A hierarchy of probabilistic system types

Falk Bartels [a,1], Ana Sokolova [b,2,*] Erik de Vink [c]

[a] *CWI Amsterdam, The Netherlands*

[b] *Eindhoven University of Technology, The Netherlands*

[c] *Eindhoven University of Technology, The Netherlands*
Leiden University, The Netherlands

Abstract

We arrange various classes of probabilistic systems studied in the literature in an expressiveness hierarchy. Our expressiveness criterion is the existence of a system translation, from the less expressive type into the more expressive type, that preserves and reflects probabilistic bisimilarity. We model the different system types as coalgebras of suitable behavior functors and argue that the corresponding coalgebraic bisimilarity coincides with probabilistic bisimilarity for the classes for which the latter notion has been proposed in the literature. The theory of coalgebras provides a unified framework for the presentation of the different classes and the system translations we needed to establish the hierarchy. All these translations arise in a standard way from natural transformations between the two behavior functors involved. Such a translation generally preserves coalgebraic bisimilarity. We exploit a new result that, under mild assumptions on the behavior functors, a system translation induced by a natural transformation with injective components also reflects bisimilarity.

Key words: probabilistic transition systems, probabilistic bisimulation, coalgebra, bisimulation, cocongruence, preservation and reflection of bisimulation.

---

* Corresponding author
Email addresses: `falk.bartels@cwi.nl` (Falk Bartels), `a.sokolova@tue.nl` (Ana Sokolova), `evink@win.tue.nl` (Erik de Vink).

1 Research supported by the NWO project ProMACS
2 Research supported by the PROGRESS project ESS.5202, (A)MaPAoTS

1 Introduction

Probabilistic systems of different kinds have been studied as semantic objects since the early nineties. Some of them arise from nondeterministic systems by adding probabilistic information to all choices; sometimes both types of uncertainty are mixed. The main motivation for considering probabilities is the need for quantitative information, as opposed to qualitative information, when reasoning about non-functional aspects of systems such as throughput, resource utilization, etc. A vast amount of research has been conducted in the area of performance analysis, in which the notion of compositionality typically does not play a major role. In the area of semantics of programming languages and program verification, however, compositionality is a central theme. Various different models with different trade-offs between odds and evens regarding performance analysis and compositionality have thus been proposed in the literature (see, e.g., [10,9,3]). A notion of probabilistic bisimulation that preserves performance metrics is a key ingredient for joint reasoning about qualitative and quantitative behaviour, and also for this many proposals have been made.

In earlier work comparison is made between a number of probabilistic process equivalences (see, e.g., [6]) and categorical formulations of Larsen-Skou bisimulation and stochastic bisimulation are given [5,25]. In recent work [23] we focused on the relationship between these and various related notions and made a taxonomy of the most prominent types of probabilistic bisimulation. In the present paper we propose a purely coalgebraic perspective on this matter, which allows us to apply a novel general result for the comparison of system types. This way the uniform coalgebraic treatment helps us considerably to clarify the picture and to organize the setting.

As to the comparison of systems, we say that one class of systems is at most as expressive as another if we can map every system of the first type into one of the second such that bisimilarity is *preserved* and *reflected*. For this we require that the transformed system has the same carrier as the original and that two states are bisimilar in the original system if and only if they are bisimilar in the translated one.

The system translations we consider all arise in a straightforward way from natural transformations τ between the two coalgebra functors involved. The translations thus obtained always preserve bisimilarity. The reflection of bisimilarity is however not guaranteed. We present a sufficient condition on the coalgebra functors involved and the natural transformation τ for this. Interestingly, in our opinion, the result builds on *cocongruences* as proposed e.g. by Kurz [14]. This notion is similar to that of a bisimulation, but based on cospans instead of spans—a change of direction which comes in handy in the proof. We exploit the fact that both notions, bisimilation and cocongruence, characterize the same behavioural equivalence in case the coalgebra functor preserves weak pullbacks.

The expressiveness hierarchy we build with these tools provides a better understanding of the relationship of the various probabilistic system types. The coalgebraic approach facilitated its construction significantly. As far as we know, this form of application of the theory of coalgebras is not reported before in the literature.

The outline of the paper is as follows: Section 2 introduces some definitions and notation. Section 3 is the coalgebraic core leading from bisimulation and cocongruences to the result on reflection of bisimilarity. In section 4 we define the different classes of probabilistic systems coalgebraically. We argue that coalgebraic bisimilarity coincides with the known concrete definitions, exemplified for the particular case of simple Segala-type systems, in Section 5. Finally, in Section 6 we apply the result from Section 3 to build the expressiveness hierarchy.

2 Preliminaries

In this section we lay down the categorical notation used in the sequel. Since we mainly work with the category of sets and total functions, which we denote by Set, we explain what the categorical notions amount to in this category.

A *span* and a *cospan* between two objects X and Y are triples $\langle S, s_1, s_2\rangle$ and $\langle C, c_1, c_2\rangle$ of objects S and C and arrows as pictured below.

$$X \xleftarrow{s_1} S \xrightarrow{s_2} Y \qquad\qquad X \xrightarrow{c_1} C \xleftarrow{c_2} Y$$

By $X \times Y$, with projections $\pi_1 : X \times Y \to X$ and $\pi_2 : X \times Y \to Y$, and $X + Y$, with injections $\iota_1 : X \to X + Y$ and $\iota_2 : Y \to X + Y$, we denote the categorical *product* and *coproduct* of the two objects X and Y. This means that, for any span $\langle S, s_1, s_2\rangle$ and cospan $\langle C, c_1, c_2\rangle$ between X and Y, there exist unique functions $\langle s_1, s_2\rangle\colon S \to X \times Y$ and $[c_1, c_2]\colon X + Y \to C$ making both parts of the respective diagram below commute.

$$\begin{tikzcd} & S \arrow[dl, "s_1"'] \arrow[d, dashed, "\langle s_1, s_2\rangle"] \arrow[dr, "s_2"] & \\ X & X \times Y \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & Y \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "\iota_1"] \arrow[dr, "c_1"'] & X + Y \arrow[d, dashed, "{[c_1, c_2]}"] & Y \arrow[l, "\iota_2"'] \arrow[dl, "c_2"] \\ & C & \end{tikzcd}$$

The categorical products and coproducts in Set are simply cartesian products and disjoint unions. We say that a span $\langle S, s_1, s_2 \rangle$ between sets X and Y is *jointly injective* if $\langle s_1, s_2 \rangle : S \to X \times Y$ is injective. Dually, the cospan $\langle C, c_1, c_2 \rangle$ is *jointly surjective* if $[c_1, c_2] : X + Y \to C$ is surjective. A relation $R \subseteq X \times Y$ gives rise to the jointly injective span $\langle R, \pi_1, \pi_2 \rangle$ between X and Y.

A *pullback* of a cospan $\langle C, c_1, c_2 \rangle$ is a span $\langle P, p_1, p_2 \rangle$ as in the left diagram below satisfying $c_1 \circ p_1 = c_2 \circ p_2$ and such that for every span $\langle S, s_1, s_2 \rangle$ with $c_1 \circ s_1 = c_2 \circ s_2$ there exists a unique mediating arrow $m : S \to P$ satisfying $s_1 = p_1 \circ m$ and $s_2 = p_2 \circ m$. Dually, a *pushout* of a span $\langle S, s_1, s_2 \rangle$ is a cospan $\langle P, p_1, p_2 \rangle$ as in the right diagram below, such that for every cospan $\langle C, c_1, c_2 \rangle$ with $c_1 \circ s_1 = c_2 \circ s_2$ there exists a unique mediating arrow $m : P \to C$ satisfying $c_1 = m \circ p_1$ and $c_2 = m \circ p_2$.

$$\begin{tikzcd}
& S \arrow[d, dashed, "m"] \arrow[ddl, bend right, "s_1"'] \arrow[ddr, bend left, "s_2"] & \\
& P \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\
X \arrow[dr, "c_1"'] & & Y \arrow[dl, "c_2"] \\
& C &
\end{tikzcd}
\qquad
\begin{tikzcd}
& S \arrow[dl, "s_1"'] \arrow[dr, "s_2"] & \\
X \arrow[dr, "p_1"'] \arrow[ddr, bend right, "c_1"'] & & Y \arrow[dl, "p_2"] \arrow[ddl, bend left, "c_2"] \\
& P \arrow[d, dashed, "m"] & \\
& C &
\end{tikzcd}$$

We also need the notion of a *weak pullback*, for which the mediating arrow m need not be unique. A functor $\mathcal{F}$ is said to *preserve weak pullbacks* if it maps a weak pullback square on a weak pullback square, i.e. if $\langle P, p_1, p_2 \rangle$ is a weak pullback of the cospan $\langle C, c_1, c_2 \rangle$, then $\langle \mathcal{F}P, \mathcal{F}p_1, \mathcal{F}p_2 \rangle$ is a weak pullback of $\langle \mathcal{F}C, \mathcal{F}c_1, \mathcal{F}c_2 \rangle$.

