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
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 coalge-
braic 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 axiomati-
izations 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

\[
\begin{array}{c}
\cdot \xrightarrow{a, \frac{1}{2}} \cdot \xrightarrow{b, \frac{1}{3}} \cdot \xrightarrow{1} \ast \\
\end{array}
\]

\[
\begin{array}{c}
\cdot \xrightarrow{a, \frac{1}{3}} \cdot \xrightarrow{b, \frac{1}{2}} \cdot \xrightarrow{1} \ast \\
\end{array}
\]

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 an-
other 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 coin-
cides 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 \(\text{Sets}\) with \(T\) a suitable monad modeling branching and \(F\) a suitable functor mod-
eling 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 machin-
ery, 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 pro-
cess 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 \(D_\omega(1 + A \times id)\),
where \(D_\omega\) 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 seman-
tics 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, respec-
tively. In Section 4 we present the syntax of expressions for quantitative transition
systems, followed by the axiomatization in Section 5 where the main results (sound-
ness 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 $\text{Sets}$, the category of sets
and functions. An $F$-coalgebra is a pair $(X, \alpha : X \to F(X))$ where $X$ is the carrier
set, the set of states, and $\alpha$ is the coalgebra transition map. An $F$-algebra is a pair
$(X, \mathfrak{a} : F(X) \to X)$. For brevity, we often identify a (co)algebra with its (co)algebra
map. Given two $F$-coalgebras $\alpha : X \to F(X)$ and $\beta : Y \to F(Y)$, a coalgebra
homomorphism from $\alpha$ to $\beta$ is a map $h : X \to Y$ such that $\beta \circ h = F(h) \circ \alpha$. Given
two $F$-algebras $\mathfrak{a} : F(X) \to X$ and $\mathfrak{b} : F(Y) \to Y$, an algebra homomorphism from
$\mathfrak{a}$ to $\mathfrak{b}$ is a map $h : X \to Y$ 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 $\alpha$ there is a unique homomorphism $\text{beh}_\alpha$ 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 $\alpha : X \to F(X)$, $x \sim y \iff \text{beh}_\alpha(x) = \text{beh}_\alpha(y)$.

**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 $X \to TF(X)$ 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 $p \eta, \mu$ 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 $\eta_X : X \to T(X)$ and $\mu_X : TT(X) \to T(X)$ for each set $X$ satisfying a naturality condition. The unit and multiplication satisfy the compatibility conditions $\mu_X \circ \eta_{T(X)} = \mu_X \circ T\eta_X = id$ 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 $\eta$ embeds a non-branching behavior as a trivial branching (with a single branch) whereas the multiplication $\mu$ “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.

```
X  Y  Z
\downarrow \downarrow \downarrow
u \mu \mu
\uparrow \uparrow \uparrow
x y z
```

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

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

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

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 $\mathcal{K}(T)$, with $\overline{F}(X) = F(X)$ and for a Kleisli arrow $f : X \to Y$, i.e., a map $f : X \to TY$, $\overline{F}(f) = \lambda \circ F(f)$. Hence $TF$-coalgebras in Sets are $\overline{F}$-coalgebras in $\mathcal{K}(T)$.
- The Kleisli category $\mathcal{K}(T)$ is suitably order-enriched, with order $\leq$ on Kleisli homsets, bottom element $\bot$ and suprema of directed subsets.
- The lifting $\overline{F} : \mathcal{K}(T) \to \mathcal{K}(T)$ 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 $\mathbf{Sets}$, then the lifted coalgebra $J(\iota^{-1}) = \eta \circ \iota^{-1}: I \to TF(I)$ is final $\overline{F}$-coalgebra in $\mathcal{K}(T)$.

