2011

Generic Trace Logics

Christian Kissig
University of Leicester

Alexander Kurz
Chapman University, akurz@chapman.edu

Recommended Citation
Christian Kissig, Alexander Kurz: Generic Trace Logics. CoRR abs/1103.3239 (2011)

Generic Trace Logics

Christian Kissig and Alexander Kurz
University of Leicester, Department of Computer Science

June 20, 2018

Abstract

We combine previous work on coalgebraic logic with the coalgebraic traces semantics of Hasuo, Jacobs, and Sokolova.

1 Introduction

The coalgebraic approach to modal logic has been pursued successfully over the last years. The basic ideas (see eg [16, 17, 19, 11]), are the following.

- A T-*coalgebra*, consisting of a carrier X and a 'next-step' map $\xi : X \to TX$, represents a transition system. For example, with $\mathcal{P}X$ the set of finite subsets of X and Act a set of actions, $X \to \mathcal{P}(Act \times X)$ is a labelled transition system.
- Any particular choice of T yields a canonical notion of T-*bisimilarity*. For example, for $X \to \mathcal{P}(Act \times X)$ we obtain the Milner-Park notion of bisimilarity [1] whereas for $X \to \mathcal{D}(Act \times X)$, with $\mathcal{D}X$ denoting the set of probability distributions on X, we obtain the notion of bisimilarity described in [3].
- Moreover, for any choice of T, we can find a logic for T-coalgebras which is expressive (ie distinguishes non-bisimilar states) and comes with a complete calculus. These logics are modal logics in the sense that formulas are invariant under T-bisimilarity.

The work on coalgebraic logic so far is focused on T-bisimilarity.

In parallel, Jacobs and collaborators [6, 5, 4] showed that coalgebras not only provide a framework for bisimilarity, but also for trace semantics:

- A (B, T)-coalgebra $X \to BTX$ is now given wrt a 'transition type' T and a 'branching type' B. For example, with $BX = \mathcal{P}X$ and $TX = \{*\}+Act\times X$, a $X \to \mathcal{P}(\{*\}+Act\times X)$ is a non-deterministic automaton.
- Different choices of B yield different notions of trace semantics. With $B = \mathcal{P}$, the trace semantics of $X \to \mathcal{P}(\{*\} + Act \times X)$ identifies states that accept the same language. With $B = \mathcal{D}$, the trace semantics of $X \to \mathcal{D}(\{*\} + Act \times X)$ identifies states that accept the same (finite) traces with the same probabilities.

The work of Jacobs et al is build on several assumptions, which limit the generality of the definition of trace semantics. For instance, it is not possible to define the trace semantics of finitely branching transition systems.

Results In this paper, we reconsider the definition of trace semantics in the category of algebras for the branching type B. This allows us to includes the often occuring finite non-determinism and finitely graded branching.

Moreover we propose a generic definition of coalgebraic logics characterising states up to trace equivalence. Our definition of trace logics is build upon a dual adjunction on the category of algebras for the branching type, and matches the definition of coalgebraic modal logics for T-bisimulation.

Structure of the paper After reviewing material known from the literature, Section 4.3 introduces trace semantcis in the category of Eilenberg-Moore algebras of the monad B describing the branching type. Section 4.4 describes trace logics using the adjunction induced by the closed structure a the commutative monad B. Section 4.5 explains how to define logics via predicate lifting, a notion known set-coalgebras, which is adapted to our setting. Section 4.6 introduces the notion of a generic trace logic and uses it to prove a particular instance to be sound, complete, and expressive.

Acknowledgements We would like to thank Ichiro Hasuo and Bart Jacobs.

2 Two Examples

Consider $\gamma : X \to \mathcal{P}_\omega(\{*\} + Act \times X)$. (X, γ) is a finitely non-deterministic automaton. Indeed, with 1 as $\{*\}$ and $+$ as (disjoint) union, we read $(a, x') \in \gamma(x)$ as x can input a and go to x' and we read $* \in \gamma(x)$ as x is an accepting state.

Now consider a logic

$$\phi ::= 0 \mid \surd \mid \phi \vee \phi \mid \langle a \rangle \phi \tag{1}$$

with compositional semantics

$$x \not\Vdash 0 \tag{2}$$
$$x \Vdash \surd \Leftrightarrow \; * \in \gamma(x) \tag{3}$$
$$x \Vdash \phi \vee \psi \;\Leftrightarrow\; x \Vdash \phi \text{ or } x \Vdash \psi \tag{4}$$
$$x \Vdash \langle a \rangle \phi \;\Leftrightarrow\; (a, x') \in \gamma(x) \text{ and } x' \Vdash \phi \tag{5}$$

and as axiomatisation the usual laws for falsum (0) and disjunction ($\vee$) plus the axioms

$$\langle a \rangle 0 = 0 \qquad \langle a \rangle (\phi \vee \psi) = \langle a \rangle \phi \vee \langle a \rangle \psi \tag{6}$$

Note that this implies the typical axiom we would expect for trace logics

$$\langle a \rangle (\langle b \rangle \phi \vee \langle c \rangle \psi) = \langle a \rangle \langle b \rangle \phi \vee \langle a \rangle \langle c \rangle \psi \tag{7}$$

Our development will not only provide a generic proof for the fact that this logic is sound, complete and expressive, but also provide conceptual explanations for why we can have falsum and disjunction, but not negation and conjunction.

To see that the interaction of the modal operators $\langle a \rangle$ with the propositional operators (0,$\vee$) is subtle, consider as a second example $\gamma : X \to \mathcal{D}(\{*\} + Act \times X)$ where $\mathcal{D}Y$ is the set of finitely supported discrete probability distributions on Y. $\gamma(x, *) \in [0, 1]$ is the probability of terminating successfully and $\gamma(x, a, x') \in [0, 1]$ is the probability of continuing with a and transiting to x'. Two states x, x' are trace equivalent if (inventing an adhoc notation similar to the logic above)

$$x \Vdash p \cdot \langle a_0 \rangle \ldots \langle a_n \rangle \surd \;\Leftrightarrow\; x' \Vdash p \cdot \langle a_0 \rangle \ldots \langle a_n \rangle \surd \tag{8}$$

which we read as stating that the probability of x (and x') to terminate successfully after the sequence $a_0 \ldots a_n$ is p.

The notation in (8) indicates that there must be a definition of logic, semantics, axiomatisation paralleling the example of non-determinstic automata and we will show how to obtain in a systematic fashion from the functors involved.

3 Preliminaries

3.1 Monads, Algebras and Coalgebras

Definition 3.1. A *coalgebra* for an endofunctor T on a category $\mathcal{C}$ is a morphism $\gamma : X \to TX$ for an object X of $\mathcal{C}$, that we call γ's domain. A T-coalgebra morphism between coalgebras $\gamma : X \to TX$ and $\delta : Y \to TY$ is a morphism $f : X \to Y$ such that $Tf \circ \gamma = \delta \circ f$ commutes. Dually, a T-algebra is an arrow $\alpha : TX \to X$.

Definition 3.2. A *monad* on *Set* is an endofunctor $B : Set \to Set$ with natural transformations $\eta : Id \Rightarrow B$ and $\mu : BB \Rightarrow B$ such that $\mu \circ \eta_T = id_T = \mu \circ T\eta$ and $\mu \circ \mu_T = \mu \circ T\mu$. If B preserves filtered colimits, the monads is called *finitary*.

Example 3.3 (finitary monads). 1. The finite powerset $\mathcal{P}_\omega$, equipped with the singleton map $\{(-)\}$ and set-union.

2. The bag functor $\mathcal{B}$ takes a set X to the set $(\mathbb{N}^X)_\omega$ of its finite multisets, and functions $f : X \to Y$ to multiset-functions $\mathcal{B}f : \mathcal{B}X \to \mathcal{B}Y$ taking multisets $m \in (\mathbb{N}^X)_\omega$ to $\lambda y. \sum_{x \in f^{-1}(y)} m(x)$.

