Trace Semantics via Determinization

Bart Jacobs[1], Alexandra Silva[1⋆], and Ana Sokolova[2]

[1] Institute for Computing and Information Sciences, Radboud University Nijmegen
[2] Department of Computer Sciences, University of Salzburg

{bart,alexandra}@cs.ru.nl, anas@cs.uni-salzburg.at

Abstract. This paper takes a fresh look at the topic of trace semantics in the theory of coalgebras. The first development of coalgebraic trace semantics used final coalgebras in Kleisli categories, stemming from an initial algebra in the underlying category. This approach requires some non-trivial assumptions, like dcpo enrichment, which do not always hold, even in cases where one can reasonably speak of traces (like for weighted automata). More recently, it has been noticed that trace semantics can also arise by first performing a determinization construction. In this paper, we develop a systematic approach, in which the two approaches correspond to different orders of composing a functor and a monad, and accordingly, to different distributive laws. The relevant final coalgebra that gives rise to trace semantics does not live in a Kleisli category, but more generally, in a category of Eilenberg-Moore algebras. In order to exploit its finality, we identify an extension operation, that changes the state space of a coalgebra into a free algebra, which abstractly captures determinization of automata. Notably, we show that the two different views on trace semantics are equivalent, in the examples where both approaches are applicable.

1 Introduction

Coalgebras provide an abstract framework for state-based computation. In general, a coalgebra is a map of the form $X \to H(X)$, where X is a state space and H is a functor that captures the kind of computation involved. Often, this H is, or contains, a monad T, providing certain computational effects, such as when T is lift $1 + (-)$, powerset $\mathcal{P}$ or distribution $\mathcal{D}$, giving partial, non-deterministic and probabilistic computation.

In the case such an H contains a monad T, it turns out that there are two archetypical forms in which T plays a role, namely in:

$$X \longrightarrow G(TX) \qquad \text{or} \qquad X \longrightarrow T(FX) \tag{1}$$

In the first case, the monad T occurs inside a functor G that typically handles input and output. In the second case the monad T is on the outside, encapsulating a form of computation over a functor F, that typically describes the transitions involved. In some cases these two forms are equivalent, for instance for non-deterministic automata with

[⋆] Also affiliated to Centrum Wiskunde & Informatica (Amsterdam, The Netherlands) and HASLab / INESC TEC, Universidade do Minho (Braga, Portugal).

labels—given by a set A—and termination. They involve the powerset monad $\mathcal{P}$ and can be described equivalently as:

$$X \longrightarrow 2 \times (\mathcal{P}X)^A \qquad\qquad \text{or} \qquad\qquad X \longrightarrow \mathcal{P}(1 + A \times X)$$

where on the left-hand-side we have the functor $G = 2 \times (-)^A$ and on the right-hand-side $F = 1 + A \times (-)$. These descriptions are equivalent because the powerset monad $\mathcal{P}$ is special (it is "additive", see [8], mapping coproducts to products). In fact, there are isomorphisms:

$$\mathcal{P}(1 + A \times X) \;\cong\; \mathcal{P}(1) \times \mathcal{P}(A \times X) \;\cong\; 2 \times (\mathcal{P}X)^A. \tag{2}$$

Classically, to study language or trace equivalence for non-deterministic automata there is a standard construction called determinization [14]. It involves changing the state space X into a state space $\mathcal{P}X$. In doing so, the transition structure becomes much simpler (in particular, deterministic).

In this paper, we are interested in a similar determinization construction, but on a more abstract level. It involves changing the state space from X into $T(X)$, for a monad T. It turns out that this can be done relatively easily for coalgebras of the form $X \to G(TX)$, on the left in (1), as illustrated in [25]: it involves a distributive law between G and T, and such law corresponds to a lifting $\widehat{G}$ of the functor G to the category $\mathcal{EM}(T)$ of Eilenberg-Moore algebras of T. Moreover, the final G-coalgebra lifts to a final $\widehat{G}$-coalgebra in $\mathcal{EM}(T)$. In this way, one obtains final coalgebra semantics in a category of algebras. It yields a first form of "$\mathcal{EM}$" extension semantics, see Definition 5.

Categorically, this extension $X \mapsto T(X)$ is given by the free algebra functor $\mathbf{C} \to \mathcal{EM}(T)$. Extension for coalgebras of the form $X \to T(FX)$ on the right in (1) is more complicated; it involves a comparison functor $\mathcal{K}\ell(T) \to \mathcal{EM}(T)$. We proceed by first translating them to coalgebras of the lifted functor $\widehat{G}$ via a suitable law $\mathfrak{e}\colon TF \Rightarrow GT$, see Section 6 (and [1]). It will be shown that the resulting trace semantics, in categories of algebras, as sketched above,

- not only includes the trace semantics in Kleisli categories developed in [12];
- but also covers more examples; in particular, it covers trace semantics for weighted automata $X \to \mathcal{M}(1 + A \times X)$, involving the multiset monad $\mathcal{M}$. This monad does not fit in the trace framework of [12] because its Kleisli category is not dcpo-enriched.

The technical (categorical) core of the paper concentrates on lifting the comparison functor $\mathcal{K}\ell(T) \to \mathcal{EM}(T)$ to categories of coalgebras $\mathbf{CoAlg}(\widehat{F}) \to \mathbf{CoAlg}(\widehat{G})$, of lifted functors $\widehat{F}, \widehat{G}$. We specialize this framework by taking G to be the functor $B \times (-)^A$ for deterministic automata. Its final coalgebra $B^{A^\star}$ gives trace semantics. In principle, our framework is general enough to allow a different semantics, like "tree" semantics, by using a different functor G.

The paper is organized as follows. The first section below recalls the basics about monads and the associated Kleisli category and category of (Eilenberg-Moore) algebras. Section 3 gives a systematic account of liftings of functors to such (Kleisli and algebra) categories, and of distributive laws; it includes a lifting result for final coalgebras to categories of Eilenberg-Moore algebras. Subsequently, Section 4 briefly recalls the coalgebraic description of deterministic automata, their final coalgebras, and the lifting of these final coalgebras to categories of Eilenberg-Moore algebras. Section 5 contains two examples of the application of $\mathcal{EM}$-extension semantics: for classical non-deterministic automata and, more interestingly, a new example mixing probability and non-determinism, for simple Segala systems. In Section 6, we develop an extension semantics for coalgebras of type TF (Definition 14, via Theorem 13) and show that the Kleisli trace semantics from [12] fits in the current setting (Proposition 15). Section 7 presents examples of the $\mathcal{K}\ell$-extension semantics, including examples already treated in [12] and, more notably, examples that cannot be studied in the framework of [12]. Section 8 contains concluding remarks and pointers for future work.

2 Monads and their Kleisli and Eilenberg-Moore categories

This section recalls the basics of the theory of monads, as needed here. For more information, see *e.g.* [19,2,18,4]. A monad is a functor $T\colon \mathbf{C} \to \mathbf{C}$ together with two natural transformations: a unit $\eta\colon \mathrm{id}_{\mathbf{C}} \Rightarrow T$ and multiplication $\mu\colon T^2 \Rightarrow T$. These are required to make the following diagrams commute, for $X \in \mathbf{C}$.

$$\begin{tikzcd}
T(X) \arrow[r, "\eta_{T(X)}"] \arrow[dr, equal] & T^2(X) \arrow[d, "\mu_X"] & T(X) \arrow[l, "T(\eta_X)"'] \arrow[dl, equal] \\
& T(X) &
\end{tikzcd}
\qquad
\begin{tikzcd}
T^3(X) \arrow[r, "\mu_{T(X)}"] \arrow[d, "T(\mu_X)"'] & T^2(X) \arrow[d, "\mu_X"] \\
T^2 \arrow[r, "\mu_X"'] & T(X)
\end{tikzcd}$$

We briefly describe the examples of monads on $\mathbf{Sets}$ that we use in this paper.

- The powerset monad $\mathcal{P}$ maps a set X to the set $\mathcal{P}X$ of subsets of X, and a function $f\colon X \to Y$ to $\mathcal{P}(f)\colon \mathcal{P}X \to \mathcal{P}Y$ given by direct image. Its unit is given by singleton $\eta(x) = \{x\}$ and multiplication by union $\mu(\{X_i \in \mathcal{P}X \mid i \in I\}) = \bigcup_{i\in I} X_i$.
- The subprobability distribution monad $\mathcal{D}$ is defined, for a set X and a function $f\colon X \to Y$, as

$$\mathcal{D}X \;=\; \{\varphi\colon X \to [0,1] \mid \textstyle\sum_{x\in X}\varphi(x) \le 1\}ascoa \qquad \mathcal{D}f(\varphi)(y) \;=\; \sum_{x\in f^{-1}(y)} \varphi(x).$$

  The support set of a distribution $\varphi \in \mathcal{D}X$ is defined as

$$\mathrm{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}.$$

  Note that we do not restrict distributions to finitely supported ones in the definition of $\mathcal{D}$. The unit of $\mathcal{D}$ is given by a Dirac distribution $\eta(x) = \delta_x = (x \mapsto 1)$ for $x \in X$ and the multiplication by $\mu(\varPhi)(x) = \sum\limits_{\varphi\in\mathrm{supp}(\varPhi)} \varPhi(\varphi)\cdot\varphi(x)$ for $\varPhi \in \mathcal{D}\mathcal{D}X$.
- For a semiring S, the multiset monad $\mathcal{M}_S$ is defined on a set X as:

$$\mathcal{M}_S X \;=\; \{\varphi\colon X \to S \mid \mathrm{supp}(\varphi) \text{ is finite }\}.$$

This monad captures multisets $\varphi \in \mathcal{M}_S X$, where the value $\varphi(x) \in S$ gives the multiplicity of the element $x \in X$. When $S = \mathbb{N}$, this is sometimes called the bag monad.

