Steps and traces

Rot, Jurriaan; Jacobs, Bart; Levy, Paul

DOI:
10.1093/logcom/exab050

Steps and traces*

JURRIAAN ROT, *Institute for Computing and Information Sciences, Radboud Universiteit, Nijmegen 6525 EC, The Netherlands.*
E-mail: jrot@cs.ru.nl

BART JACOBS, *Institute for Computing and Information Sciences, Radboud Universiteit, Nijmegen 6525 EC, The Netherlands.*

PAUL BLAIN LEVY, *School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK.*

Abstract

In the theory of coalgebras, trace semantics can be defined in various distinct ways, including through algebraic logics, the Kleisli category of a monad or its Eilenberg–Moore category. This paper elaborates two new unifying ideas: (i) coalgebraic,draftrules trace semantics is naturally presented in terms of corecursive algebras, and (ii) all three approaches arise as instances of the same abstract setting. Our perspective puts the different approaches under a common roof and allows to derive conditions under which some of them coincide.

Keywords: Coalgebra, trace semantics, corecursive algebras

1 Introduction

Traces are used in the semantics of state-based systems as a way of recording the consecutive behaviour of a state in terms of sequences of observable (input and/or output) actions. Trace semantics leads to, for instance, the notion of trace equivalence, which expresses that two states cannot be distinguished by only looking at their iterated in/output behaviour.

Trace semantics is a central topic of interest in the coalgebra community—and not only there, of course. One of the key features of the area of coalgebra is that states and their coalgebras can be considered in different universes, formalized as categories. The break-through insight is that trace semantics for a system in universe A can often be obtained by switching to a different universe B. More explicitly, where the (ordinary) behaviour of the system can be described via a final coalgebra in universe A, the trace behaviour arises by finality in the different universe B. Typically, the alternative universe B is a category of algebraic logics, the Kleisli category or the category of Eilenberg–Moore algebras, of a monad on universe A.

This paper elaborates two new unifying ideas.

1. We observe that the trace map from the state space of a coalgebra to a carrier of traces is in all three situations the unique 'coalgebra-to-algebra' map to a *corecursive algebra* [7] of traces.

[*] This is a revised and extended version of a paper which appeared in the proceedings of CMCS 2018 [21].

Vol. 31, No. 6,
Advance Access Publication on 23 August 2021 https://doi.org/10.1093/logcom/exab050

This differs from earlier work which tries to describe traces as final coalgebras. For us, it is quite natural to view languages as algebras, certainly when they consist of finite words/traces.

2. Next, these corecursive algebras, used as spaces of traces, all arise via a uniform construction, in a setting given by an adjunction together with a special natural transformation that we call a 'step'. We heavily rely on a basic result saying that in this situation, the (lifting of the) right adjoint preserves corecursive algebras, sending them from one universe to another. This is a known result [6], but its fundamental role in trace semantics has not been recognized before. For an arbitrary coalgebra, there is then a unique map to the transferred corecursive algebra; this is the trace map that we are after.

The main contribution of this paper is the unifying step-based approach to coalgebraic trace semantics: it is shown that three existing flavours of trace semantics—logical, Eilenberg–Moore and Kleisli—are all instances of our approach. Moreover, comparison results are given relating theses. We focus only on finite trace semantics and also exclude at this stage the 'iteration' based approaches, e.g. in [9, 10, 31, 38].

Outline. The paper is organized as follows. It starts in Section 1 with the abstract step-and-adjunction setting and the relevant definitions and results for corecursive algebras. In the next three sections, it is explained how this setting gives rise to trace semantics, by presenting the above-mentioned three approaches to coalgebraic trace semantics in terms of steps and adjunctions: Eilenberg–Moore (Section 3), logical (Section 4) and Kleisli (Section 5). In each case, the relevant corecursive algebra is described. These sections are illustrated with several examples. In Section 6, we study partial traces for coalgebras with input and output [5], as another instance of the step-and-adjunction setting, but it is helpful to express that setting in the language of bimodules, which we do in Appendix B.

The next section establishes a connection between the Eilenberg–Moore and the logical approach, between the Kleisli and logical approach and between the Kleisli and Eilenberg–Moore approach (Section 7). In Section 8, we show that our construction yields algebras that are not merely corecursive but completely iterative, a stronger property that provides more general trace semantics. Finally, in Section 9, we provide some directions for future work.

Notation. In the context of an adjunction $F \dashv G$, we shall use overline notation $\overline{(-)}$ for adjoint transposition. The unit and counit of an adjunction are, as usual, written as η and ε.

For an endofunctor H, we write $\mathrm{Alg}(H)$ for its algebra category and $\mathrm{CoAlg}(H)$ for its coalgebra category. For a monad (T, η, μ) on $\mathbf{C}$, we write $\mathcal{EM}(T)$ for the Eilenberg–Moore category and $\mathcal{K}\ell(T)$ for the Kleisli category.

We recall that any functor $S\colon \mathbf{Sets} \to \mathbf{Sets}$ has a unique strength st. We write $\mathrm{st}\colon S(X^A) \to S(X)^A$ for $\mathrm{st}(t)(a) = S(\mathrm{ev}_a)(t)$, where $\mathrm{ev}_a = \lambda f. f(a)\colon X^A \to X$.

2 Coalgebraic semantics from a step

This section is about the construction of corecursive algebras and their use for semantics. The notion of corecursive algebra, studied in [7, 11] as the dual of Taylor's notion of recursive coalgebra [12], is defined as follows.

Definition 2.1
Let H be an endofunctor on a category $\mathbf{C}$.

1. A *coalgebra-to-algebra morphism* from a coalgebra $c\colon X \to H(X)$ to an algebra $a\colon H(\Theta) \to \Theta$ is a map $f\colon X \to \Theta$ such that the diagram

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "c"'] & \Theta \\ H(X) \arrow[r, "H(f)"'] & H(\Theta) \arrow[u, "a"'] \end{tikzcd}$$

commutes. Equivalently: such a morphism is a fixpoint for the endofunction on the homset $\mathbf{C}(X,\Theta)$ sending f to the composite

$$X \xrightarrow{c} H(X) \xrightarrow{H(f)} H(\Theta) \xrightarrow{a} \Theta$$

2. An algebra $a\colon H(A) \to A$ is *corecursive* when for every coalgebra $c\colon X \to H(X)$ there is a unique coalgebra-to-algebra morphism $(X,c) \to (\Theta,a)$.

Here is some intuition.

- As explained in [18], the specification of a coalgebra-to-algebra morphism f is a 'divide-and-conquer' algorithm. It says: to operate on an argument, first decompose it via the coalgebra c, then operate on each component via $H(f)$, then combine the results via the algebra a.
- For each final H-coalgebra $\zeta\colon \Theta \overset{\cong}{\Rightarrow} H(\Theta)$, the inverse $\zeta^{-1}\colon H(\Theta) \to \Theta$ is a corecursive algebra. For most functors of interest, this final coalgebra gives semantics up to bisimilarity, which is finer than trace equivalence. So trace semantics requires a different corecursive algebra.

In all our examples, we use the same procedure for obtaining a corecursive algebra. It makes frequent use of the following so-called *mate correspondence* [26, 34]; also see, e.g. [23, 28, 29, 35] for special cases.

THEOREM 2.2
Given adjunctions and functors

$$\begin{tikzcd}[row sep=large] \mathbf{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] \arrow[d, "H"'] & \mathbf{D} \arrow[l, bend left, "G"] \arrow[d, "L"] \\ \mathbf{C}' \arrow[r, bend left, "F'"] \arrow[r, phantom, "\perp"] & \mathbf{D}' \arrow[l, bend left, "G'"] \end{tikzcd}$$

there are bijective correspondences between natural transformations:

$$\begin{tikzcd} \mathbf{C} \arrow[d, "H"'] & \mathbf{D} \arrow[l, "G"'] \arrow[d, "L"] \arrow[dl, phantom, "\overset{\rho_1}{\Longrightarrow}"] \\ \mathbf{C}' & \mathbf{D}' \arrow[l, "G'"] \end{tikzcd} \qquad \begin{tikzcd} \mathbf{C} \arrow[r, "F"] \arrow[d, "H"'] \arrow[dr, phantom, "\overset{\rho_2}{\Longrightarrow}"] & \mathbf{D} \arrow[d, "L"] \\ \mathbf{C}' \arrow[r, "F'"'] & \mathbf{D}' \end{tikzcd} \qquad \begin{tikzcd} \mathbf{C} \arrow[d, "H"'] & \mathbf{D} \arrow[l, "G"'] \arrow[d, "L"] \arrow[dl, phantom, "\overset{\rho_3}{\Longrightarrow}"] \\ \mathbf{C}' \arrow[r, "F'"'] & \mathbf{D}' \end{tikzcd} \qquad \begin{tikzcd} \mathbf{C} \arrow[r, "F"] \arrow[d, "H"'] \arrow[dr, phantom, "\overset{\rho_4}{\Longrightarrow}"] & \mathbf{D} \arrow[d, "L"] \\ \mathbf{C}' & \mathbf{D}' \arrow[l, "G'"] \end{tikzcd}$$

Here ρ_1 and ρ_3 correspond by adjoint transposition, and similarly for ρ_2 and ρ_4. Further, ρ_1 and ρ_2 are obtained from each other by

$$\begin{aligned} \rho_1 &= \left(HG \xRightarrow{\eta' HG} G'F'HG \xRightarrow{G'\rho_2 G} G'LFG \xRightarrow{G'L\varepsilon} G'L \right) \\ \rho_2 &= \left(F'H \xRightarrow{F'H\eta} F'HGF \xRightarrow{F'\rho_1 F} F'G'LF \xRightarrow{\varepsilon' LF} LF \right). \end{aligned}$$

It is common to refer to ρ_1 and ρ_2 as *mates*; the other two maps are their adjoint transposes, as we have seen. In diagrams, we omit the subscript i in ρ_i and let the type determine which version of ρ is meant. Further, in the remainder of this paper, we usually drop the subscript of components of natural transformations.

Our basic setting consists of an adjunction, two endofunctors and a natural transformation:

$$\begin{tikzcd} \mathbf{C} \arrow[loop left, "H"] \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{D} \arrow[l, bend left, "G"] \arrow[loop right, "L"] \end{tikzcd} \qquad \text{with} \qquad HG \overset{\rho}{\Longrightarrow} GL \tag{1}$$

The natural transformation $\rho\colon HG \Rightarrow GL$ will be called a *step*. Here H is the *behaviour functor*: we study H-coalgebras and give semantics for them in a corecursive H-algebra. This arrangement is well known in the area of coalgebraic modal logic [3, 8, 28, 35, 40], but we shall see that its application is wider. The following result shows different equivalent presentations of a step; for the proof, see Appendix A.

THEOREM 2.3
In the situation (1), there are bijective correspondences between natural transformations $\rho_1\colon HG \Rightarrow GL$, $\rho_2\colon FH \Rightarrow LF$, $\rho_3\colon FHG \Rightarrow L$ and $\rho_4\colon H \Rightarrow GLF$, as in Theorem 2.2.

Moreover, if H and L happen to be monads, then ρ_1 is an $\mathcal{EM}$-law (map $HG \Rightarrow GL$ compatible with the monad structures) iff ρ_2 is a $\mathcal{K}\ell$-law (map $FH \Rightarrow LF$ compatible with the monad structures) iff ρ_4 is a monad map; and two further equivalent characterizations are respectively a lifting of G or an extension of F:

$$\begin{tikzcd} \mathcal{EM}(H) \arrow[d] & \mathcal{EM}(L) \arrow[l, "\overline{G}"'] \arrow[d] \\ \mathbf{C} & \mathbf{D} \arrow[l, "G"] \end{tikzcd} \qquad\qquad \begin{tikzcd} \mathcal{K}\ell(H) \arrow[r, "\overline{F}"] & \mathcal{K}\ell(L) \\ \mathbf{C} \arrow[u] \arrow[r, "F"] & \mathbf{D} \arrow[u] \end{tikzcd} \qquad \square$$

Steps give rise to liftings to categories of algebras and coalgebras, as follows.

DEFINITION 2.4
In the setting (1), the step natural transformation ρ gives rise to both:

– a lifting G_ρ of the right adjoint G, called the *step-induced algebra lifting*:

$$\begin{tikzcd} \mathrm{Alg}(H) \arrow[d] & \mathrm{Alg}(L) \arrow[l, "G_\rho"'] \arrow[d] \\ \mathbf{C} & \mathbf{D} \arrow[l, "G"] \end{tikzcd} \qquad \begin{aligned} &G_\rho\Big(L(X) \overset{a}{\to} X\Big) := \\ &\quad \Big(HG(X) \overset{\rho}{\to} GL(X) \xrightarrow{G(a)} G(X)\Big). \end{aligned}$$

- dually, a lifting F^{ρ} of the left adjoint F, called the *step-induced coalgebra lifting*:

$$\begin{tikzcd}
\mathrm{CoAlg}(H) \arrow[r, "F^{\rho}"] \arrow[d] & \mathrm{CoAlg}(L) \arrow[d] \\
\mathbf{C} \arrow[r, "F"] & \mathbf{D}
\end{tikzcd}
\qquad
\begin{array}{l}
F^{\rho}\Big(X \xrightarrow{c} H(X)\Big) \;:= \\
\Big(F(X) \xrightarrow{F(c)} FH(X) \xrightarrow{\rho} LF(X)\Big).
\end{array}$$

Our approach relies on the following basic result.

PROPOSITION 2.5
([6]).
In the setting (1), for a corecursive L-algebra $a\colon L(\Theta) \to \Theta$, the transferred H-algebra $G_{\rho}(\Theta, a)\colon HG(\Theta) \to G(\Theta)$ is also corecursive.

PROOF. Let $c\colon X \to H(X)$ be an H-coalgebra. Then $F^{\rho}(X,c)$ is an L-coalgebra, which gives rise to a unique coalgebra-to-algebra map $f\colon F(X) \to \Theta$ with $a \circ L(f) \circ \rho \circ F(c) = f$. The adjoint-transpose $g\colon X \to G(\Theta)$ of f is then the unique coalgebra-to-algebra map from (X,c) to $G_{\rho}(\Theta, a)$. □

Thus, by analogy with the familiar statement that *'right adjoints preserves limits'*, we have *'step-induced algebra liftings of right adjoints preserve corecursiveness'*. Now we give the complete construction for semantics of a coalgebra.

THEOREM 2.6
Suppose that L has a final coalgebra $\zeta\colon \Psi \xrightarrow{\cong} L(\Psi)$. Then for every H-coalgebra (X,c) there is a unique coalgebra-to-algebra map $c^{\dagger}$ as on the left below:

$$\begin{tikzcd}
X \arrow[r, dashed, "c^{\dagger}"] \arrow[d, "c"'] & G(\Psi) \\
H(X) \arrow[r, dashed, "H(c^{\dagger})"] & HG(\Psi) \arrow[u, "G_{\rho}(\Psi,\zeta^{-1})"']
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
F(X) \arrow[r, dashed, "\overline{c^{\dagger}}"] \arrow[d, "F^{\rho}(X,c)"'] & \Psi \\
LF(X) \arrow[r, dashed, "L(\overline{c^{\dagger}})"] & L(\Psi) \arrow[u, "\zeta^{-1}"']
\end{tikzcd}$$

The map $c^{\dagger}$ on the left can alternatively be characterized via its adjoint transpose $\overline{c^{\dagger}}$ on the right, which is the unique coalgebra-to-algebra morphism. The latter can also be seen as the unique map to the final coalgebra $\Psi \xrightarrow{\cong} L(\Psi)$..

Note that Theorem 2.6 generalizes final coalgebra semantics: taking in (1) $F = G = \mathrm{Id}_{\mathbf{C}}$ and $H = L$, the map $c^{\dagger}$ in the above theorem is the unique homomorphism to the final coalgebra. In the remainder of this paper, we focus on instances where $c^{\dagger}$ captures traces, and we therefore refer to $c^{\dagger}$ as the *trace semantics* map.

Given steps $\rho\colon HG \Rightarrow GL$ and $\theta\colon KG \Rightarrow GM$, we can form a new step by composition, written as $\theta \circledcirc \rho$ in:

$$\theta \circledcirc \rho \quad := \quad \Big(KHG \overset{K\rho}{\Longrightarrow} KGL \overset{\theta L}{\Longrightarrow} GML\Big) \tag{2}$$

We conclude with a lemma that relates the mate construction to composition of steps. See Appendix A for a proof.

LEMMA 2.7
Let $\rho\colon HG \Rightarrow GL, \theta\colon KG \Rightarrow GM$ be steps. Then $\left(\theta \circledcirc \rho\right)_2 = M\rho_2 \circ \theta_2 H$.

3 Traces via Eilenberg–Moore

We recall the approach to trace semantics developed in [4, 22, 43], putting it in the framework of the previous section. The approach deals with coalgebras for the composite functor BT, where T is a monad that captures the 'branching' aspect. The following assumptions are required.

ASSUMPTION 3.1
(Traces via Eilenberg–Moore).
In this section, we assume the following:

1. An endofunctor $B\colon \mathbf{C} \to \mathbf{C}$ with a final coalgebra $\zeta\colon \Theta \overset{\cong}{\Rightarrow} B(\Theta)$.
2. A monad (T, η, μ), with the standard adjunction $\mathcal{F} \dashv U$ between categories $\mathbf{C} \leftrightarrows \mathcal{EM}(T)$, where U is 'forget' and $\mathcal{F}$ is for 'free algebras'.
3. A lifting $\overline{B}$ of B, as in

$$\begin{tikzcd}
\mathcal{EM}(T) \arrow[r, "\overline{B}"] \arrow[d, "U"'] & \mathcal{EM}(T) \arrow[d, "U"] \\
\mathbf{C} \arrow[r, "B"'] & \mathbf{C}
\end{tikzcd} \tag{3}$$

or, equivalently, an $\mathcal{EM}$-law $\kappa\colon TB \Rightarrow BT$.

EXAMPLE 3.2
To briefly illustrate these ingredients, we consider non-deterministic automata. These are BT-coalgebras with $B\colon \mathbf{Sets} \to \mathbf{Sets}$, $B(X) = 2 \times X^A$ where $2 = \{\bot, \top\}$ and T the finite powerset monad. The functor B has a final coalgebra carried by the set 2^{A^*} of languages. Further, $\mathcal{EM}(T)$ is the category of join semilattices (JSLs). The lifting is defined by products in $\mathcal{EM}(T)$, using the JSL on 2 given by the usual ordering $\bot \le \top$. By the end of this section, we revisit this example and obtain the usual language semantics.

The above three assumptions give rise to the following instance of our general setting (1):

$$\begin{tikzcd}
\mathbf{C} \arrow[loop left, "BT"] \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathcal{EM}(T) \arrow[l, bend left, "U"] \arrow[loop right, "\overline{B}"]
\end{tikzcd} \quad \text{with} \quad \begin{array}{l} \rho\colon BTU \Longrightarrow U\overline{B} \ \text{ where} \\ \rho_{(X,a)} = \left(BTX \xrightarrow{Ba} BX\right) \end{array} \tag{4}$$

Actually—and equivalently, by Theorem 2.3—the step ρ is most easily given in terms of $\rho_4\colon BT \Rightarrow U\overline{B}\mathcal{F}$: since $\overline{B}$ lifts B, we have $U\overline{B}\mathcal{F} = BU\mathcal{F} = BT$, so that ρ_4 is then defined simply as the identity.

