Generic Trace Semantics and Graded Monads*

Stefan Milius[1], Dirk Pattinson[2], and Lutz Schröder[1]

1 **Friedrich-Alexander-Universität Erlangen-Nürnberg**
2 **The Australian National University**

Abstract

Models of concurrent systems employ a wide variety of semantics inducing various notions of process equivalence, ranging from linear-time semantics such as trace equivalence to branching-time semantics such as strong bisimilarity. Many of these generalize to system types beyond standard transition systems, featuring, for example, weighted, probabilistic, or game-based transitions; this motivates the search for suitable coalgebraic abstractions of process equivalence that cover these orthogonal dimensions of generality, i.e. are generic both in the system type and in the notion of system equivalence. In recent joint work with Kurz, we have proposed a parametrization of system equivalence over an embedding of the coalgebraic type functor into a monad. In the present paper, we refine this abstraction to use *graded monads*, which come with a notion of depth that corresponds, e.g., to trace length or bisimulation depth. We introduce a notion of graded algebras and show how they play the role of formulas in trace logics.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages, F.4.1 Mathematical Logic, D.3.1 Formal Definitions and Theory

Keywords and phrases Traces, transition systems, monads, coalgebra, trace logics

Digital Object Identifier 10.4230/LIPIcs.CALCO.2015.251

1 Introduction

Concurrent systems are typically modelled as state-based systems of some form, with a notion of state transition. Often, transitions between states are given by a transition relation, i.e. the system records only whether or not a transition between two given states is possible. More generally, however, the transition system may implement a more fine-grained modelling that specifies also, for example, the probability or the weight of a given transition, games determining transitions depending on the choices of participating agents, or sets of jointly reachable states. The aim of universal coalgebra [25] is to provide a unified framework for the treatment of various system types such as these.

A core topic in concurrent systems are notions of observable equivalence, which range from linear-time equivalences to branching-time equivalences [33]. While branching-time equivalences, based on suitable notions of bisimilarity, fit in seamlessly with the coalgebraic paradigm [25, 32, 11], other, in particular linear-time, equivalences require more effort. Coalgebraic treatments of trace semantics have previously been based on splitting the functor into a monad, the so-called *branching type*, and a functor, the *transition type*; the transition type is then transferred, by means of suitable distributive laws, to the Kleisli category [12] or the Eilenberg-Moore category [17, 14, 30, 5] of the branching monad, and trace equivalence is cast as bisimilarity in the new category.

* The first and the last author acknowledge support by the German Research Council (DFG) in the project COAX (MI 717/5-1/SCHR 1118/12-1).

© Stefan Milius, Dirk Pattinson, and Lutz Schröder;
licensed under Creative Commons License CC-BY

In recent joint work with Kurz [19], we have proposed a quite simple-minded approach where the coalgebraic type functor, for which no particular form of decomposition needs to be assumed, is embedded into a monad M; *monadic trace semantics* is then obtained by iterating the coalgebra structure in the Kleisli category of M. This approach covers the standard examples including traces without explicit termination (not handled in other approaches) and probabilistic traces; moreover, it subsumes Kleisli-style and Eilenberg-Moore style trace semantics in the sense that these coarser semantics are obtained by applying natural quotient maps to monadic traces for the expected monads.

In more detail, monadic traces for a monad M (e.g. $MX = \mathcal{P}(\Sigma^* \times X)$ in the basic example of labelled transition systems) are sequences of elements of $M1$; the n-th element of such a sequence represents the traces obtained after n transitions. These traces typically all have length n, so the type $M1$ given to the n-th element of the trace sequence is unnecessarily general. This observation motivates using *graded monads*: a graded monad [31] associates to each set X a family of sets M_nX intuitively understood as consisting of the terms of uniform depth n over X in a given algebraic theory. It turns out that the key examples for trace semantics have easier and more general descriptions using graded monads than ordinary monads. We introduce *graded Eilenberg-Moore algebras* for graded monads, and observe that unlike in the modelling via vanilla monads, taking graded Eilenberg-Moore algebras as formulas leads to reasonable *trace logics*, i.e. logics that are invariant under trace equivalence. As a particularly tractable class of graded monads, we identify *depth*-1 *graded monads*, corresponding to algebraic theories with only shallow equations; this class contains most of the leading examples. For depth-1 graded monads, we establish a compositionality result for graded algebras, which amounts to a compositional syntax for trace formulas.

Related Work. We study finite traces, orthogonally to work on coalgebraic infinite traces [7]. The previous coalgebraic treatments of finite traces mentioned above [12, 17, 14, 30, 5] generally restrict to traces ending in accepting states, i.e. focus on language semantics in the sense of automata theory rather than on trace semantics of reactive systems [1]. Especially in the approach via the Eilenberg-Moore category [14, 30, 5], trace semantics is defined via determinization, while in the present paper we opt for a direct definition. Recent work by Klin and Rot [18] is, like the present paper and [17], concerned with trace logics. It takes a principled choice of trace logic as the *definition* of trace equivalence, while we give a semantic definition of trace equivalence and then develop logics that are invariant under trace equivalence. The path-based semantics of linear-time logics considered in [8] implicitly uses Kleisli composition in the graded monad induced by a Kleisli law.

2 Preliminaries

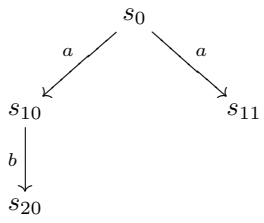
We fix a base category $\mathbf{C}$ throughout. A *G-coalgebra* (X, γ) for a functor $G : \mathbf{C} \to \mathbf{C}$ consists of a $\mathbf{C}$-object X and a morphism $\gamma : X \to GX$. A *coalgebra morphism* between G-coalgebras (X, γ) and (Y, δ) is a morphism $f : X \to Y$ such that $\delta \circ f = Gf \circ \gamma$. When $\mathbf{C} = \mathsf{Set}$, we say that states $x \in X$ and $y \in Y$ in G-coalgebras (X, γ) and (Y, δ) are *behaviourally equivalent* if there exist coalgebra morphisms f, g with common codomain such that $f(x) = g(y)$. Behavioural equivalence can be approximated using the (initial ω-segment of the) *final coalgebra sequence* $(G^n 1)_{n\in\mathbb{N}}$ where 1 is a final object in $\mathbf{C}$ and G^n is n-fold composition of G. The projections $p_n^{n+1} : G^{n+1}1 \to G^n 1$ are defined by induction where $p_0^1 : G1 \to 1$ is the unique arrow to 1 and $p_{n+1}^{n+2} = G(p_n^{n+1})$. For every G-coalgebra (X, γ), there is a *canonical cone* $\gamma_n : X \to G^n 1$ defined inductively by $\gamma_0 : X \to 1$ and $\gamma_{n+1} = G(\gamma_n)\gamma$. We say that states $x \in X$, $y \in Y$ in G-coalgebras (X, γ) and (Y, δ) are *finite-depth behaviourally*

equivalent if $\gamma_n(x) = \delta_n(y)$ for all $n \in \mathbb{N}$. If G is a finitary functor on Set, then finite-depth behavioural equivalence coincides with behavioural equivalence [34].

A *monad* is a triple (M, η, μ) where $M : \mathbf{C} \to \mathbf{C}$ is a functor, and $\eta : Id \to M$ and $\mu : MM \to M$ are natural transformations such that $\mu \circ M\eta = \mu \circ \eta M = id$ and $\mu \circ M\mu = \mu \circ \mu M$. Examples of monads are the powerset monad $\mathcal{P}$ and the distribution monad $\mathcal{D}$, which maps a set X to the set of functions $f : X \to [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. An *Eilenberg-Moore algebra* for a monad M is a morphism $a : MX \to X$ such that $a \circ \eta_X = id_X$ and $a \circ Ma = a \circ \mu_X$.

