Steps and Traces[⋆]

Jurriaan Rot[1] and Bart Jacobs[1] and Paul Levy[2]

[1] Institute for Computing and Information Sciences,
Radboud Universiteit, Nijmegen, The Netherlands
[2] University of Birmingham, UK
`jrot@cs.ru.nl` and `bart@cs.ru.nl` and `P.B.Levy@cs.bham.ac.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: 1) coalgebraic trace semantics is naturally presented in terms of corecursive algebras, and 2) 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.

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.

For many years already, 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, formalised 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 differs from earlier work which

---

[⋆] 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. This is a revised and extended version of a paper which appeared in the proceedings of CMCS 2018 [19].

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 be 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, 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 [31,25,9].

Outline. The paper is organised 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 A.

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 briefly show that our construction of corecursive algebras strengthens to a construction of completely iterative algebras. 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 [10,7] as the dual of Taylor's notion of recursive coalgebra [11], is defined as follows.

Definition 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 [15], 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 $\zeta\colon \Theta \xrightarrow{\cong} H(\Theta)$, 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* [21,28].

Theorem 2. *Given adjunctions and functors*

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

there are bijective correspondences between natural transformations:

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

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:

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

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:

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

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 [3,33,23,8,29], but we shall see that its application is wider.

Theorem 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.*

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 characterisations are respectively a lifting of G or an extension of F:

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

Definition 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*:*

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

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

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

Our approach relies on the following basic result.

Proposition 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 $a\colon L(\Theta) \to \Theta$ be a corecursive L-algebra and $c\colon X \to H(X)$ 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 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:*

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

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 6 generalises 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 $\rho \circledcirc \theta$ in:

$$\theta \circledcirc \rho \coloneqq \Big(KHG \xRightarrow{K\rho} KGL \xRightarrow{\theta L} GML\Big) \tag{2}$$

We conclude with a few (standard) technical lemmas relating a step ρ_1 to its mate ρ_2.

Lemma 7. *For any step $\rho\colon HG \Rightarrow GL$, the following diagrams commute.*

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

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

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

Lemma 8. *Let $\rho\colon HG \Rightarrow GL$, $\theta\colon KG \Rightarrow GM$ be (composable) steps. Then $\big(\theta \circledcirc \rho\big)_2 = M\rho_2 \circ \theta_2 H$.*

Proof. First, note that $\big(\theta \circledcirc \rho\big)_2 = \varepsilon MLF \circ F\theta_1 LF \circ FK\rho_1 F \circ FKH\eta$. 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 7, and definition of ρ_2 from ρ_1. □

3 Traces via Eilenberg-Moore

We recall the approach to trace semantics developed in [18,36,4], 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.

1. An endofunctor $B\colon \mathbf{C} \to \mathbf{C}$ with a final coalgebra $\zeta\colon \Theta \xrightarrow{\cong} B(\Theta)$.
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 9. 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 semi-lattices (JSLs). The lifting is defined by products in $\mathcal{EM}(T)$, using the JSL on 2 given by the usual ordering $\bot \leq \top$. 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):

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

Actually — and equivalently, by Theorem 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 [37].

Lemma 10. *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* $\mathrm{CoAlg}(\overline{B})$. $\square$

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 $\zeta\colon (\Theta, a) \to \overline{B}(\Theta, a)$ which is final. $\square$

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:

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

By Theorem 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.

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

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 11, Example 12).

In [18,36], the above trace semantics of BT-coalgebras arises through 'determinisation', 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 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 determinisation, see [18]. By Theorem 6, the trace semantics em_c can equivalently be characterised in terms of $\mathcal{F}^\rho$, as the unique map $\overline{\mathsf{em}_c}$ making the diagram below commute.

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

This is how the trace semantics via Eilenberg-Moore is presented in [18,36]: 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 [16] for a class of 'automata-like' examples.

Theorem 11. *Let Ω be a set, T a monad on* **Sets** *and $t\colon T(\Omega) \to \Omega$ an $\mathcal{EM}$-algebra. Let $B\colon$* **Sets** *$\to$* **Sets***, $B(X) = \Omega \times X^A$, and $\kappa\colon TB \Rightarrow BT$ given by*

$$\kappa_X \coloneqq \Big(T(\Omega \times X^A) \xrightarrow{\langle T(\pi_1), T(\pi_2)\rangle} T(\Omega) \times T(X^A) \xrightarrow{\quad t \times \mathrm{st} \quad} \Omega \times T(X)^A \Big).$$

Then κ *is an* $\mathcal{EM}$*-law. Moreover, the final* B*-coalgebra* (Ω^{A^*}, ζ) *together with the algebra structure* $T(\Omega^{A^*}) \xrightarrow{\mathsf{st}} T(\Omega)^{A^*} \xrightarrow{t^{A^*}} \Omega^{A^*}$ *is a final* $\overline{B}$*-coalgebra.* □

Example 12. By Theorem 11, we obtain an explicit description of the trace semantics arising from the corecursive algebra (7): for any $\langle o, f\rangle \colon X \to \Omega \times T(X)^A$, the trace semantics is the unique map em in:

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

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 determinisation constructions, see [18,36].

3.1 Generalised Eilenberg-Moore trace semantics

We generalise the above treatment of trace semantics via Eilenberg-Moore categories, to a situation where we have two functors A, B on the base category together with a step from one of them to the lifting of the other. This generalisation is inspired by the logical approach to trace semantics (Section 4), which is based on composition of steps. In particular, this allows us to treat the *extension semantics* of [18].

