

Coalgebraic Semantics of Modal Logics: an Overview

Clemens Kupke and Dirk Pattinson

Department of Computing, Imperial College London

180 Queen's Gate, London SW7 2AZ, UK

c.kupke@imperial.ac.uk, d.pattinson@imperial.ac.uk

October 7, 2011

Abstract

Coalgebras can be seen as a natural abstraction of Kripke frames. In the same sense, coalgebraic logics are generalised modal logics. In this paper, we give an overview of the basic tools, techniques and results that connect coalgebras and modal logic. We argue that coalgebras unify the semantics of a large range of different modal logics (such as probabilistic, graded, relational, conditional) and discuss unifying approaches to reasoning at this level of generality. We review languages defined in terms of the so-called cover modality, languages induced by predicate liftings as well as their common categorical abstraction, and present (abstract) results on completeness, expressiveness and complexity in these settings, both for basic languages as well as a number of extensions, such as hybrid languages and fixpoints.

1 Introduction

Many of the logics that are used in computer science today are variations of modal logics: they offer a good compromise between expressive power on the one hand, and decidability in moderately low complexity classes on the other hand. Starting from the modal logic K [13], a decidable fragment of first order logic, many variations of modal logics have been engineered to fit the need of specific applications, usually retaining the good decidability properties [101]. Various extensions of modal logics are used for a variety of different purposes, such as hybrid logic [5], description logics [6] and the modal μ -calculus [17]. These extensions add more reasoning power to basic modal logic by allowing for ontological reasoning and for reasoning about ongoing, possibly infinite behaviour. Many modal languages have been designed to fit specific semantic domains such as modal logics for Markov chains [44, 30], game frames [79] or neighbourhood structures [95, 68]. In this paper, we take the term “modal language” to be broadly construed: an extension of the language of (classical) propositional logic with additional operators.

It is the aim of coalgebraic modal logic to create a general framework for modal logics to reason about a variety of semantic domains in a uniform way. This framework not only covers structurally different classes of models, but also a wide variation of reasoning principles. Basic coalgebraic logics can be extended with features such as fixpoints and nominals to allow for applications in the area of reactive systems and knowledge representation.

A wide range of semantic domains that are usually studied individually in the literature have been demonstrated to fit into the framework of coalgebra. Coalgebras over the category of sets provide a general framework for studying state-based transition systems [83] such as labelled transition systems, discrete Markov chains and neighbourhood structures. The concrete shape of a system is specified by a given set functor T , that we think of as the type of *transitions*. These transitions can take a variety of shapes, such as labelled or unlabelled, probabilistic, weighted or alternating and can also be used to represent moves in strategic games and conditional truth. Coalgebraic modal logic tries to develop the theory of modal logic *parametric* in the type of transition, ie., parametric in the functor T that represents the transition type.

In our overview paper we focus on two related approaches to modal logics in a coalgebraic setting and discuss their common, categorical abstraction. The first approach is based on so-called *predicate liftings*, the second approach on *relation lifting*. The predicate lifting approach [75] can be seen as a direct generalisation of basic modal logic. If we think of a formula A as describing a property of states or worlds, then $\Box A$ describes a property of successor states or alternative worlds. In other words, the operation \Box lifts a state predicate to a successor predicate, and a state satisfies the lifted predicate $\Box A$ if its successor (transition) lies in the extension of $\Box A$. This approach is not fully parametric in the transition type (represented by a set functor). In order to obtain a language for a given transition type, one needs to choose a set of operators on the syntactic side and specify, for each of the operators, in what way it lifts state properties to transition properties.

The relation lifting approach appeared in the first paper on coalgebraic logic [70]. Here, we take the transition type T as defining both syntax and semantics of the modal language. In this way the language is a perfect match for the transition type under scrutiny. Compared to the predicate lifting approach, this covers a slightly smaller class of systems (transition functors need to preserve so-called weak pullbacks) and induces a somewhat non-standard syntax. However, both approaches can be unified in an abstract, categorical framework as we shall discuss.

Structure of the paper. In Section 2 we list some examples of frame classes that are amenable to the coalgebraic treatment. In Section 3, we introduce the syntax and semantics of the two families of “basic” coalgebraic modal logics that we present. Section 4 contains an overview of expressivity results for coalgebraic logics. In Section 5 we discuss reasoning and decision techniques for coalgebraic logics and coalgebraic description logics. In particular, we discuss complete derivation systems and tableau systems for these logics. Finally, in Section 6 we give an overview of coalgebraic fixpoint logics.

The notation we use in this paper is largely standard. For the most part, we restrict attention to coalgebras on sets and write $\text{Coalg}(T)$ for the category of coalgebras induced by a set-functor T . Explicitly, a T -coalgebra is a pair (W, γ) where W is a set (of worlds) and $\gamma : W \rightarrow TW$ is a (transition) function, and a *coalgebra morphism* $f : (W, \gamma) \rightarrow (W', \gamma')$ is a set theoretic function $f : W \rightarrow W'$ that satisfies $Tf \circ \gamma = \gamma' \circ f$.

2 Examples

The fact that the coalgebraic semantics applies to a large class of structurally different model classes is achieved through the parametricity of the overall theory in a given endofunctor T of type $\text{Set} \rightarrow \text{Set}$ that defines the class $\text{Coalg}(T)$ of T -coalgebras that we think of as frames in this context. Clearly, different choices of T yield different classes of models, and we begin our survey of coalgebraic logics by giving examples of modal logics, and associated semantical structures that are covered by the coalgebraic approach.

Kripke Frames & Labelled Transition Systems. Kripke frames, or the relational semantics of modal logic first put forward by Kripke [53] clearly constitute the most well-studied semantics of modal logic. We can capture Kripke frames in the coalgebraic framework by re-formulating the classical textbook definition [37, 13] where $\mathcal{P} : \text{Set} \rightarrow \text{Set}$ is the covariant powerset functor: A *Kripke-Frame* is a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{P}(W)$ is a function. It is immediate that this definition is equivalent to the classical definition of Kripke frames.

To capture labelled transition systems in the coalgebraic framework, we consider the functor $\mathcal{P}(\cdot)^{\mathcal{A}}$ where \mathcal{A} is a set (of actions, or labels) and $\mathcal{P}(X)^{\mathcal{A}}$ is the set of all functions of type $\mathcal{A} \rightarrow \mathcal{P}(X)$: A *labelled transition system* is a pair (W, γ) where $\gamma : W \rightarrow \mathcal{P}(W)^{\mathcal{A}}$ is a function. This is again equivalent to the standard definition given e.g. in [46], where a labelled transition system is understood as tuple (W, R) where W is the set of states and $R \subseteq W \times \mathcal{A} \times W$ is a labelled transition relation.

Neighbourhood Frames. Neighbourhood frames were studied by Scott [95] and Montague [68] as alternative models of modal logic. They can be captured in the coalgebraic framework by means of the functor $\mathcal{N}X = 2^{2^X}$, technically the composition of the *contravariant* powerset functor 2^- with itself. In other words, the action of \mathcal{N} on maps is given by $\mathcal{N}(f) = (f^{-1})^{-1}$ where $g^{-1} : \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$ denotes the inverse image operation induced by a function $g : X \rightarrow Y$. A *neighbourhood frame* is a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{N}W$. The induced class of \mathcal{N} -coalgebras is eminently identical to the class of neighbourhood frames, studied from a logical perspective in [21] and in [41] from a coalgebraic angle. One commonly considered variation on neighbourhoods is that the set of all neighbourhoods of a point is to be upwards closed, thus inducing monotone neighbourhood frames. In the coalgebraic framework, the class of monotone neighbourhood frames arises as a class of coalgebras for a sub-functor of \mathcal{N} . A system $S \subseteq \mathcal{P}(X)$ of subsets of a set X is *upward closed* if $A \in S$ whenever $B \in S$ and $A \supseteq B$. Let $\mathcal{M}X = \{S \in \mathcal{N}X \mid S \text{ upward closed}\}$. A *monotone neighbourhood frame* then arises in the coalgebraic framework as a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{M}W$.

Probabilistic Frames. For a function $f : X \rightarrow \mathbb{R}$ we write $\text{supp}(f) = \{x \in X \mid f(x) \neq 0\}$ for the *support* of X and let $\mathcal{D}X = \{\mu : X \rightarrow [0, 1] \mid \text{supp}(\mu) \text{ finite}, \sum_{x \in X} \mu(x) = 1\}$ be the set of finitely supported probability distributions on X . A *probabilistic frame* is a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{D}W$. Clearly every probabilistic frame defines a discrete time Markov chain with transition probabilities given by the local probability distributions. We only consider *finitely supported* probability distributions as this enables us to remain within the category of sets and functions, and we refer the reader to Moss and Viglizzio [69] for a more general treatment in the context of measurable spaces. Probabilistic frames, in the above form, are the basic ingredient in a large variety of probabilistic systems [9] but are not so well studied from a logical perspective. They are closely related to Harsanyi's type spaces (see Heifetz and Mongin [44] for a treatment in the context of modal logic) and to Fagin et. al.'s Kripke structures for knowledge and probability [30]. In essence, a probabilistic frame is the same as a finitely branching Kripke structure where successor states are assigned probabilities whereas the models considered in both [44, 30] are based on measurable spaces and therefore provide larger model classes. The point to note is that they are of the form $(W, \gamma : W \rightarrow \mathcal{D}W)$ and hence provide us with yet another example of coalgebras.

Multigraph Frames. For a set X let $\mathcal{B}X = \{f : X \rightarrow \mathbb{N} \mid \text{supp}(f) \text{ is finite}\}$ denote the collection of finitely supported multisets (or bags) over X . A *multigraph frame* is a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{B}W$, ie., a \mathcal{B} -coalgebra. In other words, each world $w \in W$ of the carrier of a multigraph frame has a (finite) number of successors, each of which comes with a weight or multiplicity. Multigraph frames [28] can be used to provide semantics for graded modal logic [31]. Multigraphs are a generalisation of Kripke frames where every edge is assigned a (non-negative)

integer weight. As we will see later, the standard semantics of graded modal logic is not coalgebraic *per se* but can be equivalently re-formulated.

Game Frames. Let N be a set (of “agents” or “players”). If X is a set, we put

$$\mathcal{G}X = \{((S_n)_{n \in N}, f) \mid \forall n \in N (\emptyset \neq S_n \subseteq \mathbb{N}) \text{ and } f : \prod_{n \in N} S_n \rightarrow W\}.$$

A *game frame* is a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{G}W$. Game frames are the semantic domain of choice for interpreting Pauly’s Coalition Logic [79]. We think of W as the set of positions on a game board. The set S_n consists of the strategies available to agent n and $f : \prod_{n \in N} S_n \rightarrow W$ is an outcome function that produces a new position on the game board, depending on the choice of chosen strategy of each agent. By restricting the strategy sets to subsets of \mathbb{N} we implicitly assume that there are at most countably many strategies. Clearly, this requirement can be relaxed, to the point where one just requires that each S_n is a set, which would however result in a class-valued functor.

Selection Function Frames. If X is a set, write $\mathcal{C}X = \{f : \mathcal{P}(X) \rightarrow \mathcal{P}(X) \mid f \text{ a function}\}$. A *selection function frame* (or conditional frame) is a pair (W, γ) where W is a set and $\gamma : W \rightarrow \mathcal{C}W$ is a function. This definition is equivalent to the one in [21] where a selection function frame is taken to be a pair (W, γ) where W is a set of worlds and $\gamma : W \times \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ is a selection function that determines a proposition $\gamma(w, A) \subseteq W$ for each world $w \in W$ and condition $A \in \mathcal{P}(X)$. The class of selection function frames provides semantics for *conditional logic* where the first argument of the transition function is interpreted as a condition that enables a certain conclusion, thus giving rise to a binary modal operator (see [21] for details).

In summary, we think of coalgebras as functions that map states or worlds to a structured set of successor states, and variations in the structure of successor states allow us to capture a large class of mathematical structures over which modal logics can be interpreted. We discuss the logical setup in the following sections.

3 Logical Languages and Semantics

We now turn to the language of coalgebraic logic, and the interpretation of formulas with respect to coalgebraic models. We distinguish two approaches to defining languages for coalgebras. The so-called *predicate lifting* approach extends classical propositional logic with modal operators inducing a standard modal language. To retain parametricity in the semantics, we do not insist on particular operators and instead postulate *coherence conditions* that relate the operators to the abstract semantics. Consequently, an instantiation of the predicate lifting approach requires that we specify the set of modal operators (and their interpretation) that we adjoin to propositional logic, and additionally the verification of suitable coherence conditions. This is not necessary in the second approach that extends propositional logic with the so-called *cover modality* ∇ that is directly induced by the endofunctor that defines the semantics. We now describe both approaches and show that they both embed into a more abstract, categorical framework.

3.1 Coalgebraic Logics via Predicate Liftings

In the predicate lifting approach, we fix a modal similarity type Λ consisting of modal operators with arities (that we leave implicit if understood from the context). The interpretation of modal operators is an ingredient of the approach and is captured by the notion of a Λ -structure.

Definition 3.1. A *modal similarity type*, or *similarity type* for short, is a set of modal operators with arities. If Λ is a similarity type, a Λ -*structure* consists of an endofunctor $T : \text{Set} \rightarrow \text{Set}$, together with an assignment of an n -ary *predicate lifting*, that is, a natural transformation of type $\llbracket \heartsuit \rrbracket : (2^-)^n \rightarrow 2^- \circ T$ where $2^- : \text{Set} \rightarrow \text{Set}^{\text{op}}$ is the contravariant powerset functor, to every n -ary operator $\heartsuit \in \Lambda$.

Conceptually, predicate liftings specify a property of structured successors (subsets of TX) given an n -tuple of properties of states (subsets of X), parametric over the class of all sets (that we think of, in this case, as the putative carriers of frames). Intuitively, the naturality condition ensures that the operators receive the ‘same meaning’ independently of the chosen carrier set, and more formally guarantees that the modal semantics is compatible with coalgebra morphisms. Every modal similarity type Λ induces a modal language that we can interpret over T -models given that T extends to a Λ -structure as follows:

Definition 3.2. The *language* induced by a modal similarity type Λ is the set $\mathcal{F}(\Lambda)$ of formulas

$$\mathcal{F}(\Lambda) \ni A, B ::= p \mid A \wedge B \mid \neg A \mid \heartsuit(A_1, \dots, A_n) \quad (p \in \mathsf{P}, \heartsuit \in \Lambda \text{ } n\text{-ary})$$

where P is a fixed and denumerable set of propositional variables. A T -*model* is a triple $M = (W, \gamma, \pi)$ where $(W, \gamma) \in \text{Coalg}(T)$ and $\pi : \mathsf{P} \rightarrow \mathcal{P}(W)$ is a valuation. Given a Λ -structure T and a T -model $M = (W, \gamma, \pi)$, the *semantics* of $A \in \mathcal{F}(\Lambda)$ is inductively given by

$$\llbracket p \rrbracket_M = \pi(p) \quad \llbracket A \wedge B \rrbracket_M = \llbracket A \rrbracket_M \cap \llbracket B \rrbracket_M \quad \llbracket \neg A \rrbracket_M = W \setminus \llbracket A \rrbracket_M$$

which gives the standard interpretation of the propositional connectives over the Boolean algebra $\mathcal{P}(W)$, and for the modal operators we put

$$\llbracket \heartsuit(A_1, \dots, A_n) \rrbracket_{(W, \gamma, \pi)} = \gamma^{-1} \circ \llbracket \heartsuit \rrbracket_W(\llbracket A_1 \rrbracket_M, \dots, \llbracket A_n \rrbracket_M).$$

Intuitively speaking, the above definition amounts to saying that a state $w \in W$ satisfies a formula $\heartsuit(A_1, \dots, A_n)$ if the transition function γ maps it to a successor $\gamma(w)$ that satisfies the property \heartsuit that may depend on A_1, \dots, A_n . We write $M, w \models A$ if $w \in \llbracket A \rrbracket_M$ and $M \models A$ if $M, w \models A$ for all $w \in W$ and finally $\text{Mod}(T) \models A$ if $M \models A$ for all $M \in \text{Mod}(T)$, where $\text{Mod}(T)$ denotes the collection of all T -models.