A pullback of a cospan $\langle C, c_1, c_2 \rangle$ between sets x and Y is the span arising from the relation

$$Q := \{\langle x, y \rangle \in X \times Y \mid f(x) = g(y)\}.$$

The characterization of a pushout is a bit more complicated, and we omit it because we shall not need it. However we note that all pullbacks and pushouts exist in Set. A weak pullback based on a relation $R \subseteq X \times Y$ is also an ordinary pullback, as one can derive from the joint injectivity of the two projections. Moreover, in Set pullbacks are jointly injective and pushouts are jointly surjective.

An object 1 of a category is called *final* if for every object X there exists precisely one arrow $! : X \to 1$. In Set the final objects are the singleton sets. When we talk about an arbitrary final set, we denote its single element by a star, i.e. $1 = \{*\}$.

3 Translation of coalgebras

We are going to model probabilistic transition systems formally as coalgebras of a suitable type functor $\mathcal{B}$ on Set. In this section we will recall the necessary definitions and prove a technical result about translations of coalgebras. For a more detailed introduction into the theory of coalgebras we refer the interested reader to, e.g., the articles of Jacobs and Rutten [11,19].

Definition 1 *Let $\mathcal{B}$ be a Set-functor. A* **$\mathcal{B}$-coalgebra** *is a pair $\langle X, \alpha\rangle$ where X is a carrier set and $\alpha : X \to \mathcal{B}X$ is a transition function. A* **homomorphism** *between two $\mathcal{B}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ is a function $h : X \to Y$ satisfying $\mathcal{B}h \circ \alpha = \beta \circ h$. The $\mathcal{B}$-coalgebras together with their homomorphisms form a category, which we denote by $\mathsf{Coalg}_{\mathcal{B}}$.*

One is often interested in the states of a coalgebra, i.e. the elements of its carrier set, only up to some sort of behavioural equivalence. The most common behavioural equivalence is *bisimilarity.*

Definition 2 *A* **bisimulation** *between two $\mathcal{B}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ is a relation $R \subseteq X \times Y$ such that there exists a coalgebra structure $\gamma : R \to \mathcal{B}R$ making the projections $\pi_1 : R \to X$ and $\pi_2 : R \to Y$ coalgebra homomorphisms between the respective coalgebras, i.e. the two squares in the following diagram commute:*

$$\begin{tikzcd}
X \arrow[d, "\alpha"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\exists\gamma"'] & Y \arrow[d, "\beta"] \\
\mathcal{B}X & \mathcal{B}R \arrow[l, "\mathcal{B}\pi_1"] \arrow[r, "\mathcal{B}\pi_2"'] & \mathcal{B}Y
\end{tikzcd}$$

Occasionally we refer to γ as the mediating coalgebra structure. We say that two states $x \in X$ and $y \in Y$ are **bisimilar**, *and write $x \sim y$, if they are related by some bisimulation between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$.*

To compare the expressiveness of coalgebras for different functors, say $\mathcal{F}$ and $\mathcal{G}$, we will study translations of $\mathcal{F}$-coalgebras into $\mathcal{G}$-coalgebras. Such a translation can easily be obtained from a natural transformation between the two functors under consideration.

Definition 3 (cf. [19, Theorem 15.1]) *A natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ gives rise to a functor $\mathcal{T}_\tau : \mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$ defined for an $\mathcal{F}$-coalgebra $\langle X, \alpha\rangle$ and an $\mathcal{F}$-homomorphism h as*

$$\mathcal{T}_\tau\langle X, \alpha\rangle := \langle X, \tau_X \circ \alpha\rangle \quad \text{and} \quad \mathcal{T}_\tau h := h.$$

To see that the above definition really defines a functor, we need to check that a homomorphism h between two $\mathcal{F}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ is also a homomorphism between the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha \rangle$ and $\mathcal{T}_\tau\langle Y, \beta \rangle$. This follows easily from the naturality of τ:

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\text{assumption } h"] & Y \arrow[d, "\beta"] \\
\mathcal{F}X \arrow[r, "\mathcal{F}h"] \arrow[d, "\tau_X"'] \arrow[dr, phantom, "\text{naturality } \tau"] & \mathcal{F}Y \arrow[d, "\tau_Y"] \\
\mathcal{G}X \arrow[r, "\mathcal{G}h"'] & \mathcal{G}Y
\end{tikzcd}$$

Since $\mathcal{T}_\tau$ preserves homomorphisms, it also preserves bisimulations. This yields that if two states $x \in X$ and $y \in Y$ are bisimilar in the $\mathcal{F}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ then they are also bisimilar in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha \rangle$ and $\mathcal{T}_\tau\langle Y, \beta \rangle$.

In order to establish that $\mathcal{G}$-coalgebras are at least as expressive as $\mathcal{F}$-coalgebras, we shall use translations $\mathcal{T}_\tau$ for which the converse holds as well, i.e. where x and y are bisimilar in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha \rangle$ and $\mathcal{T}_\tau\langle Y, \beta \rangle$ only if they are bisimilar in the original $\mathcal{F}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$. In this case we say that $\mathcal{T}_\tau$ *reflects* bisimilarity.

To this end it appears reasonable to ask that the components of $\tau : \mathcal{F} \Rightarrow \mathcal{G}$ should be injective: Assume that for some set X the component τ_X is not injective, because it identifies two distinct elements $\phi, \psi \in \mathcal{F}X$, i.e. $\tau_X(\phi) = \tau_X(\psi)$. Usually it should not be difficult to find an $\mathcal{F}$-coalgebra structure α on X such that, for two states $x, y \in X$, $\alpha(x) = \phi$ and $\alpha(y) = \psi$ but $x \not\sim y$ in $\langle X, \alpha \rangle$. Since we get $\tau_X(\alpha(x)) = \tau_X(\phi) = \tau_X(\psi) = \tau_X(\alpha(y))$, we have $x \sim y$ in $\mathcal{T}_\tau\langle X, \alpha \rangle = \langle X, \tau_X \circ \alpha \rangle$, which means that $\mathcal{T}_\tau$ does not reflect bisimilarity. (Note though that the above approach does not work in the degenerate case of a functor $\mathcal{F}$ that does not allow non-bisimilar behaviour at all, like $\mathcal{F} = \mathcal{I}d$. We shall come back to this example at the end of the section.)

In the following we show that componentwise injectivity of τ implies that $\mathcal{T}_\tau$ reflects a notion of behavioural equivalence defined not in terms of bisimulations but in terms of *cocongruences*. Then we explain that this notion coincides with bisimilarity for coalgebras of functors which preserve weak pullbacks. All coalgebra functors we shall consider have this property.

Definition 4 *A* **cocongruence** *between two $\mathcal{B}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ is a cospan $\langle U, u_1, u_2 \rangle$ between X and Y, which is jointly surjective, such that there exists a $\mathcal{B}$-coalgebra structure $\gamma : U \to \mathcal{B}U$ making u_1 and u_2 coalgebra homomorphisms. This means that the two squares in the following diagram* *commute:*

$$\begin{tikzcd} X \arrow[r, "u_1"] \arrow[d, "\alpha"'] & U \arrow[d, dashed, "\exists\gamma"'] & Y \arrow[l, "u_2"'] \arrow[d, "\beta"] \\ \mathcal{B}X \arrow[r, "\mathcal{B}u_1"'] & \mathcal{B}U & \mathcal{B}Y \arrow[l, "\mathcal{B}u_2"] \end{tikzcd}$$

We say that $x \in X$ and $y \in Y$ are behavioural equivalent*, and write $x \approx y$, in the $\mathcal{B}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$, if they are identified by some cocongruence between them.*

We took the name *cocongruence* from Kurz [14, Def. 1.2.1]. Wolter [26] calls these structures *compatible corelations*.

Theorem 5 *Let $\mathcal{F}$ and $\mathcal{G}$ be two* Set *functors. For a natural transformation $\tau \colon \mathcal{F} \Rightarrow \mathcal{G}$ with injective components we have that $\mathcal{T}_\tau : \mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$ reflects behavioural equivalence.*

For the proof of the theorem we need the following elementary fact.