On functions, $\mathcal{M}_S$ is defined like the subdistribution monad $\mathcal{D}$. Again, the support set of a multiset $\varphi \in \mathcal{M}_S X$ is defined as $\mathrm{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$. The finite support requirement is necessary for $\mathcal{M}$ to be a monad. The unit η and multiplication μ of $\mathcal{M}_S$ are defined in the same way as for $\mathcal{D}$.

$$\begin{tikzcd} \mathcal{K}\ell(T) \arrow[rr, "E"] \arrow[dr] & & \mathcal{EM}(T) \arrow[dl] \\ & \mathbf{C} & \end{tikzcd}$$

With a monad T on a category $\mathbf{C}$ one associates two categories and a comparison functor E between them, as on the right. The "Kleisli" category $\mathcal{K}\ell(T)$ is used to capture computations of type T, in the paradigm that uses monads for effects in a functional world [20]. The objects of the "Eilenberg-Moore" category $\mathcal{EM}(T)$ are algebraic structures, in abstract form. Objects of $\mathcal{EM}(T)$ are called 'algebras', or sometimes 'Eilenberg-Moore algebras' to avoid possible confusion with algebras of a functor F. The latter also form a category written as $\mathbf{Alg}(F)$. The comparison functor $E\colon \mathcal{K}\ell(T) \to \mathcal{EM}(T)$ plays here the role of pure determinization operation. The categories $\mathcal{K}\ell(T)$ and $\mathcal{EM}(T)$ are initial and final in a suitable sense, see [19] for details. This determines the determinization functor. We now describe the above points in more detail.

The Kleisli category $\mathcal{K}\ell(T)$ has the same objects as the underlying category $\mathbf{C}$, but morphisms $X \to Y$ in $\mathcal{K}\ell(T)$ are maps $X \to T(Y)$ in $\mathbf{C}$. The identity map $X \to X$ in $\mathcal{K}\ell(T)$ is T's unit $\eta_X\colon X \to T(X)$; and composition $g \odot f$ in $\mathcal{K}\ell(T)$ uses T multiplication in: $g \odot f = \mu \circ T(g) \circ f$. There is a forgetful functor $\mathcal{U}\colon \mathcal{K}\ell(T) \to \mathbf{C}$, sending X to $T(X)$ and f to $\mu \circ T(f)$. This functor has a left adjoint $\mathcal{F}$, given by $\mathcal{F}(X) = X$ and $\mathcal{F}(f) = \eta \circ f$. Such a Kleisli category $\mathcal{K}\ell(T)$ inherits colimits from the underlying category $\mathbf{C}$.

The category $\mathcal{EM}(T)$ of Eilenberg-Moore algebras has as objects maps of the form $a\colon T(X) \to X$, making the first two diagrams below commute.

$$\begin{tikzcd} X \arrow[r, "\eta"] \arrow[dr, equal] & TX \arrow[d, "a"] \\ & X \end{tikzcd} \qquad \begin{tikzcd} T^2X \arrow[r, "T(a)"] \arrow[d, "\mu"'] & TX \arrow[d, "a"] \\ TX \arrow[r, "a"'] & X \end{tikzcd} \qquad \begin{tikzcd} TX \arrow[r, "T(f)"] \arrow[d, "a"'] & TY \arrow[d, "b"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

A homomorphism of algebras $\big(TX \xrightarrow{a} X\big) \to \big(TY \xrightarrow{b} Y\big)$ is a map $f\colon X \to Y$ in $\mathbf{C}$ between the underlying objects making the diagram above on the right commute. The diagram in the middle thus says that the map a is a homomorphism $\mu \to a$. The forgetful functor $\mathcal{U}\colon \mathcal{EM}(T) \to \mathbf{C}$ has a left adjoint $\mathcal{F}$, mapping an object $X \in \mathbf{X}$ to the (free) algebra $\mu_X\colon T^2(X) \to T(X)$ with carrier $T(X)$.

Each category $\mathcal{EM}(T)$ inherits limits from the category $\mathbf{C}$. In the special case where $\mathbf{C} = \mathbf{Sets}$, the category of sets and functions (our standard universe), the category $\mathcal{EM}(T)$ is not only complete but also cocomplete (see [2, § 9.3, Prop. 4]).

The extension functor $E\colon \mathcal{K}\ell(T) \to \mathcal{EM}(T)$ sends an object $X \in \mathcal{K}\ell(T)$ to the free algebra $E(X) = (\mu\colon T^2(X) \to T(X))$. For a morphism $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, that is, $f\colon X \to T(Y)$ in $\mathbf{C}$, we have $E(f) = \mu \circ T(f)\colon T(X) \to T(Y)$. It forms a map of algebras. Sometimes this $E(f)$ is called the "Kleisli extension" of f.

3 Liftings to Kleisli and Eilenberg-Moore categories

In this section we consider the situation where we have a monad $T\colon \mathbf{C} \to \mathbf{C}$, with unit η and multiplication μ, and endofunctors $F, G\colon \mathbf{C} \to \mathbf{C}$ on the same category $\mathbf{C}$. We will be interested in distributive laws between T, and F or G. This can be of two forms, namely:

$FT \Longrightarrow TF$	"F distributes over T"	"$\mathcal{K}\ell$-law"
$TG \Longrightarrow GT$	"T distributes over G"	"$\mathcal{EM}$-law"

It is rather difficult to remember who distributes over who and so we prefer to use the terminology "$\mathcal{K}\ell$-law" and "$\mathcal{EM}$-law". This is justified by Proposition 1 below.

But first we have to be more precise about what a distributive law is. It is a natural transformation, of the form $\lambda\colon FT \Rightarrow TF$ or $\rho\colon TG \Rightarrow GT$, that commutes appropriately with the unit and multiplication of the monad. For a $\mathcal{K}\ell$-law $\lambda\colon FT \Rightarrow TF$ this means:

$$\begin{tikzcd} FX \arrow[r, equal] \arrow[d, "F(\eta_X)"'] & FX \arrow[d, "\eta_{FX}"] \\ FTX \arrow[r, "\lambda_X"'] & TFX \end{tikzcd} \qquad \begin{tikzcd} FT^2X \arrow[r, "\lambda_{TX}"] \arrow[d, "F(\mu_X)"'] & TFTX \arrow[r, "T(\lambda_X)"] & T^2FX \arrow[d, "\mu_{FX}"] \\ FTX \arrow[rr, "\lambda_X"'] & & TFX \end{tikzcd} \tag{3}$$

And for an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ it means:

$$\begin{tikzcd} GX \arrow[r, equal] \arrow[d, "\eta_{GX}"'] & GX \arrow[d, "G(\eta_X)"] \\ TGX \arrow[r, "\rho_X"'] & GTX \end{tikzcd} \qquad \begin{tikzcd} T^2GX \arrow[r, "T(\rho_X)"] \arrow[d, "\mu_{GX}"'] & TGTX \arrow[r, "\rho_{TX}"] & GT^2X \arrow[d, "G(\mu_X)"] \\ TGX \arrow[rr, "\rho_X"'] & & GTX \end{tikzcd} \tag{4}$$

The following "folklore" result gives an alternative description of distributive laws in terms of liftings to Kleisli and Eilenberg-Moore categories, see also [16] or [1].

Proposition 1 ("laws and liftings"). *Assume a monad T and endofunctors F, G on the same category $\mathbf{C}$, as above. There are bijective correspondences between $\mathcal{K}\ell/\mathcal{EM}$-laws and liftings of F to $\mathcal{K}\ell/\mathcal{EM}$-categories, in:*

$$\frac{\mathcal{K}\ell\text{-law } \lambda\colon FT \Rightarrow TF}{\begin{tikzcd} \mathcal{K}\ell(T) \arrow[r, "L"] \arrow[d] & \mathcal{K}\ell(T) \arrow[d] \\ \mathbf{C} \arrow[r, "F"'] & \mathbf{C} \end{tikzcd}} \qquad \frac{\mathcal{EM}\text{-law } \rho\colon TG \Rightarrow GT}{\begin{tikzcd} \mathcal{EM}(T) \arrow[r, "R"] \arrow[d] & \mathcal{EM}(T) \arrow[d] \\ \mathbf{C} \arrow[r, "G"'] & \mathbf{C} \end{tikzcd}}$$

Proof. Assuming a $\mathcal{K}\ell$-law $\lambda\colon FT \Rightarrow TF$ we can define $L\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ as:

$$L(X) \;=\; F(X) \qquad\quad L\big(X \xrightarrow{f} Y\big) \;=\; \big(F(X) \xrightarrow{F(f)} F(TY) \xrightarrow{\lambda_X} T(FY)\big).$$

The above two requirements (3) for λ precisely say that L is a functor.

Conversely, assume there is a functor $L\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ in a commuting square as described in the proposition. Then, on objects, $L(X) = F(X)$. Further, for a map $f\colon X \to TY$ in $\mathbf{C}$ we get $L(f)\colon FX \to TFY$ in $\mathbf{C}$. This suggests how to define a distributive law: the identity map $\mathrm{id}_{TX}\colon TX \to TX$ in $\mathbf{C}$ forms a map $TX \to X$ in $\mathcal{K}\ell(T)$, so that we can define $\lambda_X = L(\mathrm{id}_{TX})\colon FTX \to TFX$ in $\mathbf{C}$. It satisfies (3).

For the second correspondence assume we have an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$. It gives rise to a functor $R\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$ by:

$$\begin{pmatrix} TX \\ \downarrow a \\ X \end{pmatrix} \longmapsto \begin{pmatrix} TGX \\ \downarrow G(a)\circ\rho \\ GX \end{pmatrix} \qquad \text{and} \qquad f \longmapsto G(f).$$

The equations (4) guarantee that this yields a new T-algebra.

In the reverse direction, assume a lifting $R\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$. Applying it to the multiplication μ_X yields an algebra $R(\mu_X)\colon T(GTX) \to GTX$. We then define $\rho_X = R(\mu_X) \circ TG(\eta_X)\colon TGX \to GTX$. Remaining details are left to the reader. □

In what follows we shall simply write $\widehat{F}$ / $\widehat{G}$ for the lifting of F / G, both when it comes from a $\mathcal{K}\ell$-law λ or from an $\mathcal{EM}$-law ρ. Usually these laws are fixed, so confusion is unlikely, and a light, overloaded notation is preferred.

The next result (see also [1]) is not really used in this paper, but it is a natural sequel to the previous proposition since it relates the liftings $\widehat{F}, \widehat{G}$ to the standard adjunctions. Recall that we write $\mathbf{Alg}(-)$ and $\mathbf{CoAlg}(-)$ for categories of algebras and coalgebras of a *functor*, not of a (co)monad.