3 Monadic Trace Semantics, Informally

We recall monad-based trace semantics [19] using the basic example of labelled transition systems, and then motivate the transition to the more fine-grained modelling using graded monads. Consider the labelled transition system (LTS) over the alphabet $\Sigma = \{a, b\}$ depicted on the left below,



	Pretraces	Traces
Stage 0 :	$\{(\epsilon, s_0)\}$	$\{\epsilon\}$
Stage 1 :	$\{(a, s_{10}), (a, s_{11})\}$	$\{a\}$
Stage 2 :	$\{(ab, s_{20}))\}$	$\{ab\}$
Stage 3 :	$\emptyset$	$\emptyset$

whose traces at s_0 are the prefixes of ab. The idea of monadic trace semantics is to produce these traces from what we call *pretraces*, which in the case of LTS are pairs consisting of a trace and a poststate; pretraces are generated incrementally by iterating the coalgebra map representing the system in the Kleisli category of a suitable monad. LTS are standardly modelled as coalgebras for the functor G given on sets by $GX = \mathcal{P}(\Sigma \times X) \cong \mathcal{P}(X)^\Sigma$. We embed G into the monad M given by $MX = \mathcal{P}(\Sigma^* \times X)$ via an evident natural transformation α. Let γ be the G-coalgebra structure representing the above LTS. Then the pretraces of s_0 at stage n are the elements of $(\alpha\gamma)^n(s_0)$ where $(\alpha\gamma)^n$ denotes the n-fold Kleisli composite of $\alpha\gamma : X \to MX$, a morphism $X \to X$ in the Kleisli category of M. The traces are obtained as the first projections of the pretraces, i.e. at each stage the trace set is an element of $M1 \cong \mathcal{P}(\Sigma^*)$, as summarized in the table above right. We observe that the pretraces at stage n all have length n, so that viewing them just as elements of $\mathcal{P}(\Sigma^* \times X)$ loses information. One consequence of this loss of information is that a natural idea for developing a trace logic for a monadic trace semantics fails, as explained in Remark 3.2.

▶ **Remark 3.1.** A property of states is *trace-invariant* if it is closed under trace equivalence. In what follows, by a *trace logic* we mean a compositional syntax for trace-invariant properties. To set the stage for our considerations on trace logics, we remark that being trace-invariant alone is not a compositional property. E.g. in Hennessy-Milner logic, the formula $\Diamond_a \top \wedge \Diamond_b \top$ is trace-invariant – it states that a and b are both traces. Now any sufficiently expressive trace logic for LTS should presumably feature the operator $\Diamond_a$; however, $\Diamond_a(\Diamond_a \top \wedge \Diamond_b \top)$ fails to be trace invariant. Indeed, the (known) logics that characterise trace equivalence in labelled and probabilistic transition systems [3] (necessarily) do not come with the full set of standard boolean connectives.

Note also that in the case of probabilistic trace equivalence, the corresponding trace logic is not interpreted over the standard set $\{\bot, \top\}$ of truth values: for probabilistic systems, a

formula $\Diamond_{a_1} \dots \Diamond_{a_n} \top$ evaluates to the probability of a system exhibiting a trace beginning with $a_1 \dots a_n$.

▶ **Remark 3.2.** In the above standard example, trace sets (at a given stage) are elements of $M1$, the carrier of the free Eilenberg-Moore algebra for M on one generator. A putative trace logic would have formulas whose evaluation at a state depends only on the traces of that state, i.e. the evaluation map will, at each stage, factor through $M1$. It is thus tempting to postulate that the semantics of trace formulas will arise via Eilenberg-Moore algebras for M on the set of intended truth values, say, on $2 = \{\bot, \top\}$ (see also a similar suggestion by Moggi [22]). Specifically, an M-algebra on 2 consists of a complete lattice structure, w.l.o.g. the usual one, and unary operations a, b; since the monad M arises via a distributive law between $\mathcal{P}$ and $\Sigma^* \times (-)$, a and b must moreover be join-continuous. There are only two join-continuous self-maps of 2, the identity and the constant map $\bot$. Thus, an M-algebra on 2 is determined by the subset $\Sigma_0 \subseteq \Sigma$ of letters that it interprets as the identity. Such an algebra yields an operator $\langle \Sigma_0 \rangle$ of our trace logic, and the formula $\langle \Sigma_0 \rangle \top$ holds for a state if each of its traces has a prefix mentioning only actions in Σ_0. However, this constraint is trivially satisfied for every trace by considering the empty prefix. Hence, we clearly want to impose a more precise condition: $\langle \Sigma_0 \rangle \top$ should be satisfied by states all of whose traces start with an action from Σ_0.

Let us make this point more formal. For any formula ϕ of a trace logic, its semantics $[\![\phi]\!]$ should be a sequence of predicates $[\![\phi]\!]_n : M1 \to 2$, $n \in \mathbb{N}$, determining at each stage n the sets of traces satisfying ϕ. Now for any operator L of the logic we expect to have a compositional definition of the semantics of L; that is, given a formula ϕ we wish to define the semantics of $L\phi$ knowing only the semantics $[\![\phi]\!]$ and using the interpretation of L as an M-algebra $[\![L]\!] : M2 \to 2$. With only the monad structure of M available, the only natural definition of the semantics of $L\phi$ that comes to mind requires that $[\![L]\!]M[\![\phi]\!]_n : MM1 \to 2$ factors through $\mu_1 : MM1 \to M1$, yielding $[\![L\phi]\!]_n : M1 \to 2$. But then $[\![L\phi]\!]_n = [\![L\phi]\!]_n \mu_1 M\eta_1 = [\![L]\!]M[\![\phi]\!]_n M\eta_1 = [\![L]\!]M([\![\phi]\!]_n \eta_1)$ so that $[\![L\phi]\!]_n$ depends only on $[\![\phi]\!]_n \eta_1$, i.e. only on whether ϕ holds for the trace set $\{\epsilon\}$. Again, the problem here is that in the preimage of a trace set T under μ_1 we have the element $\{(\{\epsilon\}, T)\}$, i.e. we have no control over the length of prefixes that are split off T in the preimage under μ_1.

It is the core contribution of the current work to show that under a more fine-grained modelling via graded monads, discussed next, Eilenberg-Moore algebras do induce a reasonable notion of trace logic that is expressive enough in several important examples and, for so-called depth-1 graded monads, admits a compositional semantics, in fact following the ideas of the above remark.

4 Graded Monads

We discuss some of the basic theory of graded monads, originally introduced by Smirnov [31] (with grades in an arbitrary commutative monoid; here, we need only the case where grades are natural numbers). Recall that finitary monads on Set correspond to algebraic theories; under this correspondence, the functor part M of a monad may be thought of as mapping a set X to the set MX of terms over X modulo equality. A graded monad has sets M_nX for all $n \in \mathbb{N}$, which may be thought of as sets of terms of uniform depth n.

▶ **Definition 4.1** (Graded Monad)**.** A *graded monad* on $\mathbf{C}$ consists of a family of functors $M_n : \mathbf{C} \to \mathbf{C}$, $n \in \mathbb{N}$, a natural transformation $\eta : Id \to M_0$ (the *unit*), and a family of

natural transformations

$$\mu^{nk} : M_n M_k \to M_{n+k} \qquad (n, k \in \mathbb{N}),$$

the *multiplication*. These data are subject to the *unit laws* for each $n \in \mathbb{N}$ on the left below,

$$\mu^{0n}\eta M_n = id_{M_n} = \mu^{n0} M_n \eta \qquad \begin{array}{ccc} M_n M_k M_m & \xrightarrow{M_n \mu^{km}} & M_n M_{k+m} \\ {\scriptstyle \mu^{nk} M_m}\downarrow & & \downarrow{\scriptstyle \mu^{n,k+m}} \\ M_{n+k} M_m & \xrightarrow[\mu^{n+k,m}]{} & M_{n+k+m} \end{array}$$

and the *associative law* stating that for all $n, k, m \in \mathbb{N}$, the above right diagram commutes.