The following result is well known and is (in a small variation) due to [45].

LEMMA 3.3
There is a unique algebra structure $a\colon T(\Theta) \to \Theta$ making $((\Theta, a), \zeta)$ a $\overline{B}$-coalgebra. Moreover, this coalgebra is final in $\mathrm{CoAlg}(\overline{B})$.

PROOF. We recall that the map $a\colon T(\Theta) \to \Theta$ is obtained by finality in

$$\begin{tikzcd}
T(\Theta) \arrow[r, dashed, "a"] \arrow[d, "\kappa \circ T(\zeta)"'] & \Theta \arrow[d, "\zeta", "\cong"'] \\
BT(\Theta) \arrow[r, dashed, "B(a)"'] & B(\Theta)
\end{tikzcd} \tag{5}$$

This gives an Eilenberg–Moore algebra (Θ, a), with a $\overline{B}$-coalgebra $\zeta : (\Theta, a) \to \overline{B}(\Theta, a)$ which is final. □

We apply the step-induced algebra lifting $G_\rho\colon \mathrm{Alg}(\overline{B}) \to \mathrm{Alg}(BT)$ to the inverse of this final $\overline{B}$-coalgebra, obtaining a BT-algebra:

$$\big(BT(\Theta) \xrightarrow{\ell_{\mathrm{em}}} \Theta\big) \quad := \quad G_\rho((\Theta, a), \zeta^{-1}) \quad = \quad \big(BT(\Theta) \xrightarrow{B(a)} B(\Theta) \xrightarrow{\zeta^{-1}} \Theta\big). \tag{6}$$

By Theorem 2.6, this BT-algebra ℓ_{em} is corecursive, giving us trace semantics of BT-coalgebras. More explicitly, given a coalgebra $c\colon X \to BT(X)$, the trace semantics is the unique map, written as em_c, making the following square commute.

$$\begin{tikzcd}
X \arrow[r, dashed, "\mathsf{em}_c"] \arrow[d, "c"'] & \Theta \\
BT(X) \arrow[r, dashed, "BT(\mathsf{em}_c)"'] & BT(\Theta) \arrow[u, "\ell_{\mathrm{em}}"']
\end{tikzcd} \tag{7}$$

The unique map em_c in (7) appears in the literature as a 'coiteration up-to' or 'unique solution' theorem [1]. Examples follow later in this section (Theorem 3.4, Example 3.5).

In [22, 43], the above trace semantics of BT-coalgebras arises through 'determinization', which we explain next. Given a coalgebra $c\colon X \to BT(X)$, one takes its adjoint transpose:

$$\frac{c\colon X \longrightarrow BT(X) = BU\mathcal{F}(X) = U\overline{B}\mathcal{F}(X)}{\overline{c}\colon \mathcal{F}(X) \longrightarrow \overline{B}\mathcal{F}(X)}$$

It follows from Theorem 2.3 and our definition of ρ that this transpose coincides with the application of the step-induced coalgebra lifting $\mathcal{F}^\rho\colon \mathrm{CoAlg}(BT) \to \mathrm{CoAlg}(\overline{B})$ from the previous section, i.e. $\mathcal{F}^\rho(X, c) = (\mathcal{F}(X), \overline{c})$. The functor $\mathcal{F}^\rho$ thus plays the role of determinization, see [22]. By Theorem 2.6, the trace semantics em_c can equivalently be characterized in terms of $\mathcal{F}^\rho$, as the unique map $\overline{\mathsf{em}_c}$ making the diagram below commute.

$$\begin{tikzcd}
T(X) \arrow[r, dashed, "\overline{\mathsf{em}_c}"] \arrow[d, "\overline{c}"'] & \Theta \\
BT(X) \arrow[r, dashed, "B(\overline{\mathsf{em}_c})"'] & B(\Theta) \arrow[u, "\zeta^{-1}"']
\end{tikzcd} \tag{8}$$

This is how the trace semantics via Eilenberg–Moore is presented in [22, 43]: as the transpose $\mathsf{em}_c = \overline{\mathsf{em}_c} \circ \eta_X : X \to \Theta$.

We conclude this section by recalling a canonical construction of a distributive law [19] for a class of 'automata-like' examples that we will spell out in Example 3.5.

THEOREM 3.4
Let Ω be a set, T a monad on **Sets** and $t\colon T(\Omega) \to \Omega$ an $\mathcal{EM}$-algebra. Let $B\colon \mathbf{Sets} \to \mathbf{Sets}$, $B(X) = \Omega \times X^A$, and $\kappa : TB \Rightarrow BT$ given by

$$\kappa_X := \Big(\; T(\Omega \times X^A) \xrightarrow{\langle T(\pi_1),T(\pi_2)\rangle} T(\Omega) \times T(X^A) \xrightarrow{t\times \mathsf{st}} \Omega \times T(X)^A \; \Big).$$

Then κ is an $\mathcal{EM}$-law. Moreover, the algebra structure on the carrier of the final coalgebra (Ω^{A^*}, ζ) mentioned in the statement of Lemma 3.3 is given by $T(\Omega^{A*})\xrightarrow{st}T(\Omega)^{A*}\xrightarrow{t^{A*}}\Omega^{A*}$. Hence, this algebra is the carrier of a final $\overline{B}$-coalgebra.

EXAMPLE 3.5
By Theorem 3.4, we obtain an explicit description of the trace semantics arising from the corecursive algebra (7): for any $\langle o,f\rangle : X \to \Omega \times T(X)^A$, the trace semantics is the unique map em in

$$\begin{tikzcd}
X \arrow[rrr, dashed, "\mathsf{em}"] \arrow[d, "\langle o,f\rangle"'] & & & \Omega^{A^*} \\
BT(X) \arrow[r, dashed, "BT(\mathsf{em})"'] & BT(\Omega^{A^*}) \arrow[r, "B(\mathsf{st})"'] & B(T(\Omega)^{A^*}) \arrow[r, "B(t^{A^*})"'] & B(\Omega^{A^*}) \arrow[u, "\zeta^{-1}"']
\end{tikzcd}$$

We instantiate the trace semantics em for various choices of Ω, T and t. Given a coalgebra $\langle o,f\rangle\colon X \to \Omega \times T(X)^A$, we have $\mathsf{em}(x)(\varepsilon) = o(x)$ independently of these choices. The table below lists the inductive case $\mathsf{em}(x)(aw)$, respectively, for non-deterministic automata (NDA) where branching is interpreted as usual (NDA-∃), NDA where branching is interpreted conjunctively (NDA-∀) and (reactive) probabilistic automata (PA). Here $\mathcal{P}_{\mathrm{f}}$ is the finite powerset monad, and $\mathcal{D}_{\mathrm{fin}}$ the finitely supported distribution (or subdistribution) monad.

	T	Ω	$t\colon T(\Omega) \to \Omega$	$\mathsf{em}(x)(aw)$
NDA-∃	$\mathcal{P}_{\mathrm{f}}$	$2 = \{\bot, \top\}$	$S \mapsto \bigvee S$	$\bigvee_{y \in f(x)(a)} \mathsf{em}(y)(w)$
NDA-∀	$\mathcal{P}_{\mathrm{f}}$	$2 = \{\bot, \top\}$	$S \mapsto \bigwedge S$	$\bigwedge_{y \in f(x)(a)} \mathsf{em}(y)(w)$
PA	$\mathcal{D}_{\mathrm{fin}}$	$[0, 1]$	$\varphi \mapsto \sum_{p\in[0,1]} p \cdot \varphi(p)$	$\sum_{y \in X} \mathsf{em}(y)(w) \cdot f(x)(a)(y)$

For other examples, and a concrete presentation of the associated determinization constructions, see [22, 43].

3.1 *Eilenberg–Moore trace semantics for TA-coalgebras*

We now extend the above treatment of trace semantics of BT-coalgebras via Eilenberg–Moore categories, to cover coalgebras for a composite functor TA as well, where A is another endofunctor on the base category **C**. This integrates the *extension semantics* of [22] in the present setting; the latter covers examples such as non-deterministic automata (as in Example 3.9) and probabilistic systems in generative form. The approach to trace semantics of TA-coalgebras in this section extends Assumption 3.1, making use of a lifting $\overline{B}$ of a functor B to obtain traces as a suitable final coalgebra. Note that A itself is not lifted but is connected to B via a step ρ as stipulated in the global assumptions of this subsection, described next.

Assumption 3.6

In addition to Assumption 3.1, we assume a functor $A\colon \mathbf{C} \to \mathbf{C}$, and a step ρ in

$$\begin{tikzcd} \mathbf{C} \arrow[loop left, "A"] \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathcal{EM}(T) \arrow[l, bend left, "U"] \arrow[loop right, "\overline{B}"] \end{tikzcd} \qquad \text{with} \qquad \rho\colon AU \Longrightarrow U\overline{B}$$

The counit $\varepsilon\colon \mathcal{F}U \to U$ is given by $\varepsilon_{(X,a)} = a$; notice that $\mathcal{F}U(X,a) = (T(X), \mu_X)$. Applying the forgetful functor to ε gives another step $U\varepsilon\colon TU \Rightarrow U$, where the '$L$' (from (1)) in the codomain is the identity functor. We can compose the steps $U\varepsilon$ and ρ in two ways. First, we get a step for AT by composing as follows:

$$\rho \circledcirc U\varepsilon \quad = \quad \left(ATU \overset{AU\varepsilon}{\Longrightarrow} AU \overset{\rho}{\Longrightarrow} U\overline{B} \right)$$

If $A = B$, then taking ρ to be the identity is precisely the step defined in (4).

We turn to the other composition of $U\varepsilon$ and ρ, which gives a step for TA:

$$U\varepsilon \circledcirc \rho \quad = \quad \left(TAU \overset{T\rho}{\Longrightarrow} TU\overline{B} \overset{U\varepsilon\overline{B}}{\Longrightarrow} U\overline{B} \right)$$

As we will see in Proposition 3.7, steps ρ as in Assumption 3.6 correspond to natural transformations of the form $\mathfrak{e} : TA \Rightarrow BT$ making the following diagram commute:

$$\begin{tikzcd} T^2A \arrow[rr, "\mu"] \arrow[d, "T(\mathfrak{e})"'] & & TA \arrow[d, "\mathfrak{e}"] \\ TBT \arrow[r, "\kappa"] & BT^2 \arrow[r, "B(\mu)"] & BT \end{tikzcd} \tag{9}$$

In [22], a natural transformation $\mathfrak{e}$ making this diagram commute is called an *extension law* if it additionally satisfies a coherence axiom with a $\mathcal{K}\ell$-law. We will only see this later, in our comparison between different approaches for assigning trace semantics, see Section 7.3. The last line of the correspondence below involves natural transformations of the form $A \Rightarrow BT$, which are called *generic observers* in [13].

Proposition 3.7

There is a one-to-one correspondence between

$$\begin{array}{c} \text{steps} \;\; \rho\colon AU \Longrightarrow U\overline{B} \\ \hline\hline \mathfrak{e}\colon TA \Rightarrow BT \;\; \text{satisfying (9)} \\ \hline\hline A \Longrightarrow BT \end{array}$$

Proof. By Theorem 2.3, the natural transformation $\rho = \rho_1 : AU \Rightarrow U\overline{B}$ corresponds to $\mathfrak{e} = \rho_2 : \mathcal{F}A \Rightarrow \overline{B}\mathcal{F}$; the latter is a natural transformation $TA \Rightarrow BT$ whose components are maps of algebras $\mu_X \to \overline{B}(\mu_x)$, as expressed by Diagram (9). This covers the first correspondence in the proposition.

By Theorem 2.3, a natural transformation $\mathfrak{e} : TA \Rightarrow BT$ further corresponds to a natural transformation $A \Rightarrow U\overline{B}\mathcal{F} = BU\mathcal{F} = BT$. The latter is simply a natural transformation on the base category $\mathbf{C}$, which means no further coherence axioms like (9) need to be checked. □

The composed step $U\varepsilon \circledcirc \rho\colon TAU \Rightarrow U\overline{B}$ gives a corecursive algebra, by applying the step-induced algebra lifting $G_{U\varepsilon\circledcirc\rho}\colon \mathrm{Alg}(\overline{B}) \to \mathrm{Alg}(TA)$ to the final $\overline{B}$-coalgebra $((\Theta, a), \zeta)$, from Lemma 3.3.

We call this corecursive algebra $\ell^A_{\mathrm{em}}\colon TA(\Theta) \to \Theta$ to distinguish it from $\ell_{\mathrm{em}}\colon BT(\Theta) \to \Theta$. It is given by

$$\begin{tikzcd}
\ell^A_{\mathrm{em}} \coloneqq \Big(TA(\Theta) = TAU(\Theta,a) \arrow[r, "T(\rho)"] & TU\overline{B}(\Theta,a) \arrow[rr, "U(\varepsilon)"] \arrow[d, equal] & & U\overline{B}(\Theta,a) \arrow[d, equal] & \\
 & TB(\Theta) \arrow[r, "\kappa"'] & BT(\Theta) \arrow[r, "B(a)"'] & B(\Theta) \arrow[r, "\zeta^{-1}"'] & \Theta \Big)
\end{tikzcd}$$

This corecursive algebra gives semantics to TA-coalgebras. It can be expressed in terms of the corecursive BT-algebra ℓ_{em}, making use of $\rho_2\colon \mathcal{F}A \Rightarrow \overline{B}\mathcal{F}$, as follows.

LEMMA 3.8
We have $\ell^A_{\mathrm{em}} = \ell_{\mathrm{em}} \circ U(\rho_2)\colon TA(\Theta) \to \Theta$. Explicitly,

$$\big(TA(\Theta) \xrightarrow{\ell^A_{\mathrm{em}}} \Theta\big) \quad = \quad \big(TA(\Theta) \xrightarrow{U(\rho_2)} U\overline{B}\mathcal{F}(\Theta) = BT(\Theta) \xrightarrow{B(a)} B(\Theta) \xrightarrow{\zeta^{-1}} \Theta\big).$$

PROOF. We describe $\ell_{\mathrm{em}} \circ U(\rho_2)$ as south-east and ℓ^A_{em} as east-south in

$$\begin{tikzcd}
TA(\Theta) = U\mathcal{F}AU(\Theta,a) \arrow[r, "T(\rho_1)"] \arrow[dd, "U(\rho_2)"'] & U\mathcal{F}U\overline{B}(\Theta,a) \arrow[r, phantom, "="] \arrow[dd, "U(\varepsilon)"'] & TB(\Theta) \arrow[d, "\kappa"] & \\
 & & BT(\Theta) \arrow[dd, "B(a)"] & \\
U\overline{B}\mathcal{F}U(\Theta,a) \arrow[r, "U\overline{B}(\varepsilon)"] \arrow[d, equal] & U\overline{B}(\Theta,a) \arrow[dr, equal] & & \\
BT(\Theta) \arrow[rr, "B(a)"'] & & B(\Theta) \arrow[r, "\zeta^{-1}"'] & \Theta
\end{tikzcd}$$

The upper left square commutes by Lemma A.1. □

EXAMPLE 3.9
We illustrate the situation with a simple example: non-deterministic automata, viewed as coalgebras of the form $c\colon X \to \mathcal{P}_{\mathrm{f}}(\Sigma \times X + 1)$. To this end, we instantiate the setting with $\mathbf{C} = \mathbf{Sets}$, $T = \mathcal{P}_{\mathrm{f}}$ the finite powerset monad and $A(X) = \Sigma \times X + 1$. Moreover, we let $B(X) = 2 \times X^{\Sigma}$. Note that there is a difference between $\mathcal{P}_{\mathrm{f}}A$-coalgebras and $B\mathcal{P}_{\mathrm{f}}$-coalgebras, if Σ is infinite: the former are finitely branching non-deterministic automata (that is, finitely many successors) whereas the latter are image-finite non-deterministic automata (that is, finitely many successors for every alphabet letter).

The lifting $\overline{B}\colon \mathcal{EM}(\mathcal{P}_{\mathrm{f}}) \to \mathcal{EM}(\mathcal{P}_{\mathrm{f}})$ of B is given as in Example 3.2 and Theorem 3.4. In particular, the corecursive algebra

$$\ell_{\mathrm{em}}\colon 2 \times (\mathcal{P}_{\mathrm{f}}(2^{\Sigma^*}))^{\Sigma} \to 2^{\Sigma^*}$$

is given by $\ell_{\mathrm{em}}(o,\varphi)(\varepsilon) = o$ and $\ell_{\mathrm{em}}(o,\varphi)(aw) = \bigvee_{\psi \in \varphi(a)} \psi(w)$.

The relevant step $\rho\colon AU \Rightarrow U\overline{B}$ is most easily given by $\rho_4\colon A \Rightarrow U\overline{B}\mathcal{F} = B\mathcal{P}_{\mathrm{f}}$. On a component X, we define $(\rho_4)_X\colon \Sigma \times X + 1 \to 2 \times (\mathcal{P}_{\mathrm{f}}X)^{\Sigma}$ by

$$(\rho_4)_X(a,x) = \left(\bot, \lambda b. \begin{cases} \{x\} & \text{if } a = b \\ \emptyset & \text{otherwise} \end{cases}\right), \qquad (\rho_4)_X(*) = (\top, \lambda b.\emptyset)\,.$$

Then $(\rho_2)_X\colon \mathcal{P}_{\mathrm{f}}(\Sigma \times X + 1) \to 2 \times (\mathcal{P}_{\mathrm{f}}X)^{\Sigma}$ is the adjoint transpose, given by

$$\rho_2(S) = \left(\bigvee_{*\in S} \top, \lambda a.\{x \mid (a,x) \in S\}\right).$$

This coincides with the extension law given in [22].

By Lemma 3.8, the corecursive $\mathcal{P}_{\mathrm{f}}A$-algebra obtained from the final $\overline{B}$-coalgebra is given by $\ell^A_{\mathrm{em}} = \ell_{\mathrm{em}} \circ U(\rho_2)\colon \mathcal{P}_{\mathrm{f}}(\Sigma \times 2^{\Sigma^*} + 1) \to 2^{\Sigma^*}$, which is

$$\ell^A_{\mathrm{em}}(S)(\varepsilon) = \bigvee_{*\in S} \top\,, \qquad \ell^A_{\mathrm{em}}(S)(aw) = \bigvee_{(a,\psi)\in S} \psi(w)\,.$$

Given a coalgebra $c\colon X \to \mathcal{P}_{\mathrm{f}}(\Sigma \times X + 1)$, the unique coalgebra-to-algebra morphism $\mathsf{em}^A\colon X \to 2^{\Sigma^*}$ from c to ℓ^A_{em} is thus given by $\mathsf{em}^A(x)(\varepsilon) = \bigvee_{*\in c(x)} \top$ and $\mathsf{em}^A(x)(aw) = \bigvee_{(a,y)\in c(x)} \mathsf{em}^A(y)(w)$.

For examples of extension laws for weighted and probabilistic automata, see [22].

