the probability of moving from one state to another determines the behavior of the system. Actually, the probability of reaching any single state from a given state may be zero while the probability of reaching a subset of states is nonzero. A Markov process is specified by the probability of moving from any source state to any measurable subset in the σ-algebra, which is intuitively interpreted as the probability of moving from the source state to some state in the subset.

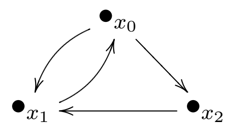
Both discrete and continuous probabilistic systems can be modeled as coalgebras and coalgebra theory has proved a useful and fruitful means to deal with probabilistic systems. In this paper, we give an overview of how to model probabilistic systems as coalgebras and survey coalgebraic results on discrete and continuous probabilistic systems. Having modeled probabilistic systems as coalgebras, there are two types of results where coalgebra meets probabilistic systems: (1) particular problems for probabilistic systems have been solved using coalgebraic techniques, and (2) probabilistic systems appear as popular examples on which generic coalgebraic results are instantiated. The results of the second kind are not to be considered of less importance: sometimes they lead to completely new results not known in the community of probabilistic systems, e.g. [2, 5, 17, 52, 65, 60]. Moreover, the variety of probabilistic systems provides a nice set of (motivating) examples for generic coalgebra results or observations, e.g. [20, 60]. Also, looking at probabilistic systems from different perspectives provides evidence in favor of behavioral equivalence rather than bisimilarity.

In the paper, we take the following route. We start with an introduction to basic coalgebra notions, and some particular results concerning bisimilarity and behavioral equivalence that are needed for what follows (Section 2). Then we discuss discrete probabilistic systems (Section 3), and the inductive class of functors that turns each of them into a coalgebra on **Sets**, the category of sets and functions. We proceed with an expressiveness comparison of discrete probabilistic systems which benefits from the coalgebraic modeling, and discuss existing results of both types mentioned above on discrete probabilistic systems. We note that from this general (coalgebraic) perspective probabilistic transitions can be viewed as transitions labeled by elements of some commutative monoid, and therefore are comparable to nondeterministic transitions. We also show an alternative way of modeling (reactive) discrete probabilistic systems as coalgebras on the category of pseudometric spaces and nonexpansive functions taken by van Breugel and Worrell that allows for a definition of behavioral distances between states [15, 16]. Next we move to continuous probabilistic systems (Section 4) where we show how they are modeled as coalgebras on **Meas**, the category of measurable spaces and measurable maps. Actually, in the literature, most of the time continuous probabilistic systems live on some special categories of measurable spaces (analytic, Polish, metric/pseudometric/ultrametric spaces). We discuss the reasons for this and present some observations (related to the dilemma of bisimilarity versus behavioral equivalence) that allow us to stay in **Meas**. We end our trip with a short discussion on how discrete systems are embedded into continuous systems, i.e., as expected, Markov chains are Markov processes. Please fasten your seat belt and enjoy the flight over the landscape of probabilistic systems and coalgebras.

2. Coalgebras, Bisimilarity, and Behavioral Equivalence

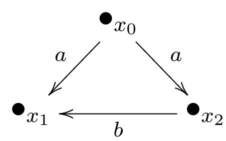
Let $\mathbb{C}$ be a category and F an endofunctor on $\mathbb{C}$. An F-coalgebra is a pair $\langle X, c\colon X \to FX\rangle$ where X in $\mathbb{C}$ is the carrier, and c is the coalgebra structure. For brevity, we often identify a coalgebra with its coalgebra structure. Given two F-coalgebras $c\colon X \to FX$ and $d\colon Y \to FY$, a coalgebra homomorphism from c to d is a map $h\colon X \to Y$ such that $d \circ h = Fh \circ c$. F-coalgebras together with their coalgebra homomorphisms form a category, denoted by Coalg_F.

In this paper we only consider coalgebras on concrete categories, i.e., categories with a faithful forgetful functor to **Sets**, the category of sets and functions. Then the carrier X of a coalgebra provides the set of states (after the application of the forgetful functor) and the coalgebra map c gives the transitions to the next state(s). The functor F determines the type of transitions. For example, coalgebras of the powerset functor $\mathcal{P}$ on **Sets** are nondeterministic transition systems in which from any state there is a set of (non-labeled) transitions to possible next states. An example of a non-deterministic transition system as coalgebra is presented below.



$$\begin{aligned}&X = \{x_0, x_1, x_2\}, \quad X \xrightarrow{c} \mathcal{P}X\\ &c(x_0) = \{x_1, x_2\}\\ &c(x_1) = \{x_0\}\\ &c(x_2) = \{x_1\}\end{aligned}$$

Coalgebras of the functor $\mathcal{P}(A \times \_)$ on **Sets**, for a set of labels A, are labeled transition systems (LTS). An example is:



$$\begin{aligned}&X = \{x_0, x_1, x_2\}, \quad X \xrightarrow{c} \mathcal{P}(A \times X)\\ &c(x_0) = \{\langle a, x_1\rangle, \langle a, x_2\rangle\}\\ &c(x_1) = \emptyset\\ &c(x_2) = \{\langle b, x_1\rangle\}\end{aligned}$$

Homomorphisms of (labeled) transition systems are transition preserving and reflecting maps. We refer the reader to [64, 44] for a gentle introduction to coalgebra and many interesting examples.

A final F-coalgebra is a final object in the category Coalg_F: from any F-coalgebra c there is a unique homomorphism beh_c to the final one. If a final coalgebra exists, it induces a final coalgebra semantics which identifies two states if and only if they are mapped to the same element of the final coalgebra via the unique homomorphism. For weak pullback preserving functors on **Sets** the final coalgebra semantics coincides with coalgebraic bisimilarity defined in the following way via the notion of a bisimulation.

Let $c\colon X \to FX$ and $d\colon Y \to FY$ be two coalgebras on **Sets**. A relation $R \subseteq X \times Y$ is a bisimulation between c and d if there exists a mediating coalgebra structure $r\colon R \to FR$ making the two projections coalgebra homomorphisms, i.e., making the following diagram commute

$$\begin{tikzcd}
X \arrow[d, "c"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\exists r"'] & Y \arrow[d, "d"] \\
FX & FR \arrow[l, "F\pi_1"] \arrow[r, "F\pi_2"'] & FY
\end{tikzcd} .$$

Two states x and y are bisimilar, notation $x \sim y$, if they are related by some bisimulation relation. Weak pullback preservation of the type functor F suffices for bisimilarity to be an equivalence. Therefore, it also suffices for bisimilarity on a given coalgebra to be the union of all equivalence bisimulation relations (bisimulations that are equivalences). In order to relate coalgebraic bisimilarity to concrete notions of bisimilarity in the literature, the notion of relation lifting is helpful.

Let $R \subseteq X \times Y$ be a relation, and F a functor on **Sets**. The relation R can be lifted to a relation $\mathrm{Rel}(F)(R) \subseteq FX \times FY$ defined by

$$\langle x, y\rangle \in \mathrm{Rel}(F)(R) \;\Leftrightarrow\; \exists z \in FR \colon F\pi_1(z) = x,\; F\pi_2(z) = y.$$

It is easy to see that relation liftings provide transfer conditions for bisimulation, as stated in the next property.

Lemma 2.1. *A relation $R \subseteq X \times Y$ is a bisimulation between the F-coalgebras $c\colon X \to FX$ and $d\colon Y \to FY$ if and only if*

$$\langle x, y\rangle \in R \implies \langle c(x), d(y)\rangle \in \mathrm{Rel}(F)(R).$$

Moreover, one can show [69] the following characterization of equivalence liftings for weak pullback preserving functors.

Lemma 2.2. *If F preserves weak pullbacks and R is an equivalence on X, then* $\mathrm{Rel}(F)(R)$ *is the pullback of the cospan* $FX \xrightarrow{Fe} F(X/R) \xleftarrow{Fe} FX$ *where $e : X \to X/R$ is the canonical map mapping each element to its equivalence class.*

As a consequence we get the following characterization of equivalence bisimulations in terms of transfer conditions.

Corollary 2.3. *A relation $R \subseteq X \times X$ is an equivalence bisimulation on the F-coalgebra $c\colon X \to FX$, where F preserves weak pullbacks, if and only if*

$$\langle x, y\rangle \in R \implies (Fe \circ c)(x) = (Fe \circ c)(y).$$

where e is the canonical map mapping each element to its equivalence class.

We will see later how Lemma 2.1, Lemma 2.2, Corollary 2.3, and some properties of relation liftings of inductively defined functors provide a modular way to show that coalgebraic and concrete bisimilarity coincide for all discrete probabilistic systems.

A way to define bisimulations on general categories is using a span of morphisms. A span $\langle R, r_1 \colon R \to X, r_2 \colon R \to Y\rangle$ is a bisimulation between two F-coalgebras $c\colon X \to FX$ and $d\colon Y \to FY$ on a category $\mathbb{C}$ if R, r_1, and r_2 satisfy some additional conditions that ensure non-triviality (e.g. r_1, r_2

being epi) and there exists a coalgebra map $r\colon R \to FR$ making both r_1 and r_2 coalgebra homomorphisms. One can then define that the coalgebras c and d are bisimilar if there is a bisimulation between them. For coalgebras on a concrete category, two states $x \in X$ and $y \in Y$ are bisimilar if there exists $z \in R$ such that $x = r_1(z)$ and $y = r_2(z)$, excluding the need of additional conditions on r_1 and r_2. On **Sets** the two definitions, general span versus relation with projections, are equivalent.

Another behavior semantics, closely related to bisimilarity, called behavioral equivalence always coincides with the final coalgebra semantics. It has proven very useful in reasoning about probabilistic systems. It is based on the notion of a cocongruence [54, 81] which is a cospan rather than a span.

A cocongruence between two F-coalgebras $c\colon X \to FX$ and $d\colon Y \to FY$ is a cospan $\langle U, u_1\colon X \to U, u_2\colon Y \to U\rangle$, with u_1 and u_2 jointly epi, such that there exists an F-coalgebra map $u\colon U \to FU$ making u_1 and u_2 coalgebra homomorphisms, i.e., making the following diagram commute

$$\begin{tikzcd} X \arrow[r, "u_1"] \arrow[d, "c"'] & U \arrow[d, dashed, "\exists u"'] & Y \arrow[l, "u_2"'] \arrow[d, "d"] \\ FX \arrow[r, "Fu_1"'] & FU & FY \arrow[l, "Fu_2"] \end{tikzcd} .$$

We say that the coalgebras c and d are behaviorally equivalent if they are connected by a cocongruence. For coalgebras on a concrete category, we say that $x \in X$ and $y \in Y$ are behaviorally equivalent, and write $x \approx y$, if they are identified by some cocongruence between them, i.e., if there exists a cocongruence $\langle U, u_1, u_2\rangle$ with $u_1(x) = u_2(y)$.