We extend the assumptions in the beginning of this section with an endofunctor $A\colon \mathbf{C} \to \mathbf{C}$, and a step ρ in:

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

Notice that applying the forgetful functor to the counit $\varepsilon_{(X,a)} = a\colon (T(X), \mu) = \mathcal{F}U(X,a) \to (X,a)$ 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:

$$\rho \circledcirc U\varepsilon \;=\; \Big(ATU \xRightarrow{AU\varepsilon} AU \xRightarrow{\rho} U\overline{B}\Big)$$

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:

$$U\varepsilon \circledcirc \rho \;=\; \Big(TAU \xRightarrow{T\rho} TU\overline{B} \xRightarrow{U\varepsilon\overline{B}} U\overline{B}\Big)$$

As we will see in Subsection 7.3, this coincides with the so-called *extension* natural transformations of [18]. The latter is a natural transformation $\mathfrak{e}\colon TA \Rightarrow BT$ making two diagrams commute, one of which is:

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

In [18] a further coherence axiom with a $\mathcal{K}\ell$-law is assumed, but we will only see this later, in our comparison between different approaches for assigning trace semantics, see Subsection 7.3.

Proposition 13. *There is a one-to-one correspondence between:*

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

Proof. By Theorem 3 the natural transformation $\rho = \rho_1\colon AU \Rightarrow U\overline{B}$ corresponds to $\mathfrak{e} = \rho_2\colon \mathcal{F}A \Rightarrow \overline{B}\mathcal{F}$; 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 3, a natural transformation $\mathfrak{e}\colon TA \Rightarrow BT$ 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. $\square$

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 10. 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 14. *We have* $\ell^A_{\mathrm{em}} = \ell_{\mathrm{em}} \circ U(\rho_2) \colon TA(\Theta) \to \Theta$. *Explicitly:*

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

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

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

The inner rectangle below commutes by Lemma 7. □

Example 15. We illustrate the situation with a simple example: non-deterministic automata, viewed as coalgebras of the form $f \colon X \to \mathcal{P}_{\mathrm{f}}(\Sigma \times X + 1)$. 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 9 and Theorem 11. 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 $\rho \colon \mathcal{F}A \Rightarrow \overline{B}\mathcal{F}$ 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

$$(\rho_4)_X(a,x) = \left(0, \lambda b. \begin{cases} \{x\} & \text{if } a = b \\ \emptyset & \text{otherwise} \end{cases}\right), \quad (\rho_4)_X(*) = (1, \lambda b.\emptyset)\,.$$

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

$$\rho_2(S) = \left(\bigvee_{* \in S} 1, \lambda a.\{x \mid (a,x) \in S\}\right)$$

This coincides with the extension law given in [18].

By Lemma 14, 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:

$$\ell^A_{\mathrm{em}}(S)(\varepsilon) = \bigvee_{*\in S} 1\,, \qquad \ell^A_{\mathrm{em}}(S)(aw) = \bigvee_{(a,\psi)\in S} \psi(w)\,.$$

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

For examples of extension laws for weighted and probabilistic automata, see [18].

4 Traces via Logic

This section illustrates how the 'logical' approach to trace semantics of [24], started in [33], fits in our general framework. In essence, traces are built up from 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.

1. An adjunction $F \dashv G$ between categories $\mathbf{C} \leftrightarrows \mathbf{D}^{\mathrm{op}}$.
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 $\alpha\colon L(\Phi) \xrightarrow{\cong} \Phi$.

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 3; here we use 'delta' instead of 'rho' notation since it is common in modal logic.

Example 16. 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 $L(X) = A \times X + 1$. 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 map ρ 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:

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

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:

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

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 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 $\mathbf{D}^{\mathrm{op}}$:

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

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

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

It is instructive to characterise 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:

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

In the remainder of this section, we show two classes of examples of the logical trace semantics. With these descriptions we retrieve most of the examples from [24] in a smooth manner.

Proposition 17. *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

$$\tau_X := \left(T(\Omega^X) \xrightarrow{\text{st}} T(\Omega)^X \xrightarrow{t^X} \Omega^X \right).$$

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

$$\Omega \times T(\Omega^{A^*})^A \xrightarrow{\text{id}\times(\text{st})^A} \Omega \times (T(\Omega)^{A^*})^A \xrightarrow{\text{id}\times(t^{A^*})^A} \Omega \times (\Omega^{A^*})^A \xrightarrow{\Omega^{\alpha^{-1}}\circ\delta} \Omega^{A^*} \quad .$$

The concrete description of log follows by spelling out the coalgebra-to-algebra diagram that characterises it. □

Example 18. We instantiate the trace semantics log from Proposition 17 for various choices of Ω, T and t. Similar to the instances in Example 12, 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 12. However, in constrast 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 $T = \mathcal{P}_{\text{f}}\mathcal{P}_{\text{f}}$; it is unclear whether T carries a suitable monad structure in that case.

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

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 $H_\Sigma(X) = \coprod_{n\in\mathbb{N}} \Sigma(n) \times X^n$. We abuse notation and write $\sigma(x_1, \ldots, x_n)$ instead of $(\sigma, x_1, \ldots x_n)$. The initial algebra of H_Σ consists of closed terms (or finite node-labelled trees) over the signature.

Proposition 19. *Let Ω be a meet semi-lattice 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 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*

$$\mathsf{log}(x)(\sigma(t_1, \ldots, t_n)) = t(T(m) \circ c(x))\,, \textit{ where}$$

$$m = \left(t \mapsto \begin{cases} \bigwedge_i \mathsf{log}(x_i)(t_i) & \textit{if } \exists x_1 \ldots x_n.\, t = \sigma(x_1, \ldots, x_n) \\ \bot & \textit{otherwise} \end{cases} \right) : H_\Sigma(X) \to \Omega$$

for all $x \in X$ *and* $\sigma(t_1, \ldots, t_n) \in \Phi$.

