

Sound and Complete Axiomatization of Trace Semantics for Probabilistic Systems

Alexandra Silva^{1,2}

Centrum Wiskunde & Informatica, The Netherlands

Ana Sokolova^{3,4}

Department of Computer Sciences, University of Salzburg, Austria

Abstract

We present a sound and complete axiomatization of finite complete trace semantics for generative probabilistic transition systems. Our approach is coalgebraic, which opens the door to axiomatize other types of systems. In order to prove soundness and completeness, we employ determinization and show that coalgebraic traces can be recovered via determinization, a result interesting in itself. The approach is also applicable to labelled transition systems, for which we can recover the known axiomatization of trace semantics (work of Rabinovich).

Keywords: probabilistic systems, trace semantics, axiomatization, coalgebra

1 Introduction

Quite some amount of work in formal methods, in particular on process algebra and process calculi, concentrates on representing processes by expressions (terms in some process algebraic language) and providing axiomatizations of behavior semantics, in most cases branching-time semantics.

Coalgebras arose as a mathematical model of state-based systems in the last couple of decades. The strength of coalgebraic modeling lies in the fact that many important notions are parametrized by the type of the system, formally given by a functor. On the one hand, the coalgebraic framework is unifying, allowing for a uniform study of different systems and making precise the connection between

¹ Email: ams@cw.i.nl

² Visiting researcher at Cornell University

³ Email: anas@cs.uni-salzburg.at

⁴ Research supported by the Austrian Science Fund (FWF) grant V00125

them. On the other hand, it can serve as a guideline for the development of basic notions for new models of computation.

In [17], Bonchi, Bonsangue, Rutten and the first author made use of the coalgebraic view on systems to devise a framework where languages of specification and axiomatizations can be uniformly derived for a large class of systems, including quantitative systems, such as weighted and probabilistic automata. The axiomatizations considered were proved, in a uniform way, to be sound and complete with respect to bisimilarity.

Bisimilarity may sometimes be considered a too strong equivalence between states of a system [13]. For applications where the branching in the system is irrelevant, linear-time semantics like trace semantics might be more appropriate. Consider for example the following two probabilistic transition systems

$$\bullet \xrightarrow{a, \frac{1}{2}} \bullet \xrightarrow{b, \frac{1}{3}} \bullet \xrightarrow{1} * \qquad \bullet \xrightarrow{a, \frac{1}{3}} \bullet \xrightarrow{b, \frac{1}{2}} \bullet \xrightarrow{1} *$$

where the labels a and b are action labels, and the labels $\frac{1}{2}$, $\frac{1}{3}$ are quantities that represent probabilistic branching (the probability of getting from one state to another with a given label). These two systems are not bisimilar, but they are trace equivalent since for (finite) trace equivalence only the total probability to reach termination with a word of labels matters (for both systems this probability is $\frac{1}{6}$ by the unique possible word ab).

In [7], Hasuo, Jacobs and the second author provided a notion of (finite) trace semantics for a large class of coalgebras and showed that their abstract notion coincides with existing notions in the literature, such as the ones for labeled transition systems or (generative) probabilistic automata. The theory works for TF -coalgebras in **Sets** with T a suitable monad modeling branching and F a suitable functor modeling linear behavior (involving the existence of a distributive law $\lambda: FT \Rightarrow TF$ that distributes branching over linear behavior). Coalgebraic trace semantics shows that linear-time semantics fits into the paradigm of final coalgebra semantics (in the Kleisli category of the monad T), and can thus benefit from the associated machinery, for instance in showing compositionality/congruence of bisimilarity and trace equivalence for various coalgebras [7]. This paper shows another benefit of the generic trace theory, allowing for new sound and complete axiomatizations of trace semantics for probabilistic transition systems in a coalgebraic view.

The paper combines the work on generic axiomatizations [17] bringing process algebra to coalgebra and coalgebraic trace semantics [7] and provides a sound and complete axiomatization of trace semantics for probabilistic transition systems. Probabilistic transition systems, in this paper, are coalgebras of type $\mathcal{D}_\omega(1 + A \times id)$, where \mathcal{D}_ω is the subdistribution monad. The work presented here can be seen as a step towards the goal to derive a framework where axiomatizations for trace semantics can be uniformly derived for a larger class of systems. However, it is difficult to describe a class of monads for which the conditions of the generic trace theory are met. The generic trace theory works for the powerset monad which allows us to use the same approach and provide sound and complete axiomatization of (finite)

trace semantics for labeled transition systems (LTS), in which case we can recover the results of Rabinovich.

We build on the framework of [17] in the sense that we keep the same specification language but add one new axiom. This is natural and also in accordance to the strategy used by Rabinovich, who presented a sound and complete axiomatization of trace semantics for LTS [14] by adding one axiom to the sound and complete axiomatization of bisimilarity proposed by Milner for the same language [11]. In our case, the additional axiom also suffices. It should be noted however that the step from qualitative, that is LTS, to general quantitative systems is not at all trivial. The main difficulty is caused by the following: while every finite LTS can be changed to a finite trace-equivalent LTS that is deterministic (in any state there is at most one a -labelled transition), this is not the case for probabilities/general weights. For a finite system (hence corresponding to an expression), there may be no finite deterministic system that is trace equivalent to it. Hence, the difficulty is in finding a “normal form” expression for all expressions that represent trace equivalent systems, since expressions correspond to finite systems only. Coalgebraic proofs of soundness and completeness [8,4,17] involve a finality argument that avoids reasoning about normal forms. This is our way out as well: We use the (infinite) determinization of a probabilistic transition system but avoid reasoning about normal forms by using a (more involved) finality argument.

Organization of the paper Section 2 and Section 3 are the introductory part of the paper introducing basics of coalgebras and coalgebraic trace semantics, and probabilistic transition systems and their trace semantics in concrete terms, respectively. In Section 4 we present the syntax of expressions for quantitative transition systems, followed by the axiomatization in Section 5 where the main results (soundness and completeness) are presented and proven. We wrap-up with concluding remarks in Section 6. The proofs are available in [18].

2 Preliminaries

In this section, we introduce the basic definitions on coalgebras and (coalgebraic) trace semantics.

Coalgebras and algebras. Let F be an endofunctor on **Sets**, the category of sets and functions. An F -coalgebra is a pair $(X, \alpha: X \rightarrow F(X))$ where X is the carrier set, the set of states, and α is the coalgebra transition map. An F -algebra is a pair $(X, \mathfrak{a}: F(X) \rightarrow X)$. For brevity, we often identify a (co)algebra with its (co)algebra map. Given two F -coalgebras $\alpha: X \rightarrow F(X)$ and $\beta: Y \rightarrow F(Y)$, a coalgebra homomorphism from α to β is a map $h: X \rightarrow Y$ such that $\beta \circ h = F(h) \circ \alpha$. Given two F -algebras $\mathfrak{a}: F(X) \rightarrow X$ and $\mathfrak{b}: F(Y) \rightarrow Y$, an algebra homomorphism from \mathfrak{a} to \mathfrak{b} is a map $h: X \rightarrow Y$ such that $\mathfrak{b} \circ F(h) = h \circ \mathfrak{a}$. F -(co)algebras together with their (co)algebra homomorphisms form a category.

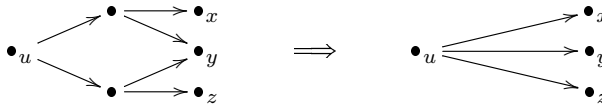
A final F -(co)algebra is a final object in the category of F -(co)algebras: From any F -(co)algebra α there is a unique homomorphism beh_α to the final one. If a final coalgebra exists, it induces a final coalgebra semantics which identifies two states if and only if they are mapped to the same element of the final coalgebra

via the unique homomorphism. In **Sets**, for weak pullback preserving functors, the final coalgebra semantics coincides with bisimilarity, i.e., for states x and y in a coalgebra $\alpha: X \rightarrow F(X)$, $x \sim y \Leftrightarrow \text{beh}_\alpha(x) = \text{beh}_\alpha(y)$.

