# Coalgebraic semantics of modal logic

Matthew Wetmore

## 1 Introduction

Modal logic is hardly new; it has been treated both formally and informally in philosophical works over the past few centuries. The modern study of modal logic was initially proof-theoretic; the first half of the 20th century saw the introduction and axiomatization of various systems of modal logic. Tarski and others pioneered an algebraic approach, studying modal logics as extensions of Boolean algebras with operators representing modalities.

The model-theoretic study of modal logic had a later start, but harboured one of the most important developments in the theory of modal logics – Kripke's introduction of possible worlds semantics (PWS) provided concrete structures over which modal formulas could be interpreted. Broadly, Kripke's "possible worlds" reside in a relational structure $(W, R)$ where $W$ is a set of "worlds", and $R$ is a relation between these worlds – given a world $w \in W$, $R$ tells us what other worlds we can get to from $w$.

While Kripke's PWS have applications in the study of the properties of modal logics – soundness, completeness, etc – on a more general, moral level they exemplify that modal logic is the "natural language" of relational structures. Blackburn et al. begin their text *Modal Logic* with slogans reflecting this:

> Modal languages are simple yet expressive languages for talking about relational structures. Modal languages provide an internal, local perspective on relational structures.

This is why modal logics have enjoyed extensive study outside of philosophy and mathematical logic departments, in areas such as computer science and linguistics. Relational structures are everywhere; the canonical example in the field of computer science is the transition system, a concept which is invaluable to formal language theory, model checking, artificial intelligence, and so on.

In this paper, we lift this understanding of (propositional) modal logics with respect to relational structures to a higher level of generality, providing semantics for modal logics in terms of the category-theoretic notion of a *coalgebra*. Coalgebras capture the general notion of state-based systems evolving over time[1] – streams, transition systems, infinite trees – so it is natural to look at modal logic as a language to express properties of coalgebras. We will follow the development of coalgebraic semantics for modal logic in a roughly historical order, and see how this coalgebraic approach affords us a unifying view of notions such as behavioural equivalence, bisimulation, and expressivity of logics. We will not really touch deduction systems, axiomatizations, or algebraic theories of modal logic; while these topics have their place in discussions of coalgebras and modal logic, we focus on the semantic aspects.

## 2 A general semantic approach to modal logic

We begin by describing a general approach to providing the semantics of a modal logic in a standard method over relational structures. Having outlined this approach, we will be able to contrast this approach with the coalgebraic methods we introduce later in the paper. Beyond that, it's nice to be precise about what it is we are generalizing in the first place.

**Definition 2.1.** A *(modal) similarity type* $\tau$ is a collection of modal operators, along with a function $ar : \tau \to \omega$ mapping an operator to its arity. While technically a similarity type is a tuple containing both the set of operators and the arity function, it is convenient to simply consider it as the set of operators and leave $ar$ implicit, and write $\heartsuit \in \tau$ for an operator $\heartsuit$.

Now, let $AP$ be a countable set of atomic propositions.

**Definition 2.2.** Given a modal similarity type $\tau$ and a set of propositions $AP$, we can inductively define the set $\mathcal{F}_\tau(AP)$ of modal $\tau$-formulas over $AP$ as follows:

$$\mathcal{F}_\tau(AP) \ni \varphi, \psi ::= p \in AP \mid \top \mid \neg\varphi \mid \varphi \vee \psi \mid \heartsuit(\varphi_1, \dots, \varphi_n) \qquad \heartsuit \in \tau, n = ar(\heartsuit)$$

As usual, we define $\varphi \to \psi$ and $\varphi \leftrightarrow \psi$ as abbreviations, $\bot = \neg\top, \varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$ and for each $\heartsuit \in \tau$ of arity $n$ we can define the dual operator $\triangle(\varphi_1, \dots, \varphi_n) = \neg\heartsuit(\neg\varphi_1, \dots, \neg\varphi_n)$. If the particular set of atomic propositions is not important, we'll merely write $\mathcal{F}_\tau$.

**Definition 2.3.** Given a similarity type $\tau$, a $\tau$*-frame* $\mathbb{F}$ is a tuple $(S, R)$ where $S$ is a set of states and $R$ is a collection of relations $R_\heartsuit$ indexed by $\heartsuit \in \tau$, where $R_\heartsuit \subseteq S^{ar(\heartsuit)+1}$.