Proposition 2. *In presence of a $\mathcal{K}\ell$-law and an $\mathcal{EM}$-law, the adjunctions $\mathbf{C} \rightleftarrows \mathcal{K}\ell(T)$ and $\mathbf{C} \rightleftarrows \mathcal{EM}(T)$ lift to adjunctions between categories of, respectively, algebras and coalgebras, as described below.*

$$\begin{tikzcd}
\mathbf{Alg}(F) \arrow[r, bend left, "\widehat{\mathcal{F}}"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathbf{Alg}(\widehat{F}) \arrow[l, bend left, "\widehat{\mathcal{U}}"] \arrow[d] \\
\mathbf{C} \arrow[loop left, "F"] \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathcal{K}\ell(T) \arrow[l, bend left, "\mathcal{U}"] \arrow[loop right, "\widehat{F}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{CoAlg}(G) \arrow[r, bend left, "\widehat{\mathcal{F}}"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathbf{CoAlg}(\widehat{G}) \arrow[l, bend left, "\widehat{\mathcal{U}}"] \arrow[d] \\
\mathbf{C} \arrow[loop left, "G"] \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathcal{EM}(T) \arrow[l, bend left, "\mathcal{U}"] \arrow[loop right, "\widehat{G}"]
\end{tikzcd}$$

There is another lifting result, for free functors only, that is relevant in this setting.

Lemma 3. *In presence of a $\mathcal{K}\ell$-law the free functor $\mathcal{F}\colon \mathbf{C} \to \mathcal{K}\ell(T)$ can be lifted, and similarly, given an $\mathcal{EM}$-law the free algebra functor $\mathcal{F}\colon \mathbf{C} \to \mathcal{EM}(T)$ can be lifted:*

$$\begin{tikzcd}
\mathbf{CoAlg}(TF) \arrow[r, "\mathcal{F}_{\mathcal{K}\ell}"] \arrow[d] & \mathbf{CoAlg}(\widehat{F}) \arrow[d] \\
\mathbf{C} \arrow[loop left, "F"] \arrow[loop below, "T"] \arrow[r, "\mathcal{F}"] & \mathcal{K}\ell(T) \arrow[loop right, "\widehat{F}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{CoAlg}(GT) \arrow[r, "\mathcal{F}_{\mathcal{EM}}"] \arrow[d] & \mathbf{CoAlg}(\widehat{G}) \arrow[d] \\
\mathbf{C} \arrow[loop left, "G"] \arrow[loop below, "T"] \arrow[r, "\mathcal{F}"] & \mathcal{EM}(T) \arrow[loop right, "\widehat{G}"]
\end{tikzcd} \qquad (5)$$

The functor $\mathbf{CoAlg}(GT) \to \mathbf{CoAlg}(\widehat{G})$ on the right gives an abstract description of what is called the generalized powerset construction in [25].

Proof. The first part is easy, since the functor $\mathcal{F}_{\mathcal{K}\ell}\colon \mathbf{CoAlg}(TF) \to \mathbf{CoAlg}(\widehat{F})$ is the identity on objects; it sends a map f of TF-coalgebras to $\mathcal{F}(f) = \eta \circ f$.

Next, assuming an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ one defines $\mathcal{F}_{\mathcal{EM}}\colon \mathbf{CoAlg}(GT) \to \mathbf{CoAlg}(\widehat{G})$ by

$$\mathcal{F}_{\mathcal{EM}}\Big(X \xrightarrow{c} GTX\Big) \;=\; \Big(TX \xrightarrow{T(c)} TGTX \xrightarrow{\rho_{TX}} GT^2X \xrightarrow{G(\mu)} GTX\Big). \qquad (6)$$

It is not hard to see that $\mathcal{F}_{\mathcal{EM}}(c)$ is a coalgebra $\mu_X \to \widehat{G}(\mu_X)$ on the free algebra μ_X. On morphisms one simply has $\mathcal{F}_{\mathcal{EM}}(f) = T(f)$. $\square$

$\mathcal{K}\ell$-laws are used to obtain final coalgebras in Kleisli categories ([12]) but require non-trivial side-conditions, like enrichement in dcpo's. For $\mathcal{EM}$-laws the situation is much easier, see below; instances of this result have been studied in [25], see also [1].

Proposition 4. *Assume a monad T and endofunctor G on a category $\mathbf{C}$, with an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ between them. If G has a final coalgebra $\zeta\colon Z \xrightarrow{\cong} GZ$ in $\mathbf{C}$, then Z carries a T-algebra structure obtained by finality, as on the left below. The map ζ then forms a map of algebras as on the right, which is the final coalgebra for the lifted functor $\widehat{G}\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$.*

$$\begin{tikzcd} GTZ \arrow[r, dashed, "G(\alpha)"] & GZ \\ TZ \arrow[u, "\rho\circ T(\zeta)"] \arrow[r, dashed, "\alpha"'] & Z \arrow[u, "\cong"', "\zeta"] \end{tikzcd} \qquad \begin{pmatrix} TZ \\ \downarrow\alpha \\ Z \end{pmatrix} \xrightarrow[\cong]{\zeta} \widehat{G}\begin{pmatrix} TZ \\ \downarrow\alpha \\ Z \end{pmatrix} = \begin{pmatrix} T(GZ) \\ \downarrow G(\alpha)\circ\rho \\ GZ \end{pmatrix}$$

Proof. We leave it to the reader to verify that α is a T-algebra. By construction of α, ζ is a map of algebras $\alpha \to \widehat{G}(\alpha)$. Suppose for an arbitrary algebra $b\colon TY \to Y$ we have a $\widehat{G}$-coalgebra $c\colon b \to \widehat{G}(b)$. Then $c\colon Y \to GY$ satisfies $G(b) \circ \rho \circ T(c) = c \circ b$. By finality in $\mathbf{C}$ there is a unique map $f\colon Y \to Z$ with $\zeta \circ f = G(f) \circ c$. This f is then the unique map $b \to \alpha$ in $\mathcal{EM}(T)$. $\square$

At this stage we can describe the first form of extension semantics, which we will call $\mathcal{EM}$-extension semantics, since it depends on an $\mathcal{EM}$-law.

Definition 5 **($\mathcal{EM}$-extension).** *Assume an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ and suppose that the final G-coalgebra $Z \xrightarrow{\cong} GZ$ exists. It can be lifted to a final $\widehat{G}$-coalgebra by Proposition 4. Hence one obtains "extension" semantics $X \to Z$ for a coalgebra $c\colon X \to GTX$ via the following three steps.*

1. *Transform c into a $\widehat{G}$-coalgebra $\mathcal{F}_{\mathcal{EM}}(c)$, via Lemma 3;*
2. *Get the resulting $\widehat{G}$-coalgebra map $TX \to Z$ by finality;*
3. *Precompose this map with the unit, yielding $X \to TX \to Z$.*

The next two sections will introduce examples of $\mathcal{EM}$-laws. Here, we briefly look at $\mathcal{K}\ell$-laws. The following result, from [12], shows that these $\mathcal{K}\ell$-laws are quite common, namely for *commutative* monads and *simple polynomial functors*—defined with identity, constants, products $\times$, and coproducts $\coprod$ only.

Lemma 6. *Let $T\colon \mathbf{Sets} \to \mathbf{Sets}$ be a commutative monad, and $F\colon \mathbf{Sets} \to \mathbf{Sets}$ a simple polynomial functor. Then there is a (canonical) $\mathcal{K}\ell$-law $\lambda\colon FT \Rightarrow TF$.* $\square$

4 Deterministic automata

This section briefly describes deterministic automata as coalgebras, recalls the final coalgebra, and introduces an associated $\mathcal{EM}$-law.

For arbitrary sets A, B there is an endofunctor $B \times (-)^A\colon \mathbf{Sets} \to \mathbf{Sets}$. Its coalgebras $\phi = \langle \phi_o, \phi_i\rangle\colon X \to B \times X^A$ are deterministic (Moore) automata. The map $\phi_o\colon X \to B$ describes the immediate output. The map $\phi_i\colon X \to X^A$ is the transition function, mapping a state $x \in X$ and an input $a \in A$ to a successor state $\phi_i(x)(a) \in X$. For the special case $B = 2 = \{0, 1\}$ it tells of a state whether it is final or not. The following result is so standard that we omit the proof.

Lemma 7. *The final coalgebra of the functor $B \times (-)^A$ on $\mathbf{Sets}$ is given by the set of functions $B^{A^\star}$, with structure:*

$$B^{A^\star} \xrightarrow{\quad \zeta = \langle \zeta_o, \zeta_i\rangle \quad} B \times \big(B^{A^\star}\big)^A$$

defined via the empty sequence $\langle\rangle \in A^\star$ and via prefixing $a \cdot \sigma$ of $a \in A$ to $\sigma \in A^\star$:

$$\zeta_o(t) \;=\; t(\langle\rangle) \qquad \text{and} \qquad \zeta_i(t)(a) \;=\; \lambda\sigma \in A^\star.\, t(a \cdot \sigma). \qquad \square$$

This result captures the paradigm of trace semantics: a state $x \in X$ of an arbitrary coalgebra $X \to B \times X^A$ has a behaviour in $B^{A^\star}$ that maps a trace-as-word of inputs in $A^\star$ to an output in B. In the sequel we consider the special case where the output set is the free algebra $T(B)$, for a monad T. In the next result we show that we then get a distributive law. We apply this result only for the category $\mathbf{Sets}$. But it will be formulated more generally, using a strong monad on a Cartesian closed category. This strength is automatic for any monad on $\mathbf{Sets}$.

Lemma 8. *Let T be a strong monad on a Cartesian closed category $\mathbf{C}$ and let $A, B \in \mathbf{C}$ be arbitrary objects. Consider the associated "machine" endofunctor $M = T(B) \times (-)^A$ on $\mathbf{C}$. Then there is an $\mathcal{EM}$-law $\rho\colon TM \Rightarrow MT$ given by:*

$$\rho_X = \Big(T\big(T(B) \times X^A\big) \xrightarrow{\langle T(\pi_1), T(\pi_2)\rangle} T^2(B) \times T(X^A) \xrightarrow{\mu \times \mathrm{st}} T(B) \times T(X)^A\Big).$$

This $\mathcal{EM}$-law can be defined slightly more generally, not with a free algebra $T(B)$ as output, but with an arbitrary algebra. But in fact, all our examples involve free algebras.

