A hierarchy of probabilistic system types

Falk Bartels [1]

CWI, Amsterdam
falk.bartels@cwi.nl

Ana Sokolova [2]

Department of Mathematics and Computer Science, TU/e, Eindhoven
a.sokolova@tue.nl

Erik de Vink

Department of Mathematics and Computer Science, TU/e, Eindhoven
LIACS, Leiden University
evink@win.tue.nl

Abstract

We study various notions of probabilistic bisimulation from a coalgebraic point of view, accumulating in a hierarchy of probabilistic system types. In general, a natural transformation between two Set-functors straightforwardly gives rise to a transformation of coalgebras for the respective functors. This latter transformation preserves homomorphisms and thus bisimulations. For comparison of probabilistic system types we also need reflection of bisimulation. We build the hierarchy of probabilistic systems by exploiting the new result that the transformation also reflects bisimulation in case the natural transformation is componentwise injective and the first functor preserves weak pullbacks. Additionally, we illustrate the correspondence of concrete and coalgebraic bisimulation in the case of general Segala-type systems.

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

[1] Research supported by the NWO project ProMACS
[2] Research supported by the PROGRESS project aMPaTS

1 Introduction

Probabilistic systems have been studied as semantic objects since the early nineties. Sometimes nondeterminism is replaced by probalility; sometimes both types of uncertainty are mixed. The main motivation for considering probabilities is based on 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, but typically the notion of compositionality does not play a major role there. In the area of semantics of programming languages and program verification, compositionality is a central theme. Various different models, with subsequently different trade-offs between odds and evens regarding performance analysis and compositionality, have been proposed in the literature that try to find a good marriage of performance analysis techniques and compositional reasoning. See, e.g., [Hil94,Her98,Ber99]. A notion of probabilistic bisimulation that preserves performance metrics is a key ingredient for this relationship to be a long and lasting one, and also for this many proposals have been made. In our view, the uniform coalgebraic treatment helps to clarify the picture and to organize the setting.

In earlier work comparison is made between a number of probabilistic process equivalences (see, e.g., [GSS95]). Categorial formulations of Larsen-Skou bisimulation and stochastic bisimulation are advocated in [DEP02,VR99]. In recent work we focused on the relationships between these and various related notions and made a taxonomy of the most prominent notions of probabilistic bisimulation. The coalgebraic framework proved useful already in [BSV02] for a unified presentation of the diverse types of systems. In the present paper we propose a purely coalgebraic perspective on this matter and provide a general result for the comparison of system types.

The general situation is as follows: Let $\mathcal{F}$, $\mathcal{G}$ be two Set-functors, and $\tau\colon \mathcal{F} \Rightarrow \mathcal{G}$ a natural transformation from $\mathcal{F}$ to $\mathcal{G}$. The component $\tau_X\colon \mathcal{F}X \to \mathcal{G}X$ transforms each $\mathcal{F}$-coalgebra $\langle X, \alpha \rangle$ into a $\mathcal{G}$-coalgebra $\langle X, \tau_X \circ \alpha \rangle$. Hence, the natural transformation τ gives rise to a functor $\mathcal{T}_\tau\colon \mathsf{Coalg}_\mathcal{F} \to \mathsf{Coalg}_\mathcal{G}$. As a consequence, an $\mathcal{F}$-bisimulation is transformed via $\mathcal{T}_\tau$ into a $\mathcal{G}$-bisimulation. So, $\mathcal{T}_\tau$ preserves bisimulation. Now suppose the $\mathcal{G}$-coalgebras $\langle X, \tau_X \circ \alpha \rangle$ and $\langle Y, \tau_Y \circ \beta \rangle$ have bisimilar states x_0 and y_0. If it is always the case that the corresponding $\mathcal{F}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ have x_0 and y_0 bisimilar as well then we say that $\mathcal{T}_\tau$ reflects bisimulation. Now, the class of systems of type $\mathcal{F}$ can be considered as a subclass of the class of systems of type $\mathcal{G}$ if there exists a functorial embedding from $\mathsf{Coalg}_\mathcal{F}$ to $\mathsf{Coalg}_\mathcal{G}$ which both preserves and reflects bisimulation.

The techical result of this paper provides sufficient conditions in terms of the functor $\mathcal{F}$ and the natural transformation τ for this being the case. Moreover, the coalgebraic analysis of the hierarchy provides a better understanding of the relationships between the various systems types. As far as we know this form of application of the theory of coalgbras is not reported before in the literature. Interestingly, in our opinion, the result itself builds on *cocongruences* as proposed e.g. by Kurz [Kur00]. 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 our proofs. We exploit the fact that both notions characterize the same behavioural equivalence in case the coalgebra functor preserves weak pullbacks.

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 bisimulation. In section 4 the result is exploited in the setting of probabilistic transition systems. For the particular case of general Segala-type systems it is shown that concrete and coalgebraic bisimulation coincide.

Acknowledgements

We would like to thank our colleagues and the CMCS referees for useful comments. Special thanks go to Holger Hermanns and Clemens Kupke for fruitful discussions about probabilistic systems and coalgebraic issues respectively.

2 Preliminaries

In the sequel we will use the following notational conventions: (i) products $X \times Y$, pairings $\langle f, g \rangle : Z \to X \times Y$ for functions $f: Z \to X$ and $g: Z \to Y$, (ii) coproducts $X + Y$, case analysis $[f, g]: X + Y \to Z$ for functions $f: X \to Z$ and $g: Y \to Z$, (iii) function images $f(X') = \{ f(x) \in Y \mid x \in X' \}$ for $f: X \to Y$ and $X' \subseteq X$.

For any set X a probability distribution μ for X is a mapping $\mu: X \to [0,1]$ such that (i) $\{ x \mid \mu(x) \neq 0 \}$ is finite or countably infinite, (ii) $\sum \{ \mu(x) \mid x \in X \} = 1$. The set $spt(\mu) = \{ x \mid \mu(x) \neq 0 \}$ is called the support of μ. For a subset $U \subseteq X$ we use $\mu[U]$ to denote the number $\sum \{ \mu(x) \mid x \in U \}$. The collection of all probability distributions for X is denoted by $\mathcal{D}_\omega(X)$.