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 17. The logic $\delta \colon H_\Sigma \Omega^- \Rightarrow \Omega^{H_\Sigma(-)}$ is:

$$\delta_X(\sigma(\phi_1, \ldots, \phi_n))(t) = \begin{cases} \bigwedge_i \phi_i(x_i) & \text{if } \exists x_1 \ldots x_n . \, t = \sigma(x_1, \ldots, x_n) \\ \bot & \text{otherwise} \end{cases}$$

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

$$TH_\Sigma(\Omega^\Phi) \xrightarrow{T(\delta)} T(\Omega^{H_\Sigma(\Phi)}) \xrightarrow{\text{st}} T(\Omega)^{H_\Sigma(\Phi)} \xrightarrow{t^{H_\Sigma(\Phi)}} \Omega^{H_\Sigma(\Phi)} \xrightarrow{\cong} \Omega^\Phi \ .$$

The explicit characterisation of log is a straightforward computation. □

Example 20. Given a signature Σ, a coalgebra $f \colon X \to \mathcal{P}_{\mathrm{f}} H_\Sigma(X)$ is a (top-down) *tree automaton.* With $\Omega = \{\bot, \top\}$ and $t(S) = \bigvee S$, Proposition 19 gives:

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

for every state $x \in X$ and tree $\sigma(t_1, \ldots, t_n)$. This is the standard semantics of tree automata. It is easily adapted to *weighted* tree automata, see [24].

In both Example 20 and Example 18, the step-induced coalgebra lifting $F_{\delta \circledcirc \tau}$ (respectively $F_{\tau \circledcirc \delta}$) of the underlying logic corresponds to reverse determinisation, see [24,35] for details. In particular, in Example 20 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 [13], 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 required.

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

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

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

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

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 [13]. 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):

$$\begin{tikzcd} \mathbf{C} \arrow[loop left, "TA"] \arrow[r, bend left, "J"] \arrow[r, phantom, "\perp"] & \mathcal{K}\ell(T) \arrow[l, bend left, "U"] \arrow[loop right, "\overline{A}"] \end{tikzcd} \quad \text{with} \quad \begin{array}{l} \rho\colon TAU \Longrightarrow U\overline{A} \ \text{ where } \rho_X = \\ \big(TATX \xrightarrow{T(\lambda)} T^2AX \xrightarrow{\mu^T} TAX\big) \end{array}$$

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

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

By Theorem 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 [13], to which we refer for examples. For later use we note the following.

Lemma 21. *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:

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

6 Partial Traces for Input/Output

Both automata theory and semantics are concerned with coalgebras, but there are differences. One is that automata are usually *finite* coalgebras, whereas coalgebras formed from a calculus or programming language are usually infinite. More importantly, the word "trace" has a rather different meaning.

- In the automata literature, the "traces" studied end in acceptance. Semanticists might call these "complete traces".
- Conversely, in the semantics literature, such as CSP [34] or game semantics [20,26,27,30], the "traces" studied have the property that any prefix of a trace is also a trace. Automata theorists might call these "partial traces".

So far in the examples we have looked at traces in the former sense. This section elaborates and example in the latter sense, which were studied coalgebraically in [5]. To this end, it is helpful to recast our step-and-adjunction setting in terms of *bimodules*; this slightly different viewpoint is set out in Appendix A.

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. Accordingly, we use the functor:

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

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 $i_r \in \mathsf{Ar}(k_r)$. 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 $s \in \sigma$.

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 form a strategy. (Note that active-ending traces need not be considered, since these are determined by the passive-ending traces. Infinite traces are not treated in [5], nor are they here.) Conversely, every strategy can be obtained in this way [5, Proposition 6.1].

The coalgebraic description of the traces uses the following categories.

- **CSL** is the category of *complete semilattices*, i.e., posets with all suprema, and monotone functions that preserve all suprema. The forgetful functor $U\colon \mathbf{CSL} \to \mathbf{Sets}$ is monadic, where the free complete semilattice on a set X is $\mathcal{P}X$, ordered by inclusion, with unit $X \to \mathcal{P}X$ sending $x \mapsto \{x\}$.

- **ACSL** is the category of *almost complete semilattices*, i.e., posets where every nonempty subset has a supremum, and monotone functions that preserve these suprema. The forgetful functor $U\colon \mathbf{ACSL} \to \mathbf{Sets}$ is monadic, where the free almost complete semilattice on a set X is the set $\mathcal{P}^{+}X$ of nonempty subsets, ordered by inclusion, with unit $X \to \mathcal{P}^{+}X$ sending $x \mapsto \{x\}$.

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.

- For a family $(B_j)_{j\in J}$ of complete semilattice lattices, $\prod_{j\in J} B_j$ is the cartesian product, with pointwise order.
- For a family $(A_j)_{j\in J}$ of almost complete semilattices, $\bigoplus_{j\in J}^{\perp} A_j$ is the set of pairs $(U,(a_j)_{j\in U})$ where $U \in \mathcal{P}J$ and $\forall j \in J.\, a_j \in A_j$, with ordering $(U,(a_j)_{j\in U}) \leqslant (V,(b_j)_{j\in V})$ when $U \subseteq V$ and $\forall j \in U.\, a_j \leqslant b_j$.

From these we build the endofunctor

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

which has a final coalgebra described as follows.

Theorem 22. [5, Theorem 6.3] *Let* Strat *be the complete semilattice of strategies, ordered by inclusion. 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* $\bigoplus_{k\in K}^{\perp} \prod_{i\in\mathsf{Ar}(k)}$*-coalgebra.*

We want to instantiate our general setting to obtain an account of traces. We take the adjunction and endofunctors

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

and an appropriate step, which we must now construct. To do so, it is convenient to take a somewhat different view of steps in terms of *bimodules*, as explained in Appendix A.

Just as our endofunctors are built out of pieces, so is our step. We first define the following 2-cells, for any set J.

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

They are defined as follows.