Proof. This follows directly from the properties of strength st. $\square$

The resulting lifting $\widehat{M}\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$ sends an algebra $a\colon TX \to X$ to the algebra $TMX \to MX$ given by:

$$T\big(T(B) \times X^A\big) \xrightarrow{\langle T(\pi_1), T(\pi_2)\rangle} T^2(B) \times T(X^A) \xrightarrow{\mu \times (a^A \circ \mathrm{st})} T(B) \times X^A$$

Proposition 4, in combination with Lemma 7, says that the final M-coalgebra $T(B)^{A^\star}$ carries a T-algebra structure that forms the final $\widehat{M}$-coalgebra in $\mathcal{EM}(T)$. With a bit of effort one shows that this algebra on $T(B)^{A^\star}$ is given by:

$$\alpha = \Big(T\big(T(B)^{A^\star}\big) \xrightarrow{\text{st}} \big(T^2(B)\big)^{A^\star} \xrightarrow{\mu^{A^\star}} T(B)^{A^\star}\Big)$$

Then $\zeta\colon \alpha \xrightarrow{\cong} \widehat{M}(\alpha)$ is the final $\widehat{M}$-coalgebra, by Proposition 4.

5 Examples of $\mathcal{EM}$-extension semantics

This section describes two examples of $\mathcal{EM}$-laws, one familiar one in the context of non-deterministic automata, and one new one for simple Segala systems.

5.1 Non-deterministic automata, in $\mathcal{EM}$-style

We briefly restate the non-deterministic automaton example from [25], but this time using the general constructions developed so far. A non-deterministic automaton is understood here as a coalgebra $c\colon X \to 2 \times (\mathcal{P}X)^A$, which is of the form $X \to G(TX)$, where G is the functor $2 \times (-)^A$ and T is the powerset monad $\mathcal{P}$ on $\mathbf{Sets}$.

Since $2 = \{0, 1\}$ is the (carrier of the) free algebra $\mathcal{P}(1)$ on the singleton set $1 = \{*\}$, Lemma 8 applies. It yields an $\mathcal{EM}$-law with components $\rho = \langle \rho_1, \rho_2 \rangle\colon \mathcal{P}(2 \times X^A) \to 2 \times (\mathcal{P}X)^A$, given by:

$$\begin{cases} \rho_1(U) = 1 \iff \exists h \in X^A.\, \langle 1, h\rangle \in U \\ x \in \rho_2(U)(a) \iff \exists \langle b, h\rangle \in U.\, h(a) = x. \end{cases}$$

Lemma 3 yields a coalgebra $\mathcal{F}_{\mathcal{EM}}(c) = \langle \phi_o, \phi_i \rangle\colon \mathcal{P}(X) \to 2 \times (\mathcal{P}X)^A$ in the category $\mathcal{EM}(\mathcal{P})$, where:

$$\begin{cases} \phi_o(U) = 1 \iff \exists x \in U.\, \pi_1 c(x) = 1 \\ y \in \phi_i(U)(a) \iff \exists x \in U.\, y \in \pi_2 c(x)(a). \end{cases}$$

By Proposition 4 the final G-coalgebra $2^{A^\star} = \mathcal{P}(A^\star)$ of languages is also final for the functor $\widehat{G}$ on $\mathcal{EM}(\mathcal{P})$. Hence we get a map $\mathcal{P}(X) \to \mathcal{P}(A^\star)$ by finality. Applying this map to the singleton set $\{x\} \in \mathcal{P}(X)$ yields the set of words that are accepted in the state $x \in X$. Thus, the $\mathcal{EM}$-semantics from Definition 5 yields the trace semantics for a non-deterministic automaton $c\colon X \to 2 \times (\mathcal{P}X)^A$, via the language accepted in a state.

5.2 Simple Segala systems, in $\mathcal{EM}$-style

Next, we consider simple Segala systems as a non-trivial example of $\mathcal{EM}$-extension semantics, which was not considered in [25]. These systems are also called simple probabilistic automata [23], Markov decision processes, or probabilistic labeled transition systems (LTSs). They are coalgebras of the form $c\colon X \to \mathcal{P}(A \times \mathcal{D}X)$, mixing probability and non-determinism. In a recent paper [9], with ideas appearing already in [21,10,6,13], it has been recognized that it might be useful for verification purposes to transform them into so-called distribution LTSs, *i.e.* into LTSs with state space $\mathcal{D}X$ of so-called uncertain or belief states.

Given a simple Segala system $c\colon X \to \mathcal{P}(A \times \mathcal{D}X)$, we denote by $c^\sharp\colon \mathcal{D}X \to \mathcal{P}(A \times \mathcal{D}X)$ its distribution LTS from [9]. It is defined by

$$\begin{aligned} c^\sharp(\varphi) &= \{\langle a, \psi\rangle \mid \exists x \in \mathrm{supp}(\varphi).\, \langle a, \psi\rangle \in c(x)\} \\ &= \big(\mu^{\mathcal{P}} \circ \mathcal{P}(c) \circ \mathrm{supp}\,\big)(\varphi). \end{aligned} \tag{7}$$

where $\mathrm{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$. In the usual notation for transitions, this means that for $\varphi, \psi \in \mathcal{D}X$,

$$\varphi \xrightarrow{a}_{c^\sharp} \psi \qquad \text{if and only if} \qquad \exists x \in \mathrm{supp}(\varphi).\, x \xrightarrow{a}_{c} \psi.$$

Here, we capture this situation via a distributive law $\rho\colon \mathcal{D}\mathcal{P}(A \times (-)) \Longrightarrow \mathcal{P}(A \times \mathcal{D})$. It is an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ for $T = \mathcal{D}$ and $G = \mathcal{P}(A \times -)$ with the property that the $\mathcal{EM}$-extension from Lemma 3 turns a simple Segala system into its distribution LTS, see Proposition 11 below. Explicitly, for a distribution $\Phi \in \mathcal{D}\mathcal{P}(A\times X)$, we define

$$\begin{aligned} \rho(\Phi) &= \{\langle a, \delta_x\rangle \mid \exists U \in \mathrm{supp}(\Phi).\, \langle a, x\rangle \in U\} \\ &= \textstyle\bigcup_{U \in \mathrm{supp}(\Phi)} \{\langle a, \delta_x\rangle \mid \langle a, x\rangle \in U\} \\ &= \big(\mu^{\mathcal{P}} \circ \mathcal{P}^2(\mathrm{id} \times \eta^{\mathcal{D}}) \circ \mathrm{supp}\,\big)(\Phi). \end{aligned} \tag{8}$$

where δ_x is the Dirac distribution assigning probability 1 to x, *i.e.* $\delta_x = \eta(x)$. The fact that ρ is an $\mathcal{EM}$-law is an instance of the following result—using that the support forms a map of monads $\mathrm{supp}\colon \mathcal{D} \Rightarrow \mathcal{P}$.

Lemma 9. *Each map of monads $\sigma\colon T \Rightarrow S$ induces an $\mathcal{EM}$-law*

$$TS(A \times -) \overset{\rho}{\Longrightarrow} S(A \times T(-))$$

of the monad T *over the* functor $S(A \times -)$. *The components of ρ are given by:*

$$\rho_X = \Big(TS(A\times X) \xrightarrow{\sigma} S^2(A\times X) \xrightarrow{S^2(id\times\eta^T)} S^2(A \times TX) \xrightarrow{\mu^S} S(A\times TX)\Big). \quad \square$$

Remark 10. As emphasized, ρ in (8) is a distributive law between the monad $\mathcal{D}$ and the *functor* $\mathcal{P}(A \times -)$. In particular for $A = 1$ it is a distributive law between the monad $\mathcal{D}$ and the *functor* $\mathcal{P}$. An important but subtle point is that ρ is not a distributive law between the $\mathcal{D}$ and the *monad* $\mathcal{P}$. There is no such distributive law as shown in [27]. The unit law for the powerset monad, required for a monad distributive law, does not hold for ρ with $A = 1$: $\rho \circ \mathcal{D}(\eta^{\mathcal{P}}) \neq \eta^{\mathcal{P}}\mathcal{D}$. Nevertheless, one can distribute probability over non-determinism, via ρ. The construction provides a non-standard LTS semantics to simple Segala systems, by first lifting these systems to distributions.

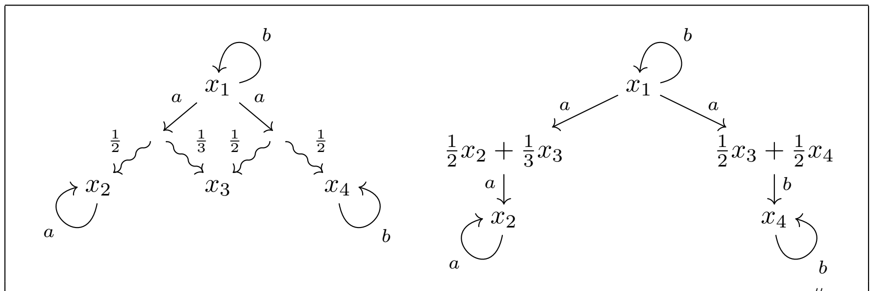
The distribution LTS in (7) from [9] is an instance of our general framework.

