Steps and Traces

Jacobs, Bart; Levy, Paul; Rot, Jurriaan

DOI:
10.1007/978-3-030-00389-0

Document Version
Peer reviewed version

Citation for published version (Harvard):
Jacobs, B, Levy, P & Rot, J 2018, Steps and Traces. in C Cirstea (ed.), 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. Lecture Notes in Computer Science , vol. 11202, Springer, pp. 122-143, 14th IFIP WG 1.3 International Workshop on Coalgebraic Methods in Computer Science (CMCS 2018), Thessaloniki, Greece, 14/04/18. https://doi.org/10.1007/978-3-030-00389-0

Steps and Traces

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

[1] Institute for Computing and Information Sciences,
Radboud Universiteit, Nijmegen, The Netherlands
[2] University of Birmingham, UK
bart@cs.ru.nl and P.B.Levy@cs.bham.ac.uk and jrot@cs.ru.nl

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

⋆ 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.

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 [5], 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 two of these forms of trace semantics, namely logic-to-Eilenberg-Moore and logic-to-Kleisli. The other combinations involve subtleties which we do not fully grasp yet. Due to space limitations, we don't cover the whole field of coalgebraic trace semantics: we focus only on finite trace semantics, and also exclude at this stage the 'iteration' based approaches, e.g., in [25,22,8].

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. The next section establishes a connection between the Eilenberg-Moore and the logical approach, and a connection between the Kleisli and logical approach (Section 6). In Section 7 we briefly show that our construction of corecursive algebras strengthens to a construction of completely iterative algebras. Finally, in Section 8 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 [9,6] as the dual of Taylor's notion of recursive coalgebra [10], 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 $a\colon H(A) \to A$ is a map $f\colon X \to A$ such that the diagram*

$$\begin{array}{ccc} X & \xrightarrow{\quad f \quad} & A \\ {\scriptstyle c}\downarrow & & \uparrow{\scriptstyle a} \\ H(X) & \xrightarrow[H(f)]{} & H(A) \end{array}$$

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

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

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

Here is some intuition.

- As explained in [14], 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 $\zeta\colon A \xrightarrow{\cong} H(A)$, the inverse $\zeta^{-1}\colon H(A) \to A$ 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, which we shall now explain. Our basic setting consists of an adjunction, two endofunctors, and a natural transformation:

$$H \circlearrowright \mathbf{C} \underset{G}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{D} \circlearrowright L \qquad \text{with} \qquad HG \xRightarrow{\ \rho\ } GL \tag{1}$$

The natural transformation $\rho\colon HG \Rightarrow GL$ will be called a *step*. Here H is the *behaviour functor*: we study H-coalgebras and give semantics for them in a corecursive H-algebra. This arrangement is well-known in the area of coalgebraic modal logic [3,27,20,7,24], but we shall see that its application is wider.

A step can be formulated in several equivalent ways [18,23].

Theorem 2. *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 $\rho_4\colon H \Rightarrow GLF$.*

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:

$$\begin{array}{ccc} \mathcal{EM}(H) & \xleftarrow{\overline{G}} & \mathcal{EM}(L) \\ \downarrow & & \downarrow \\ \mathbf{C} & \xleftarrow{\quad G \quad} & \mathbf{D} \end{array} \qquad\qquad \begin{array}{ccc} \mathcal{K}\ell(H) & \xrightarrow{\overline{F}} & \mathcal{K}\ell(L) \\ \uparrow & & \uparrow \\ \mathbf{C} & \xrightarrow{\quad F \quad} & \mathbf{D} \end{array} \qquad \square$$

Proof. We only mention the bijective correspondences: ρ_1 and ρ_3 correspond by adjoint transposition, and similarly for ρ_2 and ρ_4. Further, ρ_2 and ρ_3 are obtained from each other by:

$$\rho_3 = \Big(\; FHG \xRightarrow{\rho_2 G} LFG \xRightarrow{L\varepsilon} L \; \Big)$$
$$\rho_2 = \Big(\; FH \xRightarrow{FH\eta} FHGF \xRightarrow{\rho_3 F} LF \; \Big) \; .$$

$\square$

It is common to refer to ρ_1 and ρ_2 as *mates*; the other two maps are their adjoint transposes. 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 drop the usual subscript of components of natural transformations.

Definition 3. *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*:*

$$\begin{array}{ccc} \mathrm{Alg}(H) & \xleftarrow{G_\rho} & \mathrm{Alg}(L) \\ \downarrow & & \downarrow \\ \mathbf{C} & \xleftarrow{\quad G \quad} & \mathbf{D} \end{array} \qquad\qquad \begin{aligned} &G_\rho\Big(L(A) \xrightarrow{a} A\Big) \;\coloneqq \\ &\quad \Big(HG(A) \xrightarrow{\rho} GL(A) \xrightarrow{G(a)} G(A)\Big). \end{aligned}$$

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

$$\begin{array}{ccc} \mathrm{CoAlg}(H) & \xrightarrow{F^\rho} & \mathrm{CoAlg}(L) \\ \downarrow & & \downarrow \\ \mathbf{C} & \xrightarrow{\quad F \quad} & \mathbf{D} \end{array} \qquad\qquad \begin{aligned} &F^\rho\Big(X \xrightarrow{c} H(X)\Big) \;\coloneqq \\ &\quad \Big(F(X) \xrightarrow{F(c)} FH(X) \xrightarrow{\rho} LF(X)\Big). \end{aligned}$$

Our approach relies on the following basic result.