- Given a family of functions $(f_j \colon X_j \to B_j)$, 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 $(f x_j)_{j \in J}$.
- Given a family of functions $(f_j \colon X_j \to A_j)$, 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

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

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

$$\begin{tikzcd} \bigoplus^{\perp}_{j \in J} \mathcal{P}^{+}(X_j) \arrow[r, "\cong"] & \mathcal{P}\big(\sum_{j \in J} X_j \big) \\ (U, (Y_j)_{j \in U}) \arrow[r, mapsto] & \{ (j, x) \mid j \in U, x \in Y_j \} \end{tikzcd}$$

We thus obtain a 2-cell

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

which provides our step ρ.

From Theorem 22 with Proposition 5(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). Moreover [5, Theorem 6.6] tells us that this morphism sends $x \in X$ to its set of traces. Finally by Proposition 5(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 determinise a transition system (X, c). This is applied in [5, Section 6.2] to obtain a bisimulation method for trace equivalence.

A similar story can be told for coalgebras $c \colon X \to \prod_{k \in K} \mathcal{P} \sum_{i \in \mathsf{Ar}(k)} X$. In this case, the behaviour of a state is to first input $k \in K$ and then nondeterministically output some $i \in \mathsf{Ar}(k)$, resulting in a new state. The definition of strategy is adjusted accordingly. The step used is as follows.

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

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, by fitting the setting of [18] — which does not use corecursive algebras — in the current framework.

7.1 Eilenberg-Moore and Logic

To compare the Eilenberg-Moore approach with the logical approach, we combine their assumptions. This amounts to 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*).

The step map τ is already 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. 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 and instance of Theorem 3.

Lemma 23. *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 [14,17,12] where $\widehat{F}$ is called Pred. We use $\widehat{\cdot}$ to indicate lifting 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 $\alpha \colon L(\varPhi) \to \varPhi$ in $\mathbf{D}$ forms a final coalgebra $\alpha \colon \varPhi \to L(\varPhi)$ in $\mathbf{D}^{\mathrm{op}}$, and thus a corecursive algebra $\alpha^{-1} \colon L(\varPhi) \to \varPhi$ in $\mathbf{D}^{\mathrm{op}}$. Hence, applying G_ρ to the latter corecursive L-algebra gives a corecursive B-algebra, namely:

$$G_\rho(\varPhi, \alpha^{-1}) \coloneqq \Big(BG(\varPhi) \xrightarrow{\;\delta\;} GL(\varPhi) \xrightarrow{\;G(\alpha^{-1})\;} G(\varPhi) \Big).$$

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

$$\begin{tikzcd}
\varTheta \arrow[rr, "\mathsf{e}"] \arrow[dd, "\zeta"', "\cong"] & & G(\varPhi) \\
 & & GL(\varPhi) \arrow[u, "G(\alpha^{-1})"'] \\
B(\varTheta) \arrow[rr, "B(\mathsf{e})"] & & BG(\varPhi) \arrow[u, "\delta"']
\end{tikzcd} \tag{18}$$

This $\mathsf{e} \colon \varTheta \to G(\varPhi)$ is a morphism from the carrier of the corecursive algebra $\ell_{\mathrm{em}} \colon BT(\varTheta) \to \varTheta$, from the Eilenberg-Moore approach (6), to the carrier of the corecursive algebra $\ell^{\log} \colon BTG(\varPhi) \to G(\varPhi)$, 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 25.

Lemma 24. *The $\mathcal{EM}$-law $\kappa \colon TB \Rightarrow BT$ commutes with the step compositions in* (11)*, as in:*

$$\begin{tikzcd}
TBG \arrow[rr, "\kappa G"] \arrow[dr, "\tau \circledcirc \delta"'] & & BTG \arrow[dl, "\delta \circledcirc \tau"] \\
 & GL &
\end{tikzcd} \tag{19}$$

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

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

The functor $\widehat{G} \colon \mathbf{D}^{\mathrm{op}} \to \mathcal{EM}(T)$ is the lifting corresponding to τ, see Lemma 23.

Proof. The existence of such a ϱ 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 (19), see (11).c ☐

Theorem 25. *If the equivalent conditions in Lemma 24 hold, then the map* e *defined in* (18) *is an algebra morphism from* ℓ_{em} *to* $\ell^{\log}$*, as on the left below.*

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

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

Proof. We use that $\ell_{\mathrm{em}} = \zeta^{-1} \circ B(a) \colon BT(\Theta) \to \Theta$, where $((\Theta, a), \zeta)$ is the final $\overline{B}$-coalgebra, see Section 3. We need to prove that the outside 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:

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

And on the other hand, using that $\tau_2 = F(\tau_1 \circ T(\eta)) \circ \varepsilon$,

$$\begin{aligned}
\tau_2 \circ \overline{e} &= F(\tau_1 \circ T(\eta)) \circ \varepsilon \circ F(\mathsf{e}) \circ \varepsilon \\
&= F(\tau_1 \circ T(\eta)) \circ FG(F(\mathsf{e}) \circ \varepsilon) \circ \varepsilon \\
&= F\big(G(F(\mathsf{e}) \circ \varepsilon) \circ \tau_1 \circ T(\eta)\big) \circ \varepsilon \\
&= F\big(\tau_1 \circ TG(F(\mathsf{e}) \circ \varepsilon) \circ T(\eta)\big) \circ \varepsilon \\
&= F\big(\tau_1 \circ T(G(\varepsilon) \circ GF(\mathsf{e}) \circ \eta)\big) \circ \varepsilon \\
&= F(\tau_1 \circ T(\mathsf{e})) \circ \varepsilon \\
&= \overline{\tau_1 \circ T(\mathsf{e})}.
\end{aligned}$$

By transposing the map e in (18), it follows that $\overline{\mathsf{e}} \colon \Phi \to F(\Theta)$ is the unique morphism from the initial L-algebra $\alpha \colon L(\Phi) \xrightarrow{\cong} \Phi$ to $F(\zeta) \circ \delta_2 \colon LF(\Theta) \to F(\Theta)$. 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 3, one can show that the upper-left rectangle is equivalent to the assumption (19). To see this, we use Lemma 8 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 morphisms from c to the corecursive algebra on $G(\Phi)$. □

The equality $\mathsf{e} \circ \mathsf{em}_c = \mathsf{log}_c$ means that equivalence wrt Eilenberg-Moore trace semantics implies equivalence wrt 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 on $G(\Phi)$ 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 $\mathbf{C}$) [23].