In particular in the study of probabilistic systems coalgebraically, but also in coalgebraic modal logics in general, behavioral equivalence has advantages over bisimilarity, as we will see below. However, the good side of bisimilarity is that it is computable by efficient algorithms[2]. Another reason for working with bisimilarity is traditional, many concrete types of systems come equipped with a concrete notion of bisimilarity. Bisimilarity always implies behavioral equivalence in categories with pushouts, cf. e.g. [5, 69]. If additionally the type functor F preserves weak pullbacks, then the two notions

[2]Bisimilarity can be computed by iterative algorithms, see e.g. [39], thus making it possible to automatize coinduction proofs. The plain definition of behavioral equivalence, without knowing that it coincides with bisimilarity, does not provide such methods.

coincide (cf. e.g. [5, 69]). This property is useful in comparing expressivity of different types of coalgebras.

3. Discrete Probabilistic Systems

Discrete probabilistic systems are state-based systems in which a change of state is governed by a discrete probability distribution over possible next states. In addition, one may have labels, non-determinism, and/or termination. The basic model involving discrete probabilities is a Markov chain, given by a set of states and from each state a probability distribution over the set of states, determining the probability of transiting to any other state. We start by recalling the definition of a discrete (sub)probability distribution.

Definition 3.1. *Let X be a set. A function $\mu\colon X \to \mathbb{R}^{\geq 0}$ is a discrete probability distribution, or distribution for short, on X if $\sum_{x\in X}\mu(x) = 1$. It is a discrete subprobability distribution if $\sum_{x\in X}\mu(x) \leq 1$. The set $\{x \in X \mid \mu(x) > 0\}$ is the support of μ and is denoted by* $\mathrm{supp}(\mu)$.

Note that the general X-indexed sum is defined as

$$\mu[X] = \sum_{x\in X}\mu(x) = \sup_{\substack{X'\subseteq X\\ X' \text{ finite}}}\left\{\sum_{x\in X'}\mu(x)\right\} \ \in\ \mathbb{R}\cup\{\infty\}$$

and it is well-defined since $\mu(x) \geq 0$. One can show that for any discrete probability distribution (see e.g. [69]) the support set is at most countable, which justifies the use of the term "discrete".

A distribution that assigns probability 1 to a single element $x \in X$ is called a Dirac distribution, denoted by δ_x. Hence, $\delta_x(y) = 1$ if $y = x$ and $\delta_x(y) = 0$ otherwise.

In the remainder of this section we will discuss how to model various discrete probabilistic systems as coalgebras, starting from Markov chains as the basic discrete probabilistic system type and their bisimilarity, to more-complex inductively defined models and the relationship of their coalgebraic and concrete notions of bisimilarity. Moreover, we will briefly discuss an expressiveness comparison of the different discrete probabilistic systems which is made possible using the generality of coalgebra, as well as other specific and general results involving discrete probabilistic systems as coalgebras. Finally, we point out that although the research on discrete probabilistic systems is very elaborate and advanced, probabilities may not be that special: the finitary functors used for modeling discrete probabilistic systems have a generalization that allows studying and modeling more general monoid-valued valuations and not only discrete probability distributions.

3.1. Coalgebraic Modeling

Almost all[3] types of known discrete probabilistic systems used as models for different verification and analysis techniques can be modeled as coalgebras on **Sets**. The main step towards coalgebraic modeling of discrete probabilistic systems is in the choice of a functor to represent discrete probability distributions over a set of states.

Definition 3.2. *The probability distribution functor*

$$\mathcal{D}\colon \mathbf{Sets} \to \mathbf{Sets}$$

maps a set X to

$$\mathcal{D}X = \{\mu\colon X \to \mathbb{R}^{\geq 0} \mid \mu[X] = 1\}$$

and a function $f\colon X \to Y$ to $\mathcal{D}f\colon \mathcal{D}X \to \mathcal{D}Y$ given by

$$(\mathcal{D}f)(\mu) = \lambda y.\mu[f^{-1}(\{y\})].$$

Variants of the probability distribution functor are also used, and may be convenient depending on the application. Such are the subprobability distribution (subdistribution) functor $\mathcal{D}_{\leq 1}$ and the finitely supported (sub)distribution functors $\mathcal{D}_f$ and $\mathcal{D}_{\leq 1,f}$, whose definitions vary from the definition of $\mathcal{D}$ only on objects. We have

$$\begin{array}{rcl} \mathcal{D}_{\leq 1}X & = & \{\mu\colon X \to \mathbb{R}^{\geq 0} \mid \mu[X] \leq 1\} \\ \mathcal{D}_{f}X & = & \{\mu\colon X \to \mathbb{R}^{\geq 0} \mid \mu[X] = 1, \text{supp}(\mu) \text{ is finite}\} \\ \mathcal{D}_{\leq 1,f}X & = & \{\mu\colon X \to \mathbb{R}^{\geq 0} \mid \mu[X] \leq 1, \text{supp}(\mu) \text{ is finite}\} \end{array}$$

All these functors are well-behaved in the following sense.

[3] All with exception of the strictly alternating systems [33], which can be modeled as multi-sorted coalgebras [60].

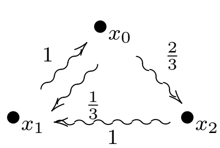
Proposition 3.3. *The (sub)probability distribution functor and its finitary variant preserve weak pullbacks. Moreover, each of them has a final coalgebra.*

The proof of the weak pullback preservation of $\mathcal{D}_f$ goes back to the first coalgebraic treatment of discrete probabilistic systems by de Vink and Rutten [80], exploiting the graph-theoretical max-flow min-cut theorem as in [46], and was also shown by Moss [57], using an elementary matrix fill-in property. Similar arguments apply to $\mathcal{D}_{\leq 1,f}$. The same result is an instance of a more general result on (finitely supported) monoid valuations (also called copower) functors, by Gumm and Schroeder [32, 31]. Finite support is not necessary for the weak pullback preservation of the probability distribution functor, also $\mathcal{D}$ preserves weak pullbacks [69] which is established by showing that the needed matrix fill-in property [57] can be used and holds for countably infinite matrices as well. Similarly $\mathcal{D}_{\leq 1}$ preserves weak pullbacks. The existence of a final coalgebra for $\mathcal{D}_f$ is trivial (the final coalgebra itself is trivial as well). In [80] it was proven that final coalgebra exists also for $\mathcal{D}_f + 1$ (allowing termination), using that the functor is bounded. Similar arguments apply to all variants. Also the finitely supported monoid valuation functor is bounded [68], generalizing the result on $\mathcal{D}_f$ and $\mathcal{D}_{\leq 1,f}$. We will discuss monoid valuation functors in Section 3.4 below.

Finally, we note that the (sub)distribution functors and their finitary versions are not only functors but also monads.

3.1.1. Markov chains

Discrete-time Markov chains (DTMCs) [49, 40], or Markov chains for short, form the basic type of discrete probabilistic systems. They are coalgebras of the discrete probability distribution functor $\mathcal{D}$. That is, a Markov chain with a state set X is a coalgebra $c\colon X \to \mathcal{D}(X)$. An example is shown below where $x \overset{p}{\rightsquigarrow} y$ for states $x, y \in X$, denotes that the probability of moving from x to y is p, i.e., $c(x)(y) = p$.



$$
\begin{aligned}
&X = \{x_0, x_1, x_2\}, \quad X \xrightarrow{c} \mathcal{D}X \\
&c(x_0) = \{x_0 \mapsto 0, x_1 \mapsto \tfrac{1}{3}, x_2 \mapsto \tfrac{2}{3}\} \\
&c(x_1) = \delta_{x_0} \\
&c(x_2) = \delta_{x_1}
\end{aligned}
$$

3.1.2. Probabilistic System Types

We can now model most of the discrete probabilistic systems from the literature, in a modular way, using an inductively defined class of functors on **Sets**. The functors are built using the following syntax

$$F ::= \_ \mid A \mid \_^A \mid \mathcal{P} \mid \mathcal{D} \mid F \circ F \mid F \times F \mid F + F$$

where the basic functors used are the identity functor $\_$; the constant functor A mapping each set to the constant set A and each map to the identity on A; the constant exponent functor $\_^A$ mapping a set X to the set of all functions from A to X and a map $f\colon X \to Y$ to the map $f^A\colon X^A \to Y^A$ given by $f^A(g) = f \circ g$; the powerset functor $\mathcal{P}$ mapping a set to the collection of its subsets and a function $f\colon X \to Y$ to $\mathcal{P}(f)\colon \mathcal{P}(X) \to \mathcal{P}(Y)$ with $\mathcal{P}(f)(X') = f(X') = \{f(x) \mid x \in X'\}$, the direct image; and the probability distribution functor $\mathcal{D}$.

An important property for some of the results presented later is weak pullback preservation. We note here that any functor in this inductively defined class preserves weak pullbacks [64, 69].

This class of functors suffices to model various discrete probabilistic systems used as mathematical models of real systems for formal verification. Most of the existing probabilistic systems arose independently in the literature to improve modeling of one or another property of a system. One motivating issue was the need to model both non-deterministic and probabilistic choice, another is compositional modeling. In Figure 3.1.2 (previously introduced in [5, 69]) we present functors that allow for coalgebraic modeling of known discrete probabilistic systems (and some standard transition systems for comparison), together with the abbreviations that we use to denote the corresponding category of coalgebras, and references to papers introducing them. For some of the systems, names used here follow [70, 5, 69] and deviate from the original names. Such are (simple) Segala systems, also known as (simple) probabilistic automata and closely related to Markov decision processes (MDPs) [6], Vardi systems which were originally introduced as concurrent Markov chains, and Pnueli-Zuck systems that were originally named probabilistic finite state programs.

