université
PAR S
D DEROT
PARIS 7

Radboud Universiteit

MASTER 2

LOGIQUE MATHÉMATIQUE ET FONDEMENTS DE L'INFORMATIQUE

Trace semantics via determinization for probabilistic transition systems

Author:
Alexandre GOY

Supervisor:
Dr. Jurriaan ROT

February 27, 2018

Contents

Abstract

A coalgebraic definition of finite and infinite trace semantics for probabilistic transition systems has recently been given using a certain Kleisli category. In this paper this semantics is developed using a coalgebraic method which is an instance of general determinization. Once applied to discrete systems, this point of view allows the exploitation of the determinized structure by up-to techniques. Thereby it becomes possible to algorithmically check the equivalence of two finite probabilistic transition systems.

Introduction

Automata theory is certainly one of the most explored branches of computer science. To meet the growing needs in probabilistic programming, model checking, or randomized algorithms, many kinds of automata are shaped with a probabilistic behaviour. Here is one of them: a generative probabilistic transition system (PTS) consists of a state space X, where every state x can either terminate or transition to any state, with a certain probability. Each non-terminating transition outputs a letter a from an alphabet A. With this informal definition, one can see that given a state x, each word w is generated by the automaton with a certain probability $[\![x]\!](w)$. The function $w \mapsto [\![x]\!](w)$ is itself a probability distribution if we take into account both finite and infinite words. The aim of this paper is to study the formal definition of these semantics, referred to as the *trace semantics*.

Automata will be described as usual using graphs. Each state is pictured by a circle and there is a distinguished terminal state $*$ which is double-circled. A transition is represented by an arrow labeled with its probability. Non-terminating arrows are further labeled with one transition letter. Consider as a first example the following PTS:

$$\text{a,}1/2 \;\circlearrowright\; x \xrightarrow{\;1/2\;} \circledast$$
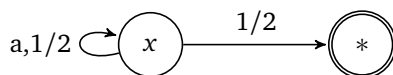
One can intuitively associate trace semantics to this automaton. Given the above PTS, the only reasonable semantics is $[\![x]\!](a^n) = \frac{1}{2^{n+1}}$. It is clearly a probability measure over A^*. Actually, some strange phenomena can occur if $|A| \geq 2$ because of the fact that $A^{\mathbb{N}}$ is uncountable. Consider the following example where $A = \{a, b\}$.

$$\text{a,}1/2 \;\circlearrowright\; y \;\circlearrowleft\; \text{b,}1/2$$
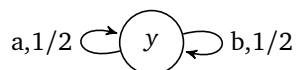
No matter how you look at it, you should end up with $[\![y]\!](w) = 0$ for every finite or infinite word w. A possible way to fix this is to ask for a subprobability over words instead of a probability. Then $[\![y]\!]$ can be defined as a subprobability with total mass 0. But a serious problem appears, because with this definition, the following state z has the same (trivial) semantics as y, so that they are deemed equivalent.

$$\text{a,}3/4 \;\circlearrowright\; z \;\circlearrowleft\; \text{b,}1/4$$
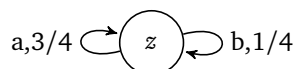
But for example, y is twice as likely as z to generate an infinite word that begins with b. This is a concrete behavioral difference. In case infinite traces are taken into account,

both states are thus not to be considered equivalent. Further, we would like to have techniques to prove that they are (not). To this purpose, it is necessary to dive into measure theory and define the arguments of $[\![x]\!]$ to be *sets of words* instead of being simply words.

To this end we will use *coalgebras*. State-based sytems are increasingly modeled using a coalgebraic point of view, which benefits from the powerful toolbox of category theory. An introduction to coalgebras can be found in [9] or [8]. This framework is notably convenient when it comes to define trace semantics; this is performed for very general PTS in [11], using a construction in a Kleisli category as in [7]. The usual counterpart of this kind of construction is to carry out a determinization process formally based on an Eilenberg-Moore category. Concretely, the state space is changed in order to make transitions become deterministic. The comparison between these two methods is discussed in [10]. Determinization allows to exploit *bisimulation up-to* techniques, a family of proof methods for behavioural equivalence of state-based systems, which have been extensively applied in concurrency theory and, more recently, in automata theory. For example, non-deterministic automata can be determinized via the well-known powerset construction. This opens the way for the HKC algorithm of [4] that checks the equivalence of non-deterministic automata using up-to techniques.

Our main contribution is to redefine the Kleisli trace semantics of [11] using the Eilenberg-Moore method. This is done for both discrete and continuous sytems. In the case of discrete systems, our approach allows to generalize the HKC algorithm of [3] to an algorithm HKC$^\infty$ that checks equivalence of states (i.e. it checks if $[\![x]\!] = [\![y]\!]$) for both finite and infinite words. Our paper is organized as follows. In section 1 are introduced the basic concepts of category theory, coalgebraic modeling and measure theory, including a part about measurable sets of words. Section 2 deals with the discrete case. The determinization process is performed by hand in order to go straight to the HKC$^\infty$ algorithm and its correctness. This is followed by a few examples. The general setting is presented in section 3, where the origin of the determinization process is further explained. Its central result is Theorem 34. Section 3 ends with the proof that our semantics is the same as the Kleisli semantics from [11]; the general framework that relates both constructions is mentioned. It might well be worth to begin by scanning section 2 to see the underlying ideas of the construction without caring about the measurability of every function, as well as some concrete brightening examples. Definitions and theorems that are not explicitly stated or referenced are considered as folklore and can be found in any basic related book.

Related work. Our main source for the overall spirit of the whole paper is undoubtedly [11], which is in turn influenced by the Kleisli constructions done in [7]. Both versions of HKC presented in [4] and [3] are good starting points to understand bisimulation up-to and the determinization method. The overhanging link between Kleisli and determinization is discussed in [10].

1 Preliminaries

The set of positive integers is $\mathbb{N}$. The set of real numbers is $\mathbb{R}$, the set of non-negative real numbers is $\mathbb{R}_+$. The notation 1 stands for a set with only one element $1 = \{*\}$. Given a set X, its power set is $\mathscr{P}(X)$.

1.1 Measure theory

Let X be any set. A σ-*algebra* on X is a subset $\Sigma_X \subseteq \mathscr{P}(X)$ such that $\emptyset \in \Sigma_X$ and Σ_X is closed under complementation and countable union. Note that this implies that $X \in \Sigma_X$ and that Σ_X is closed under countable intersection and under set difference. Given any subset $G \subseteq \mathscr{P}(X)$, there always exists a smallest σ-algebra containing G. We call it the σ-algebra *generated* by G and denote it by $\sigma_X(G)$. For example, $\mathscr{P}(X)$ is a σ-algebra on X. When working with real numbers $\mathbb{R}$, we will use the Borel σ-algebra $\mathscr{B}(\mathbb{R}) = \sigma_{\mathbb{R}}(\{(-\infty, x] \mid x \in \mathbb{R}\})$. More specifically we denote the line segment $[0, 1]$ by $\mathbb{I}$ and use $\mathscr{B}(\mathbb{I}) = \{B \cap \mathbb{I} \mid B \in \mathscr{B}(\mathbb{R})\}$ as the canonical σ-algebra on $\mathbb{I}$. If X is a set and Σ_X is a σ-algebra on X, the pair (X, Σ_X) is called a measurable space. From now we will write X for (X, Σ_X) when the σ-algebra used is clear.

Product. Given measurable spaces (X, Σ_X) and (Y, Σ_Y), we define a product σ-algebra on $X \times Y$ by $\Sigma_X \otimes \Sigma_Y = \sigma_{X \times Y}(\{S_X \times S_Y \mid S_X \in \Sigma_X, S_Y \in \Sigma_Y\})$. The product of measurable spaces is then defined by $(X, \Sigma_X) \otimes (Y, \Sigma_Y) = (X \times Y, \Sigma_X \otimes \Sigma_Y)$. Note that if $X \in G_X \subseteq \mathscr{P}(X)$ and $Y \in G_Y \subseteq \mathscr{P}(Y)$, then $\sigma_{X \times Y}(\{S_X \times S_Y \mid S_X \in G_X, S_Y \in G_Y\}) = \sigma_X(G_X) \otimes \sigma_Y(G_Y)$.

Sum. Given measurable spaces (X, Σ_X) and (Y, Σ_Y), we define a sum σ-algebra on the disjoint union $X + Y = \{(x, 0) \mid x \in X\} \cup \{(y, 1) \mid y \in Y\}$ by $\Sigma_X \oplus \Sigma_Y = \{S_X + S_Y \mid S_X \in \Sigma_X, S_Y \in \Sigma_Y\}$. The sum of measurable spaces is then defined by $(X, \Sigma_X) \oplus (Y, \Sigma_Y) = (X + Y, \Sigma_X \oplus \Sigma_Y)$. Note that if $\emptyset \in G_X \subseteq \mathscr{P}(X)$ and $Y \in G_Y \subseteq \mathscr{P}(Y)$, then $\sigma_{X+Y}(G_X \oplus G_Y) = \sigma_X(G_X) \oplus \sigma_Y(G_Y)$, see [11].

Binary products and sums can be easily generalized to finite products and sums by induction. These are denoted by $\bigotimes_{i \in I}$ and $\bigoplus_{i \in I}$ respectively.

A function $f : (X, \Sigma_X) \to (Y, \Sigma_Y)$ is *measurable* if for all $S_Y \in \Sigma_Y$, $f^{-1}(S_Y) \in \Sigma_X$. The composition of measurable functions is measurable.

Lemma 1. *If $f : (X, \Sigma_X) \to (Y, \sigma_Y(G_Y))$ is such that for all $S_Y \in G_Y$, $f^{-1}(S_Y) \in \Sigma_X$, then it is measurable.*

Proof. The proof is given as it shows the classical way of reasoning about generated σ-algebras. Let $\Sigma = \{S \subseteq Y \mid f^{-1}(S) \in \Sigma_X\} \subseteq \mathscr{P}(Y)$. Note that $\emptyset \in \Sigma$ because $f^{-1}(\emptyset) = \emptyset \in \Sigma_X$, and that Σ is closed under complementation and countable union. Indeed, if $(S_n)_{n \in \mathbb{N}} \in \Sigma^{\mathbb{N}}$ are such that $f^{-1}(S_n) \in \Sigma_X$ for all $n \in \mathbb{N}$, we have $f^{-1}(Y \setminus S_0) = X \setminus f^{-1}(S_0) \in \Sigma_X$ as Σ_X is closed under complementation, and $f^{-1}\left(\bigcup_{n \in \mathbb{N}} S_n\right) = \bigcup_{n \in \mathbb{N}} f^{-1}(S_n) \in \Sigma_X$ as Σ_X is closed under countable unions. Thus Σ is a σ-algebra. Furthermore, the hypothesis gives that $G_Y \subseteq \Sigma$. As $\sigma_Y(G_Y)$ is the *smallest* σ-algebra that contains G_Y, we get $\sigma_Y(G_Y) \subseteq \Sigma$ so that f is measurable. $\qquad\square$

For any family of measurable spaces $(Z_i, \Sigma_{Z_i})_{i \in I}$, and any family of functions $(f_i)_{i \in I}$: $Y \to Z_i$, define Σ_Y as the smallest σ-algebra on Y that makes every f_i measurable. This is the σ-algebra *generated* by $(f_i)_{i \in I}$.

Lemma 2. *Given $h : (X, \Sigma_X) \to (Y, \Sigma_Y)$ where Σ_Y is generated by some $(f_i)_{i \in I}$, assume that for all $i \in I$, $(f_i \circ h)$ is measurable. Then h is measurable.*

Proof. First prove that $\Sigma_Y = \sigma_Y(\{f_i^{-1}(S_i) \mid i \in I, S_i \in \Sigma_{Z_i}\})$. The σ-algebra on the right makes all functions f_i measurable, thus it contains Σ_Y. On the other hand, for all $i \in I$, $S_i \in \Sigma_{Z_i}$, we have $f_i^{-1}(S_i) \in \Sigma_Y$ because f_i is measurable. Now apply Lemma 1 using that for all $i \in I$ and $S_i \in \Sigma_{Z_i}$, $h^{-1}(f_i^{-1}(S_i)) = (f_i \circ h)^{-1}(S_i) \in \Sigma_X$ because $f_i \circ h$ is measurable. $\qquad\square$

For instance, using Lemma 2 facilitates measurability proofs for functions with several components. Let $(Z_i, \Sigma_i)_{i \in I}$ be a finite sequence of measurable spaces, (X, Σ_X) be a measurable space, and $f : (X, \Sigma_X) \to \left(\prod_{i \in I} Z_i, \bigotimes_{i \in I} \Sigma_i\right)$ be a function with components $(f_i : X \to Z_i)_{i \in I}$. Note that $\bigotimes_{i \in I} \Sigma_i$ is the smallest σ-algebra on $\prod_{i \in I} Z_i$ that makes every projection $\pi_j : \prod_{i \in I} Z_i \to Z_j$ measurable. (Indeed, $\pi_j^{-1}(S_j) = \prod_{i \in I} X_i^\varepsilon$ where $X_j^\varepsilon = S_j$ and $X_i^\varepsilon = X_i$ for $i \neq j$, so that $\bigotimes_{i \in I} \Sigma_i$ makes every π_j measurable; on the other hand, a generator of $\prod_{i \in I} Z_i$ can be written $\prod_{i \in I} S_i = \bigcap_{i \in I} \pi^{-1}(S_i)$ which is in any σ-algebra that makes every π_j measurable.) Thus applying Lemma 2 it is sufficient that every f_j is measurable in order for f to be measurable.

A *finite measure* on the measurable space (X, Σ_X) is a function $m : \Sigma_X \to \mathbb{R}_+$ such that $m(\emptyset) = 0$ and if $(S_n)_{n \in \mathbb{N}} \in \Sigma_X^{\mathbb{N}}$ are disjoint, then $m\left(\bigcup_{n \in \mathbb{N}} S_n\right) = \sum_{n \in \mathbb{N}} m(S_n)$ (this is called *σ-additivity*). In this case the triple (X, Σ_X, m) is called a *measure space*. If $m(X) = 1$ (resp. $m(X) \leq 1$) it is a *probability space* and m is a *probability measure* (resp. *sub-probability*). Finite measures have pleasant properties: $m(X \setminus A) = m(X) - m(A)$ for any $A \in \Sigma_X$, $m\left(\bigcup_{n \in \mathbb{N}} A_n\right) = \lim_{n \to \infty} m(A_n)$ for any increasing sequence $(A_n)_{n \in \mathbb{N}}$ of Σ_X, and analogously $m\left(\bigcap_{n \in \mathbb{N}} B_n\right) = \lim_{n \to \infty} m(B_n)$ for any decreasing sequence $(B_n)_{n \in \mathbb{N}}$ of Σ_X. These measures are called finite because they satisfy $m(X) < \infty$.

Measure over words

We begin with some reminders about languages. Any finite set A can be called an *alphabet* and its elements *letters*. The set of words of length n with letters in A is denoted by A^n. By convention $A^0 = \{\varepsilon\}$ where ε is the empty word. The set of finite words over A is denoted by $A^* = \bigcup_{n \in \mathbb{N}} A^n$, the set of infinite words by $A^\omega = A^{\mathbb{N}}$ and the set of (finite and infinite) words by $A^\infty = A^* \cup A^\omega$. A language L is a subset of $\mathscr{P}(A^*)$. It can be seen as a function $L : A^* \to \{0, 1\}$, by setting $L(w) = 1$ iff $w \in L$. The *language derivative* of L with respect to a letter a is defined by $L_a(w) = L(aw)$.

The length of $w \in A^\infty$ is denoted by $|w| \in \mathbb{N} \cup \{\infty\}$. The concatenation function $A^* \times A^\infty \to A^\infty$ is denoted by juxtaposition and defined by $uv(n) = u(n)$ if $n < |u|$ and $uv(n) = v(n - |u|)$ if $|u| \leq n < |u| + |v|$. It can be extended to languages $\mathscr{P}(A^*) \times \mathscr{P}(A^\infty) \to \mathscr{P}(A^\infty)$ by setting $LM = \{uv \mid u \in L, v \in M\}$. In the following, $\{w\}M$ will be denoted by wM for convenience.

One aim of this paper is to define certain probability measures on A^∞ which reflects

the behaviour of a given automaton. For this purpose we need to make precise which σ-algebra we use.

Definition 3. *Let $S_\infty = \{\emptyset\} \cup \{\{w\} \mid w \in A^*\} \cup \{wA^\infty \mid w \in A^*\}$. Define the σ-algebra of measurable sets of words to be $\Sigma_{A^\infty} = \sigma_{A^\infty}(S_\infty)$.*

This σ-algebra is generated by a countable family of simple generators: the empty set, the singletons of finite words, and the cones, i.e., sets wA^∞ of words that have the finite word w as a prefix. In the sequel, this is the σ-algebra on A^∞ implicitly used. As a first step, we take a look at some properties of measurable sets of words.

Proposition 4. *The following sets of words are measurable:*

  (i) The singleton $\{w\}$ for any $w \in A^\infty$;

  (ii) Any countable language;

 (iii) Any language of finite words;

 (iv) $\emptyset, A^*, A^\omega, A^\infty$;

  (v) The concatenation LS where $L \subseteq A^*$ and $S \in \Sigma_{A^\infty}$.

Proof. (i) It is already known for finite words w since $\{w\} \in S_\infty \subseteq \Sigma_{A^\infty}$. Let $w \in A^\omega$. Let $w_{|n}$ be the finite word of length n defined by $w_{|n}(k) = w(k)$ for any $k \leq n$. Then $\{w\} = \bigcap_{n \in \mathbb{N}} w_{|n} A^\infty \in \Sigma_{A^\infty}$ because Σ_{A^∞} is closed under countable intersections.

  (ii) Any countable language is a countable union of singletons, which are all measurable sets.