3. A (sub-)distribution of a set X is a function $d : X \to [0,1]$ such that $\sum_{x\in X} d(x) = 1$ ($\sum_{x\in X} d(x) \leq 1$). The (sub-)distribution functor $\mathcal{D}_{=1}$ ($\mathcal{D}_{\leq 1}$) takes a set X to the set of its (sub-)distributions, and functions $f : X \to Y$ to $\lambda m.\lambda y. \sum_{x\in f^{-1}(y)} m(x)$. For the sake of a brevity we write both, $\mathcal{D}_{=1}$ and $\mathcal{D}_{\leq 1}$, as $\mathcal{D}$ when it is clear from context, which functor we mean.

   For each X we can define functions

$$\mu_X(d' \in \mathcal{D}^2X)(x) := \textstyle\sum_{d\in\mathcal{D}X} d'(d) \cdot d(x) \qquad \eta_X(x) := \lambda y. \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise} \end{cases}$$

   μ and η are transformations natural in X and form with B a monad.

4. All of the above are examples of functors which take a set X into the set $(\mathcal{S}^X)_\omega$ of evaluations of X into a semiring $\mathcal{S}$ with finite support, and functions $f : X \to Y$ into functions $(\mathcal{S}^X)_\omega \to (\mathcal{S}^Y)_\omega$ such that $m \in (\mathcal{S}^X)_\omega \mapsto \lambda y. \sum_{x\in f^{-1}(x)} m(x)$. For $\mathcal{P}_\omega$ the semiring is the boolean algebra $\langle\{\top, \bot\}, \wedge, \vee, \top, \bot\rangle$, and for $\mathcal{B}$ the semiring are the natural numbers $\langle\mathbb{N}, +, *, 0, 1\rangle$.

5. If we take for $\mathcal{S}$ the real numbers with addition and multiplication, then the category of algebras for the semiring monad is (isomorphic to) the category of vector-spaces. See Semadeni [20] for more on this perspective. More generally, if the semiring does not happen to be a field, the category of algebras for the monad is known as the category of modules for the semiring.

6. Another example of a semiring monad uses the min-semiring $\langle\mathbb{N} \cup \{\infty\}, min, +, \infty, 0\rangle$ of natural numbers augmented with a top element, ∞, with an idempotent additive operation, min, and a commutative multiplicative operation, $+$, such that ∞ is neutral wrt min and 0 wrt $+$, and 0 absorbs wrt min.

7. Another example of semiring monads can be found in the weighted automata of Rutten [18], where the stream behaviour is an instance of the finite trace semantics presented in this paper.

An (Eilenberg-Moore-) algebra for a monad B is an algebra for the functor B satisfying additionally $\alpha \circ \mu_X = \alpha \circ B\alpha$ and $\alpha \circ \eta_X = id_X$. The algebras for a monad B form a category, the Eilenberg-Moore category B-Alg. $U : B\text{-}Alg \to \mathcal{C}$ maps an algebra to its carrier. U has a left adjoint F and we write $\eta : Id \to UF$ and $\varepsilon : FU \to Id$ for the unit and counit of the adjunction. Recall that $UF = B$ and $F\varepsilon_{UX} = \mu_X$.

Each monad admits and initial and a final B-algebra, respectively $\langle B\emptyset, \mu_\emptyset B^2\emptyset \to B\emptyset\rangle$ and $\langle\{*\}, (\lambda.*) : B\{*\} \to \{*\}\rangle$. Synonymously, we denote by 1 a singleton set, when the domain (*Set* or B-Alg) is clear from context.

For our definition of generic trace logics, it may be useful when B-Alg is *closed* in the sense that homsets in B-Alg have B-algebra structure themselves. Kock [9] showed that this is true for commutative monads.

Definition 3.4 (Strength Laws). A strength law for a monad B is a transformation $st_{X,Y} := BX \times Y \to B(X \times Y)$ natural in X and Y and commutes with the monad's unit and multiplication law such that $st_{X,Y} \circ (\eta_X \times id_Y) = \eta_{X\times Y}$ and $\mu_{X\times Y} \circ Bst_{X,Y} \circ st_{BX,Y} = st_{X\times Y} \circ (\mu_X \times id_Y)$.

A double strength law is a natural transformation given as the diagonal $dst_{X,Y} : BX \times BY \to B(X \times Y)$ of $\mu_{X\times Y} \circ Bst_{Y,X} \circ st_{X,BY} = \mu_{X\times Y} \circ Bst_{X,Y} \circ st_{Y,BX}$, given it exists consistently.

A monad is commutative if it has a double strength law.

The proof of the following can be found in [9].

Proposition 3.5. *The Eilenberg-Moore category of a commutative monad is closed.*

3.2 The Kleisli Construction and Functor Liftings

Definition 3.6 (Kleisli-Categories)**.** The Kleisli-category KlB of a monad B on $\mathcal{C}$ has as objects the objects of $\mathcal{C}$ and arrows $f : X \to Y$ are the arrows $f : X \to BY$ in $\mathcal{C}$. The identity is given by $\eta : X \to BX$ and composition of $f : X \to Y$ and $g : Y \to Z$ in KlB is given by $g \circ f := \mu_Z \circ Bg \circ f$.

The adjunction $F' \dashv U' : \mathcal{C} \to KlB$ is defined such that for all sets X, $F'X := X$, all functions $f : X \to Y$ in Set, $F'f := \eta_Y \circ f$, and for all objects X in KlB, $U'X := BX$ and for all morphisms $f : X \to Y$, $U'f := \mu_Y \circ Bf$.

Example 3.7.
1. The Kleisli-category for the powerset monad $\mathcal{P}$ is Rel, the category of sets as objects and relations as morphisms.
2. The Kleisli-category for the semiring monad $(\mathcal{S}^{(-)})_\omega$ is the category of free (left) modules for the semiring $\mathcal{S}$.

A coalgebra $\gamma : X \to BTX$ in Set is a morphisms $X \to TX$ in KlB. In order to exhibit γ as a coalgebra in KlB and to have coalgebra morphisms, one defines the lifting of Set-functors T to KlB. The lifted functor $\overline{T}$ makes $FT = \overline{T}F$ commute. The existence of the functor lifting is equivalent to the existence of a distributive law.

Definition 3.8 (Distributive Laws)**.** A distributive law for a monad B and a functor T is a natural transformation $\pi : TB \Rightarrow BT$ such that $\pi \circ T\eta = \eta_T$ and $\pi \circ T\mu = \mu_T \circ B\pi \circ \pi_B$ commute.

Example 3.9. Let $T(-) := \{*\} + Act \times (-)$ be a Set-functor for a fixed set Act. With each of the monads in Example 3.3 T has a distributive law.

1. $\pi : T\mathcal{P} \to \mathcal{P}T$: $\pi_X(*) := \{*\}$, $\pi_X(a, Y \subseteq X) := \{(a, x) \mid x \in Y\}$.
2. $\pi : T\mathcal{B} \to \mathcal{B}T$: $\pi_X(*) := \eta_{\{*\}+Act\times X}(*)$, and $\pi_X(a, m)(a, x) := \{(a, x) \mapsto m(x), (b, x) \mapsto 0, * \mapsto 0 \mid a \in Act, b \in Act, b \neq a, x \in X\}$
3. $\pi : T\mathcal{D} \to \mathcal{D}T$: $\pi_X(*) := \eta_{\{*\}+Act\times X}(*)$, and $\pi_X(a, d) := \{(a, x) \mapsto d(x), (b, x) \mapsto 0, * \mapsto 0 \mid a \in Act, b \in Act, b \neq a, x \in X\}$ where $\mathcal{D} \in \{\mathcal{D}_{\leq 1}, \mathcal{D}_{=1}\}$

Definition 3.10 (Functor Lifting by Distributive Law)**.** Given a distributive law $\pi : TB \to BT$ we can define $\overline{T}$ on objects $\overline{T}X := TX$ and on morphisms $\overline{T}(f : X \to Y) := \pi_Y \circ Tf$

There is a full and faithful functor $K : KlB \to B\text{-}Alg$ mapping X to the free algebra over X, see [15]. In other words, we can think of KlB as the full subcategory of $B\text{-}Alg$ consisiting of the free algebras.

