Generic Trace Logics

Christian Kissig[1] Alexander Kurz[2]

University of Leicester
Department of Computer Science
University Road
Leicester LE17RH
United Kingdom

Abstract

Finite trace semantics is known and well understood for classical automata and non-deterministic labelled transition systems. Jacobs et al introduced a more general definition for coalgebras which are structured in a branching type in addition to the transition type, and generalise non-deterministic and probabilistic state based transition structures.
In this work we propose a class of coalgebraic logics which adequately and expressively characterise finite trace semantics and have a compositional semantics. We obtain generic trace logics from a dual adjunction on the Eilenberg-Moore category of the monad embodying the branching type.

Keywords: Coalgebra, Coalgebraic Logic, Finite Trace Semantics, Generic Trace Theory

---

1 Introduction

The coalgebraic approach to modal logic has been pursued successfully over the last years. The basic ideas (see eg [16,17,19,10,6]), are the following.

- A T-*coalgebra*, consisting of a carrier X and a 'next-step' map $\gamma : X \to TX$, 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 [2].
- 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.

---

[1] Email:`christian@mcs.le.ac.uk`

[2] Email:`kurz@mcs.le.ac.uk`

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

In parallel, Jacobs and collaborators [5,4,3] 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 $B = \mathcal{P}$ 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 with fewer assumptions, but a limitation of trace semantics to a finite, but arbitrary depth. Nevertheless a finite horizon allows to distinguish states not finite trace equivalent.

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. Generic trace logics have the following properties.

- Formulas are invariant under trace equivalence.
- The logics are expressive: two states with different semantics are distinguished by some formula.
- The logics are complete: If two formulas are satisfied by the same states, then the formulas can be proven equivalent.

Structure of the paper We begin our argument with examples illustrating our intuitions about generic trace logics and the technical problems that arise with generality. In Section 4 we describe finite trace semantics, which leads us to our definition of generic trace logics. We show that generic trace logics are indeed invariant under finite trace equivalence and with additional assumptions expressive in Section 5. Finally, we show suitability of trace logics with four examples.

Acknowledgements

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 $\gamma : X \to \mathcal{D}(\{*\} + \mathit{Act} \times X)$ 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)

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

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

The notation in (8) indicates that there must be a definition of logic, semantics, axiomatisation paralleling the example of non-determinstic automata.

3 Preliminaries

3.1 Monads, Algebras and Coalgebras

Definition 3.1 [Coalgebras] 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.

Definition 3.2 [Monads] A *monad* on *Set* is an endofunctor $B : \mathit{Set} \to \mathit{Set}$ with a unit law, that is a natural transformation $\eta : \mathit{Id} \Rightarrow B$, and a multiplication law, that is a natural transformation $\mu : BB \Rightarrow B$. η and μ commute such that $\mu \circ \eta_T = \mathit{id}_T = \mu \circ T\eta$ and $\mu \circ \mu_T = \mu \circ T\mu$.

Example 3.3 (i) The powerset monad $\mathcal{P}$ with set-union and the singleton constructor $\{(-)\}$ is a monad, and similarily for the finitary powerset functor $\mathcal{P}_\omega$.

(ii) 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 $\lambda y. \sum_{x \in f^{-1}(y)} m(x)$. The bag functor with multiset-union and the multiset-singleton constructor constitutes a monad.

(iii) 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

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

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

(iv) $\mathcal{P}$ and $\mathcal{B}$ 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 $\mathcal{P}$ the semiring is the boolean algebra $\langle \{\top, \bot\}, \wedge, \vee, \top, \bot \rangle$, and for $\mathcal{B}$ the semiring are the natural numbers $\langle \mathbb{N}, +, *, 0, 1 \rangle$.

(v) Another example of a semiring monad uses the min-semiring $\langle \mathbb{N} \cup \{\infty\}, \mathit{min}, +, \infty, 0 \rangle$ 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. Semiring monad for the min-semiring are branching type for cost-optimal paths, and will be covered in greater detail in Section 6.4.

(vi) 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 algebra for a functor B in *Set* is a function $\alpha : BX \to X$; an algebra for a monad B additionally makes $\alpha \circ \mu_X = \alpha \circ B\alpha$ and $\alpha \circ \eta_X = id_X$, commute. A B-algebra morphism between B-algebras $\alpha : BX \to X$ and $\beta : BY \to Y$ is a function $f : X \to Y$ which makes $f \circ \alpha = \beta \circ Bf$ commute.

The algebras for a monad B form a category, the Eilenberg-Moore category B-*Alg*. From B-*Alg* there is a forgetful functor, U, which takes each algebra to its carrier. U has a left adjoint, F, which takes a set X to the free B-algebra generated from X.