Trace semantics. In this paper we are further interested in (finite) trace semantics, which also happens to be a final coalgebra semantics, only in a different category. Coalgebraic (finite) trace semantics has been developed for coalgebras of the form $X \rightarrow TF(X)$ where T is a suitable monad and F a suitable functor, see [7]. Essential for coalgebraic trace semantics is the Kleisli category of a monad. A monad (T, η, μ) , which we will frequently denote by T , on **Sets** consists of an endofunctor T on **Sets** and two natural transformations, the unit $\eta: id \Rightarrow T$ and the multiplication $\mu: TT \Rightarrow T$, that is, functions $\eta_X: X \rightarrow T(X)$ and $\mu_X: TT(X) \rightarrow T(X)$ for each set X satisfying a naturality condition. The unit and multiplication satisfy the compatibility conditions $\mu_X \circ \eta_{TX} = \mu_X \circ T\eta_X = id$ and $\mu_X \circ T\mu_X = \mu_X \circ \mu_{TX}$.

The monad structures provide a perfect way of modelling “branching”. Intuitively, the unit η embeds a non-branching behavior as a trivial branching (with a single branch) whereas the multiplication μ “flattens” two successive branchings into one branching, abstracting away internal branchings.

An example of a monad is the powerset monad \mathcal{P} with unit given by singleton, and multiplication given by union. Here, the “flattening”-of-a-“branching” metaphor is obvious, as pictured below.



A monad T on **Sets** allows for a definition of a Kleisli category $\mathcal{Kl}(T)$ whose objects are sets, and a morphism $f: X \multimap Y$ is a function $f: X \rightarrow TY$. The identity morphism on X is η_X , and composition of morphisms is defined as

$$f \bullet g = \mu \circ Tf \circ g = \left(X \xrightarrow{g} TY \xrightarrow{Tf} TTZ \xrightarrow{\mu} TZ \right).$$

There is a canonical lifting functor $J: \mathbf{Sets} \rightarrow \mathcal{Kl}(T)$ which is the identity on objects, and maps a function $f: X \rightarrow Y$ to the function $J(f) = \eta \circ f: X \rightarrow TY$.

The coalgebraic trace result of [7] applies to TF -coalgebras in **Sets** if T and F satisfy a number requirements:

- There exists a distributive law $\lambda: FT \Rightarrow TF$. As a consequence, F lifts to a functor \overline{F} on $\mathcal{Kl}(T)$, with $\overline{F}(X) = F(X)$ and for a Kleisli arrow $f: X \multimap Y$, i.e., a map $f: X \rightarrow TY$, $\overline{F}(f) = \lambda \circ F(f)$. Hence TF -coalgebras in **Sets** are \overline{F} -coalgebras in $\mathcal{Kl}(T)$.
- The Kleisli category $\mathcal{Kl}(T)$ is suitably order-enriched, with order \sqsubseteq on Kleisli homsets, bottom element \perp and suprema of directed subsets.
- The lifting $\overline{F}: \mathcal{Kl}(T) \rightarrow \mathcal{Kl}(T)$ is locally monotone.

The requirements are explained in detail in [7]. The main result of the generic trace theory [7] is:

If T and F satisfy the requirements of the generic trace theory and there exists an initial F -algebra $\iota: F(I) \xrightarrow{\cong} I$ in **Sets**, then the lifted coalgebra $J(\iota^{-1}) = \eta \circ \iota^{-1}: I \rightarrow TF(I)$ is final \overline{F} -coalgebra in $\mathcal{Kl}(T)$.

This enables defining trace semantics for TF -coalgebras in **Sets** as the final coalgebra semantics for \overline{F} -coalgebras in $\mathcal{Kl}(T)$. More precisely, for a coalgebra $\alpha: X \rightarrow TFX$ in **Sets**, i.e., $\alpha: X \dashv\rightarrow \overline{F}Y$ in $\mathcal{Kl}(T)$, we denote by tr_α the final coalgebra map in $\mathcal{Kl}(T)$, called the trace map. The trace of a state $x \in X$ is given by the image $\text{tr}_\alpha(x)$. Trace equivalence is defined by $x \sim_{\text{tr}} y \Leftrightarrow \text{tr}_\alpha(x) = \text{tr}_\alpha(y)$.

The requirements of the generic trace theory hold for the powerset monad \mathcal{P} , the subdistribution monad \mathcal{D} , and the lift monad $1 + id$, together with the inductively defined class of all “shapely functors” [7].

Slightly abusing the notation, whenever there is no risk of confusion, we will denote the lifted functor \overline{F} by F as well.

3 Probabilistic transition systems and their traces

In this paper, we consider finitely branching generative probabilistic transition systems [19] with explicit termination. They are TF coalgebras of the finitely supported subdistribution monad \mathcal{D}_ω and the linear-behavior functor $F = 1 + A \times id$ where A is a set of labels and $1 = \{*\}$ is a singleton set, used to model termination. The monad \mathcal{D}_ω assigns to a set X the set

$$\mathcal{D}_\omega(X) = \{\varphi \in [0, 1]^X \mid \varphi \text{ has finite support and } \sum_{x \in \text{supp}(\varphi)} \varphi(x) \leq 1\}$$

and to a function $f: X \rightarrow Y$, the function $\mathcal{D}_\omega(f): \mathcal{D}_\omega(X) \rightarrow \mathcal{D}_\omega(Y)$:

$$\mathcal{D}_\omega(f)(\varphi) = \lambda y. \sum_{x \in f^{-1}(\{y\})} \varphi(x).$$

The unit of \mathcal{D}_ω is given by $\eta_X(x) = (x \mapsto 1)$ and the multiplication by

$$\mu(\Phi)(x) = \sum_{\varphi \in \mathcal{D}_\omega(X)} \Phi(\varphi) \cdot \varphi(x), \quad \Phi \in \mathcal{D}_\omega \mathcal{D}_\omega(X)$$

Hence, our probabilistic transition systems are $\mathcal{D}_\omega(1 + A \times id)$ -coalgebras on **Sets**. The monad \mathcal{D}_ω provides probabilistic branching. The finite support requirement ensures finite branching and is necessary for representing probabilistic transition systems by finite expressions. The functor $1 + A \times id$ provides linear behavior, in which a state can either successfully terminate or make a labelled transition to another state.

Given a probabilistic transition system $\alpha: X \rightarrow \mathcal{D}_\omega(1 + A \times X)$ we write

$$x \xrightarrow{p} * \quad \text{if} \quad \alpha(x)(*) = p,$$

i.e., x successfully terminates with probability p , and

$$x \xrightarrow{a,p} y \quad \text{if} \quad \alpha(x)(a, y) = p,$$

i.e., if x can make an a -labelled step to y with weight p . Here, and throughout the paper, without any risk of confusion, we are omitting the coproduct injections when representing elements of $1 + A \times X$.

The monad \mathcal{D}_ω is not suitable for describing traces. The reason (intuitively) is that a trace of a state is a distribution over words. Even if the system is defined with finitely-supported distributions only, the trace will in general not have finite support. For example, consider the finite probabilistic transition system

$$\begin{array}{c} a, \frac{1}{2} \\ \curvearrowright \\ x \xrightarrow{\frac{1}{2}} * \end{array}$$

The trace of state x is the distribution that assigns probability $\frac{1}{2^{n+1}}$ to the word a^n for all $n \in \mathbb{N}$ and hence has infinite support. In terms of the generic trace theory requirements, \mathcal{D}_ω fails to satisfy the requirement of existence of suprema of directed subsets.

However, the requirements of the general trace theory do hold for the monad \mathcal{D} which is defined as \mathcal{D}_ω by dropping the finite support condition. We will apply the generic trace results by using the natural injection $i: \mathcal{D}_\omega(X) \rightarrow \mathcal{D}(X)$. The conditions for applicability of the generic trace results hold for the functor $F = 1 + A \times id$.

In particular, we need to include explicit termination since the initial algebra of the functor $A \times id$ is trivial. As a result, we can only deal with (finite) terminating traces. In case of LTS, this is no restriction: one can add the possibility to explicitly terminate to each state of an LTS, and so the finite terminating traces of this transformed LTS are all finite traces of the original one. With probabilities, this is not the case: if in a state the probability to terminate is zero and the sum of the probabilities to make a step is one, then there is no place for adding termination. Nevertheless, (finite) terminating traces are of sufficient interest and have been studied under the name completed-trace semantics in process theory.

For completeness, we mention the distributive law $\lambda: 1 + A \times \mathcal{D} \Rightarrow \mathcal{D}(1 + A \times id)$ that enables the lifting of F to $\mathcal{Kl}(\mathcal{D})$. It is defined by $\lambda_X(*) = \eta(*)$ and $\lambda_X(a, \xi) = \lambda(a, x) \cdot \xi(x)$ for $\xi \in \mathcal{D}(X)$.

It seems possible, but requires significant additional work, to extend the results presented here to an inductively defined class of so-called shapely functors (cf. [7]).