Note the connection to Kripke's possible worlds. We almost have enough information here to define the truth of a formula in $\mathcal{F}_\tau$. The remaining piece of the puzzle is:

[1]Well, in categories where it makes sense to discuss the "state" of an object, at least. In this paper we will be focusing on the category $\mathsf{Set}$.

**Definition 2.4.** Given a $\tau$-frame $\mathbb{F}$, a *valuation* is a function $V : AP \to \mathcal{P}(S)$ mapping an atomic proposition to the states in which it holds. A $\tau$*-model* $\mathbb{M}$ is a triple $(S, R, V)$, where $(S, R)$ is a $\tau$-frame and $V$ is a valuation on $S$.

Intuitively, we think of the atomic propositions as the observations we can make about states. From a logical perspective (as well as in applications), we cannot in general tell which particular state we are in – we can only tell which propositions hold at the state we are in. As we will see, this point of view will be reflected quite well in the coalgebraic setting.

Now we are set to define what it means for a formula to be true, or satisfied. Let $[\![\varphi]\!]_{\mathbb{M}}$ denote the set of states at which $\varphi$ is true. We define $[\![\varphi]\!]_{\mathbb{M}}$ inductively on the structure of $\varphi$:

$$[\![p]\!]_{\mathbb{M}} = V(p) \qquad\qquad [\![\varphi \vee \psi]\!]_{\mathbb{M}} = [\![\varphi]\!]_{\mathbb{M}} \cup [\![\psi]\!]_{\mathbb{M}}$$
$$[\![\top]\!]_{\mathbb{M}} = S \qquad\qquad [\![\neg\varphi]\!]_{\mathbb{M}} = S \setminus [\![\varphi]\!]_{\mathbb{M}}$$
$$[\![\heartsuit(\varphi_1, \ldots, \varphi_n)]\!]_{\mathbb{M}} = \{s \in S \mid \forall i.\exists s_i \in [\![\varphi_i]\!]_{\mathbb{M}}.R_{\heartsuit}(s, s_1, \ldots, s_n)\}$$

We write $\mathbb{M}, s \models \varphi$ as another way of saying $s \in [\![\varphi]\!]_{\mathbb{M}}$, and $\mathbb{M} \models \varphi$ if $[\![\varphi]\!]_{\mathbb{M}} = S$.

Of course, the study of modal logic includes natural questions such as "which formulas hold regardless of valuation?", or the stronger "which formulas hold in all frames?" – but the question with which we will be concerned with is "when do two states (or models) share a set of satisfied formulas?". Indeed, the "expressivity" of a logic is a function of its power to distinguish different properties of its models. For example, an important measure of a logic's expressivity is whether or not it has the *Hennessey-Milner property* – that is, the ability to distinguish non-bisimilar states.

Bisimilarity is an important concept in the study of the equivalence of systems. Roughly, two states $s, s'$ are bisimilar if we can't tell them apart based on observations, and for every set of states related to $s$ there is a bisimilar set of states related to $s'$. Formally, given two $\tau$-models $\mathbb{M}, \mathbb{M}'$ over the same set of atomic propositions $AP$, a bisimulation relation $E \subseteq S \times S'$ is a nonempty relation such that if $sEs'$,

1. $s \in V(p) \iff s' \in V'(p) \quad \forall p \in AP$
2. $R_{\heartsuit}(s, s_1, \ldots, s_n) \implies \forall i.\exists s'_i$ s.t. $s_i E s'_i$ and $R_{\heartsuit}(s', s'_1, \ldots, s'_n)$
3. $R_{\heartsuit}(s', s'_1, \ldots, s'_n) \implies \forall i.\exists s_i$ s.t. $s_i E i'$ and $R_{\heartsuit}(s, s_1, \ldots, s_n)$

As we will see, bisimilarity will play an important role as well when we view modal logic through the coalgebraic lens.

## 3 Coalgebras

Recall that a $\tau$-frame is a pair $(S, R)$ of states and relations on them. For example, if we look at $\tau = \{\Box\}$, then $R$ is just $R_\Box$, a binary relation on states. Let $R[s] = \{s' \in S \mid sRs'\}$; we can think of it as a function $R[\cdot] : S \to \mathcal{P}(S)$. The pair $(S, R[\cdot])$ is an example of a *coalgebra*.