The alternating systems considered here do not involve strict alternation as in the original definition of Hansson [33]. Strictly alternating systems can be modeled as multi-sorted coalgebras [60]. For more details on each of the discrete probabilistic systems, the reader is referred to [70, 5, 69]. Here we only briefly mention that both reactive and generative systems arise from LTS when replacing the non-deterministic choice modeled by $\mathcal{P}$ (in the isomorphic input view and output view on LTS) with probabilistic choice modeled by $\mathcal{D}$

Coalg_F	F	name for $X \to FX$/reference
MC	$\mathcal{D}$	Markov chains
DLTS	$(\_ + 1)^A$	deterministic automata
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$	non-deterministic automata, LTSs
React	$(\mathcal{D} + 1)^A$	reactive systems [55, 30]
Gen	$\mathcal{D}(A \times \_) + 1$	generative systems [30]
Str	$\mathcal{D} + (A \times \_) + 1$	stratified systems [30]
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$	alternating systems [33]
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$	Vardi systems [77]
SSeg	$\mathcal{P}(A \times \mathcal{D})$	simple Segala systems [67, 66]
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$	Segala systems [67, 66]
Bun	$\mathcal{D}\mathcal{P}(A \times \_)$	bundle systems [22]
PZ	$\mathcal{P}\mathcal{D}\mathcal{P}(A \times \_)$	Pnueli-Zuck systems [62]
MG	$\mathcal{P}\mathcal{D}\mathcal{P}(A \times \_ + \_)$	most general systems

Figure 1: Discrete probabilistic system types

and termination possibility. Some models make a distinction on the type of states, (non-)deterministic versus probabilistic, as in the case of alternating, stratified, or Vardi systems; some models involve both non-deterministic and probabilistic choices, like Segala systems or bundle systems. The last type of systems is added here in order to have a top element in the expressiveness hierarchy that we will discuss in Section 3.3.

3.2. Bisimulation Correspondence

All of the concrete discrete probabilistic systems come with a concrete notion of bisimilarity defined via bisimulation equivalence in terms of transfer conditions, based on the original definition of Larsen and Skou for reactive systems [55]. The main argument that justifies generic coalgebraic results for these systems is the coincidence of the concrete notions of bisimilarity with coalgebraic bisimilarity in every single case, which can be shown in a modular way using relation liftings and their properties. For the concrete bisimilarity definitions the reader is referred to [70, 69] and the complete proof of bisimilarity correspondence can be found in [69]. Here we illustrate the bisimilarity correspondence proof method for the case of Markov chains and simple Segala systems.

Definition 3.4. *An equivalence relation R on the set of states X of a Markov chain $c\colon X \to \mathcal{D}X$ is a (concrete) bisimulation if and only if $\langle x, y\rangle \in R$ implies*

$$\begin{array}{l}\textit{if } x \rightsquigarrow \mu, \textit{ then there is a distribution } \mu' \textit{ with } y \rightsquigarrow \mu' \\ \qquad\qquad \textit{and } \mu \equiv_R \mu'\end{array}$$

where $x \rightsquigarrow \mu$ denotes that $c(x) = \mu \in \mathcal{D}X$, and $\mu \equiv_R \mu'$ if and only if for any R-equivalence class C, $\mu[C] = \mu'[C]$.

The condition on the equivalence classes is closely related to the notion of distribution lifting [47], which is exactly the relation lifting for the probability distribution functor $\mathcal{D}$.

Definition 3.5. *Let $R \subseteq X \times Y$ be a relation. Let $\mu \in \mathcal{D}(X)$ and $\mu' \in \mathcal{D}(Y)$ be distributions. Define $\mu\,\overline{R}\,\mu'$ if and only if there exists a joint distribution $\nu \in \mathcal{D}(X \times Y)$ such that*

(i) ν has μ and μ' as marginals, i.e.,

1. $\nu[x, Y] = \mu(x)$ *for any* $x \in X$
2. $\nu[X, y] = \mu'(y)$ *for any* $y \in Y$

(ii) ν *satisfies* $\nu(x, y) \neq 0 \Longrightarrow \langle x, y\rangle \in R$.

It is easy to see [69] that $\mathrm{Rel}(\mathcal{D})(R) = \overline{R}$. In the special case when R is an equivalence relation on X, from Lemma 2.2, one gets that $\mathrm{Rel}(\mathcal{D})(R) = \equiv_R$, also shown directly in [48, 72, 1].

Hence, by Lemma 2.1, we get that an equivalence R on the set of states X of a Markov chain $c\colon X \to \mathcal{D}X$ is a bisimulation according to Definition 3.4 if and only if it is a coalgebraic equivalence bisimulation, showing that the concrete and the coalgebraic bisimilarity notions for Markov chains coincide. This fact was first shown by de Vink and Rutten [80]. The same technique was used by Bartels et al. [4] to sketch the correspondence of concrete bisimulation and coalgebraic bisimulation for general Segala-type systems. In [5] another, more modular proof is presented of the correspondence of concrete probabilistic bisimulation with the coalgebraic bisimulation in the case of simple Segala systems based on Corollary 2.3. At the same time, it was a proof of the correspondence for reactive systems. That technique can also be used in all the other cases. However, having Lemma 2.1 and the properties of relation liftings, it is a matter of simple, structured, and modular derivation to show the correspondence of coalgebraic and concrete bisimilarity for all of the probabilistic systems that come with a notion of bisimulation [69]. We briefly present the general method here and instantiate it to the example of simple Segala systems.

Note that bisimilarity for Markov chains is trivial, i.e., any two states in a Markov chain $c\colon X \to \mathcal{D}X$ are bisimilar since $X \times X$ is a bisimulation relation. This quickly changes in the presence of termination, action labels, and/or non-determinism.

The next lemma [41, 69] shows that for our class of inductively defined functors, $\mathrm{Rel}(F)$ can be defined by structural induction. Item (v) repeats what we already discussed above.

Lemma 3.6. *Let* $R \subseteq X \times Y$ *be a relation. Then:*

(i) $\mathrm{Rel}(\_)(R) = R$,

(ii) $\mathrm{Rel}(A)(R) = \Delta_A$, *the diagonal relation on* A,

$$\textit{(iii)}\ \ \mathrm{Rel}(\mathcal{P})(R) \ = \big\{ \langle X, Y\rangle \mid (\forall x \in X)(\exists y \in Y)\langle x,y\rangle \in R \,\wedge \\ (\forall y \in Y)(\exists x \in X)\langle x,y\rangle \in R \big\},$$

$$\textit{(iv)}\ \ \mathrm{Rel}(\_^{A})(R) = \{\langle f,g\rangle \mid (\forall a \in A)\langle f(a), g(a)\rangle \in R\},$$

(v) $\mathrm{Rel}(\mathcal{D})(R) = \overline{R}$ *(see Definition 3.5),*

$$\textit{(vi)}\ \ \mathrm{Rel}(F \circ G)(R) = \mathrm{Rel}(F)(\mathrm{Rel}(G)(R)),$$

$$\textit{(vii)}\ \ \mathrm{Rel}(F \times G)(R) \ = \big\{ \big\langle \langle x_1, x_2\rangle, \langle y_1, y_2\rangle \big\rangle \mid \langle x_1, y_1\rangle \in \mathrm{Rel}(F)(R) \,\wedge \\ \langle x_2, y_2\rangle \in \mathrm{Rel}(G)(R) \big\},$$

$$\textit{(viii)}\ \ \mathrm{Rel}(F + G)(R) \ = \big\{ \langle \kappa_1(x_1), \kappa_1(y_1)\rangle \mid \langle x_1, y_1\rangle \in \mathrm{Rel}(F)(R) \big\} \cup \\ \big\{ \langle \kappa_2(x_2), \kappa_2(y_2)\rangle \mid \langle x_2, y_2\rangle \in \mathrm{Rel}(G)(R) \big\}$$

where κ_1, κ_2 denote the injections into the coproduct, $\kappa_i \colon X_i \to X_1 + X_2$.

Now we can apply Lemma 3.6 to show the correspondence of concrete and coalgebraic bisimilarity for simple Segala systems. Similar modular arguments apply to all other discrete probabilistic systems, see [69] for the complete proof. The concrete definition of bisimulation and bisimilarity for simple Segala systems [67] is the same as the original definition of bisimulation for reactive systems introduced by Larson and Skou [55], which we recall next.

Definition 3.7. *An equivalence relation R on X is a (concrete) bisimulation on the simple Segala system $c\colon X \to \mathcal{P}(A \times \mathcal{D}X)$ if and only if $\langle x, y\rangle \in R$ implies*

$$\text{if } x \xrightarrow{a} \mu, \text{ then there exists a distribution } \mu' \text{ with } y \xrightarrow{a} \mu' \text{ and } \mu \equiv_R \mu', \\ \text{i.e., for any } R\text{-equivalence class } C,\ \mu[C] = \mu'[C]$$

where $x \xrightarrow{a} \mu$ denotes that $\langle a, \mu\rangle \in c(x)$.

Using Lemma 2.1 we are going to provide a transfer condition for coalgebraic bisimulation between two simple Segala systems and see that in case of equivalences one gets the same transfer condition as in Definition 3.7. We have that a relation $R \subseteq X \times Y$ is a bisimulation between two simple Segala systems $c\colon X \to \mathcal{P}(A \times \mathcal{D}X)$ and $d\colon Y \to \mathcal{P}(A \times \mathcal{D}Y)$ if and only if

$$\langle x, y\rangle \in R \implies \langle c(x), d(y)\rangle \in \mathrm{Rel}(\mathcal{P}(A \times \mathcal{D}))(R).$$

Using the modular properties of relation liftings, by Lemma 3.6(vi),

$$\langle c(x), d(y)\rangle \in \mathrm{Rel}(\mathcal{P}(A \times \mathcal{D}))(R)$$

if and only if

$$\begin{aligned}
&(\forall \langle a,\mu\rangle \in c(x), \exists \langle a',\mu'\rangle \in d(y):\ \langle\langle a,\mu\rangle, \langle a',\mu'\rangle\rangle \in \mathrm{Rel}(A \times \mathcal{D})(R))\ \wedge\\
&(\forall \langle a',\mu'\rangle \in d(y), \exists \langle a,\mu\rangle \in c(x):\ \langle\langle a,\mu\rangle, \langle a',\mu'\rangle\rangle \in \mathrm{Rel}(A \times \mathcal{D})(R))
\end{aligned}$$

which by Lemma 3.6(vii) is equivalent to

$$\begin{aligned}
&\big(\forall \langle a,\mu\rangle \in c(x), \exists \langle a',\mu'\rangle \in d(y):\ \langle a,a'\rangle \in \mathrm{Rel}(A)(R)\ \wedge\ \langle \mu,\mu'\rangle \in \mathrm{Rel}(\mathcal{D})(R)\big)\ \wedge\\
&\big(\forall \langle a',\mu'\rangle \in d(y), \exists \langle a,\mu\rangle \in c(x):\ \langle a,a'\rangle \in \mathrm{Rel}(A)(R)\ \wedge\ \langle \mu,\mu'\rangle \in \mathrm{Rel}(\mathcal{D})(R)\big).
\end{aligned}$$

Applying Lemma 3.6(ii),(v), we get the following equivalent condition

$$\begin{aligned}
&\big(\forall \langle a,\mu\rangle \in c(x), \exists \langle a',\mu'\rangle \in d(y):\ a = a'\ \wedge\ \mu\overline{R}\mu'\big)\ \wedge\\
&\big(\forall \langle a',\mu'\rangle \in d(y), \exists \langle a,\mu\rangle \in c(x):\ a = a'\ \wedge\ \mu\overline{R}\mu'\big).
\end{aligned}$$