Proposition 11. *Given a simple Segala system $c\colon X \to \mathcal{P}(A \times \mathcal{D}X)$, its $\mathcal{EM}$-extension $\mathcal{F}_{\mathcal{EM}}(c)$ from Lemma 3, obtained via the $\mathcal{EM}$-law ρ from* (8)*, is the same as the coalgebra $c^{\#}\colon \mathcal{D}X \to \mathcal{P}(A \times \mathcal{D}X)$ described in Equation* (7)*.*

Proof. By a straightforward calculation:

$$\begin{aligned}
\mathcal{F}_{\mathcal{EM}}(c) &\overset{(6)}{=} \mathcal{P}(\mathrm{id} \times \mu^{\mathcal{D}}) \circ \rho \circ \mathcal{D}(c) \\
&\overset{(8)}{=} \mathcal{P}(\mathrm{id} \times \mu^{\mathcal{D}}) \circ \mu^{\mathcal{P}} \circ \mathcal{P}^2(\mathrm{id} \times \eta^{\mathcal{D}}) \circ \mathrm{supp} \circ \mathcal{D}(c) \\
&= \mathcal{P}(\mathrm{id} \times \mu^{\mathcal{D}}) \circ \mathcal{P}(\mathrm{id} \times \eta^{\mathcal{D}}) \circ \mu^{\mathcal{P}} \circ \mathrm{supp} \circ \mathcal{D}(c) \\
&= \mu^{\mathcal{P}} \circ \mathcal{P}(c) \circ \mathrm{supp} \\
&\overset{(7)}{=} c^{\sharp}. \qquad \square
\end{aligned}$$

The following is a simple example of a non-determinization.


x_1
x_2
x_3
x_4
a
b
$\frac{1}{2}$
$\frac{1}{3}$
$\frac{1}{2}x_2 + \frac{1}{3}x_3$
$\frac{1}{2}x_3 + \frac{1}{2}x_4$

simple Segala system c part of its non-determinization $c^{\#}$

Remark 12. Definition 5 describes how $\mathcal{EM}$-extension semantics arises in presence of a final coalgebra. This does not directly apply in this situation because the (ordinary) powerset functor does not have a final coalgebra, for cardinality reasons. But if we restrict ourselves to finite subsets and distributions with finite support, there is still a map of monads $\mathcal{D}_{\mathrm{fin}} \Rightarrow \mathcal{P}_{\mathrm{fin}}$, so that we get an $\mathcal{EM}$-law $\mathcal{D}_{\mathrm{fin}}\mathcal{P}_{\mathrm{fin}}(A\times -) \Rightarrow \mathcal{P}_{\mathrm{fin}}(A\times\mathcal{D}_{\mathrm{fin}})$ by Lemma 9. For a "finitely branching" Segala system $c\colon X \to \mathcal{P}_{\mathrm{fin}}(A \times \mathcal{D}_{\mathrm{fin}}X)$ one obtains semantics $X \to Z$, where $Z \xrightarrow{\cong} \mathcal{P}_{\mathrm{fin}}(A \times Z)$ is the final coalgebra.

6 $\mathcal{K}\ell$-Extension semantics

In a next step we wish to develop extension semantics not only for coalgebras of the form $X \to G(TX)$, on the left in (1), but also for coalgebras $X \to T(FX)$, on the right in (1). This turns out to be more complicated. First of all, the lifting $\mathcal{F}_{\mathcal{K}\ell}\colon \mathbf{CoAlg}(TF) \to \mathbf{CoAlg}(\widehat{F})$ in Lemma 3 is not interesting in the current setting because it does not involve a state space extension $X \mapsto T(X)$.

The next thought might be to translate coalgebras $X \to T(FX)$ into coalgebras $X \to G(TX)$, via a distributive law $TF \Rightarrow GT$. This results in functors

$$\mathbf{CoAlg}(TF) \longrightarrow \mathbf{CoAlg}(GT) \xrightarrow{\text{Lemma 3}} \mathbf{CoAlg}(\widehat{G})$$

However, coalgebras $X \to T(FX)$ are usually considered as coalgebras $X \to \widehat{F}X$ of the lifted functor $\widehat{F}$ on the Kleisli category $\mathcal{K}\ell(T)$. Therefore, our aim is to obtain a functor $\mathbf{CoAlg}(\widehat{F}) \to \mathbf{CoAlg}(\widehat{G})$. This will be described below.

Recall from Section 2 that there is a comparison functor $E\colon \mathcal{K}\ell(T) \to \mathcal{EM}(T)$. In this section we show how it can be lifted to coalgebras. We consider the following situation.

1. A monad $T\colon \mathbf{C} \to \mathbf{C}$ on a category $\mathbf{C}$.
2. An endofunctor $F\colon \mathbf{C} \to \mathbf{C}$ with a $\mathcal{K}\ell$-law $\lambda\colon FT \Rightarrow TF$, leading to a lifting $\widehat{F}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ to T's Kleisli category, via Proposition 1.
3. Another endofunctor $G\colon \mathbf{C} \to \mathbf{C}$, but this time with an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$, yielding a lifting $\widehat{G}\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$ to the category of T-algebras.
4. An "extension" natural transformation $\mathfrak{e}\colon TF \Rightarrow GT$ that connects the $\mathcal{K}\ell$- and $\mathcal{EM}$-laws via the following two commuting diagrams.

$$\begin{tikzcd}
TFT \arrow[r, "T(\lambda)"] \arrow[d, "\mathfrak{e}T"'] & T^2F \arrow[r, "\mu F"] & TF \arrow[d, "\mathfrak{e}"] \\
GT^2 \arrow[rr, "G\mu"] & & GT
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2F \arrow[rr, "\mu F"] \arrow[d, "T\mathfrak{e}"'] & & TF \arrow[d, "\mathfrak{e}"] \\
TGT \arrow[r, "\rho T"] & GT^2 \arrow[r, "G\mu"] & GT
\end{tikzcd} \tag{9}$$

When such $\mathfrak{e}$ is an isomorphism, it forms a "commuting pair of endofunctors" from [1].

Theorem 13. *Assuming the above points 1–4, there is a lifting $\widehat{E}$ of the extension functor E in:*

$$\begin{tikzcd}
\mathbf{CoAlg}(\widehat{F}) \arrow[r, "\widehat{E}"] \arrow[d] & \mathbf{CoAlg}(\widehat{G}) \arrow[d] \\
\mathcal{K}\ell(T) \arrow[r, "E"] \arrow[loop left, "\widehat{F}"] & \mathcal{EM}(T) \arrow[loop right, "\widehat{G}"]
\end{tikzcd}$$

This functor $\widehat{E}$ is automatically faithful; and it is also full if the extension natural transformation $\mathfrak{e}\colon TF \Rightarrow GT$ consists of monomorphisms.

Proof. We define the functor $\widehat{E}\colon \mathbf{CoAlg}(\widehat{F}) \to \mathbf{CoAlg}(\widehat{G})$ by:

$$\begin{aligned}
\big(X \xrightarrow{c} \widehat{F}X\big) &\longmapsto \big(TX \xrightarrow{T(c)} T^2FX \xrightarrow{\mu} TFX \xrightarrow{\mathfrak{e}} GTX\big) \\
f &\longmapsto E(f) = \mu \circ T(f).
\end{aligned}$$

We need to show that $\widehat{E}(c)$ is a map of algebras $E(X) = \mu_X \to \widehat{G}(\mu_X) = \widehat{G}(EX)$ in:

$$\begin{pmatrix} T^2X \\ \downarrow \mu_X \\ TX \end{pmatrix} \xrightarrow{\widehat{E}(c)} \begin{pmatrix} TGTX \\ \downarrow G(\mu_X)\circ\rho \\ GTX \end{pmatrix} = \widehat{G}\begin{pmatrix} T^2X \\ \downarrow \mu_X \\ TX \end{pmatrix}$$

But this is simply the above requirement 4.

Assume f is a map of $\widehat{F}$-coalgebras, from $c\colon X \to \widehat{F}X$ to $d\colon Y \to \widehat{F}Y$. That is, $f\colon X \to TY$, $c\colon X \to TFX$ and $d\colon Y \to TFY$ satisfy:

$$\mu \circ T(d) \circ f \;=\; \mu \circ T(\lambda \circ F(f)) \circ c. \tag{10}$$

Then $E(f) = \mu \circ T(f)$ is a map of coalgebras $\widehat{E}(c) \to \widehat{E}(d)$, again by requirement 4:

$$\begin{aligned}
\widehat{E}(d) \circ E(f) &= \mathfrak{e}_Y \circ \mu \circ T(d) \circ \mu \circ T(f) \\
&= \mathfrak{e}_Y \circ \mu \circ \mu \circ T(T(d) \circ f) \\
&= \mathfrak{e}_Y \circ \mu \circ T(\mu \circ T(d) \circ f) \\
&\overset{(10)}{=} \mathfrak{e}_Y \circ \mu \circ T(\mu \circ T(\lambda \circ F(f)) \circ c) \\
&= \mathfrak{e}_Y \circ \mu \circ \mu \circ T^2(\lambda \circ F(f)) \circ T(c) \\
&= \mathfrak{e}_Y \circ \mu \circ T(\lambda \circ F(f)) \circ \mu \circ T(c) \\
&\overset{(9)}{=} G(\mu) \circ \mathfrak{e}_{TY} \circ TF(f) \circ \mu \circ T(c) \\
&= G(\mu \circ T(f)) \circ \mathfrak{e}_X \circ \mu \circ T(c) \\
&= \widehat{G}(Ef) \circ \widehat{E}(c).
\end{aligned}$$

Clearly, the functor $\widehat{E}$ is faithful: if $f, g \colon X \to TY$ satisfy $\widehat{E}(f) = E(f) = E(g) = \widehat{E}(g)$, then $f = E(f) \circ \eta = E(g) \circ \eta = g$.