**Definition 3.1.** Given an endofunctor $T : \mathcal{C} \to \mathcal{C}$, an $T$-*coalgebra* consists of an object $X \in \mathcal{C}$, and a morphism $\alpha : X \to FX$. We call the object the *carrier*, and the morphism the *structure map*. Occasionally we will refer to a coalgebra by its carrier set, if the structure map is unimportant in context.

So in particular, $(S, R[\cdot])$ is a $\mathcal{P}$-coalgebra, where $\mathcal{P}$ is the covariant powerset functor. If $h : X \to Y$ is a morphism in $\mathcal{C}$, where $(Y, \beta : Y \to TY)$ is another $T$-coalgebra, then $h$ is an $T$-coalgebra homomorphism if $\beta \circ h = Th \circ \alpha$. For a given endofunctor $T$, the collection of $T$-coalgebras along with their homomorphisms form a category $\mathsf{coalg}(T)$ of their own.

As our example above shows, relational structures make excellent examples of $T$-coalgebras for endofunctors on $\mathsf{Set}$. We can model a number of types of transition systems with suitable functors. For example, a labelled transition system consists of states $S$ and a relation $S \times A \times S$ for some label set $A$. This is equivalently represented by a function $S \to \mathcal{P}(S)^A$ – given a state, you get a function from a label to the set of successor states. So these transition systems make up $T$-coalgebras for the endofunctor $T = (-)^A \circ \mathcal{P}$.

In our case, the functors we will care about will often be Kripke polynomial functors, which can be defined inductively by $T, T' ::= \mathsf{id} \mid C \mid \mathcal{P} \mid T \circ T' \mid T + T' \mid T \times T' \mid T^D$, where $C$ is a constant functor and $D$ some set.

As we can model transition systems with coalgebras, it is natural to want to lift the concept of bisimulation to the categorical setting. There are a few ways to go about this. One particularly strong generalization is the following: given a set endofunctor $T$, let $\mathbb{S} = (S, \gamma)$ and $\mathbb{S}' = (S', \gamma')$ be two $T$-coalgebras. A relation $B \subseteq S \times S'$ is called a *bisimulation* between $\mathbb{S}$ and $\mathbb{S}'$ if we can endow it with a structure map $\beta : B \to TB$ such that the associated projections $\pi_1 : B \to S$ and $\pi_2 : B \to S'$ are coalgebra homomorphisms. Two states related by $B$ are said to be bisimilar.

A more intuitive, slightly weaker notion which captures similar ideas is the property of behavioural equivalence. Once again using $T$-coalgebras $\mathbb{S}, \mathbb{S}'$, we say $s \in S$ and $s' \in S'$ are behaviourally equivalent if there is a $T$-coalgebra $(X, \xi)$ and coalgebra homomorphisms $f : \mathbb{S} \to X$, $f' : \mathbb{S}' \to X$ such that $f(s) = f'(s')$. Bisimulation is a stronger condition; bisimilar states are behaviourally equivalent.

Finally[2], if a $T$-coalgebra $(Z, \zeta)$ is such that for any other $T$-coalgeba, there is a unique homomorphism into $Z$, then $(Z, \zeta)$ is the *final $T$-coalgebra*. Up to isomorphism, this $Z$ is unique. The existence of a final coalgebra is not guaranteed, when they do exist they are very useful. For example, in the presence of a final coalgebra $Z$, two states in coalgebras $S, S'$ are behaviourally equivalent $\iff$ the respective unique maps into $Z$ take the states to the same element of $Z$.

## 4 A first approach

Since coalgebras neatly encompass a general notion of evolving systems and observations, it seems natural to seek a coalgebraic approach to modal logic. The first step in this direction was an influential paper by Lawrence Moss, simply titled *Coalgebraic Logic*. Moss gives a method to obtain modal logics parametric in a set functor $T$, but these logics are somewhat unfamiliar. The logic can't be specified by a similarity type; you don't get to specify its operators. Instead, a logic obtained in Moss' manner has a single operator[3] $\nabla$ called the cover modality, whose semantics are derived in a preordained fashion straight from the functor $T$.

In the original approach, Moss defined an infinitary modal logic $\mathcal{L}_T$ given an endofunctor $T : \mathsf{SET} \to \mathsf{SET}$[4]. The class of formulas, instead of being induced from a modal similarity type, is defined to be the least class $\mathcal{L}_T$ such that $\mathcal{L}_T = \mathcal{P}\mathcal{L}_T + T\mathcal{L}_T$. For this to exist, $T$ must be monotone and *set-based*, however the details of this are unimportant for our discussion.