 (iii) A language of finite words is a subset of the countable set A^*, so it is countable, hence measurable.

 (iv) The empty set and A^∞ are in Σ_{A^∞} because this is a σ-algebra on A^∞. The set A^* is a language of finite words. Finally, $A^\omega = A^\infty \setminus A^* \in \Sigma_{A^\infty}$ because Σ_{A^∞} is closed under complementation.

  (v) As $LS = \bigcup_{w \in L} wS$ is a countable union, it is sufficient to prove that for all $S \in \Sigma_{A^\infty}$, wS is measurable for any $w \in A^*$. Let $\Sigma = \{S \subseteq A^\infty \mid \forall w \in A^*, wS \in \Sigma_{A^\infty}\}$. It is easy to see that $S_\infty \subseteq \Sigma$, because for any $w \in A^*$, $w\emptyset = \emptyset \in \Sigma_{A^\infty}$, $w\{u\} = \{wu\} \in \Sigma_{A^\infty}$ and $w(uA^\infty) = (wu)A^\infty \in \Sigma_{A^\infty}$. In particular $\emptyset \in \Sigma$. Moreover, if $(S_n)_{n \in \mathbb{N}}$ are all in Σ, then for any $w \in A^*$ we have $w\bigcup_{n \in \mathbb{N}} S_n = \bigcup_{n \in \mathbb{N}} wS_n \in \Sigma_{A^\infty}$ since Σ_{A^∞} is closed under countable union, and $w(A^\infty \setminus S_0) = wA^\infty \setminus wS_0 \in \Sigma_{A^\infty}$ since Σ_{A^∞} is closed under set difference. Thus $\bigcup_{n \in \mathbb{N}} S_n \in \Sigma$ and $A^\infty \setminus S_0 \in \Sigma$ so Σ is a σ-algebra. Since it contains S_∞, we get that $\Sigma_{A^\infty} \subseteq \Sigma$ and this is exactly what had to be proved.

    □

In the following, if m is a measure over A^∞ and $w \in A^\infty$, we will write $m(w)$ instead of $m(\{w\})$. Since S_∞ has a pleasant structure (it is a *covering semiring of sets*), we have the following key theorem:

Theorem 5 (from [11]). *Let $m : S_\infty \to \mathbb{R}_+$ be a map satisfying $m(\emptyset) = 0$. The two following conditions are equivalent.*

  (i) There exists a unique measure $\tilde{m} : \Sigma_{A^\infty} \to \mathbb{R}_+$ such that $\tilde{m}_{|S_\infty} = m$.

  (ii) For all $w \in A^*$, $m(wA^\infty) = m(w) + \sum_{a \in A} m(waA^\infty)$.

Proof. $(i) \Rightarrow (ii)$ This is obvious because the equation comes directly from the σ-additivity of $\tilde{m}$. $(ii) \Rightarrow (i)$ According to Lemma 3.18 of $[11]$, (ii) is equivalent to the fact that m is a pre-measure. Using the extension theorem (Proposition 2.4 in $[11]$), this pre-measure can be uniquely extended to a measure as in (i). $\qquad\square$

The notion of language derivative was defined above by $L_a(w) = L(aw)$. One can introduce the same operation for measures over words.

Definition 6 (Measure derivative)**.** *Let m be a measure on A^∞ and $a \in A$. The map m_a defined by $m_a(S) = m(aS)$ for any $S \in \Sigma_{A^\infty}$ is a measure, called the* measure derivative *of m (with respect to a).*

This is well-defined because sets of the shape aS for $S \in \Sigma_{A^\infty}$ are measurable according to Proposition 4 (v), and $m_a(\emptyset) = m(a\emptyset) = m(\emptyset) = 0$ and $m_a \left(\bigcup_{n \in \mathbb{N}} A_n \right) = m \left(\bigcup_{n \in \mathbb{N}} aA_n \right) = \sum_{n \in \mathbb{N}} m(aA_n) = \sum_{n \in \mathbb{N}} m_a(A_n)$ if the A_n are disjoint.

In the following, the space of sub-probability measures on $(A^\infty, \Sigma_{A^\infty})$ is denoted by $\mathscr{M}(A^\infty)$. Note that $\mathscr{M}(A^\infty)$ is closed under measure derivatives: if $m(A^\infty) \leq 1$ then $m_a(A^\infty) = m(aA^\infty) \leq m(A^\infty) \leq 1$.

1.2 Category theory

We recall here some basics of category theory. A *category* consists of a class of objects $\mathbf{C}$ and a class of morphisms (or arrows) $\mathbf{C}(X, Y)$ for every objects X, Y of $\mathbf{C}$. The notation $f : X \to Y$ stands for the sentence "f is an arrow of $\mathbf{C}(X, Y)$". For each object X of $\mathbf{C}$ there is an identity morphism $id_X : X \to X$. Furthermore, there is a composition function $\circ$ which is associative i.e. $(h \circ g) \circ f = h \circ (g \circ f)$, and such that if $f : X \to Y$, then $f \circ id_X = f = id_Y \circ f$. The composition $g \circ f$ is possible iff there exists objects X, Y, Z such that $f : X \to Y$ and $g : Y \to Z$. In the following, we will mainly work with the two following categories:

  ▷ The category **Sets** of sets and functions. Objects are usual sets, morphisms are functions, identity morphisms are identity functions and composition is given by the usual composition of functions.

  ▷ The category **Meas** of measurable sets and functions. Objects are *measurable spaces* (X, Σ_X). Morphisms are *measurable functions $f : (X, \Sigma_X) \to (Y, \Sigma_Y)$*. Identity morphisms are identity functions and composition is given by the usual composition of functions.

Let $\mathbf{C}$ be a category and A be an object of $\mathbf{C}$. The object A is *final* if for all object X of $\mathbf{C}$ there exists a unique morphism $f_X : X \to A$ called the *final morphism*. Such an object is unique up to (unique) isomorphism.

A *functor* from $\mathbf{C}$ to $\mathbf{D}$ is a mapping F that associates to every object X of $\mathbf{C}$ an object FX of $\mathbf{D}$, to every morphism $f : X \to Y$ a morphism $Ff : FX \to FY$, and such that $F(id_X) = id_{FX}$ and $Fg \circ Ff = F(g \circ f)$. An endofunctor of $\mathbf{C}$ is a functor $F : \mathbf{C} \to \mathbf{C}$. The composition of two functors is still a functor. For example, the identity functor $Id_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$ maps each object and morphism to itself. Given an object X of $\mathbf{D}$, the constant functor $X : \mathbf{C} \to \mathbf{D}$ maps each object to X and each morphism to id_X.

Let $F, G : \mathbf{C} \to \mathbf{D}$ be some functors. A *natural transformation* $\lambda : F \Rightarrow G$ consists of a

D-morphism $\lambda_X : FX \to GX$ for every $X \in \mathbf{C}$, such that for every $\mathbf{C}$-morphism $f : X \to Y$, the following diagram commutes.

$$
\begin{array}{ccc}
FX & \xrightarrow{\;\lambda_X\;} & GX \\
{\scriptstyle Ff}\Big\downarrow & & \Big\downarrow{\scriptstyle Gf} \\
FY & \xrightarrow[\;\lambda_Y\;]{} & GY
\end{array}
$$

Given a category $\mathbf{C}$, a *monad* is a triple (T, η, μ) where $T : \mathbf{C} \to \mathbf{C}$ is an endofunctor and $\eta : Id_{\mathbf{C}} \Rightarrow T$, $\mu : TT \Rightarrow T$ are natural transformations called *unit* and *multiplication* respectively, such that the following two diagrams commute.

$$
\begin{array}{ccc}
TX & \xrightarrow{\;\eta_{TX}\;} & TTX \\
{\scriptstyle T\eta_X}\Big\downarrow & \diagdown & \Big\downarrow{\scriptstyle \mu_X} \\
TTX & \xrightarrow[\;\mu_X\;]{} & TX
\end{array}
\qquad\qquad
\begin{array}{ccc}
TTTX & \xrightarrow{\;\mu_{TX}\;} & TTX \\
{\scriptstyle T\mu_X}\Big\downarrow & & \Big\downarrow{\scriptstyle \mu_X} \\
TTX & \xrightarrow[\;\mu_X\;]{} & TX
\end{array}
$$

Examples. In the context of coalgebras, monads are often used to model branching behaviour, like non deterministic branching or probabilistic branching. It will always be clear from the name of the functor which monad is used. Hence, units are always denoted by η and multiplications by μ.

  ▷ In **Sets**, the power set monad $(\mathscr{P}, \eta, \mu)$ is defined as follows. Given two sets X, Y and a function $f : X \to Y$, $\mathscr{P}X$ is the power set $\mathscr{P}(X)$ and $\mathscr{P}f : S \in \mathscr{P}X \mapsto f(S) \in \mathscr{P}Y$ is the direct image. The unit is the singleton $\eta_X(x) = \{x\}$ and the multiplication is given by $\mu_X(\mathscr{S}) = \bigcup_{S \in \mathscr{S}} S$.

  ▷ In **Sets**, the probability monad (P, η, μ) is defined as follows. For any $u : X \to \mathbb{I}$ the *support* of u is defined by $supp(u) = \{x \in X \mid u(x) \neq 0\}$. If $supp(u)$ is finite we say that u has finite support. Given two sets X, Y and a function $f : X \to Y$, define $PX = \left\{ u : X \to \mathbb{I} \mid u \text{ has finite support and } \sum_{x \in X} u(x) = 1 \right\}$ and $Pf(u)(y) = \sum_{x \in f^{-1}(\{y\})} u(x)$. The unit is the Kronecker delta $\eta_X(x)(y) = \delta_{x,y}$ and the multiplication is given by $\mu_X(U)(y) = \sum_{u \in PX} U(u)u(y)$. This monad models finitely branching probabilistic behaviour. A variant is the subprobability monad (D, η, μ). The only difference is that the sum may be less than 1: $DX = \left\{ u : X \to \mathbb{I} \mid u \text{ has finite support and } \sum_{x \in X} u(x) \leq 1 \right\}$.

Algebras and distributive laws

Let $F : \mathbf{C} \to \mathbf{C}$ be an endofunctor. An *F-algebra* is an object X together with an arrow $\alpha : FX \to X$. An F-algebra can be viewed as an operation that takes some information out of FX and constructs a new element in X. For example, in **Sets**, if some set S is given and $FX = S \times X$, $Ff = Id \times f$, then the arrow $\alpha : S \times S^{\omega} \to S^{\omega}$ given by concatenation $\alpha(h, t) = h \cdot t$ can be thought of as a way of making a list from a head element h and a tail list t. A morphism of F-algebras $\alpha : FX \to X$ and $\beta : FY \to Y$ is an arrow $f : X \to Y$ such that $f \circ \alpha = \beta \circ Ff$. The collection of F-algebras and their morphisms form a category denoted by $\mathbf{Alg}(F)$.

The last section of this paper is committed to make clear why the construction of trace semantics is canonical. Showing that it comes from a *distributive law* is a convenient way to find this out. A distributive law of a monad over a functor formally allows to reverse the order in which they are applied. It is based on a variant of algebras named Eilenberg-Moore algebras.

Let (T, η, μ) be a monad on a category **C**. An *Eilenberg-Moore T-algebra* is a T-algebra α such that the following diagrams commute.

$$
\begin{array}{ccc}
X & \xrightarrow{\eta_X} & TX \\
& \searrow{\scriptstyle id_X} & \downarrow{\scriptstyle \alpha} \\
& & X
\end{array}
\qquad\qquad
\begin{array}{ccc}
TTX & \xrightarrow{\mu_X} & TX \\
\downarrow{\scriptstyle T\alpha} & & \downarrow{\scriptstyle \alpha} \\
TX & \xrightarrow{\alpha} & X
\end{array}
$$

Given two Eilenberg-Moore T-algebras $\alpha : TX \to X$ and $\beta : TY \to Y$, a morphism from α to β is a **C**-arrow $f : X \to Y$ such that $f \circ \alpha = \beta \circ Tf$. Eilenberg-Moore T-algebras and their morphisms form a category denoted by **EM**(T).

Let $F : \mathbf{C} \to \mathbf{C}$ be an endofunctor. A natural transformation $\lambda : TF \Rightarrow FT$ is a *distributive law* if the following diagrams commute for every object X of **C**.

$$
\begin{array}{ccc}
FX & \xrightarrow{id_X} & FX \\
\downarrow{\scriptstyle \eta_{FX}} & & \downarrow{\scriptstyle F\eta_X} \\
TFX & \xrightarrow{\lambda_X} & FTX
\end{array}
\qquad
\begin{array}{ccccc}
TTFX & \xrightarrow{T\lambda_X} & TFTX & \xrightarrow{\lambda_{TX}} & FTTX \\
\downarrow{\scriptstyle \mu_{FX}} & & & & \downarrow{\scriptstyle F\mu_X} \\
TFX & & \xrightarrow{\lambda_X} & & FTX
\end{array}
$$

Coalgebras

The dual concept of F-algebra is that of an *F-coalgebra*, i.e., an object X together with an arrow $\alpha : X \to FX$. An F-coalgebra can be seen as a way of observing some behavior. For example, if $FX = S \times X$ as above, $\alpha : S^\omega \to S \times S^\omega$ given by $\alpha(h \cdot t) = (h, t)$ is a system which decomposes an infinite list into its head and its tail. The intuition behind coalgebras is that they model systems. The functor F captures the structure of the system. The object X captures the state space and the arrow α captures both the transition behaviour and the observations. It will be clearer in the end of this section, where classical deterministic or non-deterministic automata will be regarded as coalgebras. For more details about the basics intuition of coalgebras, see [9]. A morphism of F-coalgebras $\alpha : X \to FX$ and $\beta : Y \to FY$ is an arrow $f : X \to Y$ such that $Ff \circ \alpha = \beta \circ f$. For a given functor F, the collection of F-coalgebras and their morphisms form a category denoted by **Coalg**(F). This category may have a final object, called the final coalgebra, which is very interesting because it yields a canonical notion of semantics. For example, taking F as above, the final F-algebra is $\alpha : S^\omega \to S \times S^\omega$. This means that for any F-coalgebra $\beta : X \to S \times X$, there exists a unique coalgebra morphism $[\![-]\!]$ from β to α, i.e. $[\![-]\!] : X \to S^\omega$ is such that the following diagram commutes.

8

$$
\begin{array}{ccc}
X & \xrightarrow{\;\llbracket - \rrbracket\;} & A^{\omega} \\
{\scriptstyle \beta}\big\downarrow & & \big\downarrow{\scriptstyle \alpha} \\
A \times X & \xrightarrow[\;id_A \times \llbracket - \rrbracket\;]{} & A \times A^{\omega}
\end{array}
$$

Expressing $\beta = \langle o, t \rangle$ with $o : X \to A$ and $t : X \to X$, we have that $o(x) = \llbracket x \rrbracket(0)$ and $\llbracket t(x) \rrbracket(n) = \llbracket x \rrbracket(n+1)$ so that a direct formula is $\llbracket x \rrbracket(n) = o(t^n(x))$. This fits with the idea of an observable behaviour : what $\llbracket - \rrbracket$ tells is exactly the information we get if we look at the repeated output of β over a state. Actually, the object S^{ω} is the set of streams over S, i.e., of S-valued sequences. The coalgebra β generates over time the data contained in a stream, step by step.

Bisimulation (up-to)

In the framework of coalgebras, the notion of *bisimulation* is a tool which provides a family of proof techniques (see [12]). There is a general definition of bisimulations in any category using diagrams. Since this paper is only using bisimulations for discrete systems, the definition given here is specific to the category **Sets**. It is equivalent to the general definition [8].

Definition 7 (Bisimulation). *Let $F : \textbf{Sets} \to \textbf{Sets}$ be an endofunctor, let X be a set and let $R \subseteq X \times X$ be a relation. The* relation lifting *of R by F is defined by*

$$
Rel(F)(R) = \{(b, c) \in FX \times FX \mid \exists d \in FR, b = F\pi_1(d) \text{ and } c = F\pi_2(d)\}
$$

Given a coalgebra $\alpha : X \to FX$, let $b_\alpha : \mathscr{P}(X \times X) \to \mathscr{P}(X \times X)$ be defined by $b_\alpha(R) = (\alpha \times \alpha)^{-1}(Rel(F)(R))$. The relation R is called a bisimulation *on α if $R \subseteq b_\alpha(R)$.*

The greatest bisimulation on a given coalgebra (X, α) is called bisimilarity and denoted by $\sim$. The following statement expresses the principal interest of using bisimulations: bisimilarity implies behavioural equivalence.

Lemma 8. *Let (X, α) be an F-coalgebra on **Sets**. Assuming there exists a final F-coalgebra (Z, ζ), we have*
$$
\forall x, y \in X, x \sim y \Rightarrow \llbracket x \rrbracket = \llbracket y \rrbracket
$$

Proof. Let $R \subseteq X \times X$ be a bisimulation such that $(x, y) \in R$. For every $(x', y') \in X \times X$, let $d \in FR$ such that $\alpha(x') = F\pi_1(d)$ and $\alpha(y') = F\pi_2(d)$ and set $\gamma(x', y') = d$. The pair (R, γ) is an F-coalgebra. Note that π_1 is a coalgebra morphism from (R, γ) to (X, α) because $(F\pi_1 \circ \gamma)(x', y') = F\pi_1(d) = \alpha(x') = (\alpha \circ \pi_1)(x', y')$. The same is true for π_2. Thus $\llbracket - \rrbracket \circ \pi_1$ and $\llbracket - \rrbracket \circ \pi_2$ are coalgebra morphisms from (R, γ) into the final coalgebra (Z, ζ), hence they are equal. Since $(x, y) \in R$ this yields $\llbracket x \rrbracket = \llbracket y \rrbracket$. $\qquad\square$

It will be seen later that bisimulations incorporate a huge amount of elements. This is the reason why a slightly more refined notion is needed, namely bisimulation up-to.