The final $1 + A \times id$ -coalgebra in $\mathcal{Kl}(\mathcal{D})$ is $\eta \circ \iota: A^* \longrightarrow \mathcal{D}(1 + A \times A^*)$ with $\iota: A^* \xrightarrow{\cong} 1 + A \times A^*$ being the (inverse of the) initial algebra isomorphism, given by $\iota(\varepsilon) = *$ and $\iota(aw) = (a, w)$.

The trace map, for a coalgebra $X \xrightarrow{\alpha} \mathcal{D}_\omega(1 + A \times X)$, is defined by applying the generic trace theory to the coalgebra $X \xrightarrow{\alpha} \mathcal{D}_\omega(1 + A \times X) \xrightarrow{i} \mathcal{D}(1 + A \times X)$, as we depict on the right, and can be instantiated to the concrete definition:

$$\begin{aligned} \text{tr}(x)(\varepsilon) &= p, \quad \text{if } x \xrightarrow{p} * \\ \text{tr}(x)(aw) &= \sum_{x \xrightarrow{a,p} y} p \cdot \text{tr}(y)(w). \end{aligned} \quad \begin{array}{ccc} X & \xrightarrow{\text{tr}} & A^* \\ i \circ \alpha \downarrow & & \downarrow \eta \circ \iota \\ 1 + A \times X & \xrightarrow{\bullet} & 1 + A \times A^* \end{array}$$

In the diagram above the black dot on the arrows indicates Kleisli arrows and therefore the composition is Kleisli composition.

The coalgebraic trace definition provides a natural (terminating, finite) trace distribution of a state in a probabilistic transition system. We note that this trace distribution is different than the (possibly infinite) trace distribution (without explicit termination) [15] which is a probability measure over a σ -algebra generated by so-called cones. We are not aware of a possibility to deal with such trace semantics coalgebraically.

We note that, as expected, (coalgebraic) bisimilarity implies (coalgebraic) trace equivalence, i.e., $x \sim y \Rightarrow x \sim_{\text{tr}} y$.

4 Syntax

In this section, we introduce the syntax of the specification language for which we will present a sound and complete axiomatization of trace semantics. The language is an instance of the framework introduced in [17], where uniform sound and complete calculi for bisimilarity were introduced. We illustrate the definitions of this section with examples that we shall use in the subsequent sections and which capture key differences between bisimilarity and trace.

Definition 4.1 [Expressions for probabilistic transition systems] Given a set of input actions A and a set of fixed-point variables X , the set **Exp** of expressions for quantitative transition systems is given by the closed expressions contained in the following BNF, for $a \in A$ and $x \in X$:

$$\begin{aligned} E &::= \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x. E^g \mid x & (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1) \\ E^g &::= \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x. E^g & (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1) \\ F_i &::= * \mid a \cdot E \end{aligned}$$

The operator μ in the expression $\mu x. E^g$ functions as a binder for all the occurrences of the variable x in E^g . Note that the only difference between E^g and E is the occurrence of variables (E^g is an expression where variables occur *guarded*, that is only inside an expression of the shape $p \cdot a \cdot -$). An expression E is *closed* if all variables $x \in X$ occurring in E are bound.

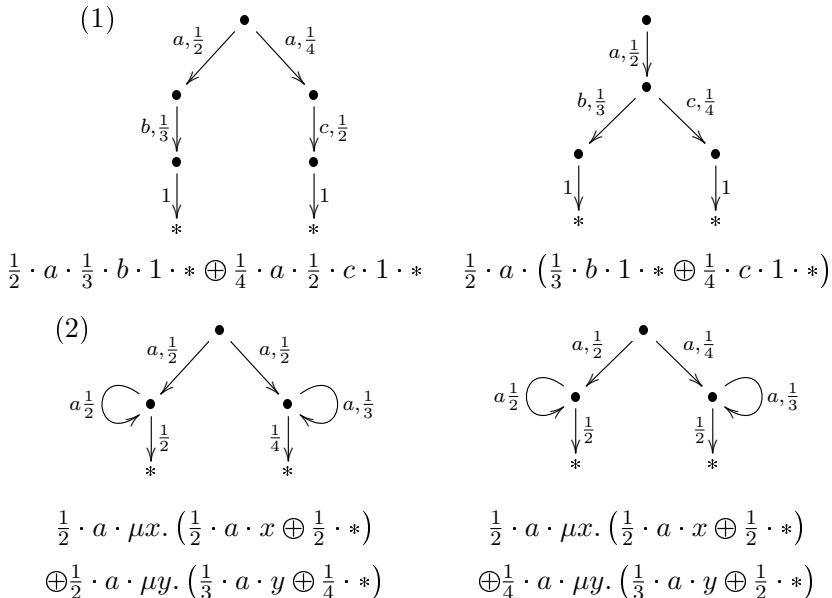
Intuitively, an expression $\bigoplus_{i \in I} p_i \cdot F_i$ behaves as the expression F_i with probability p_i , and μ -expressions are used to represent loops: a μ -expression behaves the same as its unfolding. We make this precise by providing the set of expressions with a coalgebraic structure.

We define $c: \mathbf{Exp} \rightarrow \mathcal{D}_\omega(1 + A \times \mathbf{Exp})$ by induction on the number of nested fixed-points as follows:

$$\begin{aligned} c(\bigoplus_{i \in I} p_i \cdot F_i)(*) &= \sum_{i: F_i = *} p_i \\ c(\bigoplus_{i \in I} p_i \cdot F_i)(a, E) &= \sum_{i: F_i = a \cdot E} p_i \\ c(\mu x. E^g) &= c(E^g[\mu x. E^g/x]) \end{aligned}$$

Having a coalgebra structure on the set of expressions has two advantages: it provides immediately a natural semantics, using the unique homomorphism into the final coalgebra (which can be thought of as the universe of behaviors), and it enables to define when a state s of a probabilistic transition system and an expression E are bisimilar, $s \sim E$, or trace equivalent, $s \sim_{\text{tr}} E$.

Example 4.2 [Some specifications and corresponding systems] To give an intuition for the type of systems each expression specifies, we present below a few examples of expressions and equivalent systems (more precisely, the top state of each system is bisimilar to the expression).



The systems on the right and on the left in each row are trace equivalent. However, they are not bisimilar and, therefore, each pair of expressions in each row would not be provably equivalent using the axiomatization of [17]. We will show

later how to syntactically prove the trace equivalence of the expressions, making use of the axiomatization we will introduce for trace semantics.

Theorem 4.3 (Kleene’s theorem for trace) *For every expression $E \in \text{Exp}$ there exists a (finite) probabilistic transition system (S, α) and $s \in S$ such that $E \sim_{\text{tr}} s$. Conversely, for every locally finite probabilistic transition system (S, α) and $s \in S$ there exists an expression $E \in \text{Exp}$ such that $s \sim_{\text{tr}} E$.*

Proof. Direct consequence from the similar theorem for bisimilarity [17, Theorem 4.9] and the fact that bisimilarity implies trace equivalence. \square

In the formulation of Kleene’s theorem we use *locally finite* probabilistic systems. These are probabilistic systems in which from each state only finitely many states are reachable (coalgebraically, this means that the subcoalgebra generated by each state is finite).

5 Sound and complete axiomatization for trace

In this section, we present an equational system to reason about probabilistic expressions. We will prove it sound and complete with respect to trace semantics.

For sake of simplicity, in what follows we first introduce a nullary operation \emptyset (denoting the empty \oplus -sum) and two partial operations on expressions: a binary sum $E \oplus E'$, and a unary scalar product pE for a non-negative real number p , and write the axioms with help of these auxiliary operations. They are defined as follows:

The binary sum $E \oplus E'$ is defined if and only if $E = \bigoplus_{i \in I} p_i \cdot F_i$, $E' = \bigoplus_{j \in J} q_j \cdot F'_j$, and $\sum_{i \in I} p_i + \sum_{j \in J} q_j \leq 1$, in which case it equals (as expected) the expression $\bigoplus_{k \in I+J} r_k \cdot F''_k$ with $r_k = p_i$, $F''_k = F_i$ for $k = i \in I$ and $r_k = q_j$, $F''_k = F'_j$ for $k = j \in J$. Clearly, we then have $\bigoplus_{i \in I} p_i \cdot F_i = (p_1 \cdot F_1 \oplus (p_2 \cdot F_2 \oplus (\dots)))$

Given a non-negative real number p , the scalar product pE is defined by

$$p \left(\bigoplus_{i \in I} p_i \cdot E_i \right) = \bigoplus_{i \in I} (pp_i) \cdot E_i, \quad p(\mu x.E) = p(E[\mu x.E/x]).$$

Note that $p(\bigoplus_{i \in I} p_i \cdot E_i)$ is defined if and only if $\sum_i pp_i \leq 1$.