4 Traces via logic

This section illustrates how the 'logical' approach to trace semantics of [29], ultimately based on the testing framework introduced in [40], fits in our general framework. In this approach, traces are viewed as logical formulas, also called tests, which are evaluated for states. These tests are obtained via an initial algebra of a functor L. The approach works both for *TB* and *BT*-coalgebras (and could, in principle, be extended to more general combinations). We start by listing our assumptions in this section and continue by showing how these assumptions lead to a corecursive algebra giving trace semantics in the general framework of Section 2.

Assumption 4.1
(Traces via logic).
In this section, we assume

1. An adjunction $F \dashv G$ between categories $\mathbf{C} \rightleftarrows \mathbf{D}^{\mathrm{op}}$.
2. A functor T on $\mathbf{C}$ with a step $\tau\colon TG \Rightarrow G$.
3. A functor $B\colon \mathbf{C} \to \mathbf{C}$ and a functor $L\colon \mathbf{D} \to \mathbf{D}$ with a step $\delta\colon BG \Rightarrow GL$.
4. An initial algebra $\alpha\colon L(\Phi) \overset{\cong}{\Rightarrow} \Phi.$.

We deviate from the convention of writing ρ for 'step', since the above map τ gives rise to multiple steps $\tau \circledcirc \delta$ and $\delta \circledcirc \tau$ in (11) below, in the sense of Definition 2.3; here we use 'delta' instead of 'rho' notation since it is common in modal logic.

Example 4.2
We take $\mathbf{C} = \mathbf{D} = \mathbf{Sets}$, and F, G both the contravariant powerset functor 2^{-}. Non-deterministic automata are obtained either as BT-coalgebras with $B(X) = 2 \times X^A$ and T the finite powerset functor, or as TB-coalgebras, with $B(X) = A \times X + 1$ and T again the finite powerset functor. In both

cases, L is given by $L(X) = A \times X + 1$, which has the set of words A^* as carrier of an initial algebra. The map $\tau\colon T2^- \Rightarrow 2^-$ is defined by $\tau_X(S)(x) = \bigvee_{\varphi \in S} \varphi(x)$ and intuitively models the existential choice in the semantics of non-deterministic automata. The step δ and the language semantics are defined later in this section.

The assumptions are close to the general step-and-adjunction setting (1). Here, we have an opposite category on the right and instantiate H to TB or BT:

$$\begin{tikzcd}
\mathbf{C} \arrow[loop left, "H"] \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{D}^{\mathrm{op}} \arrow[l, bend left, "G"] \arrow[loop right, "L"]
\end{tikzcd} \qquad \text{where } H = TB \text{ or } H = BT\,. \tag{10}$$

Notice that our assumptions already include a step δ (involving B, L) and a step τ, which we can compose to obtain steps for the TB respectively BT case:

$$\begin{aligned}
\tau \circledcirc \delta \quad &:= \quad \Big(TBG \xRightarrow{T\delta} TGL \xRightarrow{\tau L} GL\Big) \qquad\qquad \mathrm{CoAlg}(L)^{\mathrm{op}} \xrightarrow{G_{\tau\circledcirc\delta}} \mathrm{Alg}(TB) \\
\delta \circledcirc \tau \quad &:= \quad \Big(BTG \xRightarrow{B\tau} BG \xRightarrow{\delta} GL\Big) \qquad\qquad \mathrm{CoAlg}(L)^{\mathrm{op}} \xrightarrow{G_{\delta\circledcirc\tau}} \mathrm{Alg}(BT)
\end{aligned} \tag{11}$$

Both $\tau \circledcirc \delta$ and $\delta \circledcirc \tau$ are steps and hence give rise to step-induced algebra liftings $G_{\tau\circledcirc\delta}$ and $G_{\delta\circledcirc\tau}$ of G (Section 2). By Theorem 2.6, we obtain two corecursive algebras by applying these liftings to the inverse of the initial algebra, i.e. the (inverse of the) final coalgebra in $\mathbf{D}^{\mathrm{op}}$:

$$\begin{aligned}
\ell_{\log} \quad &:= \quad G_{\tau\circledcirc\delta}(\Phi, \alpha^{-1}) \quad = \quad \Big(TBG(\Phi) \xrightarrow{\tau\circledcirc\delta} GL(\Phi) \xrightarrow[\cong]{G(\alpha^{-1})} G(\Phi)\Big)\ , \\
\ell^{\log} \quad &:= \quad G_{\delta\circledcirc\tau}(\Phi, \alpha^{-1}) \quad = \quad \Big(BTG(\Phi) \xrightarrow{\delta\circledcirc\tau} GL(\Phi) \xrightarrow[\cong]{G(\alpha^{-1})} G(\Phi)\Big).
\end{aligned} \tag{12}$$

These corecursive algebras define trace semantics for any TB-coalgebra (X, c) and BT-coalgebra (Y, d):

$$\begin{tikzcd}
X \arrow[r, dashed, "\mathsf{log}_c"] \arrow[d, "c"'] & G(\Phi) & & Y \arrow[r, dashed, "\mathsf{log}_d"] \arrow[d, "d"'] & G(\Phi) \\
TB(X) \arrow[r, dashed, "TB(\mathsf{log}_c)"] & TBG(\Phi) \arrow[u, "\ell_{\log}"'] & & BT(Y) \arrow[r, dashed, "BT(\mathsf{log}_d)"] & BTG(\Phi) \arrow[u, "\ell^{\log}"']
\end{tikzcd} \tag{13}$$

It is instructive to characterize this trace semantics in terms of the transpose and the step-induced coalgebra liftings $F^{\tau\circledcirc\delta}$ and $F^{\delta\circledcirc\tau}$, showing how they arise as unique maps from an initial algebra:

$$\begin{tikzcd}
F(X) & \Phi \arrow[l, dashed, "\overline{\mathsf{log}_c}"'] \arrow[d, "\alpha^{-1}"] & & F(Y) & \Phi \arrow[l, dashed, "\overline{\mathsf{log}_d}"'] \arrow[d, "\alpha^{-1}"] \\
LF(X) \arrow[u, "F^{\tau\circledcirc\delta}(X,c)"] & L(\Phi) \arrow[l, dashed, "L(\overline{\mathsf{log}_c})"'] & & LF(Y) \arrow[u, "F^{\delta\circledcirc\tau}(Y,d)"] & L(\Phi) \arrow[l, dashed, "L(\overline{\mathsf{log}_d})"']
\end{tikzcd} \tag{14}$$

In the remainder of this section, we show two classes of examples of the logical approach to trace semantics. With these descriptions, we retrieve most of the examples from [29] in a smooth manner.

PROPOSITION 4.3

Let Ω be a set, $T\colon \mathbf{Sets} \to \mathbf{Sets}$ a functor and $t\colon T(\Omega) \to \Omega$ a map. Then the set of languages Ω^{A^*} carries a corecursive algebra for the functor $\Omega \times T(-)^A$. Given a coalgebra $\langle o,f\rangle\colon X \to \Omega \times T(X)^A$, the unique coalgebra-to-algebra morphism $\mathsf{log}\colon X \to \Omega^{A^*}$ satisfies

$$\mathsf{log}(x)(\varepsilon) = o(x) \qquad \mathsf{log}(x)(aw) = t\Big(T(\mathrm{ev}_w \circ \mathsf{log})(f(x)(a))\Big)$$

for all $x \in X$, $a \in A$ and $w \in A^*$.

PROOF. We instantiate the assumptions in the beginning of this section by $\mathbf{C} = \mathbf{D} = \mathbf{Sets}$, $F = G = \Omega^-$, $B(X) = \Omega \times X^A$, $L(X) = A \times X + 1$ and T the functor from the statement. The initial L-algebra is $\alpha\colon A \times A^* + 1 \overset{\cong}{\Rightarrow} A^*$. The map t extends to a modality $\tau\colon TG \Rightarrow G$, given on components by

$$\tau_X := \Big(\begin{tikzcd} T(\Omega^X) \arrow[r, "\mathrm{st}"] & T(\Omega)^X \arrow[r, "t^X"] & \Omega^X \end{tikzcd}\Big).$$

The logic $\delta\colon BG \Rightarrow GL$ is given by the isomorphism $\Omega \times (\Omega^-)^A \cong \Omega^{(A\times -)+1}$. Instantiating (12), we obtain the corecursive BT-algebra

$$\begin{tikzcd} \Omega \times T(\Omega^{A^*})^A \arrow[r, "\mathrm{id}\times(\mathrm{st})^A"] & \Omega \times (T(\Omega)^{A^*})^A \arrow[r, "\mathrm{id}\times(t^{A^*})^A"] & \Omega \times (\Omega^{A^*})^A \arrow[r, "\Omega^{\alpha^{-1}}\circ\delta"] & \Omega^{A^*} \end{tikzcd} .$$

The concrete description of log follows by spelling out the coalgebra-to-algebra diagram that characterizes it. In particular, we have

$$\begin{aligned}
\mathsf{log}(x)(aw) &= (\Omega^{\alpha-1} \circ \delta^*_A \circ \mathrm{id} \times (t^{A*} \circ \mathrm{st} \circ T(\mathsf{log}))^A \circ \langle o,f\rangle(x))(aw)\\
&= \delta^*_A(\mathrm{id} \times (t^{A*} \circ \mathrm{st} \circ T(\mathsf{log}))^A \circ \langle o,f\rangle(x))(a,w)\\
&= ((t^{A*} \circ \mathrm{st} \circ T(\mathsf{log}))^A \circ f(x))(a)(w)\\
&= (t^{A*} \circ \mathrm{st} \circ T(\mathsf{log})(f(x)(a)))(w)\\
&= t(\mathrm{st} \circ T(\mathsf{log})(f(x)(a))(w))\\
&= t(T(\mathrm{ev}_w \circ \mathsf{log})(f(x)(a)))
\end{aligned}$$

for all $x \in X$, $a \in A$ and $w \in A^*$. □

EXAMPLE 4.4

We instantiate the trace semantics log from Proposition 4.3 for various choices of Ω, T and t. Similar to the instances in Example 3.5, we consider a coalgebra $\langle o,f\rangle\colon X \to \Omega \times T(X)^A$, and we always have $\mathsf{log}(x)(\varepsilon) = o(x)$. The cases of non-deterministic automata (NDA-∃, NDA-∀) and probabilistic automata (PA) are the same as in Example 3.5. However, in contrast to the Eilenberg–Moore approach and other approaches to trace semantics, a monad structure on T is not required here. This is convenient as it also allows to treat alternating automata (AA), where $T = \mathcal{P}_\mathrm{f}\mathcal{P}_\mathrm{f}$; the latter does not carry a monad structure [30].

	T	Ω	$t\colon T(\Omega) \to \Omega$	$\mathsf{log}(x)(aw)$
NDA-∃	$\mathcal{P}_\mathrm{f}$	$2 = \{\bot, \top\}$	$S \mapsto \bigvee S$	$\bigvee_{y \in f(x)(a)} \mathsf{log}(y)(w)$
NDA-∀	$\mathcal{P}_\mathrm{f}$	$2 = \{\bot, \top\}$	$S \mapsto \bigwedge S$	$\bigwedge_{y \in f(x)(a)} \mathsf{log}(y)(w)$
PA	$\mathcal{D}_\mathrm{fin}$	$[0, 1]$	$\varphi \mapsto \sum_{p \in [0,1]} p \cdot \varphi(p)$	$\sum_{y \in X} \mathsf{log}(y)(w) \cdot f(x)(a)(y)$
AA	$\mathcal{P}_\mathrm{f}\mathcal{P}_\mathrm{f}$	$2 = \{\bot, \top\}$	$S \mapsto \bigvee_{T \in S} \bigwedge_{b \in T} b$	$\bigvee_{T \in f(x)(a)} \bigwedge_{y \in T} \mathsf{log}(y)(w)$

We also describe a logic for polynomial functors constructed from a signature. Here, we model a signature by a functor $\Sigma\colon \mathbb{N} \to \mathbf{Sets}$, where $\mathbb{N}$ is the discrete category of natural numbers. This gives rise to a functor $H_\Sigma\colon \mathbf{Sets} \to \mathbf{Sets}$ as usual by $H_\Sigma(X) = \coprod_{n \in \mathbb{N}} \Sigma(n) \times X^n$. The initial algebra of H_Σ consists of closed terms (or finite node-labelled trees) over the signature.

Proposition 4.5

Let Ω be a meet semilattice with top element $\top$ as well as a bottom element $\bot$, let $T\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor, and $t\colon T(\Omega) \to \Omega$ a map. Let (Φ, α) be the initial H_Σ-algebra. The set Ω^Φ of 'tree' languages carries a corecursive algebra for the functor TH_Σ. Given a coalgebra $c\colon X \to TH_\Sigma(X)$, the unique coalgebra-to-algebra map $\mathsf{log}\colon X \to \Omega^\Phi$ is given by

$$\mathsf{log}(x)(\sigma(u_1, \dots, u_n)) = t(T(m) \circ c(x))\,, \text{where}$$

$$m = \left(u \mapsto \begin{cases} \bigwedge_i \mathsf{log}(x_i)(u_i) & \text{if } \exists x_1 \dots x_n.\, u = (\sigma, x_1, \dots, x_n) \\ \bot & \text{otherwise} \end{cases} \right) \colon H_\Sigma(X) \to \Omega$$

for all $x \in X$ and $\sigma(u_1, \dots, u_n) \in \Phi$.

Proof. We use $\mathbf{C} = \mathbf{D} = \mathbf{Sets}$, $F = G = \Omega^-$, $B = L = H_\Sigma$. The map t extends to a modality $\tau\colon TG \Rightarrow G$ as in the proof of Proposition 4.3. The logic $\delta\colon H_\Sigma \Omega^- \Rightarrow \Omega^{H_\Sigma(-)}$ is

$$\delta_X(\sigma_1, \phi_1, \dots, \phi_n)(\sigma_2, x_1, \dots, x_m) = \begin{cases} \bigwedge_i \phi_i(x_i) & \text{if } \sigma_1 = \sigma_2 \\ \bot & \text{otherwise} \end{cases}$$

The corecursive algebra $\ell_{\log}$ is then given by

$$\begin{tikzcd} TH_\Sigma(\Omega^\Phi) \arrow[r, "T(\delta)"] & T(\Omega^{H_\Sigma(\Phi)}) \arrow[r, "\mathrm{st}"] & T(\Omega)^{H_\Sigma(\Phi)} \arrow[r, "t^{H_\Sigma(\Phi)}"] & \Omega^{H_\Sigma(\Phi)} \arrow[r, "\Omega^{\alpha^{-1}}", "\cong"'] & \Omega^\Phi \end{tikzcd} .$$

Now, given a coalgebra $c\colon X \to TH_\Sigma(X)$, we compute

$$\begin{aligned} & \mathsf{log}(x)(\sigma(u_1, \dots, u_n)) \\ & = (\Omega^{\alpha^{-1}} \circ t^{H_\Sigma \Phi} \circ \mathrm{st} \circ T(\delta_\Phi) \circ TH_\Sigma(\mathsf{log}) \circ c(x))(\sigma(u_1, \dots, u_n)) \\ & = t((\mathrm{st} \circ T(\delta_\Phi) \circ TH_\Sigma(\mathsf{log}) \circ c(x))(\alpha^{-1}(\sigma(u_1, \dots, u_n)))) \\ & = t((\mathrm{st} \circ T(\delta_\Phi) \circ TH_\Sigma(\mathsf{log}) \circ c(x))(\sigma, u_1, \dots, u_n)) \\ & = t(T(\mathrm{ev}_{(\sigma, u_1, \dots, u_n)})(T(\delta_\Phi) \circ TH_\Sigma(\mathsf{log}) \circ c(x))) \\ & = t(T(\mathrm{ev}_{(\sigma, u_1, \dots, u_n)} \circ \delta_\Phi \circ H_\Sigma(\mathsf{log}))(c(x))) \end{aligned}$$

To conclude, we analyse the map $\mathrm{ev}_{(\sigma,u_1,\dots,u_n)} \circ \delta_\Phi \circ H_\Sigma(\mathsf{log})$:

$$\begin{aligned}
&\mathrm{ev}_{(\sigma,u_1,\dots,u_n)}(\delta_\Phi(H_\Sigma(\mathsf{log})(u)))\\
&= \delta_\Phi(H_\Sigma(\mathsf{log})(u))(\sigma,u_1,\dots,u_n)\\
&= \begin{cases} \bigwedge_i \mathsf{log}(x_i)(u_i) & \text{if } \exists x_1 \dots x_n.\, u = (\sigma,x_1,\dots,x_n)\\ \bot & \text{otherwise} \end{cases} .
\end{aligned}$$

This coincides with m in the statement of the proposition. □

Example 4.6

Given a signature Σ, a coalgebra $c\colon X \to \mathcal{P}_{\mathrm{f}} H_\Sigma(X)$ is a (top-down) *tree automaton*. With $\Omega = \{\bot,\top\}$ and $t(S) = \bigvee S$, Proposition 4.5 gives

$$\mathsf{log}(x)(\sigma(t_1,\dots,t_n)) = \top \text{ iff } \exists x_1 \dots x_n.(\sigma,x_1,\dots,x_n) \in c(x) \wedge \bigwedge_{1\le i\le n} \mathsf{log}(x_i)(t_i)$$

for every state $x \in X$ and tree $\sigma(t_1,\dots,t_n)$. This is the standard semantics of tree automata. It is easily adapted to *weighted* tree automata, see [29].

In both Example 4.4 and Example 4.6, the step-induced coalgebra lifting $F^{\delta \circledcirc \tau}$ (respectively $F^{\tau \circledcirc \delta}$) of the underlying logic corresponds to reverse determinization, see [29, 42] for details. In particular, in Example 4.6, it maps a top-down tree automaton to the corresponding bottom-up tree automaton.

5 Traces via Kleisli

In this section, we briefly recall the 'Kleisli approach' to trace semantics [16] and cast it in our abstract framework. It applies to coalgebras for a composite functor TA, where T is a monad modelling the type of branching and A is a functor. For example, a coalgebra $X \to \mathcal{P}(\Sigma \times X + S)$ has an associated map $X \to \mathcal{P}(\Sigma^* \times S)$ that sends a state $x \in X$ to the set of its complete traces. (Taking $S = 1$, this is the usual language semantics of a nondeterministic automaton.) To fit this to our framework, the monad T is $\mathcal{P}$ and the functor A is $(\Sigma \times -) + S$. In general, the following assumptions are used.

Assumption 5.1
(Traces via Kleisli).
In this section, we assume

1. An endofunctor $A\colon \mathbf{C} \to \mathbf{C}$ with an initial algebra $\beta\colon A(\Psi) \xrightarrow{\cong} \Psi$.
2. A monad (T,η,μ), with the standard adjunction $J \dashv U$ between categories $\mathbf{C} \Longleftrightarrow \mathcal{K}\ell(T)$, where $J(X) = X$ and $U(Y) = T(Y)$.
3. An extension $\overline{A}$ of A, as below:

$$\begin{tikzcd}
\mathcal{K}\ell(T) \arrow[r, "\overline{A}"] & \mathcal{K}\ell(T) \\
\mathbf{C} \arrow[u, "J"] \arrow[r, "A"'] & \mathbf{C} \arrow[u, "J"']
\end{tikzcd} \qquad (15)$$

or, equivalently, a $\mathcal{K}\ell$-law $\lambda\colon AT \Rightarrow TA$.

4. $(\Psi, J(\beta^{-1}))$ is a final $\overline{A}$-coalgebra.