Each monad admits an initial and a final B-algebra, $\langle B\emptyset, \mu_\emptyset : B^2\emptyset \to B\emptyset \rangle$ and $\langle \{*\}, (\lambda.*) : B\{*\} \to \{*\} \rangle$.

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 [7] 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 [7].

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 consists of

- objects $X, Y, \ldots$ which are sets
- morphisms $f : X \to Y$ which are functions $f : X \to BY$ in Set
- composites of morphisms $f : X \to Y$ and $g : Y \to Z$ in KlB which are $g \circ f := \mu_Z \circ Bg \circ f$ in Set

The adjunction $F' \dashv U' : Set \to KlB$ 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 (i) The Kleisli-category for the powerset monad P is Rel, the category of sets as objects and relations as morphisms.

(ii) 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, we define the lifting of Set-functors T into KlB. The lifted functor $\overline{T}$ makes $F'T = \overline{T}F'$ 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. T is the functor which labels in Act or marks successful termination $(*)$. With each of the monads in Example 3.3 T has a distributive law.

(i) $\pi : T\mathcal{P} \Rightarrow \mathcal{P}T$: $\pi_X(*) := \{*\}$, $\pi_X(a, Y \subseteq X) := \{(a, x) \mid x \in Y\}$.

(ii) $\pi : T\mathcal{B} \Rightarrow \mathcal{B}T$: $\pi_X(*) := \eta_{\{*\}+Act \times X}(*)$, and $\pi_X(a, m)(a, x) := \{(a, x) \mapsto m(x), (b, x) \mapsto 0, * \mapsto 0 \mid a \in Act, b \in Act, b \neq a, x \in X\}$

(iii) $\pi : T\mathcal{D} \Rightarrow \mathcal{D}T$: $\pi_X(*) := \eta_{\{*\}+Act \times X}(*)$, 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 $D \in \{\mathcal{D}_{\leq 1}, \mathcal{D}_{=1}\}$

Definition 3.10 [Functor Lifting by Distributive Law] Given a distributive law $\pi : TB \Rightarrow BT$ we can define $\overline{T}$ from F 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 $K : \mathit{KlB} \to B\text{-}\mathit{Alg}$ mapping X to the free algebra over X, see [14]. In other words, we can think of KlB as the full subcategory of B-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 basics of coalgebraic logic (more can be found in [11]) and the fundamentals of generic trace semantics [5]).

4.1 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 $\mathit{Set} \to \mathit{Set}$ 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

$$\begin{tikzcd}
\mathit{Set} \arrow[loop left, "T"] \arrow[r, bend left, "Q"] \arrow[r, phantom, "\perp"] & BA^{op} \arrow[l, bend left, "S"] \arrow[loop right, "L"]
\end{tikzcd} \tag{9}$$

Q takes sets X contravariantly 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 $\gamma : X \to TX$ is given by the unique homomorphism from the initial L-algebra $\mathcal{L} : LI \to I$ as in:

$$\begin{tikzcd}
LI \arrow[rr, "\mathcal{L}"] \arrow[d, "L(\llbracket\cdot\rrbracket)"'] & & I \arrow[d, "\llbracket\cdot\rrbracket"] \\
LQX \arrow[r, "\delta_X"] & QTX \arrow[r, "Q\gamma"] & QX
\end{tikzcd} \tag{12}$$

Theorem 4.1 *Any* (L, δ) *with* δ *as in (11) gives rise to a logic for* T*-coalgebras. The semantics* $[\![\cdot]\!]$ *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 (i) 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 [9].

(ii) 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 generic trace semantics

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

$$\begin{tikzcd} \emptyset \arrow[r, hook, "\emptyset"] & T\emptyset \arrow[r, hook, "T\emptyset"] & T^2\emptyset \arrow[r, hook] & \cdots & T^\omega\emptyset \end{tikzcd} \tag{13}$$

In the example $T^n\emptyset = \{a_1 \ldots a_n \mid a_i \in \mathit{Act}\}$ and $T^\omega\emptyset = \mathit{Act}^*$, 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.

- 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)
- a map $e : X \to B\emptyset$ (we assume that $B\emptyset \neq \emptyset$)

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

$$\begin{tikzcd} X \arrow[r, "\gamma"] & BTX \arrow[r, "BT\gamma"] & BTBTX \arrow[r, "BTBTe"] & BTBTB\emptyset \arrow[r, "p"] & BBBTT\emptyset \arrow[r, "m"] & BT^2\emptyset \end{tikzcd}$$

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

Trace semantics in the Kleisli category [4] 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 very 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 $\mathit{Kl}(B)$ are maps $X \to BY$ in *Set*. In the example, $\mathit{Kl}(B)$ is the category of sets with relations as arrows.