In what follows, we present an axiom system for probabilistic expressions using the binary sum, the zero expression, and the scalar product. An axiom $E_1 \equiv E_2$ is to be understood as: if *both* E_1 and E_2 are well-defined expressions, then they are equivalent with respect to \equiv .

Let the relation $\equiv \subseteq \text{Exp} \times \text{Exp}$, written infix-style, be the least equivalence relation satisfying the axioms (and implication rules) from Figure 1. From the axioms, only the last is related to traces. The subset of the axioms in Figure 1 excluding the last one is sound and complete w.r.t. bisimilarity, as it was shown in [17].

$E_1 \oplus (E_2 \oplus E_3)$	$\equiv (E_1 \oplus E_2) \oplus E_3$	(A)
$E_1 \oplus E_2$	$\equiv E_2 \oplus E_1$	(C)
$E \oplus \underline{\emptyset}$	$\equiv E$	(E)
$\mu x.E$	$\equiv E[\mu x.E/x]$	(FP)
$\gamma[E/x] \equiv E$	$\Rightarrow \mu x.\gamma \equiv E$	(UFP)
$\mu x.E$	$\equiv \mu y.E[y/x]$ if y is not free in E	(α – equiv)
$E_1 \equiv E_2$	$\Rightarrow E[E_1/x] \equiv E[E_2/x]$	(Cong)
$0 \cdot E$	$\equiv \underline{\emptyset}$	(Z)
$p \cdot E \oplus p' \cdot E$	$\equiv (p + p') \cdot E$	(S)
$p_1 \cdot a \cdot pE_1 \oplus p_2 \cdot a \cdot pE_2 \equiv p \cdot a \cdot (p_1 E_1 \oplus p_2 E_2)$		(D)

Fig. 1. Axioms for trace semantics

Example 5.1 We now show some examples of the derivation of trace equivalence of two expressions. The expressions we consider in this example already appeared in Example 4.2 (1) and (2), together with equivalent transition systems. We start by showing that the expressions from Example 4.2 (1) are \equiv -equivalent, i.e.,

$$\left(\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot *\right) \oplus \left(\frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot *\right) \equiv \frac{1}{2} \cdot a \cdot \left(\frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot c \cdot 1 \cdot *\right).$$

First, we observe that $\frac{1}{3} \cdot b \cdot 1 \cdot * = \frac{1}{2}(\frac{2}{3} \cdot b \cdot 1 \cdot *)$ and $\frac{1}{2} \cdot c \cdot 1 \cdot * = \frac{1}{2}(1 \cdot c \cdot 1 \cdot *)$. Then, we apply (D) using $p = \frac{1}{2}$, $p_1 = \frac{1}{2}$ and $p_2 = \frac{1}{4}$:

$$\begin{aligned} \left(\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot *\right) \oplus \left(\frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot *\right) &\stackrel{(D)}{=} \frac{1}{2} \cdot a \cdot \left(\frac{1}{2}(\frac{2}{3} \cdot b \cdot 1 \cdot *) \oplus \frac{1}{4}(1 \cdot c \cdot 1 \cdot *)\right) \\ &= \frac{1}{2} \cdot a \cdot \left(\frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot c \cdot 1 \cdot *\right). \end{aligned}$$

A more interesting example is provided by the expressions from Example 4.2 (2). The proof of equivalence of these expressions requires the use of the (UFP) rule. We first start by observing that the left side of the sum in each expression is the same. Thus, using (Cong), it suffices to prove that

$$\frac{1}{2} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *\right) \equiv \frac{1}{4} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot *\right)$$

Soundness	Completeness
$E_1 \equiv E_2$	$E_1 \sim_{\text{tr}} E_2$
$\Leftrightarrow [E_1] = [E_2]$	$\Leftrightarrow \text{tr}(E_1) = \text{tr}(E_2)$
$\stackrel{(*)}{\Rightarrow} \text{out}_{\equiv}([E_1]) = \text{out}_{\equiv}([E_2])$	$\stackrel{(\Delta)}{\Leftrightarrow} \text{out}_{\equiv}([E_1]) = \text{out}_{\equiv}([E_2])$
$\stackrel{(\Delta)}{\Leftrightarrow} \text{tr}(E_1) = \text{tr}(E_2)$	$\stackrel{(\heartsuit)}{\Rightarrow} [E_1] = [E_2]$
$\Leftrightarrow E_1 \sim_{\text{tr}} E_2$	$\Leftrightarrow E_1 \equiv E_2$

Fig. 2. Soundness and completeness. $(*)$: existence of out_{\equiv} , (Δ) : $\text{out}_{\equiv} \circ [-] = \text{tr}$, (\heartsuit) : out_{\equiv} is injective.

In what follows let E stand for the expression $\mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot *)$

$$\begin{aligned}
& \frac{1}{2} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *) \equiv \frac{1}{4} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot *) \\
\Leftrightarrow & \frac{1}{2} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *) \equiv \frac{1}{4} \cdot a \cdot (\frac{1}{3} \cdot a \cdot E \oplus \frac{1}{2} \cdot *) \quad (\text{Cong}) \text{ and } (FP) \\
\Leftrightarrow & \frac{1}{2} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *) \equiv \frac{1}{3} \cdot a \cdot \frac{1}{4} \cdot a \cdot E \oplus \frac{1}{2} \cdot a \cdot \frac{1}{4} \cdot * \quad (D) \text{ with } p = \frac{1}{4}, p_1 = \frac{1}{3} \text{ and } p_2 = \frac{1}{2} \\
\Leftrightarrow & \frac{1}{2} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *) \equiv \frac{1}{2} \cdot a \cdot (\frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot *) \quad (D) \text{ with } p = \frac{1}{2}, p_1 = \frac{1}{3} \text{ and } p_2 = \frac{1}{2} \\
\Leftrightarrow & \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *) \equiv \frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot * \quad (\text{Cong}) \\
\Leftarrow & \frac{1}{3} \cdot a \cdot (\frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot *) \oplus \frac{1}{4} \cdot * \equiv \frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot * \quad (UFP) \\
\Leftrightarrow & \frac{1}{3} \cdot a \cdot (\frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot *) \equiv \frac{1}{6} \cdot a \cdot E \quad (\text{Cong}) \\
\Leftrightarrow & \frac{1}{6} \cdot a \cdot (\frac{1}{3} \cdot a \cdot E \oplus \frac{1}{2} \cdot *) \equiv \frac{1}{6} \cdot a \cdot E \quad (D) \text{ twice} \\
\Leftrightarrow & E \equiv E \quad (\text{Cong}) \text{ and } (FP)
\end{aligned}$$

In the next sections, we will show that the axiomatization, obtained from the sound and complete axiomatization for bisimilarity by adding one new axiom, is sound and complete with respect to trace semantics. This is the main technical result of the paper and, despite the simplicity of the axioms, proving that they are enough to achieve completeness is not a trivial task. Before we provide the technical details of the proof, let us present the intuitive idea behind it.

5.1 Soundness and completeness: An overview

We want to show that the axiomatization above is sound and complete with respect to trace semantics. That is,

$$E_1 \sim_{\text{tr}} E_2 \Leftrightarrow E_1 \equiv E_2$$

Our strategy is to show that the trace map tr is equal to a composition of two maps $\text{out}_{\equiv} \circ [-]$, where out_{\equiv} is an injective map, which we will define below, and $[-]$ is the canonical map mapping each expression to its \equiv -class. Having this, soundness and completeness follow easily, as shown in Figure 2.

We proceed as follows: in Section 5.2 we discuss determinization of probabilistic transition systems, define out_{\equiv} and show that $\text{out}_{\equiv} \circ [-]$ is a Kleisli homomorphism from $(\text{Exp}, i \circ c)$ to the final $(A^*, \eta \circ \iota)$, which by finality yields $\text{out}_{\equiv} \circ [-] = \text{tr}$ and soundness follows; in Section 5.3, we show that out_{\equiv} is an injective map, which will have as consequence completeness.