Proposition 4 ([5]). *For each corecursive L-algebra $a\colon L(A) \to A$, the transferred H-algebra $G_\rho(A,a)\colon HG(A) \to G(A)$ is also corecursive. Explicitly, for any H-coalgebra (X,c), the unique coalgebra-to-algebra map $(X,c) \to G_\rho(A,a)$ is the adjoint transpose of the unique coalgebra-to-algebra map $F^\rho(X,c) \to (A,a)$.* $\square$

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 5. *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:*

$$\begin{array}{ccc} X & \overset{c^\dagger}{\dashrightarrow} & G(\Psi) \\ {\scriptstyle c}\downarrow & & \uparrow{\scriptstyle G_\rho(\Psi,\zeta^{-1})} \\ H(X) & \overset{H(c^\dagger)}{\dashrightarrow} & HG(\Psi) \end{array} \qquad\qquad \begin{array}{ccc} F(X) & \overset{\overline{c^\dagger}}{\dashrightarrow} & \Psi \\ {\scriptstyle F^\rho(X,c)}\downarrow & & \uparrow{\scriptstyle \zeta^{-1}} \\ LF(X) & \overset{L(\overline{c^\dagger})}{\dashrightarrow} & L(\Psi) \end{array}$$

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 5 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 it as the *trace semantics* map.

3 Traces via Eilenberg-Moore

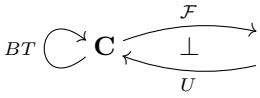
We recall the approach to trace semantics developed in [17,29,4], putting it in the framework of the previous section. The approach deals with coalgebras for the composite functor BT, where T is a monad that captures the 'branching' aspect. The following assumptions are required.

$$\begin{array}{ccc} \mathcal{EM}(T) & \xrightarrow{\overline{B}} & \mathcal{EM}(T) \\ {\scriptstyle U}\downarrow & & \downarrow{\scriptstyle U} \\ \mathbf{C} & \xrightarrow[B]{} & \mathbf{C} \end{array} \tag{2}$$

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^T, μ^T), 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 (2), or, equivalently, an $\mathcal{EM}$-law $\kappa\colon TB \Rightarrow BT$.

Example 6. To briefly illustrate these ingredients, we consider non-deterministic automata. These are BT-coalgebras with $B\colon \mathbf{Sets} \to \mathbf{Sets}$, $B(X) = 2 \times X^A$ with $2 = \{\bot, \top\}$ 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 (JSL). The lifting is defined by product 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.

These assumptions give rise to the following instance of our general setting (1):



with $\rho\colon BTU \Longrightarrow U\overline{B}$ where $\rho_{(X,a)} = \big(BTX \xrightarrow{Ba} BX\big)$

Actually — and equivalently, by Theorem 2 — 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 [30].

Lemma 7. *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.* □

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:

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

By Theorem 5, this algebra 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.

$$\begin{array}{ccc} X & \overset{\mathsf{em}_c}{\dashrightarrow} & \Theta \\ {\scriptstyle c}\downarrow & & \uparrow{\scriptstyle \ell_{\mathrm{em}}} \\ BT(X) & \underset{BT(\mathsf{em}_c)}{\dashrightarrow} & BT(\Theta) \end{array} \tag{3}$$

The unique map em_c in (3) appears in the literature as a 'coiteration up-to' or 'unique solution' theorem [1]. Examples follow later in this section (Theorem 8, Example 9).

In [17,29], 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:

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

It follows from Theorem 2 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 [17]. By Theorem 5, the trace semantics em_c can equivalently be characterised in terms of $\mathcal{F}^\rho$, as the unique map $\overline{\mathsf{em}_c}$ making (4) commute. This

$$\begin{array}{ccc} T(X) & \overset{\overline{\mathsf{em}_c}}{\dashrightarrow} & \Theta \\ {\scriptstyle \overline{c}}\downarrow & & \uparrow{\scriptstyle \zeta^{-1}} \\ BT(X) & \underset{B(\overline{\mathsf{em}_c})}{\dashrightarrow} & B(\Theta) \end{array} \tag{4}$$

is how the trace semantics via Eilenberg-Moore is presented in [17,29]: as the transpose $\mathsf{em}_c = \overline{\mathsf{em}_c} \circ \eta^T_X$.

We conclude this section by recalling a canonical construction of a distributive law [15] for a class of 'automata-like' examples.