Lemma 6 *The category* Set *has the* diagonal fill-in *property for surjective and injective functions: Assume that the outer square in the setting depicted below commutes, where e is surjective and m is injective. Then there exists a unique diagonal arrow d making both of the resulting triangles commute.*

$$\begin{tikzcd} A \arrow[r, two heads, "e"] \arrow[d, "f"'] & B \arrow[d, "g"] \arrow[dl, dashed, "\exists! d"'] \\ C \arrow[r, hook, "m"'] & D \end{tikzcd}$$

We proceed with the proof of Theorem 5.

PROOF. Let $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ be two $\mathcal{F}$-coalgebras with states $x \in X$ and $y \in Y$ such that $x \approx y$ in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau \langle X, \alpha \rangle$ and $\mathcal{T}_\tau \langle Y, \beta \rangle$. So there exists a cocongruence $\langle U, u_1, u_2 \rangle$ between the latter coalgebras identifying x and y. We shall show below that the same cospan is also a cocongruence between the $\mathcal{F}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$, so that also for them we have $x \approx y$.

Let $\gamma\colon U \to \mathcal{G}U$ be the transition structure witnessing the cocongruence property of $\langle U, u_1, u_2\rangle$, i.e. both parts of the diagram below commute.

$$
\begin{tikzcd}
X \arrow[r, "u_1"] \arrow[d, "\alpha"'] & U \arrow[dd, "\gamma"] & Y \arrow[l, "u_2"'] \arrow[d, "\beta"] \\
\mathcal{F}X \arrow[d, hook, "\tau_X"'] & & \mathcal{F}Y \arrow[d, hook, "\tau_Y"] \\
\mathcal{G}X \arrow[r, "\mathcal{G}u_1"'] & \mathcal{G}U & \mathcal{G}Y \arrow[l, "\mathcal{G}u_2"]
\end{tikzcd}
\tag{1}
$$

Using this and the naturality of τ in step $(*)$, we compute

$$
\begin{aligned}
\gamma \circ [u_1, u_2] &= [\gamma \circ u_1, \gamma \circ u_2] \\
&\overset{(1)}{=} [\mathcal{G}u_1 \circ \tau_X \circ \alpha,\ \mathcal{G}u_2 \circ \tau_Y \circ \beta] \\
&\overset{(*)}{=} [\tau_U \circ \mathcal{F}u_1 \circ \alpha,\ \tau_U \circ \mathcal{F}u_2 \circ \beta] \\
&= \tau_U \circ [\mathcal{F}u_1 \circ \alpha,\ \mathcal{F}u_2 \circ \beta].
\end{aligned}
$$

This means that the outer square of the diagram below commutes. By the definition of a cocongruence, $[u_1, u_2]$ is surjective and, by assumption, τ_U is injective, so Lemma 6 provides a diagonal fill-in, say $\tilde{\gamma} : U \to \mathcal{F}U$.

$$
\begin{tikzcd}
X + Y \arrow[r, two heads, "{[u_1,u_2]}"] \arrow[d, "{[\mathcal{F}u_1 \circ \alpha, \mathcal{F}u_2 \circ \beta]}"'] & U \arrow[d, "\gamma"] \arrow[dl, dashed, "\tilde{\gamma}"'] \\
\mathcal{F}U \arrow[r, hook, "\tau_U"'] & \mathcal{G}U
\end{tikzcd}
$$

This shows that γ factors as $\tau_U \circ \tilde{\gamma}$, and we can refine picture (1) into the one below. It follows from the commutativity of the upper left triangle in the diagram above that the two upper squares in the diagram below indeed commute. So $\tilde{\gamma}$ witnesses that – as wanted – $\langle U, u_1, u_2\rangle$ is a cocongruence between the original $\mathcal{F}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$.

$$
\begin{tikzcd}
X \arrow[r, "u_1"] \arrow[d, "\alpha"'] & U \arrow[d, "\tilde{\gamma}"'] & Y \arrow[l, "u_2"'] \arrow[d, "\beta"] \\
\mathcal{F}X \arrow[r, "\mathcal{F}u_1"] \arrow[d, hook, "\tau_X"'] & \mathcal{F}U \arrow[d, hook, "\tau_U"'] & \mathcal{F}Y \arrow[l, "\mathcal{F}u_2"'] \arrow[d, hook, "\tau_Y"'] \\
\mathcal{G}X \arrow[r, "\mathcal{G}u_1"'] & \mathcal{G}U & \mathcal{G}Y \arrow[l, "\mathcal{G}u_2"]
\end{tikzcd}
$$

$\square$

We shall show that behavioural equivalence and bisimilarity coincide for coalgebras of a weak-pullback-preserving functor, so that the above theorem implies that $\mathcal{T}_\tau$ also reflects bisimilarity under appropriate assumptions.

We first demonstrate that we can use pullbacks and pushouts to switch between bisimulations and cocongruences. The argument is standard.

Lemma 7 *Let $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ be $\mathcal{B}$-coalgebras.*

(i) If $R \subseteq X \times Y$ is a bisimulation between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ then the pushout $\langle P, p_1, p_2\rangle$ according to the diagram below is a cocongruence between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$.

$$\begin{tikzcd}
& R \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
X \arrow[dr, "p_1"'] & & Y \arrow[dl, "p_2"] \\
& P &
\end{tikzcd}$$

(ii) If $\mathcal{B}$ preserves weak pullbacks and $\langle U, u_1, u_2\rangle$ is a cocongruence between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ then the pullback $Q = \{\langle x, y\rangle \in X \times Y \mid u_1(x) = u_2(y)\}$ is a bisimulation between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$.

$$\begin{tikzcd}
& Q \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
X \arrow[dr, "u_1"'] & & Y \arrow[dl, "u_2"] \\
& U &
\end{tikzcd}$$

PROOF. (i) Let $\gamma : R \to \mathcal{B}R$ be the coalgebra structure witnessing the bisimulation property. Applying the functor $\mathcal{B}$ to the pushout square we obtain $\mathcal{B}p_1 \circ \mathcal{B}\pi_1 = \mathcal{B}p_2 \circ \mathcal{B}\pi_2$. Together with the bisimulation property this implies that the outer hexagon in the left diagram below commutes. So, with the property of the pushout, there is a unique mediating arrow $m : P \to \mathcal{B}P$ such that $m \circ p_1 = \mathcal{B}p_1 \circ \alpha$ and $m \circ p_2 = \mathcal{B}p_2 \circ \beta$, i.e. $\langle P, p_1, p_2\rangle$ is a cocongruence between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$.

$$\begin{tikzcd}
& & R \arrow[dll, "\pi_1"'] \arrow[drr, "\pi_2"] \arrow[d, "\gamma"] & & \\
X \arrow[d, "\alpha"'] \arrow[drr, "p_1"'] & & \mathcal{B}R \arrow[dll, "\mathcal{B}\pi_1"'] \arrow[drr, "\mathcal{B}\pi_2"] & & Y \arrow[d, "\beta"] \arrow[dll, "p_2"] \\
\mathcal{B}X \arrow[drr, "\mathcal{B}p_1"'] & & P \arrow[d, dashed, "m"] & & \mathcal{B}Y \arrow[dll, "\mathcal{B}p_2"] \\
& & \mathcal{B}P & &
\end{tikzcd}
\qquad
\begin{tikzcd}
& & Q \arrow[dll, "\pi_1"'] \arrow[drr, "\pi_2"] \arrow[d, dashed, "m"] & & \\
X \arrow[d, "\alpha"'] \arrow[drr, "u_1"'] & & \mathcal{B}Q \arrow[dll, "\mathcal{B}\pi_1"'] \arrow[drr, "\mathcal{B}\pi_2"] & & Y \arrow[d, "\beta"] \arrow[dll, "u_2"] \\
\mathcal{B}X \arrow[drr, "\mathcal{B}u_1"'] & & U \arrow[d, "\gamma"] & & \mathcal{B}Y \arrow[dll, "\mathcal{B}u_2"] \\
& & \mathcal{B}U & &
\end{tikzcd}$$

(ii) Since $\mathcal{B}$ preserves weak pullbacks, $\langle \mathcal{B}Q, \mathcal{B}\pi_1, \mathcal{B}\pi_2\rangle$ is a weak pullback of $\langle \mathcal{B}U, \mathcal{B}u_1, \mathcal{B}u_2\rangle$. Using this and an argument dual to the one for item (i), we get a (not necessarily unique) mediating arrow $m : Q \to \mathcal{B}Q$ in the situation pictured in the right diagram above, which witnesses that Q is a bisimulation between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$. $\square$

In Section 2 we have not given a concrete description of pushouts in Set, because the following observation about them suffices for our comparison of bisimularity and behavioural equivalence: the pushout of a relation $R \subseteq X \times Y$ identifies all elements related by R. With this we get the following corollary.