4 Coalgebraic Logic for Trace Semantics

In this section we show how to set up trace logics in a coalgebraic framework. But first we review some basic of coalgebraic logic (more can be found in [11]) and the fundamentals of generic trace semantics [6].

4.1 A Brief Review of Logics for T-Bisimilarity

Suppose we are looking for a logic for T-coalgebras built upon classical propositional logic. Such a logic would be based on Boolean algebras which precisely capture the axioms of propositional logic. Then, in the same way as T is a functor $Set \to Set$ on the models (coalgebras) side, the logic will contain modalities given in terms of a functor $L : BA \to BA$ on the category BA of Boolean algebra. The situation is depicted in

$$T \circlearrowright Set \underset{S}{\overset{Q}{\rightleftarrows}} BA^{op} \circlearrowleft L \qquad (\perp) \tag{9}$$

Q contravariantly takes sets X to their powersets 2^X and S maps a Boolean algebra to the set of maximal consistent theories (ultrafilters). For example, if $T = \mathcal{P}$ we may define L by saying that LA is the Boolean algebra generated by $\Diamond\phi, \phi \in A$, modulo the axioms

$$\Diamond 0 = 0 \qquad \Diamond(\phi \vee \psi) = \Diamond\phi \vee \Diamond\psi \tag{10}$$

Note how this definition of L captures the usual modal logic for (unlabelled) transition systems. The semantics of the logic is given by a map

$$\delta_X : LQX \to QTX \tag{11}$$

In the example we define $\delta_X(\Diamond\phi) = \{\psi \in \mathcal{P}TX \mid \phi \cap \psi \neq \emptyset\}$ in order to capture that $\Diamond\phi$ holds if the set 'of successors' ψ satisfies $\phi \cap \psi \neq \emptyset$. Finally, (L, δ) gives rise to a logic in the usual sense as follows. The set of formulas of the logic is the carrier of the initial L-algebra. The semantics of a formula wrt to a coalgebra $X \to TX$ is given by the unique homomorphism from the initial L-algebra $LI \to I$ as in:

$$\begin{array}{ccccc} LI & & \longrightarrow & & I \\ \downarrow{\scriptstyle L(\llbracket\cdot\rrbracket)} & & & & \downarrow{\scriptstyle \llbracket\cdot\rrbracket} \\ LQX & \xrightarrow{\delta_X} & QTX & \xrightarrow{Q\gamma} & QX \end{array} \tag{12}$$

Theorem 4.1. *Any (L, δ) with δ as in (11) gives rise to a logic for T-coalgebras. The semantics $\llbracket\cdot\rrbracket$ as in (12) is invariant under T-bisimilarity. The logic is expressive for (finite) coalgebras, if δ_X is onto for (finite) X and the equational logic given by the axioms defining L is complete if δ_X is injective for all X.*

Suppose we are given T, how can we find a logic (L, δ)? Two answers:

Remark 4.2.
1. Moss [16] takes LA to be the free BA generated by TUA where UA is the underlying set of A. A complete calculus has been given in [10].
2. The standard modal logic for $T = \mathcal{P}$ above arises from $LA = QTSA$ on finite A and extending continuously to all of BA [13]. It is always complete.

Both logics are expressive. A detailed comparison has been given in [12].

4.2 A Brief Review of Finite Trace Semantics

The basic construction Consider a coalgebra $X \to BTX$, the running example being $B = \mathcal{P}$ and $TX = \{*\} + Act \times X$ as discussed in Section 2. The set of traces will be the carrier of the initial T-algebra given by the colimit (or union) of the sequence

$$\emptyset \xhookrightarrow{\emptyset} T\emptyset \xhookrightarrow{T\emptyset} T^2\emptyset \hookrightarrow \cdots \qquad T^\omega\emptyset \tag{13}$$

In the example $T^n\emptyset = \{a_1 \ldots a_n \mid a_i \in Act\}$ and $T^\omega\emptyset = Act^*$, ie the set of finite words over Act. The set of traces of length n will be given by a map

$$tr_n : X \to BT^n\emptyset \tag{14}$$

In the example, $tr_n(x)$ is the set of traces of length n that lead from x to an accepting state. To compute it, we need the following ingredients.

Assumption 1.

- a map $\mu_X : BBX \to BX$ (for this we assume that B is a monad)
- a map $\pi_X : TBX \to BTX$ (for this we assume that π is a distributive law)
- an algebra morphism $e : A \to F\emptyset$ from any B-algebra A into $F\emptyset$.[1]

The maps tr_n then arise from taking n steps of γ, eg in the case $n = 2$, as

$$X \xrightarrow{\gamma} BTX \xrightarrow{BT\gamma} BTBTX \xrightarrow{BTBTe} BTBTB\emptyset \xrightarrow{p} BBBTT\emptyset \xrightarrow{m} BT^2\emptyset$$

(p stands for 3 applications of π and m for 2 applications of μ.)

Definition 4.3. Two states $x, y \in X$ of a coalgebra $X \to BTX$ are trace equivalent if $tr_n(x) = tr_n(y)$ for all $n < \omega$.

[1]This means that we assume from hereon $B\emptyset \neq \emptyset$. Also note that in all our examples B is a commutative monad, hence $B\emptyset \neq \emptyset$ implies $B\emptyset = 1$, so that $F\emptyset$ is the final algebra.

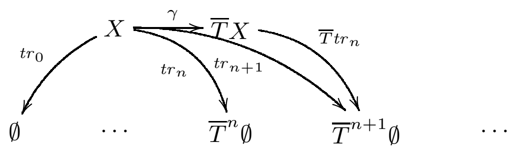
For the purposed of the current paper, we consider this the essence of the trace semantics of [5]. But [5] do much more and, in particular, they show that under additional assumptions the trace semantics can be given by a final coalgebra in the Kleisli category.

Trace semantics in the Kleisli category [5] show not only that the ingredients of a monad B and a distributive law $TB \to BT$ give rise to trace semantics, they also show that it can be elegantly formulated in the so-called Kleisli category of the monad B (see Section 3). The objects in the Kleisli category are the same as in Set, but arrows $X \to Y$ in KlB are maps $X \to BY$ in Set. In case of the powerset functor $B = \mathcal{P}$, KlB is the category of sets with relations as arrows.

The distributive law $TB \to BT$ gives rise to a lifting of $T : Set \to Set$ to $\overline{T} : KlB \to KlB$. The definition of tr_n can then be defined inductively as

$$tr_{n+1} = \overline{T}(tr_n) \circ \gamma \tag{15}$$

where we assume a morphism $tr_0 : X \to 0$ in the base case. The following diagram illustrates the above definition



(16)

Furthermore, under conditions for which we refer to [5], the final $\overline{T}$-coalgebra Z exists.[2] Therefore, with the notation of Definition 4.3, there is a map $tr : X \to BZ$ with the property

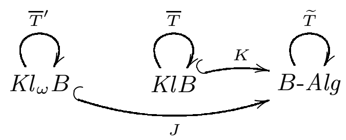
$$tr(x) = tr(y) \Leftrightarrow tr_n(x) = tr_n(y)$$

for all $n < \omega$. Thus, the trace semantics via the final coalgebra (if it exists) in the Kleisli-category is equivalent to the one of Definition 4.3. The advantage of the trace semantics via the final coalgebra in the Kleisli-category is that it gives a coinductive account of trace semantics. The disadvantage is that it excludes some natural examples such as *finite* powersets or multisets. The next section shows that these examples can be treated via final coalgebras if we move from the Kleisli-category to the category of algebras for the monad.

