Graded Monads and Graded Logics for the Linear Time – Branching Time Spectrum

Ulrich Dorsch
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stefan Milius
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Lutz Schröder
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract

State-based models of concurrent systems are traditionally considered under a variety of notions of process equivalence. In the case of labelled transition systems, these equivalences range from trace equivalence to (strong) bisimilarity, and are organized in what is known as the linear time – branching time spectrum. A combination of universal coalgebra and graded monads provides a generic framework in which the semantics of concurrency can be parametrized both over the branching type of the underlying transition systems and over the granularity of process equivalence. We show in the present paper that this framework of *graded semantics* does subsume the most important equivalences from the linear time – branching time spectrum. An important feature of graded semantics is that it allows for the principled extraction of characteristic modal logics. We have established invariance of these *graded logics* under the given graded semantics in earlier work; in the present paper, we extend the logical framework with an explicit propositional layer and provide a generic expressiveness criterion that generalizes the classical Hennessy-Milner theorem to coarser notions of process equivalence. We extract graded logics for a range of graded semantics on labelled transition systems and probabilistic systems, and give exemplary proofs of their expressiveness based on our generic criterion.

2012 ACM Subject Classification Theory of computation → Concurrency; Theory of computation → Modal and temporal logics

Keywords and phrases Linear Time, Branching Time, Monads, System Equivalences, Modal Logics, Expressiveness

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2019.36

Related Version Full version with all proof details available at `https://arxiv.org/abs/1812.01317`.

Funding This work forms part of the DFG-funded project COAX (MI 717/5-2 and SCHR 1118/12-2).

1 Introduction

State-based models of concurrent systems are standardly considered under a wide range of system equivalences, typically located between two extremes respectively representing *linear time* and *branching time* views of system evolution. Over labelled transition systems, one specifically has the well-known *linear time – branching time spectrum* of system equivalences between trace equivalence and bisimilarity [42]. Similarly, e.g. probabilistic automata have been equipped with various semantics including strong bisimilarity [29], probabilistic (convex) bisimilarity [38], and distribution bisimilarity (e.g. [11, 16]). New equivalences keep appearing in the literature, e.g. for non-deterministic probabilistic systems [5, 43].

This motivates the search for unifying principles that allow for a generic treatment of process equivalences of varying degrees of granularity and for systems of different branching types (non-deterministic, probabilistic etc.). As regards the variation of the branching type, universal coalgebra [35] has emerged as a widely-used uniform framework for state-based systems covering a broad range of branching types including besides non-deterministic and probabilistic, or more generally weighted, branching also, e.g., alternating, neighbourhood-based, or game-based systems. It is based on modelling the system type as an endofunctor on some base category, often the category of sets.

Unified treatments of system equivalences, on the other hand, are so far less well-established, and their applicability is often more restricted. Existing approaches include coalgebraic trace semantics in Kleisli [18] and Eilenberg-Moore categories [5, 6, 23, 26, 39, 43], respectively. Both semantics are based on decomposing the coalgebraic type functor into a monad, the *branching type*, and a functor, the *transition type* (in different orders), and require suitable distributive laws between these parts; correspondingly, they grow naturally out of the functor but on the other hand apply only to functors that admit the respective form of decomposition. In the present work, we build on a more general approach introduced by Pattinson and two of us, based on mapping the coalgebraic type functor into a *graded monad* [31]. Graded monads correspond to algebraic theories where operations come with an explicit notion of *depth*, and allow for a stepwise evaluation of process semantics. Maybe most notably, graded monads systematically support a reasonable notion of *graded logic* where modalities are interpreted as *graded algebras* for the given graded monad. This approach applies to all cases covered in the mentioned previous frameworks, and additional cases that do not fit any of the earlier setups. We emphasize that graded monads are geared towards *inductively* defined equivalences such as finite trace semantics and finite-depth bisimilarity; we leave a similarly general treatment of infinite-depth equivalences such as infinite trace equivalence and unbounded-depth bisimilarity to future work. To avoid excessive verbosity, we restrict to models with finite branching throughout. Under finite branching, finite-depth equivalences typically coincide with their infinite-depth counterparts, e.g. states of finitely branching labelled transition systems are bisimilar iff they are finite-depth bisimilar, and infinite-trace equivalent iff they are finite-trace equivalent.

Our goal in the present work is to illustrate the level of generality achievable by means of graded monads in the dimension of system equivalences. We thus pick a fixed coalgebraic type, that of labelled transition systems, and elaborate how a number of equivalences from the linear time – branching time spectrum are cast as graded monads. In the process, we demonstrate how to extract logical characterizations of the respective equivalences from most of the given graded monads. For the time being, none of the logics we find are sensationally new, and in fact van Glabbeek already provides logical characterizations in his exposition of the linear time – branching time spectrum [42]; an overview of characteristic logics for non-deterministic and probabilistic equivalences is given by Bernardo and Botta [2]. The emphasis in the examples is mainly on showing how the respective logics are developed uniformly from general principles.

Using these examples as a backdrop, we develop the theory of graded monads and graded logics further. In particular,

- we give a more economical characterization of depth-1 graded monads involving only two functors (rather than an infinite sequence of functors);
- we extend the logical framework by a treatment of propositional operators – previously regarded as integrated into the modalities – as first class citizens;
- we prove, as our main technical result, a generic expressiveness criterion for graded logics guaranteeing that inequivalent states are separated by a trace formula.

Our expressiveness criterion subsumes, at the branching-time end of the spectrum, the classical Hennessy-Milner theorem [19] and its coalgebraic generalization [33, 36] as well as expressiveness of probabilistic modal logic with only conjunction [12]; we show that it also covers expressiveness of the respective graded logics for more coarse-grained equivalences along the linear time – branching time spectrum. To illustrate generality also in the branching type, we moreover provide an example in a probabilistic setting, specifically we apply our expressiveness criterion to show expressiveness of a quantitative modal logic for probabilistic trace equivalence.

Related Work. Fahrenberg and Legay [17] characterize equivalences on the linear time – branching time spectrum by suitable classes of modal transition systems. We have already mentioned previous work on coalgebraic trace semantics in Kleisli and Eilenberg-Moore categories [5,6,18,23,26,39,43]. A common feature of these approaches is that, more precisely speaking, they model *language* semantics rather than trace semantics – i.e. they work in settings with explicit successful termination, and consider only successfully terminating traces. When we say that graded monads apply to all scenarios covered by these approaches, we mean more specifically that the respective language semantics are obtained by a further canonical quotienting of our trace semantics [31]. Having said that graded monads are strictly more general than Kleisli and Eilenberg-Moore style trace semantics, we hasten to add that the more specific setups have their own specific benefits including final coalgebra characterizations and, in the Eilenberg-Moore setting, generic determinization procedures. A further important piece of related work is Klin and Rot's method of defining trace semantics via the choice of a particular flavour of trace logic [28]. In a sense, this approach is opposite to ours: A trace logic is posited, and then two states are declared equivalent if they satisfy the same trace formulae. In our approach via graded monads, we instead pursue the ambition of first fixing a semantic notion of equivalence, and then designing a logic that characterizes this equivalence. Like Klin and Rot, we view trace equivalence as an inductive notion, and in particular limit attention to finite traces; coalgebraic approaches to infinite traces exist, and mostly work within the Kleisli-style setup [7–10,20,25,41]. Jacobs, Levy and Rot [22] use corecursive algebras to provide a unifying categorical view on the above-mentioned approaches to traces via Kleisli- and Eilenberg-Moore categories and trace logics, respectively. This framework does not appear to subsume the approach via graded monads, and like for the previous approaches we are not aware that it covers semantics from the linear time – branching time spectrum other than the end points (bisimilarity and trace equivalence).

2 Preliminaries: Coalgebra

We recall basic definitions and results in *(universal) coalgebra* [35], a framework for the unified treatment of a wide range of reactive systems. We write $1 = \{\star\}$ for a fixed one-element set, and $!\colon X \to 1$ for the unique map from a set X into 1. We write $f \cdot g$ for the composite of maps $g\colon X \to Y$, $f\colon Y \to Z$, and $\langle f, g\rangle\colon X \to Y \times Z$ for the pair map $x \mapsto (f(x), g(x))$ formed from maps $f\colon X \to Y$, $g\colon X \to Z$.