This enables defining trace semantics for $TF$-coalgebras in $\mathbf{Sets}$ as the final coalgebra semantics for $F$-coalgebras in $\mathcal{K}(T)$. More precisely, for a coalgebra $\alpha: X \to TFX$ in $\mathbf{Sets}$, i.e., $\alpha: X \xrightarrow{\sim} FY$ in $\mathcal{K}(T)$, we denote by $\text{tr}_\alpha$ the final coalgebra map in $\mathcal{K}(T)$, 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 $D_\omega$, and the lift monad $1 + \text{id}$, 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 $F$ 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 $D_\omega$ and the linear-behavior functor $F = 1 + A \times \text{id}$ where $A$ is a set of labels and $1 = \{\ast\}$ is a singleton set, used to model termination. The monad $D_\omega$ assigns to a set $X$ the set

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

and to a function $f: X \to Y$, the function $D_\omega(f): D_\omega(X) \to D_\omega(Y)$:

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

The unit of $D_\omega$ is given by $\eta_X(x) = (x \mapsto 1)$ and the multiplication by

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

Hence, our probabilistic transition systems are $D_\omega(1 + A \times \text{id})$-coalgebras on $\mathbf{Sets}$. The monad $D_\omega$ provides probabilistic branching. The finite support requirement ensures finite branching and is necessary for representing probabilistic transition systems by finite expressions. The functor $1 + A \times \text{id}$ provides linear behavior, in which a state can either successfully terminate or make a labelled transition to another state.
Given a probabilistic transition system $\alpha : X \rightarrow D_\omega (1 + A \times X)$ we write

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

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

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

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 $D_\omega$ 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

$$\xymatrix{ x \ar@/^/[r]^{1/2} & \ar@/_/[l]^{1/2} * }.$$

The trace of state $x$ is the distribution that assigns probability $\frac{1}{2n+1}$ to the word $a^n$ for all $n \in \mathbb{N}$ and hence has infinite support. In terms of the generic trace theory requirements, $D_\omega$ 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 $D_\omega$ by dropping the finite support condition. We will apply the generic trace results by using the natural injection $i : D_\omega(X) \rightarrow \mathcal{D}(X)$. 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 $\mathcal{K}(\mathcal{D})$. It is defined by $\lambda_X(*) = \eta(*)$ and $\lambda_X(a, \xi) = \lambda(a, x).\xi(x)$ 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 $\mathcal{K}(\mathcal{D})$ 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 $X \xrightarrow{\alpha} D_\omega(1 + A \times X)$, is defined by applying the generic trace theory to the coalgebra $X \xrightarrow{\alpha} D_\omega(1 + A \times X) \xrightarrow{\iota} D(1 + A \times X)$, as we depict on the right, and can be instantiated to the concrete definition:

$$\text{tr}(x)(\varepsilon) = p, \quad \text{if} \ x \xrightarrow{p} *, \quad \text{tr}(x)(aw) = \sum_{x \xrightarrow{a,b} y} p \cdot \text{tr}(y)(w).$$

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 $\sigma$-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., $x \sim y \Rightarrow x \sim_{tr} y$.

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 $\text{Exp}$ of expressions for quantitative transition systems is given by the closed expressions contained in the following BNF, for $a \in A$ and $x \in X$:

$$\text{E} ::= \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x. E^g \mid x \quad (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1)$$

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

$$F_i ::= * \mid a \cdot E$$

The operator $\mu$ 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 $\mu$-expressions are used to represent loops: a $\mu$-expression behaves the same as its unfolding. We make this precise by providing the set of expressions with a coalgebraic structure.

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

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

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, $s \sim_{tr} E$.

**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).

$$\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot * \quad \frac{1}{2} \cdot a \cdot (\frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot c \cdot 1 \cdot *)$$

$$\frac{1}{2} \cdot a \cdot \mu x. (\frac{1}{2} \cdot a \cdot x \oplus \frac{1}{2} \cdot *) \quad \frac{1}{2} \cdot a \cdot \mu x. (\frac{1}{2} \cdot a \cdot x \oplus \frac{1}{2} \cdot *)$$
$$\oplus_{\frac{1}{2}} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{4} \cdot *) \quad \oplus_{\frac{1}{4}} \cdot a \cdot \mu y. (\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot *)$$

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 $(S, \alpha)$ and $s \in S$ such that $E \sim_{tr} s$. Conversely, for every locally finite probabilistic transition system $(S, \alpha)$ and $s \in S$ there exists an expression $E \in \text{Exp}$ such that $s \sim_{tr} E$.