Theorem 8. *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*

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

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

Example 9. By Theorem 8, we obtain an explicit description of the trace semantics arising from the corecursive algebra (3): for any $\langle o, f\rangle\colon X \to \Omega \times T(X)^A$, the trace semantics is the unique map em in

$$\begin{array}{ccccccc} X & \dashrightarrow & & \xrightarrow{\mathsf{em}} & & \dashrightarrow & \Omega^{A^*} \\ {\scriptstyle\langle o,f\rangle}\downarrow & & & & & & \uparrow{\scriptstyle\zeta^{-1}} \\ BT(X) & \underset{BT(\mathsf{em})}{\dashrightarrow} & BT(\Omega^{A^*}) & \xrightarrow[\mathrm{st}]{} & B(T(\Omega)^{A^*}) & \xrightarrow[B(t^{A^*})]{} & BT(\Omega^{A^*}) \end{array}$$

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-$\exists$), NDA where branching is interpreted conjunctively (NDA-$\forall$) and (reactive) probabilistic automata (PA). Here $\mathcal{P}_\mathrm{f}$ is the finite powerset functor, and $\mathcal{D}_\mathrm{fin}$ the finitely supported distribution functor.

	T	Ω	$t\colon T(\Omega) \to \Omega$	$\mathsf{em}(x)(aw)$
NDA-$\exists$	$\mathcal{P}_\mathrm{f}$	$2 = \{\bot, \top\}$	$S \mapsto \bigvee S$	$\bigvee_{y \in f(x)(a)} \mathsf{em}(y)(w)$
NDA-$\forall$	$\mathcal{P}_\mathrm{f}$	$2 = \{\bot, \top\}$	$S \mapsto \bigwedge S$	$\bigwedge_{y \in f(x)(a)} \mathsf{em}(y)(w)$
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 [17,29].

4 Traces via Logic

This section illustrates how the 'logical' approach to trace semantics of [21], started in [27], 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 δ_τ and δ^τ in (6) below, in the sense of Definition 2; here we use 'delta' instead of 'rho' notation since it is common in modal logic.

Example 10. 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:

$$H \circlearrowright \mathbf{C} \underset{G}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{D}^{\mathrm{op}} \circlearrowright L \qquad \text{where } H = BT \text{ or } H = TB \tag{5}$$

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:

$$\begin{aligned} \delta_\tau &\coloneqq \Big(TBG \xRightarrow{T\delta} TGL \xRightarrow{\tau L} GL\Big) && \mathrm{CoAlg}(L) \xrightarrow{G_{\delta_\tau}} \mathrm{Alg}(TB) \\ \delta^\tau &\coloneqq \Big(BTG \xRightarrow{B\tau} BG \xRightarrow{\delta} GL\Big) && \mathrm{CoAlg}(L) \xrightarrow{G_{\delta^\tau}} \mathrm{Alg}(BT) \end{aligned} \tag{6}$$

Both δ^τ and δ_τ are steps, and hence give rise to step-induced algebra liftings G_{δ_τ} and G_{δ^τ} of G (Section 2). By Theorem 5, 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 $\mathbf{D}^{\mathrm{op}}$:

$$\begin{aligned} \ell_{\log} &\coloneqq \Big(TBG(\Phi) \xrightarrow{\delta_\tau} GL(\Phi) \xrightarrow[\cong]{G(\alpha^{-1})} G(\Phi)\Big)\,, \\ \ell^{\log} &\coloneqq \Big(BTG(\Phi) \xrightarrow{\delta^\tau} GL(\Phi) \xrightarrow[\cong]{G(\alpha^{-1})} G(\Phi)\Big)\,. \end{aligned} \tag{7}$$

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

$$\begin{array}{ccc} X & \overset{\mathsf{log}_c}{\dashrightarrow} & G(\Phi) \\ {\scriptstyle c}\downarrow & & \uparrow{\scriptstyle \ell_{\log}} \\ TB(X) & \overset{TB(\mathsf{log}_c)}{\dashrightarrow} & TBG(\Phi) \end{array} \qquad \begin{array}{ccc} Y & \overset{\mathsf{log}_d}{\dashrightarrow} & G(\Phi) \\ {\scriptstyle d}\downarrow & & \uparrow{\scriptstyle \ell^{\log}} \\ BT(Y) & \overset{BT(\mathsf{log}_d)}{\dashrightarrow} & BTG(\Phi) \end{array} \tag{8}$$

It is instructive to characterise this trace semantics in terms of the transpose and the step-induced coalgebra liftings F^{δ_τ} and F^{δ^τ}, showing how they arise as unique maps from an initial algebra:

$$\begin{array}{ccc} F(X) & \xleftarrow{\overline{\mathsf{log}_c}} & \Phi \\ {\scriptstyle F^{\delta_\tau}(X,c)}\uparrow & & \downarrow{\scriptstyle \alpha^{-1}} \\ LF(X) & \xleftarrow{L(\overline{\mathsf{log}_c})} & L(\Phi) \end{array} \qquad \begin{array}{ccc} F(Y) & \xleftarrow{\overline{\mathsf{log}_d}} & \Phi \\ {\scriptstyle F^{\delta^\tau}(Y,d)}\uparrow & & \downarrow{\scriptstyle \alpha^{-1}} \\ LF(Y) & \xleftarrow{L(\overline{\mathsf{log}_d})} & L(\Phi) \end{array} \tag{9}$$

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 [21] in a smooth manner.