Definition 9. *Let (X, α) be an F-coalgebra on **Sets** and $g : \mathscr{P}(X \times X) \to \mathscr{P}(X \times X)$ be a function. A relation $R \subseteq X \times X$ is a* bisimulation up-to g *if $R \subseteq b_\alpha(g(R))$.*

For concrete examples of bisimulation (up-to), see the following part about deterministic and non-deterministic automata.

1.3 Automata

We present here the coalgebraic view of automata in the category **Sets**, for a given alphabet A. We begin with Moore automata, which are a slight generalization of deterministic automata. Then we take a look at non-deterministic automata. The notion of bisimulation (up-to) is instantiated to deterministic and non-deterministic automata in order to present algorithms HK and HKC which are useful in section 2, and a few examples are given.

Moore and deterministic automata

Let B be a set. Define the *machine functor F_B* by $F_B X = B \times X^A$ and $Ff = id_B \times f^A$. An F_B-coalgebra models a Moore automaton with output in B. Let β be an F_B-coalgebra. We often denote such a coalgebra by $\beta = \langle o, a \mapsto t_a \rangle$. Each state is mapped to a value via the output function $o : X \to B$ and, for every letter $a \in A$, to a (unique) other state via $t_a : X \to X$. Note that automata, modeled coalgebraically in this way, do not have a notion of initial state. For such automata, it is useful to have a generalized notion of language. From now, a language will be a function $L : A^* \to B$. The language derivative is still defined by $L_a(w) = L(aw)$.

Remark. The notation 2 stands for the set $\{0, 1\}$. An F_2-coalgebra models a *deterministic automaton*. For $x \in X$, the output is 1 when the state is terminating and the output is 0 when it is not. The category $\textbf{Coalg}(F_2)$ will be denoted by **DA**.

The following proposition underlines the main interest of using the machine functor: it has a final coalgebra, consisting of languages.

Proposition 10. *There exists a final F_B-coalgebra (Ω, ω) where $\Omega = B^{A^*}$ and $\omega : B^{A^*} \to F_B B^{A^*}$ is defined by $\omega(L) = \langle L(\varepsilon), a \to L_a \rangle$.*

Proof. Let $\beta = \langle o, a \mapsto t_a \rangle : X \to FX$. We must prove that there exist a unique function $\varphi : X \to B^{A^*}$ such that the following diagram commutes.

$$
\begin{array}{ccc}
X & \xrightarrow{\ \varphi\ } & \Omega \\
\beta \downarrow & & \downarrow \omega \\
FX & \xrightarrow[F\varphi]{} & F\Omega
\end{array}
$$

This means $F\varphi \circ \beta = \omega \circ \varphi$ i.e. $\langle o(x), a \mapsto (\varphi \circ t_a)(x) \rangle = \langle \varphi(x)(\varepsilon), a \mapsto \varphi(x)_a \rangle$. Thus, we define inductively the function φ by

$$\varphi(x)(\varepsilon) = o(x) \qquad\qquad \varphi(x)(aw) = \varphi(t_a(x))(w)$$

By construction it makes the diagram commute. Furthermore, these two last equations are needed for commutation, hence such a morphism is unique. $\qquad\square$

The language accepted by a state $x \in X$ can thus be defined using the final morphism by $[\![x]\!](\varepsilon) = o(x)$ and $[\![x]\!](aw) = [\![t_a(x)]\!](w)$.

Bisimulations (up-to) with the functor F_B

It is important to notice that the notion of bisimulation for F_B-coalgebras is very well-behaved. First, an easy computation shows that for any $R \subseteq X \times X$, there is the simple expression $b_\beta(R) = \{(x, y) \in X \times X \mid o(x) = o(y) \wedge \forall a \in A, (t_a(x), t_a(y)) \in R\}$. Hence, a relation R is a bisimulation iff for all $(x, y) \in R$, $o(x) = o(y)$ and for all $a \in A$, $(t_a(x), t_a(y)) \in R$. Moreover, bisimilarity and trace equivalence coincide, as shown by the following lemma combined with Lemma 8.

Lemma 11. *Let $\beta : X \to F_B X$ be an F_B-coalgebra. Then*

$$\forall x, y \in X, [\![x]\!] = [\![y]\!] \Rightarrow x \sim y$$

Proof. Define by induction $t_\varepsilon = id_X$ and $t_{wa} = t_a \circ t_w$. Set $R = \{(t_w(x), t_w(y)) \mid w \in A^*\}$ and show that R is a bisimulation containing (x, y). This last point is obvious because $(x, y) = (t_\varepsilon(x), t_\varepsilon(y))$. Let $(x', y') \in R$ and $w \in A^*$ such that $x' = t_w(x)$ and $y' = t_w(y)$. Then $o(x') = [\![t_w(x)]\!](\varepsilon) = [\![x]\!](w) = [\![y]\!](w) = [\![t_w(y)]\!](\varepsilon) = o(y')$. Let $a \in A$, then $(t_a(x'), t_a(y')) = (t_{wa}(x), t_{wa}(y)) \in R$, so that R is a bisimulation. Consequently, $x \sim y$. $\qquad\square$

Let $x, y \in X$. The following algorithm $\texttt{Naive}(x, y)$ tries to compute the smallest bisimulation that contains (x, y), this is, $R = \{(t_w(x), t_w(y)) \mid w \in A^*\}$ as in Lemma 11. If R is not a bisimulation, the algorithm will stop at some point. If R exists and is infinite (which requires X to be infinite), then $\texttt{Naive}(x, y)$ never stops.

$$\texttt{Naive}(x, y)$$

```
(1) R := ∅; todo := ∅
(2) insert (x, y) into todo
(3) while todo is not empty do
    (3.1) extract (x′, y′) from todo
    (3.2) if (x′, y′) ∈ R then continue
    (3.3) if o(x′) ≠ o(y′) then return false
    (3.4) for all a ∈ A, insert (t_a(x′), t_a(y′)) into todo
    (3.5) insert (x′, y′) into R
(4) return true
```

Bisimulation is fine but, as we will see, bisimulation up-to g is better - provided g is *compatible* with b_β. Let $g : \mathscr{P}(X \times X) \to \mathscr{P}(X \times X)$. We say that g is compatible with b_β if it is monotone and for all $R, R' \subseteq Y \times Y$, $R \subseteq b_\beta(R') \Rightarrow g(R) \subseteq b_\beta(g(R'))$. Functions that are compatible with b_β have the pleasant following property.

Proposition 12 ([4])**.** *For all $g : X \times X \to X \times X$ compatible with b_β, any bisimulation up-to g is contained into a bisimulation.*

Proof. Let $R \subseteq X \times X$ such that $R \subseteq b_\beta(g(R))$. By compatibility and a simple induction, $g^n(R) \subseteq b_\beta(g^{n+1}(R))$ so that $\bigcup_{n \in \mathbb{N}} g^n(R) \subseteq b_\beta(\bigcup_{n \in \mathbb{N}} g^n(R))$ is a bisimulation that contains R. $\qquad\square$

Hence, if g is compatible with b_β and x, y are related by a bisimulation up-to g, then $x \sim y$.

Lemma 13. *The following functions are compatible with b_β :*

11

- $r : R \mapsto \{(x,x) \mid x \in X\}$
- $s : R \mapsto \{(y,x) \mid (x,y) \in R\}$
- $t : R \mapsto \{(x,z) \mid \exists y \in X, (x,y) \in R \text{ and } (y,z) \in R\}$
- $id : R \mapsto R$
- $f \circ g$ for all f, g compatible with b_β
- $\bigcup_{i \in I} f_i$ for all $(f_i)_{i \in I}$ compatible with b_β
- $f^\omega = \bigcup_{n \in \mathbb{N}} f^n$ for all f compatible with b_β

The last two points of the latter lemma hold in general for any compatible function. This is actually the reason why compatible functions were first introduced.

Up-to equivalence and `HK`

The `Naive` algorithm can be improved using up-to techniques. The principle of the Hopcroft-Karp algorithm (`HK`) is to reason up-to equivalence. This can significantly speed up the `Naive` algorithm in many cases. For a relation $R \subseteq X \times X$, the equivalence closure $e(R)$ is the least equivalence relation that contains R, i.e., that satisfies

$$\frac{(x,y) \in R}{(x,y) \in e(R)} \qquad \frac{}{(x,x) \in e(R)} \qquad \frac{(x,y) \in e(R)}{(y,x) \in e(R)} \qquad \frac{(x,y) \in e(R) \quad (y,z) \in e(R)}{(x,z) \in e(R)}$$

The function e is compatible with b_β because $e = (id \cup r \cup s \cup t)^\omega$. Now replace line (3.2) in `Naive(x,y)` by

(3.2) `if` $(x',y') \in e(\mathtt{R})$ `then` `continue`

This new algorithm, `HK(x,y)`, will stop faster than `Naive(x,y)` because it returns `true` as soon as it has built a bisimulation up-to e. Here is an example in the case of deterministic automata ($B = 2$, the ouput is 1 iff the state is double-circled).
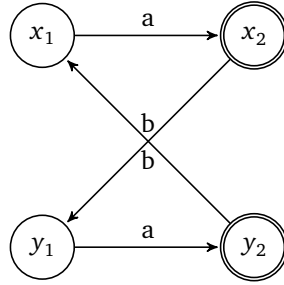
In this case, applying `Naive(`x_1, y_1`)` constructs the smallest bisimulation in 4 steps : $R = \{(x_1,y_1),(x_2,y_2),(y_1,x_1),(y_2,x_2)\}$ and then returns `true`, whereas `HK(`x_1, y_1`)` returns `true` as soon as $R' = \{(x_1,y_1),(x_2,y_2)\}$ because R' is a bisimulation up-to e.

Non-deterministic automata

A non-deterministic automaton is a coalgebra for the composite functor $F_2 \mathscr{P}$. Let α be a non-deterministic automaton, then we can write it as $\alpha = \langle o, t_a \rangle : X \to 2 \times \mathscr{P}(X)^A$. The output o still models termination. The difference with deterministic automata is

the type of $t_a : X \to \mathscr{P}(X)$. The category $\mathbf{Coalg}(F_2\mathscr{P})$ will be denoted by **NDA**. It does not have a final object, but given $\alpha : X \to 2 \times \mathscr{P}(X)^A$, the language of a state $x \in X$ is denoted by $[\![x]\!]_{\mathbf{NDA}}$ and can intuitively be defined by

$$[\![x]\!]_{\mathbf{NDA}}(\varepsilon) = o(x) \qquad\qquad [\![x]\!]_{\mathbf{NDA}}(aw) = \max_{y \in t_a(x)} [\![y]\!]_{\mathbf{NDA}}(w)$$

Bisimilarity and trace equivalence with respect to this semantics do not coincide. One can prove that $R \subseteq X \times X$ is a bisimulation iff for all $(x, y) \in R$,

$$o(x) = o(y)$$
$$\forall x' \in t_a(x), \exists y' \in X, (x', y') \in R$$
$$\forall y' \in t_a(x), \exists x' \in X, (x', y') \in R$$

Consider the following example :
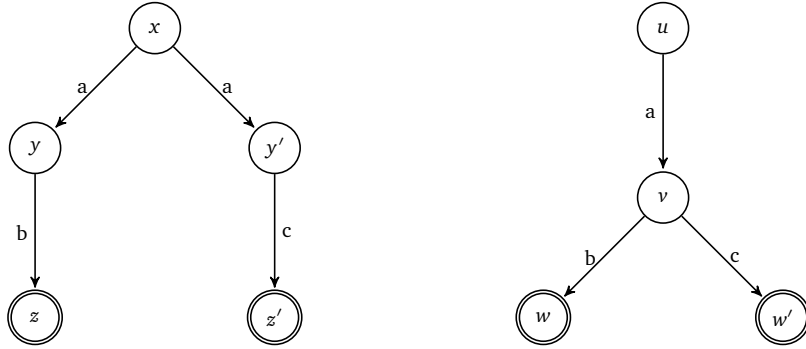
Here $[\![x]\!]_{\mathbf{NDA}} = \{ab, ac\} = [\![u]\!]_{\mathbf{NDA}}$ but x and u are not bisimilar. Indeed, if $x \sim u$ then $y \sim v$ but this is impossible because y has no arrow labeled with c, whereas v has one.

We recall the well-known power set construction : let $\langle o, a \mapsto t_a \rangle : X \to 2 \times \mathscr{P}(X)^A$. We define $\langle o^\#, a \mapsto t_a^\# \rangle : \mathscr{P}(X) \to 2 \times \mathscr{P}(X)^A$ by $o^\#(U) = \max_{x \in U} o(x)$ and $t_a^\#(U) = \bigcup_{x \in U} t_a(x)$. Note that this is a deterministic automaton. Let $[\![-]\!]_{\mathbf{DA}}$ be the final morphism from $\langle o^\#, a \mapsto t_a^\# \rangle$ to the final object in **DA**. The determinized automaton recognizes the same language as the first one, in the sense that for all $x \in X$, $[\![x]\!]_{\mathbf{NDA}} = [\![\{x\}]\!]_{\mathbf{DA}}$.

Up-to congruence and `HKC`

Let α be a non-deterministic automaton. In order to check if $[\![x]\!]_{\mathbf{NDA}} = [\![y]\!]_{\mathbf{NDA}}$, it is possible to first compute the determinized automata $\alpha^\#$ and then check if $[\![\{x\}]\!]_{\mathbf{DA}} = [\![\{y\}]\!]_{\mathbf{DA}}$. According to Lemmas 8 and 11, it is equivalent to check if $\{x\} \sim \{y\}$ in the determinized automata. To this purpose, one can use bisimulation or possibly bisimulation up-to equivalence, but an even better option is to exploit the determinized structure of $\alpha^\#$ with bisimulation up-to congruence. This is what the authors of [4] do, as an improvement of Hopcroft and Karp's algorithm for determinized automata. In section 2, we will draw inspiration from `HKC` to propose an algorithm for the trace semantics of PTS.

Let $R \subseteq \mathscr{P}(X) \times \mathscr{P}(X)$. Its congruence closure $c(R)$ is the least congruence relation that contains R, i.e., that satisfies

$$\frac{(U,V) \in R}{(U,V) \in c(R)} \qquad \overline{(U,U) \in c(R)} \qquad \frac{(U,V) \in c(R)}{(V,U) \in c(R)} \qquad \frac{(V,W) \in c(R) \quad (U,W) \in c(R)}{(U,V) \in c(R)}$$

$$\frac{(U,V) \in c(R) \quad (U',V') \in c(R)}{(U \cup U', V \cup V') \in c(R)}$$

The function c is shown to be compatible with $b_{\alpha^\#}$ in [4]. Thus, if $\{x\}$ and $\{y\}$ are related by a bisimulation up-to congruence, they are bisimilar and this yields $[\![x]\!]_{\mathrm{NDA}} = [\![y]\!]_{\mathrm{NDA}}$. Hence the following algorithm computes the trace equivalence of x and y.

$$\mathrm{HKC}(x,y)$$

(1) R $:= \emptyset$; todo $:= \emptyset$
(2) insert $(\{x\},\{y\})$ into todo
(3) while todo is not empty do
   (3.1) extract (U,V) from todo
   (3.2) if $(U,V) \in c(\mathrm{R})$ then continue
   (3.3) if $o^\#(U) \neq o^\#(V)$ then return false
   (3.4) for all $a \in A$, insert $(t_a^\#(U), t_a^\#(V))$ into todo
   (3.5) insert (U,V) into R
(4) return true

The following example taken from [2] shows how HKC quickens the computations. The alphabet is $A = \{a\}$. The initial non-deterministic automaton is:
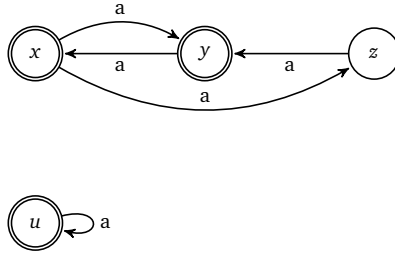
The (interesting part of the) determinized automaton is:
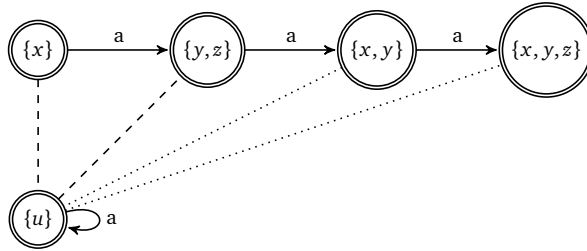
The bisimulation computed by $\mathrm{Naive}(\{x\},\{u\})$ or $\mathrm{HK}(\{x\},\{u\})$ is represented by dotted + dashed lines. The bisimulation up-to congruence computed by $\mathrm{HKC}(x,y)$ consists only in the dashed lines: after two steps, the algorithm HKC returns true.

2 Discrete case

Define the endofunctor $L : \textbf{Sets} \to \textbf{Sets}$ by setting $LX = A \times X + 1$ and $Lf = id_A \times f + id_1$. Remember that (P, η, μ) is the probability distribution monad. We can now formally define PTS from the introduction as PL-coalgebras. For a PTS $\alpha : X \to PLX$, the (finite and infinite) trace semantics $[\![-]\!] : X \to \mathscr{M}(A^{\infty})$ can be defined by the following equations.

$$[\![x]\!](\varepsilon A^{\infty}) = 1 \qquad\qquad\qquad [\![x]\!](\varepsilon) = \alpha(*)$$
$$[\![x]\!](awA^{\infty}) = \sum_{y \in X} \alpha(x)(a, y) \cdot [\![y]\!](wA^{\infty}) \qquad [\![x]\!](aw) = \sum_{y \in X} \alpha(x)(a, y) \cdot [\![y]\!](w)$$

The aim of this section is to find back this semantics via a determinization construction and to provide an efficient algorithm that takes $x, y \in X$ and checks if $[\![x]\!] = [\![y]\!]$. This last part is challenging at first sight, because this means the algorithm checks the equality of two elements of $\mathscr{M}(A^{\infty})$, which are functions on the uncountable space $\Sigma_{A^{\infty}}$.