Notice that the above definition implies that (M_0, η, μ^{00}) is a monad. More abstractly, a graded monad can be defined either as a graded monoid in the endofunctor category $[\mathbf{C}, \mathbf{C}]$ [31] or as a lax monoidal functor $\mathbb{N} \to [\mathbf{C}, \mathbf{C}]$ (with $\mathbb{N}$ viewed as a discrete monoidal category), the latter making it an instance of the notion of parametric monad [21, 16].

Two standard equivalent presentations of monads carry over *mutatis mutandis* to the graded setting, namely Kleisli triples and, over Set, algebraic theories. Graded Kleisli triples have been introduced (in a more general setting) and proved equivalent to graded monads by Katsumata [16]. All we need here is the Kleisli star notation: For $f : X \to M_k Y$, we write

$$f^*_n = \mu^{nk}_Y M_n f : M_n X \to M_{n+k} Y.$$

The presentation of graded monads in terms of graded theories is a stepping stone for isolating depth-1 theories, which in turn are the key to obtaining compositional trace logics:

▶ **Definition 4.2** (Graded theory)**.** A *graded theory* (Σ, E, d) consists of an algebraic theory, i.e. a (possibly class-sized) algebraic signature Σ and a class E of equations, and an assignment d of a *depth* $d(f) \in \mathbb{N}$ to every operation $f \in \Sigma$. This induces a notion of a term *having uniform depth* n: all variables have uniform depth 0, and $f(t_1, \dots, t_n)$ with $d(f) = k$ has uniform depth $n + k$ if all t_i have uniform depth n. (In particular, a constant c has uniform depth n for all $n \geq d(c)$). We then require that all equations $t = s$ in E have uniform depth, i.e. there exists n such that both t and s have uniform depth n. Moreover, we require that for every set X and every $k \in \mathbb{N}$, the class of terms of uniform depth k over variables from X modulo provable equality is small (i.e. in bijection with a set).

We defer the discussion of an example to the next section (Example 5.2.3).

Graded theories and graded monads on Set are essentially equivalent concepts; for the finitary case, this is implicit in [31]. In detail, a graded theory (Σ, E, d) induces a graded monad by taking $M_n X$ to be the set of Σ-terms over X that have uniform depth n, modulo equality derivable under E. Unit and multiplication are then defined as usual as conversion of variables into terms and collapsing of layered terms, respectively, noting that these operations behave as required w.r.t. uniform depth.

Conversely, a graded monad (M_n) over Set induces a graded theory (Σ, E, d) by taking Σ to be the disjoint union of all sets $M_n\kappa$ taken over all $n \in \mathbb{N}$ and all cardinals κ (so Σ is a proper class) and letting $f \in M_n\kappa$ have arity κ and depth n. Then every Σ-term t over X of uniform depth n has a canonical interpretation $[\![t]\!] \in M_n X$ defined recursively in the usual way, noting that this definition does produce an element of $M_n X$. We take E to consist of all equations $s = t$ of uniform depth n over X such that $[\![s]\!] = [\![t]\!]$ in $M_n X$.

Formally, these constructions establish an equivalence of categories between graded monads and graded monad morphisms in the obvious sense on one side, and graded theories and derived theory morphisms on the other side (i.e. maps that take signature symbols to terms, mapping axioms to derivable equations).

Examples We proceed to discuss examples of graded monads; some of these are generic constructions that depend on grades being natural numbers.

▶ **Example 4.3.** **1.** Every monad M with multiplication μ and unit η gives rise to a graded monad, by putting $M_n = M$ and $\mu^{nk} = \mu$.

2. If $F : \mathbf{C} \to \mathbf{C}$ is a functor, then $M_n = F^n$, the n-fold composition of F, defines a graded monad with unit $\eta = id$ and multiplication $\mu^{nk} = id_{F^{n+k}}$.

3. Let $\mathbf{C}$ have binary coproducts, and let $M_0 = Id$ and $M_{n+1} = FM_n + Id$. Define the natural transformations $\epsilon^{n,n+k} : M_n \to M_{n+k}$ by $\epsilon^{00} = id$, $\epsilon^{0,k+1} = \mathsf{inr}$ (right injection) and $\epsilon^{n+1,n+1+k} = F\epsilon^{nk} + Id$, and the multiplication $\mu^{nk} : M_nM_k \to M_{n+k}$ by $\mu^{0k} = id_{M_k}$ and $\mu^{n+1,k} = [\mathsf{inl} \circ F\mu^{nk}, \epsilon^{k,n+k+1}] : FM_nM_k + M_k \to FM_{n+k} + Id$. Then (M_n) is a graded monad with multiplication (μ^{nk}) and unit $\eta = id$. For $\mathbf{C} = \mathsf{Set}$, we may think of M_nX as the set of terms of depth *at most* n in an algebraic theory, i.e. (M_n) is the stratification of the free monad on F.

When $\mathbf{C}$ is monoidal, we have the following more specific example motivated fairly directly by trace semantics. For the sake of readability, we elide coherence isomorphisms.

▶ **Lemma 4.4.** *Let $(\mathbf{C}, \otimes, I)$ be a monoidal category, and let M be a strong monad on $\mathbf{C}$ with unit η, multiplication μ and strength t. Then for every object Σ of $\mathbf{C}$, the assignment*

$$M_nX = M(\Sigma^n \otimes X)$$

with the unit η and the multiplication with components

$$\mu^{nk}_X = (\, M(\Sigma^n \otimes M(\Sigma^k \otimes X)) \xrightarrow{Mt} M^2(\Sigma^{n+k} \otimes X) \xrightarrow{\mu} M(\Sigma^{n+k} \otimes X)\,),$$

where Σ^n denotes the n-th tensor power of the object Σ, defines a graded monad on $\mathbf{C}$.

($\Sigma = I$ yields Example 4.3.1.) We may think of Σ^n as an object of length-n traces; cf. Example 5.2.1.

Another example of graded monads is provided by monads that distribute over a functor by means of a so-called Kleisli law. Given a monad T with multiplication μ and unit η, and a functor F, a *Kleisli law* is a natural transformation $\lambda : FT \to TF$ such that $\lambda \circ F\eta = \eta F$ and $\lambda \circ F\mu = \mu_F \circ T\lambda \circ \lambda T$. It is well-known [23] that Kleisli laws are in 1-1 correspondence with liftings of F to the Kleisli category of M; they also induce graded monads:

▶ **Lemma 4.5.** *Let T be a monad with multiplication μ and unit η, F a functor, and $\lambda : FT \to TF$ a Kleisli-law. Define $\lambda^n : F^nT \to TF^n$ by*

$$\lambda^0 = id \qquad \text{and} \qquad \lambda^n = \lambda^{n-1}F \circ F^{n-1}\lambda.$$

Then the data

$$M_n = TF^n \qquad \mu^{nk} = \mu F^{n+k} \circ T\lambda^n F^k$$

define a graded monad whose unit is the unit η of M.

A related example is obtained from a distributive law of a monad T over an endofunctor F (also called an *EM-law*), i.e., a natural transformation $\delta : TF \to FT$ such that $\lambda \circ \eta F = F\eta$ and $\lambda \circ \mu F = F\mu \circ \lambda T \circ T\lambda$. Such distributive laws are in 1-1 correspondence with liftings of F to the category of Eilenberg-Moore algebras for T [15], and like Kleisli laws they induce graded monads:

▶ **Lemma 4.6.** *Let M be a monad with multiplication μ and unit η, F a functor, and $\lambda : TF \to FT$ an EM-law. Define $\lambda^n : TF^n \to F^nT$ by*

$$\lambda^0 = id \qquad \text{and} \qquad \lambda^n = F\lambda^{n-1} \circ \lambda F^{n-1}.$$

Then the data

$$M_n = F^nT \qquad \mu^{nk} = F^{n+k}\mu \circ F^n\lambda^k T$$

define a graded monad whose unit is the unit η of M.