Example 3.3. 1. If we take $TX = \mathcal{P}X$, we have seen that T -coalgebras are precisely Kripke frames. If we choose the similarity type $\Lambda = \{\Box\}$ we obtain the standard semantics of the modal logic K by associating \Box with the lifting $\llbracket \Box \rrbracket_X(Z) = \{Y \in \mathcal{P}X \mid Y \subseteq Z\}$.

If (W, γ, π) is a \mathcal{P} -model (a Kripke model) and $A \in \mathcal{F}(\Lambda)$ is a formula with interpretation $\llbracket A \rrbracket$, we have that

$$\llbracket \Box A \rrbracket = \gamma^{-1} \circ \llbracket \Box \rrbracket_W(\llbracket A \rrbracket) = \{w \in W \mid \gamma(w) \subseteq \llbracket A \rrbracket\}$$

so that $w \models \Box A$ iff $w' \models A$ for all $w' \in \gamma(w)$. This yields the standard Kripke semantics of modal logic [37, 13].

2. For $TX = \mathcal{P}X^{\mathcal{A}}$ we have seen previously that T -coalgebras are in one-to-one correspondence with labelled transition systems. Here, we consider the similarity type $\Lambda = \{[a] \mid a \in \mathcal{A}\}$ where each $[a]$ is a unary operator. We extend T to a Λ -structure by stipulating that

$$\llbracket [a] \rrbracket_X(Z) = \{f : \mathcal{A} \rightarrow \mathcal{P}(X) \mid f(a) \subseteq Z\}$$

The coalgebraic semantics precisely coincides with the standard semantics of Hennessy-Milner logic [46].

3. Neighbourhood frames can be seen as coalgebras for the functor $\mathcal{N}X = 2^{2^X}$. The modal logic of neighbourhood frames is induced by the similarity type $\Lambda = \{\Box\}$, and we obtain the standard semantics [21] if we interpret \Box by $\llbracket \Box \rrbracket_X(Z) = \{Y \in \mathcal{N}X \mid Z \in Y\}$.

Given a neighbourhood model $M = (W, \gamma, \pi)$ where $\gamma : W \rightarrow \mathcal{N}W$ we then obtain

$$w \models \Box A \quad \text{iff} \quad \llbracket A \rrbracket \in \gamma(w)$$

where $\llbracket A \rrbracket \subseteq W$ is the interpretation of the formula $A \in \mathcal{F}(\Lambda)$. Again this gives the standard semantics. It can be seen easily that this correspondence restricts to monotone neighbourhood frames.

4. For probabilistic frames (that is, \mathcal{D} -coalgebras) there is a large variation of modal operators that we may wish to consider. The probabilistic modal logic of Heifetz and Mongin [44] uses unary operators taken from $\Lambda = \{L_p \mid p \in [0, 1] \cap \mathbb{Q}\}$ where a formula $L_p A$ reads as ‘ A holds with probability at least p in the next state’. To capture the semantics of this logic, we use the interpretation

$$\llbracket L_p \rrbracket_X(Y) = \{\mu \in \mathcal{D}(X) \mid \mu(Y) \geq p\}$$

where we have abbreviated $\mu(Y) = \sum_{y \in Y} \mu(y)$. Given a probabilistic model (W, γ, π) where now $\gamma : W \rightarrow \mathcal{D}W$, we obtain

$$w \models L_p A \quad \text{iff} \quad \gamma(w)(\llbracket A \rrbracket) \geq p$$

which captures the semantics given in [44] in a coalgebraic setting. The logic for reasoning about probability discussed in [30] allows linear inequalities for reasoning about probabilities, and every formal rational linear inequality

$$a_1 \mu(F_1) + \dots + a_n \mu(F_n) \geq b$$

in (formula-valued) parameters F_1, \dots, F_k defines a k -ary modal operator. To express the semantics of these operators coalgebraically, we use the lifting

$$\llbracket a_1 \mu(F_1) + \dots + a_n \mu(F_n) \geq b \rrbracket_X(Y_1, \dots, Y_n) = \{\mu \in \mathcal{D}(X) \mid a_1 \mu(Y_1) + \dots + a_n \mu(Y_n) \geq b\}.$$

For finitely supported probability measures as discussed here, this semantics coincides with that given in *op.cit.*

5. From a coalgebraic perspective, the situation for graded modal logic is similar to that of probabilistic modal logic. We can take the similarity type $\Lambda = \{\langle k \rangle \mid k \in \mathbb{N}\}$ and interpret Λ -formulas via the lifting

$$\llbracket \langle k \rangle \rrbracket_X(Y) = \{f \in \mathcal{B}(X) \mid f(Y) > k\}$$

where again $f(Y) = \sum_{y \in Y} f(y)$. This definition supports the reading of $\langle k \rangle A$ of ‘ A holds in more than k successors, counted with multiplicity’. The operators $\langle k \rangle$ are known as *graded modalities* and the ensuing language of graded modal logic is usually interpreted over Kripke models [31] so that we do not recover the standard semantics directly. However, we can see image finite Kripke frames as multigraphs where every edge has multiplicity one, and transform a multigraph to a Kripke frame by creating the appropriate number of copies of each state. Similarly to probabilistic modal logic, we can also allow linear inequalities as modal operators (with an analogous definition of the associated liftings) to capture Presburger modal logic [29] or Majority logic [73].

In summary, it seems fair to say that the predicate lifting approach to coalgebraic logics subsumes a large variety of structurally different modal logics. The strength of the coalgebraic approach becomes apparent once we establish properties (such as decidability or the Hennessy-Milner property) of coalgebraic logics in the abstract framework so that we readily obtain results about concretely given logics, once they have been recognised to admit a coalgebraic semantics.

3.2 The cover modality

Logics defined in terms of predicate liftings have a general significant drawback: while the general goal of coalgebraic logic is to study a family of modal languages that is parametric in the type of transition systems, coalgebraic modal languages via predicate liftings have a second parameter: the choice of a modal similarity type Λ and the interpretation of the operators by means of a Λ -structure, where it would be desirable to have a more canonical construction of modal languages.

At first sight, it seems that there is such a canonical choice of predicate liftings: one can lift a predicate $P \xrightarrow{i} X$ to the predicate $\{t \in TX \mid \exists s \in TP \text{ s.t. } Ti(s) = t\} \subseteq TX$ (see [48, Chapter 4] where this logic is discussed in more detail). However, this resulting canonical logic has very little descriptive power (in the sense that we will discuss in Section 4 below). For example, the resulting logic for the functor $T = _ \times _$ that describes infinite binary trees disguised as T -coalgebras, fails to possess the Hennessy-Milner property. The lack of expressivity in this example can be explained intuitively by noting that the logic induced by the canonical predicate lifting is not able to express very natural, basic properties of binary trees, such as “property p holds for the left subtree”.

The problem of choosing suitable predicate liftings leads to the question whether, and how, one can define a modal language that is parametric in the choice of the functor only. This question has been answered in the ground breaking paper on coalgebraic logic by Moss [70, 71]. Moss’ idea is to simply use the functor T that encodes the type of transition system under consideration as a syntax constructor. In other words, if we are given a set of formulas $\Phi \subseteq \mathcal{L}$, then $\nabla\alpha$ for $\alpha \in T\Phi$ is also a formula, and ∇ is known as the *cover modality*. E.g. for $TX = \mathcal{D}X$ as in Section 2 this means that every finitely supported probability distribution over formulas is again a formula, whose semantics is given by matching transition probabilities.

We are going to deviate from Moss’ original definitions and notation. First, we follow [102] and denote the cover modality by ∇ rather than adhering to the original notation Δ [70, 71]. Second, again following [102], we limit our discussion to logics with finitary syntax and only allow finite disjunctions and conjunctions and use the finitary part T_ω of a given endofunctor T as syntax constructor.

For purposes of presentation, we will assume in the sequel that the functor T under consideration is standard, i.e., maps set-theoretic inclusion functions to inclusions (see [8] and [3] for background on the standardisation of set-functors). With this assumption in mind, the finitary part T_ω of a set functor is given by $T_\omega X = \bigcup_{X' \subseteq_\omega X} TX'$ for $X \in \text{Set}$ where the notation $X' \subseteq_\omega X$ means that X' is a finite subset of X . Intuitively, $T_\omega X$ contains those elements of TX that can be constructed using only finitely many elements of X . Finitary ∇ -languages now take the following form:

Definition 3.4. Let T be a set functor. The set \mathcal{L}^T of formulas of coalgebraic ∇ -logic is inductively defined as the smallest set closed under the following rules:

$$\frac{}{\top \in \mathcal{L}^T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}^T}{\bigwedge \Phi \in \mathcal{L}^T} \quad \frac{\Phi \subseteq_\omega \mathcal{L}^T}{\bigvee \Phi \in \mathcal{L}^T} \quad \frac{A \in \mathcal{L}}{\neg A \in \mathcal{L}} \quad \frac{\Phi \subseteq_\omega \mathcal{L}^T}{\nabla\alpha \in \mathcal{L}^T} \quad \alpha \in T\Phi$$

where $X \subseteq_\omega Y$ denotes that X is a finite subset of Y . The modal depth $d(A)$ of a formula is defined as usually by induction on the structure of the formula. We only mention the ∇ -case of the definition:

$$d(\nabla\alpha) = \min\{\max\{d(A) \mid A \in \Phi\} \mid \alpha \in T\Phi\} + 1.$$

Finally, we write \mathcal{L}_n^T for the collection of formulas with modal depth n .

This definition of the set of formulas ensures that each formula has a finite set of subformulas (cf. [102]). This is the justification for calling \mathcal{L}^T the *finitary* ∇ -language for T . The key for defining

the semantics of formulas in the ∇ -language is the so-called *relation lifting* associated with a given functor.

Definition 3.5. Let $T : \text{Set} \rightarrow \text{Set}$ be a functor and let $R \subseteq X_1 \times X_2$ be a binary relation. The (T) -lifted relation $\bar{T}R \subseteq TX_1 \times TX_2$ is given by

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

where $\pi_i : R \rightarrow X_i$ is the i th projection map.

The relation lifting is well-defined for an arbitrary set functor. Nevertheless, in order to ensure that the semantics of the ∇ -language is well-behaved, we make one more assumption on the functor T : we require the functor to preserve weak pullbacks. This ensures that \bar{T} can be seen as a functor on the category Rel of sets and relations.

Proposition 3.6. Let T be a set functor and \bar{T} its associated relation lifting. We have $\bar{T}(R \circ S) = \bar{T}R \circ \bar{T}S$ for all relations $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ iff T preserves weak pullbacks.

This fact is well-known and a proof can be easily obtained by carefully examining the argument in [7, page 43]. A complete proof of Prop. 3.6 can be found in [99]. Weak pullback preservation is crucial for the ∇ -language to be invariant under behavioural equivalence. Recent research [84] tries to overcome this restriction by changing the definition of relation lifting. In *loc.cit.* the authors manage to define a ∇ -logic for the monotone neighbourhood functor. The question whether the relation lifting in [84] has a nice categorical formulation which can be generalised to arbitrary set functors is under investigation.

From now on, when dealing with the ∇ -language, we fix a standard and weak pullback preserving set functor. The following proposition lists important properties of the relation lifting for such functors. We refer to [59] for proofs and a more detailed discussion.

Proposition 3.7. Let $T : \text{Set} \rightarrow \text{Set}$ be a standard, weak pullback preserving set functor and let \bar{T} the corresponding relation lifting. Then (1) \bar{T} is an endofunctor on the category Rel of sets and relations, (2) for any two relations $R, S \subseteq X \times Y$ we have $R \subseteq S$ implies $\bar{T}R \subseteq \bar{T}S$, and (3) \bar{T} commutes with taking restrictions: $\bar{T}(R \upharpoonright_{Y_1 \times Y_2}) = (\bar{T}R) \upharpoonright_{TY_1 \times TY_2}$ for any relation $R \subseteq X_1 \times X_2$ and sets $Y_1 \subseteq X_1$, $Y_2 \subseteq X_2$,

The semantics of ∇ -formulas hinges on the preliminaries above, and takes the following form:

Definition 3.8. Let $T : \text{Set} \rightarrow \text{Set}$ be a standard, weak pullback preserving set functor and let (W, γ) be a T -coalgebra. We define the satisfaction relation $\models W \times \mathcal{L}^T$ by induction as follows:

$$\begin{aligned} w \models \top & \quad \text{for all } w \in W \\ w \models \bigwedge \Phi & \quad \text{if } w \models A \text{ for all } A \in \Phi \\ w \models \bigvee \Phi & \quad \text{if there is } A \in \Phi \text{ with } w \models A \\ w \models \neg A & \quad \text{if not } w \models A \\ w \models \nabla \alpha & \quad \text{if } (\gamma(w), \alpha) \in \bar{T}(\models_{W \times \mathcal{L}_n^T}) \quad \text{for } \nabla \alpha \in \mathcal{L}_{n+1}^T. \end{aligned}$$

Finally we write $A \models B$ for two formulas $A, B \in \mathcal{L}^T$ if for all T -coalgebras (W, γ) and all states $w \in W$ we have $w \models A$ implies $w \models B$.

Remark 3.9. Note that for $\nabla\alpha \in \mathcal{L}_{n+1}^T$ we have $\alpha \in T\mathcal{L}_n^T$ and hence

$$\begin{aligned} (\gamma(w), \alpha) \in \overline{T}(\models \upharpoonright_{W \times \mathcal{L}_n^T}) & \text{ iff } (\gamma(w), \alpha) \in \overline{T}(\models) \upharpoonright_{TW \times T\mathcal{L}_n^T} \\ & \text{ iff } (\gamma(w), \alpha) \in \overline{T}(\models) \end{aligned}$$

where the first and the second equivalence follow from item (3) and item (2) of Proposition 3.7, respectively. Therefore we have $w \models \nabla\alpha$ iff $(\gamma(w), \alpha) \in \overline{T}(\models)$, which is precisely Moss' original definition of the semantics of the ∇ -operator.

The reader will have noticed that we do not include propositional variables in the ∇ -language \mathcal{L} . Variables can be treated by moving to a coloured version of the endofunctor under consideration: we put $T'X = \mathcal{P}(P) \times TX$ for a set P of propositional variables so that T -models are in 1 – 1-correspondence to T' -coalgebras. Concretely, in order to obtain a ∇ -language for Kripke models, one considers the functor $T = \mathcal{P}(P) \times \mathcal{P}_-$ where P denotes the set of propositional variables. A ∇ -formula in \mathcal{L}^T is then of the form $\nabla(C, \Phi)$ with $C \subseteq P$ and $\Phi \subseteq \mathcal{P}_\omega \mathcal{L}$. Translated to the syntax of normal modal logic, the formula $\nabla(C, \Phi)$ corresponds to the formula

$$\bigwedge_{p \in C} p \wedge \bigwedge_{p \notin C} \neg p \wedge \square \bigvee \Phi \wedge \bigwedge_{A \in \Phi} \diamond A.$$

The general spirit of ∇ -languages is probably best explained in terms of concrete examples.

Example 3.10. 1. Let $T = C \times _$ for some set C . In this case ∇ -formulas are of the form $\nabla(c, A)$ where $c \in C$ (a ‘‘colour’’) and $A \in \mathcal{L}$ is another formula. Let $(W, \gamma : W \rightarrow C \times W)$ be a T -coalgebra. Then $\nabla(c, A)$ is true at a state $w \in W$ with $\gamma(w) = (c', w')$ if $c = c'$ and $w' \models A$.