Coalgebra encapsulates the branching type of a given species of systems as a *functor*, for purposes of the present paper on the category of sets. Such a functor $G\colon \mathbf{Set} \to \mathbf{Set}$ assigns to each set X a set GX, whose elements we think of as structured collections over X, and to each map $f\colon X \to Y$ a map $Gf\colon GX \to GY$, preserving identities and composition. E.g. the *(covariant) powerset functor* $\mathcal{P}$ assigns to each set X the powerset $\mathcal{P}X$ of X, and to each map $f\colon X \to Y$ the map $\mathcal{P}f\colon \mathcal{P}X \to \mathcal{P}Y$ that takes direct images. (We mostly omit the description of the action of functors on maps in the sequel.) Systems with branching type described by G are then abstracted as *G-coalgebras*, i.e. pairs (X, γ) consisting of a set X of *states* and a map $\gamma\colon X \to GX$, the *transition map*, which assigns to each state $x \in X$ a structured collection $\gamma(x)$ of successors. For instance, a $\mathcal{P}$-coalgebra assigns to each state a set of successors, and thus is the same as a transition system.

▶ **Example 2.1.**

1. Fix a set $\mathcal{A}$ of *actions*. The functor $\mathcal{A} \times (-)$ assigns to each set X the set $\mathcal{A} \times X$; composing this functor with the powerset functor, we obtain the functor $G = \mathcal{P}(\mathcal{A} \times (-))$ whose coalgebras are precisely labelled transition systems (LTS): A G-coalgebra assigns to each state x a set of pairs (σ, y), indicating that y is a successor of x under the action σ.
2. The *(finite) distribution functor* $\mathcal{D}$ maps a set X to the set of finitely supported discrete probability distributions on X. These can be represented as probability mass functions $\mu\colon X \to [0,1]$, with $\sum_{x \in X} \mu(x) = 1$ and with the *support* $\{x \in X \mid \mu(x) > 0\}$ being finite. Coalgebras for $\mathcal{D}$ are precisely Markov chains. Composing with $\mathcal{A} \times (-)$ as above, we obtain the functor $\mathcal{D}(\mathcal{A} \times (-))$, whose coalgebras are *generative probabilistic transition systems*, i.e. assign to each state a distribution over pairs consisting of an action and a successor state.

As indicated in the introduction, we restrict attention to *finitary* functors G, in which every element $t \in GX$ is represented using only finitely many elements of X; formally, each set GX is the union of all sets $Gi_Y[GY]$ where Y ranges over finite subsets of X and i_Y denotes the injection $i_Y : Y \hookrightarrow X$. Concretely, this means that we restrict the set $\mathcal{A}$ of actions to be finite, and work with the *finite powerset functor* $\mathcal{P}_\omega$ (which maps a set X to the set of its finite subsets) in lieu of $\mathcal{P}$. ($\mathcal{D}$ as defined above is already finitary.)

Coalgebra comes with a natural notion of *behavioural equivalence* of states. A *morphism* $f\colon (X,\gamma) \to (Y,\delta)$ of G-coalgebras is a map $f\colon X \to Y$ that commutes with the transition maps, i.e. $\delta \cdot f = Gf \cdot \gamma$. Such a morphism is seen as preserving the behaviour of states (that is, behaviour is defined as being whatever is preserved under morphisms), and consequently states $x \in X$, $z \in Z$ in coalgebras (X,γ), (Z,ζ) are *behaviourally equivalent* if there exist coalgebra morphisms $f\colon (X,\gamma) \to (Y,\delta)$, $g\colon (Z,\zeta) \to (Y,\delta)$ such that $f(x) = g(z)$. For instance, states in LTSs are behaviourally equivalent iff they are bisimilar in the standard sense, and similarly, behavioural equivalence on generative probabilistic transition systems coincides with the standard notion of probabilistic bisimilarity [27]. We have an alternative notion of finite-depth behavioural equivalence: Given a G-coalgebra (X,γ), we define a series of maps $\gamma_n\colon X \to G^n 1$ inductively by taking γ_0 to be the unique map $X \to 1$, and $\gamma_{n+1} = G\gamma_n \cdot \gamma$. (These are the first ω steps of the *canonical cone* from X into the *final sequence* of G [1].) Then states x, y in coalgebras (X,γ), (Z,ζ) are *finite-depth behaviourally equivalent* if $\gamma_n(x) = \zeta_n(y)$ for all n; in the case where G is finitary, finite-depth behavioural equivalence coincides with behavioural equivalence [44].

3 Graded Monads and Graded Theories

We proceed to recall background on system semantics via graded monads introduced in our previous work [31]. We formulate some of our results over general base categories $\mathbf{C}$, using basic notions from category theory [30, 34]; for the understanding of the examples, it will suffice to think of $\mathbf{C} = \mathbf{Set}$. Graded monads were originally introduced by Smirnov [40] (with grades in a commutative monoid, which we instantiate to the natural numbers):

▶ **Definition 3.1** (Graded Monads). A *graded monad* M on a category $\mathbf{C}$ consists of a family of functors $(M_n\colon \mathbf{C} \to \mathbf{C})_{n<\omega}$, a natural transformation $\eta\colon \mathrm{Id} \to M_0$ (the *unit*) and a family of natural transformations $\mu^{nk}\colon M_n M_k \to M_{n+k}$ for $n, k < \omega$, (the *multiplication*), satisfying the *unit laws*, $\mu^{0n} \cdot \eta M_n = \mathrm{id}_{M_n} = \mu^{n0} \cdot M_n\eta$, for all $n < \omega$, and the *associative law* $\mu^{n,k+m} \cdot M_n\mu^{km} = \mu^{n+k,m} \cdot \mu^{nk} M_m$ for all $k, n, m < \omega$.

Note that it follows that (M_0, η, μ^{00}) is a (plain) monad. For $\mathbf{C} = \mathbf{Set}$, the standard equivalent presentation of monads as algebraic theories carries over to graded monads. Whereas for a monad T, the set TX consists of terms over X modulo equations of the corresponding algebraic theory, for a graded monad $(M_n)_{n<\omega}$, M_nX consists of terms of uniform depth n modulo equations:

▶ **Definition 3.2** (Graded Theories [31])**.** A *graded theory* (Σ, E, d) consists of an algebraic theory, i.e. a (possibly class-sized and infinitary) algebraic signature Σ and a class E of equations, and an assignment d of a *depth* $d(f) < \omega$ to every operation symbol $f \in \Sigma$. 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 require that all equations $t = s$ in E have uniform depth, i.e. that both t and s have uniform depth n for some n. Moreover, we require that for every set X and every $k < \omega$, the class of terms of uniform depth k over variables from X modulo provable equality is small (i.e. in bijection with a set).

Graded theories and graded monads on **Set** are essentially equivalent concepts [31, 40]. In particular, a graded theory (Σ, E, d) induces a graded monad M by taking M_nX to be the set of Σ-terms over X of uniform depth n, modulo equality derivable under E.

▶ **Example 3.3.** We recall some examples of graded monads and theories [31].

1. For every endofunctor F on $\mathbf{C}$, the n-fold composition $M_n = F^n$ yields a graded monad with unit $\eta = \mathrm{id}_{\mathrm{Id}}$ and $\mu^{nk} = \mathrm{id}_{F^{n+k}}$.
2. As indicated in the introduction, distributive laws yield graded monads: Suppose that we are given a monad (T, η, μ), an endofunctor F on $\mathbf{C}$ and a distributive law of F over T (a so-called *Kleisli law*), i.e. a natural transformation $\lambda\colon FT \to TF$ such that $\lambda \cdot F\eta = \eta F$ and $\lambda \cdot F\mu = \mu F \cdot T\lambda \cdot \lambda T$. Define natural transformations $\lambda^n\colon F^nT \to TF^n$ inductively by $\lambda^0 = \mathrm{id}_T$ and $\lambda^{n+1} = \lambda^n F \cdot F^n\lambda$. Then we obtain a graded monad with $M_n = TF^n$, unit η, and multiplication $\mu^{nk} = \mu F^{n+1} \cdot T\lambda^n F^k$. The situation is similar for distributive laws of T over F (so-called *Eilenberg-Moore laws*).
3. As a special case of 2., for every monad (T, η, μ) on **Set** and every set $\mathcal{A}$, we obtain a graded monad with $M_nX = T(\mathcal{A}^n \times X)$. Of particular interest to us will be the case where $T = \mathcal{P}_\omega$, which is generated by the algebraic theory of join semilattices (with bottom). The arising graded monad $M_n = \mathcal{P}_\omega(\mathcal{A}^n \times X)$, which is associated with trace equivalence, is generated by the graded theory consisting, at depth 0, of the operations and equations of join semilattices, and additionally a unary operation of depth 1 for each $\sigma \in \mathcal{A}$, subject to (depth-1) equations expressing that these unary operations distribute over joins.