4.3 Trace Semantics in the Eilenberg-Moore Category

In this section we propose to move the trace semantics from the Kleisli-category KlB to the category $B\text{-}Alg$ of Eilenberg-Moore-algebras. There are at least two reasons why this of interest. The first is that the duality we will exploit for the logic takes place in $B\text{-}Alg$. The second is that, in general, the limit of Diagram (16) is not a free B-algebra and hence not in KlB, but it always exists in $B\text{-}Alg$.

Let K denote the functor which embeds KlB into $B\text{-}Alg$. Our first task is to extend $\overline{T} : KlB \to KlB$ to $\widetilde{T} : B\text{-}Alg \to B\text{-}Alg$ so that $\widetilde{T}K \cong K\overline{T}$ (hence $\widetilde{T}F \cong FT$).



(17)

On the full subcategory of free algebras we can define $\widetilde{T}FX = K\overline{T}X = FTX$. To extend this to arbitrary algebras A recall first that any $A \in B\text{-}Alg$ is a coequaliser of $FU\varepsilon_A, \varepsilon_{FUA} : FUFUA \to FUA$. We then define $\widetilde{T}A$ as the coequaliser of $\widetilde{T}FU\varepsilon_A$ and $\widetilde{T}\varepsilon_{FUA}$. It can be shown that $\widetilde{T}$ is the left Kan-extension of $K\overline{T}$ along K.

Example 4.4. Let $B = \mathcal{P}_\omega$ and $T = \{\surd\} + Act \times Id$. Then $\widetilde{T}A \cong F1 + Act \cdot A$. Indeed, by definition, we have $\widetilde{T}FX = FTX \cong F1 + Act \cdot FX$. Now the claim follows from the fact that the functor $F1 + Act \cdot Id$, being a coproduct, preserves coequalisers.

[2]Moreover, [5] prove the beautiful result that show that the final $\overline{T}$-coalgebra is given by the initial T-algebra with the carrier $T^\omega\emptyset$ as in (13).

It is convenient for us to make the following assumptions.

Assumption 2. $\mathcal{B} : Set \to Set$ is a finitary commutative monad with $B\emptyset \neq \emptyset$ and $T : Set \to Set$ is a finitary functor with a distributive law $TB \to BT$.

Remark 4.5. If $\mathcal{B}$ and T are finitary, then $\widetilde{T}$ is determined by finitely generated free algebras, or, in other words, $\widetilde{T}$ preserves sifted (hence filtered) colimits [2] and falls within the framework considered in [14, 21]. For a functor $H : \mathcal{A} \to \mathcal{A}$ on a finitary algebraic category $\mathcal{A}$ to be strongly finitary means that H is determined by its action on finitely generated free algebras. More formally, H is a left Kan-extension of HK along K where K is the inclusion $\mathcal{A}_0 \to \mathcal{A}$ of the full subcategory $\mathcal{A}_0$ of finitely generated free algebras. A pleasant consequence is that all concrete calculations of some HA can be restricted to the case $A = Fn$, where F is the left adjoint of the forgetful functor $\mathcal{A} \to Set$ and n is finite. This will be exploited in the following for $\mathcal{A} = B\text{-}Alg$. Other consequences of our assumption then are:

- $F\emptyset$ is the initial and final object of $B\text{-}Alg$.
- The final $\widetilde{T}$ sequence converges after ω steps.

In a second step, we can now map a coalgebra $\gamma : X \to BTX$ (ie $\gamma : X \to \overline{T}X$) to $\widetilde{\gamma} : FX \to \widetilde{T}FX$ (ie $\widetilde{\gamma} : KX \to K\overline{T}X$). Thus $\widetilde{\gamma}$ is a coalgebra for a functor $\widetilde{T} : B\text{-}Alg \to B\text{-}Alg$. Moreover we observe that we can factor $tr_n : X \to BT^n\emptyset$ from Diagram (16) as

$$tr_n : X \to BX \cong UFX \xrightarrow{U\widetilde{tr}_n} \widetilde{T}^n F\emptyset \cong BT^n\emptyset$$

where we define $\widetilde{tr}_0$ via e as in Assumption 1 and $\widetilde{tr}_{n+1} = \widetilde{T}\widetilde{tr}_n \circ \widetilde{\gamma}$. Let us summarise this in a definition and a proposition.

Definition 4.6. Recall Assumption 2. For any coalgebra $\alpha : A \to \widetilde{T}A$ we define the trace semantics as follows. First, $\widetilde{tr} : A \to F\emptyset$ is given by finality; then, inductively $\widetilde{tr}_{n+1} = \widetilde{T}\widetilde{tr}_n \circ \widetilde{\gamma}$. This defines a cone on the final $\widetilde{T}$-sequence so we can define the trace semantics $\widetilde{tr} : A \to Z$, where $Z \to \widetilde{T}Z$ is the final $\widetilde{T}$-coalgebra. For a coalgebra $\gamma : X \to BTX$ we define $tr : X \to UZ$ as $U\widetilde{tr} \circ \eta_X$, where $\widetilde{tr}$ is the trace semantics of $\widetilde{\gamma} : FX \to \widetilde{T}FX$.

To emphasise that this definition agrees with the one of the previous subsection we state

Proposition 4.7. *Consider* $\gamma : X \to BTX$ *and* $\widetilde{\gamma} : FX \to FTX = \widetilde{T}FX$. *Then* $U\widetilde{tr}_n \circ \eta_X = tr_n$.

Thus, Z and $\widetilde{tr}$ and tr are just a convenient way to talk about the maps tr_n for all $n \in \mathbb{N}$ simultaneously. In particular, we have now again a coinductive account of trace semantics. This technique will give, for example, a short and conceptual proof of Theorem 4.16. Under Assumption 2, and if the final $\overline{T}$-coalgebra of [5] exists, then both the trace semantics in KlB and the trace semantics in $B\text{-}Alg$ are equivalent as both boil down to Definition 4.3. (Of course, this is due to the fact that the definition of $\widetilde{T}$ extends to all algebras the lifting $\overline{T}$ of T to $Kl(B)$.)

Remark 4.8. If $B\emptyset \neq 0$ then the sequence $(\widetilde{T}^n F\emptyset)_{n<\omega}$ is the finitary part of the final $\widetilde{T}$-sequence in $B\text{-}Alg$. Moreover, it follows from Remark 4.5 that if B is finitary, then the ω-limit $(\widetilde{T}^\omega F\emptyset)$ of the final sequence is the final $\widetilde{T}$-coalgebra. To summarise, in addition to the explanation of trace semantics as a final semantics in the Kleisli-category as in [5], we can also give a final semantics in the Eilenberg-Moore category. These two approaches are slightly different, for example, the approach of [5] works for $B = \mathcal{P}$ but not for $B = \mathcal{P}_\omega$, whereas for us it is more natural to work with $B = \mathcal{P}_\omega$ as we then have algebras with a finitary signature.