Lemma 26. *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 (18) 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 11), and the logical approach (Proposition 17). 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 determinisation procedure. The underlying distributive laws satisfy (19) by construction, so that the two approaches coincide (as already seen in the concrete examples).

Theorem 27. *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 11, together with δ, τ as defined in the proof of Proposition 17, satisfies* (19). *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 (19), *i.e.*, $\delta \circledcirc \tau \circ \kappa = \tau \circledcirc \delta$, we first compute, following (11),

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

Hence, we need to show that

$$\delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2) \rangle = t^{A \times X + 1} \circ \mathrm{st} \circ T(\delta_X) \quad (20)$$

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

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

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

$$\begin{aligned}\mathrm{ev}_* \circ \delta_X(\omega, f) &= \delta_X(\omega, f)(*) = \omega = \pi_1(\omega, f)\,,\\ \mathrm{ev}_{(a,x)} \circ \delta_X(\omega, f) &= \delta_X(\omega, f)(a, x) = f(a)(x) = \mathrm{ev}_x \circ \mathrm{ev}_a \circ \pi_2(\omega, f)\,.\end{aligned}$$

For the left-hand side of (20), 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 (20). 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 (20) as desired.

Since (19) is satisfied, it follows from Theorem 25 that $\mathsf{e} \circ \mathsf{em}_c = \mathsf{log}_c$. Since δ is an iso, e is an iso as well by Lemma 26. ☐

7.2 Kleisli and Logic

To compare the Kleisli approach to the logical approach, we combine their assumptions. This amounts to 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 $\beta\colon A(\Psi) \overset{\cong}{\Rightarrow} \Psi$.
2. A $\mathcal{K}\ell$-law $\lambda\colon AT \Rightarrow TA$, or equivalently, an extension/lifting $\overline{A}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ of $A\colon \mathbf{C} \to \mathbf{C}$.
3. $(\Psi, J(\beta^{-1}))$ is a final $\overline{B}$-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).

Again, we assume τ to be compatible with the monad, satisfying the equivalent conditions in Lemma 23. We obtain the following unique coalgebra-to-algebra morphism k from the initial A-algebra:

$$\begin{tikzcd}
\Psi \arrow[rr, "\mathsf{k}"] & & G(\Phi) \\
& & GL(\Phi) \arrow[u, "G(\alpha^{-1})"'] \\
A(\Psi) \arrow[uu, "\beta", "\cong"'] \arrow[rr, "A(\mathsf{k})"] & & AG(\Phi) \arrow[u, "\delta"']
\end{tikzcd} \tag{21}$$

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 25.

Lemma 28. *The distributive law $\lambda\colon AT \Rightarrow TA$ commutes with the logics in* (11), *as in:*

$$\begin{tikzcd}
ATG \arrow[rr, "\lambda G"] \arrow[dr, "\delta \circledcirc \tau"'] & & TAG \arrow[dl, "\tau \circledcirc \delta"] \\
& GL &
\end{tikzcd} \tag{22}$$

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

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

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 23.

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 $f\colon X \to T(Y)$.

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

The lower rectangle commutes by naturality, the upper is equivalent to (22). Hence, (22) 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 29. *If the equivalent conditions in Lemma 28 hold, then the map* $\overline{\mathsf{k}} = \tau_{\Phi} \circ T(\mathsf{k}) \colon T(\Psi) \to G(\Phi)$ *is an algebra morphism from* ℓ_{kl} *to* $\ell_{\log}$*, as on the left below.*

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

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

Proof. Consider the following diagram.

$$\begin{tikzcd}
TBT(\Psi) \arrow[r, "TBT(\mathsf{k})"] \arrow[d, "T(\lambda)"] \arrow[ddd, bend right, "\ell_{\mathrm{kl}}"'] & TBTG(\Phi) \arrow[rr, "TB(\tau)"] \arrow[d, "T(\lambda)"] & & TBG(\Phi) \arrow[d, "T(\delta)"'] \arrow[ddd, bend left, "\ell_{\log}"] \\
TTB(\Psi) \arrow[r, "TTB(\mathsf{k})"] \arrow[d, "\mu"] & TTBG(\Phi) \arrow[r, "TT\delta"] \arrow[d, "\mu"] & TTGL(\Phi) \arrow[r, "T(\tau)"] \arrow[d, "\mu"] & TGL(\Phi) \arrow[d, "\tau"'] \\
TB(\Psi) \arrow[r, "TB(\mathsf{k})"] \arrow[d, "T(\beta)"] & TBG(\Phi) \arrow[r, "T(\delta)"] & TGL(\Phi) \arrow[r, "\tau"] \arrow[d, "TG(\alpha^{-1})"] & GL(\Phi) \arrow[d, "G(\alpha^{-1})"'] \\
T(\Psi) \arrow[rr, "T(\mathsf{k})"'] & & TG(\Phi) \arrow[r, "\tau"'] & G(\Phi)
\end{tikzcd}$$

Everything commutes: the upper right rectangle by assumption (22), the rightmost 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 (24) and the rest by naturality. ☐

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 26, we currently do not have a converse. If δ has monic components, then it is easy to use corecursiveness to define a map from $\ell_{\log}$ to ℓ_{kl}, but this surprisingly is not sufficient to show $\overline{\mathsf{k}}$ to be monic, as confirmed by Example 30 below. In the comparison between Eilenberg-Moore and Kleisli traces in Subsection 7.3, a similar difficulty arises.