(Lemmas 4.5 and 4.6 both have Example 4.3.2 as a trivial special case.) This lemma is a 2-categorical dual of Lemma 4.5. Note that no accessibility assumptions on F or T are needed; contrastingly, to obtain a monad from F and T in the situation of the lemma as in [19], one needs to assume that F and T are finitary. Intuitively, a distributive law $TF \to FT$ allows shifting all F-operations to the top of a term only for terms of sufficiently uniform shape, i.e. those in TF^n.

5 Trace Semantics Via Graded Monads

We now give our generic definition of coalgebraic trace semantics, induced by a natural transformation from the coalgebraic type functor into a graded monad.

▶ **Definition 5.1** (Trace semantics)**.** A *trace semantics* for G-coalgebras consists of a graded monad $(M_n)_{n\in\mathbb{N}}$ and a natural transformation

$$\alpha : G \to M_1.$$

The *α-pretrace sequence* $(\gamma^{(n)} : X \to M_nX)_{n\in\mathbb{N}}$ for a G-coalgebra $\gamma : X \to GX$ is defined by induction on n: $\gamma^{(0)} = \eta_X : X \to M_0X$ and

$$\gamma^{(n+1)} = (\gamma^{(n)})^*_1 \circ \alpha_X \circ \gamma = (X \xrightarrow{\alpha\gamma} M_1X \xrightarrow{M_1\gamma^{(n)}} M_1M_nX \xrightarrow{\mu_X^{1n}} M_{n+1}X).$$

The *α-trace sequence* T^α_γ is the sequence

$$(M_n! \circ \gamma^{(n)} : X \to M_n1)_{n\in\mathbb{N}}.$$

Over an unrestricted base category, we just view the α-trace sequence as the α-trace semantics, speaking informally of *α-trace equivalence* as identification under α-trace semantics, and of properties of states being *α-trace invariant* if they depend only on the α-trace sequence. If $\mathbf{C} = \mathsf{Set}$ then states $x \in X$, $y \in Y$ in G-coalgebras (X, γ), (Y, δ) are *α-trace equivalent* if $M_n! \circ \gamma^{(n)}(x) = M_n! \circ \delta^{(n)}(y)$ for all $n \in \mathbb{N}$. We think of M_nX as containing length-n pretraces over X and of M_n1 as containing length-n traces. The morphism $M_n! : M_nX \to M_n1$ forgets the poststate of a pretrace.

The graded monad (M_n) is a parameter of the framework, and typically arises by imposing additional equational laws such as distributivity on the graded monad $(G^n)_{n<\omega}$ of Example 5.2.4.

▶ **Example 5.2.** We proceed to elaborate some concrete instances of trace semantics via graded monads, beginning with our initial motivating example. In all these examples, α is just identity. (As a trivial example where α is not identity, take $M_nX = 1$, which makes all states α-trace equivalent.)

1. *Trace semantics of labelled transition systems:* Labelled transition systems are coalgebras for the functor G defined by $GX = \mathcal{P}(\Sigma \times X)$. To capture standard trace semantics, we define a graded monad $((M_n), \eta, (\mu^{nk}))$ by

$$M_nX = \mathcal{P}(\Sigma^n \times X),$$

with $\eta_X(x) = \{(\epsilon, x)\} \in M_0(X)$ for $x \in X$, and $\mu^{nk}(S) = \{(uv, x) \mid \exists (u, V) \in S.\, (v, x) \in V\}$. This is in fact just a special case of Lemma 4.4. We then have that given a state x in a G-coalgebra $\gamma : X \to \mathcal{P}(\Sigma \times X)$, i.e. in a labelled transition system, $\gamma^{(n)}(x) \in \mathcal{P}(\Sigma^n \times X)$ is the set of pairs (w, y) where w is a length-n trace of x and y is the corresponding poststate. Thus, the n-th component $M_n!\gamma^{(n)}$ of the α-trace sequence maps x to the the set of its length-n traces.

2. *Trace semantics of probabilistic labelled transition systems:* Recall that *generative probabilistic (transition) systems* (for simplicity without the possibility of deadlock, the latter not to be confused with explicit termination) are modelled as coalgebras for the functor $\mathcal{D}(\Sigma \times -)$ where $\mathcal{D}$ denotes the discrete distribution functor (i.e. $\mathcal{D}(X)$ is the set of discrete probability distributions on X, and $\mathcal{D}(f)$ takes image measures under f). That is, a coalgebra structure $\gamma : X \to \mathcal{D}(\Sigma \times X)$ assigns to each state $x \in X$ a probability distribution over pairs of actions and successor states. As in the previous example, we obtain a graded monad for probabilistic trace semantics as an instance of the construction from Lemma 4.4. In this case, we have $M_nX = \mathcal{D}(\Sigma^n \times -)$; $\eta(x)$ is the Dirac distribution at (ϵ, x); and for $\nu \in \mathcal{D}(\Sigma^n \times \mathcal{D}(\Sigma^k \times X))$,

$$\mu^{nk}(\nu)(u, y) = \sum_{u=vw, w\in\Sigma^k, \rho\in\mathcal{D}(\Sigma^k\times X)} \nu(v, \rho)\rho(w, y)$$

for $(u, y) \in \Sigma^{n+k} \times X$. We identify M_n1 with $\mathcal{D}(\Sigma^n)$. Thus, given a state x in a G-coalgebra γ, i.e. in a generative probabilistic system, $(M_n!\gamma^{(n)})(x)(u)$ is the probability of x to exhibit a trace $u \in \Sigma^n$ when run for n steps; this captures the standard notion of probabilistic trace [6].

3. *Mazurkiewicz traces:* The trace semantics proposed by Mazurkiewicz [20] takes concurrent actions into account. Given an action alphabet Σ and an *independence relation* I, i.e., a symmetric and irreflexive relation $I \subseteq \Sigma \times \Sigma$, let W_I denote the monoid obtained by quotienting Σ^* modulo commutation of independent letters, and put $W_I^n = \{[w] \in W_I \mid |w| = n\}$. Then $M_nX = \mathcal{P}(W_I^n \times X)$ models length-n Mazurkiewicz pretraces, consisting of a Mazurkiewicz trace and a poststate. Defining the unit and multiplication analogously as for standard traces, we obtain a graded monad $\mathbb{M}$ for Mazurkiewicz traces.

By the considerations in Section 4, $\mathbb{M}$ corresponds to a graded theory. For the finitely branching case, i.e. replacing $\mathcal{P}$ with the finite powerset functor $\mathcal{P}_\mathsf{f}$, this theory is explicitly described as follows. It has the join semilattice operations and equations as operations and equations of depth 0, and one unary operation a of depth 1, called an *action*, for every $a \in \Sigma$. The theory expresses distribution of actions over the join semilattice structure, by the depth-1 equations

$$a(\bot) = \bot \qquad a(x \vee y) = a(x) \vee a(y), \tag{1}$$

and commutation of independent actions a, b by depth-2 equations $a(b(x)) = b(a(x))$.

4. *Finite-depth behavioural equivalence:* Recall that states in labelled transition systems are *finitely bisimilar* [10] if Duplicator wins all finite-length bisimulation games. More generally, two states in G-coalgebras are *finite-depth behaviourally equivalent* if their images coincide in all stages of the final sequence [29]; see Section 2. Finite-depth behavioural equivalence is an instance of α-trace equivalence: take the graded monad induced by G as in Example 4.3.2, i.e. $M_n = G^n$. Then for a state x in a G-coalgebra γ, $M! \circ \gamma^{(n)}(x) \in G^n 1$ is the image of x in the n-th stage of the final sequence under the canonical cone [19]; i.e. two states are α-trace equivalent iff they are finite-depth behaviourally equivalent.