The trace maps $(tr_n)_{n<\omega}$ arise inductively from an initial morphism $tr_0 : X \to \emptyset$ (see the assumption above) as follows.

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

The following diagram in KlB illustrates the definition.

$$\begin{tikzcd}
 & X \arrow[r, "\gamma"] \arrow[dl, bend right, "tr_0"'] \arrow[dr, bend left, "tr_n"'] \arrow[drr, "tr_{n+1}"] & \overline{T}X \arrow[dr, bend left, "\overline{T}tr_n"] & & \\
\emptyset & \cdots & \overline{T}^n\emptyset & \overline{T}^{n+1}\emptyset & \cdots
\end{tikzcd} \tag{16}$$

Finally, we should note that the above account of trace semantics also works if we take B to be *finite* powersets or multisets. Under additional assumptions we do not need for this paper (and excluding $\mathcal{P}_\omega, \mathcal{B}$), [4] prove the elegant theorem that the initial T-algebra with the carrier $T^\omega\emptyset$ as in (13) is the final $\overline{T}$-coalgebra, thus subsuming trace-equivalence under the general principle of coinduction.

4.3 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. The latter approach has been taken by Klin in [6].

$$\begin{tikzcd}
Set \arrow[r, bend left, "2^{(-)}"] \arrow[r, phantom, "\perp"] & Set^{op} \arrow[l, bend left, "2^{(-)}"]
\end{tikzcd} \tag{17}$$

In the above situation, 2 takes the role of a schizophrenic object. Analogously we may choose a B-algebra to replace 2. In most examples we have considered, $F1$ is a suitable choice. If B is commutative, $[-, F1]$ defines a contravariant endofunctor on B-Alg. For a tidy notation we abbreviate Q for $[-, F1]$.

$$\begin{tikzcd}
B\text{-}Alg \arrow[r, bend left, "Q"] \arrow[r, phantom, "\perp"] & B\text{-}Alg^{op} \arrow[l, bend left, "Q"]
\end{tikzcd} \tag{18}$$

Our logics are expressive only if $F1$ is non-trivial, that is $F1$ contains more than one element. This requirement disqualifies the exact distribution monad $B = \mathcal{D}_{=1}$, but not the sub-distribution monad $B = \mathcal{D}_{\leq 1}$.

Example 4.3 (i) When $B = \mathcal{P}_\omega$, B-Alg = $SLat$ is the category of (join) semi-lattices. For $F1$ we choose the two-element semi-lattice, so that $[-, F1]$ takes a semi-lattice A to the set of filters over A.

(ii) For $B = \mathcal{D}$, B-Alg is the category of distribution algebras. For $\mathcal{D}_{\leq 1}$, $F1$ is the set of probabilities, and for $\mathcal{D}_{=1}$, $F1$ is the trivial algebra with the only element $\{* \mapsto 1\}$.

(iii) For B being a monad for the semiring S, $F1$ is the semiring S conceived as a left S-module. For graded branching, $F1$ is the set of grades.

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

$$\begin{tikzcd} Kl(B) \arrow[loop left, "\overline{T}"] \arrow[r, "K"] & B\text{-}Alg \arrow[r, bend left, "Q"] \arrow[r, phantom, "\perp"] & B\text{-}Alg^{op} \arrow[l, bend left, "Q"] \arrow[loop right, "L"] \end{tikzcd} \tag{19}$$

L shall come with a natural transformation $LQK \Rightarrow QK\overline{T}$, the denotation of L. Analogously to the construction of coalgebraic logics for T-bisimulation in Section 4.1, we develop generic trace logics as the initial L-algebra $\mathcal{L} : LI \to I$. We suppose that I is the ω-colimit of the initial L-sequence. The semantics for generic trace logics shall be given by the initial algebra map as in the following diagram.

$$\begin{tikzcd} LI \arrow[rr, "\mathcal{L}"] \arrow[d, "L(\llbracket\cdot\rrbracket)"'] & & I \arrow[d, "\llbracket\cdot\rrbracket"] \\ LQKX \arrow[r, "\delta_X"] & QK\overline{T}X \arrow[r, "QK\gamma"] & QKX \end{tikzcd} \tag{20}$$