To begin with, let us get used to this semantics and to measures over words by doing some computations for two different PTS. The first one is defined on the alphabet $A = \{a\}$.
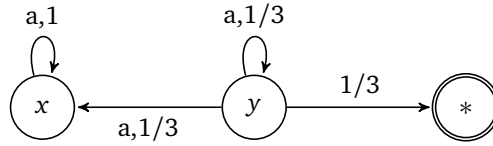

The language $[\![x]\!]$ is very easy to compute for sets of words in S_{∞} by induction: for every finite word w, $[\![x]\!](w) = 0$ and $[\![x]\!](wA^{\infty}) = 1$. Hence, we have for example $[\![x]\!](a^{\omega}) = \lim_{n \to \infty} [\![x]\!](a^n A^{\infty}) = 1$ because the sequence of sets $(a^n A^{\infty})_{n \in \mathbb{N}}$ is decreasing. Let us look at $[\![y]\!]$. First see that $[\![y]\!](\varepsilon) = 1/3$ and $[\![y]\!](\varepsilon A^{\infty}) = 1$. Let $n \in \mathbb{N} \cup \{0\}$, then:

$$[\![y]\!](a^{n+1}) = \frac{1}{3}[\![y]\!](a^n) + \frac{1}{3}[\![x]\!](a^n) = \frac{1}{3}[\![y]\!](a^n)$$
$$[\![y]\!](a^{n+1}A^{\infty}) = \frac{1}{3}[\![y]\!](a^n A^{\infty}) + \frac{1}{3}[\![x]\!](a^n A^{\infty}) = \frac{1}{3}[\![y]\!](a^n A^{\infty}) + \frac{1}{3}$$

Hence $[\![y]\!](a^n) = 1/3^{n+1}$ and $[\![y]\!](a^n A^{\infty}) = (1 + 3^{-n})/2$. Using the same arguments as for $[\![x]\!]$, we have $[\![y]\!](a^{\omega}) = \lim_{n \to \infty}(1 + 3^{-n})/2 = 1/2$. This is rather intuitive: the probability of performing n loops in state y and then getting lost forever in state x is $1/3^{n+1}$. Summing them for $n \in \mathbb{N} \cup \{0\}$ gives $1/2$.

The second PTS is defined on the alphabet $A = \{\textbf{0}, \textbf{1}, \textbf{2}\}$.
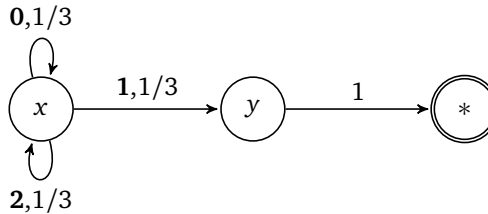
This automaton generates Cantor's space, in the sense that when generating an infinite word, the computation stops if and only if this word is not in Cantor's space (we recall this is the set of real numbers $r \in [0,1]$ such that there is no 1 in the base 3 expansion of r). More precisely, we have $[\![x]\!](w\mathbf{1}) = \left(\frac{1}{3}\right)^{|w|+1}$ if there is no $\mathbf{1}$ in w and $[\![x]\!](w) = 0$ for other finite words. Furthermore $[\![x]\!](wA^\infty) = \left(\frac{1}{3}\right)^{|w|}$ if there is no $\mathbf{1}$ in w. We can guess this way that the measure of Cantor's space in $[0,1]$ is 0 :

$$[\![x]\!](A^\omega) = [\![x]\!](A^\infty \setminus A^*) = [\![x]\!](A^\infty) - [\![x]\!](A^*) = 1 - [\![x]\!]\left(\bigcup_{w \in A^*} \{w\}\right) = 1 - \sum_{w \in A^*} [\![x]\!](w)$$

$$= 1 - \sum_{w \in \{\mathbf{0,2}\}^*} [\![x]\!](w\mathbf{1}) = 1 - \sum_{n \in \mathbb{N}} \sum_{w \in \{\mathbf{0,2}\}^*, |w|=n} \left(\frac{1}{3}\right)^{n+1} = 1 - \frac{1}{3} \sum_{n \in \mathbb{N}} 2^n \left(\frac{1}{3}\right)^n = 0$$

2.1 Trace semantics via determinization

Starting from $\alpha : X \to PLX$, we will proceed in three steps in order to define its trace semantics morphism $[\![-]\!] : X \to \mathcal{M}(A^\infty)$.

(i) Translate α into a coalgebra for the more convenient composite functor $F_{\mathbb{I} \times \mathbb{I}} D$, obtaining an $\tilde{\alpha} : X \to F_{\mathbb{I} \times \mathbb{I}} DX$. Recall that (D, η, μ) is the sub-probability distribution monad. In the sequel $F_{\mathbb{I} \times \mathbb{I}}$ will be denoted by F.

(ii) Determinize it, i.e., define an $\tilde{\alpha}^\# : DX \to FDX$ such that $\tilde{\alpha}^\# \circ \eta_X = \tilde{\alpha}$. Thus there is a final morphism $\varphi_{\tilde{\alpha}^\#} : DX \to \Omega$.

(iii) Factorize the final morphism to get a coalgebra morphism $DX \to \mathcal{M}(A^\infty)$, then precompose with η_X to get the desired trace semantics $X \to \mathcal{M}(A^\infty)$.

The whole construction is summed up in the following diagram. Here $\Omega = (\mathbb{I} \times \mathbb{I})^{A^*}$ is the set of languages with two outputs in $\mathbb{I}$ in accordance with Proposition 10.

(i) Translation: from α to $\tilde{\alpha}$

Let $\alpha : X \to PLX$ be a PL-coalgebra. The space PLX does not enlighten the information provided by α. It would be more convenient to have an explicit machine behaviour, i.e., make functor F appear. The aim of the step (i) is to formally change PLX into FDX.

Each state x gives rise to a probability distribution $\alpha(x)$ over $LX = A \times X + 1$. There are

three important pieces of information in it. First, the information that the total mass is 1. Second, the probability of termination, which is $\alpha(x)(*)$. Third, all the probabilities $\alpha(x)(a, y)$ that the system transitions to some $y \in X$ using some $a \in A$. Any faithful translation of α should keep in memory this information. We define the natural transformations $\iota : P \Rightarrow T$ and $\mathfrak{e} : DL \Rightarrow FD$.

$$\iota_Y : u \in PY \mapsto u \in DY$$

$$\mathfrak{e}_X : u \in DLX \mapsto \left\langle \sum_{z \in LX} u(z), u(*), a \mapsto [y \mapsto u(a, y)] \right\rangle \in FDX$$

The first one is from the inclusion $PY \subseteq DY$. The second one aims to clearly separate the useful information; it appears in [10] where it plays the role of the *extension natural transformation*. Now, take $\tilde{\alpha} = \langle \tilde{\alpha}_{\oplus}, \tilde{\alpha}_*, a \mapsto t_a \rangle = \mathfrak{e}_X \circ \iota_{LX} \circ \alpha$.

$$X \xrightarrow{\quad \alpha \quad} PLX \xrightarrow{\quad \iota_{LX} \quad} DLX \xrightarrow{\quad \mathfrak{e}_X \quad} FDX$$

with $\tilde{\alpha}$ spanning from X to FDX.

An explicit expression is $\tilde{\alpha}(x) = \left\langle \underbrace{\sum_{z \in LX} \alpha(x)(z)}_{=1}, \alpha(x)(*), a \mapsto [y \mapsto \alpha(x)(a, y)] \right\rangle$.

(ii) Determinization

The FD-coalgebra $\tilde{\alpha}$ has now to be changed into an F-coalgebra in order to benefit from the final F-coalgebra semantics. This is the aim of step (ii).

Let $\beta = \langle \beta_{\oplus}, \beta_*, a \to t_a \rangle : X \to FDX$ be any FD-coalgebra. Then, there is an obvious way to determinize it into an F-coalgebra $\beta^{\#} = \langle \beta_{\oplus}^{\#}, \beta_*^{\#}, a \to \tau_a \rangle : DX \to FDX$. Set for all $u \in DX$:

$$\beta_{\oplus}^{\#}(u) = \sum_{x \in X} u(x) \beta_{\oplus}(x)$$

$$\beta_*^{\#}(u) = \sum_{x \in X} u(x) \beta_*(x)$$

$$\forall a \in A \quad \tau_a(u) = y \mapsto \sum_{x \in X} u(x) t_a(x)(y)$$

This transformation is the same as in section 7.2 of [10] if the first output is dropped. The state space of the automata is basically changed from X to DX. In a sense, the function $t_a : X \to DX$ is thus "homogenized" into $\tau_a : DX \to DX$. As an F-coalgebra, $\beta^{\#}$ can be structurally seen as a Moore automaton. It takes $u \in DX$, outputs its total mass, outputs its termination mass, and every letter a makes it transition to a new subprobability distribution $v = \tau_a(u)$.

In this paragraph we explain the terminology "determinization". This refers to the classical power set construction, which consists in changing the state space X of a nondeterministic automaton β into $\mathscr{P}(X)$ to produce a *deterministic* automaton $\beta^{\#}$. This is

actually an instance of a construction discussed in section 3 (see Lemma 27): the general determinization approach involves an Eilenberg-Moore category that we overlook for the moment. The key property is that the behaviour of β starting with x as initial state is the same as the behaviour of $\beta^{\#}$ starting with $\{x\}$ as initial state. Here we are actually doing the same thing, replacing the functor $\mathscr{P}$ with the functor D, and unit $x \mapsto \{x\}$ with unit $x \mapsto \delta_{x,-}$. In terms of diagrams, what has to be checked to confer the title of determinization is that $\beta^{\#} \circ \eta_X = \beta$. This is the case:

$$(\beta_{\oplus}^{\#} \circ \eta_X)(x) = \sum_{x' \in X} \delta_{x,x'} \beta_{\oplus}(x') = \beta_{\oplus}(x)$$

$$(\beta_{*}^{\#} \circ \eta_X)(x) = \sum_{x' \in X} \delta_{x,x'} \beta_{*}(x') = \beta_{*}(x)$$

$$\forall a \in A \quad (\tau_a \circ \eta_X)(x) = \left[y \mapsto \sum_{x' \in X} \delta_{x,x'} t_a(x')(y) \right] = [y \mapsto t_a(x)(y)] = t_a(x)$$

(iii) Factorization of the final morphism

Because $\tilde{\alpha}^{\#}$ is an F-coalgebra, there exists a unique coalgebra morphism $\varphi_{\tilde{\alpha}^{\#}}$ from $\tilde{\alpha}^{\#}$ to ω. However, we want the semantics of $\tilde{\alpha}^{\#}$ to be a probability distribution over words, hence to live in $\mathscr{M}(A^{\infty})$. The step (iii) thus aims to factorize $\varphi_{\tilde{\alpha}^{\#}}$ using a new F-coalgebra $\Pi : \mathscr{M}(A^{\infty}) \to F\mathscr{M}(A^{\infty})$ and a new coalgebra morphism $[\![-]\!]$ from $\tilde{\alpha}^{\#}$ to Π.

$$\begin{array}{ccc} & \varphi_{\tilde{\alpha}^{\#}} & \\ DX \xrightarrow{\quad[\![-]\!]\quad} \mathscr{M}(A^{\infty}) \xrightarrow{\quad\varphi_{\Pi}\quad} & \Omega & (*) \end{array}$$

First, define $\Pi : \mathscr{M}(A^{\infty}) \to F\mathscr{M}(A^{\infty})$ by

$$\Pi(m) = \langle m(\varepsilon A^{\infty}), m(\varepsilon), a \mapsto m_a \rangle$$

The F-coalgebra Π will play the role of a final object in a certain subcategory of **Coalg**(F). The unique coalgebra morphism from Π to ω is denoted by φ_{Π}. It is injective (see Lemma 33). The following lemma states in which cases the factorization is possible.

Proposition 14. *Let* $\beta = \langle \beta_{\oplus}, \beta_{*}, a \mapsto \tau_a \rangle : Y \to FY$ *be an F-coalgebra. The two following conditions are equivalent:*

(i) *There exists an F-coalgebra morphism $[\![-]\!]$ from β to Π.*

(ii) *The equation $\beta_{\oplus} = \beta_{*} + \sum_{a \in A} \beta_{\oplus} \circ \tau_a$ holds.*

In this case, this morphism is unique.

Proof. Refer to the proof of Theorem 34 which is the same in the general case. $\square$

Remark. If (i) and (ii) are true, then $\varphi_{\Pi} \circ [\![-]\!]$ is a coalgebra morphism from β to ω, hence $\varphi_{\Pi} \circ [\![-]\!] = \varphi_{\beta}$ by uniqueness. Using the injectivity of φ_{Π}, one can see that for any $u, v \in TX$, $[\![u]\!] = [\![v]\!] \Longleftrightarrow \varphi_{\beta}(u) = \varphi_{\beta}(v)$. This is important because checking if $\varphi_{\beta}(u) = \varphi_{\beta}(v)$ is made easy by the use bisimulation up-to techniques.

Let us check that $\tilde{\alpha}^{\#} : DX \to FDX$ satisfies (ii) in Proposition 14.

$$\tilde{\alpha}_{\oplus}^{\#}(u) = \sum_{x \in X} u(x) \tilde{\alpha}_{\oplus}(x) = \sum_{x \in X} u(x) \sum_{z \in LX} \alpha(x)(z)$$

$$= \sum_{x \in X} u(x) \alpha(x)(*) + \sum_{x \in X} \sum_{(a,y) \in A \times X} u(x) \alpha(x)(a, y)$$

$$= \sum_{x \in X} u(x) \tilde{\alpha}_{*}(x) + \sum_{a \in A} \sum_{y \in X} \tau_a(u)(y)$$

$$= \tilde{\alpha}_{*}^{\#}(u) + \sum_{a \in A} (\tilde{\alpha}_{\oplus}^{\#} \circ \tau_a)(u) \qquad\qquad \text{(because } \tilde{\alpha}_{\oplus}(y) = 1\text{)}$$

This achieves the proof of the following proposition.

Proposition 15. *The morphism $\varphi_{\tilde{\alpha}^{\#}}$ decomposes as a unique coalgebra morphism $[\![-]\!]$ from $\tilde{\alpha}^{\#}$ to Π followed by an injective coalgebra morphism φ_{Π} from Π to ω, as in $(*)$.*

We define $[\![\,-\,]\!] = [\![-]\!] \circ \eta_X$ as our definitive semantics.

Link with the usual semantics

We conclude this section by comparing the trace semantics via determinization we just obtained with the trace semantics of PTS defined previously. First express $[\![-]\!]_{|S_{\infty}}$ (see the proof of Theorem 34). For all $u \in DX$, $a \in A$, $w \in A^*$:

$$[\![u]\!](\varepsilon A^{\infty}) = \tilde{\alpha}_{\oplus}^{\#}(u) \qquad\qquad\qquad [\![u]\!](\varepsilon) = \tilde{\alpha}_{*}^{\#}(u)$$

$$[\![u]\!](awA^{\infty}) = [\![\tau_a(u)]\!](wA^{\infty}) \qquad\qquad [\![u]\!](aw) = [\![\tau_a(u)]\!](w)$$

The following lemma makes a rather intuitive link between $[\![-]\!]$ and $[\![\,-\,]\!]$ and will help us to get back to the first semantics defined in the beginning of this section.

Lemma 16. *Let $u \in DX$. Then $[\![u]\!] = \sum_{x \in X} u(x) [\![\,x\,]\!]$.*

Proof. By induction on w (for all u). First see that $[\![u]\!](\varepsilon) = \tilde{\alpha}_{*}^{\#}(u) = \sum_{x \in X} u(x) \tilde{\alpha}_{*}(x) = \sum_{x \in X} u(x) [\![\,x\,]\!](\varepsilon)$. Then $[\![u]\!](aw) = [\![\tau_a(u)]\!](w) = \sum_{y \in X} \tau_a(u)(y) [\![\,y\,]\!](w)$ by induction hypothesis. So $[\![u]\!](aw) = \sum_{x,y \in X} u(x) t_a(x)(y) [\![\,y\,]\!](w) = \sum_{x \in X} u(x) [\![t_a(x)]\!](w)$ again by induction hypothesis. But $[\![t_a(x)]\!](w) = [\![(\tau_a \circ \eta_X)(x)]\!](w) = [\![\,x\,]\!](aw)$ because $[\![-]\!]$ is a coalgebra morphism, so $[\![u]\!](aw) = \sum_{x \in X} u(x) [\![\,x\,]\!](aw)$. The proof is the same for $[\![u]\!](wA^{\infty}) = \sum_{x \in X} u(x) [\![\,x\,]\!](wA^{\infty})$. Then, both measures coincide on S_{∞} so $[\![u]\!] = \sum_{x \in X} u(x) [\![\,x\,]\!]$ according to Theorem 5. $\qquad\square$

Using this lemma we have the following expressions.

$$[\![\,x\,]\!](\varepsilon A^{\infty}) = (\tilde{\alpha}_{\oplus}^{\#} \circ \eta_X)(x) = \tilde{\alpha}_{\oplus}(x) = \sum_{z \in X} \alpha(x)(z) = 1$$

$$[\![\,x\,]\!](\varepsilon) = (\tilde{\alpha}_{*}^{\#} \circ \eta_X)(x) = \tilde{\alpha}_{*}(x) = \alpha(*)$$

$$[\![\,x\,]\!](awA^{\infty}) = [\![(\tau_a \circ \eta_X)(x)]\!](wA^{\infty}) = [\![t_a(x)]\!](wA^{\infty}) = \sum_{y \in X} \alpha(x)(a, y) \cdot [\![\,y\,]\!](wA^{\infty})$$

$$[\![\,x\,]\!](aw) = [\![(\tau_a \circ \eta_X)(x)]\!](w) = [\![t_a(x)]\!](w) = \sum_{y \in X} \alpha(x)(a, y) \cdot [\![\,y\,]\!](w)$$

These are the same expressions as in the beginning of this section. Hence:

Proposition 17. *The two trace semantics denoted by $[\![\,-\,]\!]$ coincide.*

19

2.2 Algorithm

Let $\alpha : X \to PLX$ and $x, y \in X$. The aim of this paragraph is to give an algorithm that takes $x, y \in X$ and tells whether $[\![x]\!] = [\![y]\!]$ or not. It will directly profit from the determinization construction performed in the last section. Our algorithm is greatly inspired by HKC, and more specifically, by its generalization to weighted automata given in [3]. Indeed, the very same bisimulation up-to congruence techniques can leverage the determinized structure of $\tilde{\alpha}^{\#}$.