5. *Kleisli liftings:* If G is of the form $G = TF$ for a functor F and a monad T with a Kleisli law $\lambda : FT \to TF$, then we obtain a graded monad with $M_n = TF^n$ as in Lemma 4.5. The arising α-trace equivalence is typically finer than the language equivalence targeted in previous work on Kleisli-lifting based semantics [12], which for the sake of distinction we shall refer to as *generic language semantics*. E.g. in the base example for language semantics, non-deterministic automata, the coalgebraic model is given by $G = \mathcal{P}(1 + \Sigma \times (-))$ for an alphabet Σ, with $1 = \{\checkmark\}$ denoting explicit termination, i.e. acceptance; here, $T = \mathcal{P}$, and $F = 1 + \Sigma \times -$. Language semantics effectively has TA as its semantic domain, where A is the initial F-algebra; in this case, $TA = \mathcal{P}(\Sigma^*)$, and indeed generic language semantics instantiates exactly to the standard language semantics of nondeterministic automata. In other words, generic language semantics focusses entirely on explicit termination; in cases where the latter is not present, e.g. labelled transition systems, generic language semantics becomes trivial.

Contrastingly, α-trace equivalence in $TF^n 1 \cong \mathcal{P}(\sum_{i=0}^{n} \Sigma^i)$ records, at stage n, not only the accepted words of length at most $n-1$ (gathered in the first $n-1$ summands) but also those words of length n that are traces in the same sense as for labelled transition systems, i.e. can be run without blocking. This is similar in spirit to the denotational semantics of CSP [13], which distinguishes deadlock from successful termination $\checkmark$. Generic language semantics is obtained by just forgetting this last summand, and one can generalize this observation to show that generic language semantics is obtained as a natural quotient of α-trace semantics [19].

6. *Eilenberg-Moore liftings:* An alternative approach to generic language semantics defines trace equivalence as bisimilarity in a generic determinization that can be constructed under certain conditions [14]. We shall refer to this approach as *extension semantics*. It is based on assuming a coalgebraic type functor of the form $G = FT$ where F is a functor and T is a monad, together with a EM-law $\delta : TF \to FT$. The domain of extension semantics is the final coalgebra Z of F. The standard example of non-deterministic automata is subsumed under this approach by taking $FX = 2 \times X^\Sigma$ and $T = \mathcal{P}$. Here, $Z \cong \mathcal{P}(\Sigma^*)$, and extension semantics captures precisely the language semantics of non-deterministic automata. Like Kleisli-style generic language semantics, extension semantics becomes trivial in the absence of explicit termination; e.g. when we change F to be just $FX = X^\Sigma$, then the final F-coalgebra becomes trivial.

In our framework, we define a graded monad with $M_n = F^n T$ as in Lemma 4.6. In the example of non-deterministic automata, we have $M_n 1 = F^n \mathcal{P} 1$ with $FX = 2 \times X^\Sigma$, i.e. $M_n 1$ consists of Σ-branching trees of uniform depth n, with inner nodes labelled in $2 = \{\bot, \top\}$ and leaves in $\mathcal{P}1$. Such a tree may be identified with a set A of words w of length $\leq n$ over Σ: if $|w| < n$ then $w \in A$ iff the inner node addressed by w is labelled by $\top$; w is then *accepted*. If $|w| = n$ then $w \in A$ iff the leaf node addressed by w is labelled by 1; w is then a *trace*, i.e. a state having w in its trace sequence at stage n can execute w without deadlocking. Language equivalence is recovered from α-trace equivalence by canonically forgetting the

information about traces; see [19] for details.

6 Graded Algebras

Fix for this section a graded monad $\mathbb{M} = ((M_n), \eta, (\mu^{nk}))$. As we think of M_n as constructing terms of uniform depth n, it is natural to take graded algebras as providing an interpretation of depth-n-terms to which additional layers can be added uniformly. The M_n-algebras introduced below allow interpreting terms up to uniform depth n, and M_ω-algebras terms of arbitrary depth.

▶ **Definition 6.1** (Graded algebras). For a given natural number n, an *M_n-algebra* $A = ((A_k)_{k\leq n}, (a^{mk})_{m+k\leq n})$ consists of a family of carrier objects A_i and structure morphisms

$$a^{mk} : M_m A_k \to A_{m+k}$$

such that $a^{0m}\eta_{A_m} = id_{A_m}$ for all $m \leq n$, and whenever $m + r + k \leq n$, the diagram on the left

$$\begin{array}{ccc} M_m M_r A_k & \xrightarrow{M_m a^{rk}} & M_m A_{r+k} \\ {\scriptstyle \mu^{mr}_{A_k}}\downarrow & & \downarrow{\scriptstyle a^{m,r+k}} \\ M_{m+r} A_k & \xrightarrow[a^{m+r,k}]{} & A_{m+r+k} \end{array} \qquad \begin{array}{ccc} M_m A_k & \xrightarrow{M_m f_k} & M_m B_k \\ {\scriptstyle a^{mk}}\downarrow & & \downarrow{\scriptstyle b^{mk}} \\ A_{m+k} & \xrightarrow[f_{m+k}]{} & B_{m+k} \end{array} \tag{2}$$

commutes. A *morphism* of M_n-algebras from $A = ((A_k), (a^{mk}))$ to $B = ((B_k), (b^{mk}))$ is a family of morphisms $f_k : A_k \to B_k$, $k \leq n$, such that the right hand diagram above commutes whenever $m + k \leq n$. *M_ω-algebras* and their morphisms are defined similarly but with all indices ranging over $m, k, r \in \mathbb{N}$.

As expected, applying a graded monad to a given set yields a free algebra:

▶ **Proposition 6.2.** *For every $n \in \mathbb{N}$, $FX = ((M_m X)_{m\leq n}, (\mu^{mk})_{m+k\leq n})$ is an M_n-algebra, the free M_n-algebra over X w.r.t. the forgetful functor U_n that maps an M_n-algebra $((A_k), (a^{mk}))$ to A_0 and a morphism (f_k) to f_0; similarly for M_ω-algebras.*

In other words, M_n-algebras realize the monad M_0 by an adjunction; for $n = 0$, we just obtain the usual Eilenberg-Moore construction for M_0. For later use in the semantics of trace formulas, we note

▶ **Proposition 6.3.** *If $\mathbf{C}$ has products, then the category of M_n-algebras has products described as follows. The product of a family of M_n-algebras $A^i = ((A^i_k)_{k\leq n}, (a^{mk}_i)_{m+k\leq n})$ indexed over $i \in I$ has carriers $\prod_{i\in I} A^k_i$ for $k \leq n$ and structure morphisms being composites*

$$M_m \textstyle\prod_{i\in I} A^i_k \xrightarrow{\langle M_m \pi_i\rangle} \prod_{i\in I} M_m A^i_k \xrightarrow{\prod_{i\in I} a^{mk}_i} \prod_{i\in I} A^i_{m+k}.$$

7 Depth-1 Theories

Graded algebras in general need to be constructed monolithically – due to the entanglement between the structure morphisms imposed by Diagram (2), it is not in general possible to combine, say, an M_n-algebra and an M_k-algebra into an M_{n+k}-algebra. A combination mechanism becomes possible, however, if we restrict the depth of equations in the associated graded theory, as follows.

▶ **Definition 7.1** (Depth-1 generation and presentation)**.** We say that a graded theory is *depth-1-generated* if all its operations have depth 1, and *depth-1-presented* or just *depth-1* if additionally all its equations have depth 1. A graded monad on Set is said to have these properties if it can be generated by a corresponding graded theory.