**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 nulary 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 $r_k = p_i \cdot F''_k = F_i$ for $i \in I$ and $r_k = q_j \cdot F''_k = F'_j$ for $k = j \in J$. Clearly, we then have $\bigoplus_{i \in I} p_i \cdot F_i = (p_1 \cdot F_1 \oplus (p_2 \cdot F_2 \oplus \cdots))$

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

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

Note that $p \left( \bigoplus_{i \in I} p_i \cdot E_i \right)$ 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].
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.,

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

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 $p_2 = \frac{1}{4}$:

$$\left(\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot *\right) + \left(\frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot *\right) \overset{(D)}{=} \frac{1}{2} \cdot a \cdot \left(\frac{1}{2}(\frac{2}{3} \cdot b \cdot 1 \cdot *) + \frac{1}{4}(1 \cdot c \cdot 1 \cdot *)\right)
\quad - \frac{1}{2} \cdot a \cdot \left(\frac{1}{3} \cdot b \cdot 1 \cdot * + \frac{1}{4} \cdot c \cdot 1 \cdot *\right).$$

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

$$\frac{1}{2} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y + \frac{1}{4} \cdot *\right) \equiv \frac{1}{4} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y + \frac{1}{2} \cdot *\right)$$
Soundness | Completeness
---|---
$E_1 \equiv E_2$ | $E_1 \sim_{\text{tr}} E_2$
$\Leftrightarrow [E_1] = [E_2]$ | $\Leftrightarrow \text{tr}(E_1) = \text{tr}(E_2)$
$\Leftrightarrow \text{out}_\equiv([E_1]) = \text{out}_\equiv([E_2])$ | $\Leftrightarrow \text{out}_\equiv([E_1]) = \text{out}_\equiv([E_2])$

Fig. 2. Soundness and completeness. $(\ast)$: existence of $\text{out}_\equiv$, $(\triangle)$: $\text{out}_\equiv \circ [-] - \text{tr}$, $(\triangledown)$: $\text{out}_\equiv$ is injective.

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

\[
\begin{align*}
\frac{1}{2} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot \ast\right) &= \frac{1}{3} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot \ast\right) \\
\Leftrightarrow \frac{1}{2} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot \ast\right) &= \frac{1}{3} \cdot a \cdot \left(\frac{1}{3} \cdot a \cdot E \oplus \frac{1}{2} \cdot \ast\right) \quad (\text{Cong} \text{ and } \text{FP}) \\
\frac{1}{2} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot \ast\right) &= \frac{1}{3} \cdot a \cdot \frac{1}{3} \cdot a \cdot E \oplus \frac{1}{2} \cdot a \cdot \frac{1}{4} \cdot \ast \quad (D \text{ with } p = \frac{1}{4}, p_1 = \frac{1}{3} \text{ and } p_2 = \frac{1}{2}) \\
\frac{1}{2} \cdot a \cdot \mu y. \left(\frac{1}{3} \cdot a \cdot y \oplus \frac{1}{2} \cdot \ast\right) &= \frac{1}{3} \cdot a \cdot \left(\frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot \ast\right) \quad (D \text{ with } p = \frac{1}{2}, p_1 = \frac{1}{3} \text{ and } p_2 = \frac{1}{2}) \\
\Leftrightarrow \frac{1}{2} \cdot a \cdot \left(\frac{1}{6} \cdot a \cdot y \oplus \frac{1}{4} \cdot \ast\right) &= \frac{1}{3} \cdot a \cdot E \oplus \frac{1}{4} \cdot \ast \quad (\text{Cong}) \\
\Leftrightarrow \frac{1}{3} \cdot a \cdot \left(\frac{1}{6} \cdot a \cdot y \oplus \frac{1}{2} \cdot \ast\right) \oplus \frac{1}{4} \cdot \ast &= \frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot \ast \quad (\text{UFP}) \\
\Leftrightarrow \frac{1}{3} \cdot a \cdot \left(\frac{1}{6} \cdot a \cdot E \oplus \frac{1}{4} \cdot \ast\right) &= \frac{1}{6} \cdot a \cdot E \quad (\text{Cong}) \\
\Leftrightarrow \frac{1}{6} \cdot a \cdot \left(\frac{1}{3} \cdot a \cdot E \oplus \frac{1}{2} \cdot \ast\right) &= \frac{1}{6} \cdot a \cdot E \quad (D \text{ twice}) \\
\Leftrightarrow E = E \quad (\text{Cong} \text{ and } \text{FP})
\end{align*}
\]