Let $\mu: X \to [0,1]$ be a probability distribution and $f: X \to Y$ a mapping. The map $\mu \circ f^{-1}: Y \to [0,1]$ is given by $(\mu \circ f^{-1})(y) = \mu[f^{-1}(\{y\})]$. It follows that $\mathcal{D}_\omega$ can be considered as a Set-functor mapping $f: X \to Y$ to $\mathcal{D}_\omega(f): \mathcal{D}_\omega(X) \to \mathcal{D}_\omega(Y)$ given by $\mathcal{D}_\omega(f)(\mu) = \mu \circ f^{-1}$. In [Mos99,VR99] it is shown that $\mathcal{D}_\omega$ preserves weak pullbacks.

Let $G = (V, E)$ be a directed graph with two distinguished vertices *src* and *snk* with only outgoing and only incoming edges, respectively, and $c: E \to [0,1]$ a capacity function. The graph G is referred to as a network. A flow f for the network G is a function $f: E \to [0,1]$ such that (i) for all vertices v different from src, snk it holds that $\sum \{ f(u,v) \mid (u,v) \in E \} = \sum \{ f(v,u) \mid (v,u) \in E \}$, (ii) $f(e) \leq c(e)$ for all $e \in E$. The value of the flow f is given by $\sum\{ f(src, v) \mid (src, v) \in E \}$. A cut C for the network G is a subset $C \subseteq V$ such that $src \in C$, $snk \notin C$. The value of the cut is given by $\sum\{ c(u, v) \mid u \in C, v \notin C, (u, v) \in E \}$. The following is the well-known graph-theoretical Max-flow Min-cut Theorem.

Theorem 2.1 *Any network has a maximal flow and a minimal cut. Moreover, their values coincide.*

3 Transformation of coalgebras

In this article we model probabilistic transition systems formally as coalgebras of a suitable type functor $\mathcal{B}$ on Set, the category of sets and total functions. For a more detailed introduction into the theory of coalgebras we refer the interested reader to, e.g., the articles of Jacobs and Rutten [JR96,Rut00].

Definition 3.1 Let $\mathcal{B}$ be a Set-functor. A **$\mathcal{B}$-coalgebra** is a pair $\langle X, \alpha \rangle$ where X is a 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 the carrier set X, only up to some sort of behavioural equivalence. This is most commonly defined through bisimilarity. We adopt a categorical definition based on the notion of a *span*.

Definition 3.2(i) A **span** between two sets X and Y is a triple $\langle R, r_1, r_2 \rangle$ consisting of a set R and two functions $r_1 : R \to X$ and $r_2 : R \to Y$. We say that the pair $\langle x, y \rangle \in X \times Y$ is related by this span, notation xRy, in case there exists an element $z \in R$ with $x = r_1(z)$ and $y = r_2(z)$ (or equivalently, $\langle x, y \rangle \in \langle r_1, r_2 \rangle(R) \subseteq X \times Y$).

(ii) Let $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ be two $\mathcal{B}$-coalgebras. A **bisimulation** between $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$ is a span $\langle R, r_1, r_2 \rangle$ between the carriers X and Y, such that there exists a coalgebra structure $\gamma : R \to \mathcal{B}R$ making r_1 and r_2 coalgebra homomorphisms between the respective coalgebras, i.e. making the two squares in the following diagram commute:

$$\begin{tikzcd} X \arrow[d, "\alpha"'] & R \arrow[l, "r_1"'] \arrow[r, "r_2"] \arrow[d, dashed, "\exists\gamma"'] & Y \arrow[d, "\beta"] \\ \mathcal{B}X & \mathcal{B}R \arrow[l, "\mathcal{B}r_1"] \arrow[r, "\mathcal{B}r_2"'] & \mathcal{B}Y \end{tikzcd}$$

Occasionally we refer to $\langle R, \gamma \rangle$ as a mediating coalgebra. 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 $\langle Y, \beta \rangle$.

We now turn to transformations between coalgebras of different type functors $\mathcal{F}$ and $\mathcal{G}$. Such a transformation can easily be obtained from a natural transformation between the two functors under consideration.

Definition 3.3 [cf. [Rut00, 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 and y in the $\mathcal{F}$-coalgebras $\langle X, \alpha \rangle$ and $\langle Y, \beta \rangle$, respectively, are bisimilar, then they are also bisimilar in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau \langle X, \alpha \rangle$ and $\mathcal{T}_\tau \langle Y, \beta \rangle$.

Moreover, in order to argue that $\mathcal{G}$-coalgebras are at least as expressive as $\mathcal{F}$-coalgebras, we are interested in transformations $\mathcal{T}_\tau$ 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 $\langle Y, \beta \rangle$ already. In this case we say that $\mathcal{T}_\tau$ *reflects* bisimilarity.

To this end it appears reasonable to ask that the components of τ should be injective: Given $\phi, \psi \in \mathcal{F}S$ with $\phi \neq \psi$ we are usually able to find a coalgebra structure $\alpha : S \to \mathcal{F}S$ with $\alpha(s) = \phi$ and $\alpha(t) = \psi$ for some $s, t \in S$ such that $s \not\sim t$ (an exception being the degenerate case of a functor that does not allow nonbisimilar behaviour at all, like $\mathcal{F} = \mathcal{I}d$). So $\mathcal{T}_\tau$ would not reflect bisimilarity if τ_S identifies ϕ and ψ, since this would make s and t bisimilar in $\mathcal{T}_\tau \langle S, \alpha \rangle$.

In the following we show that componentwise injectivity of τ is already sufficient for $\mathcal{T}_\tau$ to reflect bisimilarity, at least in case $\mathcal{F}$ preserves weak pullbacks. This latter condition comes in because it allows us to resort to an alternative definition of bisimilarity which turns out to be better suited for our purposes. It is based on the notion of a *cospan*.

Definition 3.4(i) A **cospan** between the sets X and Y is a triple $\langle U, u_1, u_2 \rangle$ consisting of a set U and two functions $u_1 : X \to U$ and $u_2 : Y \to U$. The pair $\langle x, y \rangle \in X \times Y$ is identified by $\langle U, u_1, u_2 \rangle$ in case $u_1(x) = u_2(y)$.

(ii) 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 such that there exists a $\mathcal{B}$-coalgebra structure $\gamma : U \to \mathcal{B}U$ making u_1 and u_2 coalgebra homomorphisms, which 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 took the name *cocongruence* from a similar notion used by Kurz [Kur00, Def. 1.2.1]. One also finds the term *compatible corelation* in this context [Wol00].

We can use pullbacks and pushouts to switch between spans and cospans and – under further assumptions – also between bisimulations and cocongruences, as the following simple and known observations state.