2. If we consider the power set functor $T = \mathcal{P}$, we obtain ∇ -formulas of the form $\nabla\{A_1, \dots, A_n\}$ where A_1, \dots, A_n are formulas in \mathcal{L} . Note that the argument of the ∇ -operator is a *finite* set of formulas. The semantics of ∇ can be nicely expressed using the $\{\square, \diamond\}$ -syntax of ‘‘standard’’ modal logic:

$$w \models \nabla\{A_1, \dots, A_n\} \quad \text{if} \quad w \models \bigwedge_{1 \leq i \leq n} \diamond A_i \wedge \square \bigvee_{1 \leq i \leq n} A_i.$$

More formally we have that a state x in some T -coalgebra (W, γ) makes $\nabla\{A_1, \dots, A_n\}$ true if

- (i) $\forall A \in \{A_1, \dots, A_n\} \exists w' \in \gamma(w) w' \models A$
- (ii) $\forall w' \in \gamma(w) \exists A \in \{A_1, \dots, A_n\} w' \models A.$

3. Let $T = \mathcal{D}$ be the distribution functor from Section 2. After passing to a standard variant of \mathcal{D} , the language $\mathcal{L}^{\mathcal{D}}$ takes the following form: A ∇ -formula in $\mathcal{L}^{\mathcal{D}}$ is of the form $\nabla\mu$ where $\mu : \mathcal{L}^{\mathcal{D}} \rightarrow [0, 1]$ is a probability distribution over formulas with finite support. Let $(W, \gamma : W \rightarrow \mathcal{D}W)$ be an arbitrary \mathcal{D} -coalgebra. The formula $\nabla\mu$ is true at state $w \in W$ if for all $v \in W$ and all $A \in \mathcal{L}^{\mathcal{D}}$ there are real numbers $\rho_{v,A} \in [0, 1]$ such that

- $\rho_{v,A} \neq 0$ only if $v \models A$, $\xi(w)(v) \neq 0$ and $\mu(A) \neq 0$
- $\sum_{A' \in \mathcal{L}^{\mathcal{D}}} \rho_{v,A'} = \xi(w)(v)$ for all $v \in W$
- $\sum_{v \in W} \rho_{v,A} = \mu(A)$ for all $A \in \mathcal{L}^{\mathcal{D}}$.

For instance, the formula $\nabla\{A \mapsto r, \top \mapsto (1 - r)\}$ for some $A \in \mathcal{L}^{\mathcal{D}}$ and some $r \in [0, 1]$ holds at a state w in a \mathcal{D} -coalgebra (W, γ) iff

$$\sum_{v \in W, v \models A} \gamma(w)(v) \geq r$$

ie., iff the formula A holds in a successor of w with probability at least r .

Compared to predicate liftings, the drawback of ∇ -languages is the requirement for the endofunctor to preserve weak pullbacks, excluding examples such as neighbourhood frames and conditional frames. On the other hand, one obtains logics without having to specify other ingredients, such as liftings and their interpretation. This advantage will become even more apparent when we will discuss a complete axiomatisation for the ∇ -logic in the next section. First we turn to an abstract view on coalgebraic logics that unifies both the predicate lifting approach and the ∇ -approach under a common umbrella.

3.3 An Abstract View on Coalgebraic Logics

In concrete terms, we can make logical statements about coalgebras using both predicate liftings and the ∇ -modality. In either case, the logical language extends classical propositional logic which is interpreted over a coalgebra (W, γ) in a standard way using the Boolean algebra structure on $\mathcal{P}(W)$. As the languages considered above extend classical propositional logic, we may afford the luxury of not distinguishing between formulas that are propositionally equivalent. Quotienting the set \mathcal{F} of formulas by propositional equivalence (denoted by \sim), we effectively equip the set (of equivalence classes of) formulas with a Boolean algebra structure that the interpretation should preserve.

Given that \mathcal{F}/\sim carries a Boolean algebra structure, the most convenient way to extend propositional logic with modal operators is then to consider algebras for an endofunctor *in the category of Boolean algebras*. That is, extending (classical) propositional logic with modal operators amounts to moving from Boolean algebras A to L -algebras (A, α) where $A \in \mathbf{BA}$ is a Boolean algebra, $\alpha : LA \rightarrow A$ is the structure map and $L : \mathbf{BA} \rightarrow \mathbf{BA}$ is the signature functor that determines the precise nature of the (additional) operators.

In other words, we can understand the syntax of a coalgebraic modal logic as given by an endofunctor $L : \mathbf{BA} \rightarrow \mathbf{BA}$ where the free L -algebra over a set V of propositional variables plays the role of formulas, quotiented by propositional equivalence and congruence. The interpretation function arises by turning a T -model into an L -algebra. We can see this as an algebraisation functor $\mathbb{A} : \mathbf{Coalg}(T) \rightarrow \mathbf{Alg}(L)$ that is induced by a natural transformation $\delta : LP \rightarrow PT^{\text{op}}$ where $P : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{BA}$ is contravariant powerset functor. Every such natural transformation defines an algebraisation functor

$$\mathbb{A} : \mathbf{Coalg}(T) \rightarrow \mathbf{Alg}(L), (X, \gamma) \mapsto (PX, \gamma^{-1} \circ \delta_X).$$

Example 3.11. Suppose that T extends to a Λ -structure and $L : \mathbf{BA} \rightarrow \mathbf{BA}$ is given by $LA = F\{\heartsuit(a_1, \dots, a_n) \mid a_1, \dots, a_n \in UA, \heartsuit \in \Lambda \text{ } n\text{-ary}\}$ where $U : \mathbf{BA} \rightarrow \mathbf{Set}$ is the forgetful functor and F is its left adjoint. Consider the family of mappings

$$\delta_X : LPX \rightarrow PTX$$

given by the adjoint transpose

$$\frac{\delta_X : F\{\heartsuit(a_1, \dots, a_n) \mid \heartsuit \in \Lambda \text{ } n\text{-ary}, a_1, \dots, a_n \in \mathcal{P}(X)\} \rightarrow PTX}{\delta_X^\# : \{\heartsuit(a_1, \dots, a_n) \mid \heartsuit \in \Lambda \text{ } n\text{-ary}, a_1, \dots, a_n \in \mathcal{P}(X)\} \rightarrow UPTX}$$

where $\delta_X^\#(\heartsuit(a_1, \dots, a_n)) = \llbracket \heartsuit \rrbracket_X(a_1, \dots, a_n)$. The naturality of predicate liftings guarantees that δ is in fact a natural transformation. It is a matter of unfolding the definitions that the semantics via algebraisation coincides with the concretely given semantics earlier.

Example 3.12. The finitary ∇ -language for a given weak pullback preserving set functor also fits into the abstract approach. The language induces a functor $M : \text{BA} \rightarrow \text{BA}$ given by

$$MA = F\{\nabla\alpha \mid \alpha \in T_\omega UA\}$$

and the algebraisation functor $\mathbb{A} : \text{Coalg}(T) \rightarrow \text{Alg}(M)$ is induced by the natural transformation $\delta : MP \rightarrow PT^{\text{op}}$ defined by

$$\begin{aligned} \delta_C : MPC &\rightarrow PTC \\ \nabla\alpha &\mapsto \{t \in TC \mid t\bar{T}(\in)\alpha\} \end{aligned}$$

More details on this abstract algebraic view on the ∇ -logic can be found in [57].

The two above examples motivate the following definition of an abstract modal logic.

Definition 3.13. If $T : \text{Set} \rightarrow \text{Set}$ is an endofunctor, an *abstract coalgebraic logic* over T is a pair (L, δ) where $L : \text{BA} \rightarrow \text{BA}$ is an endofunctor (that determines the syntax) and $\delta : LP \rightarrow PT^{\text{op}}$ is a natural transformation (that defines the interpretation).

If (L, δ) is an abstract coalgebraic logic and V is a (fixed) set of propositional variables, we write $\mathcal{F}(L)$ for the carrier of the L -algebra that is free over $F(V)$ where again $F(V)$ is the free Boolean algebra over a set of variables V . Given $M = (W, \gamma, \pi) \in \text{Mod}(T)$, the *interpretation* of $\mathcal{F}(L)$ in M is the mapping $\llbracket \cdot \rrbracket_M : \mathcal{F}(L) \rightarrow \text{PW}$ is the unique L -algebra morphism $\llbracket \cdot \rrbracket : \mathcal{F}(L) \rightarrow \text{PW}$ from $\mathcal{F}(L)$ to $(\text{PW}, \gamma^{-1} \circ \delta_W)$ that makes the diagram

$$\begin{array}{ccc} \mathcal{F}(L) & \xrightarrow{U\llbracket \cdot \rrbracket} & \text{PW} \\ & \swarrow i & \nearrow \pi \\ & V & \end{array}$$

commute, where $i : V \rightarrow \mathcal{F}(L)$ is the injection of variables into the free algebra, and $U : \text{Alg}(L) \rightarrow \text{Set}$ is the forgetful functor.

We will see in the sequel how properties of coalgebraic logics (such as soundness, completeness) are reflected in the abstract setting and induce conditions on δ that give rise to the corresponding properties.

4 Expressivity of Modal Logics

Given a coalgebraic logic of any flavour, one may ask whether the logic does indeed provide enough power to describe particular properties of a system. The ‘gold standard’ in this context is the Hennessy-Milner property, first formulated in [46]. Informally speaking, a logic has the Hennessy-Milner property if it is powerful enough to distinguish non-bisimilar worlds in a model. In the coalgebraic context, one finds several different generalisations of bisimilarity [97] but it is argued that the notion of ‘behavioural equivalence’ provides the most natural generalisation:

Definition 4.1. Suppose $T : \text{Set} \rightarrow \text{Set}$ is an endofunctor and (C, γ) and $(D, \delta) \in \text{Coalg}(T)$. Two worlds $c \in C$ and $d \in D$ are *behaviourally equivalent* if there exists $(E, \epsilon) \in \text{Coalg}(T)$ and a pair of coalgebra morphisms $f : (C, \gamma) \rightarrow (E, \epsilon)$ and $g : (D, \delta) \rightarrow (E, \epsilon)$ such that $f(c) = g(d)$.

If we take a morphism of T -models $f : (C, \gamma, \pi) \rightarrow (D, \delta, \sigma)$ to be a coalgebra morphism $f : (C, \gamma) \rightarrow (D, \delta)$ that satisfies $f^{-1} \circ \sigma = \pi$ then the same definition applies to behavioural equivalence of T -models where we require that c and d are identified by a morphism of T -models.

We refer to [97] for a detailed discussion of different notions of observable equality in the coalgebraic context, and conditions under which they are equivalent. We note that behavioural equivalence as introduced here is defined as a *co-span*: two worlds are behaviourally equivalent if and only if they can be identified by a pair of coalgebra morphisms. If a final T -coalgebra (Z, ζ) exists, this is equivalent to both points being identified by the unique map into (Z, ζ) .

In the following, we (informally) call a logic *expressive* if any two states that are not behaviourally equivalent can be distinguished by a logical formula. In the classical case of Hennessy-Milner logic over transition systems, this only holds if the transition systems are *image finite*. Moreover, it has been shown that a functor T only admits an expressive logic if the final T -coalgebra exists [36, 58] – at least if we require that the collection of formulas of the logic is a *set*. As a consequence, we cannot expect the Hennessy-Milner property to hold without a restriction on the branching degree. As it turns out, the appropriate generalisation of image-finiteness to the coalgebraic framework is to require that the signature functor $T : \text{Set} \rightarrow \text{Set}$ is finitary.

Definition 4.2. A functor $T : \text{Set} \rightarrow \text{Set}$ is *finitary*, if it preserves ω -filtered colimits, or equivalently if for any set X and any element $t \in TX$ there exists a finite subset $Y \subseteq X$ and $s \in TY$ such that $x = (Ti)(y)$ where $i : Y \rightarrow X$ is the inclusion.

Considering the elements of TX as structured successors of elements $x \in X$, we can understand T as finitary, if each structured successor $t \in TX$ can be manufactured from a finite set $Y \subseteq X$ of states. This finiteness assumption is crucial for expressiveness both in the context of predicate liftings and for the cover modality.

4.1 Logics via Predicate Liftings

To describe the relationship between logical and behavioural equivalence in the predicate lifting approach, suppose that Λ is a similarity type and $T : \text{Set} \rightarrow \text{Set}$ is a Λ -structure. Given that behavioural equivalence is defined in terms of coalgebra morphisms, the first step in establishing a correspondence is to relate the interpretation of formulas and coalgebra morphisms. The slogan here is that ‘morphisms preserve semantics’.

Proposition 4.3. *Suppose $M = (W, \gamma, \pi)$ and $N = (V, \delta, \sigma) \in \text{Mod}(T)$ and $A \in \mathcal{F}(\Lambda)$. Then*

$$\llbracket A \rrbracket_M = f^{-1}(\llbracket A \rrbracket_N)$$

whenever $f : (W, \gamma, \pi) \rightarrow (V, \delta, \sigma)$ is a morphism of T -models.

The statement follows immediately from the naturality of predicate liftings where compatibility of f with the valuations is needed to establish the case where $A = p$ is a propositional variable. We obtain stability of logical equivalence under behavioural equivalence as a direct corollary.

Corollary 4.4. *Suppose (W, γ, π) and $(V, \delta, \sigma) \in \text{Coalg}(T)$ and $(w, v) \in C \times D$ are behaviourally equivalent. Then $w \models A$ if and only if $v \models A$ for all $A \in \mathcal{F}(\Lambda)$.*

The converse of this corollary needs additional assumptions. Even in the classical case of labelled transition systems, bisimilarity and logical equivalence only coincide for finitely branching systems. Moreover, we have to stipulate that the set Λ of predicate liftings has enough ‘distinguishing power’. We begin with the latter [86].

Definition 4.5. A Λ -structure for T is *separating* if, for all sets X , the map of type

$$TX \rightarrow \prod_{\heartsuit \in \Lambda \text{ } n\text{-ary}} \mathcal{P}(\mathcal{P}(X)^n)$$

defined by $t \mapsto \{(A_1, \dots, A_n) \in \mathcal{P}(X)^n \mid t \in \llbracket \heartsuit \rrbracket(A_1, \dots, A_n)\}$ is injective.

In other words, a Λ -structure is separating if elements of TX can be distinguished by lifted predicates. This condition will guarantee that there are ‘enough’ liftings available to distinguish individual points. For the second assumption that generalises finite branching, we need to require that the endofunctor T is finitary. Given both, we can prove:

Theorem 4.6. *Suppose that T is finitary and extends to a separating Λ -structure. Then logical equivalence and behavioural equivalence coincide, i.e. given (W, γ, π) and (V, δ, σ) in $\text{Mod}(T)$ we have that w and v are behaviourally equivalent if and only if they satisfy the same formulas.*

This theorem was first established in [76] and then generalised to the present form in [86]. In fact, *op.cit.* establishes a slightly stronger statement for κ -accessible functors, at the expense of considering a logical language with conjunctions of size less than κ , for κ a regular cardinal. Expressivity of coalgebraic logics transfers smoothly to a categorical setting that we will discuss in Section 4.3.

There are plenty of examples of functors that extend to separating Λ -structures. In particular, we obtain separation for all the examples discussed in Section 3.1, so that the Hennessy-Milner property holds as soon as the underlying endofunctor is finitary. While this is not the case for the unbounded powerset functor, the finitary powerset functor $\mathcal{P}_\omega(X) = \{A \subseteq X \mid A \text{ finite}\}$, the multiset and distribution functor are finitary; we refer to [86] for more examples.

4.2 Expressivity and the cover modality

In contrast to languages based on liftings, ∇ -languages automatically enjoy the Hennessy-Milner property as long as T is finitary.