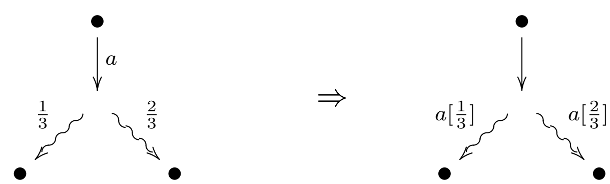
Finally, we rewrite the last condition using transition notation and obtain the transfer condition for bisimulation between simple Segala systems:

$$\begin{aligned}
&\text{if } x \xrightarrow{a} \mu,\ \text{then there exists } \mu' \text{ with } y \xrightarrow{a} \mu' \text{ and } \mu\overline{R}\mu',\ \text{and}\\
&\text{if } y \xrightarrow{a} \mu',\ \text{then there exists } \mu \text{ with } x \xrightarrow{a} \mu \text{ and } \mu\overline{R}\mu'.
\end{aligned} \tag{1}$$

If we restrict to coalgebraic bisimulations R on a simple Segala system which are equivalence relations, then as mentioned above $\overline{R} = \equiv_R$. In addition, when R is an equivalence the symmetric part of the transfer condition (1) becomes unnecessary. Hence, the transfer condition (1) is equivalent to the transfer condition of Larsen and Skou [55] (Definition 3.7) for an equivalence relation R on a simple Segala system, showing that coalgebraic and concrete bisimilarity coincide.

3.3. Expressiveness Hierarchy

Having modeled all discrete probabilistic systems as coalgebras, we can now compare their expressiveness using a single coalgebraic result [5, 69]. The question is: when can we consider one type of systems at least as expressive as another? We will soon define expressiveness precisely, its intuitive meaning being that a more expressive type can model all systems of a less expressive type. According to this intuition, it is clear that stratified systems are at least as expressive as Markov chains and Vardi systems are at least as expressive as LTS since in both cases the latter class is somehow contained in the former one. It is also clear that general Segala systems are at least as expressive as simple Segala systems since each simple Segala system can be considered a general Segala system after pushing each label into the target distribution, as in the example below.



a simple Segala transition $X \to \mathcal{P}(A \times \mathcal{D}X)$ as a Segala transition $X \to \mathcal{P}\mathcal{D}(A \times X)$

The expressiveness criterion chosen in [5, 70, 69], considers a class of systems embedded in another class of systems, in which case the latter is considered at least as expressive as the former, if there exists a translation $\mathcal{T}$ that maps any system of the former class into a system of the latter class keeping the same states such that bisimilarity is both preserved and reflected, i.e., two states are bisimilar in the original system if and only if they are bisimilar in the translated one. Since bisimilarity in all cases coincides with coalgebraic bisimilarity and the systems are modeled as coalgebras, we present a coalgebraic way of creating such translations.

We use translations of F-coalgebras into G-coalgebras in order to compare the expressiveness of coalgebras for different functors F and G. Such a translation can easily be obtained from a natural transformation between the two functors under consideration. A natural transformation $\tau\colon F \Rightarrow G$ is a set-indexed family of maps $\tau_X\colon FX \to GX$ that satisfies the naturality condition: for any map $f\colon X \to Y$, $\tau_Y \circ Ff = Gf \circ \tau_X$. A natural transformation $\tau\colon F \Rightarrow G$ induces a functor $\mathcal{T}_\tau\colon \mathsf{Coalg}_F \to \mathsf{Coalg}_G$ defined as [64]

$$\mathcal{T}_\tau\left(X \xrightarrow{c} FX \right) = \left(X \xrightarrow{c} FX \xrightarrow{\tau_X} GX \right) \quad \text{and} \quad \mathcal{T}_\tau h = h.$$

The induced functor is a translation map that preserves homomorphisms and thus preserves bisimilarity. For reflection of bisimilarity, we impose injectivity condition on the natural transformation [5, 69].

Proposition 3.8. *Let F and G be two functors on* **Sets**. *If $\tau\colon F \Rightarrow G$ is a natural transformation with injective components τ_X and the functor F preserves weak pullbacks, then the induced functor $\mathcal{T}_\tau\colon \mathsf{Coalg}_F \to \mathsf{Coalg}_G$ preserves and reflects bisimilarity.*

Interestingly, the proof of this result uses cocongruences, i.e., behavioral equivalence. One shows that if $\tau\colon F \Rightarrow G$ is a natural transformation with injective components, then $\mathcal{T}_\tau$ preserves and reflects behavioral equivalence (without imposing any conditions on the functors). In the proof of preservation of behavioral equivalence [5], one uses the diagonal fill-in property to show that the mediating coalgebra structure factors as τ after an F-coalgebra structure, for which the change of directions (cospan vs. span) is handy. Since behavioral equivalence and bisimilarity coincide for weak pullback preserving functors, one gets reflection of bisimilarity in case F preserves weak pullbacks. As noted above, all our functors preserve weak pullbacks, so in order to embed a class of F-coalgebras into a class of G-coalgebras all we need is a natural transformation with injective components from F to G. We write $\mathsf{Coalg}_F \to \mathsf{Coalg}_G$ if there is an embedding of F-coalgebras into G-coalgebras that preserves and reflects bisimilarity. Providing suitable natural transformations with injective components we can build the hierarchy of discrete probabilistic systems [5, 69] presented in Figure 2. Each concrete translation is strict in the sense that the translation map is not surjective. However, in general it is very difficult to argue that any arrow in the hierarchy is strict due to the nature of the embedding definition.

All used natural transformations are well-known ones, for example the natural transformation for the translation of simple Segala to general Segala systems is obtained from the strength of the distribution functor $\mathrm{st}^{\mathcal{D}}$. It is

$$\mathcal{P}\,\mathrm{st}^{\mathcal{D}}\colon \mathcal{P}(A \times \mathcal{D}) \Rightarrow \mathcal{P}\mathcal{D}(A \times \_\,)$$

where the strength of the distribution functor at X maps $a \in A$ and $\mu \in \mathcal{D}X$ to $st^{\mathcal{D}}_X(a, \mu) = \mu_a \in \mathcal{D}(A \times X)$ such that $\mu_a(b, x) = \mu(x)$ for $a = b \in A$ and $x \in X$, and $\mu_a(b, x) = 0$ for $a \neq b$.

3.4. Are probabilities just a special case?

Although they might seem different at first sight, the finitary powerset functor $\mathcal{P}_f$ (mapping a set to its finite subsets) and the finitely supported

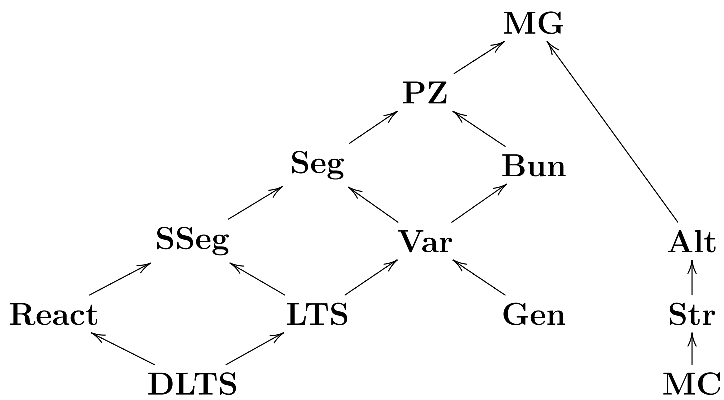


Figure 2: Hierarchy of discrete probabilistic systems

(sub)probability distribution functor $\mathcal{D}_f$ can be seen as instances of the same thing. There exists a more general functor, a (subfunctor of a) functor of finitary monoid valuations, that subsumes both. We start with the definition of the finitary monoid valuations functor [31, 32] which has attracted quite some attention lately.

Let $\mathbf{M} = \langle M, +, 0 \rangle$ be a commutative monoid. An $\mathbf{M}$-valuation on a set X is a function $v\colon X \to M$ and its support is $\mathrm{supp}(v) = \{x \in X \mid v(x) \neq 0\}$.

The finitary functor of monoid valuations $\mathbf{M}_f^{-}\colon \mathbf{Sets} \to \mathbf{Sets}$ for the monoid $\mathbf{M}$ maps a set X to the set of all finitely supported valuations on X,

$$\mathbf{M}_f^X = \{v\colon X \to M \ \mid \mathrm{supp}(v) \text{ is finite } \}$$

and a function $g\colon X \to Y$ to $\mathbf{M}_f^g\colon \mathbf{M}_f^X \to \mathbf{M}_f^Y$ given by

$$(\mathbf{M}_f^g)(v) = \lambda y.v[f^{-1}(\{y\})]$$

where for a valuation $v\colon X \to M$ and a subset $X' \subseteq X$ we write

$$v[X] = \sum_{x \in X'} v(x)$$

and the sum is defined due to the finite support property.

The finitely supported monoid valuation functor $\mathbf{M}_f^-$ and its properties have been studied by Gumm and Schröder [31, 32], showing that the functor preserves nonempty weak pullbacks along injective maps if the monoid is positive (the only invertible element is 0), and it preserves nonempty pullbacks if additionally the monoid is refinable (if $m_1+m_2=n_1+n_2$, then there exist $l_{1,1}, l_{1,2}, l_{2,1},$ and $l_{2,2}$ such that $l_{1,1}+l_{1,2}=m_1,\ l_{2,1}+l_{2,2}=m_2,\ l_{1,1}+l_{2,1}=n_1,$ and $l_{1,2}+l_{2,2}=n_2$). The same functor was also used recently by Bonchi et al. [8, 68], for deriving syntax and axioms for quantitative behaviors, where the authors show that the functor is bounded and hence has a final coalgebra.

Consider the two-valued commutative monoid $\mathbf{2}=\langle\{0,1\},\vee,0\rangle$. The functor $\mathbf{2}_f^-$ coincides with the finitary powerset functor. Note that $\mathbf{2}$ is positive and refinable, so one could also derive the weak pullback preservation of the finitary powerset functor from the results of [31, 32] as well as the existence of a final coalgebra from [68]. Another instance of the monoid valuation functor $\mathbf{M}_f^-$ is the finitely supported multiset functor $\mathcal{M}$ which maps a set X to the set of all finitely supported multisets on X. Namely, $\mathcal{M}X=\{m\colon X\to\mathbb{N}\mid \mathrm{supp}(m)\text{ is finite }\}=\mathbf{N}_f^X$ for the monoid of natural numbers $\mathbf{N}=\langle\mathbb{N},+,0\rangle$. This monoid is positive and refinable as well.