Proposition 11. *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 $\alpha\colon A \times A^* + 1 \xrightarrow{\cong} A^*$. The map t extends to a modality $\tau\colon TG \Rightarrow G$, given on components by

$$\tau_X := \big(\ T(\Omega^X) \xrightarrow{\mathrm{st}} T(\Omega)^X \xrightarrow{t^X} \Omega^X\ \big)\,.$$

The logic $\delta\colon BG \Rightarrow GL$ is given by the isomorphism $\Omega \times (\Omega^-)^A \cong \Omega^{(A\times -)+1}$. Instantiating (7) we obtain the corecursive BT-algebra

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

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

Example 12. We instantiate the trace semantics log from Proposition 11 for various choices of Ω, T and t. Similar to the instances in Example 9, 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-$\exists$, NDA-$\forall$) and probabilistic automata (PA) are the same as in Example 9. 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 $T = \mathcal{P}_{\mathrm{f}}\mathcal{P}_{\mathrm{f}}$; 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-$\exists$	$\mathcal{P}_{\mathrm{f}}$	$2 = \{\bot, \top\}$	$S \mapsto \bigvee S$	$\bigvee_{y \in f(x)(a)} \mathsf{log}(y)(w)$
NDA-$\forall$	$\mathcal{P}_{\mathrm{f}}$	$2 = \{\bot, \top\}$	$S \mapsto \bigwedge S$	$\bigwedge_{y \in f(x)(a)} \mathsf{log}(y)(w)$
PA	$\mathcal{D}_{\mathrm{fin}}$	$[0,1]$	$\varphi \mapsto \sum_{p \in [0,1]} p \cdot \varphi(p)$	$\sum_{y \in X} \mathsf{log}(y)(w) \cdot f(x)(a)(y)$
AA	$\mathcal{P}_{\mathrm{f}}\mathcal{P}_{\mathrm{f}}$	$2 = \{\bot, \top\}$	$S \mapsto \bigvee_{T \in S} \bigwedge_{b \in T} b$	$\bigvee_{T \in f(x)(a)} \bigwedge_{y \in T} \mathsf{log}(y)(w)$

We also describe a logic for polynomial functors constructed from a signature. Here, we model a signature by a functor $\Sigma\colon \mathbb{N} \to \mathbf{Sets}$, where $\mathbb{N}$ is the discrete category of natural numbers. This gives rise to a functor $H_\Sigma\colon \mathbf{Sets} \to \mathbf{Sets}$ as usual by $H_\Sigma(X) = \coprod_{n \in \mathbb{N}} \Sigma(n) \times X^n$. We abuse notation and write $\sigma(x_1, \dots, x_n)$ instead of $(\sigma, x_1, \dots x_n)$. The initial algebra of H_Σ consists of closed terms (or finite node-labelled trees) over the signature.

Proposition 13. *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*

$$\mathsf{log}(x)(\sigma(t_1, \dots, t_n)) = t(T(m) \circ c(x))\,, \text{ where}$$
$$m = \left(t \mapsto \begin{cases} \bigwedge_i \mathsf{log}(x_i)(t_i) & \text{if } \exists x_1 \dots x_n.\, t = \sigma(x_1, \dots, x_n) \\ \bot & \text{otherwise} \end{cases} \right) : H_\Sigma(X) \to \Omega$$

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

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

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

The corecursive algebra ℓ_{log} is then given by:

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

The explicit characterisation of log is a straightforward computation. □

Example 14. 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 13 gives:

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

for every state $x \in X$ and tree $\sigma(t_1, \dots, t_n)$. This is the standard semantics of tree automata. It is easily adapted to *weighted* tree automata, see [21].

In both Example 14 and Example 12, the step-induced coalgebra lifting F_{δ^τ} (respectively F_{δ_τ}) of the underlying logic corresponds to reverse determinisation, see [21,28] for details. In particular, in Example 14 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 [12], and cast it in our abstract framework. It applies to coalgebras for a composite functor TB, where T is a monad modelling the type of branching. For example, a coalgebra $X \to \mathcal{P}(A \times X + S)$ has an associated map $X \to \mathcal{P}(A^* \times S)$ 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 B is $(A \times -) + S$. In general, the following assumptions are required.

$$\begin{array}{ccc} \mathcal{K}\ell(T) & \xrightarrow{\overline{B}} & \mathcal{K}\ell(T) \\ {\scriptstyle J}\uparrow & & \uparrow{\scriptstyle J} \\ \mathbf{C} & \xrightarrow[B]{} & \mathbf{C} \end{array} \tag{10}$$

1. An endofunctor $B\colon \mathbf{C} \to \mathbf{C}$ with an initial algebra $\beta\colon B(\Psi) \xrightarrow{\cong} \Psi$.
2. A monad (T, η^T, μ^T), 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 $\overline{B}$ of B, as in (10), or, equivalently, a $\mathcal{K}\ell$-law $\lambda\colon BT \Rightarrow TB$.
4. $(\Psi, J(\beta^{-1}))$ is a final $\overline{B}$-coalgebra.