Example 4.9. Consider $B = \mathcal{P}_\omega$, $T = \{\surd\} + Act \times Id$. Then $\widetilde{T}(FX) = F\{\surd\} + Act \cdot FX$. We can identify $F\emptyset$ with $\{\emptyset\}$ and $\widetilde{T}^n(F\emptyset)$ with $\mathcal{P}_\omega(1 + Act + \dots Act^n)$. Thus, elements of $\widetilde{T}^n(F\emptyset)$ are finite sets of finite words $\langle a_1 \dots a_i\rangle$, $i \leq n$. As $F\emptyset$ is initial and final in $B\text{-}Alg$, the $\widetilde{T}^n(F\emptyset)$ are part of the initial and of the final $\widetilde{T}$-sequence. The projections $p_n^{n+1} : \widetilde{T}^{n+1}(F\emptyset) \to \widetilde{T}^n(F\emptyset)$ are finite-union-preserving maps determined by acting as the identity on singletons $\{\langle a_1 \dots a_i\rangle\}$ for $i \leq n$ and sending $\{\langle a_1 \dots a_{n+1}\rangle\}$ to $\emptyset$. The embeddings $e_{n+1}^n : \widetilde{T}^n(F\emptyset) \to \widetilde{T}^{n+1}(F\emptyset)$ are given by the obvious inclusions. Note that $p_n^{n+1} \circ e_{n+1}^n = id_n$. The colimit of the initial $\widetilde{T}$-sequence $(e_{n+1}^n)_{n<\omega}$ is given by all finite subsets of $Act^* = \coprod_{n<\omega} Act^n$. The limit of the final $\widetilde{T}$-sequence $(e_{n+1}^n)_{n<\omega}$ is given by all subsets of Act^*. Note that although all approximants $\widetilde{T}^n(F\emptyset)$ are free algebras, the limit $\mathcal{P}(Act^*)$ is not free in $B\text{-}Alg$ and hence does not appear in $Kl(\mathcal{P}_\omega)$.

4.4 Logics for Finite B-Traces

We develop logics for (B, T)-coalgebras with a semantic invariant under trace equivalence in analogy to coalgebraic modal logic for T-bisimulation.

Firstly we need a category carrying our logics. We have a number of possible replacements for BA in Diagram (9): distributive lattices for positive logic, Heyting algebras for intuitionistic logic, complete atomic Boolean algebras for infinitary logic. The minimal choice (without propositional operators) is Set itself as used for example by Klin in [8].

$$Set \underset{2^{(-)}}{\overset{2^{(-)}}{\rightleftarrows}} Set^{op} \qquad \bot \tag{18}$$

In the above situation, 2 takes the role of a schizophrenic object. Analogously we may choose a B-algebra Ω to replace 2. In most examples we have considered, $F1$ is a suitable choice, but for the moment we do not need to fix a choice.

Notation 4.10. If B is a commutative monad, we write Q for the contravariant endofunctor $[-, \Omega]$ on $B\text{-}Alg$ where Ω is for now an arbitrary but fixed object of 'truth values'.

$$B\text{-}Alg \underset{Q=[-,\Omega]}{\overset{Q=[-,\Omega]}{\rightleftarrows}} B\text{-}Alg^{op} \qquad \bot \tag{19}$$

Q_0 is the contravariant endofunctor $U[F-, \Omega] = Set(-, U\Omega)$. We have $UQA = Q_0UA$.

Example 4.11. When $B = \mathcal{P}_\omega$, $B\text{-}Alg = SLat$ is the category of (join) semi-lattices. For Ω we choose the two-element semi-lattice $F1 = \mathbb{2}$, so that $[-, F1]$ takes a semi-lattice A to the set of 'prime filters' over A. For future calculations, we record some facts about semi-lattices. First, for finite A, there are order-reversing bijections

$$A \underset{\log}{\overset{\exp}{\rightleftarrows}} [A, \mathbb{2}] \tag{20}$$

given by $\exp(a) = \lambda b.\neg(b \le a)$ and $\log(\phi) = \bigvee \phi^-$ where $\neg : \mathbb{2} \to \mathbb{2}$ is negation and $\phi^- = \{a \in A \mid \phi(a) = 0\}$. Another description of log goes as follows. Since ϕ preserves joins it has a right adjoint $\phi^\sharp$ and $\log(\phi) = \phi^\sharp(0)$. Second, if $A = FX$ with X not necessarily finite, we have the bijection

$$UQFX = U[FX, \mathbb{2}] \cong Set(X, 2) \cong \mathcal{P}X \tag{21}$$

which lifts to a semi-lattice isomorphism

$$QFX \cong (\mathcal{P}X, \emptyset, \cup) \tag{22}$$

mapping $\phi \in [FX, \mathbb{2}]$ to $\{x \in X \mid \phi(\{x\}) = 1\}$ and $S \subseteq X$ to the unique ϕ with $\phi(x) = 1 \Leftrightarrow x \in X$, or, equivalently, to $\lambda S' \in FX \,.\, S \cap S' \neq \emptyset$ (where we use (21) to identify S' with a subset of X). Taking now $X = n$ finite again, we obtain

$$QFn \cong Fn. \tag{23}$$

In this case it is more convenient to use exp and log to denote the order-preserving bijections

$$Fn \underset{\log}{\overset{\exp}{\rightleftarrows}} [Fn, \mathbb{2}] \tag{24}$$

given by $\log(\phi) = \{i \in n \mid \phi(\{i\}) = 1\}$ and $\exp(S) = \lambda S'.S \cap S' \neq \emptyset$ (where again we identify elements S, S' of Fn with subsets $S, S' \subseteq n$).

One can check that $\exp(\exp(a)) = \lambda\phi.\phi(a)$. It follows that $\exp \circ \exp : Id \to QQ$ is the unit of the adjunction (19), and, moreover, that the unit is an isomorphism on finite semi-lattices.[3] In case of $Fn \to QQFn$ we have for $S \subseteq n$ that $\exp(\exp(S))(\phi) = \log(\phi) \cap S \neq \emptyset$. The inverse

[3] This also follows from the fact that the adjunction (19) restricts to an equivalence on finite semi-lattices [7].

$QQFn \to Fn$ of $Fn \to QQFn$ maps $u : [Fn, \mathbb{2}] \to \mathbb{2}$ to $\log(\log(u)) = n \setminus \{i \in n \mid \exists \phi \,.\, u(\phi) = 0 \,\&\, \phi(\{i\}) = 1\}$.

We will also use that for finite semi-lattices coproducts and products coincide, with

$$\begin{array}{rcl} A+B & \to & A \times B \\ a & \mapsto & (a,0) \\ b & \mapsto & (0,b) \\ a \vee b & \leftarrow\!\shortmid & (a,b) \end{array} \tag{25}$$

describing the isomorphism. □

In Section 4.2 we have defined the finite trace semantics of *Set*-coalgebras $\gamma : X \to BTX$ as the final coalgebra semantics of the lifted coalgebra $\gamma : FX \to \widetilde{T}FX$ in *B-Alg*.

Secondly we need a functor L providing the modalities for our logics, as in the following diagram.

$$\widetilde{T} \circlearrowright B\text{-}Alg \underset{Q}{\overset{Q}{\rightleftarrows}} B\text{-}Alg^{op} \circlearrowleft L \qquad (\bot) \tag{26}$$

In analogy to Section 4.1, we develop finite trace logics as the initial L-algebra $\mathcal{L} : LI \to I$ in *B-Alg*. Note that under the assumptions of Remark 4.5, we have that I is the ω-colimit of the initial L-sequence:

$$0 \longrightarrow L0 \longrightarrow L^2 0 \longrightarrow \cdots \tag{27}$$

Definition 4.12. A trace logic is given by a functor $L : B\text{-}Alg \to B\text{-}Alg$ and a natural transformation $\delta : LQ \to Q\widetilde{T}$. Formulas of the logic are given by elements of the initial L-algebra. The semantics $[\![\cdot]\!]_{\widetilde{\gamma}}$ wrt a $\widetilde{T}$-coalgebra $\widetilde{\gamma} : FX \to \widetilde{T}FX$ is given by initiality as in

$$\begin{array}{ccc} LI & \xrightarrow{\qquad\qquad} & I \\ {\scriptstyle L[\![\cdot]\!]_{\widetilde{\gamma}}}\downarrow & & \downarrow{\scriptstyle [\![\cdot]\!]_{\widetilde{\gamma}}} \\ LQFX \xrightarrow{\delta_{FX}} Q\widetilde{T}FX & \xrightarrow{Q\widetilde{\gamma}} & QFX \end{array} \tag{28}$$