Depth-1 Graded Monads and Theories where operations and equations have depth at most 1 are a particularly convenient case for purposes of building algebras of graded monads; in the following, we elaborate on this condition.

▶ **Definition 3.4** (Depth-1 Graded Theory [31])**.** A graded theory is called *depth*-1 if all its operations and equations have depth at most 1. A graded monad on **Set** is *depth-1* if it can be generated by a depth-1 graded theory.

▶ **Proposition 3.5** (Depth-1 Graded Monads [31])**.** *A graded monad $((M_n), \eta, (\mu^{nk}))$ on* **Set** *is depth-1 iff the diagram below is objectwise a coequalizer diagram in $\mathbf{Set}^{M_0}$ for all $n < \omega$:*

$$M_1 M_0 M_n \underset{\mu^{10} M_n}{\overset{M_1 \mu^{0n}}{\rightrightarrows}} M_1 M_n \xrightarrow{\mu^{1n}} M_{1+n}\,. \tag{1}$$

▶ **Example 3.6.** All graded monads in Example 3.3 are depth 1: for 1., this is easy to see, for 3., it follows from the presentation as a graded theory, and for 2., see [15].

One may use the equivalent property of Proposition 3.5 to define depth-1 graded monads over arbitrary base categories [31]. We show next that depth-1 graded monads may be specified by giving only M_0, M_1, the unit η, and μ^{nk} for $n + k \leq 1$.

▶ **Theorem 3.7.** *Depth-*1 *graded monads are in bijective correspondence with* 6*-tuples* $(M_0, M_1, \eta, \mu^{00}, \mu^{10}, \mu^{01})$ *such that the given data satisfy all applicable instances of the graded monad laws.*

Semantics via Graded Monads. We next recall how graded monads define *graded semantics*:

▶ **Definition 3.8** (Graded semantics [31])**.** Given a set functor G, a *graded semantics* for G-coalgebras consists of a graded monad $((M_n), \eta, (\mu^{nk}))$ and a natural transformation $\alpha\colon G \to M_1$. The α*-pretrace sequence* $(\gamma^{(n)}\colon X \to M_nX)_{n<\omega}$ for a G-coalgebra $\gamma\colon X \to GX$ is defined by

$$\gamma^{(0)} = (X \xrightarrow{\eta_X} M_0X) \quad \text{and} \quad \gamma^{(n+1)} = (X \xrightarrow{\alpha_X \cdot \gamma} M_1X \xrightarrow{M_1\gamma^{(n)}} M_1M_nX \xrightarrow{\mu^{1n}_X} M_{n+1}X).$$

The α-*trace sequence* T^α_γ is the sequence $(M_n! \cdot \gamma^{(n)}\colon X \to M_n1)_{n<\omega}$.

In **Set**, two states $x \in X$, $y \in Y$ of coalgebras $\gamma\colon X \to GX$ and $\delta\colon Y \to GY$ are α-*trace* (or *graded*) *equivalent* if $M_n! \cdot \gamma^{(n)}(x) = M_n! \cdot \delta^{(n)}(y)$ for all $n < \omega$.

Intuitively, M_nX consists of all length-n *pretraces*, i.e. traces paired with a poststate, and M_n1 consists of all length-n traces, obtained by erasing the poststate. Thus, a graded semantics extracts length-1 pretraces from successor structures. In the following two examples we have $M_1 = G$; however, in general M_1 and G can differ (Section 4).

▶ **Example 3.9.** Recall from Section 2 that a G-coalgebra for the functor $G = \mathcal{P}_\omega(\mathcal{A} \times -)$ is just a finitely branching LTS. We recall two graded semantics that model the extreme ends of the linear time – branching time spectrum [31]; more examples will be given in the next section

1. *Trace equivalence.* For $x, y \in X$ and $w \in \mathcal{A}^*$, we write $x \xrightarrow{w} y$ if y can be reached from x on a path whose labels yield the word w, and $\mathcal{T}(x) = \{w \in \mathcal{A}^* \mid \exists y \in X.\ x \xrightarrow{w} y\}$ denotes the set of *traces* of $x \in X$. States x, y are *trace equivalent* if $\mathcal{T}(x) = \mathcal{T}(y)$. To capture trace semantics of labelled transition systems we consider the graded monad with $M_nX = \mathcal{P}(\mathcal{A}^n \times X)$ (see Example 3.3.3). The natural transformation α is the identity. For a G-coalgebra (X, γ) and $x \in X$ we have that $\gamma^{(n)}(x)$ is the set of pairs (w, y) with $w \in \mathcal{A}^n$ and $x \xrightarrow{w} y$, i.e. pairs of length-n traces and their corresponding poststate. Consequently, the n-th component $M_n! \cdot \gamma^{(n)}$ of the α-trace sequence maps x to the set of its length-n traces. Thus, α-trace equivalence is standard trace equivalence [42].
   Note that the equations presenting the graded monad M_n in Example 3.3.3 bear a striking resemblance to the ones given by van Glabbeek to axiomatize trace equivalence of processes, with the difference that in his axiomatization actions do not distribute over the empty join. In fact, $a.0 = 0$ is clearly not valid for processes under trace equivalence. In the graded setting, this equation just expresses the fact that a trace which ends in a deadlock after n steps cannot be extended to a trace of length $n + 1$.
2. *Bisimilarity.* By the discussion of the final sequence of a functor G (Section 2), the graded monad with $M_nX = G^nX$ (Example 3.3.1), with α being the identity again, captures finite-depth behavioural equivalence, and hence behavioural equivalence when G is finitary. In particular, on finitely branching LTS, α-trace equivalence is bisimilarity in this case.

4 A Spectrum of Graded Monads

We present graded monads for a range of equivalences on the linear time – branching time spectrum as well as probabilistic trace equivalence for generative probabilistic systems (GPS), giving in each case a graded theory and a description of the arising graded monads. Some of our equations bear some similarity to van Glabbeek's axioms for equality of process terms. There are also important differences, however. In particular, some of van Glabbeek's axioms are implications, while ours are purely equational; moreover, van Glabbeek's axioms sometimes nest actions, while we employ only depth-1 equations (which precludes nesting of actions) in order to enable the extraction of characteristic logics later. All graded theories we introduce contain the theory of join semilattices, or in the case of GPS convex algebras, whose operations are assigned depth 0; we mention only the additional operations needed. We use terminology introduced in Example 3.9.

Completed Trace Semantics refines trace semantics by distinguishing whether traces can end in a deadlock. We define a depth-1 graded theory by extending the graded theory for trace semantics (Example 3.3) with a constant depth-1 operation $\star$ denoting deadlock. The induced graded monad has $M_0X = \mathcal{P}_\omega(X)$, $M_1 = \mathcal{P}_\omega(\mathcal{A} \times X + 1)$ (and $M_nX = \mathcal{P}_\omega(\mathcal{A}^n \times X + \mathcal{A}^{<n})$ where $\mathcal{A}^{<n}$ denotes the set of words over $\mathcal{A}$ of length less than n). The natural transformation $\alpha_X : \mathcal{P}_\omega(\mathcal{A}\times X) \to M_1X$ is given by $\alpha(\emptyset) = \{\star\}$ and $\alpha(S) = S \subseteq \mathcal{A}\times X+1$ for $\emptyset \neq S \subseteq \mathcal{A}\times X$.