Proposition 4.7. *Let T be a finitary set functor and let $a \in \mathcal{L}^T$ be a formula. Then logical and behavioural equivalence coincide. That is, For any two T -coalgebras (W, γ) and (V, δ) and states $w \in W$ and $v \in V$ we have that w and v are behaviourally equivalent if and only if they satisfy the same formulas.*

That the truth of ∇ -formulas is invariant under behavioural equivalence can be proven using the well-known fact [83, 82] that for weak pullback preserving functors, behavioural equivalence between T -coalgebras can be characterised using the relation lifting \bar{T} . The proof of the converse direction of the proposition relies on the representation of the final T -coalgebra for a finitary endofunctor T given in [105] where every element of the carrier is approximated by its n -step behaviour that can be taken as an element of $T^n 1$. The heart of the proof consists of inductively defining for each $n \in \omega$ a (finitary) formula that characterises the n -step behaviour of a given state. To this aim we inductively define maps $\chi_n : T^n 1 \rightarrow \mathcal{L}^T$ for $n \in \omega$ by putting

$$\chi_0(w) := \top \quad \text{and} \quad \chi_{n+1}(w) := \nabla(T\chi_n)(w) \quad \text{for all } n \in \omega.$$

It is now not difficult to see that the formulas of the form $\chi_n(w)$ characterise the n -step behaviour of a given state w in a T -coalgebra (W, γ) .

4.3 Expressivity in the Abstract Approach

We have seen in Section 3.3 that both the predicate lifting approach and the ∇ -modality can be captured by the notion of ‘abstract coalgebraic logic’, defined by a functor $L : \mathbf{BA} \rightarrow \mathbf{BA}$ that admits free algebras of ‘formulas’ where the interpretation is given by a natural transformation $\delta : LP \rightarrow PT^{\text{op}}$. It is therefore natural to ask whether an abstract coalgebraic logic, given by L and δ , has the Hennessy-Milner property. In fact, this question can be asked in a much more general setting, where the adjunction $\text{Uf} \dashv \text{P}$ where $\text{P} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{BA}$ sends a set to the Boolean algebra of its subsets and $\text{Uf} : \mathbf{BA} \rightarrow \mathbf{Set}^{\text{op}}$ maps a Boolean algebra to its set of ultrafilters, can be replaced by an arbitrary adjunction. We refer to [50] for details and proofs and content ourselves with the setup described in Section 3.3.

Given that abstract languages for coalgebras are induced by a functor $L : \mathbf{BA} \rightarrow \mathbf{BA}$ and their interpretation is given by a natural transformation $\delta : LP \rightarrow PT^{\text{op}}$, it is to be expected that expressivity amounts to a certain coherence condition between L and δ . It turns out to be more convenient to formulate this coherence condition in terms of the adjoint transpose $\lambda : T^{\text{op}}\text{Uf} \rightarrow \text{Uf}L$ of δ induced by the adjunction $\text{Uf} \dashv \text{P}$:

$$\lambda : T^{\text{op}}\text{Uf} \xrightarrow{\epsilon_{T^{\text{op}}\text{Uf}}} \text{Uf}PT^{\text{op}}\text{Uf} \xrightarrow{\text{Uf}\delta_{\text{Uf}}} \text{Uf}LP\text{Uf} \xrightarrow{\text{Uf}L\eta} \text{Uf}L$$

where η and ϵ are the unit and counit of the adjunction $\text{Uf} \dashv \text{P}$, respectively. It can be shown [50, Proposition 2] that there is a 1-1 correspondence between natural transformations of type $LP \rightarrow PT^{\text{op}}$ and natural transformations of type $T^{\text{op}}\text{Uf} \rightarrow \text{Uf}L$ so that the adjoint transpose of δ is uniquely determined. The main result [50, Theorem 4] concerning the Hennessy-Milner property and abstract coalgebraic logics can now be stated as follows:

Theorem 4.8. *An abstract coalgebraic logic (L, δ) has the Hennessy-Milner property whenever all components of the adjoint transpose $\lambda : T^{\text{op}}\text{Uf} \rightarrow \text{Uf}L$ are injective.*

This theorem was anticipated in [54, Proposition 5.7] where an abstract coalgebraic logic has been derived from a given set of predicate liftings. In *loc.cit.* the expressivity of a coalgebraic logic is related to δ being surjective, but, unlike in [50], this is only carried out in a set-theoretic context. Intuitively, expressivity hinges on the fact that the semantic map given by δ must be rich enough to describe all of $T^n 1$ which is guaranteed by injectivity of the adjoint transpose.

5 Reasoning and Decision Procedures

In the previous section, we have discussed logical languages and their interpretation over coalgebraic models. However one important ingredient was missing: a purely syntactic calculus that allows us to derive all semantically valid formulas. Unsurprisingly, the deduction systems come in two flavours. In the predicate lifting approach, syntax and semantics are *a priori* independent: the syntax is given by a modal similarity type Λ whereas models are coalgebras for an endofunctor T , and the glue is provided by assigning a predicate lifting for T to every modality in Λ . As a consequence, the interpretation of a logical formula depends on the choice of predicate liftings, and a deduction system has to be coherent with this choice. For logics in terms of the ∇ -modality there is no such element of choice, and every endofunctor T (preserving weak pullbacks) directly induces both a logical language, and a deduction system. We now describe both approaches in detail.

5.1 Languages via Predicate Liftings

In the predicate lifting approach to modal logics, every similarity type Λ induces a set $\mathcal{F}(\Lambda)$ of formulas that arise by extending propositional logic with modal operators from Λ . Given that the propositional part of the logics is interpreted purely classically (over Boolean algebras $\mathcal{P}(W)$ for models (W, γ, π)), the associated deduction systems extend classical propositional logic with rules or axioms involving the modalities in Λ .

Given that our goal is to characterise the set of formulas that are universally valid over the class of *all* T -coalgebras, these axioms and rules assume a very specific form in the sense that they (only) speak about immediate successor states. In the coalgebraic formulation of modal logics, we focus on *single-step* behaviour: the essential ingredient of a coalgebraic model is the transition function $\gamma : W \rightarrow TW$ that maps states to (structured) successors where the precise structure of successors depends on the endofunctor T . Dually, the essential ingredient of a proof system has to provide a way to infer universally valid properties of successors given valid assertions over states.

$$\frac{\text{states}}{\text{successors}} \sim \frac{\text{properties of states}}{\text{properties of successors}}$$

Depending on the nature of the calculus, this is most conveniently formulated either in terms of axioms (for Hilbert-style calculi) or rules (for Gentzen Systems). In both flavours, axioms (resp. rule conclusions) purely assert properties of successor states. Syntactically, this is captured by stipulating that their modal nesting depth is uniformly equal to one. The following notations are helpful for this purpose.

Notation 5.1. Let Λ be a modal similarity type. If F is a set (of formulas), we write $\Lambda(F) = \{\heartsuit(A_1, \dots, A_n) \mid \heartsuit \in \Lambda \text{ } n\text{-ary}, A_i \in F\}$ for the set of formulas that arises from F by applying precisely one modal operator. Furthermore we write $\text{Prop}(F)$ to denote the set of Boolean combinations of formulas in F .

Informally, modal operators $\heartsuit \in \Lambda$ specify properties of successor states, and formulas in $\Lambda(F)$ therefore make assertions about successor states. We can make this intuition formal if we assume an interpretation $\llbracket A \rrbracket \subseteq X$ of formulas $A \in F$ over a state space X which induces an interpretation of formulas in $\Lambda(F)$ as predicates on successors, i.e. subsets of TX . We make this precise for the case where F consists either of propositional variables, or Boolean combinations of propositional variables.

Definition 5.2 (One-step semantics). Suppose X is a set and $\tau : \mathcal{P} \rightarrow \mathcal{P}(X)$ is a valuation. We write $\llbracket A \rrbracket_{(X, \tau)}$ for the canonical interpretation of $A \in \text{Prop}(\mathcal{P})$ in the Boolean algebra $\mathcal{P}(X)$, that is

$$\llbracket p \rrbracket_{(X, \tau)} = \tau(p) \quad \llbracket A \wedge B \rrbracket_{(X, \tau)} = \llbracket A \rrbracket_{(X, \tau)} \cap \llbracket B \rrbracket_{(X, \tau)} \quad \llbracket \neg A \rrbracket_{(X, \tau)} = X \setminus \llbracket A \rrbracket_{(X, \tau)}$$

and put $X, \pi \models A$ in case $\llbracket A \rrbracket_{(X, \tau)} = X$.

If Λ is a modal similarity type, T is a Λ -structure and $A \in \text{Prop}(\Lambda(\text{Prop}(\mathcal{P})))$, we write $\llbracket A \rrbracket_{(TX, \tau)}$ for the *one-step semantics* (or the interpretation) of A in the Boolean algebra $\mathcal{P}(TX)$ that is given by

$$\llbracket \heartsuit(A_1, \dots, A_n) \rrbracket_{(TX, \tau)} = \llbracket \heartsuit \rrbracket_X(\llbracket A_1 \rrbracket_{(X, \tau)}, \dots, \llbracket A_n \rrbracket_{(X, \tau)})$$

where the clauses for the propositional connectives are given as above. We put $TX, \tau \models A$ in case $\llbracket A \rrbracket_{(TX, \tau)} = TX$.

As the semantics of modal operators is not fixed but induced by a choice of predicate liftings, we need to ensure that logical rules and axioms are consistent with the choice of liftings. Given that coalgebraic models focus on the one-step behaviour, these coherence conditions are formulated in terms of the one-step semantics and relate one-step semantics to a suitable notion of one-step derivability which depends on the calculus. Roughly speaking, one requires that every one-step valid formula can be derived purely propositionally from rule conclusions with valid premises. We briefly sketch this process for both Hilbert and sequent calculi.

Hilbert Calculi for Coalgebraic Logics

Hilbert systems for coalgebraic logics are given by extending propositional logic by rules and axioms for reasoning with modal operators, closing under uniform substitution and modus ponens. In order to completely axiomatise the class of all T -coalgebras, for some Λ -structure T , it suffices to consider a particularly simple form of axioms and rules.

Definition 5.3. A *one-step Hilbert Rule* over a modal similarity type Λ is a pair (A, B) where $A \in \text{Prop}(\mathsf{P})$ is a propositional formula and $B \in \text{Prop}(\Lambda(\text{Prop}(\mathsf{P})))$ is a propositional combination of formulas that arise by applying precisely one modality to purely propositional formulas. Rules of the form (\top, B) with universally valid premises are called (one-step Hilbert) Axioms, and we write A/B for a rule and identify an axiom (\top, B) with its conclusion B . If \mathcal{H} is set of one-step Hilbert rules, then the derivability predicate $\mathcal{H} \vdash$ is the least set of formulas that contains all propositional tautologies and is closed under substitution instances of rules in \mathcal{H} , modus ponens and uniform substitution. We write $\mathcal{H} \vdash A$ if A is derivable in this system.

In other words, one-step axioms and rules mention *precisely* one layer of modal operators. Given a similarity type Λ and a Λ -structure T (i.e. an endofunctor together with a predicate lifting for every modality in Λ) soundness and completeness follow if the Λ -structure T is ‘compatible’ with the modal rules in a suitable way. This is formulated in terms of the one-step semantics as follows:

Definition 5.4. Suppose T is a Λ -structure and \mathcal{H} is a set of one-step rules over Λ . If X is a set and $\pi : \mathsf{P} \rightarrow \mathcal{P}(X)$ is a valuation, we say that $A \in \text{Prop}(\Lambda(\text{Prop}(\mathsf{P})))$ is *one-step derivable* relative to X and π if A is a propositional consequence of the set containing all $B\sigma$ where $A/B \in \mathcal{H}$ and $\sigma : \mathsf{P} \rightarrow \text{Prop}(\mathsf{P})$ is a substitution with $(X, \pi) \models A\sigma$. We write $\mathcal{H}X, \pi \vdash A$ if A is one-step derivable relative to X and π . We say that \mathcal{A} is *one-step sound* (resp. *one-step complete*) if

$$TX, \pi \models A \text{ if (resp. only if) } \mathcal{H}X, \pi \vdash A$$

for all sets X and all $\pi : \mathsf{P} \rightarrow \mathcal{P}(X)$.

In other words, one-step derivable formulas are propositional combinations rule conclusions with valid premises. We note that the notions of one-step soundness and one-step completeness do *not* quantify over models: they merely relate one-step validity and one-step derivability so that checking one-step soundness or completeness is much less involved than a soundness or completeness proof for the logic at large. One may wonder whether complete axiomatisations in this form always exist. This question has been answered positively in [85]:

Proposition 5.5. *The set of all one-step sound Hilbert rules for a coalgebraic logic is one-step complete.*

For concretely given logics, the question of obtaining a one-step sound and complete set of rules is therefore mainly a question of finding a suitable and easy to work with representation of the rule set. More importantly, one-step soundness and one-step completeness imply the corresponding property for the logic at large.

Theorem 5.6. *Suppose that \mathcal{H} is one-step sound (resp. complete) with respect to a Λ -structure T . Then $\text{Mod}(T) \models A$ if (only if) $\mathcal{H} \vdash A$ for all $A \in \mathcal{F}(\Lambda)$.*

This can be shown either by directly constructing a (finite) model [85] which additionally establishes the small model property or by induction on the modal rank of formulas [75].

Example 5.7. 1. For the modal logic K the rule set containing necessitation $p/\Box p$ and the distribution axiom $\Box(p \rightarrow q) \rightarrow \Box p \rightarrow \Box q$ is one-step sound and complete. This is the standard axiomatisation of the modal logic K [13]. An alternative one-step sound and complete rule set for K consists of the rules $\bigwedge_{i=1}^n p_i \rightarrow p_0 / \bigwedge_{i=1}^n \Box p_i \rightarrow \Box p_0$ for all $n \geq 0$.

2. For graded modal logic, both the axiomatisations given in [18, 31] fit the format of one-step rules, and again it can be seen that they are one-step complete. Alternatively, we may use the rule set

$$\frac{\sum_{i=1}^n r_i p_i \geq 0}{\bigvee_{i=1}^n \text{sgn}(r_i) \langle k_i \rangle p_i}$$

where $n \geq 1$ and $r_1, \dots, r_n \in \mathbb{Z} - \{0\}$, subject to the side condition $\sum_{r_i < 0} |r_i| (k_i + 1) \geq 1 + \sum_{r_i > 0} r_i k_i$. The expression in the premise refers to the arithmetic of characteristic formulas, and encodes the (propositional) formula

$$\sum_{i=1}^n r_i p_i \geq 0 = \bigvee \left\{ \left(\bigwedge_{f(i)=1} p_i \wedge \bigwedge_{f(i)=0} \neg p_i \mid f : \{1, \dots, n\} \rightarrow \{0, 1\}, \sum_{i=1}^n r_i f(i) \geq 0 \right) \right\}$$

and $\text{sgn}(r)$ is negation in case $r < 0$, and nothing otherwise. This (admittedly more complex) rule set has been shown to be one-step sound and complete in [89]. With this rule set, we only need a very limited form of propositional reasoning to show one-step completeness (weakening suffices) and therefore allows us to prove cut-elimination in a sequent calculus setting. We come back to this in Example 5.11.

Deduction in the Hilbert systems that we have presented here is mainly driven by rules rather than axioms, however [85, Proposition 15] shows that in presence of the congruence rule $\bigwedge_{i=1}^n p_i \leftrightarrow q_i / \heartsuit(p_1, \dots, p_n) \leftrightarrow \heartsuit(q_1, \dots, q_n)$ for n -ary $\heartsuit \in \Lambda$, Hilbert systems can be formulated purely in terms of axioms.

Sequent Calculi for Coalgebraic Logics

Sequent calculi provide an alternative way to formulate deductive systems over coalgebraic semantics which are better suited for the complexity analysis of coalgebraic logics: we establish soundness and completeness and then derive complexity bounds for the satisfiability problem by analysing backwards proof search in the style of Ladner [65]. As we are dealing with a classical base and a generic set of modal operators, we use one-sided sequents with explicit negation as this avoids the need to extend the modal similarity type with dual operators. As for Hilbert-systems, soundness and completeness follow from the respective one-step notions, adapted to the sequent calculus setting.