The associated left and right injections are, respectively, $\bigwedge : \mathcal{P}\mathcal{L}_T \to \mathcal{L}_T$ and $\nabla : T\mathcal{L}_T \to \mathcal{L}_T$, where the former is interpreted as infinitary conjunction and the latter introduces formulas whose structure depends on $T$. This is in important characteristic of Moss' logic: instead of introducing operators and manually specifying semantics tying the behaviour of the operator to the structure of the system captured by $T$, the structure of the system is used directly to generate a new modality. This contrasts with another approach we will outline later.

The coalgebraic aspect of the logic is the semantic interpretation of the formulas in $\mathcal{L}_T$. The semantics are defined in a somewhat opaque way; instead of presenting them, we will later look at a finitary version of this logic, and work with that instead. However, we began by presenting Moss' infinitary logic for the purposes of historical perspective. In the theory of infinitary modal logic, an important characterization theorem says that two models satisfy

[2] Hah!

[3] Moss used $\Delta$, but keeping with more modern treatments of Moss' logic we flip the $\Delta$ on its head.

[4] Where $\mathsf{SET}$ is the category of classes and class functions $f : C \to D$ s.t. $f(C) = \bigcup f(c)$, where the union is taken over all subsets $c$ of the class $C$. Such a function is called *set-continuous*

the same infinitary formulas $\iff$ the models are bisimilar. The models have a clearly coalgebraic structure, as a pair $(A, f : A \to \mathcal{P}(A) \times \mathcal{P}(AP))$, and Moss sought to generalize this sort of characterization result to other structures, generating logics which were strong enough to distinguish bisimilar models whose structure is specified by a functor $T$.

So Moss took infinitary modal logic (which has the usual operators $\Box, \Diamond$ and is interpreted over the usual Kripke structures) and generalized it to an infinitary logic with an operator $\nabla$ interpreted over structures described by $T$. However, the interpretation of $\nabla$ is not exactly intuitive. Beyond that, providing semantics for formulas in $\mathcal{L}_T$ required that $T$ preserve weak pullbacks, which is a somewhat restrictive condition. As a result, further forays into coalgebraic semantics for modal logic took a different tack.

## 5 A more intuitive approach

After Moss took the first steps, other researchers such as Alexander Kurz and Dirk Pattinson approached the problem of providing semantics over coalgebras from another angle. Instead of generating a logic parametric only in the functor $T$, and providing semantics in a manner which required $T$ to satisfy a somewhat restrictive set of conditions, this approach, called predicate lifting, hewed more closely to the standard method of providing semantics we discussed in Section 1.

In this approach, we are allowed to specify the particular operators we would like our modal logic to use – as before, we specify the with a similarity type $\tau$. Once again, we work with respect to a endofunctor $T$; in this case we will consider such endofunctors on $\mathsf{Set}$. Since we provide the operators, naturally we need to provide their interpretation in some way; to do this, we introduce a $\tau$-*structure*:

**Definition 5.1.** Given a modal similarity type $\tau$, a $\tau$-structure is a pair $(T, \{[\![\heartsuit]\!]\}_{\heartsuit \in \tau})$, of $T : \mathsf{Set} \to \mathsf{Set}$ and an "$n$-ary predicate lifting" $[\![\heartsuit]\!]_{(-)} : (2^{(-)})^n \to 2^{(-)} \circ T$, a natural transformation, for each $\heartsuit \in \tau$.

Note that throughout this paper we use $\mathcal{P}$ for the covariant powerset functor and $2^{(-)}$ for the contravariant one.

Recall that a similarity type $\tau$ induces a set of formulas $\mathcal{F}_\tau(AP)$ over a denumerable set of atomic propositions $AP$. Here, the syntax of the logic has no relation to the $\tau$-structure; in Moss' logic, the syntax was intimately tied to the particular functor $T$. Whether or not this is desirable depends on one's point of view – on one hand, we are free to use the formulas with which we are familiar, but on the other, we must specify a predicate lifting for each operator we introduce. We don't get semantics for free.

Whereas before, we defined models with respect to $\tau$, here we define them with respect to $T$:

**Definition 5.2.** A *T-model* $\mathbb{M} = (S, \gamma, V)$ is a triple such that $(S, \gamma) \in \mathsf{Coalg}(T)$ and $V : AP \to \mathcal{P}(S)$.