Corollary 8 *Let $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ be two $\mathcal{B}$ coalgebras with states $x \in X$ and $y \in Y$.*

(i) If $x \sim y$ then $x \approx y$, i.e. bisimilarity implies behavioural equivalence.
(ii) If $\mathcal{B}$ preserves weak pullbacks, then $x \approx y$ also implies $x \sim y$, i.e. bisimilarity and behavioural equivalence coincide.

PROOF. If $x \sim y$ then there exists a bisimulation $R \subseteq X \times Y$ with $\langle x, y\rangle \in R$. With Lemma 7 (i) the pushout of R is a cocongruence. Since the pushout identifies all pairs related by R, we get $x \approx y$. For item (ii), let $x \approx y$. This means that there exists a cocongruence $\langle U, u_1, u_2\rangle$ identifying x and y. According to Lemma 7 (ii), the set of all pairs identified by $\langle U, u_1, u_2\rangle$ is a bisimulation, so $x \sim y$. $\square$

From Theorem 5 and Corollary 8 we easily get our result about $\mathcal{T}_\tau$ reflecting bisimilarity.

Theorem 9 *Let $\tau\colon \mathcal{F} \Rightarrow \mathcal{G}$ be a natural transformation between the Set-functors $\mathcal{F}$ and $\mathcal{G}$. If $\mathcal{F}$ preserves weak pullbacks and all components of τ are injective then the functor $\mathcal{T}_\tau$ from Definition 3 reflects bisimilarity.*

PROOF. Let $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ be $\mathcal{F}$-coalgebras with states $x \in X$ and $y \in Y$. If $x \sim y$ in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha\rangle$ and $\mathcal{T}_\tau\langle Y, \beta\rangle$ then $x \approx y$ in the same coalgebras according to Corollary 8 (i). With Theorem 5 this implies $x \approx y$ in the original $\mathcal{F}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$. Since $\mathcal{F}$ was assumed to preserve weak pullbacks, we can apply Corollary 8 (ii) to obtain $x \sim y$ in $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ as needed. $\square$

The following example demonstrates that Theorem 9 does not hold without the assumption on weak pullback preservation. It is built on a classical example [1] of a functor not preserving weak pullbacks, which is treated in detail also by Gumm and Schröder [7].

Consider the functors

$$\mathcal{F}X := \{\langle x, y, z\rangle \in X^3 \mid |\{x, y, z\}| \leq 2\} \quad \text{and} \quad \mathcal{G}X := X^3$$

and the obvious inclusion natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$, all components of which are clearly injective. The functor $\mathcal{F}$ does not preserve weak pullbacks. To see that the translation $\mathcal{T}_\tau$ does not reflect bisimilarity, consider the $\mathcal{F}$-coalgebra $\langle X, \alpha \rangle$ with

$$X := \{s, t\}, \quad \alpha(s) := \langle s, s, t \rangle, \quad \alpha(t) := \langle s, t, t \rangle.$$

The two states s and t are bisimilar in $\mathcal{T}_\tau \langle X, \alpha \rangle$ but not in $\langle X, \alpha \rangle$. For the first claim, note that $X \times X$ is a bisimulation on $\mathcal{T}_\tau \langle X, \alpha \rangle$. For the second claim, assume there was a bisimulation $R \subseteq X \times X$ on $\langle X, \alpha \rangle$ with $\langle s, t \rangle \in R$. For the mediating coalgebra structure $\gamma : R \to \mathcal{F}R$ let $\gamma(\langle s, t \rangle) = \langle z_1, z_2, z_3 \rangle$. The homomorphism condition implies

$$\langle \pi_1(z_1), \pi_1(z_2), \pi_1(z_3) \rangle = \langle s, s, t \rangle \text{ and } \langle \pi_2(z_1), \pi_2(z_2), \pi_2(z_3) \rangle = \langle s, t, t \rangle.$$

From this we conclude $\gamma(\langle s, t \rangle) = \langle \langle s, s \rangle, \langle s, t \rangle, \langle t, t \rangle \rangle$, but, since all three pairs are different, this is not an element of $\mathcal{F}R$.

The example suggests that the assumption on the coalgebra functor in Theorem 9 is not to be seen as a limitation of the result. It is rather reflecting a limitation of the standard notion of a bisimulation to express behavioural equivalence: it fails in this case to relate s and t, although they cannot be distinguished by external observations.

Coming back to an earlier remark, we mention that componentwise injectivity of the natural transformations τ in Theorem 9 is not a necessary condition for the reflection of bisimilarity. An example of a natural transformation τ with noninjective components such that $\mathcal{T}_\tau$ still reflects bisimilarity is the natural transformation $! : \mathcal{I}d \Rightarrow 1$, where $\mathcal{I}d$ is the identity functor, with the unique maps $!_X : X \to 1$ into a singleton set $1 = \{*\}$ as components. The translation $\mathcal{T}_!$ trivially reflects bisimilarity, because all states in $\mathcal{I}d$-coalgebras are bisimilar. As it were, the natural transformation forgets only information that is not relevant for bisimilarity. We can give more interesting examples of that kind, such as the natural transformation that maps probability distributions on their set of support (see Section 4). But we are not aware of any examples involving a functor $\mathcal{F}$ such that there are $\mathcal{F}$-coalgebras with non-bisimilar states.

4 Probabilistic systems

In this section we introduce thirteen types of probabilistic systems from the literature on probabilistic modelling. A considerable amount of research has been done on each of these types of systems. They are used as mathematical models of real systems so that formal verification methods based e.g. on temporal logic or process algebra can be applied. Most of the types arose independently in order to better model one or another property of a system. One motivating issue is the need to model both non-deterministic and probabilistic choice. Another issue is the compositional modelling for which operators like hiding (restrictions by the environment) and parallel composition play a major role. Therefore some more complex models were proposed that support a definition of these operators. For example, generative systems were extended to bundle probabilistic systems because the former type did not allow for a definition of a natural asynchronous parallel composition operator. In a preceding paper [23] we gave a wider overview of these models. Here, we just note that the different classes are not defined as coalgebras in the literature. Moreover, in few cases our functorial definition varies from the original one in that we abstract from certain features that are not essential, in our understanding, to the nature of the model under consideration.

In this paper we define the systems as coalgebras of suitable behavior functors $\mathcal{B}$. The functors are built using the following syntax

$$\mathcal{B} ::= A \mid \mathcal{I}d \mid \mathcal{P} \mid \mathcal{D}_\omega \mid \mathcal{B} + \mathcal{B} \mid \mathcal{B} \times \mathcal{B} \mid \mathcal{B}^A \mid \mathcal{B}\mathcal{B}$$

where A denotes a constant functor for a set A, $\mathcal{P}$ is the powerset functor, and the composition of two functors $\mathcal{F}$ and $\mathcal{G}$ is denoted by $\mathcal{F}\mathcal{G}$. By $\mathcal{D}_\omega$ we denote the probability functor, defined by

$$\mathcal{D}_\omega S = \{\mu \colon S \to [0,1] \mid \mu[S] = 1,\ spt(\mu) \text{ finite}\} \qquad \mathcal{D}_\omega f(\mu) = \mu \circ f^{-1}$$

using the notation $\mu[X] = \sum_{x \in X} \mu(x)$ for $X \subseteq S$, $spt(\mu) = \{x \in S \mid \mu(x) > 0\}$ is the support set of μ and for $\mu \in \mathcal{D}_\omega X$, $\mu \circ f^{-1}(y) = \mu[f^{-1}(\{y\})]$.

For the proof of bisimulation correspondence (Section 5), as well as for the hierarchy results (Section 6) preservation of weak pullbacks is important. We note that

(i) the functors A, $\mathcal{I}d$, $\mathcal{P}$ and $\mathcal{D}_\omega$ [3] on Set preserve weak pullbacks,

(ii) if the Set-functors $\mathcal{F}$ and $\mathcal{G}$ preserve weak pullbacks, then so do $\mathcal{F} + \mathcal{G}$, $\mathcal{F} \times \mathcal{G}$, $\mathcal{F}^A$ and $\mathcal{F}\mathcal{G}$.

It follows that all functors involved have the desired property.

Recall that $\mathsf{Coalg}_\mathcal{B}$ denotes the category of coalgebras of the functor $\mathcal{B}$. We fix a set A to serve as a set of actions throughout this section.

[3] The preservation of weak pullbacks for $\mathcal{D}_\omega$ was shown by De Vink and Rutten [25] and by Moss [17].