Lemma 3.5*(i) If the pair $\langle x, y\rangle \in X \times Y$ is related by a span $\langle R, r_1, r_2\rangle$ between X and Y, then both elements are identified by its pushout $\langle P, p_1, p_2\rangle$.*

$$\begin{tikzcd} & R \arrow[dl, "r_1"'] \arrow[dr, "r_2"] & \\ X \arrow[dr, dashed, "p_1"'] & & Y \arrow[dl, dashed, "p_2"] \\ & P \arrow[u, phantom, "\wedge"] & \end{tikzcd} \qquad \begin{tikzcd} & Q \arrow[dl, dashed, "q_1"'] \arrow[dr, dashed, "q_2"] \arrow[d, phantom, "\vee"] & \\ X \arrow[dr, "u_1"'] & {} & Y \arrow[dl, "u_2"] \\ & U & \end{tikzcd}$$

(ii) If $x \in X$ and $y \in Y$ are identified by a cospan $\langle U, u_1, u_2\rangle$ between X and Y, then x and y are related by its pullback $\langle Q, q_1, q_2\rangle$.

Lemma 3.6 *Let $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ be $\mathcal{B}$-coalgebras.*

(i) If $\langle R, r_1, r_2\rangle$ is a bisimulation between $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ then its pushout is a cocongruence between the same coalgebras.

(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 its pullback is a bisimulation.

In the proof of our result on reflection of bisimularity we furthermore use the following well-known fact about the category Set.

Lemma 3.7 *The category Set has the following 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.*

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

The crucial property we need for our statement is isolated in this lemma.

Lemma 3.8 *Let $\tau \colon \mathcal{F} \Rightarrow \mathcal{G}$ be a natural transformation all components of which are injective. When $\langle U, u_1, u_2\rangle$ is a cocongruence between the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha\rangle$ and $\mathcal{T}_\tau\langle Y, \beta\rangle$ such that u_1 and u_2 are jointly surjective (i.e. $[u_1, u_2] \colon X+$*

$Y \to U$ is surjective) then it is also a cocongruence between the $\mathcal{F}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$.

Proof. Let $\gamma\colon U \to \mathcal{G}U$ be the transition structure witnessing the cocongruence property of $\langle U, u_1, u_2\rangle$.

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

Using (a) the commutativity of the two squares above and (b) the naturality of τ, we get

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

This means that the outer square of the diagram below commutes. By assumption, $[u_1, u_2]$ is surjective and τ_U is injective, so Lemma 3.7 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 our initial picture 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 $\langle Y, \beta\rangle$.

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

□

From this we easily get the result on reflection of bisimulation.

Theorem 3.9 *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 3.3 reflects bisimilarity.*

Proof. Let $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$ be $\mathcal{F}$-coalgebras and let $x \in X$ and $y \in Y$ be bisimilar in the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha\rangle$ and $\mathcal{T}_\tau\langle Y, \beta\rangle$. This means that there is a bisimulation $\langle R, r_1, r_2\rangle$ between them relating x and y. Let $\langle Q, q_1, q_2\rangle$ be the pushout of $\langle R, r_1, r_2\rangle$. By item (i) of Lemma 3.6 $\langle Q, q_1, q_2\rangle$ is a cocongruence between the $\mathcal{G}$-coalgebras $\mathcal{T}_\tau\langle X, \alpha\rangle$ and $\mathcal{T}_\tau\langle Y, \beta\rangle$ and by item (i) of Lemma 3.5 it identifies x and y. Since the two legs of a pushout are always jointly surjective, we can apply Lemma 3.8 to find that $\langle Q, q_1, q_2\rangle$ is also a cocongruence between the original $\mathcal{F}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$. Let $\langle P, p_1, p_2\rangle$ be the pullback of $\langle Q, q_1, q_2\rangle$. We assumed $\mathcal{F}$ to preserve weak pullbacks, so we can apply part (ii) of Lemma 3.6 to get that $\langle P, p_1, p_2\rangle$ is a bisimulation between the $\mathcal{F}$-coalgebras $\langle X, \alpha\rangle$ and $\langle Y, \beta\rangle$. By item (ii) of Lemma 3.5 this span relates x and y, which means that the states are bisimilar in the original $\mathcal{F}$-coalgebras as was to be shown. $\square$

Our argument shows that componentwise injectivity of the natural transformation τ guarantees that the translation $\mathcal{T}_\tau$ of coalgebras reflects a notion of behavioural equivalence defined in terms of cocongruences. This implies reflection of bisimilarity for the important classes of coalgebras for which the two notions coincide, as it is the case when the coalgebra functor preserves weak pullbacks. The following counter-example demonstrates that such an additional assumption is indeed necessary. It is built on a classical example of a functor not preserving weak pullbacks, which is treated in detail for instance by Gumm and Schröder [GS00].

It involves the functor $\mathcal{F}X := \{\langle x, y, z\rangle \in X^3 \mid |\{x, y, z\}| \leq 2\}$, which does not preserve weak pullbacks, the functor $\mathcal{G}X := X^3$, and the obvious inclusion natural transformation $\tau : \mathcal{F} \Rightarrow \mathcal{G}$, all components of which are clearly injective. Consider the $\mathcal{F}$-coalgebra $\langle X, \alpha\rangle$ with $X := \{s, t\}$, $\alpha(s) := \langle s, s, t\rangle$, and $\alpha(t) := \langle s, t, t\rangle$. One easily checks that s and t are not bisimilar in $\langle X, \alpha\rangle$, but they are bisimilar in $\mathcal{T}_\tau\langle X, \alpha\rangle$. To see the former, assume there was a bisimulation $\langle R, r_1, r_2\rangle$ and $z \in R$ such that $r_1(z) = s$ and $r_2(z) = t$ and let the mediating coalgebra structure $\gamma : R \to \mathcal{F}R$ map z to the triple $\langle z_1, z_2, z_3\rangle$. The homomorphism condition implies $\langle r_1(z_1), r_1(z_2), r_1(z_3)\rangle = \langle s, s, t\rangle$ and $\langle r_2(z_1), r_2(z_2), r_2(z_3)\rangle = \langle s, t, t\rangle$. From this we conclude that all z_i are different, which is a contradiction because $\langle z_1, z_2, z_3\rangle$ was assumed to be in $\mathcal{F}R$.

The counter-example suggests that the assumption on the coalgebra functor in Theorem 3.9 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.

4 Probabilistic system types

We will exploit Theorem 3.9 of the previous section to achieve the primary goal of this paper, viz. establishing a hierarchy of probabilistic system types. We first introduce a number of system types from the literature on probabilistic modelling, and subsequently prove various embedding properties.