In the case that A is the functor $(\Sigma \times -) + S$, its initial algebra is carried by $\Sigma^* \times S$, and the canonical $\mathcal{K}\ell$-law is given at X by

$$\Sigma \times TX + S \xrightarrow{[T\mathsf{inl} \circ st_{\Sigma,X}, T\mathsf{inr} \circ \eta_S]} T(\Sigma \times X + S)$$

A central observation for the Kleisli approach to traces is that the fourth assumption holds under certain order enrichment requirements on $\mathcal{K}\ell(T)$, see [16]. In particular, these hold when T is the powerset monad, the (discrete) subdistribution monad or the lift monad.

The above assumptions give rise to the following instance of our setting (1):

$$\begin{tikzcd} \mathbf{C} \arrow[loop left, "TA"] \arrow[r, bend left, "J"] \arrow[r, phantom, "\perp"] & \mathcal{K}\ell(T) \arrow[l, bend left, "U"] \arrow[loop right, "\overline{A}"] \end{tikzcd}$$

with

$$\rho\colon TAU \Longrightarrow U\overline{A} \text{ where } \rho_X = \left(TATX \xrightarrow{T(\lambda)} T^2AX \xrightarrow{\mu} TAX\right)$$

Similar to the $\mathcal{EM}$-case in Section 3, the map of adjunctions is most easily given in terms of $\rho_4\colon TA \Rightarrow U\overline{A}J$ as the identity, using that $\overline{A}$ extends A.

We apply the step-induced algebra lifting $G_\rho\colon \mathrm{Alg}(\overline{A}) \to \mathrm{Alg}(TA)$ to the inverse of the final $\overline{A}$-coalgebra and obtain a corecursive TA-algebra, called ℓ_{kl}:

$$\begin{aligned} \left(TAT(\Psi) \xrightarrow{\ell_{\mathrm{kl}}} T(\Psi)\right) &:= G_\rho(\Psi, J(\beta^{-1})^{-1}) \\ &= G_\rho(\Psi, J(\beta)) \\ &= \left(TAT(\Psi) \xrightarrow{T(\lambda)} T^2A(\Psi) \xrightarrow{\mu} TA(\Psi) \xrightarrow{T(\beta)} T(\Psi)\right) \end{aligned} \tag{16}$$

By Theorem 2.6, this algebra is corecursive, i.e. for every coalgebra $c\colon X \to TA(X)$, there is a unique map kl_c as below:

$$\begin{tikzcd} X \arrow[r, dashed, "\mathsf{kl}_c"] \arrow[d, "c"'] & T(\Psi) \\ TA(X) \arrow[r, dashed, "TA(\mathsf{kl}_c)"] & TAT(\Psi) \arrow[u, "\ell_{\mathrm{kl}}"'] \end{tikzcd} \tag{17}$$

The trace semantics is exactly as in [16], to which we refer for examples. For later use, we note the following.

LEMMA 5.2
The above map $\ell_{\mathrm{kl}}\colon TAT(\Psi) \to T(\Psi)$ is a map of Eilenberg–Moore algebras $\mu_{AT(\Psi)} \to \mu_\Psi$.

PROOF. This follows by an easy calculation:

$$\begin{aligned} \ell_{\mathrm{kl}} \circ \mu \;=\; T(\beta) \circ \mu \circ T(\lambda) \circ \mu &= T(\beta) \circ \mu \circ \mu \circ T^2(\lambda) \\ &= T(\beta) \circ \mu \circ T(\mu) \circ T^2(\lambda) \\ &= \mu \circ T^2(\beta) \circ T(\mu) \circ T^2(\lambda) \;=\; \mu \circ T(\ell_{\mathrm{kl}}). \end{aligned}$$

□

6 Partial traces for input/output

6.1 Introduction

To illustrate the versatility of our framework, we show next that it underpins a trace example quite different from the previous ones, one that arises in programming language semantics and involves both input and output actions [5].

To avoid confusion, it must be noted that the word 'trace' is used with a different meaning in the automata and semantics communities, as follows.

- In the automata literature and the previous sections, a 'trace' ends in acceptance. Semanticists would call this a 'complete trace'.
- By contrast, in the semantics literature [5, 24, 32, 33, 36, 41] and this section, a 'trace' need not end in acceptance. For example, a program that prints `Hello` and then diverges (hangs) must be distinguished from one that simply diverges, even though—since neither terminates—neither has a complete trace. Accordingly, the string `Hello` is said to be a 'trace' of the former program (but not the latter), and so is each prefix. Automata theorists would call these 'partial traces'.

This section applies our framework to traces of the second kind, but before doing that, we need two pieces of background. The first (Section 6.2) explains that, in a transition system for I/O, a state's set of traces form a *strategy*. The second (Section 6.3) characterizes the poset of all strategies as a final coalgebra. This is a result that appeared in [5].

6.2 Trace sets as strategies

The story begins by fixing a *signature*, which consists of a set K of operations, and for each $k \in K$ a set $\mathsf{Ar}(k)$ called its *arity*. Each operation $k \in K$ is regarded as an output message requesting input, and $\mathsf{Ar}(k)$ as the set of acceptable inputs.[1] Accordingly, we use the functor:

$$X \quad \mapsto \quad \mathcal{P}\Big(\textstyle\sum_{k\in K} X^{\mathsf{Ar}(k)}\Big) \quad = \quad \mathcal{P}\Big(\textstyle\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} X\Big). \tag{18}$$

A *transition system* is a coalgebra $c\colon X \to \mathcal{P}(\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} X)$. For such a system, a state $x \in X$ represents a program that nondeterministically outputs some $k \in K$, then pauses until it receives some $i \in \mathsf{Ar}(k)$ and then is in another state. We write:

$$x \overset{k}{\Longrightarrow} (y_i)_{i\in\mathsf{Ar}(k)} \qquad \text{for} \qquad (k, (y_i)_{i\in\mathsf{Ar}(k)}) \in c(x).$$

A *play* is a finite or infinite sequence $k_0, i_0, k_1, i_1, \ldots$, where $k_r \in K$ and $i_r \in \mathsf{Ar}(k_r)$. It is so called because it may be viewed a play in a game of two players, called proponent and opponent, where each output is a proponent move and each input an opponent move. (The game terminology is slightly misleading in that there is no notion of winning and play can continue forever.) A play of even length is *active ending* and one of odd length is *passive ending*.

A *strategy* (more precisely, nondeterministic finite trace strategy) is a set σ of passive-ending plays such that $sik \in \sigma$ implies $s \in \sigma$. Again, this terminology is based on the game idea, as a strategy tells proponent (nondeterminstically) how to play. The poset of all strategies, ordered by inclusion ($\subseteq$), is written Strat.

[1] Many-sorted signatures, in the guise of 'interaction structures', are used for a similar purpose in [14].

Let (X, c) be a transition system and $x \in X$ a state. A passive-ending play $k_0, i_0, \ldots, k_n$ is said to be a *trace* of x when there is a sequence

$$x = x_0 \overset{k_0}{\Longrightarrow} (y_i^0)_{i \in \mathsf{Ar}(k_0)} \ , \qquad y_{i_0}^0 = x_1 \overset{k_1}{\Longrightarrow} (y_i^1)_{i \in \mathsf{Ar}(k_1)} \ , \ \cdots$$

The set of all such traces forms a strategy. Note that active-ending traces need not be considered, since these are determined by the passive-ending traces. Infinite traces are not considered in [5], nor are they here. Conversely, every strategy can be obtained in this way [5, Proposition 6.1].

6.3 Strategies form a final coalgebra

A *complete semilattice* is a poset with all suprema. Hence it also has all infima, which allows it to be called a 'complete lattice'. Clearly, the poset Strat of all strategies, ordered by inclusion ($\subseteq$), is a complete semilattice. Let **CSL** be the category of complete semilattices and *homomorphisms*, i.e. monotone functions that preserve suprema. It was shown in [5] that Strat is a final coalgebra for a certain endofunctor on **CSL**, which we shall describe in several steps.

Firstly, an *almost complete semilattice* is a poset where every nonempty subset has a supremum. Hence, every lower-bounded subset has an infimum, but binary meets $a \wedge b$ need not exist in general. Let **ACSL** be the category of almost complete semilattices and *homomorphisms*, i.e. monotone functions that preserve suprema of nonempty sets. Informally, our motivation for using this category is the fact that, up to trace equivalence, an I/O action such as printing commutes with binary nondeterminism, and more generally with I-ary nondeterministic choice for any nonempty set I. This point (and the special role of the empty set) is developed in more detail in [5].

For any set J, we define two functors:

$$\mathbf{CSL}^J \xrightarrow{\prod_J} \mathbf{ACSL} \qquad\qquad \mathbf{ACSL}^J \xrightarrow{\bigoplus_J^\perp} \mathbf{CSL}$$

as follows. (In [5], they are linked to universal properties.)

- For a family $(A_j)_{j \in J}$ of complete semilattices, let $\prod_{j \in J} A_j$ be the cartesian product. Endowed with pointwise order, it is an almost complete (in fact complete) semilattice.
- For a family $(f_j \colon A_j \to B_j)_{j \in J}$ of complete semilattice homomorphisms, let $\prod_{j \in J} f_j \colon \prod_{j \in J} A_j \to \prod_{j \in J} B_j$ be the map sending $(a_j)_{j \in J}$ to $(f_j a_j)_{j \in J}$.
- For a family $(A_j)_{j \in J}$ of almost complete semilattices, let $\bigoplus_{j \in J}^\perp A_j$ be the set of pairs $(U, (a_j)_{j \in U})$ where $U \in \mathcal{P}J$ and $a_j \in A_j$ for all $j \in U$. It is a complete semilattice when endowed with the following order: we have $(U, (a_j)_{j \in U}) \leqslant (V, (b_j)_{j \in V})$ when $U \subseteq V$ and $a_j \leqslant b_j$ for all $j \in U$.
- For a family $(f_j \colon A_j \to B_j)_{j \in J}$ of almost complete semilattice homomorphisms, let $\bigoplus_{j \in J}^\perp f_j \colon \bigoplus_{j \in J}^\perp A_j \to \bigoplus_{j \in J}^\perp B_j$ be the map sending a pair $(U, (a_j)_{j \in U})$ to $(U, (f_j a_j)_{j \in U})$.

From these, we build our endofunctor

$$\bigoplus_{k \in K}^\perp \prod_{i \in \mathsf{Ar}(k)} : \mathbf{CSL} \to \mathbf{CSL}$$

whose final coalgebra is given as follows.

THEOREM 6.1

[5, Theorem 6.3] Let $\Psi \colon \mathsf{Strat} \to \bigoplus_{k \in K}^\perp \prod_{i \in \mathsf{Ar}(k)} \mathsf{Strat}$ send a strategy σ to $(\mathsf{Init}\,\sigma,$

$((\sigma/ki_i)_{i\in\mathsf{Ar}(k)})_{k\in\mathsf{Init}\,\sigma})$, where

$$\begin{aligned}\mathsf{Init}\,\sigma &\stackrel{\text{def}}{=} \{k\in K \mid (k)\in\sigma\}\\ \sigma/ki &\stackrel{\text{def}}{=} \{s \mid k.i.s\in\sigma\}\end{aligned}$$

Then (Strat,Ψ) is a final $\bigoplus^{\perp}_{k\in K}\prod_{i\in\mathsf{Ar}(k)}$-coalgebra.

6.4 The step

With the background completed, we now want to instantiate our general setting to form an account of traces. Our adjunction and endofunctors are as follows:

$$\begin{tikzcd}
\mathbf{Sets} \arrow[loop left, "\mathcal{P}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)}"] \arrow[r, bend left, "\mathcal{P}"] \arrow[r, phantom, "\perp"] & \mathbf{CSL} \arrow[l, bend left, "U"] \arrow[loop right, "\bigoplus^{\perp}_{k\in K}\prod_{i\in\mathsf{Ar}(k)}"]
\end{tikzcd}$$

Here $U\colon \mathbf{CSL}\to\mathbf{Sets}$ is the forgetful functor, which is monadic. Explicitly, the free complete semilattice on a set X is $\mathcal{P}X$, ordered by inclusion ($\subseteq$), with unit $X\to\mathcal{P}X$ sending $x\mapsto\{x\}$. Likewise the forgetful functor $U\colon\mathbf{ACSL}\to\mathbf{Sets}$ is monadic. Explicitly, the free almost complete semilattice on a set X is the set $\mathcal{P}^{+}X$ of nonempty subsets, ordered by inclusion ($\subseteq$), with unit $X\to\mathcal{P}^{+}X$ sending $x\mapsto\{x\}$.

Our step is formulated using bimodules and 2-cells, which are explained in the Appendix. Any functor $U\colon\mathbf{D}\to\mathbf{C}$ gives rise to a bimodule $U^{\mathsf{Right}}\colon\mathbf{C}\nrightarrow\mathbf{D}$ by Definition B.4(2), and then, for any set J, to a bimodule $(U^{\mathsf{Right}})^{J}\colon\mathbf{C}^{J}\nrightarrow\mathbf{D}^{J}$ by Definition B.3. Central to our story are the following 2-cells (in the sense of Definition B.2(2)) defined for any set J.

$$\begin{tikzcd}
\mathbf{Sets}^{J} \arrow[r, "(U^{\mathsf{Right}})^{J}", "\shortmid"{marking}] \arrow[d, "\prod"'] \arrow[dr, phantom, "\prod\Downarrow"] & \mathbf{CSL}^{J} \arrow[d, "\prod"] \\
\mathbf{Sets} \arrow[r, "U^{\mathsf{Right}}"', "\shortmid"{marking}] & \mathbf{ACSL}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{Sets}^{J} \arrow[r, "(U^{\mathsf{Right}})^{J}", "\shortmid"{marking}] \arrow[d, "\mathcal{P}\sum"'] \arrow[dr, phantom, "\sum^{\sharp}\Downarrow"] & \mathbf{ACSL}^{J} \arrow[d, "\bigoplus^{\perp}"] \\
\mathbf{Sets} \arrow[r, "U^{\mathsf{Right}}"', "\shortmid"{marking}] & \mathbf{CSL}
\end{tikzcd}$$

They are defined as follows.

- Given a family of functions $(f_j\colon X_j\to B_j)_{j\in J}$, where X_j is a set and B_j a complete semilattice, the function $\prod_{j\in J}f_j\colon\ \prod_{j\in J}X_j\to\prod_{j\in J}B_j$ sends $(x_j)_{j\in J}$ to $(fx_j)_{j\in J}$.
- Given a family of functions $(f_j\colon X_j\to A_j)_{j\in J}$, where X_j is a set and A_j an almost complete semilattice, the function $\sum^{\sharp}_{j\in J}f_j\colon\mathcal{P}\sum_{j\in J}X_j\to\bigoplus^{\perp}_{j\in J}A_j$ sends R to $(L,(y_j)_{j\in L})$ where

$$\begin{aligned}L &= \{j\in J\mid \exists x\in X_j.\,\mathsf{in}_j\,x\in R\}\\ y_j &= \bigvee_{x\in X_j\,:\,\mathsf{in}_j\,x\in R} f_j(x) \qquad \text{for } j\in L.\end{aligned}$$

Note that, as in Sections 3 and 5, the ρ_4 version of $\sum^\sharp$ is an isomorphism, namely,

$$\begin{aligned} \textstyle\bigoplus^{\perp}_{j\in J} \mathcal{P}^{+}(X_j) &\xrightarrow{\ \cong\ } \mathcal{P}\big(\textstyle\sum_{j\in J} X_j\big) \\ (U,(Y_j)_{j\in U}) &\longmapsto \{(j,x) \mid j\in U, x\in Y_j\} \end{aligned}$$

Combining these 2-cells, we obtain the following 2-cell:

$$\begin{tikzcd} \mathbf{Sets} \arrow[r, "U^{\mathsf{Right}}", "\shortmid" marking] \arrow[d, "\mathcal{P}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)}"'] \arrow[dr, phantom, "\sum^{\sharp}_{k\in K}\prod_{i\in\mathsf{Ar}(k)} \Downarrow"] & \mathbf{CSL} \arrow[d, "\bigoplus^{\perp}_{k\in K}\prod_{i\in\mathsf{Ar}(k)}"] \\ \mathbf{Sets} \arrow[r, "U^{\mathsf{Right}}"', "\shortmid" marking] & \mathbf{CSL} \end{tikzcd} \tag{19}$$

As Theorem B.6 explains, this provides our step

$$\rho\colon \mathcal{P}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} U \Rightarrow U\bigoplus_{k\in K}^{\perp}\prod_{i\in\mathsf{Ar}(k)} .$$

From Theorem 6.1 with Proposition B.9(1), we see that, for every coalgebra $c\colon X \to \mathcal{P}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} X$, there is a unique morphism to (Strat,Ψ). Specifically, [5, Theorem 6.6] tells us that what this morphism sends $x \in X$ to its set of traces. Finally, by Proposition B.9(2), $U_\rho(\mathsf{Strat},\Psi)$ is corecursive, and the map from (X,c) to it is the same, i.e. it sends $x\in X$ to its set of traces.

Note that, as in Section 3, we can use $\mathcal{P}^\rho$ to determinize a transition system (X,c). This is applied in [5, Section 6.2] to obtain a bisimulation method for trace equivalence.

6.5 Input, then output

We adapt the story above to use instead of (18) the functor

$$X \quad\mapsto\quad \textstyle\prod_{k\in K}\mathcal{P}(\mathsf{Ar}(k)\times X) \quad=\quad \prod_{k\in K}\mathcal{P}\sum_{i\in\mathsf{Ar}(k)} X.$$

Now a *transition system* is a coalgebra $c\colon X \to \prod_{k\in K}\mathcal{P}\sum_{i\in\mathsf{Ar}(k)} X$. In this case, the behaviour of a state $x\in X$ is to first input $k\in K$ and then nondeterministically output some $i\in\mathsf{Ar}(k)$, resulting in a new state x'. We write

$$x@k \overset{i}{\Longrightarrow} x' \qquad \text{for} \qquad (i,x')\in(c(x))_k.$$

Accordingly, the definitions of play, strategy and trace in Section 6.2 are adjusted as follows.

- A *play* is a finite or infinite sequence $k_0, i_0, k_1, i_1, \ldots$, where $k_r \in K$ and $i_r \in \mathsf{Ar}(k_r)$. A play of odd length is *active ending* and one of even length is *passive ending*.
- A *strategy* is a set σ of passive-ending plays such that $\varepsilon\in\sigma$ (where ε is the empty play) and $ski\in\sigma$ implies $s\in\sigma$.
- Let (X,c) be a transition system, and $x\in X$ a state. A passive-ending play $k_0, i_0, \ldots, k_n, i_n$ is said to be a *trace* of x when there is a sequence

  $$x = x_0 \qquad x_0@k_0 \overset{i_0}{\Longrightarrow} x_1\ , \qquad x_1@k_1 \overset{i_1}{\Longrightarrow} x_2\ , \ \cdots$$

  The set of all such traces form a strategy.

The poset Strat of all strategies, ordered by inclusion ($\subseteq$), forms an almost complete (in fact complete) semilattice. We adjust Theorem 6.1 to say that Strat carries a final coalgebra for the endofunctor $\prod_{k\in K}\bigoplus^{\perp}_{i\in\mathsf{Ar}(k)}$ on **ACSL**.