Readiness and Failures Semantics refine completed trace semantics by distinguishing which actions are available (readiness) or unavailable (failures) after executing a trace. Formally, given an LTS, seen as a coalgebra $\gamma : X \to \mathcal{P}_\omega(\mathcal{A}\times X)$, we write $I(x) = \mathcal{P}_\omega\pi_1 \cdot \gamma(x) = \pi_1[\gamma(x)]$ (π_1 being the first projection) for the set of actions available at x, the *ready set* of x. A *ready pair* of a state x is a pair $(w, A) \in \mathcal{A}^* \times \mathcal{P}_\omega(\mathcal{A})$ such that there exists z with $x \xrightarrow{w} z$ and $A = I(z)$; a *failure pair* is defined in the same way except that $A \cap I(z) = \emptyset$. Two states are *readiness (failures) equivalent* if they have the same ready (failure) pairs.

We define a depth-1 graded theory by extending the graded theory for trace semantics (Example 3.3) with constant depth-1 operations A for ready (failure) sets $A \subseteq \mathcal{A}$. In case of failures we add a monotonicity condition $A + A \cup B = A \cup B$ on the constant operations for the failure sets. The resulting graded monads both have $M_0X = \mathcal{P}_\omega X$, and moreover $M_1X = \mathcal{P}_\omega(\mathcal{A}\times X+\mathcal{P}_\omega\mathcal{A})$ for readiness and $M_1X = \mathcal{P}^{\downarrow}_\omega(\mathcal{A}\times X+\mathcal{P}_\omega\mathcal{A})$ for failures, where $\mathcal{P}^{\downarrow}_\omega$ is down-closed finite powerset, w.r.t. the discrete order on $\mathcal{A} \times X$ and set inclusion on $\mathcal{P}_\omega\mathcal{A}$. The natural transformation $\alpha_X : \mathcal{P}_\omega(\mathcal{A} \times X) \to M_1X$ is defined by $\alpha_X(S) = S \cup \{\pi_1[S]\}$ for readiness and $\alpha_X(S) = S \cup \{A \subseteq \mathcal{A} \mid A \cap \pi_1[S] = \emptyset\}$ for failures semantics.

Ready Trace and Failure Trace Semantics refine readiness and failures semantics, respectively, by distinguishing which actions are available (ready trace) or unavailable (failure trace) at each step of the trace. Formally, a *ready trace* of a state x is a sequence $A_0a_1A_1 \dots a_nA_n \in (\mathcal{P}_\omega\mathcal{A}\times\mathcal{A})^* \times \mathcal{P}_\omega\mathcal{A}$ such that there exist transitions $x = x_0 \xrightarrow{a_1} x_1 \dots \xrightarrow{a_n} x_n$ where each x_i has ready set $I(x_i) = A_i$. A *failure trace* has the same shape but we require that each A_i is a *failure set* of x_i, i.e. $I(x_i)\cap A_i = \emptyset$. States are *ready (failure) trace equivalent* if they have the same ready (failure) traces.

For ready traces, we define a depth-1 graded theory with depth-1 operations $\langle A, \sigma\rangle$ for $\sigma \in \mathcal{A}$, $A \subseteq \mathcal{A}$ and a depth-1 constant $\star$, denoting deadlock, and equations $\langle A, \sigma\rangle(\sum_{j\in J} x_j) = \sum_{j\in J}\langle A, \sigma\rangle(x_j)$. The resulting graded monad is simply the graded monad capturing completed trace semantics for labelled transition systems where the set of actions is changed from $\mathcal{A}$ to $\mathcal{P}_\omega\mathcal{A} \times \mathcal{A}$. For failure traces, we additionally impose the equation $\langle A, \sigma\rangle(x) + \langle A \cup B, \sigma\rangle(x) = \langle A \cup B, \sigma\rangle(x)$, which in the set-based description of the graded monad corresponds to downward closure of failure sets.

The resulting graded monads both have $M_0X = \mathcal{P}_\omega X$; for ready traces, $M_1X = \mathcal{P}_\omega((\mathcal{P}_\omega\mathcal{A} \times \mathcal{A}) \times X + 1)$ and for failure traces, $M_1X = \mathcal{P}^{\downarrow}_\omega((\mathcal{P}_\omega\mathcal{A} \times \mathcal{A}) \times X + 1)$, where $\mathcal{P}^{\downarrow}_\omega$ is down-closed finite powerset, w.r.t. the order imposed by the above equation.

For ready trace semantics we define the natural transformation $\alpha_X : \mathcal{P}_\omega(\mathcal{A} \times X) \to M_1X$ by $\alpha_X(\emptyset) = \{\star\}$ and $\alpha_X(S) = \{((\pi_1[S], \sigma), x) \mid (\sigma, x) \in S\})$ for $S \neq \emptyset$. For failure traces we define $\alpha_X(\emptyset) = \{\star\}$ and $\alpha(S) = \{((A, \sigma), x) \mid (\sigma, x) \in S, A \cap \pi_1[S] = \emptyset\}$ for $S \neq \emptyset$; note that in the latter case, $\alpha(S)$ is closed under decreasing failure sets.

Simulation Equivalence declares two states to be equivalent if they simulate each other in the standard sense. We define a depth-1 graded theory with the same signature as for trace equivalence but instead of join preservation require only that each σ is monotone, i.e. $\sigma(x + y) + \sigma(x) = \sigma(x + y)$. The arising graded monad M_n is equivalently described as follows. We define the sets M_nX inductively, along with an ordering on M_nX. We take $M_0X = \mathcal{P}_\omega X$, ordered by set inclusion. We then order the elements of $\mathcal{A} \times M_nX$ by the product ordering of the discrete order on $\mathcal{A}$ and the given ordering on M_nX, and take $M_{n+1}X$ to be the set of downclosed subsets of $\mathcal{A} \times M_nX$, denoted $\mathcal{P}^{\downarrow}_\omega(\mathcal{A} \times M_nX)$, ordered by set inclusion. The natural transformation $\alpha_X : \mathcal{P}(\mathcal{A} \times X) \to \mathcal{P}^{\downarrow}_\omega(\mathcal{A} \times \mathcal{P}_\omega(X))$ is defined by $\alpha_X(S) = {\downarrow}\{(s, \{x\}) \mid (s, x) \in S\}$, where $\downarrow$ denotes downclosure.

Ready Simulation Equivalence refines simulation equivalence by requiring additionally that related states have the same ready set. States x and y are *ready similar* if they are related by some ready simulation, and ready simulation equivalent if there are mutually ready similar. The depth-1 graded theory combines the signature for ready traces with the equations for simulation, i.e. only requires the operations $\langle A, \sigma\rangle$ to be monotone.

Probabilistic Trace Equivalence is the standard trace semantics for generative probabilistic systems (GPS), equivalently, coalgebras for the functor $\mathcal{D}(\mathcal{A} \times \mathrm{Id})$ where $\mathcal{D}$ is the monad of finitary distributions (Example 2.1). Probabilistic trace equivalence is captured by the graded monad $M_nX = \mathcal{D}(\mathcal{A}^n \times X)$, as described in Example 3.3.2. The corresponding graded theory arises by replacing the join-semilattice structure featuring in the above graded theory for trace equivalence by the one of *convex algebras*, i.e. the algebras for the monad $\mathcal{D}$. Recall [13, 14] that a convex algebra is a set X equipped with finite convex sum operations: For every $p \in [0, 1]$ there is a binary operation $\boxplus_p$ on X, and these operations satisfy the equations $x \boxplus_p x = x, x \boxplus_p y = y \boxplus_{1-p} x, x \boxplus_0 y = y, x \boxplus_p (y \boxplus_q z) = (x \boxplus_{p/r} y) \boxplus_r z$, where $p, q \in [0, 1]$, $x, y, z \in X$, and $r = (p+(1-p)q) > 0$ (i.e. $p+q > 0$) in the last equation [21]. Again, we have depth-1 operations σ for action $\sigma \in \mathcal{A}$, now satisfying the equations $\sigma(x \boxplus_p y) = \sigma(x) \boxplus_p \sigma(y)$.

5 Graded Logics

Our next goal is to extract *characteristic logics* from graded monads in a systematic way, with *characterizing* meaning that states are logically indistinguishable iff they are equivalent under the semantics at hand. We will refer to these logics as *graded logics*; the implication from graded equivalence to logical indistinguishability is called *invariance*, and the converse implication *expressiveness*. E.g. standard modal logic with the full set of Boolean connectives is invariant under bisimilarity, and the corresponding expressiveness result is known as the *Hennessy-Milner theorem*. This result has been lifted to coalgebraic generality early on, giving rise to the *coalgebraic Hennessy-Milner theorem* [33, 36]. In previous work [31], we have related graded semantics to modal logics extracted from the graded monad in the envisaged fashion. These logics are invariant by construction; the main new result we present here is a generic *expressiveness* criterion, to be discussed in Section 6. The key ingredient in this criterion are *canonical* graded algebras, which we newly introduce here, providing a recursive-evaluation style reformulation of the semantics of graded logics.