Probabilistic systems

We introduce all systems under consideration as coalgebras of a suitable functor $\mathcal{B}$. The functors are built using the following syntax

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

where $\mathcal{C}$ denotes a constant functor on Set, $\mathcal{P}$ is the powerset functor, and the composition of two functors $\mathcal{F}$ and $\mathcal{G}$ is denoted by $\mathcal{F}\mathcal{G}$. 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.

A considerable amount of research has been done on each of these thirteen types of systems we are going to consider. They are used for building models of real systems that can further be used for verification purposes either via temporal logic or verification proofs using formalisms such as process algebra. Most of the models arose independently in the literature in order to model better one or another property of a system. One motivating issue for the development of these models was a need to model both non-deterministic and probabilistic choice. Another issue from the users point of view is the need for 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 definition of these operators. For example, the bundle model arose from the generative one because of the need of defining a natural asynchronous parallel composition operator, which was not possible in the generative model itself. In [BSV02] we give a wider overview of these models. Here, we just note that the definitions in the literature are not given in terms of coalgebras of a given functor. 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. To our knowledge this is the first time that all these system types are placed and compared in one framework.

We now proceed towards the definitions of all the system types. We introduce the types of systems from the simplest ones, that even do not include probabilities, towards more complicated ones.

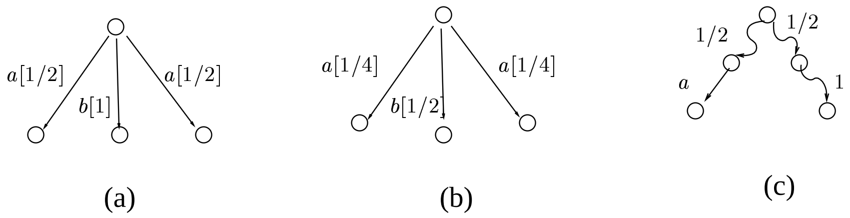
A *deterministic automaton* is a $\mathcal{B}$-coalgebra for $\mathcal{B} = (\mathcal{I}d + 1)^A$. We use $\mathbf{DA}$ for $\mathsf{Coalg}_{\mathcal{B}}$ in this case. Hence for $\langle X, \alpha \rangle \in \mathbf{DA}$, $\alpha(x)$ can be considered a partial function from A to X. A *non-deterministic automaton* is a coalgebra of the functor $\mathcal{P}(A \times \mathcal{I}d)$, the category of these coalgebras is denoted by $\mathbf{NA}$. The simplest kind of probabilistic systems that we consider are discrete time, finitely branching Markov chains. A *Markov chain* is interpreted as a coalgebra of the functor $\mathcal{D}_\omega$ and the category of such coalgebras we denote by $\mathbf{MC}$.

Next we define the reactive, generative and stratified probabilistic systems as introduced in [GSS95]. These type of systems can be considered as basic types of probabilistic transition systems. 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. There is no probability added to the choice between different actions, hence this type of systems allows first a non-deterministic choice between the actions and then a probabilistic choice governed by a given distribution. The functor that defines this class of systems is $(\mathcal{D}_\omega + 1)^A$ and the category of all such systems is denoted by **React**.

The functor defining the class of *generative probabilistic systems*, **Gen**, is $\mathcal{D}_\omega(A \times \mathcal{I}d) + 1$. We can view a generative system as obtained from a non-deterministic automaton by adding probabilities to already existing transitions such that the sum of the outgoing transition probabilities (if any) is 1 for every state. 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.

At this point we can mention a distinction between probabilistic 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 $\mathcal{BP}(A \times \mathcal{B})$. 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 *stratified system* is defined by the functor $\mathcal{D}_\omega + (A \times \mathcal{I}d) + 1$. The class of all such systems is denoted by **Str**. In a stratified system either a *purely probabilistic* transition is enabled from a state to any other state, or a single action transition is enabled, or no transition at all (deadlock state).

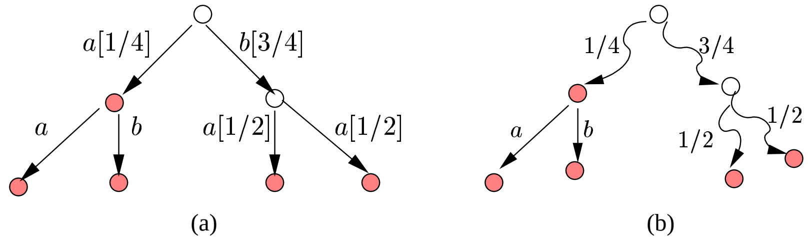


Reactive (a), generative (b) and stratified (c) system

One of the earliest models of probabilistic systems is due to Vardi [Var85]. We denote the class of *Vardi probabilistic systems* by **Var**. It is defined by the functor $\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$. The states in a Vardi system $\langle X, \alpha \rangle$ can be divided into two disjoint 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 $\alpha(x) \in \mathcal{D}_\omega(A \times X)$.

Another type of probabilistic systems that makes a distinction between non-deterministic and probabilistic states are the *alternating probabilistic systems* introduced by Hansson [Han94]. They are defined by the functor $\mathcal{B} = \mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}d)$. So, in the alternating model each state can either do a purely probabilistic or a non-deterministic transition. In this case we denote $\mathsf{Coalg}_\mathcal{B}$

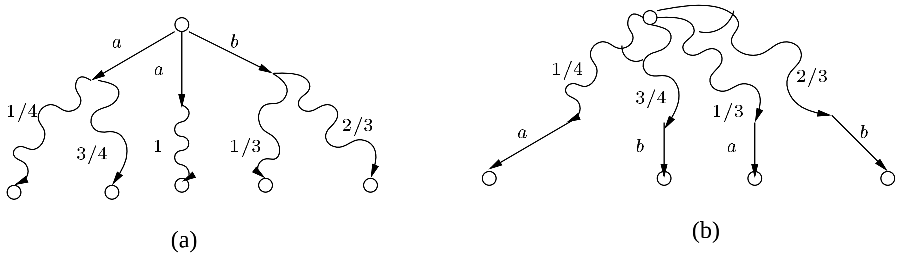
by **Alt**.



Vardi probabilistic system (a), alternating system (b)

The more complex systems that follow do not include a distinction between non-deterministic and probabilistic states, instead both non-deterministic and probabilistic choices are enabled due to the structure of the transition function. Such systems are the simple and the general Segala systems [SL94,Seg95] and the bundle [DHK98] and Pnueli-Zuck [PZ93] systems. The simple Segala model is of input type and the other models are of output type.