Finally, we have the same results as in Section 6.4, but instead of (19) we use the following 2-cell:

$$\begin{tikzcd}
\mathbf{Sets} \arrow[r, "U^{\mathsf{Right}}", "\shortmid" marking] \arrow[d, "\prod_{k\in K}\mathcal{P}\sum_{i\in\mathsf{Ar}(k)}"'] \arrow[dr, phantom, "\prod_{k\in K}\sum^{\sharp}_{i\in\mathsf{Ar}(k)}\Downarrow"] & \mathbf{ACSL} \arrow[d, "\prod_{k\in K}\bigoplus^{\perp}_{i\in\mathsf{Ar}(k)}"] \\
\mathbf{Sets} \arrow[r, "U^{\mathsf{Right}}"', "\shortmid" marking] & \mathbf{ACSL}
\end{tikzcd}$$

To summarize, we first told our story for transition systems with 'active' states, that output and then input. (These systems are sometimes called 'generative'.) In this section, we have adapted it for systems with 'passive' states that input and then output. (These systems are sometimes called 'reactive'.) Another variation would be transition systems with both active and passive states, as in [36].

7 Comparison

The presentation of trace semantics in terms of corecursive algebras allows us to compare the different approaches by constructing algebra morphisms between them. In three subsections, we compare the Eilenberg–Moore approach with the logical approach, the Kleisli approach with the logical approach, and finally we compare the Kleisli and Eilenberg–Moore approaches.

7.1 Eilenberg–Moore and logic

To compare the Eilenberg–Moore approach with the logical approach, we combine their assumptions, as follows.

Assumption 7.1
(Comparison Eilenberg–Moore and logic).
In this subsection, we assume an adjunction $F \dashv G$, endofunctors B, L and a monad T as follows:

$$\begin{tikzcd}
\mathbf{D}^{\mathrm{op}} \arrow[loop left, "L"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\perp"] & \mathbf{C} \arrow[l, bend right, "F"'] \arrow[loop above, "BT"] \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathcal{EM}(T) \arrow[l, bend left, "U"] \arrow[loop right, "\overline{B}"]
\end{tikzcd}$$

together with:

1. A final B-coalgebra $\zeta\colon \Theta \xrightarrow{\cong} B(\Theta)$.
2. An $\mathcal{EM}$-law $\kappa\colon TB \Rightarrow BT$, or equivalently, a lifting $\overline{B}$ of B.
3. An initial algebra $\alpha\colon L(\Phi) \xrightarrow{\cong} \Phi$.
4. A step $\delta\colon BG \Rightarrow GL$.
5. A step $\tau\colon TG \Rightarrow G$, whose components are $\mathcal{EM}$-algebras (a *monad action*).

Here we have assumed slightly more than the union of the assumptions of the two approaches. The step τ is an assumption of the logical approach in Section 4, but there the compatibility with the monad structure was not assumed—simply because T was not assumed to be a monad before. Here, we use this assumption as a first compatibility requirement between the logical and Eilenberg–Moore approaches.

We note that τ being a monad action is the same thing as τ being an $\mathcal{EM}$-law, involving the monad T on the left and the identity monad on the right. The next result is therefore an instance of Theorem 2.3.

Lemma 7.2
The following are equivalent:

1. a monad action $\tau_1 \colon TG \Rightarrow G$;
2. a map $\tau_2 \colon F \Rightarrow FT$, satisfying the obvious dual action equations;
3. a monad morphism $\tau_4 \colon T \Rightarrow GF$;
4. an extension $\widehat{F} \colon \mathcal{K}\ell(T) \to \mathbf{D}^{\mathrm{op}}$ $(= \mathcal{K}\ell(\mathrm{Id}))$ of F.
5. a lifting $\widehat{G} \colon \mathbf{D}^{\mathrm{op}} \to \mathcal{EM}(T)$ of G.

Such monad actions and the corresponding liftings are used, e.g. in [15, 17, 20] where $\widehat{F}$ is called Pred. We use $\widehat{\cdot}$ to indicate liftings associated with the step τ, in order to create a distinction with the lifting $\overline{\cdot}$ associated with κ.

We now start focusing on the actual comparison between the Eilenberg–Moore and logical approach. First, observe that the step $\delta \colon BG \Rightarrow GL$ gives a lifting $G_\delta \colon \mathrm{Alg}(L) \to \mathrm{Alg}(B)$, where G is a functor $\mathbf{D}^{\mathrm{op}} \to \mathbf{C}$. The 'opposite' requires some care: the initial algebra $\alpha \colon L(\Phi) \to \Phi$ in $\mathbf{D}$ forms a final coalgebra $\alpha \colon \Phi \to L(\Phi)$ in $\mathbf{D}^{\mathrm{op}}$, and thus a corecursive algebra $\alpha^{-1} \colon L(\Phi) \to \Phi$ in $\mathbf{D}^{\mathrm{op}}$. Hence, applying G_δ to the latter corecursive L-algebra gives a corecursive B-algebra, namely,

$$G_\delta(\Phi, \alpha^{-1}) := \Big(BG(\Phi) \xrightarrow{\ \delta\ } GL(\Phi) \xrightarrow{\ G(\alpha^{-1})\ } G(\Phi)\Big).$$

Since this algebra is corecursive, we obtain a unique map e as in the following diagram:

$$\begin{tikzcd}
\Theta \arrow[r, "\mathsf{e}"] \arrow[dd, "\zeta"', "\cong"] & G(\Phi) \\
 & GL(\Phi) \arrow[u, "G(\alpha^{-1})"'] \\
B(\Theta) \arrow[r, "B(\mathsf{e})"] & BG(\Phi) \arrow[u, "\delta"']
\end{tikzcd}$$

(20)

This $\mathsf{e} \colon \Theta \to G(\Phi)$ is a morphism from the carrier of the corecursive algebra $\ell_{\mathrm{em}} \colon BT(\Theta) \to \Theta$, from the Eilenberg–Moore approach (6), to the carrier of the corecursive algebra $\ell^{\log} \colon BTG(\Phi) \to G(\Phi)$, from the logical approach (12). Note that, by the above diagram, e is a B-algebra morphism, whereas ℓ_{em} and $\ell^{\log}$ are BT-algebras. We next describe a sufficient condition under which the map e is a BT-algebra morphism from ℓ_{em} to $\ell^{\log}$, which implies that the logical trace semantics factors through the Eilenberg–Moore trace semantics, see subsequent Theorem 7.4.

LEMMA 7.3
The $\mathcal{EM}$-law $\kappa : TB \Rightarrow BT$ commutes with the step compositions in (11), as in

$$\begin{tikzcd}
TBG \arrow[rr, "\kappa G"] \arrow[dr, "\tau \circledcirc \delta"'] & & BTG \arrow[dl, "\delta \circledcirc \tau"] \\
& GL &
\end{tikzcd} \tag{21}$$

iff there is a natural transformation $\widehat{\delta} \colon \overline{B}\widehat{G} \Rightarrow \widehat{G}L$ satisfying $U(\widehat{\delta}) = \delta$ in

$$\begin{tikzcd}
& \mathcal{EM}(T) \arrow[loop right, "\overline{B}"] \arrow[dd, "U"'] & \text{with} \quad \overline{B}\widehat{G} \arrow[r, Rightarrow, "\widehat{\delta}"] & \widehat{G}L \\
\mathbf{D}^{\mathrm{op}} \arrow[loop left, "L"] \arrow[ur, "\widehat{G}"] \arrow[dr, "G"'] & & & \\
& \mathbf{C} \arrow[loop right, "B"] & \text{with} \quad BG \arrow[r, Rightarrow, "\delta"] & GL
\end{tikzcd}$$

The functor $\widehat{G} \colon \mathbf{D}^{\mathrm{op}} \to \mathcal{EM}(T)$ is the lifting corresponding to τ, see Lemma 7.2.

PROOF. The existence of such a $\widehat{\delta}$ amounts to the property that each component $\delta_X \colon BG(X) \to GL(X)$ is a T-algebra homomorphism from $\overline{B}\widehat{G}(X)$ to $\widehat{G}L(X)$, i.e. the following diagram commutes:

$$\begin{tikzcd}
TBG(X) \arrow[r, "T\delta"] \arrow[d, "\kappa"'] & TGL(X) \arrow[dd, "\tau"] \\
BTG(X) \arrow[d, "B(\tau)"'] & \\
BG(X) \arrow[r, "\delta"] & GL(X)
\end{tikzcd}$$

This corresponds exactly to (21), see (11). □

THEOREM 7.4
If the equivalent conditions in Lemma 7.3 hold, then the map e defined in (20) is a BT-algebra morphism from ℓ_{em} to $\ell^{\log}$, as on the left below.

$$\begin{tikzcd}
BT(\Theta) \arrow[r, "BT(\mathsf{e})"] \arrow[d, "\ell_{\mathrm{em}}"'] & BTG(\Phi) \arrow[d, "\ell^{\log}"] \\
\Theta \arrow[r, "\mathsf{e}"'] & G(\Phi)
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
& X \arrow[dl, "\mathsf{em}_c"'] \arrow[dr, "\mathsf{log}_c"] & \\
\Theta \arrow[rr, "\mathsf{e}"'] & & G(\Phi)
\end{tikzcd}$$

In that case, for any coalgebra $X \xrightarrow{c} BT(X)$, the triangle on the right commutes.

PROOF. We use that $\ell_{\mathrm{em}} = \zeta^{-1} \circ B(a)\colon BT(\varTheta) \to \varTheta$, where $((\varTheta, a), \zeta)$ is the final $\overline{B}$-coalgebra, see Section 3. We need to prove that the outer rectangle of the following diagram commutes.

$$\begin{tikzcd}
BT(\Theta) \arrow[r, "B(a)"] \arrow[d, "BT(\mathsf{e})"'] & B(\Theta) \arrow[rr, "\zeta^{-1}", "\cong"'] \arrow[d, "B(\mathsf{e})"] & & \Theta \arrow[d, "\mathsf{e}"] \\
BTG(\Phi) \arrow[r, "B(\tau_1)"'] \arrow[rrr, bend right, "\ell^{\log}"'] & BG(\Phi) \arrow[r, "\delta"'] & GL(\Phi) \arrow[r, "\cong", "G(\alpha^{-1})"'] & G(\Phi)
\end{tikzcd}$$

The rectangle on the right commutes by definition of e. For the square on the left, it suffices to show $\mathsf{e} \circ a = \tau_1 \circ T(\mathsf{e})$; this is equivalent to $F(a) \circ \overline{\mathsf{e}} = \tau_2 \circ \overline{\mathsf{e}}$ in

$$\begin{tikzcd}
\Phi \arrow[r, "\overline{e}=F(e)\circ\varepsilon"] & F(\Theta) \arrow[r, shift left, "F(a)"] \arrow[r, shift right, "\tau_2"'] & FT(\Theta)
\end{tikzcd}$$

Indeed, by transposing, we have on the one hand:

$$\overline{\mathsf{e} \circ a} \quad = \quad F(a \circ \mathsf{e}) \circ \varepsilon \quad = \quad F(a) \circ F(\mathsf{e}) \circ \varepsilon \quad = \quad F(a) \circ \overline{\mathsf{e}}.$$

And on the other hand, using that $\tau_2 \circ \varepsilon = F(\tau_1) \circ \varepsilon$ by Lemma A.1:

$$\begin{aligned}
\tau_2 \circ \overline{\mathsf{e}} &= \tau_2 \circ F(\mathsf{e}) \circ \varepsilon \\
&= FT(\mathsf{e}) \circ \tau_2 \circ \varepsilon \\
&= FT(\mathsf{e}) \circ F(\tau_1) \circ \varepsilon \\
&= \overline{\tau_1 \circ T(\mathsf{e})}
\end{aligned}$$

By transposing the map e in (20), it follows that $\overline{\mathsf{e}}\colon \varPhi \to F(\varTheta)$ is the unique morphism from the initial L-algebra $\alpha\colon L(\Phi) \xrightarrow{\cong} \Phi$ to $F(\zeta) \circ \delta_2\colon LF(\varTheta) \to F(\varTheta)$. Hence, for the desired equality $F(a) \circ \overline{\mathsf{e}} = \tau_2 \circ \overline{\mathsf{e}}$, it suffices to prove that $F(a)$ and τ_2 are both algebra homomorphisms from $F(\zeta) \circ \delta_2$ to a common algebra, which in turn follows from commutativity of the following diagram.

$$\begin{tikzcd}
LF(\Theta) \arrow[r, "L(\tau_2)"] \arrow[dd, "\delta_2"'] & LFT(\Theta) \arrow[d, "\delta_2"] & LF(\Theta) \arrow[l, "LF(a)"'] \arrow[d, "\delta_2"] \\
 & FBT(\Theta) \arrow[d, "F(\kappa)"] & FB(\Theta) \arrow[l, "FB(a)"'] \arrow[dd, "F(\zeta)"] \\
FB(\Theta) \arrow[r, "\tau_2"] \arrow[d, "F(\zeta)"'] & FTB(\Theta) \arrow[d, "FT(\zeta)"] & \\
F(\Theta) \arrow[r, "\tau_2"'] & FT(\Theta) & F(\Theta) \arrow[l, "F(a)"]
\end{tikzcd}$$

Using the translation $(-)_1 \leftrightarrow (-)_2$ of Theorem 2.3, one can show that the upper-left rectangle is equivalent to the assumption (21). To see this, we use Lemma 2.7 to obtain $(\delta \circledcirc \tau)_2 = (\delta_1 \circ B\tau_1)_2 = \delta_2 T \circ L\tau_2$ and $(\tau \circledcirc \delta)_2 = (\tau_1 L \circ T\delta_1)_2 = \tau_2 B \circ \delta_2$. Moreover, it is easy to check that

$(\delta_1 \circ B\tau_1 \circ \kappa G)_2 = F\kappa \circ (\delta_1 \circ B\tau_1)_2$. The lower-right rectangle commutes since $((\Theta, a), \zeta)$ is a $\overline{B}$-coalgebra. The other two squares commute by naturality.

For the second part of the theorem, let $c\colon X \to BT(X)$ be a coalgebra. Since e is an algebra morphism, the equation $\mathsf{e} \circ \mathsf{em}_c = \mathsf{log}_c$ follows by uniqueness of coalgebra-to-algebra morphisms from c to $\ell^{\log}$. □

The equality $\mathsf{e} \circ \mathsf{em}_c = \mathsf{log}_c$ means that equivalence w.r.t. Eilenberg–Moore trace semantics implies equivalence w.r.t. the logical trace semantics. The converse is, of course, true if e is monic. For that, it is sufficient if $\delta\colon BG \Rightarrow GL$ is *expressive*. Here expressiveness is the property that for any B-coalgebra, the unique coalgebra-to-algebra morphism to the corecursive algebra $G_\delta(\Phi, \alpha^{-1})$ factors as a B-coalgebra homomorphism followed by a mono. This holds in particular if the components $\delta_A\colon BG(A) \to GL(A)$ are all monic (in $\mathbf{C}$) [28].

LEMMA 7.5
If $\delta\colon BG \Rightarrow GL$ is expressive, then e is monic. Moreover, if δ is an isomorphism, then e is an iso as well.

PROOF. Expressivity of δ means that we have $\mathsf{e} = m \circ h$ for some coalgebra homomorphism h and mono m. By finality of ζ there is a B-coalgebra morphism h' such that $h' \circ h = \mathrm{id}$. It follows that h is monic (in $\mathbf{C}$), so that $m \circ h = \mathsf{e}$ is monic too.

For the second claim, if δ is an isomorphism, then $G(\alpha^{-1}) \circ \delta\colon BG(\Phi) \to G(\Phi)$ is an invertible corecursive B-algebra, which implies it is a final coalgebra (see [6, Proposition 7], which states the dual). It then follows from (20) that e is a coalgebra morphism from one final B-coalgebra to another, which means it is an isomorphism. □

Previously, we have seen both a class of examples of the Eilenberg–Moore approach (Theorem 3.4) and the logical approach (Proposition 4.3). Both arise from the same data: a monad T (just a functor in the logical approach) and an $\mathcal{EM}$-algebra t. We thus obtain, for these automata-like examples, both a logical trace semantics and a matching ‘Eilenberg–Moore’ semantics, where the latter essentially amounts to a determinization procedure. The underlying distributive laws satisfy (21) by construction, so that the two approaches coincide (as already seen in the concrete examples).

THEOREM 7.6
Let Ω be a set, $T\colon \mathbf{Sets} \to \mathbf{Sets}$ a monad and $t\colon T(\Omega) \to \Omega$ an $\mathcal{EM}$-algebra. The $\mathcal{EM}$-law κ of Theorem 3.4, together with δ, τ as defined in the proof of Proposition 4.3, satisfies (21). For any coalgebra $c\colon X \to \Omega \times T(X)^A$, the map log_c coincides (up to isomorphism) with the map em_c.

PROOF. To prove (21), i.e. $\delta \circledcirc \tau \circ \kappa = \tau \circledcirc \delta$, we first compute, following (11),

$$\begin{array}{rl} (\delta \circledcirc \tau)_X : & \Omega \times (T(\Omega^X))^A \longrightarrow \Omega^{A\times X+1} \\ = & \delta_X \circ (\mathrm{id} \times \tau_X^A) \\ = & \delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \\ (\tau \circledcirc \delta)_X : & T(\Omega \times (\Omega^X)^A) \longrightarrow \Omega^{A\times X+1} \\ = & \tau_{A\times X+1} \circ T(\delta_X) \\ = & t^{A\times X+1} \circ \mathrm{st} \circ T(\delta_X). \end{array}$$

Hence, we need to show that

$$\delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2)\rangle = t^{A\times X+1} \circ \mathrm{st} \circ T(\delta_X) \qquad (22)$$

for every set X. To this end, let $S \in T(\Omega \times (\Omega^X)^A)$ and $u \in (A \times X + 1)$. We first spell out the right-hand side:

$$\begin{aligned} & (t^{A\times X+1} \circ \mathrm{st} \circ T(\delta_X)(S))(u) \\ = \; & t((\mathrm{st} \circ T(\delta_X)(S))(u)) \\ = \; & t(T(\mathrm{ev}_u \circ \delta_X)(S)) \\ = \; & \begin{cases} t(T(\pi_1)(S)) & \text{if } u = * \in 1 \\ t(T(\mathrm{ev}_x \circ \mathrm{ev}_a \circ \pi_2)(S)) & \text{if } u = (a,x) \in A \times X \end{cases} \end{aligned}$$

In the last step, we used the definition of δ:

$$\mathrm{ev}_* \circ \delta_X(\omega,f) = \delta_X(\omega,f)(*) = \omega = \pi_1(\omega,f)\,,$$

$$\mathrm{ev}_{(a,x)} \circ \delta_X(\omega,f) = \delta_X(\omega,f)(a,x) = f(a)(x) = \mathrm{ev}_x \circ \mathrm{ev}_a \circ \pi_2(\omega,f)\,.$$

For the left-hand side of (22), distinguish cases $* \in 1$ and $(a,x) \in A \times X$.

$$\begin{aligned} & (\delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2)\rangle(S))(*) \\ & = \pi_1(\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2)\rangle(S)) \\ & = t(T(\pi_1)(S)) \end{aligned}$$

which matches the right-hand side of (22). For $(a,x) \in A \times X$, we have

$$\begin{aligned} & (\delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2)\rangle(S))(a,x) \\ & = (((t^X \circ \mathrm{st})^A \circ \mathrm{st})(T(\pi_2)(S)))(a)(x) \\ & = (((t^X)^A \circ \mathrm{st}^A \circ \mathrm{st})(T(\pi_2)(S)))(a)(x) \\ & = (t^X \circ \mathrm{st}(\mathrm{st}(T(\pi_2)(S))(a)))(x) \\ & = (t^X \circ \mathrm{st}(T(\mathrm{ev}_a)(T(\pi_2)(S))))(x) \\ & = (t^X \circ \mathrm{st}(T(\mathrm{ev}_a \circ \pi_2)(S)))(x) \\ & = t(\mathrm{st}(T(\mathrm{ev}_a \circ \pi_2)(S))(x)) \\ & = t(T(\mathrm{ev}_x) \circ T(\mathrm{ev}_a \circ \pi_2)(S)) \\ & = t(T(\mathrm{ev}_x \circ \mathrm{ev}_a \circ \pi_2)(S)) \end{aligned}$$

which also matches the right-hand side; hence, we obtain (22) as desired.