A further key issue in characteristic modal logics is the choice of propositional operators; e.g. notice that when $\Diamond_\sigma$ denotes the usual Hennessy-Milner style diamond operator for an action σ, the formula $\Diamond_\sigma\top \wedge \Diamond_\tau\top$ is invariant under trace equivalence (i.e. the corresponding property is closed under under trace equivalence) but the formula $\Diamond_\sigma(\Diamond_\sigma\top \wedge \Diamond_\tau\top)$, built from the former by simply prefixing with $\Diamond_\sigma$, is not, the problem being precisely the use of conjunction. While in our original setup, propositional operators were kept implicit, that is, incorporated into the set of modalities, we provide an explicit treatment of propositional operators in the present paper. Besides adding transparency to the syntax and semantics, having first-class propositional operators will be a prerequisite for the formulation of the expressiveness theorem.

Coalgebraic Modal Logic. To provide context, we briefly recall the setup of *coalgebraic modal logic* [33, 36]. Let 2 denote the set $\{\bot, \top\}$ of Boolean truth values; we think of the set 2^X of maps $X \to 2$ as the set of predicates on X. Coalgebraic logic in general abstracts systems as coalgebras for a functor G, like we do here; fixes a set Λ of *modalities* (unary for the sake of readability); and then interprets a modality $L \in \Lambda$ by the choice of a *predicate lifting*, i.e. a natural transformation

$$[\![L]\!]_X : 2^X \to 2^{GX}.$$

By the Yoneda lemma, such natural transformations are in bijective correspondence with maps $G2 \to 2$ [36], which we shall also denote as $[\![L]\!]$. In the latter formulation, the recursive clause defining the interpretation $[\![L\phi]\!] : X \to 2$, for a modal formula ϕ, as a state predicate in a G-coalgebra $\gamma : X \to GX$ is then

$$[\![L\phi]\!] = (X \xrightarrow{\gamma} GX \xrightarrow{G[\![\phi]\!]} G2 \xrightarrow{[\![L]\!]} 2). \tag{2}$$

E.g. taking $G = \mathcal{P}_\omega(\mathcal{A}\times -)$ (for labelled transition systems), we obtain the standard semantics of the Hennessy-Milner diamond modality $\Diamond_\sigma$ for $\sigma \in \mathcal{A}$ via the predicate lifting

$$[\![\Diamond_\sigma]\!]_X(f) = \{B \in \mathcal{P}_\omega(\mathcal{A} \times X) \mid \exists x. (\sigma, x) \in B \wedge f(x) = \top\} \qquad (\text{for } f : X \to 2).$$

It is easy to see that *coalgebraic modal logic*, which combines coalgebraic modalities with the full set of Boolean connectives, is invariant under finite-depth behavioural equivalence (Section 2). Generalizing the classical Hennessy-Milner theorem [19], the *coalgebraic Hennessy-Milner theorem* [33, 36] shows that conversely, coalgebraic modal logic *characterizes* behavioural equivalence, i.e. logical indistinguishability implies behavioural equivalence, provided that G is finitary (implying coincidence of behavioural equivalence and finite-depth behavioural equivalence) and Λ is *separating*, i.e. for every finite set X, the set

$$\Lambda(2^X) = \{[\![L]\!](f) \mid f \in 2^X\}$$

of maps $GX \to 2$ is jointly injective.

We proceed to introduce the syntax and semantics of graded logics.

Syntax. We parametrize the syntax of *graded logics* over

- a set Θ of *truth constants*,
- a set $\mathcal{O}$ of *propositional operators* with assigned finite arities, and
- a set Λ of *modalities* with assigned arities.

For readability, we will restrict the technical exposition to unary modalities; the treatment of higher arities requires no more than additional indexing (and we will use 0-ary modalities in the examples). E.g. standard Hennessy-Milner logic is given by $\Lambda = \{\Diamond_\sigma \mid \sigma \in \mathcal{A}\}$ and $\mathcal{O}$ containing all Boolean connectives. Other logics will be determined by additional or different modalities, and often by fewer propositional operators. Formulae of the logic are restricted to have uniform depth, where propositional operators have depth 0 and modalities have depth 1; a somewhat particular feature is that truth constants can have top-level occurrences only in depth-0 formulae. That is, formulae $\phi, \phi_1, \dots$ of depth 0 are given by the grammar

$$\phi ::= p(\phi_1, \dots, \phi_k) \mid c \qquad (p \in \mathcal{O}\ k\text{-ary}, c \in \Theta),$$

and formulae ϕ of depth $n+1$ by

$$\phi ::= p(\phi_1, \dots, \phi_k) \mid L\psi \qquad (p \in \mathcal{O}\ k\text{-ary}, L \in \Lambda)$$

where $\phi_1, \dots, \phi_n$ range over formulae of depth $n+1$ and ψ over formulae of depth n.

Semantics. The semantics of graded logics is parametrized over the choice of *a functor G, a depth-1 graded monad $M = ((M_n)_{n<\omega}, \eta, (\mu^{nk})_{n,k<\omega})$, and a graded semantics $\alpha\colon G \to M_1$, which we fix for the remainder of the paper.* It was originally given by translating formulae into *graded algebras* and then defining formula evaluation by the universal property of $(M_n 1)$ as a free graded algebra [31]; here, we reformulate the semantics in a more standard style by recursive clauses, using canonical graded algebras. In general, the notion of graded algebra is defined as follows [31].

▶ **Definition 5.1** (Graded algebras)**.** Let $n < \omega$. A *(graded) M_n-algebra* $A = ((A_k)_{k\le n}, (a^{mk})_{m+k\le n})$ consists of carrier sets A_k and structure maps

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

satisfying the laws

$$\begin{array}{ccc} A_k & \xrightarrow{\eta_{A_k}} & M_0 A_k \\ & \searrow^{=} & \downarrow a^{0k} \\ & & A_k \end{array} \qquad \begin{array}{ccc} M_m M_r A_k & \xrightarrow{M_m a^{rk}} & M_m A_{r+k} \\ \mu^{mr}_{A_k}\downarrow & & \downarrow a^{m,r+k} \\ M_{m+r} A_k & \xrightarrow{a^{m+r,k}} & A_{m+r+k} \end{array} \tag{3}$$

for all $k \le n$ (left) and all m, r, k such that $m + r + k \le n$ (right), respectively. An *M_n-morphism* f from A to an M_n-algebra $B = ((B_k)_{k\le n}, (b^{mk})_{m+k\le n})$ consists of maps $f_k\colon A_k \to B_k$, $k \le n$, such that $f_{m+k} \cdot a^{mk} = b^{mk} \cdot M_m f_k$ for all m, k such that $m + k \le n$.

We view the carrier A_k of an M_n-algebra as the set of algebra elements that have already absorbed operations up to depth k. As in the case of plain monads, we can equivalently describe graded algebras in terms of graded theories: If M is generated by a graded theory $\mathbb{T} = (\Sigma, E, d)$, then an M_n-algebra interprets each operation $f \in \Sigma$ of arity r and depth $d(f) = m$ by maps $f^A_k\colon A^r_k \to A_{m+k}$ for all k such that $m + k \le n$; this gives rise to an inductively defined interpretation of terms (specifically, given a valuation of variables in A_m, terms of uniform depth k receive values in A_{k+m}, for $k + m \le n$), and subsequently to the expected notion of satisfaction of equations.