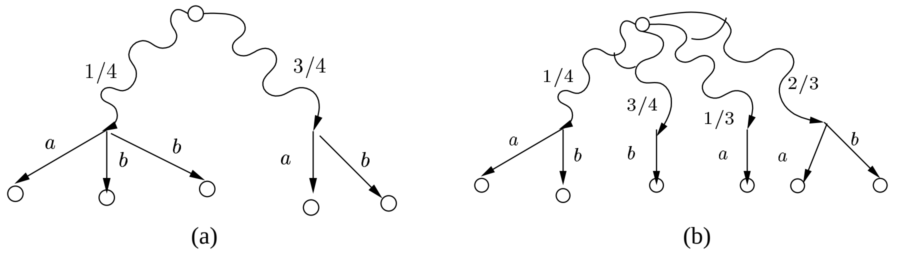
A *general Segala system* is defined by the functor $\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I}d)$, and the class of all such systems we denote by **Seg**. A Segala system $\langle X, \alpha \rangle$ is simple if for any state $x \in X$ and all $\mu \in \alpha(x)$ there exists an action $a \in A$ such that $spt(\mu) \subseteq \{a\} \times X$. This allows for a change in the functor defining the transition structure for simple systems. A *simple Segala system* is defined by the functor $\mathcal{P}(A \times \mathcal{D}_\omega)$, and the class of all such systems is denoted by **SSeg**. The two Segala types of systems are important for bridging the gap between input and output systems.



Simple Segala system (a), general Segala system (b)

The *bundle probabilistic systems*, introduced in [DHK98], are orthogonal to the general Segala systems. They are defined by the functor $\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d) + 1$. In this type of systems there is a probabilistic choice over non-deterministic bundles. Allowing also non-deterministic choice between distributions we get to the *Pnueli-Zuck probabilistic systems* of [PZ93] defined by the functor $\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$. We denote by **Bun** and **PZ** the categories of bundle and

Pnueli-Zuck systems, respectively.



Bundle system (a), Pnueli−Zuck system (b)

The class of Pnueli-Zuck systems is the most general one appearing in the literature. Finally we introduce one even more general type of systems that can act as a top element in the hierarchy of probabilistic system types. The class of *most general probabilistic systems* is defined by the functor $\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A\times\mathcal{I}d+\mathcal{I}d)$ and we denote it by **MG**.

Concrete vs. categorial bisimulation

For most of the probabilistic system types introduced above there exists in the literature a concrete definition of bisimulation. 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 or given in categorial terms of a mediating coalgebra [RT93]. In [VR99] it is shown that the concrete notion of bismulation for Markov-chains coincides with the coalgebraic notion. The proof technique extends to most other contexts involving the $\mathcal{D}_\omega$-functor, viz. **Str**, **Alt**, **React**, **SSeg**, **Seg**, and **Gen** as well. The bundle probabilistic transition systems of [DHK98] do not come equipped with a concrete notion of bisimulation. Equivalence of bundle probabilistic transition systems is defined in term of the underlying generative probabilistic transitions systems, for which concrete bisimulation coincides with the generative bisimulation. The approach of [Var85] and [PZ93] involves temporal logics. We did not unravel the explicit relationship of logically indistinguishable systems vs. bisimilar ones [LS91].

As an example we sketch the correspondence of concrete bisimulation and coalgebraic bisimulation for general Segala-type systems given by the functor $\mathcal{P}\mathcal{D}_\omega(A\times\mathcal{I}d)$ (cf. [SL94,Seg95]).

As a preparatory definition we say that, a relation $R\subseteq X\times Y$ is called z-closed if $R(x_1,y_1)\wedge R(x_2,y_1)\wedge R(x_2,y_2)\Rightarrow R(x_1,y_2)$. A component C of R is an irreducible non-empty subset of R such that for any fixed $x_0\in X$ the set $\{\,\langle x_0,y\rangle \mid y\in Y\colon R(x_0,y)\,\}$ is either disjoint from or contained in C and likewise for any fixed $y_0\in Y$. (The irreducibility refers to the property that a component has no proper subcomponent. See [VR99] for more detail.)

Definition 4.1 Let $\langle X,\alpha\rangle$, $\langle Y,\beta\rangle$ be two general Segala probabilistic transition systems. Two states $x_0\in X$, $y_0\in Y$ are called Segala-bisimilar if there exists a relation $R \subseteq X \times Y$ with $R(x_0, y_0)$ such that if $R(x, y)$ then

$$\forall \mu \in \alpha(x) \exists \nu \in \beta(y) \colon R(\mu, \nu) \land \forall \nu \in \beta(y) \exists \mu \in \alpha(x) \colon R(\mu, \nu)$$

where $R(\mu, \nu)$ iff $\mu[\{\langle a, x' \rangle \mid x' \in \pi_1[C]\}] = \nu[\{\langle a, y' \rangle \mid y' \in \pi_2[C]\}]$ for all actions a and components C of R.

It is immediate that if x_0 and y_0 are Segala-bisimilar via a relation R , then x_0 and y_0 are Segala-bisimilar via a z-closed relation $\tilde{R}$.

We have the following result.

Theorem 4.2 *Two states x_0, y_0 of two general Segala-systems $\langle X, \alpha \rangle$, $\langle Y, \beta \rangle$ are bisimilar in the sense of Definition 3.2 iff they are bisimilar in the sense of Definition 4.1.*

Proof. First, suppose $\langle R, \gamma \rangle$ is a mediating coalgebra with $x_0 R y_0$. Let $\tilde{R} \subseteq X \times Y$ be the z-closure of the set $\{ \langle x, y \rangle \mid xRy \}$.