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 $\text{tr}$ is equal to a composition of two maps $\text{out}_\equiv \circ [-]$, where $\text{out}_\equiv$ 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 $\text{out}_\equiv$ and show that $\text{out}_\equiv \circ [-]$ is a Kleisli homomorphism from $(\text{Exp}, i \circ c)$ to the final $(A^*, \eta \circ i)$, which by finality yields $\text{out}_\equiv \circ [-] = \text{tr}$ and soundness follows; in Section 5.3, we show that $\text{out}_\equiv$ 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

\[ \alpha: X \to \mathcal{D}_\omega(1 + A \times X) \]

is a “deterministic” system of type \( G = [0, 1] \times \text{id}^A \) 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 \( \eta \), 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 \( \alpha: X \to \mathcal{D}_\omega(1 + A \times X) \) its determinization is the system \( \overline{\alpha}: \mathcal{D}_\omega(X) \to [0, 1] \times (\mathcal{D}_\omega(X))^A \) defined by

\[ \overline{\alpha}(\xi) = \left( \sum_{x \in X} \xi(x) \cdot \alpha(x)(\varepsilon), \lambda a. \lambda x'. \sum_{x \in X} \xi(x) \cdot \alpha(x)(a, x') \right) \]

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

A state \( y \) in a coalgebra \( \beta: Y \to [0, 1] \times Y^A \) of type \( G \), with \( \beta(y) = (p, f) \), 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 \( \beta: Y \to [0, 1] \times Y^A \) of type \( G \), there is a canonical map \( \text{out}_\beta: Y \to [0, 1]^{A^*} \) given by

\[ \text{out}_\beta(y)(\varepsilon) = p, \quad \text{out}_\beta(y)(aw) = \text{out}_\beta(f(a))(w). \]

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

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

\[
\begin{array}{ccc}
Y & \xrightarrow{\text{out}_\beta} & [0, 1]^{A^*} \\
\beta \downarrow & & \downarrow \langle \varepsilon, (\cdot)_a \rangle \\
[0, 1] \times Y^A & \xrightarrow{id \times \text{out}_\beta} & ([0, 1]^{A^*})^A
\end{array}
\]
The concrete definition of a determinization can be structured in the following way. We observe that there is an injective natural transformation $\delta: D_\omega(1 + A \times X) \to [0, 1] \times D_\omega(X)^A$, given by $\delta(\xi) = \langle \xi(\_), \lambda a.\lambda x.\xi(a, x) \rangle$. The determinization map satisfies $\alpha = \delta \circ \mu \circ D_\omega \alpha$ and makes the following diagram commute:

$\begin{align*}
X & \xrightarrow{\eta X} D_\omega(X) \\
& \downarrow \alpha \\
D_\omega(1 + A \times X) & \xrightarrow{\alpha} \\
& \downarrow \delta \\
[0, 1] \times D_\omega(X)^A & \xrightarrow{\alpha} \\
& \downarrow D_\omega \alpha \\
& \alpha \\
& \delta \\
& \downarrow r_0, 1 \\
& \downarrow s \\
& \downarrow D_\omega \alpha.
\end{align*}$

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

$\begin{align*}
X & \xrightarrow{\eta X} D_\omega(X) \xrightarrow{\alpha} [0, 1]^A^* \\
& \downarrow \alpha \\
D_\omega(1 + A \times X) & \xrightarrow{\alpha} \\
& \downarrow \delta \\
[0, 1] \times D_\omega(X)^A & \xrightarrow{\alpha} \\
& \downarrow D_\omega \alpha \\
& \alpha \\
& \delta \\
& \downarrow r_0, 1 \\
& \downarrow s \\
& \downarrow D_\omega \alpha. 
\end{align*}$

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 $\pi: TH \to HT$. Given a coalgebra $\gamma: X \to HTX$, where $H$ and $T$ satisfy the above conditions, the coalgebra $\gamma^\sharp: TX \to HTX$ obtained by applying the generalized powerset construction to $\gamma$ 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 $\gamma$ 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 $T = D_\omega$ and $H = G$, since $G D_\omega(X)$ has a $D_\omega$-algebra structure, leading $(\delta \circ \alpha)^\sharp = \alpha$.