This induces the semantics $[\![\cdot]\!]_{\gamma}$ wrt a coalgebra $\gamma : X \to BTX$ via

$$UI \xrightarrow{U[\![\cdot]\!]_{\widetilde{\gamma}}} UQFX \xrightarrow{\cong} Q_0UFX \xrightarrow{Q_0\eta_X} Q_0X \tag{29}$$

For future reference, we record that the semantics in terms of γ and $\widetilde{\gamma}$ agree:

Proposition 4.13. *Let $\widetilde{\gamma} : FX \to \widetilde{T}FX$ be the $\widetilde{T}$-coalgebra induced by the (B,T)-coalgebra $\gamma : X \to BTX$, that is, $\gamma = U\widetilde{\gamma} \circ \eta_X$ with $\eta_X : X \to BX$ the unit of the monad B. Then $[\![\phi]\!]_{\gamma}(x) = [\![\phi]\!]_{\widetilde{\gamma}}(\eta_X(x))$.*

Example 4.14. Continuing from Example 4.11, in order to describe the logic (1), we let LA be the join-semilattice which is freely generated by $\surd$ and $\langle a \rangle \phi$ for $a \in Act$ and $\phi \in A$, quotienting by (6). To describe δ_{FX} it is convenient to note that QFX can be identified with the set of subsets of X as in (22) and $Q\widetilde{T}FX = QFTX$ with the set of subsets of TX. It therefore makes sense to define

$$\begin{aligned} \delta_{FX} : LQFX &\to Q\widetilde{T}FX \\ \surd &\mapsto \{S \subseteq TX \mid * \in S\} \\ \langle a \rangle \phi &\mapsto \{S \subseteq TX \mid \exists x (x \in \phi \;\&\; (a,x) \in S)\} \end{aligned}$$

Proposition 4.15. *(L, δ) of Example 4.14, together with (28), describes the same logic as (1) in Section 2.*

Proof. For example, we calculate $x \models \langle a \rangle \phi \;\Leftrightarrow\; \gamma(x) \in \{S \subseteq TX \mid \exists x' (x' \in \phi \;\&\; (a, x') \in S)\} \Leftrightarrow \gamma(x) \in \delta_{FX}(\langle a \rangle \phi) \Leftrightarrow x \in QF\gamma(\delta_{FX}(\langle a \rangle \phi)) \;\Leftrightarrow\; x \in [\![\langle a \rangle \phi]\!]$ where we use, respectively, (5), the definition of δ, the definition of Q, and (28). □

Theorem 4.16. *Consider a functor $T : Set \to Set$, a monad B, and a distributive law $TB \to BT$. Any (L, δ) with $L : B\text{-}Alg \to B\text{-}Alg$ and $\delta_K : LQK \to QK\overline{T}$ gives rise to a logic for BT-coalgebras invariant under B-trace semantics.*

Proof. For a given $\gamma : X \to BTX$ and formula ϕ, we have to show that $tr(x) = tr(y)$ implies $x \Vdash \phi \Leftrightarrow y \Vdash \phi$. Expressing this in $B\text{-}Alg$, this amounts to $\widetilde{tr}(\eta_X(x)) = \widetilde{tr}(\eta_X(y))$ only if $x \in [\![\phi]\!]_{\widetilde{\gamma}} \Leftrightarrow y \in [\![\phi]\!]_{\widetilde{\gamma}}$. But this is immediate from the initiality of the algebra of formulas as follows. Let (Z, ζ) be the final $\widetilde{T}$-coalgebra.

$$\begin{array}{ccccc} LI & \xrightarrow{\qquad\qquad\qquad} & & & I \\ \downarrow{\scriptstyle L[\![\cdot]\!]_\zeta} & & & & \downarrow{\scriptstyle [\![\cdot]\!]_\zeta} \\ LQZ & \xrightarrow{\delta_Z} & Q\widetilde{T}Z & \xrightarrow{Q\zeta} & QZ \\ \downarrow{\scriptstyle LQ\widetilde{tr}} & & \downarrow{\scriptstyle Q\widetilde{T}\widetilde{tr}} & & \downarrow{\scriptstyle Q\widetilde{tr}} \\ LQFX & \xrightarrow{\delta_{FX}} & Q\widetilde{T}FX & \xrightarrow{Q\widetilde{\gamma}} & QFX \end{array} \tag{30}$$

Since morphisms from the initial algebra $LI \to I$ are uniquely determined, we must have $[\![\cdot]\!]_{\widetilde{\gamma}} = Q\widetilde{tr} \circ [\![\cdot]\!]_\zeta$. ☐

4.5 Predicate Liftings

Whereas the previous section treats logics from an abstract point of view, we are now going to see how to describe them concretely using predicate liftings. First, we need to extend the set-based notion of predicate lifting [17, 19] to coalgebras over $B\text{-}Alg$.

Suppose we have L and

$$LQ \to Q\widetilde{T}.$$

Using $Id \to QQ$ from the adjunction (19) this gives us

$$L \to LQQ \to Q\widetilde{T}Q.$$

We will see below that $Q\widetilde{T}Q$ gives us predicate liftings, but first we are going to show how to recover $LQ \to Q\widetilde{T}$ from $L \to Q\widetilde{T}Q$. Write

$$J : Kl_\omega B \to B\text{-}Alg$$

for the inclusion of the category of finitely generated free algebras into $B\text{-}Alg$.

Proposition 4.17. *Let L be determined by finitely generated free algebras as in Remark 4.5. Then there is a bijection between natural transformations $LQ \to Q\widetilde{T}$ and natural transformations $LJ \to Q\widetilde{T}QJ$.*

Proof. Given $\delta : LQ \to Q\widetilde{T}$ we obtain $\rho : LJ \to Q\widetilde{T}QJ$ as $\delta Q \circ L\eta$. Conversely, given ρ, we write QA as a colimit $\phi_i : Fn_i \to QA$, which is preserved by L, and obtain δ via

$$\begin{array}{ccccc} QA & \quad & LQA & \xrightarrow{\delta_A} & Q\widetilde{T}A \\ \uparrow{\scriptstyle \phi_i} & & \uparrow{\scriptstyle L\phi_i} & & \uparrow{\scriptstyle Q\widetilde{T}\check{\phi}_i} \\ Fn_i & & LFn_i & \xrightarrow{\rho_{Fn_i}} & Q\widetilde{T}QFn_i \end{array} \tag{31}$$

where $\check{\phi}_i : A \to QFn_i$ is the adjoint transpose of ϕ_i. To check that these two assignments are inverse to each other, we first note that the diagram (31) can be rewritten as

$$\begin{array}{ccccc} & \nearrow{\scriptstyle L\phi_i} & LQA & \xrightarrow{\delta_A} & Q\widetilde{T}A \\ & & \uparrow{\scriptstyle LQ\check{\phi}_i} & & \uparrow{\scriptstyle Q\widetilde{T}\check{\phi}_i} \\ LFn_i & \xrightarrow{L\eta} & LQQFn_i & \xrightarrow[\delta QFn_i]{} & Q\widetilde{T}QFn_i \end{array} \tag{32}$$

where the triangle commutes because of the adjunction (19) and the quadrangle commutes because of naturality. It follows that starting from δ and defining ρ, the original δ satisfies (31) and therefore agrees with the δ defined from ρ. Conversely, defining δ from ρ in (31), one can choose $A = QFn$, $n_i = n$ and $\check{\phi} = id$, which shows that δ determines the ρ it comes from uniquely. □

We can interpret the proposition as follows. An element of

$$Q_0 A = UQA$$

is a predicate on A. An element of

$$[n, Q_0 A]$$

is an n-ary predicate on A. We have $[n, Q_0 A] \cong [Fn, QA] \cong [A, QFn]$ and find it useful to introduce the following notation. We want to write ϕ for n-ary predicates and if we want to make precise which of the three presentations we use, we write

$$\underline{\phi} \in [n, Q_0 A] \qquad \phi = \hat{\phi} \in [Fn, QA] \qquad \check{\phi} \in [A, QFn]. \tag{33}$$

Next we show how elements $l \in LFn$ are n-ary modal operators. Given an n-ary predicate ϕ on A, the 'modal operator' l induces an predicate on $\widetilde{T}A$ as follows.

$$\widetilde{T}A \xrightarrow{\widetilde{T}(\check{a})} \widetilde{T}QFn \xrightarrow{\rho_{Fn}(l)} \Omega \tag{34}$$

This shows that the meaning of the modal operator $l \in LFn$ is fully determined by the image $\rho_{Fn}(l) \in Q\widetilde{T}QFn$. We turn this observation into a definition.