While in general, graded algebras are monolithic objects, for depth-1 graded monads we can construct them in a modular fashion from M_1-algebras [31]; we thus restrict attention to M_0- and M_1-algebras in the following. We note that an M_0-algebra is just an Eilenberg-Moore algebra for the monad M_0. An M_1-Algebra A consists of M_0-algebras $(A_0, a^{00}\colon M_0A_0 \to A_0)$ and $(A_1, a^{01}\colon M_0A_1 \to A_1)$, and a *main structure map* $a^{10}\colon M_1A_0 \to A_1$ satisfying two instances of the right-hand diagram in (3), one of which says that a^{10} is a morphism of M_0-algebras (*homomorphy*), and the other that the diagram

$$M_1M_0A_0 \underset{M_1a^{00}}{\overset{\mu^{10}}{\rightrightarrows}} M_1A_0 \xrightarrow{a^{10}} A_1, \tag{4}$$

which by the laws of graded monads consists of M_0-algebra morphisms, commutes (*coequalization*). We will often refer to an M_1-algebra by just its main structure map.

We will use M_1-algebras as interpretations of the modalities in graded logics, generalizing the previously recalled interpretation of modalities as maps $G2 \to 2$ in branching-time coalgebraic modal logic. We fix an M_0-algebra Ω of *truth values*, with structure map $o\colon M_0\Omega \to \Omega$ (e.g. for $G = \mathcal{P}_\omega$, Ω is a join semilattice). Powers Ω^n of Ω are again M_0-algebras. A modality $L \in \Lambda$ is interpreted as an M_1-algebra $A = [\![L]\!]$ with carriers $A_0 = A_1 = \Omega$ and $a^{01} = a^{00} = o$. Such an M_1-algebra is thus specified by its main structure map $a^{10}\colon M_1\Omega \to \Omega$ alone, so following the convention indicated above we often write $[\![L]\!]$ for just this map. The evaluation of modalities is defined using canonical M_1-algebras:

▶ **Definition 5.2** (Canonical algebras)**.** The 0-*part* of an M_1-algebra A is the M_0-algebra (A_0, a^{00}). Taking 0-parts defines a functor U_0 from M_1-algebras to M_0-algebras. An M_1-algebra is *canonical* if it is free, w.r.t. U_0, over its 0-part. For A canonical and a modality $L \in \Lambda$, we denote the unique morphism $A_1 \to \Omega$ extending an M_0-morphism $f\colon A_0 \to \Omega$ to an M_1-morphism $A \to [\![L]\!]$ by $[\![L]\!](f)$, i.e. $[\![L]\!](f)$ is the unique M_0-morphism such that the following equation holds:

$$(M_1A_0 \xrightarrow{M_1f} M_1\Omega \xrightarrow{[\![L]\!]} \Omega) = (M_1A_0 \xrightarrow{a^{10}} A_1 \xrightarrow{[\![L]\!](f)} \Omega). \tag{5}$$

▶ **Lemma 5.3.** *An M_1-algebra A is canonical iff* (4) *is a (reflexive) coequalizer diagram in the category of M_0-algebras.*

By the above lemma, we obtain a key example of canonical M_1-algebras:

▶ **Corollary 5.4.** *If M is a depth-1 graded monad, then for every n and every set X, the M_1-algebra with carriers $M_nX, M_{n+1}X$ and multiplication as algebra structure is canonical.*

Further, we interpret truth constants $c \in \Theta$ as elements of Ω, understood as maps $\hat{c}\colon 1 \to \Omega$, and k-ary propositional operators $p \in \mathcal{O}$ as M_0-homomorphisms $[\![p]\!]\colon \Omega^k \to \Omega$. In our examples on the linear time – branching time spectrum, M_0 is either the identity or, most of the time, the finite powerset monad. In the former case, all truth functions are M_0-morphisms. In the latter case, the M_0-morphisms $\Omega^k \to \Omega$ are the join-continuous functions; in the standard case where $\Omega = 2$ is the set of Boolean truth values, such functions f have the form $f(x_1, \dots, x_k) = x_{i_1} \vee \dots \vee x_{i_l}$, where $i_1, \dots, i_l \in \{1, \dots, k\}$. We will see one case where M_0 is the distribution monad; then M_0-morphisms are affine maps.

The semantics of a formula ϕ in graded logic is defined recursively as an M_0-morphism $[\![\phi]\!]\colon (M_n1, \mu_1^{0n}) \to (\Omega, o)$ by

$$[\![c]\!] = (M_01 \xrightarrow{M_0\hat{c}} M_0\Omega \xrightarrow{o} \Omega) \quad [\![p(\phi_1, \dots, \phi_k)]\!] = [\![p]\!] \cdot \langle [\![\phi_1]\!], \dots, [\![\phi_k]\!] \rangle \quad [\![L\phi]\!] = [\![L]\!]([\![\phi]\!]).$$

The evaluation of ϕ in a coalgebra $\gamma\colon X \to GX$ is then given by composing with the trace sequence, i.e. as

$$X \xrightarrow{M_n! \cdot \gamma^{(n)}} M_n 1 \xrightarrow{[\![\phi]\!]} \Omega. \tag{6}$$

In particular, graded logics are, by construction, invariant under the graded semantics.

▶ **Example 5.5** (Graded logics)**.** We recall the two most basic examples, fixing $\Omega = 2$ in both cases, and $\top$ as the only truth constant:

1. *Finite-depth behavioural equivalence:* Recall that the graded monad $M_n X = G^n X$ captures finite-depth behavioural equivalence on G-coalgebras. Since M_0 is the identity monad, M_0-algebras are just sets. Thus, every function $2^k \to 2$ is an M_0-morphism, so we can use all Boolean operators as propositional operators. Moreover, M_1-algebras are just maps $a^{10}\colon GA_0 \to A_1$. Such an M_1-algebra is canonical iff a^{10} is an isomorphism, and modalities are interpreted as M_1-algebras $G2 \to 2$, with the evaluation according to (5) and (6) corresponding precisely to the semantics of modalities in coalgebraic logic (2). Summing up, we obtain precisely coalgebraic modal logic as summarized above in this case. In our running example $G = \mathcal{P}_\omega(\mathcal{A} \times (-))$, we take modalities $\Diamond_\sigma$ as above, with $[\![\Diamond_\sigma]\!]\colon \mathcal{P}_\omega(\mathcal{A} \times 2) \to 2$ defined by $[\![\Diamond_\sigma]\!](S) = \top$ iff $(\sigma, \top) \in S$, obtaining precisely classical Hennessy-Milner logic [19].
2. *Trace equivalence:* Recall that the trace semantics of labelled transition systems with actions in $\mathcal{A}$ is modelled by the graded monad $M_n X = \mathcal{P}_\omega(\mathcal{A}^n \times X)$. As indicated above, in this case we can use disjunction as a propositional operator since $M_0 = \mathcal{P}_\omega$. Since the graded theory for M_n specifies for each $\sigma \in \mathcal{A}$ a unary depth-1 operation that distributes over joins, we find that the maps $[\![\Diamond_\sigma]\!]$ from the previous example (unlike their duals $\Box_\sigma$) induce M_1-algebras also in this case, so we obtain a graded trace logic featuring precisely diamonds and disjunction, as expected.

We defer the discussion of further examples, including ones where $\Omega = [0, 1]$, to the next section, where we will simultaneously illustrate the generic expressiveness result (Example 6.5).

▶ Remark 5.6. One important class of examples where the above approach to characteristic logics will *not* work without substantial further development are simulation-like equivalences, whose characteristic logics need conjunction [42]. Conjunction is not an M_0-morphism for the corresponding graded monads identified in Section 4, which both have $M_0 = \mathcal{P}_\omega$. A related and maybe more fundamental observation is that formula evaluation is not M_0-morphic in the presence of conjunction; e.g. over simulation equivalence, the evaluation map $M_1 1 = \mathcal{P}^{\downarrow}_\omega(\mathcal{A} \times \mathcal{P}_\omega(1)) \to 2$ of the formula $\Diamond_\sigma \top \wedge \Diamond_\tau \top$ fails to be join-continuous for distinct $\sigma, \tau \in \mathcal{A}$. We leave the extension of our logical framework to such cases to future work, expecting a solution in elaborating the theory of graded monads, theories, and algebras over the category of partially ordered sets, where simulations live more naturally (e.g. [24]).