▶ **Example 7.2.** All graded monads in Example 5.2 except the one in Example 5.2.3 are depth-1.

We proceed to develop a more abstract characterization of depth-1 monads usable over arbitrary base categories. Recall that an *epi-transformation* between set functors is a natural transformation with surjective components.

▶ **Proposition 7.3.** *A graded monad $\mathbb{M} = ((M_n), \eta, (\mu^{nk}))$ on* Set *is depth-1 generated iff all μ^{nk} are epi-transformations, equivalently if all μ^{1k} are epi-transformations. Moreover, $\mathbb{M}$ is depth-1 iff additionally the diagram below is a coequalizer diagram for every n:*

$$M_1M_0M_n \underset{\mu^{10}M_n}{\overset{M_1\mu^{0n}}{\rightrightarrows}} M_1M_n \xrightarrow{\mu^{1n}} M_{1+n}\,. \tag{3}$$

▶ **Remark 7.4.** Notice that the coequalizer (3) is reflexive; indeed we have $M_1\mu^{0n} \circ M_1\eta M_n = id_{M_1M_n} = \mu^{10}M_n \circ M_1\eta M_n$ by the unit law of the graded monad.

This motivates the following definition (over unrestricted base categories).

▶ **Definition 7.5.** A graded monad $\mathbb{M} = ((M_n), \eta, (\mu^{nk}))$ is *depth-1 generated* if all μ^{nk} are epi-transformations. Moreover, $\mathbb{M}$ is *depth-1* if it is depth-1 generated and for every n, the diagram (3) is a coequalizer diagram, and $M_0\mu^{1n}$ is an epi-transformation.

▶ **Remark 7.6.** **1.** Proposition 7.3 shows that Definition 7.1 and Definition 7.5 agree where both apply, i.e. for graded monads on Set. The condition that $M_0\mu^{1n}$ be an epi-transformation is automatic in this case, since each μ^{1n}_X is a coequalizer (hence a surjective map) and every functor on Set preserves surjective maps.

2. The condition that $M_0\mu^{1n}$ is an epi-transformation holds as soon as $\mathbf{C}$ is an algebraic category such that every finitely presentable object is regular projective and M_0 is finitary. Indeed, by [2, 6.30] M_0 preserves sifted colimits (and, in particular, reflexive coequalizers). Thus, $M_0\mu^{1n}$ is a (reflexive) coequalizer and therefore an epi-transformation.

The salient point about depth-1 monads is that they allow reducing M_n-algebras to families of M_1-algebras. We begin with morphisms:

▶ **Proposition 7.7.** *If $\mathbb{M}$ is depth-1-generated then given M_n-algebras $((A_k), (a^{kl}))$ and $((B_k), (b^{kl}))$, a family of maps $f_k : A_k \to B_k$ is a morphism of M_n-algebras iff for each $l < n$, (f_l, f_{l+1}) is a morphism of M_1-algebras; i.e. $f_{1+l}a^{1l} = b^{1l}M_1f_l$, and each f_l is a morphism $(A_l, a^{0l}) \to (B_l, b^{0l})$ of M_0-algebras.*

We now present our main technical result, which states essentially that M_n-algebras for depth-1 monads can be assembled from M_1-algebras:

▶ **Theorem 7.8.** *Let $\mathbb{M} = ((M_n), \eta, (\mu^{nk}))$ be a depth-1 graded monad, and let $n \in \mathbb{N}$. Then every family of morphisms*

$$a^{1k} : M_1A_k \to A_{k+1}, \qquad a^{0k} : M_0A_k \to A_k \qquad (k \leq n)$$

such that for each $k < n$, $(a^{0k}, a^{0,k+1}, a^{1k})$ form an M_1-algebra extends uniquely to an M_n-algebra.

In other words, combining this with the previous proposition, we have that in the depth-1-case, an M_n-algebra is just a chain of M_1-algebras with compatible M_0-parts.

▶ **Remark 7.9.** In the corner case where $M_n = F^n$ for an endofunctor F (Example 4.3.2), M_0-algebras are trivial and M_1-algebras are just maps $FA_0 \to A_1$. Therefore, the *graded objects* studied by Ghilardi and Bezhanishvili [9, 4] can formally be seen as M_ω-algebras with additional structure.

8 Trace Logics

We now return to our original goal, to identify a generic notion of α-trace logic, understood as a compositional syntax for α-trace-invariant properties (see Remark 3.1). The key ingredient in our approach is the compositionality of graded algebras for depth-1 monads (Theorem 7.8): We use M_1-algebras as modal operators; by Theorem 7.8, we can build an M_n-algebra out of n such operators. By Proposition 6.2, we can then use M_n-algebras (A_k) as formulas describing α-traces of length n: we fix a truth value, i.e. an element $\tau : 1 \to A_0$, and obtain a morphism $\tau^\#$ of M_n-algebras by free extension. In particular, the diagram

$$\begin{array}{ccc} M_1M_k1 & \xrightarrow{M_1\tau_k^\#} & M_1A_k \\ {\scriptstyle \mu^{1k}}\downarrow & & \downarrow{\scriptstyle a^{1k}} \\ M_{1+k}1 & \xrightarrow[\tau_{1+k}^\#]{} & A_{1+k} \end{array}$$

commutes for $1 + k \leq n$, thus precisely realizing the idea for a compositional semantics of operators that previously failed for ordinary monads (Remark 3.2). Before we introduce more specific syntax, we formally fix the semantics as just indicated:

▶ **Definition 8.1.** An *α-trace property* (A, τ) *of rank* n consists of an M_n-algebra $A = ((A_k), (a^{mk}))$ and a distinguished global element $\tau : 1 \to A_0$ called the *base*. We think of the elements of the A_k as truth values, and refer to A_n as the *type* of (A, τ). The *evaluation* of (A, τ) on a G-coalgebra $\gamma : C \to GC$ is the morphism

$$C \xrightarrow{\gamma^{(n)}} M_nC \xrightarrow{M_n!} M_n1 \xrightarrow{\tau_n^\#} A_n\,,$$

where $\tau^\#$ is the unique homomorphism from the free M_n-algebra on 1, $(M_k1)_{k\leq n}$, to A such that $\tau_0^\#\eta = \tau$. (In particular, α-trace properties are, by definition, α-trace invariant, i.e their evaluation factors through the α-trace sequence.)

We now develop a generic notion of α-trace formula as a syntax for α-trace properties, with a number of syntactic and semantic parameters that can be chosen freely. Given an α-trace property (A, τ), the A_i are, in principle, arbitrary M_0-algebras; however, the current set of examples suggests that it suffices to choose the A_i as powers of a fixed M_0-algebra Ω of truth values. We thus arrive at the following definition of generic α-trace logic.

Syntax We parametrize the syntax over signatures Λ and Θ where Λ consists of modal operators with given finite arities and Θ of truth constants. α-Trace formulas ϕ of rank n are then defined by induction over n: the α-trace formulas of rank 0 are the truth constants; α-trace formulas ϕ of rank $n + 1$ have the form

$$\phi ::= L(\phi_1, \dots, \phi_k)$$

where $L \in \Lambda$ is k-ary, and $\phi_1, \dots, \phi_k$ are α-trace formulas of rank n. (Again, observe that this implies that when L is nullary, the formula L has rank n for every $n \geq 1$.)