Note that $\tilde{\alpha}^{\#}$ has type $DX \to \mathbb{I} \times \mathbb{I} \times (DX)^A$. Let $\varphi_{\tilde{\alpha}^{\#}}$ be the final morphism $DX \to (\mathbb{I} \times \mathbb{I})^{A^*}$. The $F_{\mathbb{I} \times \mathbb{I}}$-coalgebra $\tilde{\alpha}^{\#}$ can be easily injected into the "bigger" $F_{\mathbb{R} \times \mathbb{R}}$-coalgebra $\beta : \mathbb{R}_\omega^X \to \mathbb{R} \times \mathbb{R} \times (\mathbb{R}_\omega^X)^A$ where $\mathbb{R}_\omega^X$ is the set of finitely supported functions $X \to \mathbb{R}$. Thus we get a vector space structure (see [2]). Indeed, just define for every $u \in \mathbb{R}_\omega^X$:

$$\beta(u) = \left\langle \sum_{x \in X, a \in A, y \in X} u(x)\alpha(x)(a, y), \sum_{x \in X} u(x)\alpha(x)(*), a \mapsto \left[y \mapsto \sum_{x \in X} u(x)\alpha(x)(a, y) \right] \right\rangle$$

We study bisimulations in the sense of $\beta = \langle \beta_\oplus, \beta_*, a \mapsto \tau_a \rangle : \mathbb{R}_\omega^X \to F\mathbb{R}_\omega^X$. Note that $\beta_\oplus, \beta_*$ and each τ_a are linear functions. Let φ_β be the final morphism $\mathbb{R}_\omega^X \to (\mathbb{R} \times \mathbb{R})^{A^*}$. The following lemma shows that changing $\tilde{\alpha}^{\#}$ into β does not affect trace equivalence.

Lemma 18. *Let i be the injection $DX \to \mathbb{R}_\omega^X$ and j be the injection $(\mathbb{I} \times \mathbb{I})^{A^*} \to (\mathbb{R} \times \mathbb{R})^{A^*}$. Then $j \circ \varphi_{\alpha^{\#}} = \varphi_\beta \circ i$.*

Now we omit writing i and j. The following corollary allows us to work with bisimulation up-to congruence on $\mathbb{R}_\omega^X$, as defined below.

Corollary 19. *For every $x, y \in X$, $[\![x]\!] = [\![y]\!]$ iff $\varphi_\beta(\delta_x) = \varphi_\beta(\delta_y)$.*

Let $R \subseteq \mathbb{R}_\omega^X \times \mathbb{R}_\omega^X$. Its congruence closure $c(R)$ is the least congruence relation that contains R, i.e., that satisfies

$$\frac{(u, v) \in R}{(u, v) \in c(R)} \qquad \frac{}{(u, u) \in c(R)} \qquad \frac{(u, v) \in c(R)}{(v, u) \in c(R)} \qquad \frac{(u, v) \in c(R) \quad (v, w) \in c(R)}{(u, w) \in c(R)}$$

$$\frac{(u, v) \in c(R)}{(\lambda u, \lambda v) \in c(R)} \; (\lambda \in \mathbb{R}) \qquad \frac{(u, u') \in c(R) \quad (v, v') \in c(R)}{(u + u', v + v') \in c(R)}$$

Lemma 20. *The functions $a : R \mapsto \{z + z' \mid z \in R, z' \in R\}$ and $m_\lambda : R \mapsto \{\lambda z \mid z \in R\}$ are compatible with b_β for all $\lambda \in \mathbb{R}$.*

Proof. Let $\lambda \in \mathbb{R}$. If $R \subseteq b_\beta(R')$ and we take $(u', v') \in m_\lambda(R)$ then there exists $(u, v) \in R$ such that $(u', v') = (\lambda u, \lambda v)$. We have $(u, v) \in b_\beta(R')$ so $o(u') = o(\lambda u) = \lambda o(u) = \lambda o(v) = o(\lambda v) = o(v')$. Given $a \in A$ we have $(\tau_a(u), \tau_a(v)) \in R'$ so $(\tau_a(u'), \tau_a(v')) = (\lambda \tau_a(u), \lambda \tau_a(v)) \in m_\lambda(R')$. Thus $m_\lambda(R) \subseteq b_\beta(m_\lambda(R'))$.
Let $(u'', v'') \in a(R)$, so $(u'', v'') = (u + u', v + v')$ where $(u, v) \in R$ and $(u', v') \in R$. So $o(u'') = o(u) + o(u') = o(v) + o(v') = o(v'')$ and for all $a \in A$, $(\tau_a(u''), \tau_a(v'')) = (\tau_a(u) + \tau_a(u'), \tau_a(v) + \tau_a(v')) \in a(R')$. Thus $a(R) \subseteq b_\beta(a(R'))$. $\qquad\square$

According to Lemmas 13 and 20, the function $c : \mathscr{P}(\mathbb{R}_\omega^X \times \mathbb{R}_\omega^X) \to \mathscr{P}(\mathbb{R}_\omega^X \times \mathbb{R}_\omega^X)$ is compatible with b_β, because $c = \left(id \cup r \cup s \cup t \cup a \cup \bigcup_{\lambda \in \mathbb{R}} m_\lambda \right)^\omega$. Thus, if $u, v \in \mathbb{R}_\omega^X$ are related by a bisimulation up-to congruence, they are bisimilar, and this yields $\varphi_\beta(u) =$

$\varphi_\beta(v)$. In our case, when $u = \delta_x$ and $v = \delta_y$ for some $x, y \in X$, this further yields $[\![x]\!] = [\![y]\!]$. The following algorithm HKC^∞ computes the smallest bisimulation that relates x and y, hence it computes whether $[\![x]\!] = [\![y]\!]$ or not.

$$\text{HKC}^\infty(x, y)$$

```
(1) R := ∅; todo := ∅
(2) insert (δ_x, δ_y) into todo
(3) while todo is not empty do
    (3.1) extract (u,v) from todo
    (3.2) if (u,v) ∈ c(R) then continue
    (3.3) if β_⊕(u) ≠ β_⊕(v) then return false
    (3.3') if β_*(u) ≠ β_*(v) then return false
    (3.4) for all a ∈ A, insert (τ_a(u), τ_a(v)) into todo
    (3.5) insert (u,v) into R
(4) return true
```

Theorem 21 (inspired from theorem 4.3 in [3]). *Whenever* $\text{HKC}^\infty(x, y)$ *terminates, it returns* true *iff* $[\![x]\!] = [\![y]\!]$.

Proof. Observe that $R \subseteq b_\beta(c(R) \cup todo)$ is an invariant for the while loop at step (3). If HKC^∞ returns true then todo is empty and thus $R \subseteq b_\beta(c(R))$ so R is a bisimulation up-to c that contains (δ_x, δ_y) and we already know that this yields $[\![x]\!] = [\![y]\!]$. If HKC^∞ returns false, it encounters a pair (u, v) such that $\beta_\diamond(u) \neq \beta_\diamond(v)$ for a certain $\diamond \in \{\oplus, *\}$. There exists a word w such that $u = \tau_w(\delta_x)$ and $v = \tau_w(\delta_y)$. Therefore $\varphi_\beta(\delta_x)(w) = \varphi_\beta(\tau_w(\delta_x))(\varepsilon) = \beta_\diamond(u) \neq \beta_\diamond(v) = \varphi_\beta(\tau_w(\delta_y))(\varepsilon) = \varphi_\beta(\delta_y)(w)$ so $\varphi_\beta(\delta_x) \neq \varphi_\beta(\delta_y)$ and we know that this implies $[\![x]\!] \neq [\![y]\!]$. $\square$

Despite the fact that during the determinization the state space always becomes infinite, the following results show that if the initial state space X is finite, then HKC^∞ does terminate.

Theorem 22 (see [5]). *Let* $\mathcal{R}$ *be a ring and* X *be a finite set. Let* $R \subseteq \mathcal{R}^X \times \mathcal{R}^X$ *be a relation and let* $(v, v') \in \mathcal{R}^X \times \mathcal{R}^X$ *be a pair of vectors. We construct a generating set for a submodule of* $\mathcal{R}^X$ *by defining* $U_R = \{u - u' \mid (u, u') \in R\}$. *Then* $(v, v') \in c(R)$ *iff* $v - v' \in [U_R]$.

Proposition 23. *If* X *is finite,* $\text{HKC}^\infty(x, y)$ *terminates for every* $x, y \in \mathbb{R}^X_\omega$.

Proof. The set X is finite so $\mathbb{R}^X_\omega = \mathbb{R}^X$. Suppose we have an infinite computation and let (u_n, v_n) be the pair checked at step n. The rank of any family of vectors of $\mathbb{R}^X$ is bounded by $|X| < \infty$ and denoted by *rank*. When the stationary sequence $(rank\{(u_i - v_i)_{i \leq n}\})_{n \in \mathbb{N}}$ has reached its limit at step N, then the rest of the sequence is in the submodule of $\mathbb{R}^X$ generated by the first $(u_n - v_n)_{n \leq N}$. According to Theorem 22 with $\mathcal{R} = \mathbb{R}$, this means that the $(u_n, v_n)_{n \geq N+1}$ are in the congruence closure of the relation $\{(u_n, v_n) \mid n \leq N\}$. So at each step the test (3.2) succeeds and the cardinality of todo is decreased by 1. This cardinality is finite as step N so the number of following steps is finite too, yielding a contradiction. $\square$

Theorem 24 (Summary). *For any* $\alpha : X \to PLX$ *with* X *finite, and any* $x, y \in X$, $\text{HKC}^\infty(x, y)$ *terminates and returns* true *iff* $[\![x]\!] = [\![y]\!]$.

21

Basic example of HKC$^\infty$

To begin with, here is a very simple PTS that demonstrates the power of bisimulation up-to congruence.
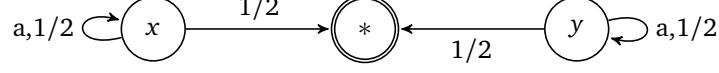
If we try to compute $\text{HK}(\delta_x, \delta_y)$ on the determinized automaton, the algorithm encounters the pairs $(\delta_x/2^k, \delta_y/2^k), k \geq 0$ and never stops. On the other hand, $\text{HKC}^\infty(x, y)$ stops after one step because it has immediately spotted that $(\delta_x/2, \delta_y/2)$ is in the congruence closure of the relation $\{(\delta_x, \delta_y)\}$.

Example of HKC$^\infty$ with bisimulation up-to as an accelerant

Let us compute HKC^∞ to know if in the following automaton the states x and z generate the same probability measure. Here $A = \{a\}$.
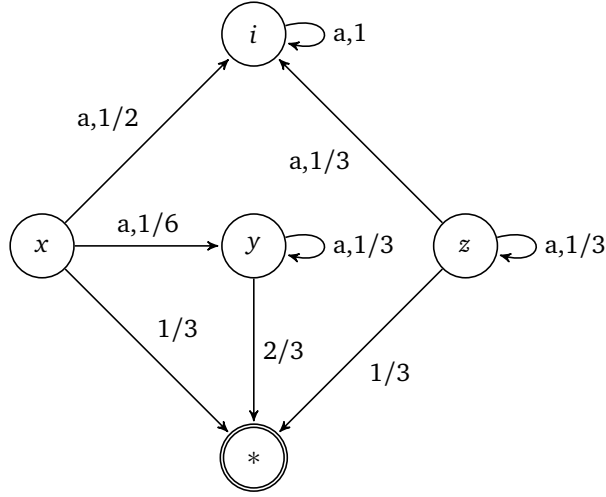
Because X is finite, $\mathbb{R}^X_\omega$ has a basis (e_x, e_y, e_z, e_i). An element $u \in \mathbb{R}^X_\omega$ is seen a column vector $u_x e_x + u_y e_y + u_z e_z + u_i e_i$ in this basis. Moreover $\tilde{\alpha}^{\#}_\oplus$ and $\tilde{\alpha}^{\#}_*$ are linear forms that can be seen as the row vectors $L_\oplus = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$ and $L_* = \begin{pmatrix} 1/3 & 2/3 & 1/3 & 0 \end{pmatrix}$, and τ_a is an endomorphism with a transition matrix M_a defined by $(M_a)_{j,k} = t_a(k)(j)$.

$$ u = \begin{pmatrix} u_x \\ u_y \\ u_z \\ u_i \end{pmatrix} \qquad M_a = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/6 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/3 & 1 \end{pmatrix} \qquad L = \begin{pmatrix} L_\oplus \\ L_* \end{pmatrix} $$

The algorithm begins with $\texttt{todo} = \{(\eta_X(x), \eta_X(z))\} = \{(e_x, e_z)\}$ and $\texttt{R} = \emptyset$. It checks that $Le_x = Le_z$, etc. as shown in the following table.

Step	(3.1)	(3.2)	(3.3)	(3.4)	(3.5)
Loop counter	(u,v) extracted from `todo`	Check $(u,v) \in c(R)$	Check $Lu = Lv$	$(M_a u, M_a v)$ added to `todo`	Cardinality of R
1	$(\begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\\0\end{pmatrix})$	Fail	$\begin{pmatrix}1\\1/3\end{pmatrix} = \begin{pmatrix}1\\1/3\end{pmatrix}$	$(\begin{pmatrix}0\\1/6\\0\\1/2\end{pmatrix}, \begin{pmatrix}0\\0\\1/3\\1/3\end{pmatrix})$	1
2	$(\begin{pmatrix}0\\1/6\\0\\1/2\end{pmatrix}, \begin{pmatrix}0\\0\\1/3\\1/3\end{pmatrix})$	Fail	$\begin{pmatrix}2/3\\1/9\end{pmatrix} = \begin{pmatrix}2/3\\1/9\end{pmatrix}$	$(\begin{pmatrix}0\\1/18\\0\\1/2\end{pmatrix}, \begin{pmatrix}0\\0\\1/9\\4/9\end{pmatrix})$	2
3	$(\begin{pmatrix}0\\1/18\\0\\1/2\end{pmatrix}, \begin{pmatrix}0\\0\\1/9\\4/9\end{pmatrix})$	Success	/	/	2
4	Empty	/	/	/	/

The check succeeds in loop 3 because $(u,v) \in c(R)$ according to theorem 22:

$$\begin{pmatrix}0\\1/18\\0\\1/2\end{pmatrix} - \begin{pmatrix}0\\0\\1/9\\4/9\end{pmatrix} = \begin{pmatrix}0\\1/18\\-1/9\\1/18\end{pmatrix} = \frac{1}{3}\begin{pmatrix}0\\1/6\\-1/3\\1/6\end{pmatrix} = \frac{1}{3}\left(\begin{pmatrix}0\\1/6\\0\\1/2\end{pmatrix} - \begin{pmatrix}0\\0\\1/3\\1/3\end{pmatrix}\right)$$

Because `todo` is eventually empty, the algorithm returns `true`. Indeed, if we compute directly the measures $[\![x]\!]$ and $[\![z]\!]$, we can see that $[\![x]\!](a^n) = 1/3^{n+1}$, $[\![x]\!](a^\omega) = 1/2$ and same for $[\![z]\!]$. Here the bisimulation up to congruence check is necessary for termination. See what is happening in the determinized infinite automaton.

$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}0\\1/6\\0\\1/2\end{pmatrix} \xrightarrow{a} \begin{pmatrix}0\\1/18\\0\\1/2\end{pmatrix} \xrightarrow{a} \cdots \xrightarrow{a} \begin{pmatrix}0\\1/(2\times 3^n)\\0\\1/2\end{pmatrix} \xrightarrow{a} \cdots$$

$$\begin{pmatrix}0\\0\\1\\0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}0\\0\\1/3\\1/3\end{pmatrix} \xrightarrow{a} \begin{pmatrix}0\\0\\1/9\\4/9\end{pmatrix} \xrightarrow{a} \cdots \xrightarrow{a} \begin{pmatrix}0\\0\\1/3^n\\(1-3^{-n})/2\end{pmatrix} \xrightarrow{a} \cdots$$

The construction of the bisimulation up to equivalence (dashed + dotted lines) would take an infinite number of steps. But the construction of the bisimulation up to congruence (dashed lines) takes only 2 steps.