The finitely supported (sub)probability distributions functor is not exactly an instance of the finitely supported monoid valuations functor. Rather it is an instance of its subfunctor, due to the condition that each (sub)distribution assigns probability (less than or equal to) 1 to the set on which it is defined.

Let $\mathbf{M}=\langle M,+,0\rangle$ be a commutative monoid, and S a subset of M. The functor of finitely supported monoid valuations in S, $\mathbf{M}_{S,f}^-\colon\mathbf{Sets}\to\mathbf{Sets}$ maps a set X to the set

$$\mathbf{M}_{S,f}^X=\{v\colon X\to M\ \mid \mathrm{supp}(v)\text{ is finite and }v[X]\in S\}$$

and a function in the same way as $\mathbf{M}_f^-$ does. Clearly, $\mathbf{M}_f^-=\mathbf{M}_{M,f}^-$. The functor $\mathbf{M}_{S,f}^-$ was used by Klin [50] for deriving structural operational semantics for weighted transition systems.

Consider the commutative additive monoid of non-negative real numbers $\mathbf{R}=\langle\mathbb{R}^{\geq 0},+,0\rangle$ and its subsets the interval $[0,1]$ and the singleton $\{1\}$. The functor $\mathbf{R}_{[0,1],f}^-$ coincides with the finitely supported subprobability distribution functor, whereas $\mathbf{R}_{\{1\},f}^-$ coincides with the finitely supported probability distribution functor. The monoid $\mathbf{R}$ is also positive and refinable, so $\mathbf{R}_f^-$ preserves nonempty weak pullbacks and has a final coalgebra.

Note that (sub)probability distributions have also other properties that are not explicitly captured by a functor $\mathbf{M}_{S,f}^-$. For example, for any subdistribution μ on a set X, not only $\mu[X]\in[0,1]$ but also $\mu[X']\in[0,1]$ for any subset $X'\subseteq X$. In other words, any subdistribution is a discrete subprobability measure, which is not obvious from the definition of $\mathbf{M}^{-}_{S,f}$. In order to make this property of subdistributions explicit, one needs to highlight some additional properties of the involved monoid: the functor $\mathbf{M}^{-}_{S,f}$ can be specialized further as in [45] requiring that $\mathbf{M}$ be a partially ordered monoid with the property that $x \leq x + y$ for all $x, y \in M$ and S a downward-closed set.

It is interesting to note that algebraic properties of the involved monoid may have far-reaching consequences on the behavior of the coalgebras of a monoid valuations functor. For example, as noted in [45], the fact that Boolean logic with standard modalities is expressive for bisimilarity of the finitary powerset functor in contrast to finite conjunctions being sufficient for the expressivity of probabilistic/graded modal logic for the finitary probability (sub)distribution/multiset functor, is a consequence of an algebraic property of the involved monoids: both $\mathbf{R}$ and $\mathbf{N}$ are cancellative monoids ($x + y = x + z \Rightarrow y = z$), whereas $\mathbf{2}$ is not.

An alternative precise way to model (and generalize) the distribution functor $\mathcal{D}_f$ is via valuations to effect algebras (which are partial commutative monoids with "orthosupplement") [43]. Effect algebras capture key properties of the unit interval [0, 1].

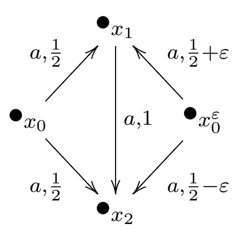
In order to capture the monad structure of both $\mathcal{P}_f$ and $\mathcal{D}_f$, additional algebraic structure is needed: instead of monoid valuations or effect-algebra valuations, one considers semiring valuations with additional properties or valuations to effect monoids [43] (effect algebras with multiplication satisfying certain properties). The unit interval [0,1] is an example of an effect monoid.

3.5. Coalgebra Results on Discrete Probabilistic Systems

In the past decade, there has been quite some research on coalgebra involving discrete probabilistic systems. Due to their variety and inductive definition, but also due to their importance in concurrency theory and verification in general, discrete probabilistic systems are popular examples in many works in the area of coalgebra. Moreover, all general results on coalgebras on **Sets** could be instantiated to probabilistic systems and in some cases they provide existing notions and results to justify the general ones, but more importantly in some cases they provide completely new results, e.g. [2, 5, 17, 65, 52, 60] that should be of interest to the probabilistic systems community. We mention a few here. In addition, there have been coalgebraic approaches to solving particular problems for probabilistic systems that are not of a generic kind, but coalgebraic notions are key ingredients in obtaining the results, as in the line of work on behavioral distances described right below.

Behavioral Distances

A significant amount of work deals with behavioral distances on discrete and continuous probabilistic systems as coalgebras. We focus on the work of van Breugel and Worrell [14, 13, 15, 16] since it is essentially coalgebraic. For more information on other work on behavioral distances we refer the reader to [15, 59] that describe the situation and provide references to the literature. The main motivation behind this work is to define a quantitative analogue of bisimilarity, that will not only relate bisimilar states but provide metrics on how close the behavior of states are. The work of van Breugel and Worrell applies to reactive probabilistic systems. To start with, we borrow the following example from [15]. Consider the reactive probabilistic system below. Its states x_0 and x_0^ε



are only bisimilar if $\varepsilon = 0$. However, they behave almost the same for small ε different from 0. The authors define a pseudometric (two elements can have distance 0 even if not equal) on the states of such coalgebras, using finality in a category of coalgebras on pseudometric spaces and nonexpansive maps, for a functor describing reactive systems. Since the construction is general, i.e., it works for both continuous and discrete systems, we will get back to the underlaying theory in Section 4. Here we only mention that the authors provide algorithms [13, 16] for computing the behavior distances on finite-state systems, which rely on solving a particular linear programming model. Moreover, they provide a nice comparison to other existing (non-coalgebraic) notions of behavioral distance in [14].

Modal Logics

The well-developed theory of coalgebraic modal logics starting from Moss, extensively expanded by Cirstea, Kurz, Pattinson, Schröder, Venema, and others, which also involves work on modular logics for inductively defined functors, frequently employs discrete probabilistic systems as examples, cf. e.g. [57, 17, 18, 20, 19, 65, 60]. Many of the general results when instantiated to probabilistic systems provide new insight or results that the probabilistic systems community may not be aware of, e.g. [17, 65, 60]. The value of the results on coalgebraic modal logics is inverse proportional to the space devoted here, the main reason for such a treatment being that more information on coalgebraic modal logics can be found in the survey article by Kupke and Pattinson [53] in this very issue.

Structural Operational Semantics

Coalgebra theory leads to significant results on discrete probabilistic systems in structural operational semantics. Bartels [2, 3] instantiated the general categorical framework of bialgebras for structural operational semantics by Turi and Plotkin [73] to reactive probabilistic systems and simple Segala systems, providing the first formats for structural operational semantics that guarantee that the operations are well-defined and well-behaved. The latter means that bisimilarity/behavioral equivalence is a congruence for each operation defined by rules of the given format. Recently in the work of Klin and Sassone [52] the same framework was instantiated to stochastic systems. More information on structural operational semantics and the work mentioned here can be found in the survey article by Klin [51] in this issue. Worth noting here is that stochastic transition systems, describing models like labeled CTMCs, are reactive versions of coalgebras of the monoid valuation functor $\mathbf{R}_f^-$, namely they are coalgebras of the functor $(\mathbf{R}_f^-)^A$.

Traces

We mentioned before that $\mathcal{D}$ (and each of its variants) is not only a functor but also a monad modeling probabilistic choice. The monad structure is important, e.g. for linear-time trace semantics. The coalgebraic trace theory of Hasuo et al. [37] applies to TF-coalgebras on **Sets** where T is a monad modeling branching, and F is a functor modeling linear behavior. Under certain conditions on T and F, the functor F lifts to the Kleisli category of T (with objects sets and morphisms $f\colon X \to Y$ being functions $f\colon X \to TY$), where branching is hidden. The main result of [37] shows that, under additional order-theoretic conditions on T and F, the initial F-algebra in **Sets** is a final coalgebra of the lifting of F in the Kleisli category. The final coalgebra semantics in the Kleisli category provides trace semantics for

TF-coalgebras. The theory is applicable to the subprobability distribution monad $\mathcal{D}_{\leq 1}$ (but not to the variants) and the functor $F = 1 + A \times \_$ resulting in the usual trace semantics for generative probabilistic systems.

The powerset functor $\mathcal{P}$ is also a monad, modeling non-deterministic choice. However, the combination $\mathcal{PD}$ is unfortunately not a monad [76, 75] (due to nonexistence of a required distributive law). In the work of Varacca [76, 75], the problem of constructing a monad for non-determinism and probability is addressed and two solutions are proposed: either to replace the distribution monad by a new monad of indexed valuations, which combined with the powerset provides a monad due to the existence of suitable distributive law, or to use one monad of convex subsets, on a different category, for the whole combination. The latter monad was recently used to describe traces of simple Segala systems by Jacobs [42].

Recent work on generic forward and backward simulations by Hasuo [34, 36] also instantiates to generative probabilistic systems, providing new notions of forward and backward simulations for these systems. Forward and backward simulations are sound in the sense that they imply trace inclusion and are therefore useful as a proof method for trace equivalence. An interesting application of this method to probabilistic anonymity and probable innocence can be found in the work of Hasuo et al. [38, 35].

Kleene Algebras

A recent line of research in coalgebra by Silva, Bonchi et al., focuses on deriving languages of generalized regular expressions, and their sound and complete axiomatizations, for transition systems modeled as coalgebras, generalizing the results of Kleene, on regular languages and deterministic finite automata, and Milner, on regular behaviors and finite LTS. This work focuses on an inductively defined class of functors which involves the monoid valuation functor and also leads to results for discrete probabilistic systems [8, 68].

Weak Bisimulation

A long open problem in coalgebra theory is the problem of a coalgebraic characterization of weak bisimulation. One of the few approaches towards this [71] is actually inspired by (and somewhat tailored to) concrete work on probabilistic weak bisimulation. The approach is to transform a given system to its "double-arrow" system whose transitions are all "weak transitions" of the original system, and then define weak bisimulation as a bisimulation on the transformed system. The paper lists some required properties of such a transformation, but no generic construction is identified that actually provides the transformation. One can come up with a suitable transformation for generative probabilistic systems and for LTS.