Example 4.4 Continuing from Example 4.3, 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 $\phi \in A$, modding out by (6). To describe δ it is convenient to note that QKX can be identified with the set of subsets of X and $QK\overline{T}X$ with the set of subsets of TX. It therefore makes sense to define

$$\begin{aligned} \delta_X : LQKX &\to QK\overline{T}X \\ \surd &\mapsto \{S \subseteq TX \mid * \in S\} \\ \langle a\rangle\phi &\mapsto \{S \subseteq TX \mid \exists x'(x' \in \phi \;\&\; (a, x') \in S)\} \end{aligned}$$

Proposition 4.5 *(L, δ) of Example 4.4, together with (20), describes the same logic as (1) in Section 2.*

Proof. For example, we calculate $x \Vdash \langle a\rangle\phi \;\Leftrightarrow\; \gamma(x) \in \{S \subseteq TX \mid \exists x'(x' \in \phi \;\&\; (a, x') \in S)\} \Leftrightarrow \gamma(x) \in \delta_X(\langle a\rangle\phi) \Leftrightarrow x \in QK\gamma(\delta_X(\langle a\rangle\phi)) \;\Leftrightarrow\; x \in \llbracket\langle a\rangle\phi\rrbracket$ where we use, respectively, (5), the definition of δ, the definition of Q (and K), and (20).□

Theorem 4.6 *Consider a functor $T : Set \to Set$, a monad B, and a distributive law $TB \Rightarrow BT$. Any (L, δ) with $L : B\text{-}Alg \to B\text{-}Alg$ and $\delta : LQK \Rightarrow QK\overline{T}$ gives rise to a logic for (B, T)-coalgebras invariant under B-trace semantics.*

Remark 4.7 As in [8] the logic will be complete if δ is in injective and expressive if δ is surjective and L respectively preserves these properties.

5 Proof of the Theorem

We prove that generic trace logics are invariant under finite trace equivalence, by showing that $\llbracket-\rrbracket$ factors through tr_n for each $n < \omega$. We use that generic trace logics arise as the ω-colimit of the initial L sequence to show a stratification.

Every L-algebra forms a cocone over the initial L-sequence. Let $(i_n : L^n K\emptyset \to I)_{n<\omega}$ be the (colimiting) cocone for I, and for the L-algebra $\delta_X \circ QK\gamma$ the cocone be $(\llbracket-\rrbracket_n : L^n \to QKX)_{n<\omega}$ such that $\llbracket-\rrbracket_0$ is the initial object map $\llbracket-\rrbracket_0 : K\emptyset \to QKX$

and

$$\llbracket - \rrbracket_{n+1} = QK\gamma \circ \delta_X \circ L\llbracket - \rrbracket_n \tag{21}$$

as in

$$\begin{tikzcd}
 & QKX & QK\overline{T}X \arrow[l, "QK\gamma"'] & LQKX \arrow[l, "\delta_X"'] & \\
K\emptyset \arrow[ur, bend left, "\llbracket - \rrbracket_0"] \arrow[r] & \cdots \arrow[r] & L^nK\emptyset \arrow[ul, "\llbracket - \rrbracket_n"] \arrow[r] & L^{n+1}K\emptyset \arrow[ull, "\llbracket - \rrbracket_{n+1}"] \arrow[u, bend right, "L\llbracket - \rrbracket_n"'] \arrow[r] & \cdots
\end{tikzcd} \tag{22}$$

All triangles above commute, viz $\llbracket - \rrbracket \circ i_n = \llbracket - \rrbracket_n$ for all n.

We obtain the invariance result once we have matched the initial L-sequence and the inital $\overline{T}$-sequence.

$$\begin{tikzcd}
K\emptyset \arrow[r, "j"] \arrow[dd, "k_0"] & LK\emptyset \arrow[r, "Lj"] \arrow[d, "Lk_0"] & L^2K\emptyset \arrow[r, "L^2j"] \arrow[d, "L(\delta_0 \circ Lk_0)"] & \cdots \\
 & LQK\emptyset \arrow[d, "\delta_\emptyset"] & LQK\overline{T}\emptyset \arrow[l, "LQKe"] \arrow[d, "\delta_{\overline{T}\emptyset}"] & \cdots \arrow[l, "LQK\overline{T}e"] \\
QK\emptyset & QK\overline{T}\emptyset \arrow[l, "QKe"] & QK\overline{T}^2\emptyset \arrow[l, "QK\overline{T}e"] & \cdots \arrow[l, "QK\overline{T}^2e"]
\end{tikzcd} \tag{23}$$

where k_0 is the uniquely determined initial object map. By an easy inductive argument one can show that the above diagram commutes. The left square commutes because $K\emptyset$ is initial in $B\text{-}Alg$; the right lower square commutes by naturality of δ; and the right upper square is the image of the left one under L and thus commutes as well. The commutativity of the diagram means that generic trace logics and its semantics is stable under application of L.

$$k_{n+1} := \delta_{\overline{T}^n\emptyset} \circ Lk_n \tag{24}$$

To prove generic trace logics invariant under finite trace equivalence using the idea of stratification, we show for each $n < \omega$ that Qtr_n factors $\llbracket - \rrbracket_n$ as follows.