Given a $\tau$-structure and a $T$-model $\mathbb{M}$, we are set to define the semantics of formulas in $\mathcal{F}_\tau$; compare this to the situation in Section 1. For the usual Boolean operations, the semantics are the same as in Section 1. However, the situation changes for the operators we introduced:

$$[\![\heartsuit(\varphi_1, \ldots, \varphi_n)]\!]_{\mathbb{M}} = \gamma^{-1} \circ [\![\heartsuit]\!]_S([\![\varphi_1]\!]_{\mathbb{M}}, \ldots, [\![\varphi_n]\!]_{\mathbb{M}})$$

Roughly, $\mathbb{M}, s \models \heartsuit(\varphi_1, \ldots, \varphi_n) \iff \gamma(w)$ satisfies the property defining $\heartsuit$; the idea which remains fixed in this generalization is that the operator specifies properties of "structured successors". Of course, the question is how to define the predicate lifting. To get a handle on how they should be defined, it's easiest to look at specific examples. The modal logic **K**, i.e. the weakest normal modal logic, interpreted over Kripke frames, is simple and hence a good first example. Here, $\tau = \{\Box\}$, and the $T$ we are interested in is simply the covariant powerset functor $\mathcal{P}$. For a set of states $S$, what should $[\![\Box]\!]_S$ be?

The formula $\Box\varphi$ will be interpreted to hold true for all states $s$ such that $\gamma(s)$ is in $[\![\Box]\!]_S([\![\varphi]\!]_{\mathbb{M}})$. We want it to hold for $s$ such that $\gamma(s)$ all satisfy $\varphi$; i.e. we want $\gamma(s) \subseteq [\![\varphi]\!]_{\mathbb{M}}$. Hence we define $[\![\Box]\!]_S(Z) = \{Y \subseteq S \mid Y \subseteq Z\}$. We can check that this has the correct type as specified in definition 5.1: indeed, $[\![\Box]\!]_S$ maps a set of states ($2^S$) to a set of sets of states ($2^{\mathcal{P}(S)}$).

Another, more interesting example is defining Hennessy-Milner logic in this setting. Here, $\tau = \{[a] \mid a \in A\}$ for some set of actions $A$, and we interpret the formulas over labelled transition systems which can be described (as we mentioned earlier) with the set functor $T = (-)^A \circ \mathcal{P}$. Through a similar process of reasoning we can come to the conclusion that $[\![[a]]\!]_S(Z) = \{f : A \to \mathcal{P}S \mid f(a) \subseteq Z\}$.

As we mentioned earlier, Moss' approach included the limitation that the functor specifying the type of transition system must preserve weak pullbacks. We would be remiss if we mentioned that the predicate lifting approach doesn't have this restriction without giving an example of a transition system type we can treat with this approach but not Moss'; otherwise one might conclude that the weak pullback-preservation isn't much of a restriction after all.

Neighbourhood frames are another structure for interpreting standard modal logic; instead of looking at relational structures $(W, R)$ like we described in the introduction, the relation between worlds is captured by a function $N : W \to 2^{2^W}$, where the subsets of $W$ in $N(w)$ represent sets of worlds obeying some proposition. As an example, in this framework one would say $\mathbb{M}, w \models \Box\varphi \iff [\![\varphi]\!]_{\mathbb{M}} \in N(w)$. Neighbourhood frames were studied independently by both Dana Scott and Richard Montague. In this case, the functor describing the structure is $T = 2^{(-)} \circ 2^{(-)}$; this does not preserve weak pullbacks.

## 6 Return to the cover modality

Following our historical treatment of the development of coalgebraic modal logics, we return to the cover modality approach initiated by Moss. As the predicate lifting approach allowed one to provide coalgebraic semantics for a logic whose syntax was supplied, rather than generated, and supported a wider variety of functors $T$, attention was directed away from Moss' approach. However, results outside of the coalgebraic program would help turn some of the spotlight back onto Moss' approach.

The paper *Automata for the modal $\mu$-calculus and related results* by David Janin and Igor Walukiewicz reconstructed classical modal languages using a connective $a \to \Phi$, where $\Phi$ is a set of formulas and $a \in A$, an action set; this reconstruction was an important proof technique in showing some nice results about the $\mu$-calculus. The semantics of this connective were identical to the cover modality in such a setting; noting this connection, Yde Venema and Kupke showed that many important results in the theory of fixpoint logics could be lifted to the generalized coalgebraic setting. In the following years, more focus was devoted to Moss' approach.