Semantics We assume from now on that $\mathbf{C}$ has finite products. As parameters of the semantics, we fix an M_0-algebra Ω with structure map $\omega : M_0\Omega \to \Omega$ serving as an object of truth values, and interpretations of the signature symbols. We let Ω^n denote the n-th Cartesian power of Ω as an M_0-algebra, with structure map $\omega^{(n)}$ (formed as in Proposition 6.3). An n-ary modal operator $L \in \Lambda$ is interpreted as a morphism $[\![L]\!] : M_1(\Omega^n) \to \Omega$ such that $(\omega^{(n)}, \omega, [\![L]\!])$ form an M_1-algebra with carriers $\Omega_0 = \Omega^n$, $\Omega_1 = \Omega$; explicitly, ω and $\omega^{(n)}$ are algebras for the monad M_0 and the diagrams

$$\begin{array}{ccc} M_1M_0(\Omega^n) & \xrightarrow{M_1\omega^{(n)}} & M_1(\Omega^n) \\ {\scriptstyle \mu^{10}}\downarrow & & \downarrow{\scriptstyle [\![L]\!]} \\ M_1\Omega^n & \xrightarrow[[\![L]\!]]{} & \Omega \end{array} \qquad \begin{array}{ccc} M_0M_1(\Omega^n) & \xrightarrow{M_0[\![L]\!]} & M_0\Omega \\ {\scriptstyle \mu^{01}}\downarrow & & \downarrow{\scriptstyle \omega} \\ M_1(\Omega^n) & \xrightarrow[[\![L]\!]]{} & \Omega \end{array}$$

commute. Finally, a truth constant $c \in \Theta$ is interpreted as a truth value $[\![c]\!] : 1 \to \Omega$.

The semantics of an α-trace formula ϕ is an α-trace property $[\![\phi]\!]$ of type Ω, defined recursively as follows. For $c \in \Theta$, we put (overloading notation)

$$[\![c]\!] = (\Omega, [\![c]\!]),$$

an α-trace property of rank 0. For an α-trace formula $L(\phi_1, \dots, \phi_k)$ of rank $n+1$, we form the product of the rank-n α-trace properties $[\![\phi_1]\!], \dots, [\![\phi_k]\!]$; explicitly, this product is formed by taking products of M_n-algebras as in Proposition 6.3, and by tupling the bases (observe that the evaluation of the product according to Definition 8.1 is the tuple formed from the evaluations of the component properties). We thus obtain a rank-n α-trace property $(((A_r)_{r\leq n}, (a^{mr})_{m+r\leq n}), \tau)$ of type Ω^k. Using Theorem 7.8, we then extend the latter to a rank-$(n+1)$ α-trace property $(((A_r)_{r\leq n+1}, (a^{mr})_{m+r\leq n+1}), \tau)$ of type Ω by taking $A_{n+1} = \Omega$, $a^{0,n+1} = \omega$, and $a^{1n} = [\![L]\!] : A_n = \Omega^k \to \Omega = A_{n+1}$.

▶ **Example 8.2.** **1.** *Labelled transition systems.* As truth value object, we take $2 = \{\bot, \top\}$ with the usual join semilattice structure; we put $\Theta = \{\top\}$ and $[\![\top]\!] = \top : 1 \to 2$. We could then take Λ to consist just of unary modal operators of the form $\langle a \rangle$, interpreted as

$$[\![\langle a \rangle]\!] : \mathcal{P}(\Sigma \times 2) \to 2, \quad S \mapsto \begin{cases} \top & (a, \top) \in S \\ \bot & \text{otherwise} \end{cases}$$

much as in Remark 3.2. This defines exactly the usual trace logic for LTS (in particular is already sufficient to distinguish states up to trace equivalence): $\langle a \rangle \phi$ says that there exists a trace that begins with a and continues with a trace satisfying ϕ.

We obtain a slightly more interesting logic by extending Λ with operators of higher arity. Due to the equations imposed by the graded monad $(M_n) = (\mathcal{P}(\Sigma^n \times -))$, an M_1-algebra with carriers 2^k, 2 interprets $a \in \Sigma$ as a join-continuous map $2^k \to 2$; such maps have the form $(b_1, \dots, b_k) \mapsto \bigvee_{i\in I} b_i$ for some $I \subseteq \{1, \dots, k\}$. Thus, we can introduce k-ary operators L of the form

$$L(\phi_1, \dots, \phi_k) = \bigvee_{a\in\Sigma} \langle a \rangle \bigvee_{i\in I_a} \phi_i \qquad (I_a \subseteq \{1, \dots, k\});$$

that is, we enrich the language with disjunction.

2. *Probabilistic trace logic:* Recall that generative probabilistic transition systems are coalgebras for $\mathcal{D}(\Sigma \times -)$, and their trace semantics is given by the graded monad $(M_n) = (\mathcal{D}(\Sigma^n \times -))$; in particular, $M_0 \cong \mathcal{D}$. To obtain a trace logic, we take $\Lambda = \{\langle\Sigma_0\rangle \mid \Sigma_0 \subseteq \Sigma\}$, and $\Theta = \{1\}$. We choose $\Omega = [0,1]$ as the object of truth values, made into a $\mathcal{D}$-algebra by taking expected values, i.e. a formal convex combination $\sum p_i q_i$ over $[0,1]$ is mapped to the arithmetic sum $\sum p_i q_i$. We put $[\![1]\!] = 1 \in [0,1]$. Finally, we interpret the modal operator $\langle\Sigma_0\rangle$ by

$$[\![\langle\Sigma_0\rangle]\!] : \mathcal{D}(\Sigma \times [0,1]) \to [0,1], \qquad \mu \mapsto \textstyle\sum_{a\in\Sigma_0, p\in[0,1]} p\mu(\{(a,p)\}).$$

Then a formula $\langle\Sigma_n\rangle \cdots \langle\Sigma_1\rangle p$ evaluates, at a state c, to p times the probability that c takes a trace in $\Sigma_n \cdots \Sigma_1$; up to the slightly more general syntax, this is exactly the usual trace logic for generative probabilistic transition systems (see, e.g., [3]). Similarly as in the previous example, we can move to a richer language with higher-arity modal operators. As the distributive law behind the multiplication of $\mathcal{D}(\Sigma^n \times -)$ (Example 5.2.2) amounts to requiring that an M_1-algebra with carriers $[0,1]^k$, $[0,1]$ interprets every $a \in \Sigma$ as a morphism $[0,1]^k \to [0,1]$ of $\mathcal{D}$-algebras, we thus extend the language with affine maps (in analogy to adding disjunction in the case of LTS), i.e. with formulas $c + \sum_i q_i\phi_i$, subject to the proviso that $(x_i) \mapsto c + \sum q_i x_i$ defines a map $[0,1]^k \to [0,1]$. In particular, the extended language includes fuzzy negations $1 - \phi$.

3. *Coalgebraic modal logic:* Recall that finite-depth behavioural equivalence on G-coalgebras is α-trace equivalence for the graded monad $M_n X = G^n X$ (Example 5.2.4). Now finite-depth behavioural equivalence is precisely the equivalence described by coalgebraic modal logic for a separating set of predicate liftings (no assumptions are needed on the functor) [24, 26, 28]. The simplest example is Hennessy-Milner logic over labelled transition systems; other examples include probabilistic, graded, and neighbourhood-based logics [27]. In fact, coalgebraic modal logic can be seen as an α-trace logic. Specifically, let Λ be a signature of finitary (possibly nullary) modal operators L, with given interpretations as *predicate liftings* for G. The latter are equivalent to subsets of $G(2^k)$ where k is the arity, i.e. to maps $[\![L]\!] : G(2^k) \to 2$. Predicate liftings are closed under Boolean combination [26], so we can assume that Λ is closed under Boolean combinations. Therefore, we can restrict the syntax of coalgebraic modal logic to nothing but closed terms formed from the operations in Λ (the only other standard ingredient are Boolean operators, now absorbed by Λ).

We define an α-trace logic by taking the same Λ, and $\Theta = \emptyset$; moreover, we take $\Omega = 2$ to be the truth value object. The interpretations Λ already have the required type $G(2^k) \to 2$; since M_0 is the identity monad, and the M_0-algebra structure on 2 is therefore trivial, there are no further conditions to check. Trace formulas over Λ and Θ are, then, exactly the same as formulas in coalgebraic modal logic over Λ, and the semantics is the same in both settings.