Example of $\mathtt{HKC}^\infty$ where infinite traces matter

Get back to the example of the introduction, taking $A = \{a, b\}$.

a,1/2 $\circlearrowright$ (y) $\circlearrowleft$ b,1/2   a,3/4 $\circlearrowright$ (z) $\circlearrowleft$ b,1/4

23

Take (e_y, e_z) as a basis of $\mathbb{R}^X$. Then

$$u = \begin{pmatrix} u_y \\ u_z \end{pmatrix} \qquad M_a = \begin{pmatrix} 1/2 & 0 \\ 0 & 3/4 \end{pmatrix} \qquad M_b = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/4 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

In the first loop, everything is fine because $L \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = L \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Then $\texttt{todo} = \left\{ \left(\begin{pmatrix} 1/2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 3/4 \end{pmatrix} \right), \left(\begin{pmatrix} 1/2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1/4 \end{pmatrix} \right) \right\}$ so both possible next equality checks fail: $\begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \neq \begin{pmatrix} 3/4 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \neq \begin{pmatrix} 1/4 \\ 0 \end{pmatrix}$. Thus, $\texttt{HKC}^\infty$ returns $\texttt{false}$. Remark that not caring about the first line of L amounts to use $\texttt{HKC}$, which returns $\texttt{true}$ because it does not take into account infinite words.

3   General case

In this section, we generalize the trace semantics previously defined. We work with continuous PTS, defined later as coalgebras for the analogue of functor PL in the category **Meas**. The underlying distributive law is brought to light, so that the origin of the determinization process is better understood. The following table sums up the analogies and differences with the discrete case.

	Discrete case	General case
Category	**Sets**	**Meas**
Usual operation	$\sum$	$\int$
Language functor	L	Measurable version of L
Machine functor	F	Measurable version of F
Probability monad	Probability monad P	Giry's monad $\mathbb{P}$
Determinization monad	Sub-probability monad D	Sub-Giry's monad $\mathbb{D}$
Initial state space	Set X	Measurable space (X, Σ_X)
Determinized state	Finitely supported vector	Finite measure (≤ 1)
Transitions	Matrix $t_a : X \times X \to \mathbb{I}$	Kernel $t_a : X \times \Sigma_X \to \mathbb{I}$
Final F-coalgebra	ω	Measurable version of ω
Measure coalgebra	Π	Measurable version of Π
Pseudo-final morphism	$[\![-]\!] : DX \to \mathscr{M}(A^\infty)$	$[\![-]\!] : \mathbb{D}X \to \mathbb{D}A^\infty$

In this section we work in the category **Meas** of measurable spaces and functions. It is easy to adapt functors L and F, but considering the monads we will need some additional measure-theoretic background.

Given measurable spaces X, Y and a measurable function $f : X \to Y$, define $LX = A \times X + 1$ along with its σ-algebra $\Sigma_{LX} = \mathscr{P}(A) \otimes \Sigma_X \oplus \mathscr{P}(1)$, and $Lf = id_A \times f + id_1$. Moreover, define $FX = \mathbb{I} \times \mathbb{I} \times X^A$ along with its σ-algebra $\mathscr{B}(\mathbb{I}) \otimes \mathscr{B}(\mathbb{I}) \otimes \bigotimes_{a \in A} \Sigma_X$ and $Ff = id_\mathbb{I} \times id_\mathbb{I} \times f^A$.

Integration

Let (X, Σ_X, m) be a measure space and $f : X \to \mathbb{R}$ be a measurable function. If $f(X) = \{\alpha_1, ... \alpha_n\}$ for some $\alpha_1, ... \alpha_n \in \mathbb{R}_+$, then f is called a *simple* function and its integral can be defined as $\int_X f \, dm = \sum_{i=1}^n \alpha_i m(f^{-1}(\{\alpha_i\}))$. If $f \geq 0$ pointwise, define

$\int_X f \, dm = \sup \left\{ \int_X g \, dm \mid g \le f, g \text{ simple} \right\} \in [0, \infty]$. Finally, for any $f : X \to \mathbb{R}$, decompose $f = f^+ - f^-$ where $f^+ \ge 0$ and $f^- \ge 0$. If their integrals are not both ∞, define $\int_X f \, dm = \int_X f^+ dm - \int_X f^- dm$. If this is finite, we say that f is m-integrable. Furthermore, for any $S \in \Sigma_X$, the indicator function $\mathbf{1}_S$ is measurable and we define $\int_S f \, dm = \int_X \mathbf{1}_S f \, dm$.

Given any measurable function $g : X \to Y$ and any measure $m : \Sigma_X \to \mathbb{R}_+$, the *image measure* of m by g is $m \circ g^{-1}$. For any measurable $f : Y \to \mathbb{R}$, f is $m \circ g^{-1}$-integrable iff $f \circ g$ is m-integrable and in this case, $\int_Y f \, d(m \circ g^{-1}) = \int_X (f \circ g) dm$.

Actually, each positive measurable function $X \to \mathbb{R}_+$ is the pointwise limit of an increasing sequence of simple functions. In order to prove some property for every positive measurable function, one can prove it for simple functions (or for indicator functions, if it is preserved by linear combinations) and show that the property is preserved when taking the limit. Many such proofs use the monotone convergence theorem, which states that if $(f_n)_{n \in \mathbb{N}}$ is an increasing sequence of positive functions with pointwise limit f, then f is measurable and $\int_X f \, dm = \lim \int_X f_n dm$.

The Giry monad

The Giry monad [6] provides a link between probability theory and category theory.

Definition 25. *In **Meas**, the Giry monad $(\mathbb{P}, \eta, \mu)$ is defined as follows. For any measurable space X, $\mathbb{P}X$ is the space of probability measures over (X, Σ_X), and $\Sigma_{\mathbb{P}X}$ is the σ-algebra generated by the functions $e_S^X : \mathbb{P}X \to \mathbb{I}$ defined by $e_S^X(m) = m(S)$. For any measurable function $g : X \to Y$, $(\mathbb{P}g)(m) = m \circ g^{-1}$.*
The unit is defined by $\eta_X(x)(S) = \mathbf{1}_S(x)$ and the multiplication by $\mu_X(\Phi)(S) = \int_{\mathbb{P}X} e_S^X d\Phi$.
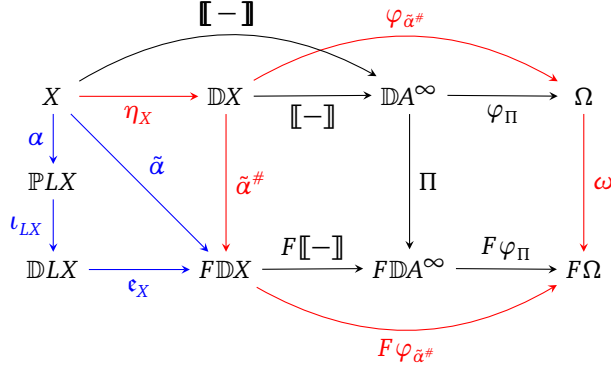
In the same way, one can define the sub-Giry monad $(\mathbb{D}, \eta, \mu)$. The only difference is that $\mathbb{D}X$ is then the space of sub-probability measures over (X, Σ_X). There is a natural transformation $\iota : \mathbb{P} \Rightarrow \mathbb{D}$ that comes from the inclusion : $\iota_X(m) = m$.

3.1 Trace semantics via determinization

The aim of this section is to define trace semantics for continuous PTS, i.e., coalgebras of the form $\alpha : X \to \mathbb{P}LX$. We proceed in the same way as for discrete systems.

(i) Transform α into a more convenient coalgebra $\tilde{\alpha} : X \to F\mathbb{D}X$.

(ii) Determinize $\tilde{\alpha}$ into an F-coalgebra $\tilde{\alpha}^\# : \mathbb{D}X \to F\mathbb{D}X$.

(iii) Factorize the final morphism : $\varphi_{\tilde{\alpha}^\#} = \varphi_\Pi \circ [\![-]\!]$ and take $[\![-]\!] = [\![-]\!] \circ \eta_X$.

The following diagram sums up the construction. Here $\Omega = (\mathbb{I} \times \mathbb{I})^{A^*}$ and Σ_Ω is the Σ-algebra generated by the functions $L \mapsto L(w)$.

$$
\begin{array}{ccccccc}
& & \xrightarrow{\quad[\![-]\!]\quad} & & \xrightarrow{\quad\varphi_{\tilde{\alpha}^{\#}}\quad} & & \\
X & \xrightarrow{\eta_X} & \mathbb{D}X & \xrightarrow{[\![-]\!]} & \mathbb{D}A^{\infty} & \xrightarrow{\varphi_{\Pi}} & \Omega \\
\downarrow{\alpha} & \searrow^{\tilde{\alpha}} & \downarrow{\tilde{\alpha}^{\#}} & & \downarrow{\Pi} & & \downarrow{\omega} \\
\mathbb{P}LX & & & & & & \\
\downarrow{\iota_{LX}} & & & & & & \\
\mathbb{D}LX & \xrightarrow{\mathfrak{e}_X} & F\mathbb{D}X & \xrightarrow{F[\![-]\!]} & F\mathbb{D}A^{\infty} & \xrightarrow{F\varphi_{\Pi}} & F\Omega \\
& & & \xrightarrow{\quad F\varphi_{\tilde{\alpha}^{\#}}\quad} & & &
\end{array}
$$

(i) Translation: from α to $\tilde{\alpha}$

Proposition 26. *For any measurable space X, the function* $\mathfrak{e}_X : \mathbb{D}LX \to F\mathbb{D}X$ *defined by*

$$\mathfrak{e}_X(m) = \langle m(LX), m(1), a \mapsto [S \mapsto m(\{a\} \times S)]\rangle$$

is measurable. Moreover, $\mathfrak{e} : \mathbb{D}L \Rightarrow F\mathbb{D}$ *is a natural transformation.*

Proof. First, see that this is a measurable function. Note that we can write $\mathfrak{e}_X = \langle e^{LX}_{LX}, e^{LX}_1, a \mapsto \phi_a\rangle$, where $\phi_a : \mathbb{D}LX \to \mathbb{D}X$ is defined by $\phi_a(m)(S) = m(\{a\} \times S)$. According to Lemma 2, it suffices to prove that the $|A| + 2$ components are measurable functions. The first two are, by definition of $\Sigma_{\mathbb{D}X}$. Given $a \in A$, use Lemma 1 and see that for any $S \in \Sigma_X$, $e^X_S \circ \phi_a = e^{LX}_{\{a\}\times S}$ is measurable because $\{a\} \times S \in \Sigma_{LX}$. For naturality, let $f : X \to Y$ be a morphism. Note that if $B_A \in \Sigma_A$, $B_Y \in \Sigma_Y$ and $B_1 \in \Sigma_1$, then $Lf^{-1}(B_A \times B_Y + B_1) = B_A \times f^{-1}(B_Y) + B_1$. Thus, $\mathfrak{e}$ is a natural transformation because

$$
\begin{aligned}
(\mathfrak{e}_Y \circ \mathbb{D}Lf)(m) &= \langle m(Lf^{-1}(A \times Y + 1)), m(Lf^{-1}(1)), a \mapsto [S \mapsto m(Lf^{-1}(\{a\} \times S))]\rangle \\
&= \langle m(A \times X + 1), m(1), a \mapsto [S \mapsto m(\{a\} \times f^{-1}(S))]\rangle \\
&= \langle e^{LX}_{LX}, e^{LX}_1, a \mapsto \mathbb{D}f \circ \phi_a\rangle(m) = (F\mathbb{D}f \circ \mathfrak{e}_X)(m)
\end{aligned}
$$

$\square$

Now take $\tilde{\alpha} = \mathfrak{e}_X \circ \iota_{LX} \circ \alpha$. We will write its components $\langle \tilde{\alpha}_{\oplus}, \tilde{\alpha}_{*}, a \mapsto t_a\rangle$. A direct expression of $\tilde{\alpha}$ is given by $\tilde{\alpha}(x) = \langle \underbrace{\alpha(x)(LX)}_{1}, \alpha(x)(1), a \mapsto [S \mapsto \alpha(x)(\{a\} \times S)]\rangle$.

(ii) Determinization

The following lemma (inspired from [13]) establishes some conditions under which there is a canonical notion of determinization.

Lemma 27. *Let* $\mathbf{C}$ *be a category,* $F : \mathbf{C} \to \mathbf{C}$ *be an endofunctor and* (T, η, μ) *be a monad on* $\mathbf{C}$*. Let* $f : X \to TFX$ *be a* TF*-coalgebra and* $h : TFTX \to FTX$ *be an Eilenberg-Moore* T*-algebra. Then there exists a unique* T*-algebra morphism* $f^{\#} : (TX, \mu_X) \to (FTX, h)$ *such that* $f = f^{\#} \circ \eta_X$.

Proof. Uniqueness. Let $f^{\#}$ be such a morphism, then, using the first diagram of monads (i) and that $f^{\#}$ is a T-algebra morphism (ii), we get

$$f^{\#} \underset{(i)}{=} f^{\#} \circ \mu_X \circ T\eta_X \underset{(ii)}{=} h \circ Tf^{\#} \circ T\eta_X = h \circ T(f^{\#} \circ \eta_X) = h \circ Tf$$

Existence. Take $f^{\#} = h \circ Tf$, then, using the naturality of $\mu : TT \Rightarrow T$ (i) and that h is an Eilenberg-Moore T-algebra (ii), we get that $f^{\#}$ is a T-algebra morphism because

$$f^{\#} \circ \mu_X = h \circ Tf \circ \mu_X \underset{(i)}{=} h \circ \mu_{FTX} \circ TTf \underset{(ii)}{=} h \circ Th \circ TTf = h \circ T(h \circ Tf) = h \circ Tf^{\#}$$

Furthermore, using the naturality of $\eta : Id_{\mathbf{C}} \Rightarrow T$ and that h is an Eilenberg-Moore T-algebra, $f^{\#} \circ \eta_X = h \circ Tf \circ \eta_X = h \circ \eta_{FTX} \circ f = f$. $\qquad\square$

Lemma 28. *With the same notations as for Lemma 27, and given a distributive law $\lambda : TF \Rightarrow FT$, then $h = F\mu_X \circ \lambda_{TX} : TFTX \to FTX$ is an Eilenberg-Moore T-algebra.*

Proof. The first diagram of distributive laws (i) and monads (ii) yields

$$h \circ \eta_{FTX} = F\mu_X \circ \lambda_{TX} \circ \eta_{FTX} \underset{(i)}{=} F\mu_X \circ F\eta_{TX} \circ id_{FX} = F(\mu_X \circ \eta_{TX}) \underset{(ii)}{=} Fid_{TX} = id_{TFX}$$

Furthermore

$$
\begin{aligned}
h \circ Th &= F\mu_X \circ \lambda_{TX} \circ TF\mu_X \circ T\lambda_{TX} \\
&= F\mu_X \circ FT\mu_X \circ \lambda_{TTX} \circ T\lambda_{TX} && \text{(naturality of } \lambda\text{)} \\
&= F(\mu_X \circ T\mu_X) \circ \lambda_{TTX} \circ T\lambda TX \\
&= F(\mu_X \circ \mu_{TX}) \circ \lambda_{TTX} \circ T\lambda_{TX} && \text{(second diagram of the monad)} \\
&= F\mu_X \circ F\mu_{TX} \circ \lambda_{TTX} \circ T\lambda_{TX} \\
&= F\mu_X \circ \lambda_{TX} \circ \mu_{FTX} && \text{(second diagram of distributive laws)} \\
&= h \circ \mu_{FTX}
\end{aligned}
$$

$\qquad\square$

The next step is to define a distributive law $\lambda : \mathbb{D}F \Rightarrow F\mathbb{D}$ in order to apply Lemmas 27 and 28. In the following we write $id_{FX} = \langle \pi_X^{\oplus}, \pi_X^*, a \mapsto \pi_X^a \rangle$. Note that $\pi^{\epsilon} : F \Rightarrow \mathbb{I}$ (for $\epsilon \in \{*, \oplus\}$) and $\pi^a : F \Rightarrow Id_{\mathbf{C}}$ (for $a \in A$) are natural transformations.

Lemma 29. *Let $f : X \to \mathbb{I}$ be a measurable function. Then $g_f : \mathbb{D}X \to \mathbb{I}$ defined by $g_f(m) = \int_X f\,dm$ is measurable. Furthermore, if $g : \mathbb{D}\mathbb{I} \to \mathbb{I}$ is the measurable function defined by $g(m) = \int_{\mathbb{I}} id_{\mathbb{I}} dm$, we have $g_f \circ \eta_X = f$ and $g_f \circ \mu_X = g \circ \mathbb{D}g_f$.*

Proof. Measurability. If $f = \mathbf{1}_B$ for some $B \in \Sigma_X$, remark that $g_f(m) = \int_X \mathbf{1}_B dm = m(B) = e_B^X(m)$ so g_f is measurable. By linearity of the integral, g_f is measurable for any simple function. Finally if f is the pointwise limit of an increasing sequence of simple functions $(f_n)_{n \in \mathbb{N}}$, we have that $g_f(m) = \int_X f\,dm = \int_X \lim f_n dm = \lim \int_X f_n dm = \lim g_{f_n}(m)$ by the monotone convergence theorem.

Equations. If $f = \mathbf{1}_B$ for some $B \in \Sigma_X$, then $g_f \circ \eta_X = e_B^X \circ \eta_X = \mathbf{1}_B = f$ and for any $\Phi \in \mathbb{D}\mathbb{D}X$, $(g_f \circ \mu_X)(\Phi) = \mu_X(\Phi)(B) = \int_{\mathbb{D}X} e_B^X d\Phi = \int_{\mathbb{I}} id_{\mathbb{I}} d(\mathbb{D}e_B^X(\Phi)) = (g \circ \mathbb{D}g_f)(\Phi)$. Note that both equations are preserved by linear combinations, hence the result is true for any simple function. Finally, if f is the pointwise limit of an increasing sequence

27

of simple functions $(f_n)_{n \in \mathbb{N}}$ for which the result is true, we already know that g_f is the pointwise limit of the increasing sequence g_{f_n}. In particular $g_{f_n} \circ \eta_X \to g_f \circ \eta_X$ and $g_{f_n} \circ \mu_X \to g_f \circ \mu_X$. For any $\Phi \in \mathbb{DDI}$, $(g \circ \mathbb{D}g_{f_n})(\Phi) = \int_{\mathbb{I}} id_{\mathbb{I}} d(\mathbb{D}g_{f_n}(\Phi)) = \int_{\mathbb{DI}} g_{f_n} d\Phi \to \int_{\mathbb{DI}} g_f d\Phi = (g \circ \mathbb{D}g_f)(\Phi)$. Hence both equations are true for all measurable $f : X \to \mathbb{I}$. $\square$

Taking $f = id_{\mathbb{I}}$ and using this lemma we get that $g : \mathbb{DI} \to \mathbb{I}$ satisfies $g \circ \eta_{\mathbb{I}} = id_{\mathbb{I}}$ and $g \circ \mu_{\mathbb{I}} = g \circ \mathbb{D}g$, hence g is an Eilenberg-Moore $\mathbb{D}$-algebra. For any object X of **Meas**, define $\lambda_X : \mathbb{D}FX \to F\mathbb{D}X$ by

$$\lambda_X = \langle g \circ \mathbb{D}\pi_X^{\oplus}, g \circ \mathbb{D}\pi_X^*, a \mapsto \mathbb{D}\pi_X^a \rangle$$

This is a measurable function because each component is measurable.