Definition 4.18. Elements of $Q\widetilde{T}QFn$ are called n-ary predicate liftings. Each $\lambda \in Q\widetilde{T}QFn$ induces a natural transformation

$$\begin{array}{ll} [Fn, QA] & \to Q\widetilde{T}A \\ \phi & \mapsto \lambda \circ \widetilde{T}(\check{\phi}) \end{array} \tag{35}$$

Example 4.19. Consider $B = \mathcal{P}_\omega$, $T = \{*\} + Act \times Id$, $\widetilde{T}(A) = F\{*\} + Act \cdot A$. As in Example 4.9, we identify $F\emptyset$ with $\{\emptyset\}$ and $\widetilde{T}^n(F\emptyset)$ with $\mathcal{P}_\omega(1 + Act + \ldots Act^n)$. The initial and final $\widetilde{T}$-algebras are then $\mathcal{P}_\omega(Act^*)$ and $\mathcal{P}(Act^*)$, respectively. Recall that $QA = [A, F1] = [A, 2]$ and we write $0, 1 \in 2$. Further note that, for finite n, there is a bijection $UQFn = U[Fn, 2] \cong Set(n, 2) \cong Bn = UFn$ which extends to a semi-lattice isomorphism $QFn \cong Fn$.

In order to obtain the clause for $\surd$, we instantiate (35) with $n = \emptyset$ (because $\surd$ is a constant) and let $\lambda_\surd$ be the unique isormorphism

$$\widetilde{T}QF\emptyset \cong \widetilde{T}F\emptyset = F\{*\} + Act \cdot F\emptyset \cong F\{*\} \longrightarrow 2. \tag{36}$$

Consider A and $\phi : F\emptyset \to QA$ and $\check{\phi} : A \to QF\emptyset \cong F\emptyset$. This gives us the semantics of $\surd$ as follows. $\delta_A(\surd) \in Q\widetilde{T}A$ as in (31) is the map

$$\begin{array}{ccc} F\{*\} + Act \cdot A & \xrightarrow{\delta(\surd)} & 2 \\ {\scriptstyle F\{*\}+Act\cdot\check{\phi}}\downarrow & \nearrow{\scriptstyle \lambda_\surd} & \\ F\{*\} + Act \cdot F\emptyset & & \end{array} \tag{37}$$

Finally, putting this together with (28) and (29) we find that, as expected,

$$x \Vdash \surd \;\Leftrightarrow * \in \gamma(x).$$

In order to obtain the clause for $\langle a \rangle \phi$, we instantiate (35) with $n = 1$ and let λ_a be given by the map

$$\widetilde{T}QF1 \cong \widetilde{T}F1 = F\{*\} + Act \cdot F1 \longrightarrow 2 \tag{38}$$

which sends all generators $*$ and $b \in A, b \neq a$ to 0 and a to 1. Consider A and choose some $\phi : F1 \to QA$. Note that $\check{\phi} : A \to QF1 \cong F1 \cong \mathbb{2}$. This gives us the semantics of $\langle a \rangle \phi$ as follows. $\delta(\langle a \rangle \phi) \in Q\widetilde{T}A$ as in (31) is the map

$$\begin{array}{ccc} F\{*\} + Act \cdot A & \xrightarrow{\delta(\langle a \rangle \phi)} & \mathbb{2} \\ {\scriptstyle F\{*\}+Act\cdot\check{\phi}} \downarrow & \nearrow_{\lambda_a} & \\ F\{*\} + Act \cdot F1 & & \end{array} \tag{39}$$

Finally, putting this together with (28) and (29) we find that, as expected,

$$x \Vdash \langle a \rangle \phi \ \Leftrightarrow \ (a, x') \in \gamma(x) \text{ and } x' \Vdash \phi.$$

Every collection of predicate liftings defines a functor.

Definition 4.20. Given a collection of predicate liftings Λ let $L_\Lambda A = F \coprod_{\lambda \in \Lambda} [F(n_\lambda), A]$, where n_λ is the arity of λ. The semantics δ_Λ acts on a generator $(\lambda, \phi) \in Q\widetilde{T}QFn \times [Fn, QA]$ as given by (35).

Example 4.21. Let $\Lambda = \{\lambda_{\surd}\} \cup \{\lambda_a \mid a \in Act\}$ as in Example 4.19. Then $L_\Lambda A \cong F1 + Act \cdot FUA$ and δ_Λ is given by (37) and (39).

It is possible to incorporate logical laws into the functor.

Example 4.22. Let $\Lambda = \{\lambda_{\surd}\} \cup \{\lambda_a \mid a \in Act\}$ as in Example 4.19 and consider the set E of equations given by (6). Then $L_{\Lambda E} \cong F1 + Act \cdot Id$ and $\delta_{\Lambda E}$ is given by (37) and (39). Furthermore, we have

$$\begin{array}{ccc} F1 + Act \cdot Q & \xrightarrow{\kappa} & \\ \cong \downarrow & & Q\widetilde{T} \\ L_{\Lambda E} Q & \xrightarrow{\delta_{\Lambda E}} & \end{array} \tag{40}$$

where, on finite A, κ_A is the isomorphism

$$F1 + Act \cdot QA \longrightarrow F1 \times \textstyle\prod_{Act} QA \longrightarrow Q(F1 + Act \cdot A) \longrightarrow Q\widetilde{T}A \tag{41}$$

where the first iso comes from (25), the second is due to Q being a hom-functor, and the third is from the definition of $\widetilde{T}$.

To summarise, we have extracted from the example in Section 2 a general framework that allows to define trace logics for general functors T and monads B satisfying Assumption 2.

4.6 A generic trace logic

In this section, we show how to define a logic (L_T, δ_T) for general functors T and monads B satisfying Assumption 2. We show that the example from the previous section arises in that way.

Definition 4.23. The functor $L_T : B\text{-}Alg \to B\text{-}Alg$ is defined on finitely generated free algebras Fn as $L_T Fn = Q\widetilde{T}QFn$. Since every $A \in B\text{-}Alg$ is a colimit of finitely generated free algebras, this extends continuously to all $A \in B\text{-}Alg$.

Definition 4.24. The semantics $\delta_T : L_T Q \to Q\widetilde{T}$ is given by considering QA as a colimit $\phi_i : Fn_i \to QA$, which is, by construction, preserved by L_T. More explicitly, $(\delta_T)_X$ is the unique arrow making the following diagram

$$\begin{array}{cccc} QA & L_T QA & \xrightarrow{(\delta_T)_A} & Q\widetilde{T}A \\ {\scriptstyle \phi_i} \uparrow & {\scriptstyle L_T \phi_i} \uparrow & & \uparrow {\scriptstyle Q\widetilde{T}\check{\phi}_i} \\ Fn_i & L_T Fn_i & \xrightarrow{=} & Q\widetilde{T}QFn_i \end{array} \tag{42}$$

commute for each i; as in (33), the arrow $\check{\phi}_i$ comes from applying the isomorphism $B\text{-}Alg(Fn_i, QA) \cong B\text{-}Alg(A, QFn_i)$ to ϕ_i.