4. Continuous Probabilistic Systems

By continuous probabilistic systems we mean probabilistic systems on continuous state spaces. The notion of interest is no longer a (discrete) probability distribution, but a (continuous) probability measure. As explained well by Panangaden [59], the point is that the elements of the space are no longer the "atoms" of the measure, each element may have probability zero and yet a subset may have a nonzero probability. The work on continuous probabilistic systems is in large part coalgebraic, or more precisely it is categorical and coalgebra-aware. Most of the work on continuous probabilistic systems is due to Desharnais, Panangaden [59] et al., on labeled Markov processes, to Doberkat [27, 28] on stochastic relations, and to van Breugel and Worrell [15] on behavioral distances. Our aim here is to present a very brief, and as a consequence somewhat shallow, overview of continuous probabilistic systems as coalgebras. Coalgebraic treatment of continuous probabilistic systems also originates in the work of de Vink and Rutten [80], on the category of ultrametric spaces and nonexpansive functions. We start with the basic definition of a measurable space, measurable function, and (sub)probability measure.

A measurable space $\mathcal{X} = \langle X, S_X \rangle$ is a pair of a set X and a σ-algebra S_X on X, i.e., a collection of subsets of X, $S_X \subseteq \mathcal{P}X$ with the properties

(1) $\emptyset \in S_X$,

(2) $S \in S_X \implies X \setminus S \in S_X$, and

(3) $\bigcup_i S_i \in S_X$, for any countable family of measurable sets $S_i \in S_X$.

The elements of the σ-algebra are called measurable sets.

Let $\mathcal{X} = \langle X, S_X \rangle$ and $\mathcal{Y} = \langle Y, S_Y \rangle$ be measurable spaces. A measurable function from $\mathcal{X}$ to $\mathcal{Y}$ is any function $f\colon X \to Y$ with the property that inverse image of a measurable set is a measurable set, i.e., $f^{-1}(S_Y) \subseteq S_X$.

A probability measure on $\mathcal{X}$ is a function $\mu\colon S_X \to [0, 1]$ with the properties that

(1) $\mu(\emptyset) = 0$,

(2) $\mu(X) = 1$,

(3) $\mu(\bigcup_i S_i) = \sum_i \mu(S_i)$, for any countable family of pairwise-disjoint measurable sets $S_i \in S_X$.

A function $\mu\colon S_X \to [0,1]$ is a subprobability measure on $\mathcal{X}$ if it satisfies the properties (1) and (3), which means that the measure of the whole space is less than or equal to 1 but not necessarily 1.

Definition 4.1. *A Markov process on a measurable space $\mathcal{X} = \langle X, S_X \rangle$ is a pair $\langle X, (\mu_x)_{x \in X} \rangle$ with X being the state set and, for each $x \in X$, $\mu_x\colon S_X \to [0,1]$ is a transition subprobability measure, i.e., a subprobability measure with the additional property that for each $S \in S_X$ the function $m_S\colon X \to [0,1]$ given by $m_S(x) = \mu_x(S)$ is a measurable function from $\mathcal{X}$ to the measurable space $[0,1]$ with the σ-algebra of Borel sets, the smallest σ-algebra containing all open sets.*

Hence, a Markov process is a continuous-space transition system. Given a Markov process on $\mathcal{X}$, one interprets $\mu_x(S)$ as the probability that the system starting in state x makes a transition to one of the states in S. As in the case of Markov chains, labels make the behavior of Markov processes more interesting, leading to labeled Markov processes. Labels are reactive, i.e., in a labeled Markov process on $\mathcal{X} = \langle X, S_X \rangle$, for each $x \in X$ and each label $a \in A$, $\mu_{x,a}\colon S_X \to [0,1]$ is a transition subprobability measure. We borrow the following example from [23, 59].

Consider a process with two labels a, b. The state space is the real plane $\mathbb{R}^2$. When the process makes an a-move from state (x_0, y_0), it jumps to (x, y_0), where the probability distribution for x is given by the density $K_\alpha \exp(-\alpha(x-x_0)^2)$, where $K_\alpha = \sqrt{\alpha/\pi}$ is the normalizing factor. When it makes a b-move it jumps from state (x_0, y_0) to (x_0, y), where the distribution for y is given by the density function $K_\beta \exp(-\beta(y-y_0)^2)$. The meaning of these densities is as follows. The probability of jumping from (x_0, y_0) to a state with x-coordinate in the interval $[s,t]$ under an a-move is $\int_s^t K_\alpha \exp(-\alpha(x-x_0)^2)dx$. Note that the probability of jumping to any given point is, of course, 0. In this process the interaction with the environment controls whether the jump is along the x-axis or along the y-axis but the actual extent of the jump is governed by a probability distribution. With a single label, the process amounts to an ordinary (time-independent) Markov process.

We refer the reader to [23, 24, 25, 59] for more interesting examples and a very good explanation of the importance of modeling and verifying such systems.

4.1. Coalgebraic Modeling

In order to model Markov processes as coalgebras one needs a suitable category and a suitable functor. The most natural choice for a category is **Meas**, the category of measurable spaces and measurable functions. For different reasons, some of which are explained in Section 4.2, in most of the work on Markov processes different categories were considered. In this brief survey we mainly remain in **Meas** and argue that for coalgebraic treatment of Markov processes it suffices to work with general measurable spaces unless one prefers bisimilarity to behavioral equivalence. In addition, we explain how probabilistic systems are modeled in categories of metric spaces for the purpose of studying behavioral distances.

No matter which category is considered, all works on continuous probabilistic systems agree on the functor: the Giry functor (actually monad) [29]. The initial idea to look for such a monad goes back to unpublished work by Lawvere [56].

Definition 4.2. *Given a measurable space $\mathcal{X} = \langle X, S_X \rangle$ the Giry functor*

$$\mathcal{G} \colon \mathbf{Meas} \to \mathbf{Meas}$$

maps $\mathcal{X}$ to the measurable space

$$\mathcal{G}\mathcal{X} = \langle \mathcal{G}_{\mathcal{X}}, \mathcal{S}\mathcal{G}\mathcal{X} \rangle$$

where $\mathcal{G}_{\mathcal{X}}$ is the set of subprobability measures on $\mathcal{X}$ and $\mathcal{S}\mathcal{G}\mathcal{X}$ is the smallest σ-algebra making all evaluation maps ev_S, for $S \in S_X$, measurable, where for $S \in S_X$, the evaluation map $ev_S \colon \mathcal{G}_{\mathcal{X}} \to [0, 1]$ is given by $\mu \mapsto \mu(S)$.

A morphism $f \colon \mathcal{X} \to \mathcal{Y}$, i.e., a measurable function $f \colon X \to Y$ is mapped to $\mathcal{G}f \colon \mathcal{G}\mathcal{X} \to \mathcal{G}\mathcal{Y}$ where $\mathcal{G}f(\mu) = \mu \circ f^{-1}$.

As noticed in [45, 58], the σ-algebra $\mathcal{S}\mathcal{G}\mathcal{X}$ is generated by the collection:

$$\big\{ L_r(S) \mid r \in \mathbb{Q} \cap [0, 1], S \in S_X \big\}$$

where

$$L_r(S) = \{\mu \in \mathcal{G}_{\mathcal{X}} \mid \mu(S) \geq r\} = ev_S^{-1}([r, 1])$$

which are the usual probabilistic modalities, providing an intrinsic connection between probabilistic modal logics and Giry coalgebras.

Moss and Viglizzo [58] show that every functor on the category **Meas** built from the identity and constant functors using products, coproducts, and the Giry functor has a final coalgebra. The construction uses modal logics: the elements of the final coalgebra are theories (sets of modal formulas) satisfied by states in all possible coalgebras. Viglizzo [79] also provided another construction of a final coalgebra for the same class of functors, avoiding the logic.

Proposition 4.3. *Markov processes are exactly the $\mathcal{G}$-coalgebras on* **Meas**.

The proposition holds since, by the construction of the σ-algebra $\mathcal{SGX}$ for $\mathcal{X} = \langle X, S_X \rangle$, we have that a function $c\colon X \to \mathcal{G}_{\mathcal{X}}$ is measurable if and only if $ev_S \circ c$ is measurable for all $S \in S_X$. Therefore, $c\colon \mathcal{X} \to \mathcal{GX}$ is a Giry coalgebra if and only if $\langle X, (\mu_x)_{x \in X} \rangle$ with $\mu_x = c(x)$ is a Markov process, since $ev_S \circ c = m_S$.

A large part of the research on continuous probabilistic systems focuses on stochastic relations and stochastic coalgebraic logic [27, 28]. A stochastic relation in **Meas** is a Kleisli morphism of the Giry monad, i.e., a map $s\colon \mathcal{X} \to \mathcal{GY}$. Hence, every Markov process (a Giry coalgebra on **Meas**) is a stochastic relation (for $\mathcal{X} = \mathcal{Y}$), but not the other way around. Most of the research on stochastic relations is located in other categories (analytic or Polish spaces) than general measurable spaces, for reasons that we will discuss in the following section.

4.2. Bisimilarity Problems and Solutions

It is very difficult to show that bisimilarity is an equivalence for Markov process, in particular it is difficult to show that it is transitive [59, 27, 21]. The reason is the following. Assume $c \sim d$ and $d \sim e$ for three Markov processes, coalgebras on **Meas**, $c\colon \mathcal{X} \to \mathcal{GX}$, $d\colon \mathcal{Y} \to \mathcal{GY}$ and $e\colon \mathcal{Z} \to \mathcal{GZ}$. Let $\langle \langle \mathcal{Q}, q \rangle, q_1, q_2 \rangle$ be a witnessing bisimulation for $c \sim d$ and $\langle \langle \mathcal{R}, r \rangle, r_1, r_2 \rangle$ for $d \sim e$. Hence we have the following situation:

$$\begin{tikzcd}[column sep=small]
 & \langle \mathcal{Q}, q \rangle \arrow[dl, "q_1"'] \arrow[dr, "q_2"] & & \langle \mathcal{R}, r \rangle \arrow[dl, "r_1"'] \arrow[dr, "r_2"] & \\
\langle \mathcal{X}, c \rangle & & \langle \mathcal{Y}, d \rangle & & \langle \mathcal{Z}, e \rangle
\end{tikzcd}$$

Then, for transitivity, it would suffice to complete the following cospan on the left, to a square on the right.

$$
\begin{tikzcd}
\langle \mathcal{Q}, q\rangle \arrow[dr, "q_2"'] & & \langle \mathcal{R}, r\rangle \arrow[dl, "r_1"] \\
& \langle \mathcal{Y}, d\rangle &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \langle \mathcal{P}, p\rangle \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\
\langle \mathcal{Q}, q\rangle \arrow[dr, "q_2"'] & & \langle \mathcal{R}, r\rangle \arrow[dl, "r_1"] \\
& \langle \mathcal{Y}, d\rangle &
\end{tikzcd}
$$

which turns out to be difficult for Markov processes. The category **Meas** has pullbacks, so if the coalgebra functor would preserve (weak) pullbacks then one can complete the square. However, as shown by Viglizzo [78], the Giry monad on **Meas** does not preserve weak pullbacks.