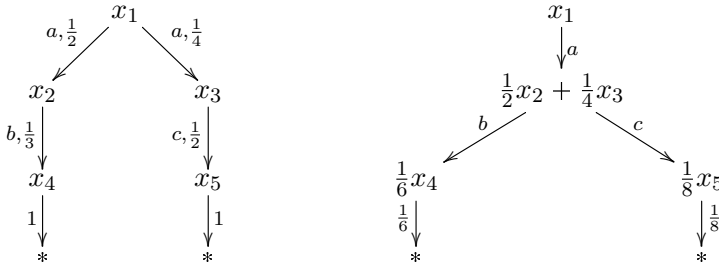
5.2 A way out: Determinization of probabilistic transition systems

The determinization of a probabilistic transition system

$$\alpha: X \rightarrow \mathcal{D}_\omega(1 + A \times X)$$

is a “deterministic” system of type $G = [0, 1] \times id^A$ and state space $\mathcal{D}_\omega(X)$. The idea is that in the determinization, states are uncertain, i.e., we only know that with a given probability the system is in one of the original states.

We start by an example of such determinization: the automaton on the right is part of the determinization of the one on the left. In general, the determinization yields an infinite automaton. In this example, we show the accessible part when starting from the state $\eta(x_1)$, the Dirac distribution of x_1 , and we denote the distributions by formal sums.



The actual definition of the determinization is as follows. Given a probabilistic transition system $\alpha: X \rightarrow \mathcal{D}_\omega(1 + A \times X)$ its determinization is the system $\bar{\alpha}: \mathcal{D}_\omega(X) \rightarrow [0, 1] \times (\mathcal{D}_\omega(X))^A$ defined by

$$\bar{\alpha}(\xi) = \langle \sum_{x \in X} \xi(x) \cdot \alpha(x)(*), \lambda a. \lambda x'. \sum_{x \in X} \xi(x) \cdot \alpha(x)(a, x') \rangle$$

for a distribution $\xi \in \mathcal{D}_\omega(X)$.

A state y in a coalgebra $\beta: Y \rightarrow [0, 1] \times Y^A$ of type G , with $\beta(y) = \langle p, f \rangle$, either terminates with probability p or given a label a it transits to a unique next state $f(a)$. Moreover, for any such deterministic coalgebra $\beta: Y \rightarrow [0, 1] \times Y^A$ of type G , there is a canonical map $out_\beta: Y \rightarrow [0, 1]^{A^*}$ given by

$$out_\beta(y)(\varepsilon) = p, \quad out_\beta(y)(aw) = out_\beta(f(a))(w).$$

In the example above, $out(\eta(x_1))(ab) = \frac{1}{6}$ and $out(\frac{1}{2}x_2 + \frac{1}{4}x_3)(c) = \frac{1}{8}$.

The map out_β is actually the unique homomorphism from β into the final G -coalgebra. The final G -coalgebra is $([0, 1]^{A^*}, \langle \varepsilon?, (-)_a \rangle)$ where for a map $\xi: A^* \rightarrow [0, 1]$, we have $\varepsilon^?(\xi) = \xi(\varepsilon)$ and $(\xi)_a = \lambda a. \lambda w. \xi(aw)$. Hence, the following diagram commutes.

$$\begin{array}{ccc} Y & \xrightarrow{\quad out_\beta \quad} & [0, 1]^{A^*} \\ \beta \downarrow & & \downarrow \langle \varepsilon?, (-)_a \rangle \\ [0, 1] \times Y^A & \xrightarrow{\quad id \times out_\beta^A \quad} & [0, 1] \times ([0, 1]^{A^*})^A \end{array}$$

The concrete definition of a determinization can be structured in the following way. We observe that there is an injective natural transformation $\delta: \mathcal{D}_\omega(1 + A \times X) \rightarrow [0, 1] \times \mathcal{D}_\omega(X)^A$, given by $\delta(\xi) = \langle \xi(*), \lambda a. \lambda x. \xi(a, x) \rangle$. The determinization map satisfies $\bar{\alpha} = \delta \circ \mu \circ \mathcal{D}_\omega \alpha$ and makes the following diagram commute

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & \mathcal{D}_\omega(X) \\ \alpha \downarrow & & \nearrow \bar{\alpha} \\ \mathcal{D}_\omega(1 + A \times X) & & \\ \delta \downarrow & \swarrow & \\ [0, 1] \times \mathcal{D}_\omega(X)^A & & \end{array}$$

To summarize, the situation is shown in the following diagram:

$$\begin{array}{ccccc} X & \xrightarrow{\eta_X} & \mathcal{D}_\omega(X) & \dashrightarrow^{\text{out}} & [0, 1]^{A^*} \\ \alpha \downarrow & & \nearrow \bar{\alpha} & & \downarrow \langle \varepsilon?, (-)_a \rangle \\ \mathcal{D}_\omega(1 + A \times X) & & & & \\ \delta \downarrow & \swarrow & & & \\ [0, 1] \times \mathcal{D}_\omega(X)^A & \dashrightarrow^{\text{id} \times \text{out}^A} & & \dashrightarrow & [0, 1] \times ([0, 1]^{A^*})^A \end{array}$$

More generally, this fits into the generalized powerset construction [16]. The generalized powerset construction can be applied to a coalgebra of type HT , for T a monad and H a functor with a T -algebra lifting ($HT(X)$ has a T -algebra structure h), or equivalently, for H such that there is a distributive law $\pi: TH \rightarrow HT$. Given a coalgebra $\gamma: X \rightarrow HTX$, where H and T satisfy the above conditions, the coalgebra $\gamma^\sharp: TX \rightarrow HTX$ obtained by applying the generalized powerset construction to γ is defined as $\gamma^\sharp = h \circ T\gamma = H\mu \circ \pi \circ T\gamma$. It can be thought of as a determinization of γ in the sense that any side effects modeled by the monad T will now be buried in the state space of the new coalgebra. Taking T to be the powerset monad and $H = 2 \times (-)^A$, the functor defining the type of deterministic automata, one obtains the usual powerset construction, which allows to define a deterministic automaton language-equivalent to a given non-deterministic automaton. The construction is applicable to $T = \mathcal{D}_\omega$ and $H = G$, since $G\mathcal{D}_\omega(X)$ has a \mathcal{D}_ω -algebra structure, leading $(\delta \circ \alpha)^\sharp = \bar{\alpha}$.

Remark 5.2 There seems to be a relationship between the functor G and the functor F , that may shed light on how to extend the current work to other functors in place of F , e.g. shapely functors. Given a functor H that is inductively built from the identity functor, constant functors, finite products and coproducts (or even if infinite coproducts in which case H can be any shapely functor), we can define a corresponding functor G_H as follows: $G_{id} = id$, $G_A = \mathcal{D}_\omega(A)$, $G_{H_1 + H_2} = G_{H_1} \times G_{H_2}$ and $G_{H_1 \times H_2} = (G_{H_2})^{H_1}$. Note that in our particular example $F = 1 + A \times -$ and $G_F = G = [0, 1] \times (-)^A$, where $[0, 1] = \mathcal{D}_\omega(1)$. Such a functor G_H may be useful to determinize $\mathcal{D}_\omega H$ -coalgebras, and a corresponding natural transformation $\delta_H: \mathcal{D}_\omega H \Rightarrow G_H \mathcal{D}_\omega$ could also be inductively defined. The details

of this generalization remain future work. In addition, the definition of expressions should change accordingly (the F-type expressions) and the trace semantics needs to be instantiated to such functors H in order to gain understanding of the situation.

We now need to formally connect the semantics given by *out* and the trace semantics given by *tr*. The first observation is the following.

Lemma 5.3 *Starting from a coalgebra $X \xrightarrow{\alpha} \mathcal{D}_\omega(1 + A \times X)$, the image of the map *out*, as depicted in the commuting diagram below, is in $\mathcal{D}(A^*)$.*

$$\begin{array}{ccccc}
 X & \xrightarrow{\eta_X} & \mathcal{D}_\omega(X) & \xrightarrow{\text{out}} & \mathcal{D}(A^*) & \xrightarrow{\quad} & [0, 1]^{A^*} \\
 \alpha \downarrow & & \nearrow \bar{\alpha} & & \downarrow \mathcal{D}_\iota & & \downarrow \langle \varepsilon?, (-)_a \rangle \\
 \mathcal{D}_\omega(1 + A \times X) & & & & \mathcal{D}(1 + A \times A^*) & & \\
 \delta \downarrow & & & & \delta \downarrow & & \\
 [0, 1] \times \mathcal{D}_\omega(X)^A & \xrightarrow{\text{id} \times \text{out}^A} & [0, 1] \times (\mathcal{D}(A^*))^A & \xrightarrow{\quad} & [0, 1] \times ([0, 1]^{A^*})^A
 \end{array}$$

Remark 5.4 A consequence of our further results, which we can also show independently, is that $\text{out} \circ \eta = \text{tr}$, which is also expected from the definition of *out* and the determinization. This is in itself a very interesting result since it shows that coalgebraic traces can be recovered via determinization. However, for the axiomatization we need another map out_\equiv and its connection to coalgebraic traces.