▶ **Remark 8.3.** The above examples seem to indicate that there is no single canonical choice for the truth value object Ω. In some cases, the free M_0-algebra $M_0 1$ will do, as in Example 8.2.1 or in a variant of Example 8.2.2 that uses subprobabilities instead of probabilities. As it stands, Ω is isomorphic to $M_0 2$ in Example 8.2.2, similarly in Example 8.2.3. Given Ω, a morphism $M_1(\Omega^n) \to \Omega$ corresponds to an n-ary lifting of Ω-valued predicates for M_1, i.e. a transformation $(\Omega^X)^n \to \Omega^{M_1 X}$, natural in X [26]; we leave the analysis of the predicate liftings arising from the interpretations $[\![L]\!] : M_1(\Omega^n) \to \Omega$ of n-ary modal operators L to future work.

▶ **Remark 8.4.** In all the above examples, the trace logic is *expressive*, i.e. logically equivalent states are α-trace equivalent. In the general case, it is trivial to come up with an expressive

set of α-trace *properties*: just take, for each n, the free M_n-algebra over 1 (Proposition 6.2) as an α-trace property of rank n. Of course, this is uninteresting, as it amounts to just taking trace sets as logical formulas; also, it does not constitute a compositional syntax for α-trace *formulas*. We leave the identification of criteria for expressiveness of a given trace logic to future research.

9 Conclusions and Future Work

We have shown how many forms of trace semantics of coalgebras, including the usual trace semantics of nondeterministic and probabilistic labelled transition systems and Mazurkiewicz traces as well as finite-depth behavioural equivalence, can be modelled uniformly by embedding the coalgebraic type functor into a graded monad. A salient point about this approach is that it constitutes, to our best understanding, the first native semantic definition of generic trace equivalence, while existing approaches start either from a determinization procedure or a trace logic.

We have introduced a notion of graded algebras, which serve as trace-invariant properties. As our main technical result, we have shown that for the more restrictive class of depth-1 monads, graded algebras can be built in a modular fashion. This gives rise to a compositional syntax for trace-invariant logics. We have illustrated how such logics arise for our main examples of trace semantics, thus regaining and extending standard logics in the case of plain and probabilistic traces, and coalgebraic modal logic in the case of finite-depth behavioural equivalence.

Future investigations will be directed at analysing the expressivity as well as algorithmic aspects of trace logics, including the exploration of temporal extensions.

Acknowledgements We wish to thank Alexander Kurz and Tadeusz Litak for useful discussions and pointers to the literature, and Erwin R. Catesbeiana for hints on inconsistent graded monads.

References

1 L. Aceto, A. Ingólfsdóttir, K. Larsen, and J. Srba. *Reactive systems: modelling, specification and verification*. Cambridge Univ. Press, 2007.
2 J. Adámek, J. Rosický, and E. Vitale. *Algebraic Theories*. Cambridge Univ. Press, 2011.
3 M. Bernardo and S. Botta. A survey of modal logics characterising behavioural equivalences for non-deterministic and stochastic systems. *Math. Struct. Comput. Sci.*, 18:29–55, 2008.
4 N. Bezhanishvili and S. Ghilardi. The bounded proof property via step algebras and step frames. *Ann. Pure Appl. Logic*, 165:1832–1863, 2014.
5 M. Bonsangue, S. Milius, and A. Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *ACM Trans. Comput. Log.*, 14, 2013.
6 I. Christoff. Testing equivalences and fully abstract models for probabilistic processes. In *Theories of Concurrency, CONCUR 1990*, vol. 458 of *LNCS*, pp. 126–140. Springer, 1990.
7 C. Cîrstea. A coalgebraic approach to linear-time logics. In *Foundations of Software Science and Computation Structures, FoSSaCS 2014*, vol. 8412 of *LNCS*, pp. 426–440. Springer, 2014.
8 C. Cîrstea. Canonical coalgebraic linear time logics. In *Proc. CALCO*, 2015. This volume.
9 S. Ghilardi. An algebraic theory of normal forms. *Ann. Pure Appl. Logic*, 71:189–245, 1995.
10 V. Goranko and M. Otto. Model theory of modal logic. In P. Blackburn, J. van Benthem, and F. Wolter, eds., *Handbook of Modal Logic*, pp. 249–329. Elsevier, 2006.

11 D. Gorín and L. Schröder. Simulations and bisimulations for coalgebraic modal logics. In *Algebra and Coalgebra in Computer Science, CALCO 2013*, vol. 8089 of *LNCS*, pp. 253–266. Springer, 2013.

12 I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Log. Meth. Comput. Sci.*, 3, 2007.

13 A. Hoare. *Communicating sequential processes*. Prentice Hall, 1985.

14 B. Jacobs, A. Silva, and A. Sokolova. Trace semantics via determinization. In *Coalgebraic Methods in Computer Science, CMCS 2012*, vol. 7399 of *LNCS*, pp. 109–129. Springer, 2012.

15 P. Johnstone. Adjoint lifting theorems for categories of algebras. *Bull. London Math. Soc.*, 7:294–297, 1975.

16 S. Katsumata. Parametric effect monads and semantics of effect systems. In *Principles of Programming Languages, POPL 2014*, pp. 633–646. ACM, 2014.

17 C. Kissig and A. Kurz. Generic trace logics. arXiv preprint 1103.3239, 2011.

18 B. Klin and J. Rot. Coalgebraic trace semantics via forgetful logics. In *Foundations of Software Science and Computation Structures, FoSSaCS 2015*, LNCS. Springer, 2015.

19 A. Kurz, S. Milius, D. Pattinson, and L. Schröder. Simplified coalgebraic trace equivalence. In *Software, Services, and Systems*, vol. 8950 of *LNCS*, pp. 75–90. Springer, 2015.

20 A. Mazurkiewicz. *Concurrent Program Schemes and Their Interpretation.* Aarhus University, Comp. Sci. Depart., DAIMI PB-78, July 1977.

21 P.-A. Mellies. The parametric continuation monad. Preprint, 2015.

22 E. Moggi. Notions of computation and monads. *Inf. Comput.*, 93:55–92, 1991.

23 P. Mulry. Lifting theorems for Kleisli categories. In *Mathematical Foundations of Programming Semantics, MFPS 1993*, vol. 802 of *LNCS*, pp. 304–319. Springer, 1994.

24 D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame J. Formal Logic*, 45:19–33, 2004.

25 J. Rutten. Universal coalgebra: A theory of systems. *Theor. Comput. Sci.*, 249:3–80, 2000.

26 L. Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theor. Comput. Sci.*, 390:230–247, 2008.

27 L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. *ACM Trans. Comput. Log.*, 10:13:1–13:33, 2009.

28 L. Schröder and D. Pattinson. Coalgebraic correspondence theory. In *Foundations of Software Structures and Computer Science, FoSSaCS 2010*, vol. 6014 of *LNCS*, pp. 328–342. Springer, 2010.

29 L. Schröder and D. Pattinson. Rank-1 modal logics are coalgebraic. *J. Log. Comput.*, 20, 2010.

30 A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing determinization from automata to coalgebras. *Log. Meth. Comput. Sci*, 9(1:9), 2013.

31 A. Smirnov. Graded monads and rings of polynomials. *J. Math. Sci.*, 151:3032–3051, 2008.

32 S. Staton. Relating coalgebraic notions of bisimulation. *Log. Meth. Comput. Sci.*, 7, 2011.

33 R. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *Theories of Concurrency, CONCUR 1990*, vol. 458 of *LNCS*, pp. 278–297. Springer, 1990.

34 J. Worrell. On the final sequence of a finitary set functor. *Theor. Comput. Sci.*, 338:184–199, 2005.