Assume xRy. Pick $z \in R$ such that $r_1(z) = x$, $r_2(z) = y$. Let $\mu \in \alpha(x)$. Note $\alpha \circ r_1 = \mathcal{P}(\mathcal{D}_\omega(A \times r_1)) \circ \gamma$. Choose $\rho \in \gamma(z)$ such that $\mu = \rho \circ (A \times r_1)^{-1}$ or, equivalently, $\mu = \mathcal{D}_\omega(A \times r_1)(\rho)$. Put $\nu = \rho \circ (A \times r_2)^{-1}$. Let $a \in A$ and C be a component of $\tilde{R}$ with faces E, F, i.e. $\pi_1[C] = E$, $\pi_2[C] = F$. We then have $\mu[\{ \langle a, x' \rangle \mid x' \in E \}] = (\rho \circ (A \times r_1)^{-1})(\{ \langle a, x' \rangle \mid x' \in E \}) = \rho[\{ \langle a, z' \rangle \mid r_1(z') \in E \}] = \rho[\{ \langle a, z' \rangle \mid z' \in C \}] = \ \ldots \ = \nu[\{ \langle a, y' \rangle \mid y' \in F \}]$ (where, for $z \in \tilde{R} \setminus R$, $\rho(a, z) = 0$ by definition). So, for any $\mu \in \alpha(x)$ there exists $\nu \in \beta(y)$ such that $\mu[\{ \langle a, x' \rangle \mid x' \in E \}] = \nu[\{ \langle a, y' \rangle \mid y' \in F \}]$ for all a and C. Symmetrically we have that for any $\nu \in \beta(y)$ there exists $\mu \in \alpha(x)$ such that $\mu[\{ \langle a, x' \rangle \mid x' \in E \}] = \nu[\{ \langle a, y' \rangle \mid y' \in F \}]$ for all a and C.

Now assume $x\tilde{R}y$. This implies that there exist $x_1, y_1, \ldots, x_n, y_n$ such that $x_1 R y_1$, $y_1 R^{-1} x_2$, …, $y_{n-1} R^{-1} x_n$, $x_n R y_n$ with $x_1 = x$ and $y_n = y$. By the above it then follows (by induction on n) that $\forall \mu \in \alpha(x_1) \exists \nu \in \beta(y_n)$: $\mu[\{ \langle a, x' \rangle \mid x' \in \pi_1[C] \}] = \nu[\{ \langle a, y' \rangle \mid y' \in \pi_2[C] \}]$ for all actions a and components C. Hence, $x = x_1$, $y = y_n$ are bisimilar according to Definition 4.1.

Second, suppose $R \subseteq X \times Y$ is a Segala-bisimulation relation with $R(x_0, y_0)$. Without loss of generality we can assume that R is z-closed. Let $x \in X, y \in Y$ such that $R(x, y)$, $\mu \in \alpha(x)$, $\nu \in \beta(y)$ such that $\mu[\{ \langle a, x' \rangle \mid x' \in \pi_1[C] \}] = \nu[\{ \langle a, y' \rangle \mid y' \in \pi_2[C] \}]$ for all a, C.

Consider the following network with distinguished elements *src* and *snk*: $\{ (src, \langle a, x' \rangle) \mid x' \in X \} \cup \{ (\langle a, x' \rangle, \langle a, y' \rangle) \mid R(x', y') \} \cup \{ (\langle a, y' \rangle, snk) \mid y' \in Y \}$. Decorate the source edges and sink edges with capacities $c(src, \langle a, x' \rangle) = \mu(\langle a, x' \rangle)$, $c(\langle a, y' \rangle, snk) = \nu(\langle a, y' \rangle)$, respectively. Let the capacities for the remaining edges be 1. For any component C the nodes $\{ (\langle a, x' \rangle, \langle a, y' \rangle) \mid C(x', y') \}$ span a complete bi-partite subgraph. Moverover, $\sum \{ c(src, \langle a, x' \rangle) \mid x' \in \pi_1[C] \} = \sum \{ \mu(\langle a, x' \rangle) \mid x' \in \pi_1[C] \} = \mu[\{ \langle a, x' \rangle \mid x' \in \pi_1[C] \}] = \nu[\{ \langle a, y' \rangle \mid y' \in \pi_2[C] \}] = \sum \{ \nu(\langle a, y' \rangle) \mid y' \in \pi_2[C] \} = \sum \{ c(\langle a, y' \rangle, snk) \mid y' \in \pi_2[C] \}$.

We observe that for each component either the E-face or the F-face is included in the cut. By Segala-bisimilarity we have that the E-face and the F-face of a component contribute equally to a cut. It follows that the value of the minimal cut is the value of the cut between the source and the rest of the graph. By construction the value of this cut equals 1. By the Max-flow Min-cut Theorem 2.1 it follows that there is a flow of value 1, i.e. there exist weights $wgt(\langle a,x'\rangle,\langle a,y'\rangle)$ such that for fixed x' and y', respectively, $\mu(\langle a,x'\rangle) = \sum\{\, wgt(\langle a,x'\rangle,\langle a,y'\rangle) \mid C(x',y')\,\}$ and $\nu(\langle a,y'\rangle) = \sum\{\, wgt(\langle a,x'\rangle,\langle a,y'\rangle) \mid C(x',y')\,\}$. Now, define the probability distribution $\rho_{\mu,\nu}\colon R \to [0,1]$ by $\rho_{\mu,\nu}(\langle a,(x',y')\rangle) = wgt(\langle a,x'\rangle,\langle a,y'\rangle)$ and put $\gamma(x,y) = \{\, \rho_{\mu,\nu} \mid \mu \in \alpha(x), \nu \in \beta(y), R(\mu,\nu)\,\}$. We claim that

$$\mathcal{P}(\mathcal{D}_\omega(A\times\pi_1))(\gamma(x,y)) = \alpha(x) \text{ and } \mathcal{P}(\mathcal{D}_\omega(A\times\pi_2))(\gamma(x,y)) = \beta(y).$$

We only check the case for $\alpha(x)$: Note $\mathcal{P}(\mathcal{D}_\omega(A\times\pi_1))(\gamma(x,y)) = \{\, \mathcal{D}_\omega(A\times\pi_1)(\rho) \mid \rho \in \gamma(x,y)\,\} = \{\, \rho\circ(A\times\pi_1)^{-1} \mid \rho = \rho_{\mu,\nu}, \mu\in\alpha(x), \nu\in\beta(y), R(\mu,\nu)\,\}$. "$\subseteq$" Pick $\mu \in \alpha(x)$, $\nu \in \beta(y)$ with $R(\mu,\nu)$. We then have, for any pair $\langle a,x'\rangle$, $(\rho\circ(A\times\pi_1)^{-1})(\langle a,x'\rangle) = \rho[\{\,\langle a,(x',y')\rangle \mid R(x',y')\,\}] = \mu(\langle a,x'\rangle)$. Hence $\rho\circ(A\times\pi_1)^{-1} = \mu$ and, in particular $\rho\circ(A\times\pi_1)^{-1} \in \alpha(x)$.