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

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

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 [12]. 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):

$$TB \circlearrowright \mathbf{C} \underset{U}{\overset{J}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathcal{K}\ell(T) \circlearrowright \overline{B} \qquad \text{with} \qquad \begin{array}{l} \rho\colon TBU \Longrightarrow U\overline{B} \text{ where } \rho_X = \\ \big(TBTX \xrightarrow{T(\lambda)} T^2BX \xrightarrow{\mu^T} TBX\big) \end{array}$$

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

We apply the step-induced algebra lift $G_\rho\colon \mathrm{Alg}(\overline{B}) \to \mathrm{Alg}(TB)$ to the inverse of the final $\overline{B}$-coalgebra, and call it ℓ_{kl}:

$$\begin{aligned} \Big(TBT(\Psi) \xrightarrow{\ell_{\mathrm{kl}}} T(\Psi)\Big) &\coloneqq G_\rho(\Psi, J(\beta^{-1})^{-1}) \\ &= G_\rho(\Psi, J(\beta)) \\ &= \Big(TBT(\Psi) \xrightarrow{T(\lambda)} T^2B(\Psi) \xrightarrow{\mu^T} TB(\Psi) \xrightarrow{T(\beta)} T(\psi)\Big). \end{aligned}$$

By Theorem 5, this algebra is corecursive, i.e., for every coalgebra $c\colon X \to TB(X)$, there is a unique map kl_c as below:

$$\begin{array}{ccc} X & \overset{\mathsf{kl}_c}{\dashrightarrow} & T(\Psi) \\ {\scriptstyle c}\downarrow & & \uparrow{\scriptstyle \ell_{\mathrm{kl}}} \\ TB(X) & \overset{TB(\mathsf{kl}_c)}{\dashrightarrow} & TBT(\Psi) \end{array}$$

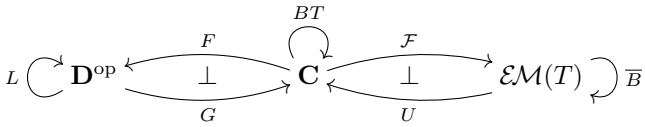
The trace semantics is exactly as in [12], to which we refer for examples.

6 Comparison

The presentation of trace semantics in terms of corecursive algebras allows us compare the different approaches by constructing algebra morphisms between them. In this section, we compare the Eilenberg-Moore against the logical approach, and the Kleisli against the logical approach as well. For a comparison between Kleisli and Eilenberg-Moore we refer to [17]. The latter is not in terms of corecursive algebras; we leave such a reformulation for future work. In [21], logical traces are also compared to determinisation constructions. But the technique is different, with the primary difference that no corecursive algebras are used there.

6.1 Eilenberg-Moore and Logic

To compare the Eilenberg-Moore approach to the logical approach, we combine their assumptions. This amounts to an adjunction $F \dashv G$, endofunctors B, L and a monad T as follows:



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 map τ is an assumption of the logical approach, but the compatibility with the monad structure was not assumed before (in the logical approach, T is not assumed to be a monad). 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). Therefore, by Theorem 2:

Lemma 15. *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 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 [13,16,11] where $\widehat{F}$ is called Pred. We turn back to the comparison between the Eilenberg-Moore and logical approach. First, observe that since $\delta \colon BG \Rightarrow GL$ is a step, it induces a corecursive B-algebra $BG(\Phi) \xrightarrow{\delta} GL(\Phi) \xrightarrow{G(\alpha^{-1})} G(\Phi)$. Hence, we obtain a unique map e as in the following diagram:

$$\begin{array}{ccccc} \Theta & & \xrightarrow{\quad\mathsf{e}\quad} & & G(\Phi) \\ {\scriptstyle\zeta}\downarrow & & & & \uparrow{\scriptstyle G(\alpha^{-1})} \\ B(\Theta) & \xrightarrow{B(\mathsf{e})} & BG(\Phi) & \xrightarrow{\delta} & GL(\Phi) \end{array} \tag{11}$$

This is a map from the carrier of the corecursive algebra ℓ_{em} (from the Eilenberg-Moore approach) to the carrier of the corecursive algebra $\ell^{\log}$ (from the logical approach). Note that, by the above diagram, it is a B-algebra morphism, whereas ℓ_{em} and $\ell^{\log}$ are BT-algebras. The following is 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 (Theorem 17).

Lemma 16. *The distributive law* κ *commutes with the logics in* (6)*, as in:*

$$\begin{array}{ccc} TBG & \xrightarrow{\quad\kappa G\quad} & BTG \\ & {\scriptstyle\delta_\tau}\searrow \quad \swarrow{\scriptstyle\delta^\tau} & \\ & GL & \end{array} \tag{12}$$

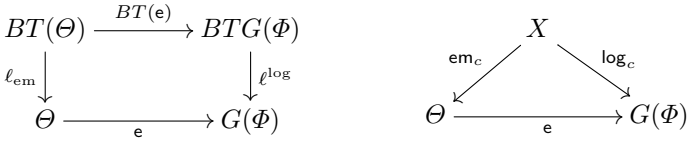
iff there is a natural transformation $\varrho \colon \overline{B}\widehat{G} \Rightarrow \widehat{G}L$ *such that* $U(\varrho) = \delta$ *— where the functor* $\widehat{G} \colon \mathbf{D}^{\mathrm{op}} \to \mathcal{EM}(T)$ *is the lifting corresponding to* τ *(Lemma 15).*

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:

$$\begin{array}{ccc} TBGX & \xrightarrow{T\delta} & TGLX \\ {\scriptstyle\kappa G}\downarrow & & \\ BTGX & & \downarrow{\scriptstyle\tau L} \\ {\scriptstyle B\tau}\downarrow & & \\ BGX & \xrightarrow{\delta} & GLX \end{array}$$