Our goal in the remainder of this section is to define out_\equiv and show that $\text{out}_\equiv \circ [-]$ is a Kleisli homomorphism from $(\text{Exp}, i \circ c)$ to the final F -coalgebra in $\mathcal{Kl}(\mathcal{D})$, $(A^*, \eta \circ \iota)$.

Let us start with summarizing in a diagram some of the maps we are dealing with:

$$\begin{array}{ccccc}
 & & [-] & & \\
 & \text{Exp} & \xrightarrow{[-]_\sim} & \text{Exp}/\sim & \xrightarrow{[-]_\equiv} & \text{Exp}/\equiv \\
 c \downarrow & & & c_0 \downarrow & & \\
 \mathcal{D}_\omega(1 + A \times \text{Exp}) & \xrightarrow{\mathcal{D}_\omega(1 + A \times [-]_\sim)} & \mathcal{D}_\omega(1 + A \times \text{Exp}/\sim)
 \end{array}$$

Here, $[-]_\sim$ denotes the surjective equivalence map which quotients only using the axioms for bisimilarity (all axioms except (D)), and $[-]_\equiv$ quotients with the axiom (D) . The commutativity of the square above was proved in [17], and had as consequence the soundness of the axioms w.r.t. bisimilarity. We know, however, that we cannot fill the diagram on the right side in the same way, that is, Exp/\equiv will never have a coalgebra structure making $[-]_\equiv$ a coalgebra homomorphism. Hence, we will take a different approach, inspired by [12,3].

From now on *positive convex structures* [5,6] play an important role in our work. They are the Eilenberg-Moore algebras of the monad \mathcal{D}_ω [6]. In concrete terms, a positive convex structure is an algebra with a finite convex sum operation $\bigoplus_{i \in I} p_i x_i$

for $p_i \in [0, 1]$ and $\sum_{i \in I} p_i \leq 1$, satisfying the axioms:

$$(i) \quad \bigoplus_{i \in I} p_{i,k} x_i = x_k \quad \text{if } p_{i,k} = 1 \text{ for } i = k \text{ and } p_{i,k} = 0 \text{ otherwise}$$

$$(ii) \quad \bigoplus_{i \in I} p_i \left(\bigoplus_{j \in J} q_{i,j} x_j \right) = \bigoplus_{j \in J} \left(\sum_{i \in I} p_i q_{i,j} \right) x_j.$$

Given a positive convex structure \bigoplus on a set X , it provides a \mathcal{D}_ω -algebra $\mathbf{a}: \mathcal{D}_\omega(X) \rightarrow X$ by $\mathbf{a}(\xi) = \bigoplus_{x \in \text{supp}(\xi)} \xi(x)x$. Our first observation is that Exp/\sim carries a positive convex structure.

Proposition 5.5 (Exp/ \sim is a PCA) *The set Exp/\sim has a positive convex algebra structure, that is, for every $[E_1]_\sim, \dots, [E_n]_\sim \in \text{Exp}/\sim$ and $p_1, \dots, p_n \in [0, 1]$ satisfying $\sum_{i=1}^n p_i \leq 1$, the operation given by*

$$\bigoplus_i p_i [E_i]_\sim = \left[\bigoplus_i p_i E_i \right]_\sim$$

is a positive convex structure.

Next, we observe that Exp/\equiv also has a positive convex structure.

Proposition 5.6 (Exp/ \equiv is a PCA) *The set Exp/\equiv has a positive convex algebra structure, that is, for every $[E_1], \dots, [E_n] \in \text{Exp}/\equiv$ and $p_1, \dots, p_n \in [0, 1]$ satisfying $\sum_{i=1}^n p_i \leq 1$, the operation given by*

$$\bigoplus_i p_i [E_i] = \left[\bigoplus_i p_i E_i \right]$$

is a positive convex structure. Moreover $[-]_\equiv$ is an algebra homomorphism from Exp/\sim to Exp/\equiv .

Let \mathbf{a}_\sim denote the algebra map on Exp/\sim , $\mathbf{a}_\sim: \mathcal{D}_\omega(\text{Exp}/\sim) \rightarrow \text{Exp}/\sim$, given by the positive convex structure and \mathbf{a}_\equiv the algebra map on Exp/\equiv , $\mathbf{a}_\equiv: \mathcal{D}_\omega(\text{Exp}/\equiv) \rightarrow \text{Exp}/\equiv$, making $[-]_\equiv$ an algebra homomorphism.

We can then expand the above diagram in the following way, where the coalgebra

structure d exists because of Lemma 5.7 below:

$$\begin{array}{ccccc}
 & & [-] & & \\
 & \text{Exp} & \xrightarrow{[-]_{\sim}} & \text{Exp}/_{\sim} & \xrightarrow{[-]_{\equiv}} & \text{Exp}/_{\equiv} \\
 & \downarrow c & & \downarrow c_0 & & \downarrow d \\
 \mathcal{D}_{\omega}(1 + A \times \text{Exp}) & \xrightarrow{\mathcal{D}_{\omega}(1 + A \times [-]_{\sim})} & \mathcal{D}_{\omega}(1 + A \times \text{Exp}/_{\sim}) & & \\
 & & \downarrow \delta & & \\
 & & [0, 1] \times \mathcal{D}_{\omega}(\text{Exp}/_{\sim})^A & & \\
 & & \downarrow id \times \mathbf{a}_{\sim}^A & & \\
 & & [0, 1] \times \text{Exp}/_{\sim}^A & \xrightarrow{id \times [-]_{\equiv}^A} & [0, 1] \times \text{Exp}/_{\equiv}^A
 \end{array}$$

Lemma 5.7 *Let $E_1, E_2 \in \text{Exp}$ such that $E_1 \equiv E_2$. Then*

$$G[-]_{\equiv} \circ G\mathbf{a}_{\sim} \circ \delta \circ c_0([E_1]_{\sim}) = G[-]_{\equiv} \circ G\mathbf{a}_{\sim} \circ \delta \circ c_0([E_2]_{\sim}).$$

In concrete terms, the coalgebra $(\text{Exp}/_{\equiv}, d)$ behaves as follows. Let E be an expression. We first notice that there exists an unfolded expression $p \cdot * \oplus_i p_i \cdot a_i \cdot E_i$ such that $E \equiv p \cdot * \oplus_i p_i \cdot a_i \cdot E_i$. Then we have

$$d([E]) = \langle p, f \rangle \quad \text{where} \quad f: A \rightarrow \text{Exp}/_{\equiv}, \quad f(a) = \bigoplus_{i: a_i = a} p_i [E_i]. \quad (1)$$

Let out_{\equiv} be the unique homomorphism from $(\text{Exp}/_{\equiv}, d)$ to the final G -coalgebra. This is the map we are after, in order to show soundness and completeness.

There are many generic properties of determinizations that are out of the scope of this paper and we leave their elaboration for future work. We only state few here, in order to reveal connections between $(\text{Exp}/_{\equiv}, d)$ and the determinization of $(\text{Exp}/_{\sim}, c_0)$ and shed some light on the overall situation. First, we note that

$$c_0 \circ \mathbf{a}_{\sim} = \mu \circ \mathcal{D}_{\omega} c_0 \quad (2)$$

which is a consequence of the definitions and the property $c(\oplus_i p_i E_i) = \sum p_i \cdot c(E_i)$, that can readily be checked. This means that c_0 is an algebra homomorphism from $(\text{Exp}/_{\sim}, \mathbf{a}_{\sim})$ to $(\mathcal{D}_{\omega}(1 + A \times \text{Exp}/_{\sim}), \mu)$, the free PCA, and yields that the determinization $\overline{c_0}$ of $(\text{Exp}/_{\sim}, c_0)$ satisfies

$$\overline{c_0} = \delta \circ c_0 \circ \mathbf{a}_{\sim} \quad (3)$$

implying further that \mathbf{a}_{\sim} is a coalgebra homomorphism from the determinization $(\mathcal{D}_{\omega} \text{Exp}/_{\sim}, \overline{c_0})$ to the G -coalgebra $(\text{Exp}/_{\sim}, G\mathbf{a}_{\sim} \circ \delta \circ c_0)$, i.e.

$$G\mathbf{a}_{\sim} \circ \overline{c_0} = G\mathbf{a}_{\sim} \circ \delta \circ c_0 \circ \mathbf{a}_{\sim}. \quad (4)$$

Let out be the unique homomorphism from the determinization $(\mathcal{D}_{\omega} \text{Exp}/_{\sim}, \overline{c_0})$ to the final G -coalgebra and out_{\sim} the unique homomorphism from $(\text{Exp}/_{\sim}, G\mathbf{a}_{\sim} \circ \delta \circ c_0)$ to the final G -coalgebra.