"$\supseteq$" Pick $\mu\in\alpha(x)$. Since $R(x,y)$ we can choose $\nu\in\beta(y)$ such that $R(\mu,\nu)$. Then $\rho_{\mu,\nu} \in \gamma(x,y)$ by definition of γ, and, following a similar argument as above, $\rho_{\mu,\nu}\circ(A\times\pi_1)^{-1} = \mu$, and, in particular, $\mu \in \mathcal{P}(\mathcal{D}_\omega(A\times\pi_1))(\gamma(x,y))$.

We conclude that $\mathcal{P}(\mathcal{D}_\omega(A\times\pi_1))(\gamma(x,y)) = \alpha(x)$ for any $(x,y)\in R$. Hence $\mathcal{P}(\mathcal{D}_\omega(A\times\pi_1))\circ\gamma = \alpha\circ\pi_1$. By symmetry, $\mathcal{P}(\mathcal{D}_\omega(A\times\pi_2))\circ\gamma = \beta\circ\pi_2$. So, $\langle R,\gamma\rangle$ is a mediating **Seg**-coalgebra for $\langle X,\alpha\rangle$, and $\langle Y,\beta\rangle$. By assumption x_0 and y_0 are connected by R. □

The Max-flow Min-cut Theorem as applied above, following [Jon89,VR99], is an elegant tool for the construction of the mediating morphism γ. However, because of the special form of the network at hand, we could have dispensed with it as an ad-hoc argument would have sufficed. In a different situation, viz. in proving full abstraction, Worell circumvents the application of the graph-theoretical theorem by exploiting the notion of a so-called F-simulation (cf. [Wor00]).

A hierarchy of probabilistic system types

In this part we compare the introduced probabilistic system types, using the results of Section 3.

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 bisimulation then we say that the class $\mathsf{Coalg}_\mathcal{F}$ is *coalgebraically embedded* in the class $\mathsf{Coalg}_\mathcal{G}$, denoted by $\mathsf{Coalg}_\mathcal{F} \to \mathsf{Coalg}_\mathcal{G}$. Clearly $\to$ is reflexive and transitive.

The next theorem states the results on coalgebraic embeddings between the previously introduced probabilistic system types. For readability we recall the

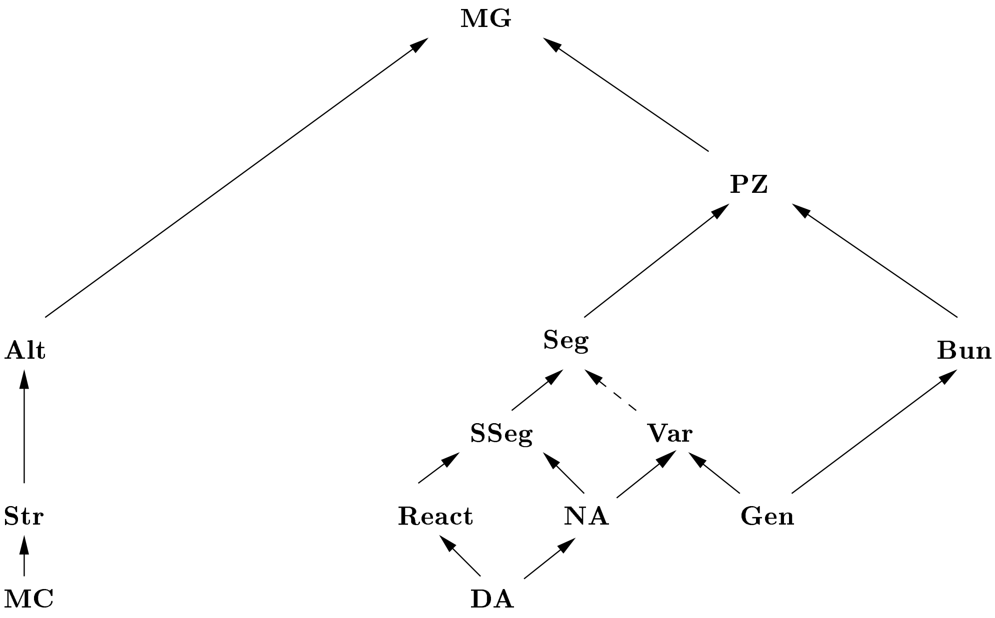


Fig. 1. Hierarchy of probabilistic system types

defining functors and the notation used for each class in the following table.

MC	$\mathcal{D}_\omega$	**Alt**	$\mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}d)$
DA	$(\mathcal{I}d + 1)^A$	**Seg**	$\mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I}d)$
NA	$\mathcal{P}(A \times \mathcal{I}d)$	**SSeg**	$\mathcal{P}(A \times \mathcal{D}_\omega)$
React	$(\mathcal{D}_\omega + 1)^A$	**Bun**	$\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d) + 1$
Gen	$\mathcal{D}_\omega(A \times \mathcal{I}d) + 1$	**PZ**	$\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$
Str	$\mathcal{D}_\omega + (A \times \mathcal{I}d) + 1$	**MG**	$\mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d + \mathcal{I}d)$
Var	$\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$		

Theorem 4.3 *The coalgebraic embeddings presented in Figure 1 hold among the probabilistic system types.*

Proof. By Theorem 3.9, if $\mathcal{F}, \mathcal{G}$ are functors on Set such that $\mathcal{F}$ preserves weak pullbacks and there is a componentwise injective natural transformation from $\mathcal{F}$ to $\mathcal{G}$, then $\mathsf{Coalg}_{\mathcal{F}} \to \mathsf{Coalg}_{\mathcal{G}}$. We note that:

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

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

It follows that all functors involved have the desired property. Hence in all of the cases it is enough to construct a componentwise injective natural transformation. We start by defining some elementary natural transformations and collecting some simple properties. Let $\mathcal{F}, \mathcal{G}, \mathcal{H}$ be functors on Set.