Now we will describe a finitary version of the logic $\mathcal{L}_T$ described before; this presentation is due to Kupke and Pattinson. Let $T : \mathsf{Set} \to \mathsf{Set}$ be a standard[5] functor, and let $T_\omega$ denote the finitary part of $T$; $T_\omega X = \bigcup TY$, where the union is taken over all finite subsets $Y \subseteq X$ (we denote this $Y \subseteq_\omega X$). We define $\mathcal{L}_T$ as the smallest set of formulas closed under the following rules:

$$\frac{}{\top \in \mathcal{L}_T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}_T}{\bigwedge \Phi \in \mathcal{L}_T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}_T}{\bigvee \Phi \in \mathcal{L}_T} \quad \frac{\varphi \in \mathcal{L}_T}{\neg\varphi \in \mathcal{L}_T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}_T \quad \alpha \in T_\omega \Phi}{\nabla\alpha \in \mathcal{L}_T}$$

In order to define semantics for this logic, we need to introduce the concept of *relational lifting*. Given our set endofunctor $T$, and $R \subseteq X_1 \times X_2$, the *(T-)lifted* relation $\overline{T}R \subseteq TX_1 \times TX_2$ is the set

$$\overline{T}R = \{(t_1, t_2) \,|\, \exists z \in TR.T\pi_i(z) = t_i \text{ for } i = 1, 2\},$$

It is a theorem that if $T$ preserves weak pullbacks, then $\overline{T}(R \circ S) = \overline{T}R \circ \overline{T}S$, where $R \circ S$ denotes relation composition. This is why we imposed such a requirement on our $T$. While in general it may be nontrivial to understand the lifting of some relation, it is easy to compute for Kripke polynomial functors.

Now, to provide the semantics of $\varphi \in \mathcal{L}_T$, we fix a $T$-coalgebra $(S, \gamma)$ and define the satisfaction relation $\models \subseteq W \times \mathcal{L}_T$ by induction in the obvious way for conjunctions, disjunctions,

[5] Maps set-theoretic inclusions to inclusions.

$\top$, and negations. The interesting case is the semantics of the cover modality[6]:

$$s \models \nabla\alpha \text{ if } (\gamma(s), \alpha) \in \overline{T}(\models)$$

As in the case of predicate liftings, we can gain a better understanding of what's going by looking at an example. In the case of a Kripke structure, the relevant functor is $T = \mathcal{P}$. Hence we get modal formulas $\nabla\{\varphi_1, \ldots, \varphi_n\}$, since $\nabla\alpha$ is a formulas for $\alpha \in \mathcal{P}_\omega\Phi$ for some finite subset $\Phi \subseteq \mathcal{L}_{\mathcal{P}}$. If we chase through the definitions, we see that $s \models \nabla\{\varphi_1, \ldots, \varphi_n\}$ iff

$$\forall i \,.\, \exists s' \in \gamma(s) \text{ s.t. } s' \models \varphi_i \quad \text{and} \quad \forall s' \in \gamma(s) \,.\, \exists i \text{ s.t. } s' \models \varphi_i$$

Or, using the standard $\Box, \Diamond$ semantics, $s \models \nabla\{\varphi_i\}$ if $s \models \bigwedge_i \Diamond\varphi_i \wedge \Box \bigvee_i \varphi_i$, for some finite set of formulas $\{\varphi_i\}$. In particular, we can recover the usual operators for modal logic over Kripke structures using the cover modality (this is the property Janin and Walukiewicz used, for their connective).

## 7 Expressivity

As we've mentioned before, the ability for logics to distinguish bisimilar states is an important landmark on the scale of expressivity. In the coalgebraic setting we've introduced the notions of bisimilarity and behavioural equivalence as appropriate generalizations. Since behavioural equivalence is a weaker but still sufficient characterization of our usual idea of bisimilarity, we'll use it as our line in the sand – we will say a logic is expressive if two non-similar states can be distinguished by a formula. Let's look at how the approaches of predicate lifting and relational lifting allow us to test for expressivity.

As we mentioned when discussing the origin of Moss' (relational lifting) approach, it was an attempt to generalize expressivity properties of infinitary modal logics to structures described by a suitable functor $T$. As long as $T$ is finitary, then the $\nabla$ logic we recover it via relational lifting is expressive[7]. For weak pullback-preserving functors, behavioural equivalence can be characterized by the lifting $\overline{T}$, so any $\nabla$ formula's truth value is the same for behaviourally equivalent states.