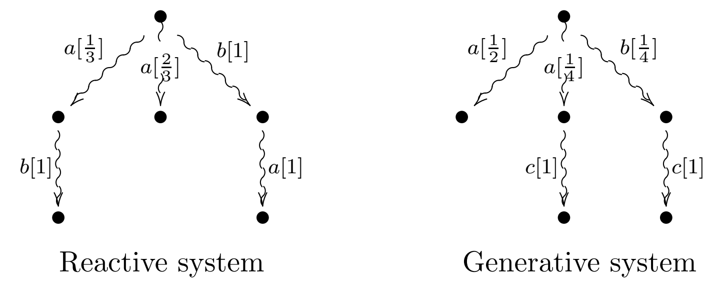
$\mathsf{Coalg}_{\mathcal{B}}$	$\mathcal{B}$	name/reference
MC	$\mathcal{D}_\omega$	Markov chains
DA	$(\mathcal{I}d+1)^A$	deterministic automata
NA	$\mathcal{P}(A\times\mathcal{I}d)\cong\mathcal{P}^A$	non-deterministic automata, LTSs
React	$(\mathcal{D}_\omega+1)^A$	reactive systems [15,6]
Gen	$\mathcal{D}_\omega(A\times\mathcal{I}d)+1$	generative systems [6]
Str	$\mathcal{D}_\omega+(A\times\mathcal{I}d)+1$	stratified systems [6]
Alt	$\mathcal{D}_\omega+\mathcal{P}(A\times\mathcal{I}d)$	alternating systems [8]
Var	$(\mathcal{D}_\omega(A\times\mathcal{I}d)+\mathcal{P}(A\times\mathcal{I}d))/\bowtie$	Vardi systems [24]
SSeg	$\mathcal{P}(A\times\mathcal{D}_\omega)$	simple Segala systems [22,21]
Seg	$\mathcal{P}\mathcal{D}_\omega(A\times\mathcal{I}d)$	Segala systems [22,21]
Bun	$\mathcal{D}_\omega\mathcal{P}(A\times\mathcal{I}d)$	bundle systems [4]
PZ	$\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A\times\mathcal{I}d)$	Pnueli-Zuck systems [18]
MG	$\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A\times\mathcal{I}d+\mathcal{I}d)$	most general systems

Fig. 1. Probabilistic system types

We now present the probabilistic system types and the functors defining them via Figure 1. For each system type the table lists the notation, the functor and the name. For some systems we also include a reference to the bibliographic source of the system. The names used for these systems follow the overview paper [23]. Some of them are otherwise not present in the literature. For the Vardi systems sometimes the term *concurrent Markov chains* is used, for the Segala systems the name *(simple) probabilistic automata* is used while the systems introduced by Pnueli and Zuck are called *probabilistic finite state programs*. We introduce the last type of systems ourselves as a generalization of the class **PZ** in order to have a top element in our hierarchy.

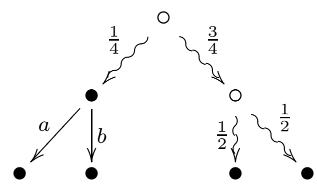
Basically, every type of probabilistic systems arises from the plain definition of a transition system with or without labels. Probabilities can then be added either to every transition, or to transitions labelled with the same action, or there can be a distinction between probabilistic and ordinary (non-deterministic) states, where only the former ones include probabilistic information, or the transition function can be equipped with structure that provides both non-determinism and probability distributions.

The simplest kind of probabilistic systems that we consider are discrete time, finitely branching Markov chains. Two other classical basic models of probabilistic systems are the reactive and the generative systems. They arise from LTSs when changing the powerset functor $\mathcal{P}$ to the distribution functor $\mathcal{D}_\omega$. At this point we can mention a distinction between systems, the one between *input* type and *output* type of systems. An input system is one defined by a functor of the kind $\mathcal{B}^A$ while an output system has a functor of the form $\mathcal{B}\mathcal{P}(A \times \mathcal{B})$. Note that LTSs can be viewed as both input and output type of systems, due to the isomorphism $\mathcal{P}(A \times \mathcal{I}d) \cong \mathcal{P}^A$. In the probabilistic case this is not the case. As the names already suggest, a reactive system is a probabilistic input system, reacting to the input by the environment, while a generative system is a typical output system, producing output depending on the probability distribution. A reactive system can transit from a given state with a given action to any other state according to the probability distribution that governs this transition. On the other hand in a generative system the distributions involve actions. The generative systems are *fully probabilistic* in the sense that it is enough to erase the action labels on the transitions in order to obtain a Markov chain from a generative system.

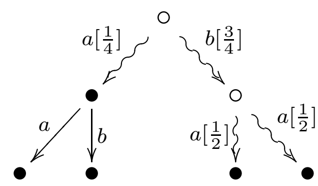


Some of the system types introduced above make a distinction between types of states. Such are the stratified, the alternating and the Vardi systems. If a state in such a system allows a probabilistic transition, then it is a probabilistic state. If, on the other hand, it allows a (non-)deterministic transition, then it is a (non-)deterministic state. The functor defining the Vardi systems needs more explanation. In a Vardi system $\langle X, \alpha\rangle$, the states can be divided into two sets, a set of non-deterministic states $x \in X$ such that $\alpha(x) \in \mathcal{P}(A \times X)$ and a set of probabilistic states $x \in X$ for which $\alpha(x) \in \mathcal{D}_\omega(A \times X)$. The probabilistic states show a generative behavior. Furthermore, by $\bowtie$ we identify some degenerated steps. If from a state $x \in X$ the system can only move, via an action a, to a state $y \in X$, then it is the same as saying that from x, via a, with probability 1 the system moves to y. Therefore, the equivalence $\bowtie$ identifies the Dirac distribution $\mu^1_{\langle a,x\rangle} \in \mathcal{D}_\omega(A \times X)$, for $\mu^1_{\langle a,x\rangle}(\langle a, x\rangle) = 1$ and the singleton set $\{\langle a, x\rangle\} \in \mathcal{P}(A \times X)$. This way, there are states in a Vardi system that are both non-deterministic, with one outgoing transition and probabilistic with a Dirac outgoing transition. By considering $(\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$ instead of $\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$, the functorial properties are still preserved.

Unlike reactive and generative ones, systems with the above distinction between states can simulate full non-determinism. When drawing diagrams of these types of systems, we use curly arrows for probabilistic transitions, and ordinary arrows for non-deterministic transitions. Furthermore, a circle represents a probabilistic state and a bullet stands for a non-deterministic state.

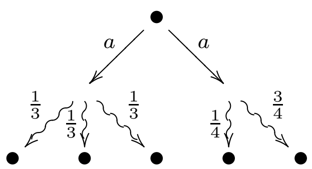


alternating system

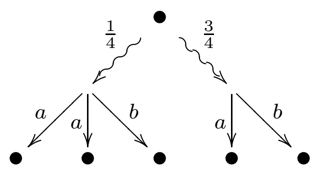


Vardi system

Another way of allowing both full non-determinism and probabilities, without distinguishing between states, is by equipping the transition function with a structure, as in the case of Segala, simple Segala, bundle and Pnueli-Zuck systems. The simple Segala model is of input type, enriching the reactive model with full non-determinism, and the other models are of output type, allowing non-determinism in the generative setting.



simple Segala system



bundle probabilistic automaton

5 Concrete vs. categorial bisimulation

For most of the probabilistic system types introduced above, a concrete definition of bisimulation is given in the literature. A cornerstone of the coalgebraic approach to bisimulation is the correspondence of bisimilarity of deterministic and non-deterministic transition systems given in concrete terms of transfer properties [16] or given in categorial terms of a mediating coalgebra [1] (see also [20]). De Vink and Rutten have shown [25], following Jones' use of the graph-theoretical max-min theorem [12], that the concrete notion of bismulation for Markov chains coincides with the coalgebraic notion. The proof technique extends to most other systems involving the functor $\mathcal{D}_\omega$ in their definition, viz. **Str**, **Alt**, **React**, **SSeg**, **Seg**, and **Gen**. As an example, in [2], we sketched the correspondence of concrete bisimulation and coalgebraic bisimulation for the general Segala-type systems (cf. [22,21]) which we modelled as coalgebras of the functor $\mathcal{PD}_\omega(A \times \mathcal{I}d)$. The bundle probabilistic transition systems [4] do not come equipped with a concrete notion of bisimulation. Equivalence of bundle probabilistic transition systems is defined in terms of the underlying generative probabilistic transitions systems, for which concrete bisimulation coincides with the coalgebraic bisimulation. The approach of Vardi [24] and Pnueli and Zuck [18] involves temporal logics. We do not unravel the explicit relationship of logically indistinguishable systems vs. bisimilar ones [15]. However, the familiarity with coalgebraic bisimulation makes it easy to formulate concrete definitions of bisimulation in the cases of bundle, Vardi and Pnueli-Zuck systems (cf. [23]).