To show that the example of the previous section is actually the generic one, we need a lemma helping us to compare the two logics.

Lemma 4.25. *Let $(L, \delta), (L', \delta')$ be two logics and ρ, ρ' as in (31). If there is an isomorphism $\alpha : LJ \to L'J$ such that for all finite sets n we have*

$$\begin{array}{ccc} LFn & \xrightarrow{\rho} & \\ {\scriptstyle \alpha_n}\downarrow & & Q\widetilde{T}QFn \\ L'Fn & \xrightarrow{\rho'} & \end{array} \tag{43}$$

then this extends to an isomorphism $\beta : L \to L'$ of logics, ie, β satisfies

$$\begin{array}{ccc} LQ & \xrightarrow{\delta} & \\ {\scriptstyle \beta Q}\downarrow & & Q\widetilde{T} \\ L'Q & \xrightarrow{\delta'} & \end{array} \tag{44}$$

Moreover, $\beta_{Fn} = \alpha_n$.

Consequently, any collection of isomorphisms $Ln \to Q\widetilde{T}QFn$, $n \in \mathbb{N}$, defines the same logic, or, more precisely:

Corollary 4.26. *The generic logic L_T is determined up to isomorphism, that is, for any other logic (L, δ) with the $LFn \to Q\widetilde{T}QFn$ as in (31) being isos, there is a unique isomorphism $L \to L_T$ such that*

$$\begin{array}{ccc} LQ & \xrightarrow{\delta} & \\ \downarrow & & Q\widetilde{T} \\ L_TQ & \xrightarrow{\delta_T} & \end{array} \tag{45}$$

Finally, we can show that the generic logic of this subsection agrees with the logic defined, in different ways, by (1)-(6), or again in Example 4.14 or in Example 4.22.

Proposition 4.27. *Going back to Example 4.22, there is an isomorphism such that*

$$\begin{array}{ccc} L_{\wedge E}Q & \xrightarrow{\delta_{\wedge E}} & \\ {\scriptstyle \cong}\downarrow & & Q\widetilde{T} \\ L_TQ & \xrightarrow{\delta_T} & \end{array} \tag{46}$$

Proof. We write (L, δ) for $(L_{\wedge E}, \delta_{\wedge E})$ and ρ for the natural transformation as in (31). According to Corollary 4.26, it is enough to show that $\rho_{Fn} : LFn \to Q\widetilde{T}QFn$ is an isomorphism. From the proof of Proposition 4.17, we know that $\rho_{Fn} = \delta_{QFn} \circ L\eta$. Since η is an isomorphism for finite semi-lattices, the result now follows from δ_{QFn} being iso, see Example 4.22. □

Finally, Definition 4.24 does not depend on the choice of a partiuclar T or B, so we can summarise this section as follows.

Theorem 4.28. *For every monad B on Set and functor $T : Set \to Set$ satisfying Assumption 2 there is a generic trace logic.*

Of course, given B and T, the real work consists in finding a good explicit description of the generic logic. We have illustrated this for the moment only with one example.

We can apply the general framework to obtain results about generic logics. For example, we have

Theorem 4.29. *The logic of Example 4.22 is expressive and complete.*

Proof. We write (L, δ) for $(L_{\wedge E}, \delta_{\wedge E})$. The proof is straightforward due to the following facts: B and $\widetilde{T}$ preserve finite algebras and on finite algebras we have that δ is an isomorphism. In detail:

Expressiveness means that any two non-trace equivalent states can be separated by a formula. Consider a coalgebra $X \to BTX$ with $x, x' \in X$ and suppose x accepts trace t and x' does not. Since the initial L-algebra is the free B-algebra over the set of traces, t can be considered as a formula and we have $x \Vdash t$ and $x' \nVdash t$.

Completeness means that if L does not prove $\phi = \phi'$, then there must be a coalgebra $X \to BTX$ and $x \in X$ such that, wlog, $x \Vdash \phi$ and $x \nVdash \phi'$. Since ϕ and ϕ' appear at some stage n in the initial algebra construction of L, the semantics of ϕ and ϕ' is determined at stage n. Since δ is an iso on finite algebras, the images of ϕ and ϕ' in $Q\widetilde{T}^n F\emptyset$ are different. It follows from a standard argument that there is a $\widetilde{T}$-coalgebra $\widetilde{\gamma} : \widetilde{T}^n F\emptyset \to \widetilde{T}(\widetilde{T}^n F\emptyset)$ that refutes the equation $\phi = \phi'$. In particular, $[\![\phi]\!]_{\widetilde{\gamma}} \neq [\![\phi']\!]_{\widetilde{\gamma}}$ are two different morphisms $FT^n\emptyset = Q\widetilde{T}^n F\emptyset \to 2$, so they must differ on some generator $\eta_X(x)$ where $\eta_X : X \to BX$ maps elements x to singletons $\{x\}$. It follows now from Proposition 4.13 that the (B, T)-coalgebra $U\gamma \circ \eta_X : X \to BTX$ contains a state x with $x \Vdash \phi$ and $x \nVdash \phi$. □

References

[1] P. Aczel. *Non-Well-Founded Sets.* CSLI, Stanford, 1988.

[2] J. Adámek, J. Rosický, and E. Vitale. *Algebraic Theories.* Cambridge University Press, 2011.

[3] E. de Vink and J. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. In *ICALP'97.*

[4] I. Hasuo. *Tracing Anonymity with Coalgebras.* PhD thesis, University of Nijmegen, 2008.

[5] I. Hasuo, B. Jacobs, and A. Sokolova. Generic Trace Theory. In *CMCS'06.*

[6] B. Jacobs. Trace Semantics for Coalgebras. *Electronic Notes in Theoretical Computer Science*, 106, 2004.

[7] P. Johnstone. *Stone Spaces.* Cambridge University Press, 1982.

[8] B. Klin. Bialgebraic operational semantics and modal logic. In *LICS'07.*

[9] A. Kock. Monads on symmetric monoidal closed categories. *Archiv der Mathematik*, 21(1):1–10, Dec. 1970.

[10] C. Kupke, A. Kurz, and Y. Venema. Completeness of the finitary Moss logic. In *AiML'08.*

[11] A. Kurz. Coalgebras and their logics. *SIGACT News*, 37, 2006.

[12] A. Kurz and R. Leal. Equational Coalgebraic Logic. In *MFPS'09.*

[13] A. Kurz and J. Rosický. The Goldblatt-Thomason-theorem for coalgebras. In *CALCO'07.*

[14] A. Kurz and J. Rosický. Strongly complete logics for coalgebras. July 2006.

[15] S. MacLane. *Categories for the Working Mathematician.* Graduate Texts in Mathematics. Springer, New York, 2nd edition edition, 1998.

[16] L. Moss. Coalgebraic logic. *Ann. Pure Appl. Logic*, 96, 1999.

[17] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theor. Comp. Sci.*, 309, 2003.

[18] J. J. Rutten. Coinductive Counting With Weighted Automata, 2002. *Journal of Automata, Languages and Combinatorics*, 8, 2003.

[19] L. Schröder. Expressivity of Coalgebraic Modal Logic: The Limits and Beyond. In *FOSSACS'05.*

[20] Z. Semadeni. *Monads and their Eilenberg-Moore algebras in functional analysis.* Queen's Papers in Pure and Applied Mathematics, No. 33. Queen's University, Kingston, Ont., 1973.

[21] J. Velebil and A. Kurz. Equational presentations of functors and monads. *Math. Struct. Comput. Sci.*, 2011.