Lemma 5.8 *The final coalgebra homomorphisms satisfy*

$$out = out_{\sim} \circ \mathbf{a}_{\sim}, \quad out_{\sim} = out \circ \eta, \quad out_{\sim} = out_{\equiv} \circ [-]_{\equiv}.$$

Lemma 5.3 stated that the image of out , coming from a determinization, is in $\mathcal{D}(A^*)$. Now, using Lemma 5.8 we can show that the image of out_{\equiv} is in $\mathcal{D}(A^*)$ as well.

Lemma 5.9 *The unique homomorphism out_{\equiv} into the final G -coalgebra from the G -coalgebra $(\text{Exp}/_{\equiv}, d)$ makes the following diagram commute.*

$$\begin{array}{ccccc}
 \text{Exp}/_{\equiv} & \xrightarrow{\quad} & \xrightarrow{\quad out_{\equiv} \quad} & \mathcal{D}(A^*) & \xrightarrow{\quad} & [0, 1]^{A^*} \\
 \downarrow d & & & \downarrow \mathcal{D}\iota & & \downarrow \langle \varepsilon?, (-)_a \rangle \\
 & & & \mathcal{D}(1 + A \times A^*) & & \\
 & & & \downarrow \delta & & \\
 [0, 1] \times \text{Exp}/_{\equiv}^A & \xrightarrow{\quad} & \xrightarrow{\quad} & [0, 1] \times (\mathcal{D}(A^*))^A & \xrightarrow{\quad} & [0, 1] \times ([0, 1]^{A^*})^A
 \end{array}$$

Having this as a first step, we can relate the semantics induced by out_{\equiv} with trace semantics.

Proposition 5.10 *The map $out_{\equiv} \circ [-]$ is a Kleisli homomorphism from $(\text{Exp}, i \circ c)$ to $(A^*, \eta \circ \iota)$. Therefore, by finality, $out_{\equiv} \circ [-] = tr$.*

This yields the soundness of the axiomatization, see Figure 2, and paves the road to completeness.

Theorem 5.11 (Soundness) *For all $E_1, E_2 \in \text{Exp}$, $E_1 \equiv E_2 \Rightarrow E_1 \sim_{\text{tr}} E_2$.*

5.3 Completeness

To prove completeness, as announced in Figure 2, it remains to prove that out_{\equiv} is an injective map. Borrowing inspiration from [8], we proceed as follows. We first factorize the map out_{\equiv} into a surjective map followed by an injective one:

$$out_{\equiv} = \text{Exp}/_{\equiv} \xrightarrow{e} I \xrightarrow{m} \mathcal{D}_{\omega}(A^*)$$

Then we show that a “variant” of $(\text{Exp}/_{\equiv}, d)$ is final in a certain category of coalgebras to which the factorization carries over. Finally, we show that a “variant” of (I, g) , induced by the factorization, is in the same category and is final as well. This proves that e is an isomorphism, and hence out_{\equiv} is mono.

The difficulty is that $(\text{Exp}/_{\equiv}, d)$ is not final in the category of G -coalgebras on **Sets**, since d is not an isomorphism. Therefore, we move to another category (of coalgebras) which was already implicitly present for a while.

5.4 Coalgebras over algebras

As base category, instead of **Sets**, we take **PCA**, the category of Eilenberg-Moore algebras of \mathcal{D}_{ω} .

Then we consider coalgebras on **PCA**. For a functor F on **PCA**, an F -coalgebra is a pair $((X, \mathfrak{a}), \alpha)$ where α is an algebra homomorphism from (X, \mathfrak{a}) to $F(X, \mathfrak{a})$, both in **PCA**. An F -coalgebra homomorphism from $((X, \mathfrak{a}), \alpha)$ to $((Y, \mathfrak{b}), \beta)$ is a map $h: X \rightarrow Y$ that is both an algebra and a coalgebra homomorphism, i.e., $\mathfrak{b} \circ \mathcal{D}_\omega(h) = h \circ \mathfrak{a}$ and $\beta \circ h = F(h) \circ \alpha$.

The functor $G = [0, 1] \times (-)^A$ on **Sets** lifts to a functor on **PCA**, denoted also by G , as follows. We have $G(X, \mathfrak{a}) = (GX, \mathfrak{a}_G)$ where \mathfrak{a}_G is defined “pointwise” by

$$\bigoplus_i p_i \langle o_i, f_i \rangle = \langle \sum_i p_i \cdot o_i, f \rangle, \quad \text{with } f(a) = \bigoplus_i p_i f_i(a)$$

for $\langle o_i, f_i \rangle \in GX$. Note that the second \bigoplus is from the algebra (X, \mathfrak{a}) .

Any G -coalgebra (X, α) on **Sets** with a PCA structure \mathfrak{a} such that α is an algebra homomorphism from (X, \mathfrak{a}) to $G(X, \mathfrak{a})$ is a G -coalgebra $((X, \mathfrak{a}), \alpha)$ on **PCA**.

Example 5.12 Every determinization is a G -coalgebra on **PCA**, with carrier the free PCA $(\mathcal{D}_\omega(X), \mu)$. Moreover, $((\text{Exp}/\sim, \mathfrak{a}_\sim), G\mathfrak{a}_\sim \circ \delta \circ c_0)$ and $((\text{Exp}/\equiv, \mathfrak{a}_\equiv), d)$ are G -coalgebras on **PCA**.

The carrier of the final G -coalgebra on **Sets** has a PCA structure \mathfrak{z} , making it final in the category of G -coalgebras on **PCA**. This is a consequence of a general result, see e.g. [2,8], and the fact that the coalgebra structure on the final is an algebra homomorphism. In concrete terms \mathfrak{z} is given by

$$\bigoplus_i p_i \xi_i(w) = \sum_i p_i \cdot \xi_i(w), \quad \text{for } \xi_i \in [0, 1]^{A^*}, w \in A^*.$$

Therefore, for any G -coalgebra (X, α) on **Sets** such that X has a PCA structure \mathfrak{a} , the final coalgebra map out is also the final coalgebra map from the G -coalgebra $((X, \mathfrak{a}), \alpha)$ on **PCA**. That is, out is also an algebra homomorphism from (X, \mathfrak{a}) to $([0, 1]^{A^*}, \mathfrak{z})$, as shown in the diagram below.

$$\begin{array}{ccc} \mathcal{D}_\omega X & \xrightarrow{\quad \mathcal{D}_\omega out \quad} & \mathcal{D}_\omega([0, 1]^{A^*}) \\ \mathfrak{a} \downarrow & & \downarrow \mathfrak{z} \\ X & \xrightarrow{\quad out \quad} & [0, 1]^{A^*} \\ \alpha \downarrow & & \downarrow \langle \varepsilon?, (-)_a \rangle \\ [0, 1] \times X^A & \xrightarrow{\quad id \times out^A \quad} & [0, 1] \times ([0, 1]^{A^*})^A \end{array}$$

At this point it is important to mention that $\mathcal{D}(A^*)$ also has a PCA structure, namely $\mu \circ i$. Moreover, the inclusion $\mathcal{D}(A^*) \rightarrow [0, 1]^{A^*}$ is an algebra homomorphism from $(\mathcal{D}(A^*), \mu \circ i)$ to $([0, 1]^{A^*}, \mathfrak{z})$. Also $((\mathcal{D}(A^*), \mu \circ i), \delta \circ \mathcal{D}l)$ is a G -coalgebra on **PCA**. As a result, we get the following lemma which is applicable to any determinization (by Lemma 5.3), as well as to out_\sim and out_\equiv (by Lemma 5.9),

Lemma 5.13 *If the image of the final coalgebra homomorphism out of a G -coalgebra $((X, \mathfrak{a}), \alpha)$ lives in $\mathcal{D}(A^*)$, then out is a coalgebra homomorphism from $((X, \mathfrak{a}), \alpha)$ to $((\mathcal{D}(A^*), \mu \circ i), \delta \circ \mathcal{D}l)$.*