Here we present a new and more modular proof of the correspondence of concrete probabilistic bisimulation with the coalgebraic bisimulation in the case of simple Segala systems. At the same time, it is a proof of the correspondence for reactive systems. The same technique can be used in all the other cases.

Definition 10 *Let $\langle S, \alpha\rangle$ be a simple Segala system. An equivalence relation R on S is a simple Segala bisimulation [22,21] if for all $\langle s, t\rangle \in R$ and for all actions $a \in A$:*

$$\text{if } s \xrightarrow{a}\rightsquigarrow \mu \text{ then } t \xrightarrow{a}\rightsquigarrow \mu' \text{ and } \mu \equiv_{R^*} \mu' \text{ for some distribution } \mu'$$

where $\mu \equiv_{R^*} \mu'$ if and only if $\forall C \in S/R \colon \mu[C] = \mu'[C]$.

Two states s and t of a simple Segala system $\langle S, \alpha\rangle$ are bisimilar, denoted by $s \sim_{sseg} t$ if and only if there exists a simple Segala bisimulation R on S with $\langle s, t\rangle \in R$.

The notation $s \xrightarrow{a}\rightsquigarrow \mu$ stands for $\langle a, \mu\rangle \in \alpha(s)$.

Let $\mathcal{F} = \mathcal{P}(A \times \mathcal{D}_\omega)$ be the functor defining the simple Segala systems. Let $\sim_{\mathcal{F}}$ denote the bisimilarity relation for $\mathsf{Coalg}_{\mathcal{F}} = \mathbf{SSeg}$. Let $\langle S, \alpha\rangle, \langle T, \beta\rangle \in \mathsf{Coalg}_{\mathcal{F}}$. By definition, $s \sim_{\mathcal{F}} t$ for $s \in S, t \in T$ if and only if there exists a (coalgebraic) bisimulation $R \subseteq S \times T$ with $\langle s, t\rangle \in R$.

In order to relate the concrete and coalgebraic notion of bisimulation in the case of simple Segala systems we lift a relation on sets to a relation on distributions on sets [13].

Definition 11 *Let $R \subseteq S \times T$ be a relation and let $\mu \in \mathcal{D}_\omega S$ and $\mu' \in \mathcal{D}_\omega T$ be distributions. Define $\mu \equiv_R \mu'$ if and only if there exists a distribution $\nu \in \mathcal{D}_\omega R$ such that*

$$(\mathcal{D}_\omega \pi_1)(\nu) = \mu \qquad \text{and} \qquad (\mathcal{D}_\omega \pi_2)(\nu) = \mu'.$$

The relation $\equiv_R \subseteq \mathcal{D}_\omega S \times \mathcal{D}_\omega T$ is called the lifting of R to $\mathcal{D}_\omega$.

By Definition 11 there exists a surjective map $\eta : \mathcal{D}_\omega R \to \equiv_R$ defined by

$\eta(\nu) = \langle \mathcal{D}_\omega\pi_1(\nu), \mathcal{D}_\omega\pi_2(\nu)\rangle$ such that the following diagram commutes.

$$\begin{tikzcd}
 & \equiv_R \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
\mathcal{D}_\omega S & \mathcal{D}_\omega R \arrow[l, "\mathcal{D}_\omega\pi_1"] \arrow[r, "\mathcal{D}_\omega\pi_2"'] \arrow[u, "\eta"'] & \mathcal{D}_\omega S
\end{tikzcd} \tag{2}$$

With the notion of lifting, the following characterization of coalgebraic bisimulation for $\mathcal{F}$ in terms of a relation and transfer conditions can be formulated.

Lemma 12 *A relation $R \subseteq S \times T$ is a coalgebraic bisimulation (cf. Definition 2) between the simple Segala systems $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ if and only if for all $\langle s, t\rangle \in R$, and for all $a \in A$:*

1. *if $s \xrightarrow{a}\rightsquigarrow \mu$ then there exists $\mu' \in \mathcal{D}_\omega T$ such that $t \xrightarrow{a}\rightsquigarrow \mu'$ and $\mu \equiv_R \mu'$,*
2. *if $t \xrightarrow{a}\rightsquigarrow \mu$ then there exists $\mu' \in \mathcal{D}_\omega S$ such that $s \xrightarrow{a}\rightsquigarrow \mu'$ and $\mu \equiv_R \mu'$.*

PROOF. The proof follows the same reasoning used in the proof of coincidence of coalgebraic and concrete bisimulation for labelled transition systems (cf. [20,19]). Let $\langle S, \alpha\rangle, \langle T, \beta\rangle \in \mathbf{SSeg}$ and let $R \subseteq S \times T$ be a coalgebraic bisimulation with mediating coalgebra structure γ. Assume $\langle s, t\rangle \in R$ and $s \xrightarrow{a}\rightsquigarrow \mu$. Hence $\langle a, \mu\rangle \in \alpha \circ \pi_1(\langle s, t\rangle)$ and since π_1 is a homomorphism from $\langle R, \gamma\rangle$ to $\langle S, \alpha\rangle$ we get $\langle a, \mu\rangle \in \mathcal{F}\pi_1 \circ \gamma(\langle s, t\rangle)$, i.e. there exists $\nu \in \mathcal{D}_\omega R$ such that $\langle s, t\rangle \xrightarrow{a}\rightsquigarrow \nu$ in $\langle R, \gamma\rangle$ and $\mathcal{D}_\omega\pi_1(\nu) = \mu$. Put $\mu' = \mathcal{D}_\omega\pi_2(\nu)$. Then $\mu \equiv_R \mu'$. Since π_2 is a homomorphism from $\langle R, \gamma\rangle$ to $\langle T, \beta\rangle$ we get that $\langle a, \mu'\rangle \in \beta \circ \pi_2(\langle s, t\rangle)$ i.e. $t \xrightarrow{a}\rightsquigarrow \mu'$. Clause 2. can be proven symmetrically. For the opposite direction, assume $R \subseteq S \times T$ satisfies the clauses 1. and 2. Then $\gamma : R \to \mathcal{F}R$ with

$$\gamma(\langle s, t\rangle) = \{\langle a, \nu\rangle \mid \langle a, \mu\rangle \in \alpha(s), \langle a, \mu'\rangle \in \beta(t) \text{ and } \nu \text{ witnesses that } \mu \equiv_R \mu'\}$$

is well defined. By Definition 11 it follows that π_1 and π_2 are homomorphisms from $\langle R, \gamma\rangle$ to $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$, respectively, which completes the proof. $\square$

A simple Segala bisimulation is a relation on the states of one system, while a coalgebraic bisimulation is a relation between the state sets of two systems. We will restrict to coalgebraic bisimulations on the state set of one system and show that two states are related with some coalgebraic bisimulation if and only if they are related with some simple Segala bisimulation, which gives us the correspondence of simple Segala and coalgebraic bisimilarity. Note that restricting to the state set of one system is without loss of generality. It can be shown (provided that $\mathcal{F}$ preserves weak pullbacks) that two states $s \in S$ and $t \in T$ of two $\mathcal{F}$-systems $\langle S, \alpha\rangle$ and $\langle T, \beta\rangle$ are related by a bisimulation between

S and T if and only if they are related by a bisimulation on the coproduct of the two systems, i.e., $\langle S+T, [\mathcal{F}\iota_1, \mathcal{F}\iota_2] \circ (\alpha+\beta)\rangle$.

The lifting of an equivalence relation on a set to a relation on distributions can be characterized nicely with the following statement [13].

Lemma 13 *If R is an equivalence relation, then $\equiv_R = \equiv_{R^*}$.*

An elementary proof of this property is given by Jonsson et al. [13], and a similar construction was already used by De Vink and Rutten [25]. However, we give a more abstract proof here in order to emphasize that this property follows directly from the weak pullback preservation of the functor $\mathcal{D}_\omega$.

PROOF. [Lemma 13] Let R be an equivalence relation on a set S. Then the following diagram commutes

$$\begin{tikzcd} & R \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\ S \arrow[dr, "c"'] & & S \arrow[dl, "c"] \\ & S/R & \end{tikzcd} \tag{3}$$

where c is the canonical morphism, mapping each element of S to its equivalence class under R.

In order to prove the equality of $\equiv_R$ and $\equiv_{R^*}$, we show that both relations are pullbacks of the cospan $\langle \mathcal{D}_\omega(S/R), \mathcal{D}_\omega c, \mathcal{D}_\omega c\rangle$.

For $\equiv_{R^*}$ this follows directly from the characterization of pullbacks in Set (cf. Section 2) and the fact that $\mu \equiv_{R^*} \mu'$ is equivalent to $\mathcal{D}_\omega c(\mu) = \mathcal{D}_\omega c(\mu')$, as one easily verifies.