Now assume the components $\mathfrak{e}_X \colon TFX \to GTX$ are monomorphisms in $\mathbf{C}$. Towards fulness of $\widehat{E}$, let $h \colon \widehat{E}(c) \to \widehat{E}(d)$ be a morphism in $\mathbf{CoAlg}(\widehat{G})$. It is a map $h \colon TX \to TY$ that is both a map of algebras and of coalgebras, so:

$$\begin{tikzcd}
T^2X \arrow[r, "T(h)"] \arrow[d, "\mu"'] & T^2Y \arrow[d, "\mu"] & & GTX \arrow[r, "G(h)"] & GTY \\
TX \arrow[r, "h"'] & TY & & TX \arrow[u, "\mathfrak{e}_X \circ \mu \circ T(c)"] \arrow[r, "h"'] & TY \arrow[u, "\mathfrak{e}_Y \circ \mu \circ T(d)"']
\end{tikzcd} \qquad (11)$$

We now take $f = h \circ \eta \colon X \to T(Y)$ and need to show that $\widehat{E}(f) = E(f) = h$ and that it is a map of $\widehat{F}$-coalgebras $c \to d$. The first is easy since:

$$E(f) \;=\; \mu \circ T(h \circ \eta) \;=\; h \circ \mu \circ T(\eta) \;=\; h.$$

In order to show that f is a map of coalgebras we use that $\mathfrak{e}$ consists of monos, in:

$$\begin{aligned}
\mathfrak{e} \circ \big(d \odot f\big) &= \mathfrak{e} \circ \mu \circ T(d) \circ f \\
&= \mathfrak{e} \circ \mu \circ T(d) \circ h \circ \eta \\
&\overset{(11)}{=} G(h) \circ \mathfrak{e} \circ \mu \circ T(c) \circ \eta \\
&= G(h) \circ \mathfrak{e} \circ \mu \circ \eta \circ c \\
&= G(h) \circ \mathfrak{e} \circ c \\
&= G(h) \circ G(\mu \circ T(\eta)) \circ \mathfrak{e} \circ c \\
&\overset{(11)}{=} G(\mu) \circ GT(h \circ \eta) \circ \mathfrak{e} \circ c \\
&= G(\mu) \circ \mathfrak{e} \circ TF(f) \circ c \\
&\overset{(9)}{=} \mathfrak{e} \circ \mu \circ T(\lambda \circ F(f)) \circ c \\
&= \mathfrak{e} \circ \big(\widehat{F}(f) \odot c\big).
\end{aligned}$$

□

On a more abstract level, what the previous result does is lift $\mathfrak{e}\colon TF \Rightarrow GT$ to a natural transformation $\widehat{\mathfrak{e}}\colon E\widehat{F} \Rightarrow \widehat{G}E$. In this way we can also define the functor $\widehat{E}\colon \mathbf{CoAlg}(\widehat{F}) \to \mathbf{CoAlg}(\widehat{G})$ by:

$$\big(X \xrightarrow{c} \widehat{F}X\big) \longmapsto \big(EX \xrightarrow{\widehat{\mathfrak{e}}\circ E(c)} \widehat{G}(EX)\big) \qquad \text{and} \qquad f \longmapsto E(f) = \mu \circ T(f).$$

We can now define extension semantics for coalgebras $X \to T(FX)$, analogously to Definition 5.

Definition 14 **($\mathcal{K}\ell$-extension).** *Assume, in addition to the points 1–4 from the beginning of this section, that the functor G has a final coalgebra $Z \xrightarrow{\cong} G(Z)$. By Proposition 4 it lifts to a final $\widehat{G}$-coalgebra. Thus, for a coalgebra $c\colon X \to T(FX)$ one obtains its $\mathcal{K}\ell$-extension semantics in three steps, like in Definition 5:*

1. *Transform c into a $\widehat{G}$-coalgebra $\widehat{E}(c)$, by Theorem 13;*
2. *Obtain a map $TX \to Z$ in $\mathcal{EM}(T)$ by finality;*
3. *Get $X \to Z$ by precomposition with the unit $X \to TX$.*

We conclude this section by showing how the "Kleisli" trace semantics from [12] fits in the current setting. Thus, we assume in the situation of Definition 14 that the functor F has an initial algebra $\iota\colon F(W) \xrightarrow{\cong} W$.

Proposition 15. *Assume the map $\mathcal{F}(\iota^{-1})\colon W \to \widehat{F}(W)$ is final $\widehat{F}$-coalgebra. Each coalgebra $c\colon X \to TF(X)$ then gives rise to a "Kleisli" trace map in the Kleisli category (as in [12]), namely:*

$$\begin{tikzcd}
\widehat{F}X \arrow[r, dashed] & \widehat{F}(W) \\
X \arrow[u, "c"] \arrow[r, dashed, "\mathrm{tr}_{\mathcal{K}\ell}(c)"] & W \arrow[u, "\cong"', "\mathcal{F}(\iota^{-1})"]
\end{tikzcd}$$

When we apply the functor $\widehat{E}$ from Theorem 13 to this diagram we get the rectangle on the left in:

$$\begin{tikzcd}
& \widehat{G}(TX) \arrow[r] & \widehat{G}(TW) \arrow[r, dashed] & \widehat{G}(Z) \\
X \arrow[r, "\eta"] \arrow[rr, bend right, "\mathrm{tr}_{\mathcal{K}\ell}(c)"'] & TX \arrow[u, "\widehat{E}(c)"] \arrow[r, "\widehat{E}(\mathrm{tr}_{\mathcal{K}\ell}(c))"] & TW \arrow[u, "\cong"', "\widehat{E}(\mathcal{F}(\iota^{-1}))"] \arrow[r, dashed] & Z \arrow[u, "\cong"', "\zeta"]
\end{tikzcd} \tag{12}$$

The resulting $\mathcal{K}\ell$-extension semantics map $X \to Z$ is then the $\mathcal{K}\ell$-extension semantics of the final $\widehat{F}$-coalgebra $\mathcal{F}(\iota^{-1})\colon W \to \widehat{F}(W)$, precomposed with the Kleisli trace semantics $\mathrm{tr}_{\mathcal{K}\ell}(c)$. □

7 Determinization and trace semantics

In this section we specialize the $\mathcal{K}\ell$-extension framework developed so far to deterministic automata, leading to a novel definition of trace semantics, namely via $\mathcal{K}\ell$-extension semantics. Several illustrations will be given, including the standards ones from the trace semantics of [12].

Definition 16. *Assume a monad T and an endofunctor F on the category* $\mathbf{Sets}$*, with a* $\mathcal{K}\ell$*-law* $\lambda\colon FT \Rightarrow TF$ *between them. Assume further sets A, B, leading to endofunctor* $M = T(B) \times (-)^A$*, which comes with an* $\mathcal{EM}$*-law* $\rho\colon TM \Rightarrow MT$ *like in Lemma 8. Finally, we assume an extension law* $\mathfrak{e}\colon TF \Rightarrow MT = T(B) \times T(-)^A$ *like in the previous section. In this situation,*

– *the determinization of a coalgebra* $c\colon X \to TFX$ *is the* $\widehat{M}$*-coalgebra* $\widehat{E}(c)$ *in the category of algebras* $\mathcal{EM}(T)$ *given by:*

$$T(X) \xrightarrow{T(c)} T^2FX \xrightarrow{\mu} TFX \xrightarrow{\mathfrak{e}} T(B) \times T(X)^A$$

Thus, determinization turns the coalgebra c on X into a deterministic automaton on TX.

– *the trace map* $\mathrm{tr}(c)\colon X \to T(B)^{A^\star}$ *of such a coalgebra c is obtained via the unique coalgebra map* $T(X) \to T(B)^{A^\star}$ *that arises by finality in* $\mathcal{EM}(T)$ *in:*

$$\begin{tikzcd}
 & T(B) \times T(X)^A = \widehat{M}(TX) \arrow[r, dashed] & \widehat{M}\big(T(B)^{A^\star}\big) = T(B) \times \big(T(B)^{A^\star}\big)^A \\
X \arrow[r, "\eta"] \arrow[rr, bend right, "\mathrm{tr}(c)"'] & TX \arrow[u, "\widehat{E}(c)"] \arrow[r, dashed] & T(B)^{A^\star} \arrow[u, "\cong"', "\zeta"]
\end{tikzcd}$$

7.1 Non-deterministic automata, in $\mathcal{K}\ell$-style

In Subsection 5.1 we have seen how to obtain traces for non-deterministic automata via determinization (like in [25]). Now we re-describe the same example in $\mathcal{K}\ell$-style, via the isomorphisms (2). In essence we translate the "Kleisli" trace semantics approach of [12] into the current setting, like in Proposition 15. Thus we start with a non-deterministic automata as coalgebras of the form $c\colon X \to \mathcal{P}(1 + A \times X)$, for a fixed set of labels A and $1 = \{*\}$ modeling termination. A state $x \in X$ of such a coalgebra is final if and only if $* \in c(x)$. In this case the monad is the powerset monad $\mathcal{P}$ and the functor is $F = 1 + A \times (-)$ with finite lists $A^\star$ as initial algebra. We recall that the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ is the category $\mathbf{Rel}$ of sets and relations between them, and the category $\mathcal{EM}(\mathcal{P})$ is the category $\mathbf{CL}$ of complete lattices with join-preserving maps.

The functor F lifts to $\widehat{F}\colon \mathbf{Rel} \to \mathbf{Rel}$ by Lemma 6. We instantiate M for the set $B = 1$ and the powerset monad, and get $M = 2 \times (-)^A$ since $\mathcal{P}(1) \cong 2$. Then there is an extension law $\mathfrak{e}\colon \mathcal{P}(1 + A \times (-)) \Rightarrow 2 \times \mathcal{P}^A$ with

$$\mathfrak{e}(U) \;=\; \langle o(U), \lambda a. \{x \mid \langle a, x\rangle \in U\}\rangle \qquad \text{where} \qquad o(U) \;=\; \begin{cases} 1 & \text{if } * \in U \\ 0 & \text{if } * \notin U. \end{cases}$$

One can easily check that $\mathfrak{e}$ is injective (it is actually an isomorphism) and that it satisfies the requirements (9) for an extension law.

Given a coalgebra $c\colon X \to \mathcal{P}(1 + A \times X)$ its determinization is the deterministic automaton $\widehat{E}(c)\colon \mathcal{P}(X) \to 2 \times \mathcal{P}(X)^A$ with subsets $V \subseteq X$ as states, given by $\widehat{E}(c) = \mathfrak{e} \circ \mu \circ \mathcal{P}(c)$ or, more concretely,

$$\widehat{E}(c)(V) \;=\; \left\langle o\left(\bigcup_{x\in V} c(x)\right),\; \lambda a.\ \bigcup_{x \in V} \{y \mid \langle a, y\rangle \in c(x)\}\right\rangle.$$

This determinization coincides with the well-known subset construction [14]. The trace map $\mathrm{tr}(c)$ associates with each state of the original non-deterministic automaton the language it recognizes.