**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: $G_{id} = id$, $G_A = D_\omega(A)$, $G_{H_1 \times H_2} = G_{H_1} \times G_{H_2}$ and $G_{H_1 + H_2} = (G_{H_2})^{H_1}$. Note that in our particular example $F = 1 + A \times -$ and $G_F = G = [0, 1] \times (-)^A$, where $[0, 1] = D_\omega(1)$. Such a functor $G_H$ may be useful to determinize $D_\omega H$-coalgebras, and a corresponding natural transformation $\delta_H: D_\omega H \Rightarrow G_H D_\omega$ 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 $\text{out}$ and the trace semantics given by $\text{tr}$. 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 $\text{out}$, as depicted in the commuting diagram below, is in $\mathcal{D}(A^*)$.

\[
\begin{array}{ccc}
X \xrightarrow{\eta_X} \mathcal{D}_\omega(X) & \xrightarrow{\text{out}} & D(A^*) \\
\downarrow{\alpha} & & \downarrow{\delta} \\
\mathcal{D}_\omega(1 + A \times X) & \xrightarrow{\text{id} \times \text{out}^A} & [0, 1] \times \mathcal{D}(A^*)^A \\
\downarrow{\delta} & & \downarrow{\delta} \\
[0, 1] \times \mathcal{D}_\omega(X)^A & \xrightarrow{\text{out}} & [0, 1] \times ([0, 1] \times \mathcal{D}(A^*)^A)^A 
\end{array}
\]

**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 $\text{out}$ 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 $\text{out}_\equiv$ and its connection to coalgebraic traces.

Our goal in the remainder of this section is to define $\text{out}_\equiv$ and show that $\text{out}_\equiv \circ [-] = \text{tr}$, which is also expected from the definition of $\text{out}$ 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 $\text{out}_\equiv$ and its connection to coalgebraic traces.

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

\[
\begin{array}{ccc}
\mathcal{D}_\omega(1 + A \times \text{Exp}) & \xrightarrow{\mathcal{D}_\omega(1 + A \times [-]_\sim)} & \mathcal{D}_\omega(1 + A \times \text{Exp}/\sim) \\
\text{co} & & \text{co} \\
\text{Exp} & \xrightarrow{[-]_\sim} & \text{Exp}/\sim \\
\end{array}
\]

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, $\text{Exp}/\equiv$ 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 $\mathcal{D}_\omega$ [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) $\bigoplus_{i \in I} p_{i,k} x_i = x_k$ if $p_{i,k} = 1$ for $i = k$ and $p_{i,k} = 0$ otherwise

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

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

**Proposition 5.5 (Exp/_\sim is a PCA)** The set $\text{Exp}_\sim$ has a positive convex algebra structure, that is, for every $[E_1]_\sim, \ldots, [E_n]_\sim \in \text{Exp}_\sim$ and $p_1, \ldots, p_n \in [0,1]$ 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 $\text{Exp}_\neq$ also has a positive convex structure.

**Proposition 5.6 (Exp/_\neq is a PCA)** The set $\text{Exp}_\neq$ has a positive convex algebra structure, that is, for every $[E_1], \ldots, [E_n] \in \text{Exp}_\neq$ and $p_1, \ldots, p_n \in [0,1]$ 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 $[-]_\neq$ is an algebra homomorphism from $\text{Exp}_\sim$ to $\text{Exp}_\neq$.