To show that $\equiv_R$ is a pullback of the same cospan note that, in (3), $\langle R, \pi_1, \pi_2\rangle$ is a pullback of $\langle S/R, c, c\rangle$. Having that $\mathcal{D}_\omega$ preserves weak pullbacks, the following is a weak pullback diagram.

$$\begin{tikzcd} & \mathcal{D}_\omega R \arrow[dl, "\mathcal{D}_\omega \pi_1"'] \arrow[dr, "\mathcal{D}_\omega \pi_2"] & \\ \mathcal{D}_\omega S \arrow[dr, "\mathcal{D}_\omega c"'] & & \mathcal{D}_\omega S \arrow[dl, "\mathcal{D}_\omega c"] \\ & \mathcal{D}_\omega(S/R) & \end{tikzcd} \tag{4}$$

From (2) and (4) and the surjectivity of η, we get that $\langle \equiv_R, \pi_1, \pi_2\rangle$ is a weak pullback of $\langle \mathcal{D}_\omega(S/R), \mathcal{D}_\omega c, \mathcal{D}_\omega c\rangle$ as well, and since it is based on a relation,

$\langle \equiv_R, \pi_1, \pi_2 \rangle$ is a pullback of $\langle \mathcal{D}_\omega(S/R), \mathcal{D}_\omega c, \mathcal{D}_\omega c \rangle$. $\square$

Having Lemma 12 and Lemma 13, for the correspondence theorem we only need to restrict to coalgebraic bisimulations which are equivalences. This can be done because $\sim$ is an equivalence for weak-pullback-preserving functors (cf. [19, Corollary 5.6]).

Theorem 14 *Let $\langle S, \alpha \rangle \in$ **SSeg** and $s, t \in S$. Then $s \sim_{sseg} t$ if and only if $s \sim_{\mathcal{F}} t$.* $\square$

6 A hierarchy of probabilistic system types

We will exploit Theorem 9 of Section 3 to achieve the primary goal of this paper, viz. establishing a hierarchy of probabilistic system types.

Let $\mathcal{F}$ and $\mathcal{G}$ be functors on Set. If there exists a translation functor from $\mathsf{Coalg}_{\mathcal{F}}$ to $\mathsf{Coalg}_{\mathcal{G}}$ that both preserves and reflects bisimilarity then we say that the class $\mathsf{Coalg}_{\mathcal{F}}$ is *coalgebraically embedded* in the class $\mathsf{Coalg}_{\mathcal{G}}$. This relation is clearly reflexive and transitive.

The expressiveness criterion makes sure that if a class of systems **A** is coalgebraically embedded in a class **B** then a "copy" of any system belonging to **A** exists in **B**, and therefore we consider the class **B** at least as expressive as the class **A**. Another hierarchy result, using a different expressiveness criterion is given for the reactive, generative and stratified systems by Van Glabbeek et al [6]. According to the expressiveness criterion of Van Glabbeek et al the class **A** is at least as expressive as the class **B** if there exists a translation functor from **A** to **B** that preserves bisimilarity. Their expressiveness criterion is local: any system of **A** can be considered as expressing at least as much as its image in **B**, while our expressiveness criterion is global: each system in **A** expresses exactly the same as its image, but the class **B** may be "bigger".

The next theorem lists some coalgebraic embeddings between the probabilistic system types introduced in Figure 1.

Theorem 15 *The coalgebraic embeddings presented in Figure 2 hold among the probabilistic system types, where an arrow* $\mathbf{A} \to \mathbf{B}$ *expresses that the class* **A** *is coalgebraically embeddable in the class* **B**.

PROOF. By Theorem 9, if $\mathcal{F}, \mathcal{G}$ are functors on Set such that $\mathcal{F}$ preserves

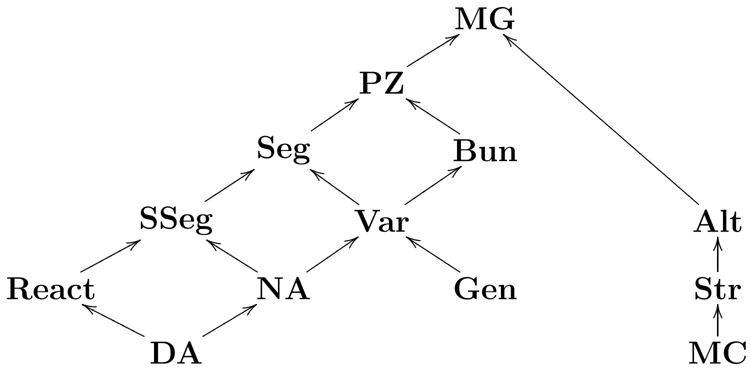


Fig. 2. Hierarchy of probabilistic system types

weak pullbacks and there is a componentwise injective natural transformation from $\mathcal{F}$ to $\mathcal{G}$, then $\mathsf{Coalg}_{\mathcal{F}}$ is coalgebraically embeddable in $\mathsf{Coalg}_{\mathcal{G}}$.

Having the weak pullback preservation for all functors from Figure 1, it is enough to construct a componentwise injective natural transformation for each embedding. We start by defining some elementary natural transformations and collecting some simple properties. Let $\mathcal{F}, \mathcal{G}, \mathcal{H}$ be functors on Set.

- We define the *empty* natural transformation $1 \overset{\eta}{\Rightarrow} \mathcal{P}$, for $\eta_X(*) = \emptyset$.
- The left and right coproduct injections ι_1 and ι_2 are natural transformations $\mathcal{F} \overset{\iota_1}{\Rightarrow} \mathcal{F} + \mathcal{G}$, $\mathcal{G} \overset{\iota_2}{\Rightarrow} \mathcal{F} + \mathcal{G}$ with injective components.
- For every set X, the injective functions $\sigma_X : X \to \mathcal{P}X$ where $\sigma_X(x) = \{x\}$ form a natural transformation $\mathcal{I}d \overset{\sigma}{\Rightarrow} \mathcal{P}$, the *singleton* natural transformation.
- For every set X, the injective functions $\delta_X : X \to \mathcal{D}_\omega X$ where $\delta_X(x) = \mu_x^1$, $\mu_x^1(x) = 1$ form the *Dirac* natural transformation $\mathcal{I}d \overset{\delta}{\Rightarrow} \mathcal{D}_\omega$.
- For any set X, the injective functions $\phi_X : (X + 1)^A \to \mathcal{P}(A \times X)$ defined by $\phi_X(f) = \mathit{Graph}(f) = \{\langle a, f(a) \rangle \mid f(a) \in X\}$ for $f : A \to X + 1$, form a natural transformation $(\mathcal{I}d + 1)^A \overset{\phi}{\Rightarrow} \mathcal{P}(A \times \mathcal{I}d)$
- From $\mathcal{F} \overset{\tau_1}{\Rightarrow} \mathcal{H}$ and $\mathcal{G} \overset{\tau_2}{\Rightarrow} \mathcal{H}$ we get a natural transformation $\mathcal{F} + \mathcal{G} \overset{[\tau_1, \tau_2]}{\Rightarrow} \mathcal{H}$.
- If $\mathcal{F}_1 \overset{\tau_1}{\Rightarrow} \mathcal{G}_1$ and $\mathcal{F}_2 \overset{\tau_2}{\Rightarrow} \mathcal{G}_2$ are componentwise injective, then so is the natural transformation $\mathcal{F}_1 + \mathcal{F}_2 \overset{\tau_1 + \tau_2}{\Rightarrow} \mathcal{G}_1 + \mathcal{G}_2$.
- If $\mathcal{F} \overset{\tau}{\Rightarrow} \mathcal{G}$ is componentwise injective, then so is $\mathcal{F}\mathcal{H} \overset{\tau\mathcal{H}}{\Rightarrow} \mathcal{G}\mathcal{H}$, where $(\tau\mathcal{H})_X = \tau_{\mathcal{H}X}$.
- From $\mathcal{F} \overset{\tau}{\Rightarrow} \mathcal{G}$ we get a natural transformation $\mathcal{H}\mathcal{F} \overset{\mathcal{H}\tau}{\Rightarrow} \mathcal{H}\mathcal{G}$ with $(\mathcal{H}\tau)_X = \mathcal{H}(\tau_X)$. If the functor $\mathcal{H}$ preserves injectivity and all components of τ are injective, then so are the components of $\mathcal{H}\tau$. For the first condition, since every Set-functor preserves injectives with nonempty domain, we just need to check that $\mathcal{H}$ maps functions from the empty set to injective functions. This is the case for $\mathcal{P}$, $\mathcal{D}_\omega$, and the other functors we use below, as one easily verifies.