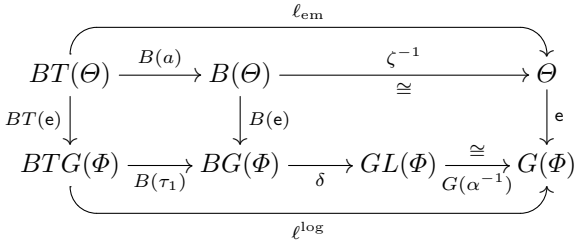
This corresponds exactly to (12). □

Theorem 17. *If the equivalent conditions in Lemma 16 hold, then the map* e *defined in* (11) *is an algebra morphism from* ℓ_{em} *to* $\ell^{\log}$*, as on the left below.*

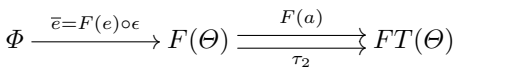


In that case, for any coalgebra $X \xrightarrow{c} BT(X)$ *the triangle on the right commutes.*

Proof. We use that $\ell_{\mathrm{em}} = \zeta^{-1} \circ B(a) \colon BT(\Theta) \to \Theta$, 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.



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:



Indeed, by transposing we have on the one hand:

$$\overline{\mathsf{e} \circ a} = F(a \circ \mathsf{e}) \circ \epsilon_\Phi = F(a) \circ F(\mathsf{e}) \circ \epsilon_\Phi = F(a) \circ \overline{\mathsf{e}}$$

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

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

By transposing the maps in (11), it follows that $\overline{\mathsf{e}} \colon \Phi \to F(\Theta)$ is the unique morphism from the initial L-algebra 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.

$$\begin{array}{ccccc}
LF(\Theta) & \xrightarrow{L(\tau_2)} & LFT(\Theta) & \xleftarrow{LF(a)} & LF(\Theta) \\
 & & \downarrow{\scriptstyle \delta_2 T} & & \downarrow{\scriptstyle \delta_2} \\
\downarrow{\scriptstyle \delta_2} & & FBT(\Theta) & \xleftarrow{FB(a)} & FB(\Theta) \\
 & & \downarrow{\scriptstyle F\kappa} & & \\
FB(\Theta) & \xrightarrow{\tau_2 B} & FTB(\Theta) & & \downarrow{\scriptstyle F(\zeta)} \\
\downarrow{\scriptstyle F(\zeta)} & & \downarrow{\scriptstyle FT(\zeta)} & & \\
F(\Theta) & \xrightarrow[\tau_2]{} & FT(\Theta) & \xleftarrow[F(a)]{} & F(\Theta)
\end{array}$$

Using the translation $(-)_1 \leftrightarrow (-)_2$ (of Theorem 2), one shows that the upper-left rectangle is equivalent to the assumption (12). To see this, we use that $(\delta^\tau)_2 = (\delta_1 \circ B\tau_1)_2 = \delta_2 T \circ L\tau_2$ and $(\delta_\tau)_2 = (\tau_1 L \circ T\delta_1)_2 = \tau_2 B \circ \delta_2$ (as stated, e.g., in [21]); 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 $\mathbf{C}$) [20].

Lemma 18. *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 [5, Proposition 7], which states the dual). It then follows from (11) 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 8), and the logical approach (Proposition 11). 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 (12) by construction, so that the two approaches coincide (as already seen in the concrete examples).

Theorem 19. *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 8, together with δ, τ as defined in the proof of Proposition 11, satisfies* (12)*. 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 (12), *i.e.*, $\delta^\tau \circ \kappa = \delta_\tau$, we first compute, following (6),

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

Hence, we need to show that

$$\delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2) \rangle = t^{A \times X + 1} \circ \mathrm{st} \circ T(\delta_X) \quad (13)$$

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 (13), 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 (13). For $(a, x) \in A \times X$, we have:

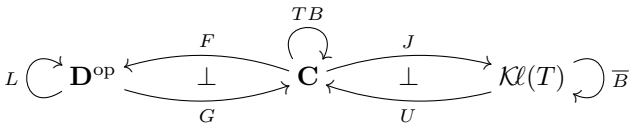
$$\begin{aligned}
&(\delta_X \circ (\mathrm{id} \times (t^X \circ \mathrm{st})^A) \circ (t \times \mathrm{st}) \circ \langle T(\pi_1), T(\pi_2)\rangle(S))(a, x) \\
&= (((t^X \circ \mathrm{st})^A \circ \mathrm{st})(T(\pi_2)(S)))(a)(x) \\
&= (((t^X)^A \circ \mathrm{st}^A \circ \mathrm{st})(T(\pi_2)(S)))(a)(x)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

which also matches the right-hand side, hence we obtain (13) as desired.