Lemma 5.1 *For all $n < \omega$, $\llbracket - \rrbracket_n = QKtr_n \circ k_n$.*

$$\begin{tikzcd}
 & & QKX & QK\overline{T}X \arrow[l, "QK\gamma"'] & LQKX \arrow[l, "\delta_X"'] \\
QK\emptyset & \cdots \arrow[l, "QKe"] & QK\overline{T}^n\emptyset \arrow[l] \arrow[u, "QKtr_n"] & QK\overline{T}^{n+1}\emptyset \arrow[l, "QK\overline{T}^ne"] \arrow[u, "QK\overline{T}tr_n"'] & \cdots \arrow[l] \\
K\emptyset \arrow[u, "k_0"] \arrow[r, "j"'] & \cdots \arrow[r] & L^nK\emptyset \arrow[u, "k_n"] \arrow[uu, bend left, "\llbracket - \rrbracket_n"] \arrow[r, "L^nj"'] & L^{n+1}K\emptyset \arrow[u, "k_{n+1}"] \arrow[uur, bend right, "L\llbracket - \rrbracket_n"'] \arrow[r] & \cdots
\end{tikzcd} \tag{25}$$

Proof. We show the lemma by induction. For $n = 0$, $QKtr_0 \circ k_0 = \llbracket - \rrbracket_0$ commutes because $K\emptyset$ is initial. As the inductive hypothesis we suppose $QK\mathit{tr}_n \circ k_n = [\![-]\!]_n$.

$$\begin{aligned}
&QK\mathit{tr}_n \circ k_n = [\![-]\!]_n && \text{apply } L\\
&LQK\mathit{tr}_n \circ Lk_n = L[\![-]\!]_n && \text{compose with } QK\gamma \circ \delta_X\\
&QK\gamma \circ \delta_X \circ LQK\mathit{tr}_n \circ Lk_n = QK\gamma \circ \delta_X \circ L[\![-]\!]_n && \text{by naturality of } \delta\\
&QK\gamma \circ QK\overline{T}\mathit{tr}_n \circ \delta_{\overline{T}\emptyset} \circ Lk_n = QK\gamma \circ \delta_X \circ L[\![-]\!]_n && (15),(20),(21)\\
&QK\mathit{tr}_{n+1} \circ k_{n+1} = [\![-]\!]_{n+1}
\end{aligned}$$

□

This concludes the proof of Theorem 4.6. Under additional assumptions, we can make a statement about the expressiveness of generic trace logics.

The map $k_0 : K\emptyset \to QK\emptyset$ is epi, because we assumed that the domain $B\emptyset$ of $F\emptyset$ is not empty, and $QK\emptyset$ has only one element. In fact all k_n are epi under the assumptions that δ is epi, and L preserves epis.

Lemma 5.2 *If δ is epi and L preserves epis, then k_n is epi for each $n < \omega$.*

It remains to show that generic trace logics distinguish states not finite trace equivalent, which means that generic trace logics are expressive. Expressiveness turns out to depend on $(k_n)_{n<\omega}$ being epi which follows from δ being naturally epi and L preserving epis. See Lemma 5.2. Under the same assumptions, however, we obtain a stronger property, namely that generic trace logics can distinguish states not finite trace equivalent at a definitive finite depth. This establishes the finitariness of generic trace logics.

Theorem 5.3 *For the branching type B, transition type T, logic functor L with denotation $\delta : LQK \Rightarrow QK\overline{T}$ is expressive if δ is naturally epi and L preserves epis.*

Proof. Suppose points $x, y \in X$ are separable by trace semantics at some depth n, viz $\mathit{tr}_n(x) \neq \mathit{tr}_n(y)$, then there is a morphism $f : K\overline{T}^n\emptyset \to F1$ separating $\mathit{tr}_n(x)$ and $\mathit{tr}_n(y)$. If k_n is epi, there is a depth n formula $\phi \in L^n 0$ such that $k_n(\phi) = f$. Due to the commutativity of $[\![-]\!]_n = QK\mathit{tr}_n \circ k_n$, ϕ logically distinguishes x and y. □

6 Applications

In Section 2 we began with adhoc attempts of constructing trace logics, and then gave a rather abstract construction of trace logics in Section 4. Next we show that our categorical machinery yields concrete, usable logics that meet the anticipated criteria, for instance of Equation (7).