Proposition 30. *Let X, Y be objects of **Meas** and $f : X \to Y$ be a measurable function. The following diagrams commute. Consequently, $\lambda : \mathbb{D}F \Rightarrow F\mathbb{D}$ is a distributive law.*

$$
\begin{array}{ccc}
\mathbb{D}FX \xrightarrow{\mathbb{D}Ff} \mathbb{D}FY & \quad FX \xrightarrow{id_{FX}} FX & \quad \mathbb{DD}FX \xrightarrow{\mathbb{D}\lambda_X} \mathbb{D}F\mathbb{D}X \xrightarrow{\lambda_{\mathbb{D}X}} F\mathbb{DD}X \\
\lambda_X \downarrow \quad \downarrow \lambda_Y & \eta_{FX} \downarrow \quad \downarrow F\eta_X \quad \mu_{FX} \downarrow & \quad \downarrow F\mu_X \\
F\mathbb{D}X \xrightarrow{F\mathbb{D}f} F\mathbb{D}Y & \mathbb{D}FX \xrightarrow{\lambda_X} F\mathbb{D}X & \mathbb{D}FX \xrightarrow{\quad\quad\lambda_X\quad\quad} F\mathbb{D}X
\end{array}
$$

Proof.

$$
\begin{aligned}
\lambda_Y \circ \mathbb{D}Ff &= \langle g \circ \mathbb{D}\pi_Y^{\oplus} \circ \mathbb{D}Ff, g \circ \mathbb{D}\pi_Y^* \circ \mathbb{D}Ff, a \mapsto \mathbb{D}\pi_Y^a \circ \mathbb{D}Ff \rangle \\
&= \langle g \circ \mathbb{D}(\pi_Y^{\oplus} \circ Ff), g \circ \mathbb{D}(\pi_Y^* \circ Ff), a \mapsto \mathbb{D}(\pi_Y^a \circ Ff) \rangle \\
&= \langle g \circ \mathbb{D}(id_{\mathbb{I}} \circ \pi_X^{\oplus}), g \circ \mathbb{D}(id_{\mathbb{I}} \circ \pi_X^*), a \mapsto \mathbb{D}(f \circ \pi_X^a) \rangle \\
&= \langle g \circ \mathbb{D}\pi_X^{\oplus}, g \circ \mathbb{D}\pi_X^*, a \mapsto \mathbb{D}f \circ \mathbb{D}\pi_X^a \rangle && (\pi^{\oplus}, \pi^* \text{ natural}) \\
&= F\mathbb{D}f \circ \lambda_X
\end{aligned}
$$

$$
\begin{aligned}
\lambda_X \circ \eta_{FX} &= \langle g \circ \mathbb{D}\pi_X^{\oplus} \circ \eta_{FX}, g \circ \mathbb{D}\pi_X^* \circ \eta_{FX}, a \mapsto \mathbb{D}\pi_X^a \circ \eta_{FX} \rangle \\
&= \langle g \circ \eta_{\mathbb{I}} \circ \pi_X^{\oplus}, g \circ \eta_I \circ \pi_X^*, a \mapsto \eta_X \circ \pi_X^a \rangle && (\text{naturality of } \eta) \\
&= \langle \pi_X^{\oplus}, \pi_X^*, a \mapsto \eta_X \circ \pi_X^a \rangle && (g \text{ is an EM algebra}) \\
&= F\eta_X
\end{aligned}
$$

$$
\begin{aligned}
&F\mu_X \circ \lambda_{\mathbb{D}X} \circ \mathbb{D}\lambda_X \\
&= F\mu_X \circ \langle g \circ \mathbb{D}\pi_{\mathbb{D}X}^{\oplus} \circ \mathbb{D}\lambda_X, g \circ \mathbb{D}\pi_{\mathbb{D}X}^* \circ \mathbb{D}\lambda_X, a \mapsto \mathbb{D}\pi_{\mathbb{D}X}^a \circ \mathbb{D}\lambda_X \rangle && (\text{definition of } \lambda) \\
&= F\mu_X \circ \langle g \circ \mathbb{D}(\pi_{\mathbb{D}X}^{\oplus} \circ \lambda_X), g \circ \mathbb{D}(\pi_{\mathbb{D}X}^* \circ \lambda_X), a \mapsto \mathbb{D}(\pi_{\mathbb{D}X}^a \circ \lambda_X) \rangle \\
&= F\mu_X \circ \langle g \circ \mathbb{D}(g \circ \mathbb{D}\pi_X^{\oplus}), g \circ \mathbb{D}(g \circ \mathbb{D}\pi_X^*), a \mapsto \mathbb{DD}\pi_X^a \rangle && (\text{definition of } \lambda) \\
&= \langle g \circ \mathbb{D}g \circ \mathbb{DD}\pi_X^{\oplus}, g \circ \mathbb{D}g \circ \mathbb{DD}\pi_X^*, a \mapsto \mu_X \circ \mathbb{DD}\pi_X^a \rangle \\
&= \langle g \circ \mu_{\mathbb{I}} \circ \mathbb{DD}\pi_X^{\oplus}, g \circ \mu_{\mathbb{I}} \circ \mathbb{DD}\pi_X^*, a \mapsto \mu_X \circ \mathbb{DD}\pi_X^a \rangle && (g \text{ EM algebra}) \\
&= \langle g \circ \mathbb{D}\pi_X^{\oplus} \circ \mu_{FX}, g \circ \mathbb{D}\pi_X^* \circ \mu_{FX}, a \mapsto \mathbb{D}\pi_X^a \circ \mu_{FX} \rangle && (\text{naturality of } \mu) \\
&= \lambda_X \circ \mu_{FX}
\end{aligned}
$$

$\square$

Let us compute the value of our resulting determinization. Given $\tilde{\alpha} : X \to F\mathbb{D}X$, we recall the notation $\tilde{\alpha} = \langle \tilde{\alpha}_\oplus, \tilde{\alpha}_*, a \mapsto t_a \rangle$, then take $h = F\mu_X \circ \lambda_{\mathbb{D}X}$ (Lemma 28) and $\tilde{\alpha}^\# = \tilde{\alpha} \circ \mathbb{D}h$ (Lemma 27). We get

$$
\begin{aligned}
\tilde{\alpha}^\# &= h \circ \mathbb{D}\tilde{\alpha} \\
&= F\mu_X \circ \lambda_{\mathbb{D}X} \circ \mathbb{D}\tilde{\alpha} \\
&= F\mu_X \circ \langle g \circ \mathbb{D}(\pi^\oplus_{\mathbb{D}X} \circ \tilde{\alpha}), g \circ \mathbb{D}(\pi^*_{\mathbb{D}X} \circ \tilde{\alpha}), a \mapsto \mathbb{D}(\pi^a_{\mathbb{D}X} \circ \tilde{\alpha}) \rangle \\
&= \langle g \circ \mathbb{D}\tilde{\alpha}_\oplus, g \circ \mathbb{D}\tilde{\alpha}_*, a \mapsto \mu_X \circ \mathbb{D}t_a \rangle
\end{aligned}
$$

Let $m \in \mathbb{D}X$. This more explicit expression shows that the coalgebra that arises from the determinization is natural in the sense that the components of $\tilde{\alpha}^\#$ are basically obtained by integrating the information provided by α.

$$
\begin{aligned}
\tilde{\alpha}^\#(m) &= \left\langle \int_X \tilde{\alpha}_\oplus \, dm, \int_X \tilde{\alpha}_* \, dm, a \mapsto \left[S \mapsto \int_X t_a(-)(S) \, dm \right] \right\rangle \\
&= \left\langle \int_X \alpha(-)(LX) \, dm, \int_X \alpha(-)(1), a \mapsto \left[S \mapsto \int_X \alpha(-)(\{a\} \times S) \, dm \right] \right\rangle
\end{aligned}
$$

(iii) Final coalgebra

This heavy determinization part allows us to work on F-coalgebras, which are nice ones because there exists a final object in **Coalg**(F).

Proposition 31. *Let $\Omega = (\mathbb{I} \times \mathbb{I})^{A^*}$ and Σ_Ω be the smallest σ-algebra that makes the functions $e_w : \Omega \to \mathbb{I} \times \mathbb{I}$ defined by $e_w(L) = L(w)$ measurable for every $w \in A^*$. Let $\omega : \Omega \to F\Omega$ be defined by $\omega(L) = \langle L(\varepsilon), a \mapsto L_a \rangle$. Then (Ω, ω) is the final F-coalgebra.*

Proof. First, ω is measurable. Indeed $\pi_1 \circ e_\varepsilon$ and $\pi_2 \circ e_\varepsilon$ are (where $\pi_i : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$) and for $a \in A$, the function $\phi_a : L \mapsto L_a$ is measurable because $e_w \circ \phi_a = e_{aw}$ is measurable for every $w \in A^*$. Let $\beta = \langle \beta_\oplus, \beta_*, a \mapsto \tau_a \rangle : X \to FX$ be an F-coalgebra. It is easy to see that there is at most one coalgebra morphism φ from β to ω, because the commutation of the following diagram yields $\varphi(x)(\varepsilon) = \langle \beta_\oplus(x), \beta_*(x) \rangle$ and $\varphi(x)(aw) = \varphi(\tau_a(x))(w)$.

$$
\begin{array}{ccc}
X & \xrightarrow{\varphi} & \Omega \\
\beta \downarrow & & \downarrow \omega \\
FX & \xrightarrow{F\varphi} & F\Omega
\end{array}
$$

The only thing to check is that this φ is measurable. We prove it by induction on words. The function $e_\varepsilon \circ \varphi = \langle \beta_\oplus, \beta_* \rangle$ is measurable, and if $e_w \circ \varphi$ is measurable then $e_{aw} \circ \varphi = (e_w \circ \varphi) \circ \tau_a$ is measurable too by induction hypothesis. $\square$

Thus for any F-coalgebra β the final morphism towards ω, denoted φ_β, gives a canonical notion of semantics. What we want it something slightly more specific that takes into account the way $\tilde{\alpha}^\#$ was built to produce a probability measure in $\mathbb{D}A^\infty$. This is obtained via a pseudo-final coalgebra $\Pi : \mathbb{D}A^\infty \to F\mathbb{D}A^\infty$ as follows.

Proposition 32. *Let $\pi : A^\infty \to LA^\infty$ be defined by $\pi(\varepsilon) = *$ and $\pi(aw) = (a, w)$. This is the final L-coalgebra.*

Proof. The σ-algebra on LA^∞ is generated by the sets 1 and $\{a\} \times S$ for $S \in \Sigma_{A^\infty}$. Applying Lemma 4 (v), see that $\pi^{-1}(1) = \{\varepsilon\} \in \Sigma_{A^\infty}$ and $\pi^{-1}(\{a\} \times S) = aS \in \Sigma_{A^\infty}$. Because of Lemma 1 this shows that π is measurable. Let $\gamma : X \to LX$ be an L-coalgebra. There is at most one coalgebra morphism from γ to π. Indeed, the commutation of the following diagram yields that $\pi(\varphi(x)) = (id_A \times \varphi + id_1)(\gamma(x))$ so if $\gamma(x) = *$ then $\varphi(x) = \varepsilon$, and if $\gamma(x) = (a, y)$ then $\varphi(x) = a \cdot \varphi(y)$.

$$\begin{array}{ccc} X & \xrightarrow{\varphi} & A^\infty \\ \gamma \downarrow & & \downarrow \pi \\ LX & \xrightarrow{F\varphi} & LA^\infty \end{array}$$

Using Lemma 1, we check that φ is measurable by focusing on sets in S_∞. First see that $\varphi^{-1}(\{\varepsilon\}) = \gamma^{-1}(1) \in \Sigma_X$ because γ is measurable and that $\varphi^{-1}(\varepsilon A^\infty) = X \in \Sigma_X$. Assume that $\varphi^{-1}(\{w\})$ and $\varphi^{-1}(wA^\infty) \in \Sigma_X$, then $\varphi^{-1}(\{aw\}) = \gamma^{-1}(\{a\} \times \varphi^{-1}(\{w\}))$ and $\varphi^{-1}(awA^\infty) = \gamma^{-1}(\{a\} \times \varphi^{-1}(wA^\infty))$ are in Σ_X because γ is measurable. $\square$

Let $\Pi : A^\infty \to FA^\infty$ be the F-coalgebra $\Pi = \mathfrak{e}_{A^\infty} \circ \mathbb{D}\pi$. It has a direct expression involving the measure derivative; it is exactly the same as the Π of section 2.

$$\Pi(m) = \langle m(\pi^{-1}(LA^\infty)), m(\pi^{-1}(1)), a \mapsto [S \mapsto m(\pi^{-1}(\{a\} \times S))] \rangle$$
$$= \langle m(A^\infty), m(\varepsilon), a \mapsto m_a \rangle$$

The aim is now to factorize the semantics obtained via ω into semantics obtained via Π. The following result is kind of a completeness property for this operation.

Lemma 33. *The final morphism φ_Π from Π to ω is injective.*

Proof. For any $m, m' \in \mathbb{D}A^\infty$, in order to have $m = m'$, it is sufficient to prove that $m_{|S_\infty} = m'_{|S_\infty}$ according to Theorem 5. By induction on w, we prove that for all $m, m' \in \mathbb{D}A^\infty$ such that $\varphi_\Pi(m) = \varphi_\Pi(m')$, then $\langle m(wA^\infty), m(w) \rangle = \langle m'(wA^\infty), m'(w) \rangle$. First, $\langle m(\varepsilon A^\infty), m(\varepsilon) \rangle = \varphi_\Pi(m)(\varepsilon) = \varphi_\Pi(m')(\varepsilon) = \langle m'(\varepsilon A^\infty), m'(\varepsilon) \rangle$. Note that $\varphi_\Pi(m) = \varphi_\Pi(m')$ implies $\varphi_\Pi(m_a)(w) = \varphi_\Pi(m)(aw) = \varphi_\Pi(m')(aw) = \varphi_\Pi(m'_a)(w)$ so that $\varphi_\Pi(m_a) = \varphi_\Pi(m'_a)$. Use the induction hypothesis to see that $\langle m(awA^\infty), m(aw) \rangle = \langle m_a(wA^\infty), m_a(w) \rangle = \langle m'_a(wA^\infty), m'_a(w) \rangle = \langle m'(awA^\infty), m'(aw) \rangle$. This achieves the induction, so m and m' coincide on S_∞, hence $m = m'$. $\square$

The following proposition states precisely in which cases the factorization can be done. This is a variant of Theorem 5 in which we really see that our system is making one step. This version is a bit higher-end than Theorem 14 because it also proves that the involved functions are measurable.

Theorem 34. *Let $\beta = \langle \beta_\oplus, \beta_*, a \mapsto \tau_a \rangle : Y \to FY$ be an F-coalgebra. The two following conditions are equivalent:*

(i) *There exists an F-coalgebra morphism $[\![-]\!]$ from β to Π.*

(ii) *The equation $\beta_\oplus = \beta_* + \sum_{a \in A} \beta_\oplus \circ \tau_a$ holds.*

In this case, this morphism is unique.

For convenience we will now denote $e_S^{A^\infty} \circ [\![-]\!]$ by $[\![-]\!](S)$, and $\phi_a \circ [\![-]\!]$ by $[\![-]\!]_a$, where the measure derivative function $\phi_a : m \mapsto m_a$ is measurable as a component of Π.

30

Proof. $(i) \Rightarrow (ii)$ Assume that $[\![-]\!]$ is a coalgebra morphism from β to Π. Commutation of the diagram yields $\langle \beta_\oplus, \beta_*, a \mapsto [\![-]\!] \circ \tau_a \rangle = \langle [\![-]\!](A^\infty), [\![-]\!](\varepsilon), a \mapsto [\![-]\!]_a \rangle$. Let $y \in Y$. Because $[\![y]\!]$ is a measure, $\beta_\oplus(y) = [\![y]\!](\varepsilon A^\infty) = [\![y]\!](\varepsilon) + \sum_{a \in A} [\![y]\!](bA^\infty)$. Thus $\beta_\oplus(y) = \beta_*(y) + \sum_{a \in A} [\![\tau_a(y)]\!](A^\infty) = \beta_*(y) + \sum_{a \in A} (\beta_\oplus \circ \tau_a)(y)$.

Uniqueness. If $[\![-]\!]'$ is another such morphism, we have $[\![-]\!](A^\infty) = [\![-]\!]'(A^\infty)$, $[\![-]\!](\varepsilon) = [\![-]\!]'(\varepsilon)$ and for any $a \in A$, $[\![-]\!] \circ \tau_a = [\![-]\!]_a$ and $[\![-]\!]' \circ \tau_a = [\![-]\!]'_a$. An immediate induction yields $[\![-]\!]_{|S_\infty} = [\![-]\!]'_{|S_\infty}$, thus $[\![-]\!] = [\![-]\!]'$ by Theorem 5.