A large amount of research dealt with this problem and in all cases the shift to other categories was taken. In the first coalgebraic treatment of Markov processes by de Vink and Rutten [80], the category of ultrametric spaces and nonexpansive maps was considered. The main motivation for doing so was reusing a theorem that guarantees existence of a final coalgebra for locally contractive functors on ultrametric spaces [74]. The authors mention that it would be good to have weak pullback preservation which would result in a full-abstractness result implying that bisimilarity is well-behaved, but leave the issue for future work. A very involved construction [23] of a so-called semi-pullback provides a way to complete the cospan above to a square, for (labeled) Markov processes on analytic spaces, showing that bisimilarity is transitive. This result was followed by a deep analysis of semi-pullbacks by Doberkat [26, 27] for stochastic relations and thus Markov processes on Polish and analytic spaces. None of this proves that bisimilarity of (labeled) Markov processes on **Meas** is not transitive, but it is quite likely so, as conjectured already in [23].

Finally, we note that there is no problem with behavioral equivalence: it is always an equivalence in categories with pushouts, as is the case with **Meas**. This fact was explicitly recognized first by Danos et al. [21] where cocongruences (event bisimulations) were considered for (labeled) Markov processes. Event bisimulations provide a concrete definition of behavioral equivalence.

4.3. Comments on Modal Logics for Continuous Probabilistic Systems

A significant part of this work is related to modal logic, in particular the first result showing that negation-free probabilistic modal logic (with finite conjunctions) is expressive for bisimilarity of probabilistic systems comes from the work on labeled Markov processes [7, 23, 59] on Polish/analytic spaces. In the context of stochastic relations, different bisimulation notions are considered in order to show expressivity of probabilistic modal logic with finite conjunctions, again on Polish/analytic spaces [27, 28]. Expressivity, also called Hennessy-Milner property, of a logic for bisimilarity means that two states are bisimilar if and only if they satisfy the same formulas.

Danos et al. [21] showed that negation-free probabilistic modal logic (with finite conjunctions) is expressive for behavioral equivalence of Markov processes on **Meas**. Recently [45], Jacobs and the author of this paper have presented another proof of the expressivity of probabilistic modal logic with finite conjunctions (and no negations) with respect to behavioral equivalence for Markov processes as coalgebras of the Giry functor on **Meas**, by providing a dual adjunction between **Meas** and **MSL**, the category of meet-semilattices. The same paper also provides a proof of the expressivity of the same logic for Markov chains, via a related dual adjunction between **Sets** and **MSL**.

4.4. Behavioral Distances via Finality: Metric vs. Measurable Spaces

The work on behavioral pseudometrics for probabilistic systems by van Breugel and Worrell, cf. e.g. [14, 15], builds-up on the work of de Vink and Rutten [80] in the sense that continuous (and as a special case also discrete) probabilistic systems are modeled as coalgebras of a variant of the Giry functor on a category of 1-bounded pseudometric spaces.

A 1-bounded pseudometric space is a pair $\langle X, d_X \rangle$ where X is a set and $d_X \colon X \times X \to [0, 1]$ is a pseudometric satisfying the symmetry condition and the triangle inequality. It is 1-bounded since all distances are bounded by 1, and pseudo since different elements may have distance 0. 1-bounded pseudometric spaces and nonexpansive maps (functions that do not increase distances) form a category. The authors show the existence of a final coalgebra for locally contractive functors in this category, by slightly generalizing the result of [74].

Every (pseudo)metric space is a measurable space when equipped with the Borel σ-algebra, the smallest σ-algebra containing all open sets. In order to model probabilistic systems as coalgebras on the category of pseudometric spaces, the following definition yields a functor M: it maps a metric space $\langle X, d_X \rangle$ to the set of all (tight) Borel probability measures on X, with the Hutchinson metric. On functions, M is defined just like the Giry functor. In order to model probabilistic systems, the authors modify the functor to include (reactive) labels and a so-called discount factor $c \in (0, 1)$, resulting in a functor P. The discount factor ensures that the functor P is locally contractive, and intuitively it discounts the future: the smaller the discount factor, the more the future is discounted. As a consequence, P has a final coalgebra with carrier $\Omega = \langle fix(P), d_{fix(P)} \rangle$. The space Ω is a compact metric space.

Now, by finality, this induces a metric on the states on any other P-coalgebra, as follows. Let $\mathcal{X} = \langle X, d_X \rangle$ be a 1-bounded pseudometric space and $c\colon \mathcal{X} \to P\mathcal{X}$ a P-coalgebra. Let $\varphi\colon \mathcal{X} \to \Omega$ be the unique homomorphism obtained by finality. For any two states $x, y \in X$ the behavioral distance from x to y, also called coalgebraic distance, is defined by

$$d(x, y) = d_{fix(P)}(\varphi(x), \varphi(y)).$$

The obtained behavior distance is a pseudometric: bisimilar states have distance 0. Moreover, states have distance 0 if and only if they are bisimilar. As explained by the authors [15]: the distance between states is a trade-off between the depth of observations needed to distinguish the states and the amount each observation differentiates the states.

The authors present several useful characterizations of the introduced coalgebraic pseudometric, and a comparison to existing (non-coalgebraic) behavior metrics of probabilistic systems. In particular they show that the coalgebraic pseudometric coincides (up to the discount factor) with the first behavioral pseudometric, the logical pseudometric of Desharnais et al. [25]. They also present an algorithm for computing behavioral distances on finite systems based on a linear programming problem [13, 16]. Moreover, the coalgebraic distance can be approximated, i.e., the finality homomorphism φ is a fixpoint of a certain function and can be computed by a sequence of approximations φ_n each inducing a pseudometric d_n on the states of a P-coalgebra that approximate the pseudometric d. To calculate d with a prescribed degree of accuracy α, the authors show that it suffices to calculate d_i for $i = 1, \ldots, \log_c(\alpha/2)$.

Furthermore, van Breugel et al. characterize approximate bisimilarity independently of the Hutchinson metric (independently of integration) in [10], and in [9] provide a more general final coalgebra theorem for accessible categories and accessible functors, that subsume the categories of coalgebras studied before and cover the case of no discount ($c = 1$). In a later work [12], van Breugel, Sharma, and Worrell present an algorithm for approximating the pseudometric also in case of systems that do not discount the future ($c = 1$).

4.5. Embedding Discrete into Continuous Systems

As expected, Markov chains embed into Markov processes. Clearly, any discrete probability distribution μ on a set X extends to a probability measure, $\overline{\mu}$ on the discrete measurable space $\mathcal{X} = \langle X, \mathcal{P}X \rangle$ with the discrete σ-algebra of all subsets. We have $\overline{\mu}(X') = \mu[X']$. Therefore, any Markov chain "is" a Markov process.

However, Markov chains and Markov processes live in different categories. For a precise embedding, in line with the translation embeddings of Section 3.3, more machinery is needed. As shown in [45], there is a translation functor $\mathcal{T}$ that maps a Markov chain to a Markov process on the same state set so that behavioral equivalence is preserved and reflected. The functor is induced by a suitable injective natural transformation. It is non-trivial to show that behavioral equivalence on a Markov chain $c\colon X \to \mathcal{D}X$ and on the corresponding discrete Markov processes $\mathcal{T}(c)$ coincide [45] providing an embedding as in Section 3.3 from chains into processes.

The situation is as follows

$$\begin{tikzcd}
\mathsf{Coalg}_{\mathcal{D}} \arrow[r, "\mathcal{T}"] \arrow[d] & \mathsf{Coalg}_{\mathcal{G}} \arrow[d] \\
\mathbf{Sets} \arrow[r, "\mathrm{Disc}"] & \mathbf{Meas}
\end{tikzcd}$$

where Disc denotes the functor from **Sets** to **Meas** mapping a set X to the discrete measurable space on X, and the vertical arrows represent forgetful functors mapping a coalgebra to its carrier. The functor Disc is a left adjoint of the forgetful functor from **Meas** to **Sets** mapping a measurable space $\mathcal{X} = \langle X, S_X \rangle$ to the set X. The adjunction plays an important role in the definition of $\mathcal{T}$.

Following this embedding, it may be of interest to build another floor in the hierarchy of probabilistic systems for different (labeled) continuous probabilistic systems on the category **Meas**.

5. Conclusions

We have presented a brief survey of discrete and continuous probabilistic systems as coalgebras. Via the probability distribution functor on **Sets** and the Giry functor on **Meas** (and related categories) probabilistic systems enter coalgebra. Discrete systems are inductively built and therefore present a nice class of examples for other research in coalgebra theory. Treating continuous systems is not so straightforward and requires moving to other categories than **Meas** in order that bisimilarity is an equivalence. A solution is to consider behavioral equivalence instead of bisimilarity.

Another fruitful direction in the coalgebraic treatment of probabilistic systems that we have briefly highlighted is through metric spaces, providing behavioral distances between states rather than equivalence relations on states. This work is significant for two reasons: (1) behavioral pseudometrics provide not only information about whether states of a probabilistic system behave in the same way or not, but also quantitative information about how close or how far apart the behavior of such states are; (2) part of the work on behavioral metrics is essentially coalgebraic, the pseudometrics being defined via finality.

This survey is already quite long, and we have just flown over the topic of coalgebraic treatment of probabilistic systems. Aware that in many respects we lack in explanation and detail, we hope that at least references to the literature may guide you to your particular destination topic of interest.

Acknowledgements

I am grateful to the organizers of CMCS 2010 for providing me an opportunity to give a talk and publish this work. Furthermore, I would like to thank Dirk Pattinson for discussions and pointers to references; Lutz Schröder for a comment during my talk at CMCS 2010, a prompt answer to a question, and just-in-time pointers to the literature; and Mathieu Tracol for an informative discussion on behavioral pseudometrics. Last but not least, I thank the anonymous referees for their thorough reviews, feedback, and very useful comments.

References

[1] C. Baier. *On Algorithmic Verification Methods for Probabilistic Systems.* Habilitationsschrift, FMI, Universitaet Mannheim, 1998.

[2] F. Bartels. GSOS for probabilistic transition systems. *Electronic Notes in Theoretical Computer Science*, 65(1), 2002. Proc. CMCS'02.

[3] F. Bartels. *On generalised coinduction and probabilistic specification formats: distributive laws in coalgebraic modelling.* PhD thesis, Vrije Universiteit, Amsterdam, 2004.