Throughout this section we will consider labelled transition systems (LTSs), only, that are coalgebras of transition type $T(-) = \{*\} + \mathit{Act} \times (-)$. Other examples, such as labelled binary trees $(\mathit{Act} \times (-) \times (-))$, context-free grammars $(\mathit{List}(\mathit{Act} \times (-)))$, etc work similarily. We suppose that Act is finite.

6.1 Non-deterministic Labelled Transition Systems

The branching type of non-deterministic LTSs is $B = \mathcal{P}$. Algebras for $\mathcal{P}$ are complete join semi-lattices. For the schizophrenic object $F1$ we choose $F1$, the free two-element semi-lattice $\{\bot, \surd\}$ of truth values. The logic functor $L(-) = F\{\surd\} \times (-)^{Act}$ takes a semi-lattice A to a pair $(t, f : Act \to A)$ where $t = \surd$ ($t = \bot$) means that the current state is successfully terminating (failing) and f characterises for each $a \in Act$ the sucessor states.

Formulas, that is elements of the carrier I of the initial L-algebra, are of the form

$$\phi ::= \surd \mid \langle a \rangle \phi \mid \bigvee \Phi \tag{26}$$

where Φ is a set of formulas ϕ. Trace logics for $\mathcal{P}$ and T satisfy all laws of complete semilattices, and moreover

- (distributivity) $\langle a \rangle \bigvee \Phi = \bigvee \{\langle a \rangle \phi \mid \phi \in \Phi\}$

because I^{Act} (in $LI = F\{\surd\} \times I^{Act}$) is a complete semilattice itself.

The logic becomes finitary (with finitary disjunctions) if we restrict ourselves to finite branching, $B = \mathcal{P}_\omega$. Then we recover the logic of Section 2 and Example 4.4.

6.2 Graded Labelled Transition Systems

The branching type of graded LTSs is the Bag monad $\mathcal{B}$. $\mathcal{B}$-algebras are semiring modules in the natural numbers, $\mathbb{N}$, with addition and multiplication as known in arithmetic. The schizophrenic object $F1$ is $\mathbb{N}$ itself.

$L(-) = F\{\surd\} \times (-)^{Act}$ is a natural choice for the logic functor, as $\delta : LQK \Rightarrow QK\overline{T}$ is then the natural isomorphism $[K(\{*\} + Act \cdot (-)), F1] \cong F\{\surd\} \times [K-, F1]^{Act}$. However, finite products and coproducts coincide in categories of semiring modules under the isomorphism taking tuples (a, b) to linear combinations $a \oplus b$, so that we can define equivalently $L(-) := F\{\surd\} + Act \cdot (-)$. Note also, that $[F\{*\}, F1] \cong F\{\surd\}$.

We obtain the language of trace logics for $\mathcal{B}$ and T by iteration along the initial L-sequence.

- $K\emptyset = \{0\}$ contains the empty linear combination
- $LK\emptyset = F\{\surd\} = \{n \cdot \surd \mid n \in \mathbb{N}\}$ contains the formulas specifying with which grade a successful termination can be reached
- $L^2K\emptyset = F\{\surd\} + Act \cdot (F\{\surd\}) = \{n \cdot \surd, \langle a \rangle \cdot (n \cdot \surd) \mid n \in \mathbb{N}, a \in Act\}$ contains the formulas specifying the grade of successful termination and for each label $a \in Act$, the grade of successsful termination after passing through a
- and so forth

We see that formulas of I are of the form

$$\phi ::= \surd \mid \langle a \rangle \phi \mid \bigoplus n \cdot \phi \tag{27}$$

where $n \in \mathbb{N}$ and $a \in Act$. For notational convenience we use the binary $\oplus$ in alternation with the finitary $\bigoplus$.

The logic is subject to the module axioms

- (commutativity) $a \oplus b = b \oplus a$
- (identity) $0 \oplus a = a$
- (associativity) $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- (cumulation) $(n \cdot a) \oplus (m \cdot a) = (n + m) \cdot a$
- (multiplication) $n \cdot (m \cdot a) = (n * m) \cdot a$

and the distributivity axiom, which stems from $Act \cdot I$ (in $LI = F\{\surd\} + Act \cdot I$ being a semiring modules.