Definition 5.8. A *sequent* over a similarity type Λ is a finite multiset of Λ -formulas and $\mathcal{S}(\Lambda)$ denotes the set of Λ -sequents. A *one-step sequent rule* over Λ is of the form $\Gamma_1 \dots \Gamma_n / \Gamma_0$, where $\Gamma_1, \dots, \Gamma_n \subseteq \mathcal{P} \cup \neg\mathcal{P}$ and $\Gamma_0 \subseteq \Lambda(\mathcal{P}) \cup \neg\Lambda(\mathcal{P})$ are finite sets of formulas and $\neg X = \{\neg A \mid A \in X\}$.

The derivability predicate $\mathcal{G} \vdash$ is the least set of sequents that is closed under the propositional rules

$$\frac{\Gamma, A, B}{\Gamma, A \wedge B} \quad \frac{\Gamma, \neg A \quad \Gamma, \neg B}{\Gamma, \neg(A \wedge B)} \quad \frac{\Gamma, A}{\Gamma, \neg\neg A} \quad \frac{}{\Gamma, p, \neg p}$$

and closed under substitution instances of the one-step rules in \mathcal{G} augmented with weakening, i.e. if $\Gamma_1 \dots \Gamma_n / \Gamma_0 \in \mathcal{G}$ is a rule with $\mathcal{G} \vdash \Gamma_i \sigma$ for $i = 1, \dots, n$, then $\mathcal{G} \vdash \Gamma_0 \sigma, \Delta$ for all Λ -sequents Δ .

As usual, we identify a singleton (multi)set A with its unique element $\{A\}$ and write Γ, Δ for the (multiset) union of Γ and Δ . Note that premise and conclusion of a one-step rule are *sets* rather than *multisets* of formulas and that our axiomatisation does not include either cut or contraction. An easy adaptation of the proof given in [100, Section 3.6] shows that the propositional rules are sound and complete with respect to (classical) propositional logic. Semantically, we read a sequent disjunctively, so that if T is a Λ -structure, $M \in \text{Mod}(T)$, we put

$$\llbracket \Gamma \rrbracket_M = \llbracket \bigvee \Gamma \rrbracket_M \quad \text{and} \quad \text{Mod}(T) \models \Gamma \iff \text{Mod}(T) \models \bigvee \Gamma$$

for a Λ -sequent Γ . One-step soundness and one-step completeness in the context of sequent calculi now take the following form:

Definition 5.9. Suppose \mathcal{G} is a set of one-step sequent rules over Λ and T is a Λ -structure. We say that $\Gamma \in \mathcal{S}(\Lambda(\mathcal{P}))$ is *one-step derivable* from \mathcal{G} relative to a set X and a valuation $\pi : \mathcal{P} \rightarrow \mathcal{P}(X)$ if there exists a rule $\Gamma_1 \dots \Gamma_n / \Gamma_0 \in \mathcal{G}$ and a substitution $\sigma : \mathcal{P} \rightarrow \mathcal{P}$ such that $(X, \pi) \models \bigvee \Gamma_i \sigma$ for all $i = 1, \dots, n$ and $\Gamma_0 \sigma \subseteq \Gamma$. We write $\mathcal{G}X, \pi \vdash \Gamma$ if Γ is one-step derivable under \mathcal{G} relative to X and π . Now \mathcal{G} is *one-step sound* (resp. one-step cut-free complete) if

$$\mathcal{G}X, \pi \vdash \Gamma \text{ if (resp. only if) } TX, \pi \models \Gamma$$

for all sets X , all valuations $\pi : \mathcal{P} \rightarrow \mathcal{P}(X)$ and all $\Gamma \in \mathcal{S}(\Lambda(\mathcal{P}))$.

The main difference to the corresponding notions for Hilbert-systems (Definition 5.4) is the notion of one-step derivability, where we require that a derivable sequent must be obtained as a superset of a (substituted) rule conclusion with valid premises, whereas in Hilbert-systems, we are allowed to combine multiple premises with the help of (classical) propositional reasoning. This closely resembles the difference between the reasoning process in Hilbert and Gentzen systems, where we can only introduce (but not eliminate) new connectives in the latter. As is the case for Hilbert-systems, we can establish completeness of the sequent calculus at large, given the (stronger) coherence conditions.

Theorem 5.10. Suppose \mathcal{G} is a set of sequent rules over a similarity type Λ and T is a Λ -structure. Then, for all $\Gamma \in \mathcal{S}(\Lambda)$ we have

$$\text{Mod}(T) \models \Gamma \text{ if (resp. only if) } \mathcal{G} \vdash \Gamma$$

in case \mathcal{G} is one-step sound (resp. one-step cut-free complete).

For the proof, one can stratify the set of Λ -sequents by modal rank, and show that one-step cut-free completeness propagates completeness for sequents of rank $< n$ to sequents of rank n [78]. Alternatively, one can exploit completeness of the Hilbert system that arises by translating sequent

rules to Hilbert-rules (where the rule $\Gamma_1 \dots \Gamma_n / \Gamma_0$ induces the Hilbert-rule $\bigvee \Gamma_1 \wedge \dots \wedge \bigvee \Gamma_n / \bigvee \Gamma_n$). Technically, this amounts to establishing cut-elimination, as the propositional reasoning in the Hilbert-calculus can only be simulated with the cut-rule in a sequent system [78]. To illustrate cut-free completeness, we revisit the logics discussed in Example 5.7.

Example 5.11. 1. For the basic modal logic K the rule set, containing for all $n \in \omega$ the rule $\neg p_1, \dots, \neg p_n, p_0 / \neg \Box p_1, \dots, \neg \Box p_n, \Box p_0$, is one-step cut-free complete.

2. The rule set

$$\frac{\sum_{i=1}^n r_i p_i \geq 0}{\{\text{sgn}(r_i) \langle k_i \rangle p_i \mid i = 1, \dots, n\}}$$

where $n \geq 1$ and $r_1, \dots, r_n \in \mathbb{Z} - \{0\}$, subject to the side condition $\sum_{r_i < 0} |r_i| (k_i + 1) \geq 1 + \sum_{r_i > 0} r_i k_i$, is one-step cut-free complete. Here we read the premise as a set of sequents by converting into conjunctive normal form.

In comparison with Example 5.7, it is easy to see that in general, cut-free completeness implies completeness but the converse is not necessarily the case. We refer to [89] for more examples.

Decision Procedures

To establish complexity bounds and decidability of coalgebraic logics, one can either take a semantic route ([45] is an example for the modal logic K) or implement satisfiability via backwards proof-search in a cut-free sequent calculus. In the latter approach, the subformula property of the sequent calculus under scrutiny together with suitable bounds on the height of the proof tree entails finiteness of the search space. Given that the height of proof trees is polynomially bounded by the end sequent this can be implemented on an alternating Turing machine [20] where the existential steps correspond to guessing a rule and the universal steps require us to establish provability of all rule premises. As a consequence, one can show that the satisfiability problem for coalgebraic logics is in PSPACE provided that rules can be decided in NP. This is not immediate for all logics of interest: in graded or probabilistic modal logic, the rule premises are too large to be polynomially representable, but can be represented by polynomial-size codes. Formally, one defines a rule set to be *PSPACE-tractable* if rules can be represented by a code of polynomial size (measured in the size of the conclusion) and it can be decided in NP whether a sequent is a premise of any given rule [89]. While it is easy to construct synthetic counterexamples that violate this property, it is satisfied in all logics of interest, including probabilistic and graded modal logic.

Theorem 5.12. *Suppose that \mathcal{G} is PSPACE-tractable. Then the problem of deciding whether $\mathcal{G} \vdash \Gamma$ is in PSPACE. As a consequence, if moreover \mathcal{G} is one-step cut-free complete and one-step sound over a Λ -structure T , the satisfiability problem of $\mathcal{F}(\Lambda)$ over $\text{Mod}(T)$ is in PSPACE.*

The above result has first been established, in a variant that does not explicitly mention sequent systems, in [87, 89] and has been related to proof search in Sequent calculi in [78]. All logics discussed in Section 3.1 can be axiomatised via a PSPACE-tractable rule set, and in particular the rule sets presented in Example 5.11 are PSPACE-tractable.

Given the prominence of the one-step transition behaviour, an alternative semantical approach to satisfiability is to construct the models step-by-step. If the semantics is defined in terms of a Λ -structure T , this hinges on the ability to represent enough elements of TX syntactically so that every pair (A, B) of formulas where $A \in \text{Prop}(\mathcal{P})$ and $B \in \text{Prop}(\Lambda(\mathcal{P}))$ that has a one-step model does also have a one-step model of polynomial size. If we think of the formulas A as a description of states and B as a property of successors, a one-step model consists of a pair (X, τ) where X is a valuation

with $X, \tau \models A$ and a point $t \in TX$ such that $t \in \llbracket B \rrbracket_{TX, \tau}$. In other words, the one-step polysize model property asserts that everything that is one-step satisfiable can be represented by a model of polynomial size (we refer to [88] for details). The proof of specific complexity bounds then hinges on whether we can check whether any given syntactic representation is indeed a one-step model of a one-step pair. If we can do this fast enough (in PSPACE) then this implies an overall PSPACE upper bound. In other words, we have the following theorem, first established in [88]:

Theorem 5.13. *If a Λ -structure T has the one-step polysize model property, and the one-step model checking problem is in PSPACE, then the satisfiability of $A \in \mathcal{F}(\Lambda)$ over $\text{Coalg}(T)$ can be decided in polynomial space.*

The decision procedure that underlies the proof of the above theorem essentially guesses a one-step model at each step, and the correctness is witnessed by the fact that all these one-step models can be glued together to obtain a satisfying model for the initial formula. Remarkably, this theorem also permits us to go beyond one of the limitations that we had previously considered: It does not only apply to satisfiability in the class of *all* T -coalgebras, but also allows us to capture so-called *non-iterative* logics, i.e. logics whose semantics imposes a local relation between states and successors, such as reflexivity in the case of relational semantics.

Non-Iterative Axioms

At this point, the diligent reader will have noticed that the format of one-step rules is rather rigid. For instance, axioms that – in the relational case – encode transitivity ($\Box\Box p \rightarrow \Box p$) or reflexivity ($\Box p \rightarrow p$) do not fit into the format of one-step rules. Semantically, on the other hand, we have formulated soundness and completeness with respect to the class of *all* models based on a given endofunctor T , whereas the class of transitive or reflexive frames does not arise as the class of *all* models over a given endofunctor. In order to accommodate axioms that go beyond what is expressible as one-step rules, we therefore should take a different approach, and expect soundness and completeness only with respect to a *subclass* of coalgebras based on an endofunctor T . This is again in analogy to the relational case: the logic T (an extension of an axiomatisation of K with the reflexivity axiom $\Box p \rightarrow p$) and the logic $K4$ (that extends K with $\Box p \rightarrow \Box\Box p$) are only (sound and) complete with respect to the class of reflexive (resp. transitive) Kripke frames. In full generality, there is little hope of establishing a general coalgebraic completeness result, as incompleteness already abounds in the relational setting: [16, 11, 26] give examples of extensions of the logic K that are necessarily incomplete with respect to any given class of Kripke frames. The source of incompleteness are axioms of modal rank ≥ 2 and incompleteness disappears if axioms are restricted to modal rank ≤ 1 .

Intuitively, a modal axiom of rank ≤ 1 asserts a local condition (such as reflexivity) that is required to hold between the ‘current’ state and its successor(s), whereas general axioms (like transitivity) impose a global condition on the entirety of the model. In the coalgebraic framework, the local coherence conditions between states and successors can be accommodated, so that every extension of a one-step complete rule set with so-called shallow (or non-iterative) axioms is complete with respect to the frame class it defines. More formally, one can prove the following:

Theorem 5.14. *Suppose that T is a Λ -structure and \mathcal{H} is one-step sound and complete with respect to T . If \mathcal{A} is a set of Λ -formulas with $\text{rank}(A) \leq 1$ for all $A \in \mathcal{A}$, then*

$$\text{Mod}(\mathcal{A}) \models B \iff \mathcal{H} + \mathcal{A} \vdash B$$

for all $B \in \mathcal{F}(\Lambda)$.

In the above theorem, $\text{Mod}(A)$ is the class of T -coalgebras that satisfies all substitution instances of $A \in \mathcal{A}$ and $\mathcal{H} + \mathcal{A} \vdash B$ if B can be derived using (substitution instances of) elements of \mathcal{A} as additional axioms. The proof [77] first constructs (finite) algebraic models that are then translated to T -coalgebras in a semantics-preserving way. While this works for Hilbert calculi, it is at present unknown how additional axioms of rank ≤ 1 can be integrated into a cut-free sequent calculus but Theorem 5.13 extends to the case of non-iterative frame conditions so that we obtain PSPACE upper bounds for decidability in many cases.

Global Assumptions and Nominals

The coalgebraic approach is readily extended to deal with so-called *global assumptions* and *nominals*. The former are a set of formulas that constrain the class of models over which logics are interpreted whereas the latter provide us with the possibility to pinpoint individual states in a model.

Definition 5.15. Suppose that Λ is a modal similarity type. If $A \subseteq \mathcal{F}(\Lambda)$ is a set (of global assumptions) then $\text{Mod}(A)$ consists of all $M \in \text{Mod}(T)$ so that $M \models A$ for all $A \in A$. We write $\text{Mod}(A) \models B$ if $M \models B$ for all $M \in \text{Mod}(A)$.

It turns out that both complete axiomatisations and decidability, as discussed before, can be readily extended to deal with global assumptions. In particular, completeness and complexity of coalgebraic logics over a set A of global assumptions follows from precisely the same coherence conditions that were also used to establish the corresponding properties in absence of global assumptions. This has been formulated in terms of a tableau calculus in [38] where the (tableau) rules for the calculus are the dualised sequent rules, extended with global assumptions:

Theorem 5.16. *Suppose that \mathcal{G} is one-step sound and one-step complete over a Λ -structure T . Then \mathcal{G} induces a tableau calculus that characterises satisfiability in $\text{Mod}(A)$ where $A \subseteq \mathcal{F}(\Lambda)$ is a finite set of global assumptions. Moreover satisfiability can be decided in EXPTIME if this is the case for \mathcal{G} .*

The main algorithmic ingredient of the above theorem is the use of *global caching* first applied to modal logics with relational semantics [39] that realises the (optimal) EXPTIME-bound for satisfiability over global assumptions, which also allows us to relax the complexity of \mathcal{G} from PSPACE to EXPTIME.

Nominals, or individuals as they are called in description logic, provide us with the facility to name individual states in a model. The extension of the modal logic K with nominals is usually referred to as *hybrid logic* [5]. Formally, this is captured by the following extended language, where as usual Λ denotes an arbitrary similarity type, and \mathbb{N} a denumerable set of nominals:

$$\mathcal{F}(\Lambda) \ni A, B ::= p \mid A \wedge B \mid \neg A \mid \heartsuit(A_1, \dots, A_n) \mid n \mid @_n A$$

where $p \in \mathbb{P}$ is a propositional variable, $n \in \mathbb{N}$ a nominal and $\heartsuit \in \Lambda$ is n -ary. Compared to the language introduced in Definition 3.2, we may now use nominals in place of propositional variables, and $@_n A$ asserts that formula A is satisfied at the point named n in a model (the operators $@_n$ are often called *satisfaction operators*).

Given a Λ -structure T , our notion of model changes slightly in that we also have to provide a valuation of nominals. That is, we consider *hybrid models* (the collection of which we still denote by $\text{Mod}(T)$) that now take the form (W, γ, π) where $(W, \gamma) \in \text{Coalg}(T)$ and $\pi : \mathbb{P} \cup \mathbb{N} \rightarrow \mathcal{P}(W)$ is a *hybrid valuation*, that is, $\pi(n)$ is a singleton set for all $n \in \mathbb{N}$. The interpretation of this extended language extends the clauses given in Definition 3.2 by

$$\llbracket n \rrbracket_M = \pi(n) \quad \llbracket @_n A \rrbracket_M = \{w \in W \mid \pi(n) \in \llbracket A \rrbracket_M\}$$

where $M = (W, \gamma, \pi)$ is a hybrid model, and we have identified $\pi(n)$ with its unique element in the clause above. The satisfaction operator $@_n$ effectively moves the evaluation context to the point $\pi(n)$ that n denotes in M , note that the interpretation of satisfaction operators is either the empty set or the whole carrier of the model.