Let $a_\sim$ denote the algebra map on $\text{Exp}_\sim$, $a_\sim: \mathcal{D}_\omega(\text{Exp}_\sim) \to \text{Exp}_\sim$, given by the positive convex structure and $a_\neq$ the algebra map on $\text{Exp}_\neq$, $a_\neq: \mathcal{D}_\omega(\text{Exp}_\neq) \to \text{Exp}_\neq$, making $[-]_\neq$ 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:

\[
\begin{array}{c}
\text{Exp} \\
\downarrow c
\end{array} \xrightarrow{\sim} \begin{array}{c}
\text{Exp}/\sim \\
\downarrow c_0
\end{array} \xrightarrow{\sim} \begin{array}{c}
\text{Exp}/= \\
\downarrow \delta
\end{array} \xrightarrow{\sim} \begin{array}{c}
\mathcal{D}_\omega(1 + A \times \text{Exp}) \\
\downarrow \delta
\end{array} \xrightarrow{\sim} \begin{array}{c}
\mathcal{D}_\omega(1 + A \times \text{Exp}/\sim) \\
\downarrow [0, 1] \times \mathcal{D}_\omega(\text{Exp}/\sim)^A
\end{array} \xrightarrow{\sim} [0, 1] \times \text{Exp}/=.
\]

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

\[
G[-]= \circ \text{Ga}_- \circ \delta \circ c_0 ([E_1]_-) = G[-]= \circ \text{Ga}_- \circ \delta \circ c_0 ([E_2]_-).
\]

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

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

Let \(\text{out}_=\) be the unique homomorphism from \((\text{Exp}/=, d)\) 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 \((\text{Exp}/=, d)\) and the determinization of \((\text{Exp}/\sim, c_0)\) and shed some light on the overall situation. First, we note that

\[
c_0 \circ a_- = \mu \circ \mathcal{D}_\omega c_0 \tag{2}
\]

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 \((\text{Exp}/\sim, a_-)\) 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

\[
\overline{c_0} = \delta \circ c_0 \circ a_- \tag{3}
\]

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

\[
\text{Ga}_- \circ \overline{c_0} = \text{Ga}_- \circ \delta \circ c_0 \circ a_- \tag{4}
\]

Let \(\text{out}_\sim\) be the unique homomorphism from the determinization \((\mathcal{D}_\omega \text{Exp}/\sim, \overline{c_0})\) to the final \(G\)-coalgebra and \(\text{out}_\sim\) the unique homomorphism from \((\text{Exp}/\sim, \text{Ga}_- \circ \delta \circ c_0)\) to the final \(G\)-coalgebra.
Lemma 5.8 The final coalgebra homomorphisms satisfy
\[ \text{out} = \text{out}_\sim \circ \text{a}_\sim, \quad \text{out}_\sim = \text{out} \circ \eta, \quad \text{out}_\sim = \text{out}_\equiv \circ [\cdot]_\equiv. \]

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

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

\[
\begin{array}{c}
\text{Exp}_\equiv \\
d \downarrow \quad \downarrow \delta \\
[0, 1] \times \text{Exp}_\equiv & \rightarrow & [0, 1] \times (\mathcal{D}(A^*))^A & \rightarrow & [0, 1] \times ([0, 1]A^*)^A
\end{array}
\]

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

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

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

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

5.3 Completeness

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

\[
\text{out}_\equiv = \text{Exp}_\equiv \xrightarrow{e} I \xrightarrow{m} \mathcal{D}_\omega(A^*)
\]

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 \( \text{out}_\equiv \) is mono.

The difficulty is that \( (\text{Exp}_\equiv, d) \) is not final in the category of \( G \)-coalgebras on \text{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 \text{Sets}, we take \text{PCA}, the category of Eilenberg-Moore algebras of \( \mathcal{D}_\omega \).
Then we consider coalgebras on \( \mathbf{PCA} \). For a functor \( F \) on \( \mathbf{PCA} \), an \( F \)-coalgebra is a pair \( ((X, a), \alpha) \) where \( \alpha \) is an algebra homomorphism from \( (X, a) \) to \( F(X, a) \), both in \( \mathbf{PCA} \). An \( F \)-coalgebra homomorphism from \( ((X, a), \alpha) \) to \( ((Y, b), \beta) \) is a map \( h : X \to Y \) that is both an algebra and a coalgebra homomorphism, i.e., \( b \circ D_\omega(h) = h \circ a \) and \( \beta \circ h = F(h) \circ \alpha \).