- (distributivity) $\langle a \rangle \bigoplus_{i<n} n_i \cdot \phi_i = \bigoplus_{i<n} n_i \cdot \langle a \rangle \phi_i$

Note that the semiring $\langle \mathbb{N}, +, *, 0, 1 \rangle$ can be replaced with any other semiring, for instance with $\langle \mathbb{N}, +, min, 0, \infty \rangle$.

6.3 Probabilistic Labelled Transition Systems

Probabilistic branching cannot be modelled with a semiring monad, because instead of binary linear combination, probabilism requires convex combinations. The subdistribution monad provides the type for probabilistic LTSs.

Similarly to graded LTSs, we can find the following two definitions for L. First, $L(-) = F\{\surd\} \times (-)^{Act}$ with denotation $\delta : LQK \Rightarrow QK\overline{T}$ being the natural isomorphism $F\{\surd\} \times [K(-), F1]^{Act} \cong [F\{*\} + Act \cdot (-), F1]$. Second, $L(-) = F\{\surd\} + Act \cdot (-)$ with the denotation δ defined such that

$$\delta_X(p \cdot \surd \in F1) := \left\{ \begin{cases} q \cdot \surd \mapsto (p * q) \cdot \surd \\ \langle a \rangle x \mapsto 0 \end{cases} \mid a \in Act, x \in X \right\}$$

$$\delta_X(\langle a \rangle (f : X \to F1) \in Act \cdot \mathcal{D}_{\leq 1}\text{-}Alg(X, F1)) := \left\{ \begin{cases} q \cdot \surd \mapsto 0 \\ \langle a \rangle x \mapsto f(x) \\ \langle b \rangle x \mapsto 0 \end{cases} \mid b \in Act, b \neq a, x \in X \right\}$$

Because products and coproducts do not coincide in $\mathcal{D}_{\leq 1}$-Alg, we obtain in fact two different logics. We choose the more natural functor $L(-) = F\{\surd\} + (-)^{Act}$. The initial sequence L resembles the one for graded LTSs, because also for $\mathcal{D}_{\leq 1}$, $[F1, F1] \cong F1$. Note that the choice of L and δ has implications on the expressivity of the so obtained trace logics.

Formulas of trace logics for probabilistic LTSs are then

$$\phi ::= \surd \mid \langle a \rangle \phi \mid \bigoplus p \cdot \phi \tag{28}$$

where $n \in [0, 1]$ and $a \in Act$.

As a $\mathcal{D}_{\leq 1}$-algebra, trace logic for $\mathcal{D}_{\leq 1}$ and T are subject to the following axioms.

- (commutativity) $\bigoplus_{i<n} p_i \cdot \psi_i = \bigoplus_{i<n} p_{\rho(i)} \cdot \psi_{\rho(i)}$ for any permutation ρ on $\{i < n\}$
- (cumulation) let $\psi_k = \psi_l$ for some $k < l < n$, then $\bigoplus_{i<n} p_i \cdot \psi_i = (\bigoplus_{i<k} p_i \cdot \psi_i) \oplus ((p_k + p_l) \cdot \psi_k) \oplus (\bigoplus_{k<i<l} p_i \cdot \psi_i) \oplus (\bigoplus_{l<n} p_i \cdot \psi_i)$

- (padding) $\bigoplus_{i<n} p_i \cdot \psi_i = \bigoplus_{i<m} p_i \cdot \psi_i$ such that $n < m$ and $p_i = 0$ for all $n \leq i < m$.

The padding axioms gives generality to the next axiom, which describes the distribution of branching over transition steps.

- (distributivity) $\langle a \rangle \bigoplus_{i<n} p_i \cdot \psi_{ij} = \bigoplus_{i<n} p_i \cdot \langle a \rangle \psi_{ij}$

The above axioms allow formulas of I to be normalised into linear combinations of formulas specifying individual finite Act-traces.

6.4 *Labelled Transition Systems of Cost Optimal Paths*

Labelled transition systems of cost optimal paths are branching in a monad for the min-semiring $\langle \mathbb{N} \cup \{\infty\}, min, +, \infty, 0 \rangle$ as in Example 3.3. We read natural numbers and ∞ as costs, where ∞ means unattainable. Costs accumulate along paths $(+)$ and at each node of branching, the optimal branch is chosen (min). In finite trace semantics of such transition systems, we can adjoin more expensive paths of the same length and with the same labels without changing the semantics.

Besides the usual axioms for semiring modules, such as commutativity, associativity, and compatibility as laid out in Section 6.2, we note the following axioms resulting from the properties of the min-semiring:

- (idempotency) $n \cdot x \oplus n \cdot x = n \cdot x$
- (absorption) $0 \cdot x \oplus n \cdot x = 0 \cdot x$

Essentially, finite trace semantics assigns to each point x the set of finite cost-optimal traces descending from x. Such traces are essentially finite Act-words with total costs adjoint.