It turns out that many tools and techniques known from hybrid logic (with relational semantics) can be generalised to its coalgebraic counterpart: we can capture universal validity both with a sequent calculus and a Hilbert-system, and we obtain complexity results for the satisfiability problem.

To establish completeness with respect to a Hilbert-calculus, we essentially have to axiomatise the algebra of satisfaction operators (see e.g. the chapter on Hybrid Logic in [13]). The only specific additional axiom that is needed in the coalgebraic context specifies how satisfaction operators indicate with the modalities that are semantically captured by predicate liftings. Here, one axiomatises the fact that an @-formula $@_n A$ is either valid in the entirety of the model, or nowhere at all. This leads to the axiom

$$(\text{mob})@_n p \rightarrow (\heartsuit(q_1, \dots, q_n) \leftrightarrow \heartsuit(@_n p \wedge q_1, \dots, @_n p \wedge q_n))$$

where (mob) stands for ‘make-or-break’. Intuitively, if we assume that $@_n p$ is valid at a certain point in a model, it will be universally valid in this model, so that the truth-sets of q and $@_n p \wedge q$ coincides irrespective of the interpretation of q .

Given a one-step sound and complete axiomatisation of the underlying modal operators (Definition 5.4) the above axioms are sufficient to achieve a sound and complete Hilbert-style calculus for hybrid coalgebraic logics, as shown in [72].

Theorem 5.17. *Suppose that \mathcal{H} is a set of one-step sound and complete one-step rules. If $\mathcal{H} \vdash$ is the derivability predicate induced \mathcal{H} , (mob) and the axiomatisation of the algebra of satisfaction operators, then \mathcal{H} is sound and complete with respect to $\text{Mod}(T)$, that is*

$$\mathcal{H} \vdash A \iff \text{Mod}(T) \models A$$

for all hybrid formulas $A \in \mathcal{F}(\Lambda)$.

The proof of the above theorem can be seen as an adaptation of the finite model construction given in [85] where one has to pay additional attention to the fact that the extension of nominals are actually singleton sets. A sequent-style axiomatisation of hybrid logic can be achieved by converting a one-step cut-free axiomatisation to @-prefixed form where every one-step sequent rule $\Gamma_1 \dots \Gamma_k / \Gamma_0$ induces the hybrid rule

$$\frac{@_n \Gamma_1 \sigma, @_t \Gamma_0 \sigma, \Delta \quad \dots \quad @_n \Gamma_k \sigma, @_t \Gamma_0 \sigma, \Delta}{@_t \Gamma_0 \sigma, \Delta} (n \notin \Gamma_0 \sigma, \dots, \Gamma_k \sigma, \Delta)$$

where σ is a substitution that is applied pointwise to sequents, and Δ is an arbitrary hybrid sequent and $@_n \Gamma$ is the result of prefixing every formula in Γ with $@_n$. The side condition ensures that n is a ‘fresh’ nominal which allows us to propagate information from conclusion to premise.

The main result here is completeness of an ensuing cut-free system that we can later use to determine the complexity of the satisfiability problem [72].

Theorem 5.18. *The hybrid sequent system induced by a one-step sound and cut-free complete set of one-step rules is sound, complete, and admits cut elimination. If \mathcal{G} is moreover PSPACE-tractable, then satisfiability of $A \in \mathcal{F}(\Lambda)$ in $\text{Mod}(T)$ can be decided in polynomial space.*

The system to which this theorem refers to contains, beside the hybridised versions of the one-step rules, the (hybridised) rules of propositional logic, and an axiomatisation of satisfaction operators. We refer to [72] for details.

Global assumptions and nominals can of course be combined in a single setting, where we can think of the global assumptions as collating knowledge about a specific application domain and nominals as specific individuals, leading to description logic reasoning [6] over coalgebraic models. The main question here is the satisfiability problem over a set of global assumptions in hybrid coalgebraic logic: Given a modal similarity type Λ together with a Λ -structure T , a set $\Sigma \subseteq \mathcal{F}(\Lambda)$ (of global assumptions), is a formula $A \in \mathcal{F}(\Lambda)$ satisfiable in a hybrid model M over T so that $M \models B$ for all $B \in \Sigma$?

To answer this question, we need to account for both the global assumptions and nominals which necessitates to combine both approaches laid out earlier. To decide satisfiability, one can either ‘guess’ the theory of the individual named points of a putative model [92] or design a special-purpose calculus where knowledge about individuals is propagated [38] leading to the same complexity bounds:

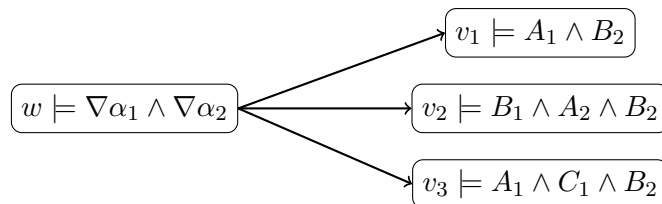
Theorem 5.19. *Let Λ be a similarity type and fix a Λ -structure T . If \mathcal{G} is one-step sound and cut-free complete, and moreover EXPTIME-tractable, then satisfiability of $A \in \mathcal{F}(\Lambda)$ in the class $\text{Mod}(\Sigma)$ can be decided in exponential time.*

5.2 An axiomatisation of the cover modality

For almost ten years the coalgebraic cover modality has been mainly studied semantically. Moss’ original paper [70, 71] mentions some sound logical principles, but a complete set axioms seemed out of reach. Therefore the recent observation that a relatively simple and natural set of axioms suffices to axiomatise ∇ was a positive surprise. Compared to the proof systems for coalgebraic modal logics using predicate liftings, the axioms for ∇ have the advantage that their shape is independent of the functor T under consideration.

We are now going to discuss a sound and complete set of axioms for the coalgebraic ∇ -logic that has been presented in [57]. In comparison to *op.cit.*, however, the axiomatisation we are presenting contains only three instead of four axioms for the ∇ -operator. The reason for that is that the axiom in [57] that concerned negation turned out to be redundant (this has been observed first in [12] for the case $T = \mathcal{P}$ and established in [55] for the general case). To simplify our presentation, we make the assumption that the set functor T under consideration maps finite sets to finite sets. The set of axioms for an arbitrary set functor is very similar and can be found in [57]. The axiomatisation of logics in terms of the cover modality ∇ hinges on the notion of *slim redistribution*, which is probably best understood by first looking at examples.

Example 5.20. Consider the case $T = \mathcal{P}$ and let (W, γ) be a T -coalgebra with $w \in W$ such that $w \models \nabla\alpha_1 \wedge \nabla\alpha_2$ for $\alpha_1 = \{A_1, B_1, C_1\} \subseteq \mathcal{L}^{\mathcal{P}}$, $\alpha_2 = \{A_2, B_2\} \subseteq \mathcal{L}^{\mathcal{P}}$ and with $\gamma(w) = \{v_1, v_2, v_3\}$. Then a possible model can be depicted as follows:



It is not difficult to see that in fact the following ∇ -formula that contains conjunctions of subformulas of α_1 and α_2 holds at w :

$$w \models \nabla \{ \bigwedge \{A_1, B_2\}, \bigwedge \{B_1, A_2, B_2\}, \bigwedge \{A_1, C_1, B_2\} \}.$$

The set $\Phi = \{ \{A_1, B_2\}, \{B_1, A_2, B_2\}, \{A_1, C_1, B_2\} \}$ is called a slim redistribution of $\{\alpha_1, \alpha_2\}$. More generally, as we will see, any conjunction of ∇ -formulas is equivalent to a disjunction of ∇ 's:

$$w \models \bigwedge \{ \nabla \alpha \mid \alpha \in \Gamma \} \quad \text{iff} \quad w \models \bigvee \{ \nabla (T \bigwedge) \Phi \mid \Phi \in \text{SRD}(\Gamma) \}$$

where $\text{SRD}(\Gamma)$ is the collection of slim redistributions of Γ .

Formally, (slim) redistributions are defined as follows.

Definition 5.21. Let T be a set functor. An element $\Phi \in T\mathcal{P}X$ is a *redistribution* of a set $\Gamma \in \mathcal{P}TX$ if $\Gamma \subseteq \{ \alpha \in TX \mid \alpha(\bar{T}) \in \Phi \}$, that is, every element of Γ is a lifted member of Φ . In case $\Gamma \in \mathcal{P}_\omega T_\omega X$, we call a redistribution Φ *slim* if $\Phi \in T_\omega \mathcal{P}_\omega (\bigcup_{\alpha \in \Gamma} \text{Base}(\alpha))$. Here for $\alpha \in T_\omega X$ we write $\text{Base}(\alpha)$ to denote the \subseteq -smallest subset U of X such that $\alpha \in TU$ (cf. [102] for well-definedness of Base). The set of slim redistributions of Γ is denoted as $\text{SRD}(\Gamma)$.

The motivation for introducing *slim* redistributions is to identify redistributions that agree on relevant formulas. While there are in general infinitely many redistributions of some $\Gamma \in \mathcal{P}_\omega T_\omega X$, there are only finitely many pairwise distinct slim ones.

Example 5.22. Let $T = C \times _$ be the functor mapping a set X to the set $C \times X$ and a function $f : X \rightarrow Y$ to the function $\text{id}_C \times f : C \times X \rightarrow C \times Y$. Consider an arbitrary set $\Gamma = \{ (c_1, x_1), \dots, (c_n, x_n) \} \in \mathcal{P}_\omega T_\omega X$. A redistribution of Γ is a pair $(c, \phi) \in T\mathcal{P}X$ with $c \in C$, $\phi \subseteq_\omega X$, such that (i) $c = c_i$ for all $i \in \{1, \dots, n\}$, and (ii) $\phi \supseteq \{x_1, \dots, x_n\}$. In particular, this means that $\text{SRD}(\Gamma) = \emptyset$ if there are $i_1, i_2 \in \{1, \dots, n\}$ with $c_{i_1} \neq c_{i_2}$. A redistribution (c, ϕ) of Γ is slim if $\phi = \{x_1, \dots, x_n\}$.

We now turn to the formulation of the axioms of the logic. Roughly speaking, there is one rule that replaces the congruence rule from modal logic, one axiom that allows us to replace a conjunction of ∇ -formulas by a disjunction of ∇ -formulas and one axiom that expresses the fact that ∇ can be distributed over conjunctions.

In order to be able to formulate these axioms we need to consider conjunctions and disjunctions as functions $\bigwedge : \mathcal{P}_\omega \mathcal{L}^T \rightarrow \mathcal{L}^T$ and $\bigvee : \mathcal{P}_\omega \mathcal{L}^T \rightarrow \mathcal{L}^T$ to which we can apply a functor, e.g. an application of T_ω yields two functions

$$\begin{aligned} T_\omega \bigvee : T_\omega \mathcal{P}_\omega X &\rightarrow T_\omega X \\ T_\omega \bigwedge : T_\omega \mathcal{P}_\omega X &\rightarrow T_\omega X. \end{aligned}$$

The following axioms constitute a sound and complete axiomatisation for functors T that preserve finite sets, ie. with the property that TX is finite whenever the set X is finite.

$$\frac{\{A \leq B \mid (A, B) \in Z\}}{\nabla \alpha \leq \nabla \beta} \quad (\alpha, \beta) \in \bar{T}(Z) \quad (\nabla 1)$$

$$\bigwedge \{ \nabla \alpha \mid \alpha \in \Gamma \} \leq \bigvee \{ \nabla (T_\omega \bigwedge) \Phi \mid \Phi \in \text{SRD}(\Gamma) \} \quad (\nabla 2_f)$$

$$\nabla (T_\omega \bigvee) \Phi \leq \bigvee \{ \nabla \beta \mid \beta \bar{T}(\epsilon) \in \Phi \} \quad (\nabla 3_f)$$

Here $A, B \in \mathcal{L}^T$, $\alpha, \beta \in T_\omega \mathcal{L}^T$, $\Gamma \in \mathcal{P}_\omega T_\omega \mathcal{L}^T$, $\Phi \in T_\omega \mathcal{P}_\omega \mathcal{L}^T$ and $Z \subseteq \mathcal{L}^T \times \mathcal{L}^T$ denotes a finite binary relation between formulas.

The fact that the functor T preserves finite sets ensures that the axioms $(\nabla 2_f)$ and $(\nabla 3_f)$ are well-formed: for arbitrary set functors T the disjunctions on the right hand side of $(\nabla 2_f)$ and $(\nabla 3_f)$ could become infinite. In the general case axioms $(\nabla 2_f)$ and $(\nabla 3_f)$ have to be replaced by rules [57]. Axiom $(\nabla 2_f)$ and axiom $(\nabla 3_f)$ could be formulated as equivalences: both the converse of $(\nabla 2_f)$ and the converse of $(\nabla 3_f)$ can be derived using $(\nabla 1)$.

The rule $(\nabla 1)$, which had been already used by Moss to define a “minimal” ∇ -logic, can be seen as a direct generalization of the congruence rule from modal logic.

Example 5.23. 1. Let $T = C \times _$. In this case $(\nabla 1)$ can be expressed in the following simpler form:

$$\frac{A \leq B}{\nabla(c, A) \leq \nabla(c, B)} \quad c \in C$$

where $A, B \in \mathcal{L}^T$.

2. Let $T = \mathcal{P}$ and consider the formulas $\nabla\{A, \perp\}$ and $\nabla\{B, C, \top\}$ with arbitrary formulas A and B . We put $Z = \{(A, \top), (\perp, B), (\perp, C)\}$. Obviously $(A', B') \in Z$ implies $A' \leq B'$ for all $A', B' \in \mathcal{L}^{\mathcal{P}}$ and we have $(\nabla\{A, \perp\}, \nabla\{B, C, \top\}) \in \overline{T}(Z)$. Therefore the following is a valid instance of the rule $(\nabla 1)$:

$$\frac{\{A' \leq B' \mid (A', B') \in Z\}}{\nabla\{A, \perp\} \leq \nabla\{B, C, \top\}}$$

and thus we can conclude that $\nabla\{A, \perp\} \leq \nabla\{B, C, \top\}$.

The axiom $(\nabla 2_f)$ expresses the most fundamental property of ∇ : every conjunction of ∇ -formulas can be (equivalently) written as a disjunction of ∇ -formulas. Ultimately this implies that any formula in \mathcal{L} is equivalent to a formula that does not contain conjunctions.

Example 5.24. 1. Consider the functor $T = C \times _$ from Example 5.22 and consider the set $\Gamma = \{(c, A_1), \dots, (c, A_n)\} \in \mathcal{P}_\omega T_\omega \mathcal{L}^T$. In this case we have $\text{SRD}(\Gamma) = \{(c, \{A_1, \dots, A_n\})\}$ and axiom $(\nabla 2_f)$ yields

$$\bigwedge \{\nabla(c, A_i) \mid (c, A_i) \in \Gamma\} \leq \nabla \left(c, \bigwedge \{A_1, \dots, A_n\} \right).$$

2. Let $T = \mathcal{P}$ and consider the set $\Gamma = \{\{A_1, A_2\}, \{B\}\} \subseteq T_\omega \mathcal{L}^T$. In this case

$$\begin{aligned} \text{SRD}(\Gamma) = \{ & \Phi \mid \Phi \subseteq \{\{A_1, A_2, B\}, \{A_1, B\}, \{A_2, B\}\} \text{ and } \{A_1, A_2, B\} \in \Phi \\ & \text{or } \{A_1, B\}, \{A_2, B\} \in \Phi \} \end{aligned}$$

and we get

$$\begin{aligned} \nabla\{A_1, A_2\} \wedge \nabla\{B\} \leq & \nabla\{\bigwedge\{A_1, A_2, B\}\} \vee \nabla\{\bigwedge\{A_1, B\}, \bigwedge\{A_2, B\}\} \vee \\ & \nabla\{\bigwedge\{A_1, A_2, B\}, \bigwedge\{A_1, B\}\} \vee \nabla\{\bigwedge\{A_1, A_2, B\}, \bigwedge\{A_1, B\}\} \vee \\ & \nabla\{\bigwedge\{A_1, A_2, B\}, \bigwedge\{A_1, B\}, \bigwedge\{A_2, B\}\} \end{aligned}$$

as an instance of axiom $(\nabla 2_f)$.