Example 30. We give an example where $\delta \colon BG \Rightarrow GL$ is monic and (22) commutes, but where nevertheless logical equivalence is stronger than 'Kleisli' trace equivalence. Let $\mathbf{C} = \mathbf{D} = \mathbf{Sets}$, $F = G = 2^{-}$, $A = L = (\Sigma \times -) + 1$, $T = \mathcal{P}$, $\tau \colon \mathcal{P}2^{-} \Rightarrow 2^{-}$ given by union as before, and define the step δ by $\delta_X(a, \varphi)(t) = \top$ iff $\exists x. t = (a, x) \wedge \varphi(x)$, and $\delta_X(*)(t) = \top$ (the latter differs from the step in Proposition 19). Notice that δ indeed has monic components.

Let $\lambda\colon AT \Rightarrow TA$ be the distributive law from [13], given by $\lambda_X(a, S) = \{(a, x) \mid x \in S\}$ and $\lambda(*) = \{*\}$. Then (22) is satisfied:

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

It is straightforward to check that this commutes. However, given a coalgebra $f\colon X \to TB(X)$, the induced logical semantics $\mathsf{log}\colon X \to 2^{\Sigma^*}$ is: $\mathsf{log}(x)(w) = \top$ iff $* \in f(x)$ or $\exists a \in \Sigma, v \in \Sigma^*, y \in X. w = av \wedge (a, y) \in f(x) \wedge \mathsf{log}(y)(v) = \top$. In particular, this means that if $* \in f(x)$ and $* \in f(y)$ 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 [13].

Cîrstea [9] compares logical traces to a 'path-based semantics', which resembles the Kleisli approach (as well as [25]) 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 [13]. Cîrstea shows that, with $\Omega = T(1)$, $t = \mu_1\colon TT(1) \to T(1)$ and δ from the proof of Proposition 19 (assuming $T1$ to have enough structure to define that logic), the triangle (22) commutes (see [9, Lemma 5.12]).

7.3 Kleisli and Eilenberg-Moore

First, we combine the assumptions of the Eilenberg-Moore approach, in this case for TA-coalgebras, and the Kleisli approach. This amounts to endofunctors A, B and a monad T, on a base category $\mathbf{C}$, 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 $\rho\colon AU \Rightarrow U\overline{B}$.

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 $U\varepsilon\colon UT \Rightarrow U$. Its components are $\mathcal{EM}$-algebras, since this is a monad action. For the trace semantics of TA-coalgebras via Eilenberg-Moore, see Section 3.1, we make use of the composed step:

$$U\varepsilon \circledcirc \rho = \Big(TAU \xRightarrow{T\rho} TU\overline{B} \xRightarrow{U\varepsilon\overline{B}} U\overline{B}\Big) \tag{23}$$

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 29. 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:

$$\begin{tikzcd}
\Psi \arrow[r, dashed, "\mathsf{k}"] & \Theta & \\
& B(\Theta) \arrow[u, "\zeta^{-1}"'] & \\
& U\overline{B}(\Theta, a) \arrow[u, equal] & BT(\Theta) \arrow[ul, "B(a)"'] \arrow[uul, bend right, "\ell_{\mathrm{em}}"'] \\
& AU(\Theta, a) \arrow[u, "\rho"'] & TA(\Theta) \arrow[u, "U(\rho_2)"'] \\
A(\Psi) \arrow[uuuu, "\beta", "\cong"'] \arrow[r, dashed, "A(\mathsf{k})"] & A(\Theta) \arrow[u, equal] \arrow[ur, bend right, "\eta"'] &
\end{tikzcd} \tag{24}$$

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

$$\begin{aligned}
B(a) \circ U(\rho_2) \circ \eta &= B(a) \circ B(\mu) \circ \kappa \circ T(\rho) \circ TA(\eta) \circ \eta \quad \text{by Theorem 3} \\
&= B(a) \circ B(\mu) \circ \kappa \circ \eta \circ \rho \circ A(\eta) \\
&= B(a) \circ B(\mu) \circ B(\eta) \circ \rho \circ A(\eta) \\
&= B(a) \circ \rho \circ A(\eta) \\
&= \rho \circ A(a) \circ A(\eta) \qquad \text{since } a\colon (T(\Theta), \mu) \to (\Theta, a) \text{ in } \mathcal{EM}(T) \\
&= \rho.
\end{aligned}$$

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 algebras $\ell_{\mathrm{em}}\colon BT(\Theta) \to \Theta$ from (6) and (7), with $G_{U(\rho_2)}(\ell_{\mathrm{em}}) = \ell_{\mathrm{em}} \circ U(\rho_2)\colon TA(\Theta) \to \Theta$, giving Eilenberg-Moore trace semantics. The map $\overline{\mathsf{k}}$ thus relates the carriers of these corecursive TA-algebras.

Like in the previous sections, we now give a sufficient condition for the map $\overline{\mathsf{k}}\colon T(\Psi) \to \Theta$ to be an algebra morphism.

Proposition 31. *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:*

$$\begin{tikzcd} ATU \arrow[rr, "\lambda"] \arrow[dr, "\rho\circledcirc U(\varepsilon)"'] & & TAU \arrow[dl, "U(\varepsilon)\circledcirc\rho"] \\ & U\overline{B} & \end{tikzcd} \tag{25}$$

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

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

   The functor $\widehat{\mathcal{F}}\colon \mathcal{K}\ell(T) \to \mathcal{EM}(T)$ is the extension corresponding to $U(\varepsilon)$, according to Theorem 3; it is often called the 'comparison' functor, and then written as K.

3. *The following 'extension requirement' from [18] commutes:*

$$\begin{tikzcd} TAT \arrow[rr, "U(\rho_2)"] \arrow[d, "T(\lambda)"'] & & BTT \arrow[d, "B(\mu)"] \\ TTA \arrow[r, "\mu"'] & TA \arrow[r, "U(\rho_2)"'] & BT \end{tikzcd} \tag{26}$$

Proof. The equivalence of points (1) and (2) is an instance of Lemma 28, where it should be noted that we are instantiating $\mathbf{D}$ with $\mathcal{EM}(T)^{\mathrm{op}}$, which causes the two natural transformation in the above diagram to be in opposite direction.