$(ii) \Rightarrow (i)$ Assume that (ii) holds. Let us define $[\![-]\!]_{|S_\infty}$ by induction:

$$[\![y]\!]_{|S_\infty}(\varepsilon A^\infty) = \beta_\oplus(y) \qquad\qquad [\![y]\!]_{|S_\infty}(\varepsilon) = \beta_*(y)$$
$$[\![y]\!]_{|S_\infty}(awA^\infty) = [\![\tau_a(y)]\!]_{|S_\infty}(wA^\infty) \qquad\qquad [\![y]\!]_{|S_\infty}(aw) = [\![\tau_a(y)]\!]_{|S_\infty}(w)$$

We must prove that it can be extended to a measure, using Theorem 5. First, note that $[\![y]\!]_{|S_\infty}(\varepsilon A^\infty) = \beta_\oplus(y) = \beta_*(y) + \sum_{a \in A}(\beta_\oplus \circ \tau_a)(y) = [\![y]\!]_{|S_\infty}(\varepsilon) + \sum_{a \in A} [\![y]\!]_{|S_\infty}(aA^\infty)$. If it is known that for all $y \in Y$, $[\![y]\!]_{|S_\infty}(wA^\infty) = [\![y]\!]_{|S_\infty}(w) + \sum_{a \in A} [\![y]\!]_{|S_\infty}(waA^\infty)$ then for any $b \in A$ we have $[\![y]\!]_{|S_\infty}(bwA^\infty) = [\![\tau_b(y)]\!]_{|S_\infty}(wA^\infty) = [\![\tau_b(y)]\!]_{|S_\infty}(w) + \sum_{a \in A} [\![\tau_b(y)]\!]_{|S_\infty}(waA^\infty) = [\![y]\!]_{|S_\infty}(bw) + \sum_{a \in A} [\![y]\!]_{|S_\infty}(bwaA^\infty)$. This proves the (ii) of Theorem 5. We denote by $[\![-]\!]$ the extension of $[\![-]\!]_{|S_\infty}$. We postpone the proof of the measurability of $[\![-]\!]$; what is left is the commutation of the coalgebra diagram. The first line of the definition of $[\![-]\!]_{|S_\infty}$ gives directly that $\beta_\oplus = [\![-]\!](A^\infty)$ and $\beta_* = [\![-]\!](\varepsilon)$. Let $a \in A$. For any $y \in Y$, according to the second line of the definition of $[\![-]\!]_{|S_\infty}$, the measures $[\![\tau_a(y)]\!]$ and $[\![y]\!]_a$ coincide on S_∞, hence are equal according to Theorem 5, so $[\![-]\!] \circ \tau_a = [\![-]\!]_a$. This achieves the proof that the diagram commutes.

Measurability. It is not immediate to notice why $[\![-]\!] : Y \to \mathbb{D}A^\infty$ is a measurable function. What has to be shown according to Lemma 2 is that for any $S \in \Sigma_{A^\infty}$, $[\![-]\!](S)$ is measurable. This is true when $S \in S_\infty$. Indeed, $[\![-]\!](\emptyset)$ is the zero function, which is measurable. For the rest we proceed by induction. Obviously $[\![-]\!](\varepsilon A^\infty) = \beta_\oplus$ and $[\![-]\!](\varepsilon) = \beta_*$ are measurable because β is. Furthermore, $[\![-]\!](awA^\infty) = [\![-]\!]_a(wA^\infty) = [\![-]\!](wA^\infty) \circ \tau_a$ and $[\![-]\!](aw) = [\![-]\!]_a(w) = [\![-]\!](w) \circ \tau_a$ are measurable by induction hypothesis and composition.

Let Z be a set. A set $P \subseteq \mathscr{P}(Z)$ is a *π-system* if it is non-empty and closed under finite intersections. A set $D \subseteq \mathscr{P}(Z)$ is a *λ-system* if it contains Z and is closed under difference (if $A, B \in D$ and $A \subseteq B$ then $B \setminus A \in D$) and countable *increasing* union. A widely known theorem of measure theory, namely the $\pi - \lambda$ theorem (see [1], lemma 4.11) is that given P a π-system, D a λ-system such that $P \subseteq D$, then $\sigma_Z(P) \subseteq D$.

Take $Z = A^\infty$, $P = S_\infty$ and $D = \{S \in \Sigma_{A^\infty} \mid [\![-]\!](S) \text{ is measurable}\}$. It is easy to see that S_∞ is a π-system. Moreover, D is a λ-system. Indeed, $A^\infty \in D$ (see above), if $(S_n)_{n \in \mathbb{N}}$ is an increasing sequence of sets in D, then $[\![-]\!](S_1 \setminus S_0) = [\![-]\!](S_1) - [\![-]\!](S_0)$ is measurable as a difference of measurable functions and $[\![-]\!]\left(\bigcup_{n \in \mathbb{N}} S_n\right) = \lim_{n \to \infty} [\![-]\!](S_n)$ is measurable as a pointwise limit of measurable functions. Finally, given the preceding paragraph, we have $S_\infty \subseteq D$. The $\pi - \lambda$ theorem therefore yields $\Sigma_{A^\infty} \subseteq D$. $\qquad \square$

An interpretation of the last proposition is that, in the subcategory of F-coalgebras that satisfy the equation (ii), the final object is Π. If Proposition 34 holds, then note that $\varphi_\Pi \circ [\![-]\!]$ is a coalgebra morphism from β into the final coalgebra ω. Hence by finality

$\varphi_\Pi \circ [\![-]\!] = \varphi_\beta$. This is kind of a soundness property for our factorization. Soundness and completeness together yield the following proposition, which is exactly the same as in section 2.

Proposition 35. *Let $\beta : Y \to FY$ be an F-coalgebra for which Proposition 34 holds. Then for any $y, z \in Y$, $[\![y]\!] = [\![z]\!]$ iff $\varphi_\beta(y) = \varphi_\beta(z)$.*

Proof. By Lemma 33, $[\![y]\!] = [\![z]\!]$ iff $(\varphi_\Pi \circ [\![-]\!])(y) = (\varphi_\Pi \circ [\![-]\!])(z)$ iff $\varphi_\beta(y) = \varphi_\beta(z)$. $\square$

Back to $\alpha : X \to \mathbb{P}LX$ we check that Proposition 34 holds for $\tilde{\alpha}^\# = \langle \tilde{\alpha}_\oplus^\#, \tilde{\alpha}_*^\#, a \mapsto \tau_a \rangle$. Note that because $\alpha(x)(LX) = 1$, we have for all $m \in \mathbb{D}X$ that $m(X) = \int_X 1 dm = \int_X \alpha(x)(LX)dm = \tilde{\alpha}_\oplus^\#(m)$. This justifies the last equality:

$$\tilde{\alpha}_\oplus^\#(m) = \int_X \alpha(-)(LX)dm = \int_X \left(\alpha(-)(1) + \sum_{a \in A} \alpha(-)(\{a\} \times X) \right) dm$$

$$= \int_X \alpha(-)(1)dm + \sum_{a \in A} \int_X \alpha(-)(\{a\} \times X)dm$$

$$= \tilde{\alpha}_*^\#(m) + \sum_{a \in A} \tau_a(m)(X) = \tilde{\alpha}_*^\#(m) + \sum_{a \in A} (\tilde{\alpha}_\oplus^\# \circ \tau_a)(m)$$

Conclusion. To any $\alpha : X \to \mathbb{P}LX$ can be given a canonical trace semantics via a determinization process. This is a function $[\![-]\!] : \mathbb{D}X \to \mathbb{D}A^\infty$.

3.2 Related results

Link with Kerstan's trace semantics

In [11], given an $\alpha : X \to \mathbb{P}LX$, the trace semantics $\mathbf{tr} : X \to \mathbb{P}A^\infty$ is defined by

$$\mathbf{tr}(x)(\varepsilon A^\infty) = \alpha(x)(LX) \quad (= 1) \qquad \mathbf{tr}(\varepsilon) = \alpha(x)(1)$$

$$\mathbf{tr}(x)(awA^\infty) = \int_X \mathbf{tr}(-)(wA^\infty)dt_a(x) \qquad \mathbf{tr}(x)(aw) = \int_X \mathbf{tr}(-)(w)dt_a(x)$$

We will hereby prove that this sematics fits with ours, in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
X & \xrightarrow{\ \mathbf{tr}\ } & \mathbb{P}A^\infty \\
{\scriptstyle \eta_X} \downarrow & & \downarrow {\scriptstyle \iota_{A^\infty}} \\
\mathbb{D}X & \xrightarrow[{[\![-]\!]}]{} & \mathbb{D}A^\infty
\end{array}
$$

Define $[\![-]\!] = [\![-]\!] \circ \eta_X$ as in the discrete case.

Lemma 36. *For any $m \in \mathbb{D}X$ and any $S \in S_\infty$, $[\![m]\!](S) = \int_X [\![-]\!](S)dm$.*

Proof. In this proof there may be times when $\int_X f\,dm$ is denoted by $\int_{x\in X} f(x)m(dx)$. First, show that for any measurable function $f: X \to \mathbb{I}$,

$$\int_X f\,d\tau_a(m) = \int_{x\in X} \left(\int_X f\,dt_a(x) \right) m(dx)$$

Using a density argument, first look at the case $f = \mathbf{1}_B$ for some $B \in \Sigma_X$. The equality becomes $\tau_a(m)(B) = \int_X t_a(-)(B)dm$, which is true by definition of τ_a. Furthermore, the property is clearly preserved by linear combination, so it is true for simple functions. Now let $(f_n)_{n\in\mathbb{N}}$ be an increasing sequence of simple functions with pointwise limit f. Then $\int_X f\,dt_a(m) = \lim \int_X f_n dt_a(m) = \lim \int_{x\in X} \left(\int_X f_n dt_a(x) \right) m(dx) = \int_{x\in X} \left(\int_X \lim f_n dt_a(x) \right) m(dx) = \int_{x\in X} \left(\int_X f\,dt_a(x) \right) m(dx)$. The exchanges between limit and integral are justified by the monotone convergence theorem.

Note further that $[\![x]\!](\varepsilon A^\infty) = (\tilde{\alpha}_\oplus^\# \circ \eta_X)(x) = \tilde{\alpha}_\oplus(x) = \alpha(x)(LX)$ and in the same way $[\![x]\!](\varepsilon) = \alpha(x)(1)$. Now let us prove the lemma by induction, for all $m \in \mathbb{D}X$. First

$$[\![m]\!](\varepsilon A^\infty) = \tilde{\alpha}_\oplus^\#(m) = \int_X \alpha(-)(LX)dm = \int_X [\![-]\!](\varepsilon A^\infty)dm$$

$$[\![m]\!](\varepsilon) = \tilde{\alpha}_*^\#(m) = \int_X \alpha(-)(1)dm = \int_X [\![-]\!](\varepsilon)dm$$

Assume the result is true for wA^∞ and w. Take $\diamond \in \{\{\varepsilon\}, A^\infty\}$.

$$[\![m]\!](aw\diamond) = [\![\tau_a(m)]\!](w\diamond) = \int_X [\![-]\!](w\diamond)d\tau_a(m) \qquad \text{(induction hypothesis)}$$

$$= \int_{x\in X} \left(\int_X [\![-]\!](w\diamond)dt_a(x) \right) m(dx) \qquad \text{(preliminary lemma)}$$

$$= \int_{x\in X} [\![\tau_a(\eta_X(x))]\!](w\diamond)m(dx) \qquad \text{(definition of } \tau_a)$$

$$= \int_X [\![-]\!](aw\diamond)dm$$

$\square$

Using this last lemma and that $\tau_a \circ \eta_X = t_a$, we have for any $x \in X$:

$[\![x]\!](\varepsilon A^\infty) = \alpha(x)(LX)$

$[\![x]\!](\varepsilon) = \alpha(x)(1)$

$[\![x]\!](awA^\infty) = [\![(\tau_a \circ \eta_X)(x)]\!](wA^\infty) = [\![t_a(x)]\!](wA^\infty) = \int_X [\![-]\!](wA^\infty)dt_a(x)$

$[\![x]\!](aw) = [\![(\tau_a \circ \eta_X)(x)]\!](w) = [\![t_a(x)]\!](w) = \int_X [\![-]\!](w)dt_a(x)$

Thus, for any $x \in X$, $[\![-]\!](x)$ and $(\iota_{A^\infty} \circ \mathbf{tr})(x)$ are measures in $\mathbb{D}A^\infty$ that coincide on S_∞. Because of Theorem 5, they are equal. Consequently the above diagram commutes, which mean that the trace semantics we get via determinization and Eilenberg-Moore algebras is the same as the Kleisli trace semantics of [11].

Proposition 37. *The two trace semantics denoted by $[\![-]\!]$ and $\mathbf{tr}$ coincide.*

Link with the discrete case

In the event that $\alpha : X \to \mathbb{P}LX$ can be seen as a discrete system, i.e., for all $x \in X$, $\alpha(x)$ is a linear sum of Dirac distributions, then the general semantics coincide with those obtained in section 2.

Link with a more general correspondence

In [10], an abstract link is established between Kleisli trace semantics and determinized trace semantics. It turns out that when both constructions are possible, and under some compatibility conditions, the two trace semantics can be compared. In the setting of our paper, it is proved above that indeed the trace semantics are the same. But we do not know if the general result can be directly applied here for at least one reason: the correspondence stated in [10] uses only one monad for both constructions. This allows to relate both constructions in an easier fashion, via for example the extension natural transformation $\mathfrak{e} : \mathbb{D}L \Rightarrow F\mathbb{D}$. The moment we violate this is when we use the injection natural transformation $\iota : \mathbb{P} \Rightarrow \mathbb{D}$. It seems actually impossible to choose to use only $\mathbb{P}$ or $\mathbb{D}$. The Giry monad is necessary because we do need the sums-to-1 condition to ensure that Theorem 34 is satisfied. The sub-Giry monad is necessary because the components of $\tilde{\alpha}$ do not sum to 1. One may argue that $\mathbb{P}(A \times X + 1) \simeq \mathbb{D}(A \times X)$ (via the function $m \mapsto m_{|\Sigma_A \otimes \Sigma_X}$), but this does not solve this issue. Indeed, Kleisli semantics of PTS of the form $X \to \mathbb{D}(A \times X)$ is trivial (see Theorem 3.33 in [11]).

Our work is largely done by hand because there are no general enough results about systems of the shape $X \to \mathbb{P}(A \times X + 1)$. Either this is because such systems are a really specific case and the existence of measure semantics is a little wonder, or this may be because we were not able to see a more stylish way to proceed.

Conclusion

The recent formalization of automata through coalgebras allows to take a step back and understand why a trace semantics is the good one in a certain sense. In addition, it provides some tools such as bisimulation (up-to). We took as a starting point the trace semantics for continuous PTS given by Kerstan in [11] and redefined it using a determinization process in both discrete and continuous cases. It seems that a Kleisli approach is inadequate for taking into account infinite traces in **Sets**, whereas our determinization approach can do this using a small and very localized amount of measure theory. For discrete PTS, bisimulations up-to turned out to be a fertile ground for finding an algorithm that checks trace equivalence.

In section 3 the algorithmic considerations of section 2 could be adapted, but this would be irrelevant. Indeed, such algorithms for continuous systems will not be computable as they involve infinite sums or integrals, in contrast to HKC^∞ which only needs some matrix multiplications. Moreover, for any interesting general system, i.e., any system that can not be reduced to the discrete case, such algorithms will not terminate. The only case for which they would be useful is to prove that a given general system is equivalent to a given discrete system.

One could have worked with only one output in the machine functor, as this is usually the case. For example, if the total mass output is dropped, its information is not lost forever as we can compute it by summing the mass on every word using the termination output and the transitions. However, our way of doing is shaped to highlight the step-by-step motion of the automaton and the fact that each state carries two equally important pieces of information. This makes the coalgebraic treatment via the machine functor sound in regards to bisimulation. Another reason is that using two outputs lead "easily" to the statement about the existence of a pseudo-final morphism into $\mathbb{D}A^\infty$.

References

[1] Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: a Hitchhiker's Guide*. Springer, Berlin; London, 2006.

[2] Filippo Bonchi, Marcello Bonsangue, Michele Boreale, Jan Rutten, and Alexandra Silva. A Coalgebraic Perspective on Linear Weighted Automata. *Information and Computation*, 211:77–105, 2012.

[3] Filippo Bonchi, Barbara König, and Sebastian Küpper. Up-to techniques for weighted systems (extended version). *CoRR*, abs/1701.05001, 2017.

[4] Filippo Bonchi and Damien Pous. Checking NFA equivalence with bisimulations up to congruence. In *Principle of Programming Languages (POPL)*, pages 457–468, Roma, Italy, January 2013. ACM. 16p.

[5] Michele Boreale. Weighted bisimulation in linear algebraic form. In *CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings*, pages 163–177, 2009.

[6] Michèle Giry. A categorical approach to probability theory. *Lecture Notes in Mathematics*, 915:68–85, 1982.

[7] Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *CoRR*, abs/0710.2505, 2007.

[8] Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.

[9] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:62–222, 1997.

[10] Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. *Journal of Computer and System Sciences*, 81(5):859 – 879, 2015. 11th International Workshop on Coalgebraic Methods in Computer Science, CMCS 2012 (Selected Papers).

[11] Henning Kerstan and Barbara König. Coalgebraic trace semantics for continuous probabilistic transition systems. *Logical Methods in Computer Science*, 9(4), 2013.

[12] Jurriaan Cornelis Rot. *Enhanced coinduction*. Faculty of Science, Leiden University, 2015.

[13] Alexandra Silva, Filippo Bonchi, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Generalizing the powerset construction, coalgebraically. In *FSTTCS*, 2010.