6 Expressiveness

We now present our main result, an expressiveness criterion for graded logics, which states that a graded logic characterizes the given graded semantics if it has enough modalities propositional operators, and truth constants. Both the criterion and its proof now fall into place naturally and easily, owing to the groundwork laid in the previous section, in particular the reformulation of the semantics in terms of canonical algebras:

▶ **Definition 6.1.** We say that a graded logic with set Ω of truth values and sets Θ, $\mathcal{O}$, Λ of truth constants, propositional operators, and modalities, respectively, is

1. *depth-0 separating* if the family of maps $[\![c]\!]\colon M_0 1 \to \Omega$, for truth constants $c \in \Theta$, is jointly injective; and
2. *depth-1 separating* if, whenever A is a canonical M_1-algebra and $\mathfrak{A}$ is a jointly injective set of M_0-homomorphisms $A_0 \to \Omega$ that is closed under the propositional operators in $\mathcal{O}$ (in the sense that $[\![p]\!] \cdot \langle f_1, \dots, f_k \rangle \in \mathfrak{A}$ for $f_1, \dots, f_k \in \mathfrak{A}$ and k-ary $p \in \mathcal{O}$), then the set $\Lambda(\mathfrak{A}) := \{[\![L]\!](f)\colon A_1 \to \Omega \mid L \in \Lambda, f \in \mathfrak{A}\}$ of maps is jointly injective.

▶ **Theorem 6.2** (Expressiveness)**.** *If a graded logic is both depth-0 separating and depth-1 separating, then it is expressive.*

▶ **Example 6.3** (Logics for bisimilarity)**.** We note first that the existing coalgebraic Hennessy-Milner theorem, for branching time equivalences and coalgebraic modal logic with full Boolean base over a finitary functor G [33,36], as recalled in Section 5, is a special case of Theorem 6.2: We have already seen in Example 5.5 that coalgebraic modal logic in the above sense is an instance of our framework for the graded monad $M_nX = G^nX$. Since $M_0 = \mathrm{id}$ in this case, depth-0 separation is vacuous. As indicated in Example 5.5, canonical M_1-algebras are w.l.o.g. of the form $\mathrm{id}\colon GX \to GX$, where for purposes of proving depth-1 separation, we can restrict to finite X since G is finitary. Then, a set $\mathfrak{A}$ as in Definition 6.1 is already the whole powerset 2^X, so depth-1 separation is exactly the previous notion of separation.

A well-known particular case is probabilistic bisimilarity on Markov chains, for which an expressive logic needs only probabilistic modalities $\Diamond_p$ "with probability at least p" and conjunction [12]. This result (later extended to more complex composite functors [32]) is also easily recovered as an instance of Theorem 6.2, using the same standard lemma from measure theory as in *op. cit.*, which states that measures are uniquely determined by their values on a generating set of the underlying σ-algebra that is closed under finite intersections (corresponding to the set $\mathfrak{A}$ from Definition 6.1 being closed under conjunction).

▶ Remark 6.4. For behavioural equivalence, i.e. $M_nX = G^nX$ as in the above example, the inductive proof of our expressiveness theorem essentially instantiates to Pattinson's proof of the coalgebraic Hennessy-Milner theorem by induction over the terminal sequence [33]. One should note that although the coalgebraic Hennessy-Milner theorem can be shown to hold for larger cardinal bounds on the branching by means of a direct quotienting construction [36], the terminal sequence argument goes beyond finite branching only in corner cases.

▶ **Example 6.5** (Expressive graded logics on the linear time – branching time spectrum)**.** We next extract graded logics from some of the graded monads for the linear time – branching time spectrum introduced in Section 4, and show how in each case, expressiveness is an instance of Theorem 6.2. Bisimilarity is already covered by the previous example. Depth-0 separation is almost always trivial and not mentioned further. Unless mentioned otherwise, all logics have disjunction, enabled by M_0 being powerset as discussed in the previous section. Most of the time, the logics are essentially already given by van Glabbeek (with the exception that we show that one can add disjunction) [42]; the emphasis is entirely on uniformization.

1. *Trace equivalence:* As seen in Example 5.5, the graded logic for trace equivalence features (disjunction and) diamond modalities $\Diamond_\sigma$ indexed over actions $\sigma \in \mathcal{A}$. The ensuing proof of depth-1 separation uses canonicity of a given M_1-algebra A only to obtain that the structure map a^{10} is surjective. The other key point is that a jointly injective collection $\mathfrak{A}$ of M_0-homomorphisms $A_0 \to 2$, i.e. join preserving maps, has the stronger separation property that whenever $x \not\leq y$ then there exists $f \in \mathfrak{A}$ such that $f(x) = \top$ and $f(y) = \bot$.

2. Graded logics for completed traces, readiness, failures, ready traces, and failure traces are developed from the above by adding constants or additionally indexing modalities over sets of actions, with only little change to the proofs of depth-1 separation. For completed trace equivalence, we just add a 0-ary modality $\star$ indicating deadlock. For ready trace equivalence, we index the diamond modalities $\Diamond_\sigma$ with sets $I \subseteq \mathcal{A}$; formulae $\Diamond_{\sigma,I}\phi$ are then read 'the current ready set is I, and there is a σ-successor satisfying ϕ'. For failure trace equivalence we proceed in the same way but read the index I as 'I is a failure set at the current state'. For readiness equivalence and failures equivalence, we keep the modalities $\Diamond_\sigma$ unchanged from trace equivalence and instead introduce 0-ary modalities r_I indicating that I is the ready set or a failure set, respectively, at the current state, thus ensuring that formulae do not continue after postulating a ready set.

▶ **Example 6.6** (Probabilistic traces)**.** We have recalled in Section 4 that probabilistic trace equivalence of generative probabilistic transition systems can be captured as a graded semantics using the graded monad $M_nX = \mathcal{D}(\mathcal{A}^n \times X)$, with M_0-algebras being convex algebras. In earlier work [31] we have noted that a logic over the set $\Omega = [0,1]$ of truth values (with the usual convex algebra structure) featuring rational truth constants, affine combinations as propositional operators (as indicated in Section 5), and modal operators $\langle\sigma\rangle$, interpreted by M_1-algebras $[\![\langle\sigma\rangle]\!]\colon M_1[0,1] \to [0,1]$ defined by $[\![\langle\sigma\rangle]\!](\mu) = \sum_{r\in[0,1]} r\mu(\sigma,r)$ is invariant under probabilistic trace equivalence. By our expressiveness criterion, we recover the result that this logic is expressive for probabilistic trace semantics (see e.g. [2]).

7 Conclusion and Future Work

We have provided graded monads modelling a range of process equivalences on the linear time – branching time spectrum, presented in terms of carefully designed graded algebraic theories. From these graded monads, we have extracted characteristic modal logics for the respective equivalences systematically, following a paradigm of graded logics that grows out of a natural notion of graded algebra. Our main technical results concern the further development of the general framework for graded logics; in particular, we have introduced a first-class notion of propositional operator, and we have established a criterion for *expressiveness* of graded logics that simultaneously takes into account the expressive power of the modalities and that of the propositional base. (An open question that remains is whether an expressive logic always exists, as it does in the branching-time setting [36].) Instances of this result include, for instance, the coalgebraic Hennessy-Milner theorem [33, 36], Desharnais et al.'s expressiveness result for probabilistic modal logic with only conjunction [12], and expressiveness for various logics for trace-like equivalences on non-deterministic and probabilistic systems. The emphasis in the examples has been on well-researched equivalences and logics for the basic case of labelled transition systems, aimed at demonstrating the versatility of graded monads and graded logics along the axis of granularity of system equivalence. The framework as a whole is however parametric also over the branching type of systems and in fact over the base category determining the structure of state spaces; an important direction for future research is therefore to capture (possibly new) equivalences and extract expressive logics on other system types such as probabilistic systems (we have already seen probabilistic trace equivalence as an instance; see [4] for a comparison of some equivalences on probabilistic automata, which combine probabilities and non-determinism) and nominal systems, e.g. nominal automata [3, 37]. Moreover, we plan to extend the framework of graded logics to cover also temporal logics, using graded algebras of unbounded depth.

References

1 Jirí Adámek and Václav Koubek. Remarks on Fixed Points of Functors. In *Fundamentals of Computation Theory, FCT 1977*, LNCS, pages 199–205. Springer, 1977. doi:10.1007/3-540-08442-8_86.

2 Marco Bernardo and Stefania Botta. A survey of modal logics characterising behavioural equivalences for non-deterministic and stochastic systems. *Math. Struct. Comput. Sci.*, 18:29–55, 2008.