Finally, axiom $(\nabla 3_f)$ says that the ∇ -operator can be distributed over disjunctions.

Example 5.25. Let $T = \mathcal{P}$ be the power set functor and let $A_1, A_2, A_3 \in \mathcal{L}^{\mathcal{P}}$ be formulas. An instance of $(\nabla 3_f)$ looks as follows: $\nabla\{\bigvee\{A_1, A_2\}, \bigvee\{A_3\}\} \leq \nabla\{A_1, A_3\} \vee \nabla\{A_2, A_3\} \vee \nabla\{A_1, A_2, A_3\}$.

We will not be very precise in our notion of derivability. Roughly speaking, a derivation of some inequality $A \leq B$ is a finite tree whose root is labelled with $A \leq B$ and in which each parent node is the conclusion of a rule and the children of the same node are the premises of that rule.

Definition 5.26. Fix a sound and complete set \mathcal{R} of axioms and deduction rules for propositional logic. We write $\vdash A \leq B$ and say $A \leq B$ is *derivable* if there is a proof of $A \leq B$ using the rules and axioms in \mathcal{R} together with $(\nabla 1)$, $(\nabla 2_f)$ and $(\nabla 3_f)$.

Under this notion of derivability we can show that the axioms $(\nabla 2_f)$ and $(\nabla 3_f)$ are in fact equivalences. Soundness and completeness now take the following form:

Theorem 5.27. For all formulas $A, B \in \mathcal{L}^T$ we have

$$\vdash A \leq B \quad \text{iff} \quad A \models B$$

where $A \models B$ denotes containment of truth sets (Definition 3.8).

The completeness proof in [57] uses a stratification of the logic. Any inequality $\vdash A \leq B$ with $A, B \in \mathcal{L}_n$ can be derived with a derivation that contains only formulas of modal depth at most n . Soundness and completeness of the derivation system restricted to formulas of modal depth 0 is then an immediate consequence of soundness and completeness of \mathcal{R} . Furthermore, soundness and completeness for formulas of modal depth at most n can be lifted to soundness and completeness for formulas of modal depth at most $n + 1$.

Essential for the formulation of this lifting of soundness and completeness is the abstract view on coalgebraic logic that we discussed in Section 3.3. Instead of the functor $M : \mathbf{BA} \rightarrow \mathbf{BA}$ from Example 3.11 we consider the functor $\mathbb{M} : \mathbf{BA} \rightarrow \mathbf{BA}$ that can be seen as a quotient of M by the rules and axioms of the ∇ -logic. Let B_n denote the Boolean algebra obtained by considering the collection of formulas of depth at most n modulo derivable equivalence. One can show that $B_n \cong \mathbb{M}^n \mathbb{2}$, where $\mathbb{2}$ denotes the two-element Boolean algebra. The existence of an injective natural transformation $\delta : \mathbb{M}\mathbf{P} \rightarrow \mathbf{P}T^{\text{op}}$, where $\mathbf{P} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{BA}$ is again the contravariant power set functor, is sufficient for proving soundness and completeness of the logic. The details of the proof can be found in [55].

The first complete set of deduction rules and axioms for a ∇ -modality was presented in [74] for the case $T = \mathcal{P}$. This axiomatisation contained in total 7 rules and axioms. The formulation of the deduction system in [74] used the specific structure of formulas in $\mathcal{L}^{\mathcal{P}}$ and therefore it is not possible to directly use the rules and axioms from [74] also for the axiomatisation of \mathcal{L}^T for set functors T different from the power set functor.

A complete set of rules for the power set functor that is in line with the rules and axioms above is presented in [12] where the authors also devise a cut-free complete Gentzen system for the coalgebraic ∇ -modality for \mathcal{P} . The authors of [12] have recently devised a cut-free complete Gentzen system for the general case which is, however, not yet published at the time of writing of this article.

5.3 The Abstract Approach: Semantic Conditions on Completeness

Compared to concrete representations of coalgebraic logics, there is considerably fewer work on conditions that guarantee completeness for coalgebraic logics presented in an abstract way. Given that abstract logics are given by a pair (L, δ) where $L : \mathbf{BA} \rightarrow \mathbf{BA}$ and $\delta : \mathbf{LP} \rightarrow \mathbf{P}T^{\text{op}}$ is natural, there is even no direct notion of completeness: this relies on a *representation* of the endofunctor L which gives rise to a concrete syntax. In brief, a presentation of L amounts to specifying LA by means of generators G_A and relations R_A that depend functorially on A [62, Definition 3.3]. This allows to

define an equational logic, given by a presentation of L , on top of the equational theory of Boolean algebras. In this setting, Theorem 6.15 of *op.cit.* establishes the following:

Theorem 5.28. *Suppose (L, δ) is an abstract logic for coalgebras and suppose δ is injective. Then the equational theory induced by any presentation of L is complete w.r.t. T -coalgebras.*

The above theorem can be formulated in terms of a contravariant adjunction between the category of sets and Boolean algebras. One obtains a much tighter fit if this adjunction is in fact an equivalence, in other words, the syntactical presentation is an exact mirror image of the semantics. This is the goal of domain theory in logical form [1] which can also be fruitfully applied in a coalgebraic context. The first paper in this area was [56] which is based on the dual equivalence between Boolean algebras and Stone spaces. In particular, *op.cit.* argues that coalgebras over the category of Stone spaces are a natural semantics for modal languages where soundness and completeness are a basic consequence of duality. Technically, [56] exploits a duality between algebras over Boolean algebras and coalgebras over Stone spaces to show that descriptive general frames are in fact nothing but coalgebras for the Vietoris functor (over Stone spaces). In a more general context, [14] studies modal logics for transition systems that are modelled as coalgebras and presents sound and complete axiomatisations over several base categories.

The same approach has also been instantiated over other base categories. Applied to categories of presheaves, the denotational semantics of the π -calculus over bifinite domains is dualised to obtain a sound and complete logic for the π -calculus [15]. Apart from the abstract theory, all that is required is to give a syntactical presentation of the endofunctor that defines a coalgebraic semantics of the calculus.

We note that the abstract approach to completeness for coalgebraic logic is mainly concerned with completeness for *equational* theories of modal logic. This arises naturally via the algebra/coalgebra duality, where properties of coalgebras correspond to algebraic terms that induce equational reasoning. It is currently an open problem to relate the abstract approach to coalgebraic logics to different reasoning styles such as sequent systems.

6 Coalgebraic Fixpoint Logics

In this section we are going to discuss extensions of coalgebraic logics with fixpoint operators. These extensions generalise the modal μ -calculus in the same way as coalgebraic logics without fixpoint operators generalise basic modal logic.

The formulas of all coalgebraic logics discussed so far can only speak about a *finite* number of subsequent transitions and the ensuing states. In application scenarios it is on the other hand often required to be able to formalise assertions the entire (infinite) evolution of a system from a given state. This can be achieved by extending the basic logical language with *fixpoint operators*. Given that the modal operators are monotone, a formula $A(p)$ where p occurs only under an even number of negations defines a monotone operator $M_A : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ on the set of states of a model with carrier W by mapping a subset $S \subseteq W$ to the truth-set of $A(p)$ where the extension of the variable p is taken to be S .

Every (monotone) modal logic gives rise to an associated μ -calculus that arises by adding formulas $\mu x.A$ and $\nu x.A$ to express the least and greatest fixpoint of the operator M_A . In this setting, both the least and greatest fixpoint of M_A conveys information about the infinite evolution of a system: typically, least fixpoints express safety properties, as their violation can be detected after finitely

many transition steps, whereas greatest fixpoints express liveness. In Lamport’s words, safety properties assert that “something bad never happens” [66] and naturally make assertions about the entire evolution of a process.

Fixpoint Logics via Predicate Liftings

While the extension of the modal logic K with fixpoint operators is well studied [17] there is hardly any literature that deals with the extension of other modal logics with fixpoint operators. In the coalgebraic setting, it turns out that fixpoint operators can be added to the syntax in an orthogonal way: it is precisely the same coherence conditions (Definition 5.9) that guarantee soundness and completeness of a logical calculus. As for reasoning under global assumptions, it is more convenient to use tableaux rather than a sequent calculus. To accommodate fixpoint formulas, we extend this tableau calculus with unfolding rules for the fixpoint operators. Since unfolding does not distinguish between least and greatest fixpoints, we need to make sure that – for a satisfiable formula – outermost least fixpoints are not unfolded infinitely often. This gives rise to a parity game [40] where the priorities reflect both nesting of fixpoint operators and distinguish between least and greatest fixpoints. This leads to a global condition that defines closed tableaux: every leaf of a closed tableau is an axiom, and on every infinite path an outermost least fixpoint is unfolded infinitely often.

Given a set Λ of modal operators, we deal with formulas in negation normal form, given by the grammar

$$\mathcal{F}(\Lambda) \ni A, B ::= p \mid \bar{p} \mid A \wedge B \mid A \vee B \mid \heartsuit(A_1, \dots, A_n) \mid \bar{\heartsuit}(A_1, \dots, A_n) \mid \mu p.A \mid \nu p.A$$

where $\heartsuit \in \Lambda$ is an n -ary operator and we require that \bar{p} does not occur in A in the last two clauses. As usual, \bar{p} denotes the negation of the propositional variable p , $\bar{\heartsuit}$ denotes the dual modal operator for each $\heartsuit \in \Lambda$ and $\mu p.A$ and $\nu p.A$ denote the least and greatest fixpoint of the operator M_A discussed above which would fail to be monotone if \bar{p} were to occur in A .

For the semantics, we need to restrict to *monotone* Λ -structures, where a Λ -structure T with associated predicate liftings $\llbracket \heartsuit \rrbracket$ is monotone if every component $\llbracket \heartsuit \rrbracket_X : \mathcal{P}(X)^n \rightarrow \mathcal{P}(TX)$ of the natural transformation $\llbracket \heartsuit \rrbracket$ is monotone in each component. The formal semantics is as expected, and given a T -model $M = (W, \gamma, \pi)$ we have

$$\llbracket p \rrbracket = \pi(p) \quad \llbracket \bar{p} \rrbracket_M = W \setminus \pi(p) \quad \llbracket \bar{\heartsuit}(A_1, \dots, A_n) \rrbracket_M = TW \setminus \llbracket \heartsuit \rrbracket_M(W \setminus A_1, \dots, W \setminus A_n)$$

and for the fixpoint formulas we have

$$\llbracket \mu p.A \rrbracket_M = \text{LFP}(M_A) \quad \llbracket \nu p.A \rrbracket_M = \text{GFP}(M_A)$$

where $\text{LFP}(M_A)$ and $\text{GFP}(M_A)$ denote the least and greatest fixpoint of the operator $M_A : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ given by $M_A(X) = \llbracket A \rrbracket_{M'}$ for $M' = (W, \gamma, \pi')$ where $\pi'(p) = X$ and $\pi'(q) = \pi(q)$ for $q \neq p$. As before we write $M, w \models A$ if $w \in \llbracket A \rrbracket_M$.

Crucially, the satisfaction relation $w \models A$ can be characterised in terms of a two-player parity game [40] that can be seen as a generalisation of the model checking game [98] to the coalgebraic context. This game is played on model/formula pairs where the priority of a position is determined by the formula: the unfolding of a least (resp. greatest) fixpoint attracts an odd (resp. even) priority, and the priority of a fixpoint unfolding increases with the height of the corresponding operator in the construction tree of a formula. Validity of a formula at a point can then be characterised by the parity condition that ensures that no outermost least fixpoint is unfolded infinitely often (we refer to

[24] for details). This game-theoretic characterisation is the basis of a characterisation of satisfiability in terms of a tableau calculus. This calculus is build from the usual propositional tableau rules

$$\frac{p, \bar{p}, \Gamma}{\quad} \quad \frac{A \wedge B, \Gamma}{A, B, \Gamma} \quad \frac{A \vee B, \Gamma}{A, \Gamma \quad B, \Gamma} \quad \frac{\eta p. A, \Gamma}{A[p := \eta p. A], \Gamma}$$

together with unfolding rules for least and greatest fixpoints where $\eta \in \{\mu, \nu\}$ along with a set \mathcal{R} of Λ -rules. To characterise satisfiability, we need to insist, as in the model checking game, that outermost least fixpoints are not unfolded infinitely often. As a consequence we need to impose a global condition on tableaux to characterise satisfiability. Here, we characterise *unsatisfiability* of the root formula in terms of the existence of a *closed* tableau, where on every infinite path at least one outermost fixpoint has to be unfolded infinitely often. As in the model checking game, this condition is defined in terms of a (the same) parity function. In contrast to the model checking game, tableaux are defined in terms of sequents (sets of formulas), and the condition on infinite paths requires to trace the evolution of single formulas that occur in tableau sequents. We refer to [24] for full details of the following:

Theorem 6.1. *Suppose that the set \mathcal{G} of sequent rules is one-step sound and one-step cut-free complete with respect to a given Λ -structure T . Then \mathcal{G} induces a tableau calculus with the property that a formula $A \in \mathcal{F}(\Lambda)$ is satisfiable in $\text{Mod}(T)$ if and only if there is no closed tableau with root A .*

In order to determine satisfiability, one therefore has to check for the existence of closed tableaux, which in turn can be formulated in terms of a parity game. Crucially, in the construction of the tableau we need to check the condition on traces. This is achieved by constructing the tableau and checking the condition on traces by a word automaton at the same time. The ensuing game, where now priorities are determined by the parity function of the word automaton, is again a parity game. Provided that the set \mathcal{G} of Λ -rules can be decided in exponential time, this gives an overall EXPTIME complexity bound for the existence of closed tableau, and therefore for the satisfiability problem.

Theorem 6.2. *Suppose that a given set \mathcal{G} of Λ -rules can be decided in EXPTIME. Then the problem of deciding the existence of a closed tableau with root $A \in \mathcal{F}(\Lambda)$ is in EXPTIME.*

All monotone Λ -structures presented in Section 2 allow for a one-step complete axiomatisation in terms of one-step rules that are decidable in EXPTIME, which immediately gives rise to complexity bounds for various fixpoint logics, such as fixpoint extensions of coalition logic, monotone modal logic, graded modal logic and – of course – the modal μ -calculus. We conclude the section on the coalgebraic μ -calculus by mentioning closely related work by Fontaine, Leal and Venema [32] in which the authors devise automata which can be used to obtain Theorem 6.2 and a general small model property theorem for the coalgebraic μ -calculus.

Automata and fixpoint extensions of the ∇ -logic

An important area in which variants of the ∇ -operator have been used is the area of modal fixpoint logics, in particular, the modal μ -calculus. The μ -automata by Janin & Walukiewicz [51] and the tableaux for the modal μ -calculus in [104] both are tailored for logics with ∇ -operator rather than for modal logic in the standard syntax using \square 's and \diamond 's. This observation led Venema [102] to the development of a coalgebraic fixpoint logic that uses the cover modality. The language of this logic consists of the coalgebraic ∇ -language *without negation* enriched by fixpoint variables and least and greatest fixpoint operators.

Definition 6.3. Let T be a set functor. The collection of formulas of Venema’s coalgebraic fixpoint language is defined as follows:

$$\mu\mathcal{L}_T^\mathcal{V} \ni A ::= p \in \mathcal{V} \mid A \wedge A \mid A \vee A \mid \nabla\alpha, \alpha \in T_\omega(\mu\mathcal{L}_T^\mathcal{V}) \mid \mu p.A, p \in \mathcal{V} \mid \nu p.A, p \in \mathcal{V},$$

where \mathcal{V} is a set of fixpoint variables. We write $\mu\mathcal{L}_T$ for the collection of sentences, ie., formulas without free occurrences of fixpoint variables.