The dashed map $TW \dashrightarrow Z$ in (12) in Proposition 15 is the obvious isomorphism $\mathcal{P}(A^\star) \xrightarrow{\cong} 2^{A^\star}$ for non-deterministic automata. Therefore, Proposition 15 yields that "Kleisli" trace and trace via $\mathcal{K}\ell$-extension semantics coincide, *i.e.* $\mathrm{tr}(c) = \mathrm{tr}_{\mathcal{K}\ell}(c)$ for any non-deterministic automaton $c\colon X \to \mathcal{P}(1 + A \times X)$.

7.2 Generative probabilistic systems

Next, we consider generative probabilistic systems with explicit termination. They also fit in the "Kleisli" trace approach of [12]. Their determinization was introduced by the last two authors in [26] and motivated us to look at the framework of the present paper.

Generative systems are coalgebras for the functor $\mathcal{D}(1 + A \times (-))$ where $\mathcal{D}$ is the subprobability distribution monad. The functor $F = 1 + A \times (-)$ lifts to $\widehat{F}\colon \mathcal{K}\ell(\mathcal{D}) \to \mathcal{K}\ell(\mathcal{D})$ by Lemma 6. The category $\mathcal{EM}(\mathcal{D})$ is the category $\mathbf{PCA}$ of positive convex algebras and convex maps [11]. We instantiate the functor M to $B = 1$ and the subprobability distribution monad $\mathcal{D}$, and get $M = [0,1] \times (-)^A$ since $\mathcal{D}(1) \cong [0,1]$. Define $\mathfrak{e}\colon \mathcal{D}(1 + A \times (-)) \Rightarrow [0,1] \times \mathcal{D}^A$ by

$$\mathfrak{e}(\xi) \;=\; \langle\, \xi(*),\, \lambda a.\, \lambda y.\, \xi(a,y)\,\rangle.$$

It is not difficult to check that $\mathfrak{e}$ is an extension law and that it is injective.

Given a coalgebra $c\colon X \to \mathcal{D}(1 + A \times X)$ its determinization is the deterministic automaton $\widehat{E}(c)\colon \mathcal{D}(X) \to [0,1] \times \mathcal{D}(X)^A$ with states $\mathcal{D}(X)$, given by $\widehat{E}(c) = \mathfrak{e} \circ \mu \circ \mathcal{D}(c)$ or, more concretely,

$$\widehat{E}(c)(\xi) \;=\; \big\langle\, \textstyle\sum_{x \in X} \xi(x) \cdot c(x)(*),\; \lambda a.\, \lambda y.\, \sum_{x \in X} \xi(x) \cdot c(x)(a,y)\,\big\rangle\,.$$

We show an example of such determinization: the automaton on the right is part of the determinization of the one on the left. The full determinization is an infinite automaton. We show the accessible part when starting from the state $\eta(x_1)$, the Dirac distribution of x_1, and we denote the distributions by formal sums. We omit zero output probabilities.

$$\begin{tikzcd}
 & x_1 \arrow[dl, "{a,\frac{1}{2}}"'] \arrow[dr, "{a,\frac{1}{3}}"] & \\
x_2 \arrow[d, "{b,\frac{1}{4}}"'] & & x_3 \arrow[d, "{c,\frac{1}{2}}"'] \\
x_4 \arrow[d] & & x_5 \arrow[d] \\
1 & & 1
\end{tikzcd}
\qquad
\begin{tikzcd}
 & x_1 \arrow[d, "a"] & \\
 & \frac{1}{2}x_2 + \frac{1}{3}x_3 \arrow[dl, "b"'] \arrow[dr, "c"] & \\
\frac{1}{8}x_4 \arrow[d] & & \frac{1}{6}x_5 \arrow[d] \\
\frac{1}{8} & & \frac{1}{6}
\end{tikzcd}$$

generative system c part of its determinization $\widehat{E}(c)$

The trace map $\mathrm{tr}(c)$ associates with each state of the original generative system a subprobability distribution on words, giving the probability to terminate with a given word starting from the given state. For instance, for state x_1 above, we have $\mathrm{tr}(c)(x_1) = \frac{1}{8}ab + \frac{1}{6}ac$.

The dashed map in (12) in Proposition 15 is in this case the inclusion map $\mathcal{D}(A^*) \to [0,1]^{A^*}$. Therefore, again, Proposition 15 yields that "Kleisli" trace and trace via extension semantics "coincide", *i.e.* $\mathrm{tr}(c)(x)(w) = \mathrm{tr}_{\mathcal{K}\ell}(c)(x)(w)$ for any generative system $c\colon X \to \mathcal{D}(1 + A \times X)$, any of its states $x \in X$, and any word $w \in A^*$. Moreover, it shows that the trace map $\mathrm{tr}(c)$ is not just any map from A^* to $[0,1]$, but a subprobability distribution on words. This coincidence was shown in [26] in concrete terms.

7.3 Weighted automata

The restrictions imposed on the monad in order for the trace semantics of [12] to work rule out several interesting monads, such as the multiset monads $\mathcal{M}_S$ (including the free vector space monad when S is a field), used for coalgebraic modeling of weighted systems. In this example, we show how the new framework allows us to deal with trace semantics for weighted systems using extension semantics. We consider the same functor $F = 1 + A \times (-)$ as before, with the multiset monad $\mathcal{M}_S$ over a semiring S. Recall that it maps a set X to the set of all finitely supported maps from X to S. Having finite support is crucial for $\mathcal{M}_S$ to be a monad, and one of the reasons why this monad does not fit in the "Kleisli" traces framework of [12]. Similar to probability distributions, we denote multisets by formal sums, now with coefficients in the semiring S. The Kleisli category $\mathcal{K}\ell(\mathcal{M}_S)$ is, for instance, the category of free commutative monoids when $S = \mathbb{N}$, and the category $\mathcal{EM}(\mathcal{M}_S)$ is the category of modules over S.

Coalgebras of the functor $\mathcal{M}_S(1 + A \times (-))$ are precisely weighted automata with weights over the set S. Given a coalgebra $c\colon X \to \mathcal{M}_S(1 + A \times X)$, each state $x \in X$ has an output weight $c(x)(*) \in S$ and an a-labelled transition into state y with weight $c(x)(\langle a, y\rangle) \in S$.

We instantiate the deterministic automaton functor M with $B = 1$ and the multiset monad $\mathcal{M}_S$, and get $M = S \times (-)^A$ since $\mathcal{M}_S(1) \cong S$. Then there is an extension natural transformation $\mathfrak{e}\colon \mathcal{M}_S(1 + A \times (-)) \Rightarrow S \times \mathcal{M}_S^A$ given, as for the subdistribution monad, by

$$\mathfrak{e}(\xi) \;=\; \langle\, \xi(*),\; \lambda a.\, \lambda x.\, \xi(\langle a, x\rangle)\,\rangle.$$

Again, $\mathfrak{e}$ satisfies all the conditions and it is injective.

For weighted automata, just like for generative systems, the determinization construction gives rise to a deterministic automaton with an infinite state-space even if the original automaton is finite. For a weighted automaton $c\colon X \to \mathcal{M}_S(1 + A \times X)$, the determinization $\widehat{E}(c)\colon \mathcal{M}_S(X) \to S \times \mathcal{M}_S(X)^A$ is given by

$$\widehat{E}(c)(\xi) \;=\; \langle\, \textstyle\sum_{x \in X} \xi(x) \cdot c(x)(*),\; \lambda a.\, \lambda y.\, \textstyle\sum_{x \in X} \xi(x) \cdot c(x)(a, y)\,\rangle$$

for a multiset $\xi\colon X \to S$ with finite support $\{x \in X \mid \xi(x) \neq 0\}$.

The following is a very simple example of determinization, for $S = \mathbb{N}$.

$$\begin{tikzcd}
 & x_1 \to 1 \arrow[dl, "{a,1}"'] \arrow[dr, "{a,2}"] & \\
x_2 \to 1 \arrow[loop below, "{a,1}"] & & x_3 \to 1 \arrow[loop below, "{a,1}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
x_1 \arrow[r] \arrow[d, "a"'] & 1 \\
x_2 + 2x_3 \to 3 \arrow[loop below, "a"] &
\end{tikzcd}$$

weighted automaton c part of its determinization $\widehat{E}(c)$

In general, the transitions of the determinization of the example weighted automaton c are given by $k_1x_1 + k_2x_2 + k_3x_3 \xrightarrow{a} (k_1 + k_2)x_2 + (2k_1 + k_3)x_3$ and the termination weight of a state $k_1x_1 + k_2x_2 + k_3x_3$ equals $k_1 + k_2 + k_3$.

The trace map $\mathrm{tr}(c) \colon X \to \mathbb{N}^{A^\star}$ associates with each state x of the original weighted automaton the weighted language that it accepts. For instance, in the example above, we have $\mathrm{tr}(c)(x_1)(\langle\rangle) = 1$ and $\mathrm{tr}(c)(x_1)(a^n) = 3$ for $n \geq 1$.

Remark 17. More specifically, we can consider weighted automata with weights over a field $\mathbb{F}$, which are coalgebras for the functor $\mathcal{M}_{\mathbb{F}}(1 + A \times (-))$, where $\mathcal{M}_{\mathbb{F}}$ is the free vector space monad. This monad is special: the Kleisli category $\mathcal{K}\ell(\mathcal{M}_{\mathbb{F}})$ and the category $\mathcal{EM}(\mathcal{M}_{\mathbb{F}})$ coincide, since each vector space has a basis. They are the category **Vec** of vector spaces and linear maps over $\mathbb{F}$. The determinization automaton $\widehat{E}(c)$ for a weighted automaton c coincides then with the *linear weighted automaton* of [5].