Since (21) is satisfied, it follows from Theorem 7.4 that $\mathsf{e} \circ \mathsf{em}_c = \mathsf{log}_c$. Since δ is an iso, e is an iso as well by Lemma 7.5. □

7.2 Kleisli and logic

To compare the Kleisli approach to the logical approach, we combine their assumptions. Further, similar to the comparison between Eilenberg–Moore and logic in the previous section, we assume a first compatibility criterion by requiring the components τ to be componentwise Eilenberg–Moore algebras.

Assumption 7.7
(Comparison Kleisli and logic).
In this subsection, we assume an adjunction $F \dashv G$, endofunctors A, L and a monad T as follows:

$$\begin{tikzcd}
\mathbf{D}^{\mathrm{op}} \arrow[loop left, "L"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\perp"] & \mathbf{C} \arrow[l, bend right, "F"'] \arrow[loop above, "TA"] \arrow[r, bend left, "J"] \arrow[r, phantom, "\perp"] & \mathcal{K}\ell(T) \arrow[l, bend left, "V"] \arrow[loop right, "\overline{A}"]
\end{tikzcd}$$

together with

1. An initial algebra $\beta\colon A(\Psi) \xrightarrow{\cong} \Psi$.
2. A $\mathcal{K}\ell$-law $\lambda\colon AT \Rightarrow TA$, or equivalently, an extension $\overline{A}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ of $A\colon \mathbf{C} \to \mathbf{C}$.
3. $(\Psi, J(\beta^{-1}))$ is a final $\overline{A}$-coalgebra.
4. An initial algebra $\alpha\colon L(\Phi) \xrightarrow{\cong} \Phi$.
5. A step $\delta\colon AG \Rightarrow GL$.
6. A step $\tau\colon TG \Rightarrow G$, whose components are $\mathcal{EM}$-algebras (a monad action).

By the last assumption, τ satisfies the equivalent conditions in Lemma 7.2; again, this is in itself not part of the logical approach but is used in the comparison to the Kleisli approach to trace semantics. We obtain the following unique coalgebra-to-algebra morphism k from the initial A-algebra:

$$\begin{tikzcd}
\Psi \arrow[r, "\mathsf{k}"] & G(\Phi) \\
 & GL(\Phi) \arrow[u, "G(\alpha^{-1})"'] \\
A(\Psi) \arrow[uu, "\beta", "\cong"'] \arrow[r, "A(\mathsf{k})"] & AG(\Phi) \arrow[u, "\delta"']
\end{tikzcd} \tag{23}$$

Since τ is a monad action, for every X, $G(X)$ carries an Eilenberg–Moore algebra τ_X. Thus, we can take the adjoint transpose (w.r.t. the Eilenberg–Moore adjunction) $\overline{\mathsf{k}} = \tau_\Phi \circ T(\mathsf{k})\colon T(\Psi) \to G(\Phi)$. We then have the following analogue of Theorem 7.4.

Lemma 7.8
The distributive law $\lambda\colon AT \Rightarrow TA$ commutes with the logics in (11), as in

$$\begin{tikzcd}
ATG \arrow[rr, "\lambda G"] \arrow[dr, "\delta \circledcirc \tau"'] & & TAG \arrow[dl, "\tau \circledcirc \delta"] \\
 & GL &
\end{tikzcd} \tag{24}$$

iff there is a natural transformation $\widehat{\delta}\colon L\widehat{F} \Rightarrow \widehat{F}\overline{A}$ satisfying $\widehat{\delta}J = \delta$ in

$$\begin{tikzcd}
 & \mathcal{K}\ell(T) \arrow[loop right, "\overline{A}"] \arrow[dl, "\widehat{F}"'] & \text{with} & L\widehat{F} \arrow[r, Rightarrow, "\widehat{\delta}"] & \widehat{F}\overline{A} \\
\mathbf{D}^{\mathrm{op}} \arrow[loop left, "L"] & & & & \\
 & \mathbf{C} \arrow[loop right, "A"] \arrow[ul, "F"] \arrow[uu, "J"'] & \text{with} & LF \arrow[r, Rightarrow, "\delta"] & FA
\end{tikzcd}$$

The two natural transformation on the right are written in $\mathbf{D}$ instead of $\mathbf{D}^{\mathrm{op}}$. The functor $\widehat{F}\colon \mathcal{K}\ell(T) \to \mathbf{D}^{\mathrm{op}}$ is the extension corresponding to τ, Lemma 7.2.

PROOF. The condition $\widehat{\delta}J = \delta$ simply means that $\widehat{\delta}_X = \delta_X$ for every object X in $\mathbf{C}$. Naturality of $\widehat{\delta}$ amounts to commutativity of the outside of the diagram below, for every map $f\colon X \to T(Y)$.

$$\begin{tikzcd}
LF(Y) \arrow[r, "\delta"] \arrow[dd, "L(\tau)"'] \arrow[ddd, bend right=60, "L\widehat{F}(f)"'] & FA(Y) \arrow[d, "\tau"] \arrow[ddd, bend left=60, "\widehat{F}\overline{A}(f)"] \\
 & FTA(Y) \arrow[d, "F(\lambda)"] \\
LFT(Y) \arrow[r, "\delta"] \arrow[d, "LF(f)"'] & FAT(Y) \arrow[d, "FA(f)"] \\
LF(X) \arrow[r, "\delta"'] & FA(X)
\end{tikzcd}$$

The lower rectangle commutes by naturality; the upper is equivalent to (24). Hence, (24) implies naturality. Conversely, if $\widehat{\delta}$ is natural, then the upper rectangle commutes for each Y by taking $f = \mathrm{id}[TY]$ (the identity map in $\mathbf{C}$). □

THEOREM 7.9
If the equivalent conditions in Lemma 7.8 hold, then the map $\overline{\mathsf{k}} = \tau_{\Phi} \circ T(\mathsf{k})\colon T(\Psi) \to G(\Phi)$ is a TA-algebra morphism from ℓ_{kl} to $\ell_{\log}$, as on the left below.

$$\begin{tikzcd}
TAT(\Psi) \arrow[r, "TA(\overline{\mathsf{k}})"] \arrow[d, "\ell_{\mathrm{kl}}"'] & TAG(\Phi) \arrow[d, "\ell_{\log}"] \\
T(\Psi) \arrow[r, "\overline{\mathsf{k}}"] & G(\Phi)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & X \arrow[dl, "\mathsf{kl}_c"'] \arrow[dr, "\mathsf{log}_c"] & \\
T(\Psi) \arrow[rr, "\overline{\mathsf{k}}"] & & G(\Phi)
\end{tikzcd}$$

In that case, for any coalgebra $c\colon X \to TA(X)$, there is a commuting triangle as on the right above.

PROOF. Consider the following diagram.

$$\begin{tikzcd}
TAT(\Psi) \arrow[r, "TAT(\mathsf{k})"] \arrow[d, "T(\lambda)"] \arrow[ddd, bend right, "\ell_{\mathrm{kl}}"'] & TATG(\Phi) \arrow[rr, "TA(\tau)"] \arrow[d, "T(\lambda)"] & & TAG(\Phi) \arrow[d, "T(\delta)"'] \arrow[ddd, bend left, "\ell_{\log}"] \\
TTA(\Psi) \arrow[r, "TTA(\mathsf{k})"] \arrow[d, "\mu"] & TTAG(\Phi) \arrow[r, "TT\delta"] \arrow[d, "\mu"] & TTGL(\Phi) \arrow[r, "T(\tau)"] \arrow[d, "\mu"] & TGL(\Phi) \arrow[d, "\tau"'] \\
TA(\Psi) \arrow[r, "TA(\mathsf{k})"] \arrow[d, "T(\beta)"] & TAG(\Phi) \arrow[r, "T(\delta)"] & TGL(\Phi) \arrow[r, "\tau"] \arrow[d, "TG(\alpha^{-1})"] & GL(\Phi) \arrow[d, "G(\alpha^{-1})"'] \\
T(\Psi) \arrow[rr, "T(\mathsf{k})"'] & & TG(\Phi) \arrow[r, "\tau"'] & G(\Phi)
\end{tikzcd}$$

Everything commutes: the upper right rectangle by assumption (24), the right-most square in the middle row since τ is an action, the outer shapes by definition of ℓ_{kl} and $\ell_{\log}$, the lower left rectangle by (23) and the rest by naturality.

For the second part of the theorem, since $\overline{\mathsf{k}}$ is an algebra morphism, we have that $\overline{\mathsf{k}} \circ \mathsf{kl}_c$ is a coalgebra-to-algebra morphism from c to the corecursive algebra $\ell_{\log}$. Hence, $\overline{\mathsf{k}} \circ \mathsf{kl}_c = \mathsf{log}_c$ by uniqueness of such morphisms. □

The above result gives a sufficient condition under which Kleisli trace equivalence implies logical trace equivalence. However, contrary to the case of traces in Eilenberg–Moore, in Lemma 7.5, we currently do not have a converse. The condition that δ has monic components is, surprisingly, not sufficient for $\overline{\mathsf{k}}$ to be monic, as confirmed by Example 7.10 below. In the comparison between Eilenberg–Moore and Kleisli traces in Section 7.3, a similar difficulty arises.

EXAMPLE 7.10

We give an example where $\delta \colon AG \Rightarrow GL$ is monic and (24) commutes, but where nevertheless logical equivalence does not imply Kleisli trace equivalence. Let $\mathbf{C} = \mathbf{D} = \mathbf{Sets}$, $F = G = 2^{-}$, $A = L = (-) + 1$, $T = \mathcal{P}_{\mathrm{f}}$, $\tau \colon \mathcal{P}_{\mathrm{f}} 2^{-} \Rightarrow 2^{-}$ given by union as before, and define the step δ, for $\varphi \in 2^X$, by $\delta_X(\varphi)(t) = \top$ iff $t \in X \wedge \varphi(t)$, and $\delta_X(*)(t) = \top$ (the latter differs from the step in Proposition 4.5). Notice that δ indeed has monic components. (In the conference version [21], we used a more general setting with $A = L = (\Sigma \times -) + 1$, where Σ is a fixed set. However, the associated δ is not monic if Σ contains more than one element, contrary to what is stated there. Indeed, we need to choose Σ to be a singleton for the example to go through.)

Let $\lambda \colon AT \Rightarrow TA$ be the distributive law from [16], given by $\lambda_X(S) = \{x \mid x \in S\}$ for $S \in \mathcal{P}_{\mathrm{f}}(X)$, and $\lambda(*) = \{*\}$. Then (24) is satisfied:

$$\begin{tikzcd}
\mathcal{P}_{\mathrm{f}}(2^X) + 1 \arrow[rr, "\lambda"] \arrow[d, "\tau+1"'] & & \mathcal{P}_{\mathrm{f}}(2^X + 1) \arrow[d, "\mathcal{P}_{\mathrm{f}}(\delta)"] \\
2^X + 1 \arrow[r, "\delta"] & 2^{X+1} & \mathcal{P}_{\mathrm{f}}(2^{X+1}) \arrow[l, "\tau"']
\end{tikzcd}$$

It is straightforward to check that this commutes. However, given a coalgebra $c \colon X \to \mathcal{P}_{\mathrm{f}}A(X)$, the induced logical semantics $\mathsf{log} \colon X \to 2^{\mathbb{N}}$ is $\mathsf{log}(x)(n) = \top$ iff $* \in c(x)$ or $n > 0 \wedge \exists y \in c(x).\, \mathsf{log}(y)(n-1) = \top$. In particular, this means that if $* \in c(x)$ and $* \in c(y)$ for some states x, y, then they are trace equivalent. This differs from the Kleisli semantics, which amounts to the usual language semantics of non-deterministic automata (over a singleton alphabet) [16].

Cîrstea [9] compares logical traces to a 'path-based semantics', which resembles the Kleisli approach (as well as [31]) but does not require a final $\overline{A}$-coalgebra. In particular, given a commutative monad T on **Sets** and a signature Σ, she considers a canonical distributive law $\lambda\colon H_\Sigma T \Rightarrow TH_\Sigma$, which coincides with the one in [16]. Cîrstea shows that, with $\Omega = T(1)$, $t = \mu_1\colon TT(1) \to T(1)$ and δ from the proof of Proposition 4.5 (assuming $T1$ to have enough structure to define that logic), the triangle (24) commutes (see [9, Lemma 5.12]).

7.3 Kleisli and Eilenberg–Moore

To compare the Eilenberg–Moore and Kleisli approaches, we first combine their assumptions. The Kleisli approach applies to TA-coalgebras; to match this, we make use of the variant of the Eilenberg–Moore approach for TA-coalgebras presented in Section 3.1. The latter approach uses a lifting of a functor B as well as a step relating A and B.

ASSUMPTION 7.11
(Comparison Kleisli and Eilenberg–Moore).
In this subsection, we assume to endofunctors A, B and a monad T, on a base category **C**, and liftings $\overline{A}, \overline{B}$ to Kleisli and Eilenberg–Moore-categories as follows:

$$\begin{tikzcd}
\mathcal{EM}(T) \arrow[loop left, "\overline{B}"] \arrow[r, bend right, "U"'] \arrow[r, phantom, "\perp"] & \mathbf{C} \arrow[l, bend right, "\mathcal{F}"'] \arrow[loop above, "TA"] \arrow[r, bend left, "J"] \arrow[r, phantom, "\perp"] & \mathcal{K}\ell(T) \arrow[l, bend left, "V"] \arrow[loop right, "\overline{A}"]
\end{tikzcd}$$

In this situation, we further assume the following ingredients, which combine earlier assumptions.

1. An initial algebra $\beta\colon A(\Psi) \xrightarrow{\cong} \Psi$.
2. A $\mathcal{K}\ell$-law $\lambda\colon AT \Rightarrow TA$, or equivalently, an extension $\overline{A}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ of the functor A.
3. $(\Psi, J(\beta^{-1}))$ is a final $\overline{A}$-coalgebra.
4. An $\mathcal{EM}$-law $\kappa\colon TB \Rightarrow BT$, or equivalently, a lifting $\overline{B}\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$.
5. A final coalgebra $\zeta\colon \Theta \xrightarrow{\cong} B(\Theta)$.
6. A step $\rho\colon AU \Rightarrow U\overline{B}$.

The step $\rho\colon AU \Rightarrow U\overline{B}$ is an assumption of the Eilenberg–Moore approach for TA-coalgebras in Section 3.1, defined on top of the assumptions for the Eilenberg–Moore approach for BT-coalgebras. Under a further assumption such a law corresponds to an *extension* natural transformation as in [22], see Proposition 7.12.

Recall from Section 3 that the final B-coalgebra (Θ, ζ) gives rise to a final $\overline{B}$-coalgebra $((\Theta, a), \zeta)$. We will make use of the counit ε of the $\mathcal{EM}$-adjunction $\mathcal{F} \dashv U$ as a step $U\varepsilon\colon TU \Rightarrow U$. Its components are $\mathcal{EM}$-algebras. For the trace semantics of TA-coalgebras via Eilenberg–Moore, see Section 3.1, we make use of the composed step:

$$U\varepsilon \circledcirc \rho \quad = \quad \left(TAU \xRightarrow{T\rho} TU\overline{B} \xRightarrow{U\varepsilon\overline{B}} U\overline{B} \right) \quad . \tag{25}$$

These assumptions form an instance of the assumptions in Section 7.2, where we compared Kleisli to logical trace semantics. In particular, in the latter we instantiate $\mathbf{D}^{\mathrm{op}}$ with $\mathcal{EM}(T)$, L with $\overline{B}$, δ with

$\rho\colon AU \Rightarrow U\overline{B}$ and τ with $U\varepsilon\colon UT \Rightarrow U$. Thus, we immediately obtain the comparison result from Theorem 7.9. For presentation purposes, we restate the relevant results and definitions.

There is the following unique coalgebra-to-algebra morphism k from the initial A-algebra:

$$\begin{tikzcd}
\Psi \arrow[r, dashed, "\mathsf{k}"] & \Theta & \\
& B(\Theta) \arrow[u, "\zeta^{-1}"'] \arrow[d, equal] & \\
& U\overline{B}(\Theta, a) & BT(\Theta) \arrow[ul, bend right, "B(a)"'] \arrow[uul, bend right, "\ell_{\mathrm{em}}"'] \\
& AU(\Theta, a) \arrow[u, "\rho"'] \arrow[d, equal] & TA(\Theta) \arrow[u, "U(\rho_2)"'] \\
A(\Psi) \arrow[uuuu, "\beta", "\cong"'] \arrow[r, dashed, "A(\mathsf{k})"] & A(\Theta) \arrow[ur, bend right, "\eta"'] &
\end{tikzcd} \tag{26}$$

The rectangle on the right commutes since $\ell_{\mathrm{em}} = \zeta^{-1} \circ B(a)$, by definition (6), and

$$\begin{aligned}
B(a) \circ U(\rho_2) \circ \eta &= B(a) \circ \rho \circ A(\eta) && \text{by Lemma A.1} \\
&= \rho \circ A(a) \circ A(\eta) && \text{since}a\colon (T(\Theta), \mu) \to (\Theta, a) \text{ in } \mathcal{EM}(T) \\
&= \rho.
\end{aligned}$$

Taking the adjoint transpose, w.r.t. the Eilenberg–Moore adjunction $\mathcal{F} \dashv U$, of this map $\mathsf{k}\colon \Psi \to \Theta = U(\Theta, a)$, yields a map of Eilenberg–Moore algebras:

$$\overline{\mathsf{k}} \;=\; \Big(\mathcal{F}(\Psi) \xrightarrow{\mathcal{F}(\mathsf{k})} \mathcal{F}U(\Theta, a) \xrightarrow{\varepsilon} (\Theta, a)\Big) \;=\; \Big(T(\Psi) \xrightarrow{T(\mathsf{k})} TU(\Theta, a) \xrightarrow{a} \Theta\Big).$$

We have seen in (16) that $T(\Psi)$ is the carrier of the corecursive algebra $\ell_{\mathrm{kl}}\colon TAT(\Psi) \to T(\Psi)$ giving Kleisli trace semantics. At the same time, Θ is the carrier of the corecursive algebra $\ell_{\mathrm{em}}\colon BT(\Theta) \to \Theta$ from (6) as well as the corecursive algebra

$$G_{U\varepsilon \circledcirc \rho}((\Theta, a), \zeta) = \ell_{\mathrm{em}}^{A} = \ell_{\mathrm{em}} \circ U(\rho_2)\colon TA(\Theta) \to \Theta\,,$$

where $((\Theta, a), \zeta)$ is the final $\overline{B}$-coalgebra, and the equality on the right is given by Lemma 3.8. Thus, the map $\overline{\mathsf{k}}\colon T(\Psi) \to \Theta$ relates the carriers of the corecursive TA-algebras ℓ_{kl} and ℓ_{em}^{A}. Like in the previous sections, we now give a sufficient condition for $\overline{\mathsf{k}}$ to be an algebra morphism.