Formulas for generic trace logics for such transition systems inhabit (right) min-semiring modules. The semantic map $[\![-]\!]$ assigns to a formula and a point in the modelling coalgebra the least grade with which that point satisfies the formula. In free (un-quotiented) trace logics, a formula can be thought of as a statement about the costs of the cost-optimal finite paths with given labels.

7 Future Work

We believe that the generality of monads as branching types bears many more interesting examples that could enrich coalgebra as the theory of general state-based transition systems, and that an investigation of trace logics could contribute to the understanding of branching types for coalgebras. We outline an example underpinning our opinion.

Bistarelli, Montanari and Rossi described [15] constraint satisfaction problems (CSPs) as elements of a semiring with multiplication $\times$ being projection and with $+$ being composition of CSPs. The functor which takes a set X into the set of maps $X \to S$ into a semiring S with finite support is a monad, and so is the functor which takes a set X into the set of finite constraint problems over X. The multiplication of the monad uses the multiplication of the semiring. A trace through a coalgebra with branching in a CSP semiring is a CSP over structures of transitions, and a statement in trace logics could be conceived as an assertion about the structure of transitions to pass.

References

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

[2] de Vink, E. P. and J. J. M. M. Rutten, *Bisimulation for probabilistic transition systems: a coalgebraic approach*, in: *Proceedings of ICALP'97*, LNCS **1256**, 1997.

[3] Hasuo, I., "Tracing Anonymity with Coalgebras," Ph.D. thesis (2008).

[4] Hasuo, I., B. Jacobs and A. Sokolova, *Generic trace theory*, in: *International Workshop on Coalgebraic Methods in Computer Science (CMCS 2006)*, Elect. Notes in Theor. Comp. Sci. **164** (2006), pp. 47–65.

[5] Jacobs, B., *Trace semantics for coalgebras*, Electronic Notes in Theoretical Computer Science **106** (2004), pp. 167–184.
URL `http://dx.doi.org/10.1016/j.entcs.2004.02.031`

[6] Klin, B., *Bialgebraic operational semantics and modal logic*, in: *LICS '07: Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science* (2007), pp. 336–345.
URL `http://dx.doi.org/10.1109/LICS.2007.13`

[7] Kock, A., *Monads on symmetric monoidal closed categories*, Archiv der Mathematik **21** (1970), pp. 1–10.
URL `http://dx.doi.org/10.1007/BF01220868`

[8] Kupke, C., A. Kurz and D. Pattinson, *Algebraic semantics for coalgebraic logics*, in: J. Adàamek and S. Milius, editors, *Coalgebraic Methods in Computer Science (CMCS 2004)*, Electr. Notes in Theoret. Comp. Sci. **106**, 2004, pp. 219–241.

[9] Kupke, C., A. Kurz and Y. Venema, *Completeness of the finitary Moss logic*, in: C. Areces and R. Goldblatt, editors, *Advances in Modal Logic*.

[10] Kurz, A., *Coalgebras and their logics*, SIGACT News **37** (2006), pp. 57–77.
URL `http://dx.doi.org/10.1145/1140612.1140628`

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

[12] Kurz, A. and R. Leal, *Equational coalgebraic logic*, in: *MFPS'09*.

[13] Kurz, A. and J. Rosický, *The goldblatt-thomason theorem for coalgebras*, 2007 pp. 342–355.
URL `http://dx.doi.org/10.1007/978-3-540-73859-6_23`

[14] Maclane, S., "Categories for the Working Mathematician," Graduate Texts in Mathematics, Springer, New York, 1998, 2nd edition edition.

[15] Montanari, U., F. Rossi and S. Bistarelli, *Constraint solving over semirings - kfupm eprints* (1995).
URL `http://eprints.kfupm.edu.sa/32056/`

[16] Moss, L. S., *Coalgebraic logic*, Annals of Pure and Applied Logic **96** (1999), pp. 277–317.
URL `citeseer.ist.psu.edu/moss99coalgebraic.html`

[17] Pattinson, D., *Coalgebraic modal logic: Soundness, completeness and decidability of local consequence*, Theoretical Computer Science **309** (2003).

[18] Rutten, J. J. M. M., *Coinductive counting with weighted automata* (2002).
URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.2930`

[19] Schröder, L., *Expressivity of coalgebraic modal logic: The limits and beyond*, 2005 pp. 440–454.
URL `http://www.springerlink.com/content/66nr7029lrryc1ue`