Now we prove all the coalgebraic embeddings, by building the needed natural transformations from the elementary ones mentioned above.

MC → Str: $\mathcal{D}_\omega \overset{\iota_1}{\Rightarrow} \mathcal{D}_\omega + (A \times \mathcal{I}d) + 1$

DA → NA: $(\mathcal{I}d + 1)^A \overset{\phi}{\Rightarrow} \mathcal{P}(A \times \mathcal{I}d)$

DA → React: $(\mathcal{I}d + 1)^A \overset{\mathcal{F}\delta}{\Rightarrow} (\mathcal{D}_\omega + 1)^A$, for $\mathcal{F} = (\mathcal{I}d + 1)^A$.

React → SSeg: $(\mathcal{D}_\omega + 1)^A \overset{\phi\mathcal{D}_\omega}{\Rightarrow} \mathcal{P}(A \times \mathcal{D}_\omega)$

NA → SSeg: $\mathcal{P}(A \times \mathcal{I}d) \overset{\mathcal{F}\delta}{\Rightarrow} \mathcal{P}(A \times \mathcal{D}_\omega)$, for $\mathcal{F} = \mathcal{P}(A \times \mathcal{I}d)$.

NA → Var: $\mathcal{P}(A \times \mathcal{I}d) \overset{\xi \circ \iota_2}{\Rightarrow} (\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$ for $\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d) \overset{\xi}{\Rightarrow} (\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$ being the canonical natural transformation, that maps every element to its class. Although ξ is not injective, $\xi \circ \iota_2$ is.

Gen → Var: $\mathcal{D}_\omega(A \times \mathcal{I}d) + 1 \overset{\xi \circ (id + \eta\mathcal{F})}{\Rightarrow} (\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie$, for $\mathcal{F} = A \times \mathcal{I}d$. The transformation $\xi \circ (id + \eta\mathcal{F})$ is componentwise injective, since $id + \eta\mathcal{F}$ does not reach $\bowtie$-identifiable elements in $\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$.

Var → Seg: $(\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie \overset{[\sigma\mathcal{D}_\omega, \mathcal{P}\delta]\mathcal{F}}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I}d)$ for $\mathcal{F} = A \times \mathcal{I}d$. Note that the natural transformation factors through the equivalence classes, because the $\bowtie$-identified elements are mapped to the same Segala behaviour. The transformation is injective.

Var → Bun: $(\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d))/\bowtie \overset{[\mathcal{D}_\omega\sigma, \delta\mathcal{P}]\mathcal{F}}{\Rightarrow} \mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$ for $\mathcal{F} = A \times \mathcal{I}d$. As in the case **Var → Seg**, the $\bowtie$-identified elements are mapped to the same bundle behaviour, and the transformation is injective.

SSeg → Seg: $\mathcal{P}(A \times \mathcal{D}_\omega) \overset{\mathcal{P}\tau}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I}d)$ where $(A \times \mathcal{D}_\omega) \overset{\tau}{\Rightarrow} \mathcal{D}_\omega(A \times \mathcal{I}d)$ is given by $\tau_X(\langle a, \mu \rangle) = \mu_a^1 \times \mu$, where $\mu \times \mu'(\langle x, x' \rangle) = \mu(x) \cdot \mu'(x')$ and μ_a^1 is the Dirac distribution for a. All components of τ are injective.

Str → Alt: $\mathcal{D}_\omega + (A \times \mathcal{I}d) + 1 \overset{id + [\sigma, \eta]\mathcal{F}}{\Rightarrow} \mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = A \times \mathcal{I}d$. Componentwise injectivity holds.

Seg → PZ: $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I}d) \overset{\mathcal{P}\mathcal{D}_\omega\sigma\mathcal{F}}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = A \times \mathcal{I}d$.

Bun → PZ: $\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d) \overset{\sigma\mathcal{F}}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = \mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$.

PZ → MG: $\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d) \overset{\mathcal{P}\mathcal{D}_\omega\mathcal{P}\iota_1}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d + \mathcal{I}d)$

Alt → MG: $\mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}d) \overset{\sigma\mathcal{H} \circ [\mathcal{D}_\omega(\sigma\mathcal{F} \circ \iota_2), \delta\mathcal{G} \circ \mathcal{P}\iota_1]}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d + \mathcal{I}d)$. Here injections go to $A \times \mathcal{I}d + \mathcal{I}d$ and $\mathcal{F} = A \times \mathcal{I}d + \mathcal{I}d$, $\mathcal{G} = \mathcal{P}\mathcal{F}$, $\mathcal{H} = \mathcal{D}_\omega\mathcal{G} = \mathcal{D}_\omega\mathcal{P}\mathcal{F}$. Again, there is no overlap between the images in the two cases.

$\square$

We note here that we are not yet able to prove absence of arrows in the hierarchy presented. Some more arrows than those presented in Figure 2 may exist. For instance in case of a finite label set A, we get **React → Gen** by the transformation $\tau : (\mathcal{D}_\omega + 1)^A \Rightarrow \mathcal{D}_\omega(A \times \mathcal{I}d) + 1$ defined in the following way. Fix a distribution $\mu \in \mathcal{D}_\omega A$ such that $spt(\mu) = A$. For any set X and any $\phi : A \to \mathcal{D}_\omega + 1$, define $\tau_X(\phi) = *$ if and only if $\phi(a) = *$ for all $a \in A$

and otherwise, $\tau_X(\phi) = \nu \in \mathcal{D}_\omega(A \times \mathcal{I}d)$ where for $a \in A, x \in X$

$$\nu(a,x) = \begin{cases} 0 & \text{if } \phi(a)(x) = *, \\ \frac{\phi(a)(x)\cdot\mu(a)}{\mu[\{b\in A \mid \phi(b)\neq *\}]} & \text{otherwise.} \end{cases}$$

The transformation τ is natural and its components are injective.

References

[1] Peter Aczel and Nax Mendler. A final coalgebra theorem. In D.H. Pitt, D.E. Rydeheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Proc. 3rd CTCS*, volume 389 of *LNCS*, pages 357–365. Springer, 1989.

[2] F. Bartels, A. Sokolova, and E.P. de Vink. A hierarchy of probabilistic system types. In H. P. Gumm, editor, *Proc. CMCS'03*, volume 82 of *ENTCS*, 2003.

[3] M. Bernardo. *Theory and Application of Extended Markovian Process Algebra.* PhD thesis, University of Bologna, 1999.

[4] P. D'Argenio, H. Hermanns, and J.-P. Katoen. On generative parallel composition. In *Proc. Probmiv'98*, volume 22 of *ENTCS*. Elsevier, 1998.

[5] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for Labelled Markov Processes. *Information and Computation*, 179:163–193, 2002.

[6] R.J. van Glabbeek, S.A. Smolka, and B. Steffen. Reactive, generative, and stratified models of probabilistic processes. *Information and Computation*, 121:59–80, 1995.

[7] H. Peter Gumm and Tobias Schröder. Coalgebraic structure from weak pullback preserving fuctors. In Horst Reichel, editor, *Proc. CMCS 2000*, volume 33 of *ENTCS*. Elsevier, 2000.

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

[9] H. Hermanns. *Interactive Markov Chains.* PhD thesis, Universiät Erlangen-Nürnberg, 1998. Revised version appeared as Interactive Markov Chains and the Quest for Quantified Quality, LNCS 2428, 2002.

[10] J. Hillston. *A compositional approach to performance modelling.* PhD thesis, University of Edinburgh, 1994. Also appeared in the CPHC/BCS Distinguished Dissertation Series, Cambridge University Press, 1996.

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

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

[13] B. Jonsson, K.G. Larsen, and W. Yi. Probabilistic extensions of process algebras. In *Handbook of Process Algebras*. Elsevier, North Holland, 2001.

[14] A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.

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

[16] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

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

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

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

[20] J.J.M.M. Rutten and D. Turi. Initial algebra and final coalgebra semantics for concurrency. In *REX School/Symposium*, pages 530–582. LNCS 666, 1993.

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

[22] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. Concur'94*, pages 481–496. LNCS 836, 1994.

[23] A. Sokolova and E.P. de Vink. Probabilistic automata: system types, parallel composition and comparison. In *Dagstuhl seminar on Validation of Stochastic Systems, December 2002*. LNCS, 2003. In press. Available via: `http://www.win.tue.nl/~ana/`.

[24] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. FOCS'85*, pages 327–338, Portland, Oregon, 1985.

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

[26] U. Wolter. On corelations, cokernels, and coequations. In H. Reichel, editor, *Proc. CMCS 2000*. ENTCS 33, 2000. 20pp.