On the other hand, if $T$ is finitary, then $\mathsf{Coalg}(T)$ contains a final coalgebra $(Z, \zeta)$ whose structure we can understand – in a technique which seems indispensable for theoretical computer science, we can generate the carrier $Z$ as a greatest fixed point with iterates $T^n 1$,

[6]Technically this should be treated in levels by defining the modal depth of a formula, and for $\nabla\alpha$ of modal depth $n+1$, lifting the restriction of the $\models$ relation to formulas of modal depth at most $n$, so the definition is well-founded. However, if $T$ is standard and preserves weak pullbacks, the definition will be equivalent. For more details, see the references.

[7]In fact, for weak pullback-preserving functors, behavioural equivalence and bisimilarity coincide.

for $n < \omega$. Then any state's is characterized by some finite process; we can inductively define formulas to characterize states by this $n$-step behaviour.

The situation is different for logics defined via predicate liftings. We do not get such an elegant characterization; instead, since we defined the semantics, we must deal with the fact that our definitions may lead to a weak logic. While it is true for any $\tau$-structure that behavioural equivalence will lead to the two states satisfying all the same formulas, the other direction is more stringent. In particular, $T$ must be finitary, and the map

$$TS \to \prod_{\heartsuit \in \tau \ n\text{-ary}} \mathcal{P}(\mathcal{P}(X)^n)$$

defined on each factor by sending $t \mapsto \{(A_1, \ldots, A_n) \subseteq X^n \mid t \in [\![\heartsuit]\!](A_1, \ldots, A_n)\}$ must be injective. It isn't too hard to see why, at least intuitively – we need to be able to distinguish structured successors by our operators, so in a sense there must be "enough" operators to tell them apart. If the map defined above is injective, then we say the $\tau$-structure is *separating.*

To summarize, in both cases we need $T$ to be finitary in order to distinguish states which aren't behaviourally equivalent. However, this is a sufficient condition as well for logics defined via relational lifting (Moss' approach). If the logic is defined via predicate lifting, then the associated $\tau$-structure must also be separating.

# 8 Conclusion

Coalgebras give us a way to study the properties of many different modal logics in a unified framework. By providing a uniform template into which we can mould a variety of models for modal logics, we can derive conditions on expressivity which apply to each logic we place in this framework. This gives us a reliable process to follow in order to prove properties of a new logic – if we can describe it in the coalgebraic framework, a number of theorems automatically apply.

Beyond that, it's pleasing to see these ideas in the right place. By fitting pieces of the theory of modal logic into a grander framework, we are less likely to miss the forest for the trees. Certainly there are areas in the study of modal logic which require a more involved and particular approach, but by isolating the aspects which can be solved by more general means we get a better view of the surrounding territory.

## References


[BdRV02] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2002.

[BV11] Jort Bergfeld and Yde Venema. Model constructions for moss coalgebraic logic. In Andrea Corradini, Bartek Klin, and Corina Crstea, editors, *Algebra and Coalgebra in Computer Science*, volume 6859 of *Lecture Notes in Computer Science*, pages 100–114. Springer Berlin Heidelberg, 2011.

[JR11] Bart Jacobs and JJMM Rutten. An introduction to (co) algebra and (co) induction. 2011.

[JW95] David Janin and Igor Walukiewicz. Automata for the modal ?-calculus and related results. In Ji? Wiedermann and Petr Hjek, editors, *Mathematical Foundations of Computer Science 1995*, volume 969 of *Lecture Notes in Computer Science*, pages 552–562. Springer Berlin Heidelberg, 1995.

[KKV12] Clemens Kupke, Alexander Kurz, and Yde Venema. Completeness for the coalgebraic cover modality. *Logical Methods in Computer Science*, 8(3), 2012.

[KP11] Clemens Kupke and Dirk Pattinson. Coalgebraic semantics of modal logics: An overview. *Theoretical Computer Science*, 412(38):5070 – 5094, 2011. {CMCS} Tenth Anniversary Meeting.

[Mos99] Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(13):277 – 317, 1999.

[Rut00] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3 – 80, 2000. Modern Algebra.

[Ven07] Yde Venema. 6 algebras and coalgebras. In Johan Van Benthem Patrick Blackburn and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 331 – 426. Elsevier, 2007.