Next we factorize. Factorizations in **Sets** carry over to factorizations of coalgebras over Eilenberg-Moore algebras, see e.g. [10, Theorem 1.3.7]. We have the following situation.

$$\begin{array}{ccccc}
 (\text{Exp}/\equiv, \mathbf{a}_\equiv) & \xrightarrow{\quad e \quad} & (I, \mathbf{a}_I) & \xrightarrow{\quad m \quad} & (\mathcal{D}(A^*), \mu) \\
 \downarrow d & & \downarrow g & & \downarrow \delta \circ \mathcal{D}_\iota \\
 G(\text{Exp}/\equiv, \mathbf{a}_\equiv) & \longrightarrow & G(I, \mathbf{a}_I) & \longrightarrow & G(\mathcal{D}_\omega(A^*), \mu)
 \end{array}$$

out_\equiv (curved arrow from $(\text{Exp}/\equiv, \mathbf{a}_\equiv)$ to $(\mathcal{D}(A^*), \mu)$)

Still, G -coalgebras on **PCA** are not sufficient for our goal of showing finality of $((\text{Exp}/\equiv, \mathbf{a}_\equiv), d)$ since d is still not an isomorphism. For this reason we consider (yet) another functor \hat{G} on **PCA** which is a subfunctor of G . Let (X, \mathbf{a}) be a PCA. The functor \hat{G} is defined on object as

$$\begin{aligned}
 \hat{G}(X, \mathbf{a}) = \{ \langle o, f \rangle \in [0, 1] \times X^A \mid \forall a \in A. \exists p_a^i \in [0, 1], v_a^i \in V. f(a) = \bigoplus_i p_a^i v_a^i \\
 \text{and } \sum_a \sum_i p_a^i \leq 1 - o \}
 \end{aligned}$$

and on arrows just like G .

5.5 Finality for completeness

We show that $((\text{Exp}/\equiv, \mathbf{a}_\equiv), d)$ is final in the category of *locally-finite* \hat{G} -coalgebras, denoted by $\mathbf{PCA}_f(\hat{G})$, that we define next.

A \hat{G} -coalgebra $((X, \mathbf{a}), \alpha)$ is locally finite if for every $x \in X$ there exists a finitely generated subalgebra of (X, \mathbf{a}) with states Y and $x \in Y$ which is a subcoalgebra of $((X, \mathbf{a}), \alpha)$, i.e., Y is closed under the coalgebra structure α . An algebra (Y, \mathbf{a}_Y) in **PCA** is finitely generated if there exists a split epi e_B from $(\mathcal{D}_\omega B, \mu)$ to (Y, \mathbf{a}_Y) for some finite set B .

Proposition 5.14 $((\text{Exp}/\equiv, \mathbf{a}_\equiv), d)$ is final in $\mathbf{PCA}_f(\hat{G})$.

The next property follows from [1, Proposition 1.69] and ensures that $((I, \mathbf{a}_I), g)$ is also final in $\mathbf{PCA}_f(\hat{G})$.

Lemma 5.15 The category $\mathbf{PCA}_f(\hat{G})$ is closed under homomorphic images.

Hence we have reached our goal, stated in the following lemma.

Lemma 5.16 The map $\text{out}_\equiv: \text{Exp}/\equiv \rightarrow \mathcal{D}(A^*)$ is injective.

This is the last ingredient we needed for completeness.

Theorem 5.17 (Completeness) For all $E_1, E_2 \in \text{Exp}$, $E_1 \sim_{\text{tr}} E_2 \Rightarrow E_1 \equiv E_2$.

6 Conclusions

In this paper, we presented the first sound and complete axiomatization of (finite, terminating) trace semantics for generative probabilistic transition systems (with explicit termination).

Inspired by the work of Rabinovich, who axiomatized trace semantics for LTS, we took as basis a calculus sound and complete w.r.t. bisimilarity and we extended it with an extra axiom. Our approach is coalgebraic. This means that constructions and results are phrased in quite general terms which might be helpful to pinpoint which conditions on the functor type of the system are crucial and which generalizations are possible.

The fact that a sound and complete calculus w.r.t. bisimilarity can be extended to a sound and complete calculus w.r.t. coalgebraic language equivalence has recently been studied by Bonsangue, Milius and the first author [3]. The class of systems they consider is however different from the one considered in this paper (formally, they consider coalgebras for FT , with F a functor and T a monad, such that F preserves T -algebras). In the determinization step, we relate to the powerset construction [16] which also served as basis for the proofs in [3]. However, we had to deal with the extra difficulty of showing that the semantics of the determinized automaton is actually a subdistribution over words (that is, an element of $\mathcal{D}(A^*)$) and not just any arbitrary function $[0, 1]^{A^*}$. This fact is quite instructive and we believe that it will serve as basis to clarify the connection between the coalgebraic trace semantics of [7] and the coalgebraic language equivalence of [3] and describe a framework in which both semantics can be considered.

Acknowledgement

We are indebted to Marcello Bonsangue and Stefan Milius for valuable suggestions and pointers to the literature.

References

- [1] Jiří Adámek and Jiří Rosický. *Locally Presentable and Accessible Categories* (London Mathematical Society Lecture Note Series 189), Cambridge University Press (1994)
- [2] Falk Bartels. On generalized coinduction and probabilistic specification formats. Ph.D. thesis, Vrije Universiteit Amsterdam (2004)
- [3] Marcello Bonsangue, Stefan Milius, and Alexandra Silva. Sound and complete axiomatizations of coalgebraic language equivalence. preprint, 2011.
- [4] Marcello M. Bonsangue, Jan J. M. M. Rutten, and Alexandra Silva. An algebra for Kripke polynomial coalgebras. In *Proc. LICS*, pages 49–58. IEEE Computer Society, 2009.
- [5] Ernst-Erich Doberkat. Eilenberg-Moore algebras for stochastic relations. *Inf. Comput.*, 204(12):1756–1781, 2006.
- [6] Ernst-Erich Doberkat. Erratum and addendum: Eilenberg-Moore algebras for stochastic relations. *Inf. Comput.*, 206(12):1476–1484, 2008.
- [7] Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Logical Methods in Computer Science*, 3(4), 2007.

- [8] Bart Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In K. Futatsugi et al., editors, *Essays Dedicated to Joseph A. Goguen*, volume 4060 of *Lecture Notes in Computer Science*, pages 375–404. Springer, 2006.
- [9] Peter Johnstone. Adjoint lifting theorems for categories of algebras. *Bull. London Math. Soc.* 7, 294–297 (1975)
- [10] Alexander Kurz. Logics for coalgebras and applications to computer science. Ph.D. thesis, Ludwig-Maximilian-Universität München (2000)
- [11] Robin Milner. A complete inference system for a class of regular behaviours. *J. Comput. Syst. Sci.*, 28(3):439–466, 1984.
- [12] Stefan Milius. A sound and complete calculus for finite stream afteruits. In *Proc. LICS*, pages 421–430. IEEE Computer Society, 2010.
- [13] Sumit Nain and Moshe Y. Vardi. Trace semantics is fully abstract. In *Proc. LICS*, pages 59–68. IEEE Computer Society, 2009.
- [14] Alexander Moshe Rabinovich. A complete axiomatisation for trace congruence of finite state behaviors. In S. Brookes et al., editors, *Proc. MFPS*, volume 802 of *Lecture Notes in Computer Science*, pages 530–543. Springer, 1993.
- [15] Roberto Segala. A compositional trace-based semantics for probabilistic automata. In *Proc. CONCUR*, volume 9622 of *Lecture Notes in Computer Science*, pages 234–248. Springer, 1995.
- [16] Alexandra Silva, Filippo Bonchi, Marcello Bonsangue, and Jan Rutten. Generalizing the powerset construction, coalgebraically. In K. Lodaya and M. Mahajan, editors, *Proc. FSTTCS 2010*, volume 8 of *LIPICs*, pages 272–283, 2010.
- [17] Alexandra Silva, Filippo Bonchi, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Quantitative Kleene coalgebras. *Inf. Comput.* 209(5): 822–849, 2011.
- [18] Alexandra Silva and Ana Sokolova. Sound and complete axiomatization of trace semantics for probabilistic systems. Technical report SEN-1108, CWI, 2011.
- [19] Rob J. van Glabbeek, Scott A. Smolka, and Bernhard Steffen. Reactive, generative and stratified models of probabilistic processes. *Inf. Comput.*, 121(1):59–80, 1995.