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

\[
\bigoplus_i p_i(o_i, f_i) = \left( \sum_i p_i \cdot o_i, f \right), \quad \text{with } f(a) = \bigoplus_i p_i f_i(a)
\]

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

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

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

The carrier of the final \( G \)-coalgebra on \( \mathbf{Sets} \) has a PCA structure \( \bar{3} \), making it final in the category of \( G \)-coalgebras on \( \mathbf{PCA} \). 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 \( \bar{3} \) is given by

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

Therefore, for any \( G \)-coalgebra \( (X, \alpha) \) on \( \mathbf{Sets} \) such that \( X \) has a PCA structure \( a \), the final coalgebra map \( \text{out} \) is also the final coalgebra map from the \( G \)-coalgebra \( ((X, a), \alpha) \) on \( \mathbf{PCA} \). That is, \( \text{out} \) is also an algebra homomorphism from \( (X, a) \) to \( ([0, 1]^A \wedge, \bar{3}) \), as shown in the diagram below.

\[
\begin{array}{ccc}
\mathcal{D}_\omega X & \xrightarrow{i} & \mathcal{D}_\omega([0, 1]^A) \\
\downarrow \alpha & & \downarrow \bar{3} \\
X & \xrightarrow{\text{out} \wedge} & [0, 1]^A \wedge \\
\downarrow \iota & & \downarrow \langle ?, (-) \rangle \\
[0, 1] \times X^A & \xrightarrow{id \times \text{out} \wedge} & [0, 1] \times ([0, 1]^A) \wedge
\end{array}
\]

At this point it is important to mention that \( \mathcal{D}(A^\wedge) \) also has a PCA structure, namely \( \mu \circ i \). Moreover, the inclusion \( \mathcal{D}(A^\wedge) \xrightarrow{\iota} [0, 1]^A \wedge \) is an algebra homomorphism from \( (\mathcal{D}(A^\wedge), \mu \circ i) \) to \( ([0, 1]^A \wedge, \bar{3}) \). Also \( ((\mathcal{D}(A^\wedge), \mu \circ i), \delta \circ \mathcal{D}i) \) is a \( G \)-coalgebra on \( \mathbf{PCA} \). As a result, we get the following lemma which is applicable to any determinization (by Lemma 5.3), as well as to \( \text{out}_\sim \) and \( \text{out}_\approx \) (by Lemma 5.9),

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

\[
\begin{array}{rcl}
(\text{Exp}/=, a_\equiv) & \xrightarrow{e} & (I, a_I) \xrightarrow{m} (\mathcal{D}(A^*), \mu) \\
\downarrow d & & \downarrow \delta \circ \mathcal{D}_i \\
G(\text{Exp}/=, a_\equiv) & \xrightarrow{g} & G(I, a_I) \xrightarrow{\delta \circ \mathcal{D}_i} G(\mathcal{D}_\omega(A^*), \mu)
\end{array}
\]

Still, \(G\)-coalgebras on **PCA** are not sufficient for our goal of showing finality of \((\text{Exp}/=, a_\equiv), d\) 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 \((X, a)\) be a PCA. The functor \(\hat{G}\) is defined on object as

\[
\hat{G}(X, a) = \{ (o, f) \in [0, 1] \times X^A \mid \forall a \in A. \exists p_a^i \in [0, 1], v_i \in V. f(a) = \bigoplus_i p_a^i v_i^i \quad \text{and} \quad \sum_a \sum_i p_a^i \leq 1 - o \}
\]

and on arrows just like \(G\).

### 5.5 Finality for completeness

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

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

**Proposition 5.14** \((\text{Exp}/=, a_\equiv), d\) is final in **PCA\(_f\)(\(\hat{G}\))**.

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

**Lemma 5.15** The category **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 \(\text{out}\_\equiv : \text{Exp}/= \to \mathcal{D}(A^*)\) is injective.

This is the last ingredient we needed for completeness.

**Theorem 5.17** (Completeness) For all \(E_1, E_2 \in \text{Exp}\), \(E_1 \sim_{tr} E_2 \Rightarrow E_1 \equiv E_2\).
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 $D(A^*)$) and not just any arbitrary function $[0,1]^{T^*}$. 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