Since (12) is satisfied, it follows from Theorem 17 that $\mathsf{e} \circ \mathsf{em}_c = \mathsf{log}_c$. Since δ is an iso, e is an iso as well by Lemma 18. □

6.2 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 B, L and a monad T as follows:



together with:

1. An initial algebra $\beta\colon B(\Psi) \xrightarrow{\cong} \Psi$.
2. A $\mathcal{K}\ell$-law $\lambda\colon BT \Rightarrow TB$, or equivalently, an extension $\overline{B}$ of B.
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 $\delta\colon BG \Rightarrow GL$.
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 15. Since δ is a step, we obtain the following unique coalgebra-to-algebra morphism k from the initial B-algebra:

$$\begin{array}{ccccc}
\Psi & & \xrightarrow{\quad\quad\mathsf{k}\quad\quad} & & G(\Phi) \\
{\scriptstyle\beta^{-1}}\downarrow & & & & \uparrow{\scriptstyle G(\alpha^{-1})} \\
B(\Psi) & \xrightarrow{B(\mathsf{k})} & BG(\Phi) & \xrightarrow{\delta} & GL(\Phi)
\end{array} \tag{14}$$

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 $\overline{\mathsf{k}} = \tau_\Phi \circ T(\mathsf{k})\colon T(\Psi) \to G(\Phi)$. We have the following analogue of Theorem 17.

Lemma 20. *The distributive law λ commutes with the logics in* (6)*, as in:*

$$\begin{array}{ccc} BTG & \xrightarrow{\lambda G} & TBG \\ & {\scriptstyle \delta^\tau}\searrow \quad \swarrow{\scriptstyle \delta_\tau} & \\ & GL & \end{array} \tag{15}$$

iff there is a natural transformation $\varrho\colon L\widehat{F} \Rightarrow \widehat{F}\overline{B}$ given by $\varrho J = \delta$ — where the functor $\widehat{F}\colon \mathcal{K}\ell(T) \to \mathbf{D}^{\mathrm{op}}$ is the extension corresponding to τ (Lemma 15).

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

$$\begin{array}{ccccc} LF(Y) & \xrightarrow{\delta} & FB(Y) & \xrightarrow{\tau B} & FTB(Y) \\ {\scriptstyle L\tau}\downarrow & & & & \downarrow{\scriptstyle F\lambda} \\ LFT(Y) & & \xrightarrow[\delta T]{} & & FBT(Y) \\ {\scriptstyle LF(f)}\downarrow & & & & \downarrow{\scriptstyle FB(f)} \\ LF(X) & & \xrightarrow[\delta]{} & & FB(X) \end{array}$$

The lower rectangle commutes by naturality, the upper is equivalent to (15). Hence, (15) implies naturality. Conversely, if ϱ is natural, then the upper rectangle commutes for each Y by taking $f = \mathrm{id}_{TY}$ (the identity map in $\mathbf{C}$). □

Theorem 21. *If the equivalent conditions in Lemma 20 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.*

$$\begin{array}{ccc} TBT(\Psi) & \xrightarrow{TB(\overline{\mathsf{k}})} & TBG(\Phi) \\ {\scriptstyle \ell_{\mathrm{kl}}}\downarrow & & \downarrow{\scriptstyle \ell_{\log}} \\ T(\Psi) & \xrightarrow{\overline{\mathsf{k}}} & G(\Phi) \end{array} \qquad\qquad \begin{array}{ccc} & X & \\ & {\scriptstyle \mathsf{kl}_c}\swarrow \quad \searrow{\scriptstyle \mathsf{log}_c} & \\ T(\Psi) & \xrightarrow{\overline{\mathsf{k}}} & G(\Phi) \end{array}$$

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.

$$\begin{array}{ccccccc} TBT(\Psi) & \xrightarrow{TBT(\mathsf{k})} & TBTG(\Phi) & & \xrightarrow{\qquad TB\tau \qquad} & & TBG(\Phi) \\ \downarrow{\scriptstyle T\lambda} & & \downarrow{\scriptstyle T\lambda G} & & & & \downarrow{\scriptstyle T\delta} \\ TTB(\Psi) & \xrightarrow{TTB(\mathsf{k})} & TTBG(\Phi) & \xrightarrow{TT\delta} & TTGL(\Phi) & \xrightarrow{T\tau L} & TGL(\Phi) \\ {\scriptstyle \ell_{\mathrm{kl}}}\downarrow{\scriptstyle \mu B} & & \downarrow{\scriptstyle \mu BG} & & \downarrow{\scriptstyle \mu GL} & & {\scriptstyle \tau L}\downarrow \quad {\scriptstyle \ell_{\log}} \\ TB(\Psi) & \xrightarrow{TB(\mathsf{k})} & TBG(\Phi) & \xrightarrow{T\delta} & TGL(\Phi) & \xrightarrow{\tau L} & GL(\Phi) \\ \downarrow{\scriptstyle T(\beta)} & & & & \downarrow{\scriptstyle TG(\alpha^{-1})} & & {\scriptstyle G(\alpha^{-1})}\downarrow \\ T(\Psi) & & \xrightarrow[T(\mathsf{k})]{} & & TG(\Phi) & \xrightarrow[\tau]{} & G(\Phi) \end{array}$$

Everything commutes: the upper right rectangle by assumption (15), 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 (14) and the rest by naturality. $\square$

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 18, 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 22 below. In the comparison between Eilenberg-Moore and Kleisli traces [17], a similar difficulty arises: it is unclear under what conditions the map from the final coalgebra in Kleisli to the final coalgebra in Eilenberg-Moore obtained there is mono (and hence, if Eilenberg-Moore trace equivalence implies Kleisli trace equivalence).