3 Mikołaj Bojańczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10(3), 2014. doi:10.2168/LMCS-10(3:4)2014.

4 Filippo Bonchi, Alexandra Silva, and Ana Sokolova. The Power of Convex Algebras. In *Concurrency Theory, CONCUR 2017*, volume 85 of *LIPIcs*, pages 23:1–23:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

5 Filippo Bonchi, Ana Sokolova, and Valeria Vignudelli. The Theory of Traces for Systems with Nondeterminism and Probability. In *Logic in Computer Science, LICS 2019*. IEEE, 2019.

6 Marcello Bonsangue, Stefan Milius, and Alexandra Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *ACM Trans. Comput. Log.*, 14, 2013.

7 Corina Cîrstea. Maximal traces and path-based coalgebraic temporal logics. *Theoret. Comput. Sci.*, 412(38):5025–5042, 2011. doi:10.1016/j.tcs.2011.04.025.

8 Corina Cîrstea. A Coalgebraic Approach to Linear-Time Logics. In *Foundations of Software Science and Computation Structures, FoSSaCS 2014*, volume 8412 of *LNCS*, pages 426–440. Springer, 2014.

9 Corina Cîrstea. Canonical Coalgebraic Linear Time Logics. In *Algebra and Coalgebra in Computer Science, CALCO 2015*, Leibniz International Proceedings in Informatics, 2015.

10 Corina Cîrstea. From Branching to Linear Time, Coalgebraically. *Fund. Inform.*, 150(3-4):379–406, 2017. doi:10.3233/FI-2017-1474.

11 Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, and Carroll Morgan. Characterising Testing Preorders for Finite Probabilistic Processes. *Log. Meth. Comput. Sci.*, 4(4), 2008. doi:10.2168/LMCS-4(4:4)2008.

12 Josee Desharnais, Abbas Edalat, and Prakash Panangaden. A Logical Characterization of Bisimulation for Labeled Markov Processes. In *Logic in Computer Science, LICS 1998*, pages 478–487. IEEE Computer Society, 1998.

13 Ernst-Erich Doberkat. Eilenberg-Moore algebras for stochastic relations. *Inf. Comput.*, 204(12):1756–1781, 2006. doi:10.1016/j.ic.2006.09.001.

14 Ernst-Erich Doberkat. Erratum and Addendum: Eilenberg-Moore algebras for stochastic relations. *Inf. Comput.*, 206(12):1476–1484, 2008. doi:10.1016/j.ic.2008.08.002.

15 Ulrich Dorsch, Stefan Milius, and Lutz Schröder. Graded Monads and Graded Logics for the Linear Time – Branching Time Spectrum, 2019. arXiv:1812.01317.

16 Laurent Doyen, Thomas Henzinger, and Jean-François Raskin. Equivalence of Labeled Markov Chains. *Int. J. Found. Comput. Sci.*, 19(3):549–563, 2008. doi:10.1142/S0129054108005814.

17 Uli Fahrenberg and Axel Legay. A Linear-Time-Branching-Time Spectrum of Behavioral Specification Theories. In *Theory and Practice of Computer Science, SOFSEM 2017*, volume 10139 of *LNCS*, pages 49–61. Springer, 2017. doi:10.1007/978-3-319-51963-0.

18 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic Trace Semantics via Coinduction. *Log. Meth. Comput. Sci.*, 3, 2007. doi:10.2168/LMCS-3(4:11)2007.

19 M. Hennessy and R. Milner. Algebraic Laws for Non-Determinism and Concurrency. *J. ACM*, 32:137–161, 1985.

20 Bart Jacobs. Trace Semantics for Coalgebras. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science, CMCS 2004*, volume 106 of *ENTCS*, pages 167–184. Elsevier, 2004. doi:10.1016/j.entcs.2004.02.031.

21 Bart Jacobs. Convexity, Duality and Effects. In C.S. Calude and V. Sassone, editors, *Proc. TCS 2010*, volume 323 of *IFIP AICT*, pages 1–19, 2010.

22 Bart Jacobs, Paul B. Levy, and Jurriaan Rot. Steps and Traces. In Corina Cîrstea, editor, *Proc. CMCS 2018*, volume 11202 of *LNCS*, pages 122–143. Springer, 2018.

23 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace Semantics via Determinization. In *Coalgebraic Methods in Computer Science, CMCS 2012*, volume 7399 of *LNCS*, pages 109–129. Springer, 2012. doi:10.1007/978-3-642-32784-1.

24 Krzysztof Kapulkin, Alexander Kurz, and Jiri Velebil. Expressiveness of Positive Coalgebraic Logic. In *Advances in Modal Logic, AiML 2012*, pages 368–385. College Publications, 2012.

25 Henning Kerstan and Barbara König. Coalgebraic Trace Semantics for Continuous Probabilistic Transition Systems. *Log. Meth. Comput. Sci.*, 9(4), 2013. doi:10.2168/LMCS-9(4:16)2013.

26 Christian Kissig and Alexander Kurz. Generic Trace Logics. arXiv preprint 1103.3239, 2011.

27 Bartek Klin. Structural Operational Semantics for Weighted Transition Systems. In *Semantics and Algebraic Specification*, volume 5700 of *LNCS*, pages 121–139. Springer, 2009.

28 Bartek Klin and Juriaan Rot. Coalgebraic trace semantics via forgetful logics. In *Foundations of Software Science and Computation Structures, FoSSaCS 2015*, volume 9034 of *LNCS*, pages 151–166. Springer, 2015.

29 K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94:1–28, 1991.

30 Saunders MacLane. *Categories for the working mathematician.* Springer, 2nd edition, 1998.

31 Stefan Milius, Dirk Pattinson, and Lutz Schröder. Generic Trace Semantics and Graded Monads. In *Algebra and Coalgebra in Computer Science, CALCO 2015*, Leibniz International Proceedings in Informatics, 2015.

32 Lawrence Moss and Ignacio Viglizzo. Final coalgebras for functors on measurable spaces. *Inf. Comput.*, 204(4):610–636, 2006. doi:10.1016/j.ic.2005.04.006.

33 D. Pattinson. Expressive Logics for Coalgebras via Terminal Sequence Induction. *Notre Dame J. Formal Log.*, 45:19–33, 2004.

34 Benjamin Pierce. *Basic category theory for computer scientists.* MIT Press, 1991.

35 J. Rutten. Universal Coalgebra: A Theory of Systems. *Theoret. Comput. Sci.*, 249:3–80, 2000.

36 Lutz Schröder. Expressivity of Coalgebraic Modal Logic: The Limits and Beyond. *Theoret. Comput. Sci.*, 390:230–247, 2008.

37 Lutz Schröder, Dexter Kozen, Stefan Milius, and Thorsten Wißmann. Nominal Automata with Name Binding. In *Foundations of Software Science and Computation Structures, FOSSACS 2017*, volume 10203 of *LNCS*, pages 124–142, 2017. doi:10.1007/978-3-662-54458-7.

38 Roberto Segala and Nancy Lynch. Probabilistic Simulations for Probabilistic Processes. In *Concurrency Theory, CONCUR 1994*, volume 836 of *LNCS*, pages 481–496. Springer, 1994. doi:10.1007/978-3-540-48654-1.

39 Alexandra Silva, Filippo Bonchi, Marcello Bonsangue, and Jan Rutten. Generalizing determinization from automata to coalgebras. *Log. Methods Comput. Sci*, 9(1:9), 2013.

40 A. Smirnov. Graded Monads and Rings of Polynomials. *J. Math. Sci.*, 151:3032–3051, 2008.

41 Natsuki Urabe and Ichiro Hasuo. Coalgebraic Infinite Traces and Kleisli Simulations. In *Algebra and Coalgebra in Computer Science, CALCO 2015*, volume 35 of *LIPIcs*, pages 320–335. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015.

42 R. van Glabbeek. The Linear Time – Branching Time Spectrum I; The Semantics of Concrete, Sequential Processes. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, pages 3–99. Elsevier, 2001. doi:10.1016/B978-044482830-9/50019-9.

43 Gerco van Heerdt, Justin Hsu, Joël Ouaknine, and Alexandra Silva. Convex Language Semantics for Nondeterministic Probabilistic Automata. In *Theoretical Aspects of Computing. ICTAC 2018*, volume 11187 of *LNCS*, pages 472–492. Springer, 2018. doi:10.1007/978-3-030-02508-3.

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