PROPOSITION 7.12

In the above setting, the following three statements are equivalent.

1. The distributive law $\lambda\colon AT \Rightarrow TA$ commutes with the two composed steps $\rho \circledcirc U(\varepsilon)$ and $U(\varepsilon) \circledcirc \rho$, as in

$$\begin{tikzcd}
ATU \arrow[rr, "\lambda"] \arrow[dr, "\rho \circledcirc U(\varepsilon)"'] & & TAU \arrow[dl, "U(\varepsilon) \circledcirc \rho"] \\
& U\overline{B} &
\end{tikzcd} \tag{27}$$

2. There is a natural transformation $\mathfrak{c}\colon \widehat{\mathcal{F}}\overline{A} \Rightarrow \overline{B}\widehat{\mathcal{F}}$ satisfying $\mathfrak{c}J = \rho_2$ in

$$\begin{tikzcd}
 & \mathcal{K}\ell(T) \arrow[loop right, "\overline{A}"] \arrow[dl, "\widehat{\mathcal{F}}=K"'] & \text{with } \widehat{\mathcal{F}}\overline{A} \xRightarrow{\mathfrak{c}} \overline{B}\widehat{\mathcal{F}} \\
\mathcal{EM}(T) \arrow[loop left, "\overline{B}"] & & \\
 & \mathbf{C} \arrow[loop right, "A"] \arrow[ul, "\mathcal{F}"] \arrow[uu, "J"'] & \text{with } \mathcal{F}A \xRightarrow{\rho_2} \overline{B}\mathcal{F}
\end{tikzcd}$$

The functor $\widehat{\mathcal{F}}\colon \mathcal{K}\ell(T) \to \mathcal{EM}(T)$ is the extension corresponding to $U(\varepsilon)$, according to Theorem 2.3; it is often called the 'comparison' functor and then written as K.

3. The following 'extension requirement' from [22] commutes

$$\begin{tikzcd}
TAT \arrow[rr, "U(\rho_2)"] \arrow[d, "T(\lambda)"'] & & BTT \arrow[d, "B(\mu)"] \\
TTA \arrow[r, "\mu"'] & TA \arrow[r, "U(\rho_2)"'] & BT
\end{tikzcd} \tag{28}$$

PROOF. The equivalence of points (1) and (2) is an instance of Lemma 7.8, where it should be noted that we are instantiating $\mathbf{D}$ with $\mathcal{EM}(T)^{\mathrm{op}}$, which causes the two natural transformations in the above diagram to be in opposite direction.

We show the equivalence of (1) and (3). Using Lemma 2.7, it is straightforward to check, via Theorem 2.3, that $\big(\rho \circledcirc U(\varepsilon)\big)_2 = \big(\rho_1 \circ AU\varepsilon\big)_2 = \overline{B}\varepsilon \circ \rho_2$ and $U\varepsilon \circ T\rho_1 \circ \lambda = \rho_2 \circ \varepsilon \circ \mathcal{F}(\lambda)$. As a consequence, commutativity of Diagram (27) is equivalent to commutativity of the following diagram:

$$\begin{tikzcd}
\mathcal{F}AT \arrow[rr, "\rho_2"] \arrow[d, "\mathcal{F}(\lambda)"'] & & \overline{B}\mathcal{F}T \arrow[d, "\overline{B}(\varepsilon)"] \\
\mathcal{F}TA \arrow[r, "\varepsilon"'] & \mathcal{F}A \arrow[r, "\rho_2"'] & U\overline{B}
\end{tikzcd}$$

This amounts to the diagram in point (3). □

Under the above equivalent conditions, we obtain the desired algebra morphism.

THEOREM 7.13
If the equivalent conditions in Proposition 7.12 hold, then the map $\overline{\mathsf{k}}\colon T(\Psi) \to \Theta$ obtained from (26), is a TA-algebra morphism between corecursive algebras ℓ_{kl} and $\ell^{A}_{\mathrm{em}} = \ell_{\mathrm{em}} \circ U(\rho_2)$, as on the left below.

$$\begin{tikzcd}
TAT(\Psi) \arrow[r, "TA(\overline{\mathsf{k}})"] \arrow[d, "\ell_{\mathrm{kl}}"'] & TA(\Theta) \arrow[d, "\ell^{A}_{\mathrm{em}}=\ell_{\mathrm{em}}\circ U(\rho_2)"] \\
T(\Psi) \arrow[r, "\overline{\mathsf{k}}"] & \Theta
\end{tikzcd}
\qquad
\begin{tikzcd}
 & X \arrow[dl, "\mathsf{kl}_c"'] \arrow[dr, "\mathsf{em}^{A}_{c}"] & \\
T(\Psi) \arrow[rr, "\overline{\mathsf{k}}"] & & \Theta
\end{tikzcd}$$

In that case, for any coalgebra $c\colon X \to TA(X)$, there is a commuting triangle as on the right above, where em^A_c is the unique map from (X, c) to the corecursive algebra $\ell_{\mathrm{em}} \circ U(\rho_2)$.

PROOF. In order to prove commutation of the rectangle, we need to combine many earlier facts:

$$\begin{aligned}
\overline{\mathsf{k}} \circ \ell_{\mathrm{kl}} &\overset{(16)}{=} a \circ T(\mathsf{k}) \circ T(\beta) \circ \mu \circ T(\lambda) \\
&\overset{(26)}{=} a \circ T(\zeta^{-1} \circ B(a) \circ U(\rho_2) \circ \eta \circ A(\mathsf{k})) \circ \mu \circ T(\lambda) \\
&\overset{(5)}{=} \zeta^{-1} \circ B(a) \circ \kappa \circ TB(a) \circ TU(\rho_2) \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ BT(a) \circ \kappa \circ TU(\rho_2) \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ B(\mu) \circ \kappa \circ TU(\rho_2) \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&\overset{(9)}{=} \zeta^{-1} \circ B(a) \circ U(\rho_2) \circ \mu \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ U(\rho_2) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ BT(\mathsf{k}) \circ U(\rho_2) \circ \mu \circ T(\lambda) \\
&\overset{(28)}{=} \zeta^{-1} \circ B(a) \circ BT(\mathsf{k}) \circ B(\mu) \circ U(\rho_2) \\
&= \zeta^{-1} \circ B(a) \circ B(\mu) \circ BT^2(\mathsf{k}) \circ U(\rho_2) \\
&= \zeta^{-1} \circ B(a) \circ BT(a) \circ BT^2(\mathsf{k}) \circ U(\rho_2) \\
&= \ell_{\mathrm{em}} \circ BT(\overline{k}) \circ U(\rho_2) \\
&= \ell_{\mathrm{em}} \circ U(\rho_2) \circ TA(\overline{k}).
\end{aligned}$$

Now let a coalgebra $c\colon X \to TA(X)$ be given. We need to prove that $\overline{\mathsf{k}} \circ \mathsf{kl}_c$ satisfies the defining property of em^A_c. But this easy using the rectangle in the theorem:

$$\ell^A_{\mathrm{em}} \circ TA(\overline{\mathsf{k}} \circ \mathsf{kl}_c) \circ c \quad = \quad \overline{\mathsf{k}} \circ \ell_{\mathrm{kl}} \circ TA(\mathsf{kl}_c) \circ c \quad \overset{(17)}{=} \quad \overline{\mathsf{k}} \circ \mathsf{kl}_c$$

□

Just like in the comparison between Kleisli and logic, the above result gives a sufficient condition for the Eilenberg–Moore trace semantics to factor through the Kleisli trace semantics. However, again we do not know of a reasonable condition to ensure that the map $\overline{\mathsf{k}}$ is monic. Such a result is important for the comparison: it would ensure that two states are equivalent w.r.t. Kleisli traces iff they are equivalent w.r.t. Eilenberg–Moore traces (right now, we only have the implication from left to right). In [22], such a condition is also missing; monicity of $\overline{\mathsf{k}}$ is only shown to hold in several concrete examples.

In [22, Section 6], the Kleisli approach to coalgebraic trace semantics is compared with the Eilenberg–Moore approach, making use of an 'extension' natural transformation $\mathfrak{e}$ satisfying two requirements, namely,

$$\begin{tikzcd}
TAT \arrow[r, "T(\lambda)"] \arrow[d, "\mathfrak{e}"'] & T^2A \arrow[r, "\mu"] & TA \arrow[d, "\mathfrak{e}"] \\
BT^2 \arrow[rr, "B(\mu)"] & & BT
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2A \arrow[rr, "\mu"] \arrow[d, "T(\mathfrak{e})"'] & & TA \arrow[d, "\mathfrak{e}"] \\
TBT \arrow[r, "\kappa"] & BT^2 \arrow[r, "B(\mu)"] & BT
\end{tikzcd}$$

In the present step-based setting, the rectangle on the right occurred as (9) in Section 3.1, which contains the generalization of Eilenberg–Moore trace semantics that is used here. The first of the above two rectangles captures compatibility in Proposition 7.12 and is used for a comparison of Kleisli and Eilenberg–Moore semantics in Theorem 7.13. The conclusion is that the approach of this paper not only covers the approach of [22, Section 6] but also puts it in a wider step-based perspective, using corecursive algebras.

8 Completely iterative algebras

Milius [37] introduced a notion of 'complete iterativity' of algebras that is stronger than corecursiveness and has the advantage of being preserved by various constructions. So, whenever we encounter a corecursive algebra, it is natural to ask whether it is in fact completely iterative. This section shows that all our corecursive algebras are completely iterative (Theorem 8.3), and that this yields trace maps in more general settings.

Definition 8.1

Let **C** have binary coproducts. For an endofunctor H on **C**, an H-algebra $a\colon HA \to A$ is *completely iterative* when $[\mathrm{id}, a]$ is a corecursive $A + H$-coalgebra. Explicitly: when for every $c\colon X \to A + HX$ there is a unique $f : X \to A$ such that the following diagram commutes.

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "c"'] & A \\
A + HX \arrow[r, "A+Hf"'] & A + HA \arrow[u, "{[\mathrm{id},a]}"']
\end{tikzcd}$$

The following gives two useful ways of constructing such algebras.

Proposition 8.2

1. If $\zeta : A \to HA$ is a final H-coalgebra, then (A, ζ^{-1}) is completely iterative.
2. Given a step as in Section 2, the functor G_ρ preserves complete iterativity.

Proof. Part (1) is included in [37, Theorem 2.8], and part (2) is the dual of [18, Theorem 5.6]. □

We may thus say: *'step-induced algebra liftings of right adjoints preserve complete iterativity'*. We deduce the following strengthening of Theorem 2.6.

Theorem 8.3

Given a step as in Section 2 and a final coalgebra $\zeta : A \to HA$, the algebra $G_\rho(A, \zeta^{-1})$ is completely iterative.

if L has a final coalgebra (Ψ, ζ), then $G_\rho(A, \zeta^{-1})$ is completely iterative.

For example, in the setting of Section 3, we obtain the following variation of (7). Given a coalgebra $c\colon X \to \Theta + BT(X)$, there is a unique map, written as em_c, making the following square commute:

$$\begin{tikzcd}
X \arrow[rr, dashed, "\mathsf{em}_c"] \arrow[d, "c"'] & & \Theta \\
\Theta + BT(X) \arrow[rr, dashed, "\Theta+BT(\mathsf{em}_c)"'] & & \Theta + BT(\Theta) \arrow[u, "{[\mathrm{id},\ell_{\mathrm{em}}]}"']
\end{tikzcd} \tag{29}$$

In what sense is em_c a 'trace map'? Let us look at the special case of Example 3.2, where $B(X) = 2 \times X^A$ so that Θ is the set 2^{A^*} of languages, and T is the finite powerset monad. Think of c as a *generalized* nondeterministic automaton: as well as the usual accepting and rejecting states, there can also be 'semantic states' that are labelled with a language and from which there are no transitions. For a state $x \in X$, a *trace* of x is either a word appearing along a path from x to an accepting state (as usual), or a concatenation of words s and t, where s appears along a path from x to a semantic state labelled by L, and $t \in L$. With this definition, we see that em_c sends $x \in X$ to its set of traces.

Each of our examples is similar to this one: the completely iterative algebra yields trace semantics for a generalized transition system in which the semantics may sometimes be given directly.

9 Future work

The main contribution of this paper is a general treatment of trace semantics via corecursive algebras, constructed through an adjunction and a step, covering the 'Eilenberg–Moore', 'Kleisli' and 'logic' approaches to trace semantics. It is expected that our framework also works for other examples, such as the 'quasi-liftings' in [2], but this is left for future work. In [27], several examples of adjunctions are discussed in the context of automata theory, some of them the same as the adjunctions here, but with the aim of lifting them to categories of coalgebras, under the condition that what we call the step is an iso. In our case, it usually is not an iso, since the behaviour functor is a composite *TB* or *BT*; however, it remains interesting to study cases in which such adjunction liftings appear, as used for instance in the aforementioned paper and [29, 42]. Further, our treatment in Section 3 (Eilenberg–Moore) assumes a monad to construct the corecursive algebra, but it was shown by Bartels [1] that this algebra is also corecursive when the underlying category has countable coproducts (and dropping the monad assumption). We currently do not know whether this fits our abstract approach. Finally, the Kleisli/logic and Kleisli/Eilenberg–Moore comparisons (Section 7) are similar, but the Eilenberg–Moore/logic comparison seems different. So far we have been unable to derive a general perspective on such comparisons that covers all three.

A further direction of research is provided by the recent [10], where graded monads are used to define trace semantics in a general way, together with associated expressive logics. On the one hand it would be interesting to try and capture this within our steps-and-adjunctions framework; but this would require to capture graded semantics via some notion of finality or corecursiveness, which we are not currently aware of. On the other hand, as pointed out by one of the reviewers, the comparison results of Section 7 can be viewed as expressivity results of one semantics w.r.t. the other—this insight is interesting on its own, and might help in devising a more general method for making comparisons as in Section 7, also discussed above. Further, the expressiveness criteria in [10] may be useful to address the issues in the comparison of Kleisli semantics to Eilenberg–Moore and logical trace semantics. We leave these considerations for future work.

Funding

The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement nr. 320571.

Acknowledgements

We are grateful to the anonymous referees of both the conference version and this extended paper for various comments and suggestions.

References

[1] F. Bartels. Generalised coinduction. *Mathematical Structures in Computer Science*, **13**, 321–348, 2003.

[2] F. Bonchi, A. Silva and A. Sokolova. The power of convex algebras. In *CONCUR*, pp. 23:1–23:18. Vol. 85 of *LIPIcs*. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2017.

[3] M. M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *FoSSaCS*, pp. 455–469. Vol. 3441 of *Lect. Notes Comp. Sci.* Springer, 2005.

[4] M. M. Bonsangue, S. Milius and A. Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *ACM Transactions on Computational Logic*, **14**, 7:1–7:52, 2013.

[5] N. J. Bowler, P. B. Levy and G. D. Plotkin. Initial algebras and final coalgebras consisting of nondeterministic finite trace strategies. In *MFPS*, pp. 23–44. Vol. 341 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, 2018.

[6] V. Capretta, T. Uustalu and V. Vene. Recursive coalgebras from comonads. *Information and Computation*, **204**, 437–468, 2006.

[7] V. Capretta, T. Uustalu and V. V. C. algebras. A study of general structured corecursion. In *SBMF*, pp. 84–100. Vol. 5902 of *Lect. Notes Comp. Sci.* Springer, 2009.

[8] L.-T. Chen and A. Jung. On a categorical framework for coalgebraic modal logic. *Electronic Notes in Theoretical Computer Science*, **308**, 109–128, 2014.

[9] C. Cîrstea. A coalgebraic approach to quantitative linear time logics. CoRR, abs/1612.07844, 2016. https://arxiv.org/abs/1612.07844v1.

[10] U. Dorsch, S. Milius and L. Schröder. Graded monads and graded logics for the linear time—branching time spectrum. In *CONCUR*, pp. 36:1–36:16. Vol. 140 of *LIPIcs*. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2019.

[11] A. Eppendahl. Coalgebra-to-algebra morphisms. *Electronic Notes in Theoretical Computer Science*, **29**, 42–49, 1999.

[12] J. Y. Girard, Y. Lafont and P. Taylor. *Proofs and Types*. Vol. 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1988.

[13] S. Goncharov. Trace semantics via generic observations. In *CALCO*, pp. 158–174. Vol. 8089 of *Lect. Notes Comp. Sci.* Springer, 2013.

[14] P. Hancock and P. Hyvernat. Programming interfaces and basic topology. *Annals of Pure and Applied Logic*, **137**, 189–239, 05 2009.

[15] I. Hasuo. Generic weakest precondition semantics from monads enriched with order. *Theoretical Computer Science*, **604**, 2–29, 2015.

[16] I. Hasuo, B. Jacobs and A. Sokolova. Generic trace semantics via coinduction. *Logical Methods in Computer Science*, **3**, 2007.

[17] W. Hino, H. Kobayashi, I. Hasuo and B. Jacobs. Healthiness from duality. In *LICS*. IEEE, Computer Science Press, 2016.

[18] R. Hinze, N. Wu and J. Gibbons. Conjugate hylomorphisms—or: the mother of all structured recursion schemes. In *POPL*, pp. 527–538. ACM, 2015.

[19] B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Essays Dedicated to Joseph A. Goguen*, pp. 375–404. Vol. 4060 of *Lect. Notes Comp. Sci.* Springer, 2006.

[20] B. Jacobs. A recipe for state and effect triangles. *Logical Methods in Computer Science*, **13**, 2017.

[21] B. Jacobs, P. Levy and J. Rot. Steps and traces. In *CMCS*, pp. 122–143. Vol. 11202 of *Lect. Notes Comp. Sci.* Springer, 2018.

[22] B. Jacobs, A. Silva and A. Sokolova. Trace semantics via determinization. *Journal of Computer and System Sciences*, **81**, 859–879, 2015.

[23] B. Jacobs and A. Sokolova. Exemplaric expressivity of modal logics. *Journal of Logic and Computation*, **20**, 1041–1068, 2010.

[24] R. Jagadeesan, C. Pitcher and J. Riely. Open bisimulation for aspects. In *AOSD*, pp. 107–120. Vol. 208 of *ACM International Conference Proceeding Series*. ACM, 2007.

[25] P. Johnstone. Adjoint lifting theorems for categories of algebras. *Bulletin of the London Mathematical Society*, **7**, 294–297, 1975.

[26] G. M. Kelly and R. Street. Review of the elements of 2-categories. In *Category Seminar: Proceedings Sydney Category Seminar 1972/1973*, G. M. Kelly, ed. Vol. 420 in Lecture Notes in Mathematics. Springer, 1974.