Example 22. We give an example where $\delta\colon BG \Rightarrow GL$ is monic and (15) commutes, but where nevertheless logical equivalence is stronger than 'Kleisli' trace equivalence. Let $\mathbf{C} = \mathbf{D} = \mathbf{Sets}$, $F = G = 2^-$, $B = L = (A \times -) + 1$, $T = \mathcal{P}$, $\tau\colon \mathcal{P}2^- \Rightarrow 2^-$ 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 13). Notice that δ indeed has monic components.

Let $\lambda\colon BT \Rightarrow TB$ be the distributive law from [12], given by $\lambda_X(a, S) = \{(a, x) \mid x \in S\}$ and $\lambda(*) = \{*\}$. Then (15) is satisfied:

$$\begin{array}{ccccc} A \times \mathcal{P}(2^X) + 1 & & \xrightarrow{\quad\lambda\quad} & & \mathcal{P}(A \times 2^X + 1) \\ {\scriptstyle \mathrm{id} \times \tau + 1}\downarrow & & & & \downarrow{\scriptstyle \mathcal{P}(\delta)} \\ A \times 2^X + 1 & \xrightarrow{\delta} & 2^{A \times X + 1} & \xleftarrow{\tau} & \mathcal{P}(2^{A \times X + 1}) \end{array}$$

It is straightforward to check that this commutes. However, given a coalgebra $f\colon X \to TB(X)$, the induced logical semantics $\mathsf{log}\colon X \to 2^{A^*}$ is: $\mathsf{log}(x)(w) = \top$ iff $* \in f(x)$ or $\exists a \in A, v \in A^*, y \in X. w = av \wedge (a, y) \in f(x) \wedge \mathsf{log}(y)(v) = \top$. 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 [12].

Cîrstea [8] compares logical traces to a 'path-based semantics', which resembles the Kleisli approach (as well as [22]) but does not require a final $\overline{B}$-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 [12]. Cîrstea shows that, with $\Omega = T(1)$, $t = \mu_1\colon TT(1) \to T(1)$ and δ from the proof of Proposition 13 (assuming $T1$ to have enough structure to define that logic), the triangle (15) commutes (see [8, Lemma 5.12]).

7 Completely Iterative Algebras

In this paper, we constructed several corecursive algebras. We briefly show that they all satisfy the following stronger property [26].

Definition 23. *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.*

$$\begin{array}{ccc} X & \xrightarrow{\quad f \quad} & A \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle [\mathrm{id},a]} \\ A + HX & \xrightarrow[A+Hf]{} & A + HA \end{array}$$

Following [14,26], we have two ways of constructing such algebras.

Proposition 24.

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 5, 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.

8 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 [19], 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 [28,21]. 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 Eilenberg-Moore/logic and Kleisli/logic comparisons (Section 6) seem to share certain aspects (the conditions look very similar), but so far we have been unable to derive a general perspective on such comparisons that covers both, and possibly also the Eilenberg-Moore/Kleisli comparison of [17].

Acknowledgement. We are grateful to the anonymous referees 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. V. Capretta, T. Uustalu, and V. Vene. Recursive coalgebras from comonads. *Inf. Comput.*, 204(4):437–468, 2006.
6. 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.
7. L-T. Chen and A. Jung. On a categorical framework for coalgebraic modal logic. *Electr. Notes Theor. Comput. Sci*, 308:109–128, 2014.
8. C. Cîrstea. A coalgebraic approach to quantitative linear time logics. *CoRR*, abs/1612.07844, 2016.
9. A. Eppendahl. Coalgebra-to-algebra morphisms. *Electr. Notes Theor. Comput. Sci*, 29:42–49, 1999.
10. J. Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1988.
11. I. Hasuo. Generic weakest precondition semantics from monads enriched with order. *Theor. Comput. Sci.*, 604:2–29, 2015.
12. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Log. Meth. Comp. Sci.*, 3(4), 2007.
13. W. Hino, H. Kobayashi, I. Hasuo, and B. Jacobs. Healthiness from duality. In *Logic in Computer Science*. IEEE, Computer Science Press, 2016.
14. 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.
15. 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.
16. B. Jacobs. A recipe for state and effect triangles. *Log. Meth. Comp. Sci.*, 13(2), 2017. See `https://lmcs.episciences.org/3660`.
17. B. Jacobs, A. Silva, and A. Sokolova. Trace semantics via determinization. *Journ. of Computer and System Sci.*, 81(5):859–879, 2015.
18. 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.
19. 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.

20. 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.
21. 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`.
22. 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.
23. T. Leinster. *Higher Operads, Higher Categories*, volume 298 of *London Mathematical Society Lecture Notes*. Cambridge University Press, 2004.
24. 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.
25. 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.
26. Stefan Milius. Completely iterative algebras and completely iterative monads. *Inf. Comput*, 196(1):1–41, 2005.
27. 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.
28. J. Rot. Coalgebraic minimization of automata by initiality and finality. *Elect. Notes in Theor. Comp. Sci.*, 325:253–276, 2016.
29. A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing determinization from automata to coalgebras. *Log. Meth. Comp. Sci.*, 9(1), 2013.
30. 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.