We show the equivalence of (1) and (3). Using Lemma 8, it is straightforward to check, via Theorem 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 (25) 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 32. *If the equivalent conditions in Proposition 31 hold, then the map* $\overline{\mathsf{k}}\colon T(\Psi) \to \Theta$ *obtained from* (24)*, is a* TA*-algebra morphism between corecursive algebras* ℓ_{kl} *and* $G_{U(\rho_2)}(\ell_{\mathrm{em}}) = \ell_{\mathrm{em}} \circ U(\rho_2)$*, 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:

$$\begin{aligned}
\overline{\mathsf{k}} \circ \ell_{\mathrm{kl}} &\overset{(16)}{=} a \circ T(\mathsf{k}) \circ T(\beta) \circ \mu \circ T(\lambda) \\
&\overset{(24)}{=} a \circ T(\zeta^{-1} \circ B(a) \circ U(\rho_2) \circ \eta \circ A(\mathsf{k})) \circ \mu \circ T(\lambda) \\
&\overset{(5)}{=} \zeta^{-1} \circ B(a) \circ \kappa \circ TB(a) \circ TU(\rho_2) \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ BT(a) \circ \kappa \circ TU(\rho_2) \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ B(\mu) \circ \kappa \circ TU(\rho_2) \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&\overset{(9)}{=} \zeta^{-1} \circ B(a) \circ U(\rho_2) \circ \mu \circ T(\eta) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ U(\rho_2) \circ TA(\mathsf{k}) \circ \mu \circ T(\lambda) \\
&= \zeta^{-1} \circ B(a) \circ BT(\mathsf{k}) \circ U(\rho_2) \circ \mu \circ T(\lambda) \\
&\overset{(26)}{=} \zeta^{-1} \circ B(a) \circ BT(\mathsf{k}) \circ B(\mu) \circ U(\rho_2) \\
&= \zeta^{-1} \circ B(a) \circ B(\mu) \circ BT^2(\mathsf{k}) \circ U(\rho_2) \\
&= \zeta^{-1} \circ B(a) \circ BT(a) \circ BT^2(\mathsf{k}) \circ U(\rho_2) \\
&= \ell_{\mathrm{em}} \circ BT(\overline{k}) \circ U(\rho_2) \\
&= \ell_{\mathrm{em}} \circ U(\rho_2) \circ TA(\overline{k}).
\end{aligned}$$

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 em^A_c. But this easy using the rectangle in the theorem:

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

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 [18], such a condition is also missing; monicity of $\overline{\mathsf{k}}$ is only shown to hold in several concrete examples.

In [18, §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 in Subsection 3.1 as (9) in our generalization of Eilenberg-Moore trace semantics that is used here. The first of the above two rectangles captures compatibility in Proposition 31 and is used for a comparison of Kleisli and Eilenberg-Moore semantics in Theorem 32. The conclusion is that the approach of this paper not only covers the approach of [18, §6] but also puts it in a wider step-based perspective, using corecursive algebras.

8 Completely Iterative Algebras

In this paper, we constructed several corecursive algebras. We briefly show that they all satisfy the following stronger property [32].

Definition 33. *For an endofunctor H on* $\mathbf{C}$*, 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 $f\colon X \to A$ such that the following diagram commutes.*

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

Following [15,32], we have two ways of constructing such algebras.

Proposition 34.

1. *If $\zeta\colon A \to HA$ 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.*

We may thus say: *"step-induced algebra liftings of right adjoints preserve complete iterativity"*. Consequently, by analogy with Theorem 6, if L has a final coalgebra (Ψ, ζ) then $G_\rho(A, \zeta^{-1})$ is completely iterative. For our examples, this may be seen as a trace semantics for a coalgebra c that may sometimes stop following the behaviour functor and instead provide semantics 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 [22], 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 [35,24]. 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.

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

References

1. F. Bartels. Generalised coinduction. *Mathematical Structures in Computer Science*, 13(2):321–348, 2003.
2. F. Bonchi, A. Silva, and A. Sokolova. The power of convex algebras. In R. Meyer and U. Nestmann, editors, *28th International Conference on Concurrency Theory, CONCUR 2017*, volume 85 of *LIPIcs*, pages 23:1–23:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
3. M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *Foundations of Software Science and Computational Structures, 8th International Conference, FOSSACS 2005*, pages 455–469, 2005.
4. M. Bonsangue, S. Milius, and A. Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *ACM Trans. Comput. Log.*, 14(1):7:1–7:52, 2013.
5. Nathan Bowler, Paul Blain Levy, and Gordon Plotkin. Initial algebras and final coalgebras consisting of nondeterministic finite trace strategies. *Electronic Notes in Theoretical Computer Science*, 341:23 – 44, 2018. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).
6. V. Capretta, T. Uustalu, and V. Vene. Recursive coalgebras from comonads. *Inf. Comput.*, 204(4):437–468, 2006.
7. V. Capretta, T. Uustalu, and V. Vene. Corecursive algebras: A study of general structured corecursion. In M. Vinicius Medeiros Oliveira and J. Woodcock, editors,

Formal Methods: Foundations and Applications, number 5902 in Lect. Notes Comp. Sci., pages 84–100. Springer, Berlin, 2009.