In the remainder of this section we consider the quantum walks example from [15], which can be described as a coalgebra $c \colon \mathbb{Z} + \mathbb{Z} \to \mathcal{M}_{\mathbb{C}}(\mathbb{Z} + \mathbb{Z})$, where the state space $\mathbb{Z}+\mathbb{Z}$ represents positions on a line $\mathbb{Z}$, with a direction (namely $\uparrow$ for the left component in $\mathbb{Z} + \mathbb{Z}$ and $\downarrow$ for the other, downwards direction). This description only involves the (unitary) transition function given by a superposition of left and right steps. Here we adapt this example a little bit so that we can compute the resulting probabilities via traces. We take the functor $F(X) = (\mathbb{Z} + \mathbb{Z}) + X$, and coalgebra $c \colon \mathbb{Z} + \mathbb{Z} \to \mathcal{M}_{\mathbb{C}}((\mathbb{Z} + \mathbb{Z}) + (\mathbb{Z} + \mathbb{Z})) = \mathcal{M}_{\mathbb{C}}(F(\mathbb{Z} + \mathbb{Z}))$ given by:

$$\begin{array}{rcl}
c(\uparrow k) & = & 1\ell(k) + \frac{1}{\sqrt{2}} \uparrow (k-1) + \frac{1}{\sqrt{2}} \downarrow (k+1) \\
c(\downarrow k) & = & 1r(k) + \frac{1}{\sqrt{2}} \uparrow (k-1) - \frac{1}{\sqrt{2}} \downarrow (k+1).
\end{array}$$

The right-hand-side of these equations is a formal sum over elements of the set $F(\mathbb{Z} + \mathbb{Z}) = (\mathbb{Z}+\mathbb{Z}) + (\mathbb{Z}+\mathbb{Z})$. What is possibly confusing is that in this quadruple coproduct of integers the left part $\mathbb{Z} + \mathbb{Z}$ is used for output, and the right part $\mathbb{Z} + \mathbb{Z}$ as state, for further computation. In these definitions the first parts $\ell(k)$ and $r(k)$ are the left and right part of this output. The second part involves multiplication with scalars $\pm\frac{1}{\sqrt{2}} \in \mathbb{C}$ and elements $\uparrow (k \pm 1), \downarrow (k \pm 1)$ in the right (state) component of $(\mathbb{Z} + \mathbb{Z}) + (\mathbb{Z} + \mathbb{Z})$.

As machine functor we take $M(X) = \mathcal{M}_{\mathbb{C}}(\mathbb{Z} + \mathbb{Z}) \times X$, with label set $A = 1$, and with final coalgebra $Z = \mathcal{M}_{\mathbb{C}}(\mathbb{Z}+\mathbb{Z})^{\mathbb{N}}$ of streams. The extension natural tranformation follows from additivity of the multiset monad $\mathcal{M}_{\mathbb{C}}$ (like in (2), see [8]), in:

$$\begin{tikzcd}
\mathcal{M}_{\mathbb{C}}((\mathbb{Z} + \mathbb{Z}) + X) \arrow[r, "\mathfrak{e} = \lambda\varphi.\, \langle \varphi \circ \kappa_1, \varphi \circ \kappa_2 \rangle", "\cong"'] \arrow[d, phantom, "\parallel"] & \mathcal{M}_{\mathbb{C}}(\mathbb{Z} + \mathbb{Z}) \times \mathcal{M}_{\mathbb{C}}(X) \arrow[d, phantom, "\parallel"] \\
\mathcal{M}_{\mathbb{C}}(F(X)) & M(\mathcal{M}_{\mathbb{C}}(\mathbb{Z}))
\end{tikzcd}$$

The coalgebra $\widehat{E}(c)\colon \mathcal{M}_{\mathbb{C}}(\mathbb{Z}+\mathbb{Z}) \to \mathcal{M}_{\mathbb{C}}(\mathbb{Z}+\mathbb{Z}) \times \mathcal{M}_{\mathbb{C}}(\mathbb{Z}+\mathbb{Z})$ in the category of vector spaces over $\mathbb{C}$, resulting from Theorem 13, gives rise to a trace map $\mathrm{tr}(c)\colon \mathbb{Z}+\mathbb{Z} \to \mathcal{M}_{\mathbb{C}}(\mathbb{Z}+\mathbb{Z})^{\mathbb{N}}$. Thus, for the initial (upwards) state $\uparrow 0 \in \mathbb{Z}+\mathbb{Z}$ we get the probability $P(n,k)$ of ending up after n steps at position $k \in \mathbb{Z}$ on the line via the Born rule:

$$P(n,k) \;=\; \Big|\,\mathrm{tr}(c)(\uparrow 0)(n)(\ell k)\,\Big|^2 + \Big|\,\mathrm{tr}(c)(\uparrow 0)(n)(rk)\,\Big|^2.$$

These probabilities are computed in an *ad hoc* manner in [15].

8 Discussion

In this paper, we have systematically studied trace semantics, bringing together two perspectives: the coalgebraic trace semantics of [12] and the coalgebraic language equivalence via a determinization construction of [25]. Having the two perspectives together enables us to extend the class of systems that fits the framework of [12], while guaranteeing that the coalgebraic trace semantics from [12] fits in the current setting (Proposition 15). We illustrated the whole approach with several non-trivial examples, including quantum walks and simple Segala systems.

Our set-up has a certain overlap with [1], but the focus there is on getting coincidence of initial algebras and final coalgebras in categories $\mathcal{EM}(T)$, using stronger assumptions than here, namely commutativity of endofunctors $TF \cong GT$, see Section 6, where we only have a law $TF \Rightarrow GT$.

In certain cases one can also use duality results to obtain trace semantics, see [22,17]. How this relates to the current setting is unclear at this stage.

We see several (other) directions for future work. We have studied most of our examples using the deterministic automata functor. By varying this functor, as in the example of Segala systems, one moves from a traces-as-words semantics to a more elaborate one, *e.g* with traces-as-trees. We would like to further explore this and to study whether we can use our framework to give a coalgebraic account of temporal logics such as LTL and CTL, for which some work has appeared in [7]. The systematic development of sound and complete calculi for coalgebras is studied in [24,26,3]. In [26], the axiomatization is provided for a system modeled as a TF coalgebra, but for the completeness proof there is a somewhat unexplained move to coalgebras of type GT. A general question is whether we can use our framework to justify this step and whether we can exploit it for a more general transfer result of soundness and completeness.

Acknowledgments. We are grateful to the anonymous referees for valuable comments. The work of Alexandra Silva is partially funded by the ERDF through the Programme COMPETE and by the Portuguese Foundation for Science and Technology, project ref. `PTDC/EIA-CCO/122240/2010` and `SFRH/BPD/71956/2010`. The work of Ana Sokolova is partially supported by the Austrian Science Fund (FWF) grant V00125.

References

1. A. Balan and A. Kurz. On coalgebras over algebras. *Theor. Comp. Sci.*, 412(38):4989–5005, 2011.

2. M. Barr and Ch. Wells. *Toposes, Triples and Theories*. Springer, Berlin, 1985. Revized and corrected version available from URL: `www.cwru.edu/artsci/math/wells/pub/ttt.html`.
3. M. M. Bonsangue, Milius S, and A. Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *CoRR*, abs/1104.2803, 2011.
4. F. Borceux. *Handbook of Categorical Algebra*, volume 50, 51 and 52 of *Encyclopedia of Mathematics*. Cambridge Univ. Press, 1994.
5. M. Boreale. Weighted bisimulation in linear algebraic form. In *Proc. of CONCUR '09*, pages 163–177. Springer, 2009. LNCS 5710.
6. P. Castro, P. Panangaden, and D. Precup. Equivalence relations in fully and partially observable Markov decision processes. In *Proc. IJCAI 2009*, pages 1653–1658, 2009.
7. C. Cîrstea. Maximal traces and path-based coalgebraic temporal logics. *Theor. Comput. Sci.*, 412(38):5025–5042, 2011.
8. D. Coumans and B. Jacobs. Scalars, monads and categories. In C. Heunen and M. Sadrzadeh, editors, *Compositional methods in Physics and Linguistics*. Oxford Univ. Press, 2012.
9. S. Crafa and F. Ranzato. A spectrum of behavioral relations over LTSs on probability distributions. In *Proc. CONCUR 2011*, pages 124–139. LNCS 6901, 2011.
10. Y. Deng, R. van Glabbeek, M. Hennessy, and C. Morgan. Characterising testing preorders for finite probabilistic processes. *Logical Methods in Computer Science*, 4(4), 2008.
11. E. Doberkat. Eilenberg-moore algebras for stochastic relations. *Information and Computation*, 204(12):1756–1781, 2006.
12. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace theory via coinduction. *Logical Methods in Computer Science*, 3(4:11), 2007.
13. H. Hermanns, A. Parma, R. Segala, B. Wachter, and L. Zhang. Probabilistic logical characterization. *Information and Computation*, 209(2):154–172, 2011.
14. J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Wesley, 2006.
15. B. Jacobs. Coalgebraic walks, in quantum and Turing computation. In M. Hofmann, editor, *Foundations of Software Science and Computation Structures*, number 6604 in Lect. Notes Comp. Sci., pages 12–26. Springer, Berlin, 2011.
16. P.T. Johnstone. Adjoint lifting theorems for categories of algebras. *Bull. London Math. Soc.*, 7:294–297, 1975.
17. Ch. Kissig and A. Kurz. Generic trace logics, 2011. `arxiv.org/abs/1103.3239`.
18. E.G. Manes. *Algebraic Theories*. Springer, Berlin, 1974.
19. S. Mac Lane. *Categories for the Working Mathematician*. Springer, Berlin, 1971.
20. E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.
21. A. Parma and R. Segala. Logical characterizations of bisimulations for discrete probabilistic systems. In *Proc. FoSSaCS 2007*, pages 287–301. LNCS 4423, 2007.
22. D. Pavlović, M. Mislove, and J. Worrell. Testing semantics: Connecting processes and process logics. In M. Johnson and V. Vene, editors, *Algebraic Methods and Software Technology*, number 4019 in Lect. Notes Comp. Sci., pages 308–322. Springer, Berlin, 2006.
23. R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. Concur'94*, pages 481–496. LNCS 836, 1994.
24. A. Silva. *Kleene coalgebra*. PhD thesis, Radboud University Nijmegen, 2010.
25. A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing the powerset construction, coalgebraically. In *Proc. FSTTCS 2010*, volume 8 of *LIPIcs*, pages 272–283, 2010.
26. A. Silva and A. Sokolova. Sound and complete axiomatization of trace semantics for probabilistic systems. In *Proc. MFPS 2011*, pages 291–311. ENTCS 276, 2011.
27. D. Varacca and G. Winskel. Distributing probability over non-determinism. *Mathematical Structures in Computer Science*, 16(1):87–113, 2006.