- $1 \overset{\eta}{\Rightarrow} \mathcal{P}$, where $\eta_X(*) = \emptyset$.
- The left and right coproduct injections i_l and i_r are natural transformations $\mathcal{F} \overset{i_l}{\Rightarrow} \mathcal{F} + \mathcal{G}$, $\mathcal{G} \overset{i_r}{\Rightarrow} \mathcal{F} + \mathcal{G}$ 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 $\delta_X(x) = \mu^1_x$, $\mu^1_x(x) = 1$ 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 $\phi_X(f) = \mathit{Graph}(f) = \{(a, f(a)) \mid f(a) \in X\}$ for $f : A \to X + 1$, form a natural transformation $(\mathcal{I}d + 1)^A \overset{\phi}{\Rightarrow} \mathcal{P}(A \times \mathcal{I}d)$
- If $\mathcal{F} \overset{\tau_1}{\Rightarrow} \mathcal{H}$ and $\mathcal{G} \overset{\tau_2}{\Rightarrow} \mathcal{H}$ then $\mathcal{F} + \mathcal{G} \overset{[\tau_1,\tau_2]}{\Rightarrow} \mathcal{H}$ holds.
- If $\mathcal{F}_1 \overset{\tau_1}{\Rightarrow} \mathcal{G}_1$ and $\mathcal{F}_2 \overset{\tau_2}{\Rightarrow} \mathcal{G}_2$, then $\mathcal{F}_1 + \mathcal{F}_2 \overset{\tau_1 + \tau_2}{\Rightarrow} \mathcal{G}_1 + \mathcal{G}_2$. If τ_1 and τ_2 are componentwise injective, so is $\tau_1 + \tau_2$.
- If $\mathcal{F} \overset{\tau}{\Rightarrow} \mathcal{G}$ then $\mathcal{F}\mathcal{H} \overset{\tau\mathcal{H}}{\Rightarrow} \mathcal{G}\mathcal{H}$ where $\tau\mathcal{H}_X = \tau_{\mathcal{H}X}$. If τ is componentwise injective, then so is $\tau\mathcal{H}$.
- If $\mathcal{F} \overset{\tau}{\Rightarrow} \mathcal{G}$ then $\mathcal{H}\mathcal{F} \overset{\mathcal{H}\tau}{\Rightarrow} \mathcal{H}\mathcal{G}$ where $\mathcal{H}\tau_X = \mathcal{H}(\tau_X)$. 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 chech 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: $\mathcal{D}_\omega \overset{i_l}{\Rightarrow} \mathcal{D}_\omega + (A \times \mathcal{I}d) + 1$

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: $\mathcal{P}(A \times \mathcal{I}d) \overset{i_r}{\Rightarrow} \mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$

Gen → Var: $\mathcal{D}_\omega(A \times \mathcal{I}d) + 1 \overset{id + \eta\mathcal{F}}{\Rightarrow} \mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = A \times \mathcal{I}d$.

Gen → Bun: $\mathcal{D}_\omega(A \times \mathcal{I}d) + 1 \overset{\mathcal{D}_\omega\sigma\mathcal{F} + id}{\Rightarrow} \mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d) + 1$, for $\mathcal{F} = A \times \mathcal{I}d$.

Var → Seg: $\mathcal{D}_\omega(A \times \mathcal{I}d) + \mathcal{P}(A \times \mathcal{I}d) + 1 \overset{[\sigma\mathcal{D}_\omega, \mathcal{P}\delta, \eta]\mathcal{F}}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega(A \times \mathcal{I}d)$ for $\mathcal{F} = A \times \mathcal{I}d$ and the transformation $[\sigma\mathcal{D}_\omega, \mathcal{P}\delta, \eta]\mathcal{F}$ is componentwise injective up to identification of the two degenerated steps i.e. identification of $\mu^1_{\langle a,x\rangle}$ and $\{\langle a,x\rangle\}$. Note that in the picture we draw a dashed arrow for this coalgebraic embedding. As a remark, the transitive solid arrow **Gen** $\to$ **Seg** still holds.

SSeg $\to$ **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 $\tau_X(\langle a, \mu\rangle) = \mu^1_a \times \mu$, where $\mu \times \mu'(\langle x, x'\rangle) = \mu(x) \cdot \mu'(x')$. All components of τ are injective.

Str $\to$ **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 $\to$ **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 $\to$ **PZ:** $\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d) + 1 \overset{[\sigma,\eta]\mathcal{F}}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$, for $\mathcal{F} = \mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d)$, and $[\sigma,\eta]\mathcal{F}$ is componentwise injective.

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

Alt $\to$ **MG:** $\mathcal{D}_\omega + \mathcal{P}(A \times \mathcal{I}d) \overset{\sigma\mathcal{H}\circ[\mathcal{D}_\omega(\sigma\mathcal{F}\circ i_r),\delta\mathcal{G}\circ\mathcal{P}i_l]}{\Rightarrow} \mathcal{P}\mathcal{D}_\omega\mathcal{P}(A \times \mathcal{I}d + \mathcal{I}d)$. Here injections go from $\mathcal{I}d$ 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.

□

References

[Ber99] M. Bernardo. *Theory and Application of Extended Markovian Process Algebra*. PhD thesis, University of Bologna, 1999.

[BSV02] F. Bartels, A. Sokolova, and E.P. de Vink. Probabilistic automata based models. In *VOSS GI-Dagstuhl seminar*. LNCS, 2002. to appear.

[DEP02] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for Labelled Markov Processes. *Information and Computation*, 179:163–193, 2002.

[DHK98] P. D'Argenio, H. Hermanns, and J.-P. Katoen. On generative parallel composition. In *Proc. Probmiv'98*, volume 22 of *ENTCS*. Elsevier, 1998.

[GS00] 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.

[GSS95] 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.

[Han94] H. Hansson. Time and probability in formal design of distributed systems. In *Real-Time Safety Critical Systems*, volume 1. Elsevier, 1994.

[Her98] 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.

[Hil94] 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.

[Jon89] C. Jones. *Probabilistic Non-determinism.* PhD thesis, University of Edinburgh, 1989.

[JR96] 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.

[Kur00] A. Kurz. *Logics for Coalgebras and Applications to Computer Science.* PhD thesis, Ludwig-Maximilians-Universität München, 2000.

[LS91] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[Mos99] L.S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.

[PZ93] A. Pnueli and L. Zuck. Probabilistic verification. *Information and Computation*, 103:1–29, 1993.

[RT93] 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.

[Rut00] J.J.M.M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[Seg95] R. Segala. *Modeling and verification of randomized distributed real-time systems.* PhD thesis, MIT, Department of EECS, 1995.

[SL94] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. Concur'94*, pages 481–496. LNCS 836, 1994.

[Var85] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. FOCS'85*, pages 327–338, Portland, Oregon, 1985.

[VR99] 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.

[Wol00] U. Wolter. On corelations, cokernels, and coequations. In H. Reichel, editor, *Proc. CMCS 2000.* ENTCS 33, 2000. 20pp.

[Wor00] J. Worell. Coinduction for recursive data types: partial orders, metric spaces and ω-categories. In H. Reichel, editor, *Proc. CMCS 2000.* ENTCS 33, 2000. 20pp.