8. L-T. Chen and A. Jung. On a categorical framework for coalgebraic modal logic. *Electr. Notes Theor. Comput. Sci*, 308:109–128, 2014.
9. C. Cîrstea. A coalgebraic approach to quantitative linear time logics. *CoRR*, abs/1612.07844, 2016.
10. A. Eppendahl. Coalgebra-to-algebra morphisms. *Electr. Notes Theor. Comput. Sci*, 29:42–49, 1999.
11. J. Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1988.
12. I. Hasuo. Generic weakest precondition semantics from monads enriched with order. *Theor. Comput. Sci.*, 604:2–29, 2015.
13. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Log. Meth. Comp. Sci.*, 3(4), 2007.
14. W. Hino, H. Kobayashi, I. Hasuo, and B. Jacobs. Healthiness from duality. In *Logic in Computer Science.* IEEE, Computer Science Press, 2016.
15. R. Hinze, N. Wu, and J. Gibbons. Conjugate hylomorphisms - or: The mother of all structured recursion schemes. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the Symposium on Principles of Programming Languages, POPL 2015*, pages 527–538. ACM, 2015.
16. B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, pages 375–404, 2006.
17. B. Jacobs. A recipe for state and effect triangles. *Log. Meth. Comp. Sci.*, 13(2), 2017. See `https://lmcs.episciences.org/3660`.
18. B. Jacobs, A. Silva, and A. Sokolova. Trace semantics via determinization. *Journ. of Computer and System Sci.*, 81(5):859–879, 2015.
19. Bart Jacobs, Paul Levy, and Jurriaan Rot. Steps and traces. In Corina Cîrstea, editor, *Coalgebraic Methods in Computer Science - 14th IFIP WG 1.3 International Workshop, CMCS 2018, Colocated with ETAPS 2018, Thessaloniki, Greece, April 14-15, 2018, Revised Selected Papers*, volume 11202 of *Lecture Notes in Computer Science*, pages 122–143. Springer, 2018.
20. R. Jagadeesan, C. S. Pitcher, and J. Riely. Open bisimulation for aspects. In *Proceedings, International Conference on Aspect-Oriented Software Development.* ACM Press, 2007.
21. G. M. Kelly and R. Street. Review of the elements of 2-categories. In Gregory M. Kelly, editor, *Category Seminar: Proceedings Sydney Category Seminar 1972/1973*, number 420 in Lecture Notes in Mathematics. Springer-Verlag, 1974.
22. H. Kerstan, B. König, and B. Westerbaan. Lifting adjunctions to coalgebras to (re)discover automata constructions. In M. Bonsangue, editor, *Coalgebraic Methods in Computer Science (CMCS 2014), Revised Selected Papers*, volume 8446 of *Lect. Notes Comp. Sci.*, pages 168–188. Springer, 2014.
23. B. Klin. Coalgebraic modal logic beyond sets. In M. Fiore, editor, *Math. Found. of Programming Semantics*, number 173 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2007.
24. B. Klin and J. Rot. Coalgebraic trace semantics via forgetful logics. *Log. Meth. Comp. Sci.*, 12(4:10), 2016. See `https://lmcs.episciences.org/2622`.
25. A. Kurz, S. Milius, D. Pattinson, and L. Schröder. Simplified coalgebraic trace equivalence. In *Software, Services, and Systems - Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering*, pages 75–90, 2015.

26. J. Laird. A fully abstract trace semantics for general references. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Proceedings, 34th International Colloquium on Automata, Languages and Programming, Wroclaw, Poland*, volume 4596 of *Lecture Notes in Computer Science*, pages 667–679. Springer, 2007.
27. S. B. Lassen and P. B. Levy. Typed normal form bisimulation. In J Duparc and TA Henzinger, editors, *Proc. , 23rd Conf. on Comp. Sci. and Logic*, volume 4646 of *LNCS*, 2007.
28. T. Leinster. *Higher Operads, Higher Categories*, volume 298 of *London Mathematical Society Lecture Notes.* Cambridge University Press, 2004.
29. P. Levy. Final coalgebras from corecursive algebras. In L. Moss and P. Sobocinski, editors, *Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*, volume 35 of *LIPIcs*, pages 221–237. Schloss Dagstuhl, 2015.
30. Paul Blain Levy and Sam Staton. Transition systems over games. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 64:1–64:10. ACM, 2014.
31. S. Milius, D. Pattinson, and L. Schröder. Generic trace semantics and graded monads. In *6th Conference on Algebra and Coalgebra in Computer Science, CALCO 2015, June 24-26, 2015, Nijmegen, The Netherlands*, pages 253–269, 2015.
32. Stefan Milius. Completely iterative algebras and completely iterative monads. *Inf. Comput*, 196(1):1–41, 2005.
33. D. Pavlović, M. Mislove, and J. Worrell. Testing semantics: Connecting processes and process logics. In M. Johnson and V. Vene, editors, *Algebraic Methods and Software Technology*, number 4019 in Lect. Notes Comp. Sci., pages 308–322. Springer, Berlin, 2006.
34. A. W. Roscoe, S.D. Brookes, and C.A.R. Hoare. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, July 1984.
35. J. Rot. Coalgebraic minimization of automata by initiality and finality. *Elect. Notes in Theor. Comp. Sci.*, 325:253–276, 2016.
36. A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing determinization from automata to coalgebras. *Log. Meth. Comp. Sci.*, 9(1), 2013.
37. Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, pages 280–291. IEEE Computer Society, 1997.

A 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 35. *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 $\mathbf{C} \nrightarrow \mathbf{D}$ corresponds 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 36.

1. *A* map *of bimodules*

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

sends each $\mathbf{O}$*-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

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

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

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

Here are two ways of constructing a bimodule $\mathbf{C} \nrightarrow \mathbf{D}$.

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

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 $n : \mathbf{O} \cong F^{\mathsf{Left}}$.

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 correspond to 2-cells of bimodules, as follows.

Theorem 37. *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*

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

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* $\left(F'HX \xrightarrow{\rho_2} LFX \xrightarrow{Lm(g)} LY \right)$. *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:

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

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 38.

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 hom-set $\mathbf{C}(X, \Theta)$ 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 39.

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 5 into two parts.

Proposition 40.

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 [Propositions 16–17][29].