[4] F. Bartels, A. Sokolova, and E. d. Vink. A hierarchy of probabilistic system types. *Electronic Notes in Theoretical Computer Science*, 82(1), 2003. Proc. CMCS'03.

[5] F. Bartels, A. Sokolova, and E. d. Vink. A hierarchy of probabilistic system types. *Theoretical Computer Science*, 327:3–22, 2004.

[6] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[7] R. Blute, J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. In *Proc. LICS*, pages 149–158. IEEE, 1997.

[8] F. Bonchi, M. Bonsangue, J. Rutten, and A. Silva. Deriving syntax and axioms for quantitative regular behaviours. In *Proc. CONCUR*, pages 146–162. LNCS 5710, 2009.

[9] F. v. Breugel, C. Hermida, M. Makkai, and J. Worrell. An accessible approach to behavioural pseudometrics. In *Proc. ICALP*, pages 1018–1030. LNCS 3580, 2005.

[10] F. v. Breugel, M. Mislove, J. Ouaknine, and J. Worrell. An intrinsic characterization of approximate probabilistic bisimilarity. In *Proc. FoSSaCS*, pages 200–215. LNCS 2620, 2003.

[11] F. v. Breugel, M. Mislove, J. Ouaknine, and J. Worrell. Domain theory, testing and simulation for labelled Markov processes. *Theoretical Computer Science*, 333:2239–2259, 2005.

[12] F. v. Breugel, B. Sharma, and J. Worrell. Approximating a behavioural pseudometric without discount for probabilistic systems. In *Proc. FoSSaCS*, pages 123–137. LNCS 4423, 2007.

[13] F. v. Breugel and J. Worrell. An algorithm for quantitative verification of probabilistic transition systems. In *Proc. CONCUR*, pages 336–350. LNCS 2154, 2001.

[14] F. v. Breugel and J. Worrell. Towards quantitative verification of probabilistic transition systems. In *Proc. ICALP*, pages 421–432. LNCS 2076, 2001.

[15] F. v. Breugel and J. Worrell. A behavioural pseudometric for probabilistic transition systems. *Theoretical Computer Science*, 331(1):115–142, 2005.

[16] F. v. Breugel and J. Worrell. Approximating and computing behavioural distances in probabilistic transition systems. *Theoretical Computer Science*, 360(1-3):373–385, 2006.

[17] C. Cîrstea. A modular approach to defining and characterising notions of simulation. *Information and Computation*, 204(4):469–502, 2006.

[18] C. Cîrstea. Modularity in coalgebra. *Electronic Notes in Theoretical Computer Science*, 164(1):3–26, 2006.

[19] C. Cîrstea, A. Kurz, D. Pattinson, L. Schröder, and Y. Venema. Modal logics are coalgebraic. In *BCS Int. Acad. Conf.*, pages 128–140. British Computer Society, 2008.

[20] C. Cîrstea and D. Pattinson. Modular construction of complete coalgebraic logics. *Theoretical Computer Science*, 388(1-3):83–108, 2007.

[21] V. Danos, J. Desharnais, F. Laviolette, and P. Panangaden. Bisimulation and cocongruence for probabilistic systems. *Information and Computation*, 204:503–523, 2006.

[22] P. D'Argenio, H. Hermanns, and J.-P. Katoen. On generative parallel composition. *Electronic Notes in Theoretical Computer Science*, 22:105–122, 1998. Proc. PROBMIV'98.

[23] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. *Information and Computation*, 179:163–193, 2002.

[24] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labelled Markov processes. *Information and Computation*, 184(1):160–200, 2003.

[25] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

[26] E.-E. Doberkat. Semi-pulbacks and bisimulations in categories of stochastic relations. In *Proc. ICALP*, pages 996–1070. LNCS 2719, 2003.

[27] E.-E. Doberkat. *Stochastic relations: Foundations for Markov transition systems*. ChapmanHall/CRC, 2007.

[28] E.-E. Doberkat. *Stochastic Coalgebraic Logic*. Springer, 2010.

[29] M. Giry. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, pages 68–85. LNM 915, 1982.

[30] R. v. Glabbeek, S. Smolka, B. Steffen, and C. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proc. LICS*, pages 130–141. IEEE, 1990.

[31] H.-P. Gumm. Copower functors. *Theoretical Computer Science*, 410:1129–1142, 2009.

[32] H.-P. Gumm and T. Schröder. Monoid-labeled transition systems. *Electronic Notes in Theoretical Computer Science*, 44(1):185–204, 2001. Proc. CMCS'01.

[33] H. Hansson. Time and probability in formal design of distributed systems. *Real-Time Safety Critical Systems*, 1, 1994.

[34] I. Hasuo. Generic forward and backward simulations. In *Proc. CONCUR*, pages 406–420. LNCS 4137, 2006.

[35] I. Hasuo. *Tracing Anonymity with Coalgebras*. PhD thesis, Radboud University Nijmegen, 2008.

[36] I. Hasuo. Generic forward and backward simulations II: Probabilistic simulation. In *Proc. CONCUR*, pages 447–461. LNCS 6269, 2010.

[37] I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Logical Methods in Computer Science*, 3(4:11), 2007.

[38] I. Hasuo, Y. Kawabe, and H. Sakurada. Probabilistic anonymity via coalgebraic simulations. *Theoretical Computer Science*, 411:2239–2259, 2007.

[39] D. Hausmann, T. Mossakowski, and L. Schröder. Iterative circular coinduction for cocasl in isabelle/hol. In *Proc. FASE*, pages 341–356. LNCS 3442, 2005.

[40] R. Howard. *Dynamic Probabilistic Systems*. John Wiley & Sons, Inc., New York, 1971.

[41] B. Jacobs. Exercises in coalgebraic specification. In *Algebraic and coalgebraic methods in the mathematics of program construction*, pages 237–281. LNCS 2297, 2002.

[42] B. Jacobs. Coalgebraic trace semantics for combined possibilitistic and probabilistic systems. *Electronic Notes in Theoretical Computer Science*, 203(5):131–152, 2008. Proc. CMCS'08.

[43] B. Jacobs. Probabilities, distribution monads, and convex categories. *Theoretical Computer Science*, 2011. To appear.

[44] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS*, 62:222–259, 1996.

[45] B. Jacobs and A. Sokolova. Exemplaric expressivity of modal logics. *Journal of Logic and Computation*, 20(5):1041–1068, 2010.

[46] C. Jones. *Probabilistic Non-determinism*. PhD thesis, University of Edinburgh, 1989.

[47] B. Jonsson and K. Larsen. Specification and refinement of probabilistic processes. In *Proc. LICS*, pages 266–277. IEEE, 1991.

[48] B. Jonsson, K. Larsen, and W. Yi. Probabilistic extensions of process algebras. In *Handbook of Process Algebras*, pages 685–710. Elsevier, North Holland, 2001.

[49] J. Kemeny and J. Snell. *Finite Markov Chains*. Springer-Verlag, New York, 1976.

[50] B. Klin. Structural operational semantics for weighted transition systems. In *Semantics and Algebraic Specification, Essays Dedicated to Peter D. Mosses on the Occasion of His 60th Birthday*, pages 121–139. LNCS 5700, 2009.

[51] B. Klin. Structural operational semantics coalgebraically. *Theoretical Computer Science*, 2010. This issue.

[52] B. Klin and V. Sassone. Structural operational semantics for stochastic process calculi. In *Proc. FoSSaCS*, pages 428–442. LNCS 4962, 2008.

[53] C. Kupke and D. Pattinson. Coalgebraic modal logics. *Theoretical Computer Science*, 2010. This issue.

[54] A. Kurz. *Logics for coalgebras and applications to computer science.* PhD thesis, Ludwig-Maximilians-Universität München, 2000.

[55] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[56] F. Lawvere. The category of probabilistic mappings, 1962. Preprint.

[57] L. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.

[58] L. Moss and I. Viglizzo. Final coalgebras for functors on measurable spaces. *Information and Computation*, 240:610–636, 2006.

[59] P. Panangaden. *Labelled Markov Processes.* Imperial College Press, 2009.

[60] D. Pattinson and L. Schröder. Modular algorithms for heterogeneous modal logics via multi-sorted coalgebra. *Mathematical Structures in Computer Science*, 21:235–266, 2010.

[61] A. Paz. *Introduction to probabilistic automata (Computer science and applied mathematics).* Academic Press, Inc., Orlando, FL, USA, 1971.

[62] A. Pnueli and L. Zuck. Probabilistic verification. *Information and Computation*, 103:1–29, 1993.

[63] M. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

[64] J. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[65] L. Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theoretical Computer Science*, 390(2-3):230–247, 2008.

[66] R. Segala. *Modeling and verification of randomized distributed real-time systems*. PhD thesis, MIT, 1995.

[67] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. CONCUR*, pages 481–496. LNCS 836, 1994.

[68] A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Quantitative Kleene coalgebras. *Information and Computation*, 2010. To appear.

[69] A. Sokolova. *Coalgebraic analysis of probabilistic systems*. PhD thesis, Eindhoven University of Technology, 2005.

[70] A. Sokolova and E. d. Vink. Probabilistic automata: system types, parallel composition and comparison. In *Validation of Stochastic Systems: A Guide to Current Research*, pages 1–43. LNCS 2925, 2004.

[71] A. Sokolova, E. d. Vink, and H. Woracek. Coalgebraic weak bisimulation for action-type systems. *Scientific Annals of Computer Science*, 19:93–144, 2009.

[72] M. Stoelinga. An introduction to probabilistic automata. *Bulletin of the EATCS*, 78:176–198, 2002.

[73] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS*, pages 280–291. IEEE, 1997.

[74] D. Turi and J. Rutten. On the foundations of final coalgebra semantics. *Mathematical Structures in Computer Science*, 8(5):481–540, 1998.

[75] D. Varacca. *Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation*. PhD thesis, Univ. Aarhus, 2003. BRICS Dissertation Series, DS-03-14.

[76] D. Varacca and G. Winskel. Distributing probabililty over nondeterminism. *Mathematical Structures in Computer Science*, 16(1):87–113, 2006.

[77] M. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *Proc. FOCS*, pages 327–338, Portland, Oregon, 1985. IEEE.

[78] I. Viglizzo. *Coalgebras on measurable spaces.* PhD thesis, Indiana University, 2005.

[79] I. Viglizzo. Final sequences and final coalgebras for measurable spaces. In *CALCO*, pages 395–407. LNCS 3629, 2005.

[80] E. d. Vink and J. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theoretical Computer Science*, 221:271–293, 1999.

[81] U. Wolter. On corelations, cokernels, and coequations. *Electronic Notes in Theoretical Computer Science*, 33, 2000. Proc. CMCS'00.