[27] H. Kerstan, B. König and B. Westerbaan. Lifting adjunctions to coalgebras to (re)discover automata constructions. In *CMCS*, pp. 168–188. Vol. 8446 of *Lect. Notes Comp. Sci.* Springer, 2014.

[28] B. Klin. Coalgebraic modal logic beyond sets. In *MFPS*, M. Fiore, ed. Vol. 173 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2007.

[29] B. Klin and J. Rot. Coalgebraic trace semantics via forgetful logics. *Logical Methods in Computer Science*, **12**, 2016.

[30] B. Klin and J. Salamanca. Iterated covariant powerset is not a monad. *Electronic Notes in Theoretical Computer Science*, **341**, 261–276, 2018.

[31] A. Kurz, S. Milius, D. Pattinson and L. Schröder. Simplified coalgebraic trace equivalence. In *Software, Services, and Systems*, pp. 75–90. Vol. 8950 of *Lect. Notes Comp. Sci.* Springer, 2015.

[32] J. Laird. A fully abstract trace semantics for general references. In *ICALP*, pp. 667–679. Vol. 4596 of Lect. Notes Comp. Sci. Springer, 2007.

[33] S. B. Lassen and P. B. Levy. Typed normal form bisimulation. In *CSL*, pp. 283–297. Vol. 4646 of *Lect. Notes Comp. Sci.* Springer, 2007.

[34] T. Leinster. *Higher Operads, Higher Categories*. Vol. 298 of *London Mathematical Society Lecture Notes*. Cambridge University Press, 2004.

[35] P. B. Levy. Final coalgebras from corecursive algebras. In *CALCO*, pp. 221–237. Vol. 35 of *LIPIcs*. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2015.

[36] P. B. Levy and S. Staton. Transition systems over games. In *CSL-LICS*, pp. 64:1–64:10. ACM, 2014.

[37] S. Milius. Completely iterative algebras and completely iterative monads. *Information and Computation*, **196**, 1–41, 2005.

[38] S. Milius, D. Pattinson and L. Schröder. Generic trace semantics and graded monads. In *CALCO*, pp. 253–269. Vol. 35 of *LIPIcs*. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2015.

[39] P. S. Mulry. Lifting theorems for Kleisli categories. In *MFPS*, pp. 304–319. Vol. 802 of *Lect. Notes Comp. Sci.* Springer, 1993.

[40] D. Pavlovic, M. W. Mislove and J. Worrell. Testing semantics: connecting processes and process logics. In *AMAST*, pp. 308–322. Vol. 4019 of *Lect. Notes Comp. Sci.* Springer, 2006.

[41] A. W. Roscoe, S. D. Brookes and C. A. R. Hoare. A theory of communicating sequential processes. *Journal of the ACM*, **31**, 560–599, 1984.

[42] J. Rot. Coalgebraic minimization of automata by initiality and finality. *Electronic Notes in Theoretical Computer Science*, **325**, 253–276, 2016.

[43] A. Silva, F. Bonchi, M. M. Bonsangue and J. J. M. M. Rutten. Generalizing determinization from automata to coalgebras. *Logical Methods in Computer Science*, **9**, 2013.

[44] R. Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 149–168, 1972.

[45] D. Turi and G. D. Plotkin. Towards a mathematical operational semantics. In *LICS*, pp. 280–291. IEEE Computer Society, 1997.

A Details for Section 2

We recall a (standard) lemma that relates a step ρ_1 to its mate ρ_2.

LEMMA A.1
For any step $\rho\colon HG \Rightarrow GL$, the following diagrams commute.

$$\begin{tikzcd}
FHG \arrow[r, "F\rho_1"] \arrow[d, "\rho_2 G"'] & FGL \arrow[d, "\varepsilon L"] \\
LFG \arrow[r, "L\varepsilon"'] & L
\end{tikzcd}
\qquad
\begin{tikzcd}
H \arrow[r, "H\eta"] \arrow[d, "\eta H"'] & HGF \arrow[d, "\rho_1 F"] \\
GFH \arrow[r, "G\rho_2"'] & GLF
\end{tikzcd}$$

PROOF. By unpacking the definitions and using naturality, e.g. in the first diagram:

$$\begin{aligned}
L\varepsilon \circ \rho_2 G &= L\varepsilon \circ \varepsilon LFG \circ F\rho_1 FG \circ FH\eta G \\
&= \varepsilon L \circ FGL\varepsilon \circ F\rho_1 FG \circ FH\eta G \\
&= \varepsilon L \circ F\rho_1 \circ FHG\varepsilon \circ FH\eta G \\
&= \varepsilon L \circ F\rho_1 .
\end{aligned}$$

□

The above lemma is useful in the proofs of Theorem 2.3 and Lemma 2.7, presented next.

PROOF OF THEOREM 2.3 The correspondence between ρ_1, ρ_2, ρ_3 and ρ_4 follows from Theorem 2.2. For the second part of the statement, suppose H and L have monad structures (H, η^H, μ^H) and (L, η^L, μ^L), respectively. The fact that ρ_1 is an $\mathcal{EM}$-law iff ρ_4 is a monad map can be reconstructed from [44].

We show that if ρ_1 is an $\mathcal{EM}$-law then ρ_2 is a Kleisli law—the converse follows analogously. To this end, for the compatibility of ρ_2 with μ, consider the following diagram.

$$\begin{tikzcd}
FHH \arrow[rr, "F\mu^H"] \arrow[d, "\rho_2 H"] \arrow[dr, "FHH\eta"] & & FH \arrow[d, "FH\eta"] \arrow[dddd, bend left, "\rho_2"] \\
LFH \arrow[d, "LFH\eta"] \arrow[ddd, bend right, "L\rho_2"'] & FHHGF \arrow[r, "F\mu^H LF"] \arrow[dl, "\rho_2 HGF"] \arrow[d, "FH\rho_1 F"] & FHGF \arrow[dd, "F\rho_1 F"] \\
LFHGF \arrow[d, "LF\rho_1 F"] & FHGLF \arrow[dl, "\rho_2 GLF"] \arrow[d, "F\rho_1 LF"] & \\
LFGLF \arrow[d, "L\varepsilon LF"] & FGLLF \arrow[r, "FG\mu^L F"] \arrow[dl, "\varepsilon LLF"] & FGLF \arrow[d, "\varepsilon LF"] \\
LLF \arrow[rr, "\mu^L F"] & & LF
\end{tikzcd}$$

The outside shapes commute by definition of ρ_2. For the (inner) rectangle, everything commutes, clockwise starting at the north by naturality, the fact that ρ_1 is an $\mathcal{EM}$-law, naturality, Lemma A.1, and twice naturality. For the unit axiom, we have the following diagram:

$$\begin{tikzcd}
FH \arrow[r, "FH\eta"'] \arrow[rrr, bend left, "\rho_2"] & FHGF \arrow[r, "F\rho_1 F"'] & FGLF \arrow[r, "\varepsilon LF"'] & LF \\
F \arrow[u, "F\eta^H"] \arrow[r, "F\eta"] \arrow[rrr, bend right, "\mathrm{id}"'] & FGF \arrow[u, "F\eta^H GF"'] \arrow[r, equal] & FGF \arrow[u, "FG\eta^L F"'] \arrow[r, "\varepsilon F"] & F \arrow[u, "\eta^L F"']
\end{tikzcd}$$

which commutes by naturality, a triangle identity of the adjunction, definition of ρ_2 and the fact that ρ_1 is an $\mathcal{EM}$-law (middle shape).

Finally, the correspondence between $\mathcal{EM}$-laws and liftings was shown in [25] and the variant for Kleisli laws in [39]. □

Proof of Lemma 2.7 First, note that $\left(\theta \circledcirc \rho\right)_2 = \varepsilon MLF \circ F\theta_1 LF \circ FK\rho_1 F \circ FKH\eta$. It thus suffices to prove that the following diagram commutes.

$$\begin{tikzcd}
FKH \arrow[rr, "FKH\eta"] \arrow[d, "\theta_2 H"'] & & FKHGF \arrow[dl, "\theta_2 HGF"'] \arrow[d, "FK\rho_1 F"] \\
MFH \arrow[r, "MFH\eta"] \arrow[dd, "M\rho_2"'] & MFHGF \arrow[d, "MF\rho_1 F"'] & FKGLF \arrow[dl, "\theta_2 GLF"'] \arrow[dd, "F\theta_1 LF"] \\
& MFGLF \arrow[dl, "M\varepsilon LF"'] & \\
MLF & & FGMLF \arrow[ll, "\varepsilon MLF"]
\end{tikzcd}$$

All the inner parts commute, clockwise starting from the top by naturality of θ_2 (twice), Lemma A.1, and definition of ρ_2 from ρ_1. □

B Steps and bimodules

For the example of partial traces for I/O given in Section 6, it is convenient to take a different view of our step-and-adjunction setting, using the following notion.

Definition B.1
For categories $\mathbf{C}$ and $\mathbf{D}$, a *bimodule* $\mathbf{O}\colon \mathbf{C} \nrightarrow \mathbf{D}$ consists of the following data.

- A family of sets $(\mathbf{O}(X,Y))_{X\in\mathbf{C},Y\in\mathbf{D}}$, where $g \in \mathbf{O}(X,Y)$ is called an $\mathbf{O}$-*morphism* $g\colon X \to Y$.
- Each $g\colon X \to Y$ can be composed with a $\mathbf{C}$-map $f\colon X' \to X$ or $\mathbf{D}$-map $h\colon Y \to Y'$.

For $g\colon X \to Y$, we must have the following. (We use semicolon for diagrammatic-order composition.)

$$\begin{aligned}
\mathrm{id}_X; g &= g \\
(f';f); g &= f'; (f; g) \\
g; \mathrm{id}_Y &= g \\
g; (h; h') &= (g; h); h' \\
(f; g); h &= f; (g; h)
\end{aligned}$$

For example, for an endofunctor H on $\mathbf{C}$, the coalgebra-to-algebra morphisms constitute a bimodule $\mathrm{CoAlg}(H) \nrightarrow \mathrm{Alg}(H)$. Bimodules $\mathbf{C} \nrightarrow \mathbf{D}$ correspond to functors $\mathbf{C}^{\mathrm{op}} \times \mathbf{D} \to \mathbf{Sets}$ and are also called *distributors* or *profunctors* (but some authors reverse the direction).

Definition B.2

1. A *map* of bimodules

$$\begin{tikzcd}
\mathbf{C} \arrow[r, bend left, "O", "|"{marking}] \arrow[r, bend right, "O'"', "|"{marking}] \arrow[r, phantom, "R\Downarrow"] & \mathbf{D}
\end{tikzcd}$$

   sends each **O**-morphism $g\colon X \to Y$ to an $\mathbf{O}'$-morphism $Rg\colon X \to Y$ with the following commuting:

$$\begin{tikzcd}
X' \arrow[d, "f"'] \arrow[dr, "R(f;g)"] & \\
X \arrow[r, "Rg"'] & Y
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "Rg"] \arrow[dr, "R(g;h)"'] & Y \arrow[d, "h"] \\
& Y'
\end{tikzcd}$$

2. More generally, a *2-cell*

$$\begin{tikzcd}
\mathbf{C} \arrow[r, "O", "|"{marking}] \arrow[d, "H"'] \arrow[dr, phantom, "R\Downarrow"] & \mathbf{D} \arrow[d, "L"] \\
\mathbf{C}' \arrow[r, "O'"', "|"{marking}] & \mathbf{D}'
\end{tikzcd}$$

   sends each **O**-morphism $g\colon X \to Y$ to an $\mathbf{O}'$-morphism $Rg\colon HX \to LY$ with the following commuting:

$$\begin{tikzcd}
HX' \arrow[d, "Hf"'] \arrow[dr, "R(f;g)"] & \\
HX \arrow[r, "Rg"'] & LY
\end{tikzcd}
\qquad
\begin{tikzcd}
HX \arrow[r, "Rg"] \arrow[dr, "R(g;h)"'] & LY \arrow[d, "Lh"] \\
& LY'
\end{tikzcd}$$

The product construction on categories extends to bimodules:

Definition B.3
Let $(\mathbf{O}_j\colon \mathbf{C}_j \nrightarrow \mathbf{D}_j)_{j\in J}$ be a family of bimodules. Then the bimodule $\prod_{j\in J}\mathbf{O}_j\colon \prod_{j\in J}\mathbf{C}_j \nrightarrow \prod_{j\in J}\mathbf{D}_j$ is defined by saying that a morphism $(X_j)_{j\in J} \to (Y_j)_{j\in J}$ is a family $(g_j\colon X_j \to Y_j)_{j\in J}$, where g_j is an $\mathbf{O}_j$-morphism for all $j \in J$. Composition is defined componentwise.

Of course this construction extends also to maps and 2-cells between bimodules.
Here are two ways of constructing a bimodule $\mathbf{C} \nrightarrow \mathbf{D}$.

Definition B.4

1. A functor $F\colon \mathbf{C} \to \mathbf{D}$ gives $F^{\mathsf{Left}}\colon \mathbf{C} \nrightarrow \mathbf{D}$, where $F^{\mathsf{Left}}(X, Y) \stackrel{\text{def}}{=} \mathbf{D}(FX, Y)$.
2. A functor $G\colon \mathbf{D} \to \mathbf{C}$ gives $G^{\mathsf{Right}}\colon \mathbf{C} \nrightarrow \mathbf{D}$, where $G^{\mathsf{Right}}(X, Y) \stackrel{\text{def}}{=} \mathbf{C}(X, GY)$.

Definition B.5
For a bimodule $\mathbf{O}\colon \mathbf{C} \nrightarrow \mathbf{D}$,

- a *left representation* consists of a functor $F\colon \mathbf{C} \to \mathbf{D}$ and an isomorphism $m : \mathbf{O} \cong F^{\mathsf{Left}}$
- a *right representation* consists of a functor $G\colon \mathbf{C} \to \mathbf{D}$ and an isomorphism $n : \mathbf{O} \cong G^{\mathsf{Right}}$.

Note that an adjunction

$$\begin{tikzcd}
\mathbf{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{D} \arrow[l, bend left, "G"]
\end{tikzcd}$$

may be viewed as a bimodule isomorphism $F^{\mathsf{Left}} \cong G^{\mathsf{Right}}$. Conversely, a bimodule $\mathbf{C} \nrightarrow \mathbf{D}$ equipped with both a left and a right representation constitutes an adjunction.

The natural transformations in Theorem 2.2 correspond to 2-cells of bimodules, as follows.

Theorem B.6
Suppose we have left representations $m\colon \mathbf{O} \cong F^{\mathsf{Left}}$ and $m'\colon \mathbf{O}' \cong F'^{\mathsf{Left}}$. Then a 2-cell

$$\begin{tikzcd}
\mathbf{C} \arrow[r, "O", "\mid" marking] \arrow[d, "H"'] \arrow[dr, phantom, "R \Downarrow"] & \mathbf{D} \arrow[d, "L"] \\
\mathbf{C}' \arrow[r, "O'"', "\mid" marking] & \mathbf{D}'
\end{tikzcd}$$

corresponds to a natural transformation

$$\begin{tikzcd}
\mathbf{C} \arrow[r, "F"] \arrow[d, "H"'] \arrow[dr, phantom, "\overset{\rho_2}{\Rightarrow}"] & \mathbf{D} \arrow[d, "L"] \\
\mathbf{C}' \arrow[r, "F'"'] & \mathbf{D}'
\end{tikzcd}$$

where R sends an $\mathbf{O}$-morphism $g\colon X \to Y$ to $\left(F'HX \xrightarrow{\rho_2} LFX \xrightarrow{L_{m(g)}} LY\right)$. The analogous statements hold for ρ_1, ρ_3 and ρ_4.

Now we give a more refined account of steps. Suppose we have a bimodule, two endofunctors and a 2-cell:

$$\begin{tikzcd}
\mathbf{C} \arrow[r, "O", "|"{marking}] \arrow[d, "H"'] \arrow[dr, phantom, "R\Downarrow"] & \mathbf{D} \arrow[d, "L"] \\
\mathbf{C} \arrow[r, "O"', "|"{marking}] & \mathbf{D}
\end{tikzcd}$$

We call R a 'step'. Given a left representation $m\colon \mathbf{O} \cong F^{\mathsf{Left}}$ we have ρ_2 and the functor F^{ρ}. Given a right representation $n\colon \mathbf{O} \cong G^{\mathsf{Right}}$, we have ρ_1 and the functor G_{ρ}.

DEFINITION B.7

1. A *coalgebra morphism* from an H-coalgebra $c\colon X \to H(X)$ to an L-coalgebra $d\colon \Theta \to L(\Theta)$ is an $\mathbf{O}$-morphism $g\colon X \to \Theta$ such that the following commutes:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "c"'] & \Theta \arrow[d, "d"] \\
H(X) \arrow[r, "R(f)"'] & L(\Theta)
\end{tikzcd}$$

   This gives a bimodule $\mathrm{CoAlg}(H) \nrightarrow \mathrm{CoAlg}(L)$.
2. A *coalgebra-to-algebra morphism* from an H-coalgebra $c\colon X \to H(X)$ to an L-algebra $a\colon L(\Theta) \Rightarrow \Theta$ is an $\mathbf{O}$-morphism $g\colon X \to \Theta$ such that the following commutes:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "c"'] & \Theta \\
H(X) \arrow[r, "R(f)"'] & L(\Theta) \arrow[u, "a"']
\end{tikzcd}$$

   Equivalently: such a morphism is a fixpoint for the endofunction on the homset $\mathbf{C}(X,\Theta)$ sending f to the composite $X \xrightarrow{c} H(X) \xrightarrow{R(f)} L\ (\theta) \xrightarrow{a} \theta$. This gives a bimodule $\mathrm{CoAlg}(H) \nrightarrow \mathrm{Alg}(L)$.

DEFINITION B.8

1. A final coalgebra $d\colon \Theta \Rightarrow L(\Theta)$ is said to *extend across* $\mathbf{O}$ when from each H-coalgebra $c\colon X \to H(X)$ there is a unique morphism to (Θ, d).
2. A corecursive algebra $a\colon L(\Theta) \Rightarrow \Theta$ is said to *extend across* $\mathbf{O}$ when from each H-coalgebra $c\colon X \to H(X)$ there is a unique morphism to (Θ, a).

Now let us decompose Proposition 2.5 into two parts.

PROPOSITION B.9

1. Let $\mathbf{O}$ have a left representation $m\colon \mathbf{O} \cong F^{\mathsf{Left}}$. Then any corecursive L-algebra (Θ, a) extends across $\mathbf{O}$. (And hence also any final L-algebra.) Explicitly, the map $(X, c) \to (\Theta, a)$ is m^{-1} applied to the map $F^{\rho}(X, c) \to (\Theta, a)$.
2. Let $\mathbf{O}$ have a right representation $n\colon \mathbf{O} \cong G^{\mathsf{Right}}$. Then any corecursive L-algebra (Θ, a) extending across $\mathbf{O}$ is sent by G_{ρ} to a corecursive H-algebra. Explicitly, the map $(X, c) \to G_{\rho}(\Theta, a)$ is n applied to the map $(X, c) \to (\Theta, a)$.

Note that this story also appears, in contravariant form, in [35, Propositions 16–17].

Received 27 January 2019