Remark 6.4. The variables occurring in the language $\mu\mathcal{L}_T^\mathcal{V}$ are not propositional variables but fixpoint variables and ultimately we are only interested in the sentences of the language. Therefore, as a fixpoint variable only can occur positively in a formula, Definition 6.3 does not contain any negated variables. This is different from the definition of the coalgebraic μ -calculus using predicate liftings, that includes (possibly negated) propositional variables.

The semantics of a formula is defined as for the language without fixpoint operators.

Definition 6.5. Let T be a set functor and let (X, γ) be a T -coalgebra. For any valuation $h : \mathcal{V} \rightarrow \mathcal{P}(X)$ we define the satisfaction relation \models_h for the modal part of the language as before, adding the obvious clause for fixpoint variables that $x \models_h p$ if $x \in h(p)$, and we let $\llbracket A \rrbracket_h = \{x \in X \mid x \models_h A\}$. Furthermore for any formula A and any variable $p \in \mathcal{V}$ we define an operator

$$\begin{aligned} \llbracket A \rrbracket_h^p : \mathcal{P}(X) &\rightarrow \mathcal{P}(X) \\ U &\mapsto \llbracket A \rrbracket_{h[p \mapsto U]} \end{aligned} \quad \text{where} \quad h[p \mapsto U](q) = \begin{cases} U & \text{if } p = q \\ h(q) & \text{otherwise.} \end{cases}$$

and we put $\llbracket \mu p.A \rrbracket_h = \text{LFP}(\llbracket A \rrbracket_h^p)$ and $\llbracket \nu p.A \rrbracket_h = \text{GFP}(\llbracket A \rrbracket_h^p)$.

In the remainder of this section we are going to have a closer look at coalgebra automata. Coalgebra automata are expressively equivalent to coalgebraic fixpoint logics (cf. Thm. 6.9 below for a precise statement) and, as we will see, this equivalence can be used in order to study interesting properties of coalgebraic fixpoint logics. But coalgebra automata provide not only a tool for coalgebraic fixpoint logics. In addition to that, they also provide a general framework in which important results in automata theory can be proven in a uniform way.

Definition 6.6. An alternating T -coalgebra automaton is a quadruple $\mathbb{A} = (A, \Delta, a_I, \Omega)$ where A is a finite set of states, $\Delta : A \rightarrow \mathcal{P}\mathcal{P}(TA)$ is a transition function, $a_I \in A$ is the initial state and $\Omega : A \rightarrow \omega$ is a function (the parity function). We call \mathbb{A} non-deterministic if $\Delta(a)$ contains only singleton sets for all $a \in A$ and we call \mathbb{A} deterministic if $\Delta(a)$ contains precisely one singleton set. For a non-deterministic automaton we consider the transition function to be of type $\Delta : A \rightarrow \mathcal{P}(TA)$ and, similarly, for a deterministic automaton we consider the transition function to be of type $\Delta : A \rightarrow TA$.

Coalgebra automata for a set functor T should be thought of as devices that accept or reject pointed T -coalgebras, ie., pairs (\mathbb{X}, x) where $\mathbb{X} = (X, \gamma)$ is a T -coalgebra and $x \in X$. Acceptance is formulated in terms of a parity graph game. In order to understand the type of the transition function Δ , one should think of Δ mapping any state a of the automaton to a lattice expression of the form $\bigvee_{\Phi \in \Delta(a)} \bigwedge_{t \in \Phi} t$, where the t ’s represent requirements (or formulas) that the successor of a given state has to fulfil. The outer occurrence of \mathcal{P} represents a disjunction and the inner occurrence of \mathcal{P} represents a conjunction over these requirements. This leads to a definition of acceptance as a parity game [102], played by \exists and \forall where accepting a pointed coalgebra amounts to \exists having a winning strategy.

Definition 6.7. Let T be a standard and weak pullback preserving functor, let $\mathbb{X} = (X, \gamma)$ be a T -coalgebra and let $\mathbb{A} = (A, \Delta, a_I, \Omega)$ be an alternating T -coalgebra automaton. The game board of the acceptance game $\mathcal{G}(\mathbb{X}, \mathbb{A})$ defined as follows:

Position: b	$P(b)$	Admissible moves: $E[b]$	$\overline{\Omega}(b)$
$(x, a) \in X \times A$	\exists	$\{(x, \Phi) \in X \times \mathcal{P}(TA) \mid \Phi \in \Delta(a)\}$	$\Omega(a)$
$(x, \Phi) \in X \times \mathcal{P}(TA)$	\forall	$\{(x, \phi) \in X \times TA \mid \phi \in \Phi\}$	0
$(x, \phi) \in X \times TA$	\exists	$\{Z \in \mathcal{P}(X \times A) \mid (\gamma(x), \phi) \in \overline{T}Z\}$	0
$Z \in \mathcal{P}(X \times A)$	\forall	Z	0

In other words $\mathcal{G}(\mathbb{X}, \mathbb{A})$ is a parity graph game where \exists 's positions are defined as $B_{\exists} = (X \times A) \cup (X \times TA)$, \forall 's positions are given by $B_{\forall} = X \times \mathcal{P}(TA) \cup \mathcal{P}(X \times A)$ and the edge relation E and parity function $\overline{\Omega}$ are defined in the above table. The acceptance game of a non-deterministic and deterministic automaton is obtained by removing the first choice of \forall or both the first choices of \exists and \forall , respectively. We say that \mathbb{A} accepts a given pointed coalgebra (\mathbb{X}, x) if player \exists has a winning strategy at position (x, a_I) in $\mathcal{G}(\mathbb{X}, \mathbb{A})$. The language $L(\mathbb{A})$ accepted by \mathbb{A} is given by $L(\mathbb{A}) = \{(\mathbb{X}, x) \in \text{PCoalg}(T) \mid \mathbb{A} \text{ accepts } (\mathbb{X}, x)\}$, where $\text{PCoalg}(T)$ denotes the collection of pointed T -coalgebras.

This notion of automaton is a natural generalization of the standard notions of automata operating on structures with (possibly) infinite behaviour.

Example 6.8. 1. Consider the functor $T = C \times _ \times _$ for some set C . Then T -coalgebra automata correspond to parity automata that operate on C -labelled binary trees (cf. [40] for a definition). These automata are the key tool for proving Rabin's theorem [80] stating that the monadic second-order theory **S2S** of infinite binary trees is decidable.

2. $T = \mathcal{P}(\mathcal{P}) \times \mathcal{P}$ where \mathcal{P} is a set of proposition letters. In this case T -automata are exactly the modal μ -automata from [51] that are expressively equivalent to the modal μ -calculus. These automata constitute a key ingredient of Walukiewicz's proof in [104] that Kozen's axiomatisation of the modal μ -calculus is complete.

3. $T = \mathcal{P}(\mathcal{P}) \times \mathcal{D}$ where \mathcal{P} is a set of proposition letters. In this case T -automata operate on discrete Markov chains with finite support whose states are labelled with sets of propositional variables. The coalgebraic definition is to the best of our knowledge the first definition for automata that operate on Markov chains, T -automata are expressively equivalent to a probabilistic μ -calculus. Recent research on probabilistic fixpoint logics focuses, however, on the more complex path-based logics PCTL (cf. [42] and cf. [23] for a coalgebraic perspective) for which a more sophisticated automaton model is required [47].

As mentioned earlier on, the importance of coalgebra automata from a logical perspective lies in the fact that these automata are expressively equivalent to Venema's coalgebraic fixpoint logic [102].

Theorem 6.9. *Let T be a standard, weak pullback preserving set functor.*

1. *Every formula $A \in \mu\mathcal{L}_T$ can be transformed into a T -coalgebra automaton \mathbb{A}_A such that for any pointed T -coalgebra (\mathbb{X}, x) we have $(\mathbb{X}, x) \in L(\mathbb{A}_A)$ iff $x \models A$.*

2. *If T maps finite sets to finite sets then every T -coalgebra automaton \mathbb{A} can be transformed into a formula $A_{\mathbb{A}} \in \mu\mathcal{L}_T$ such that for any pointed T -coalgebra (\mathbb{X}, x) we have $x \models A_{\mathbb{A}}$ iff $(\mathbb{X}, x) \in L(\mathbb{A})$.*

Remark 6.10. The restriction to functors mapping finite sets to finite sets in the second half of the theorem could be dropped if we require the transition function of a T -automaton to be of type $\Delta : A \rightarrow \mathcal{P}_\omega \mathcal{P}_\omega T A$ (this ensures that every player has only a finite number of choices even if TA is infinite). It is however unclear whether this change of transition function would possibly invalidate Theorem 6.13 below in the case of functors that do *not* preserve finite sets.

Intuitively speaking, automata and formulas can be seen as different, equivalent representations of the same concepts. Coalgebra automata enjoy several closure properties [59, 52] that form the basis of the correspondence between automata and fixpoint formulas.

Theorem 6.11. *Let T be a standard, weak pullback preserving set functor.*

1. *For any two T -coalgebra automata \mathbb{A}_1 and \mathbb{A}_2 we can construct automata \mathbb{A}_\cup and \mathbb{A}_\cap such that $L(\mathbb{A}_\cup) = L(\mathbb{A}_1) \cup L(\mathbb{A}_2)$ and $L(\mathbb{A}_\cap) = L(\mathbb{A}_1) \cap L(\mathbb{A}_2)$.*

2. *Let C be a set (of colours). For any $C \times T$ -coalgebra automaton \mathbb{A} we can construct a T -coalgebra automaton $\pi_C \mathbb{A}$ with*

$$L(\pi_C \mathbb{A}) = \{((X, \gamma), x) \in \text{PCoalg}(T) \mid \exists ((Y, \langle \sigma, \delta \rangle), y) \in L(\mathbb{A}) \text{ such that} \\ ((X, \gamma), x) \text{ and } ((Y, \delta), y) \text{ are behaviourally equivalent}\}$$

3. *Let T be a functor that maps finite sets to finite sets. For every T -coalgebra automaton \mathbb{A} we can construct a T -coalgebra automaton \mathbb{A}^c such that $L(\mathbb{A}^c) = \text{PCoalg}(T) \setminus L(\mathbb{A})$.*

Remark 6.12. For set functors that map finite sets to finite sets, item 1 of Theorem 6.11 can be obtained as a rather trivial corollary of Theorem 6.9. Item 1 has, however, also a simple automata-theoretic proof that does not use any detour via the equivalence between formulas and automata and that works for any weak pullback preserving set functor.

The closure properties from the previous theorem form the basis for the correspondence between formulas and automata. This is obvious for closure under union, intersection and complement as these operations have their logical counterpart. The second closure property, closure under “projection”, however, seems to have no obvious logical correspondent. It can be shown that the projection operator captures the semantics of the so-called bisimulation quantifiers from [103, 27]. The fact that the correspondence between formulas and coalgebra automata can be obtained without adding bisimulation quantifiers explicitly to the language of coalgebraic fixpoint logic means that those quantifiers are already definable implicitly in the logic. In other words: coalgebraic fixpoint logics are closed under bisimulation quantifiers. This fact can be used to establish that Venema’s coalgebraic fixpoint logic enjoys the so-called *uniform* interpolation property. This has been observed by Venema but has not been published to this day. Note that even for modal logics without fixpoint operators the cover modality is useful for proving uniform interpolation [19].

An important motivation for establishing an equivalence between coalgebra automata and formulas of coalgebraic fixpoint logic is to obtain a decision procedure for coalgebraic fixpoint logics: In order to check whether or not a given formula A is satisfiable, it suffices to check whether the language of the corresponding automaton $L(\mathbb{A}_A)$ is non-empty. The non-emptiness test, however, only works for non-deterministic coalgebra automata. Therefore the following theorem [59] is crucial.

Theorem 6.13. *Let T be a standard, weak pullback preserving functor. For any alternating automaton \mathbb{A} we can construct an equivalent non-deterministic automaton \mathbb{A}^{nd} with $L(\mathbb{A}) = L(\mathbb{A}^{\text{nd}})$.*

The key role in the coalgebraic part of the construction is played by redistributions and the distributive law $(\nabla 2_f)$ from the ∇ -logic that we discussed in Section 5.2 because this law is the key that allows us to rearrange the order in which the players in the acceptance game are moving. The automata-theoretic core of the construction is the well-known Safra construction that is used to determinise ω -word automata: the set of states of the new automaton \mathbb{A}^{nd} are binary relations over the set of states of the original automaton \mathbb{A} and we have to construct a deterministic ω -word parity automaton that recognises those infinite sequence of relations that do only contain sequences of states of \mathbb{A} that satisfy \mathbb{A} 's parity condition. One way of summarising the proof of [59] is to say that it reduces the problem of non-determinising a T -coalgebra automaton for an arbitrary transition type T to the problem of determinising an ω -word automaton. The complexity of the construction of a non-deterministic coalgebra automaton is given by the complexity of the Safra construction [59].

Theorem 6.14. *Let $\mathbb{A} = (A, \Delta, a_I, \Omega)$ be a non-deterministic T -coalgebra automaton. Then $L(\mathbb{A}) \neq \emptyset$ iff there exists a T -coalgebra $(A, \delta : A \rightarrow TA)$ with $\delta(a) \in \Delta(a)$ for all $a \in A$ such that $((A, \delta), a_I) \in L(\mathbb{A})$.*

This important result provides a powerful tool for deciding satisfiability of a given formula of coalgebraic fixpoint logic: Let $A \in \mu\mathcal{L}_T$ be a formula and let \mathbb{A}_A be the corresponding T -coalgebra automaton according to Theorem 6.9. By Theorem 6.13 we may assume that \mathbb{A}_A is nondeterministic. In order to check whether A is satisfiable it suffices to check whether a is satisfiable on a pointed T -coalgebra that can be constructed from the automaton \mathbb{A}_A as described in Theorem 6.14. This will be in many cases decidable, in particular, if we assume that the functor T maps finite sets to finite sets.

Corollary 6.15. *Let T be a standard and weak pullback preserving set functor that maps finite sets to finite sets. Then for any formula of coalgebraic fixpoint logic $A \in \mu\mathcal{L}^T$ it is decidable whether or not A is satisfiable in some T -coalgebra.*

More concretely, Corollary 6.15 can be used to obtain a number of decidability results in a uniform manner, e.g. decidability of the modal μ -calculus (for $T = \mathcal{P}(\mathcal{P}) \times \mathcal{P}_.$) and of **S2S** (for $T = C \times _. \times _.$).

7 Conclusions

In order to keep the size of our article within certain boundaries we had to confine ourselves to providing only the basic facts on coalgebraic modal logics in detail. Over the last decade the theory of coalgebra and logic has seen a rapid development and it will come as no surprise that we had to make many omissions.

The most fundamental choice we made was to only present coalgebraic *modal* logics. This means that we did not discuss the coequational logic from [2, 94] and also not the line of research into coalgebraic logics by Goldblatt [34, 35, 33] that has a distinctive equational flavour. We also did not discuss the recent developments on Kleene coalgebra [96]. Compared to coalgebraic modal logic the work on Kleene coalgebra is not so much about the specification of certain aspects of the behaviour of a given coalgebra but about the complete specification of a transition system using algebraic terms. Nevertheless there is a close connection between Kleene coalgebra and this connection is also the topic of active research.

Inside the field of coalgebraic modal logic we also made several omissions: The inductively defined logics going back to the works by Jacobs & Rößiger [81, 49] influenced the more recent work on modularity [22, 25, 91]: both proof systems and decision procedures can be derived compositionally from the components of combined modal logics.

We represented the predicate lifting approach and the relation lifting approach but we did not mention the work on translations between the two families of coalgebraic modal logics [67, 60]. The work on coalgebraic fixpoint logic lead to a general completeness result for flat coalgebraic fixpoint logics [93]. Finally, there are many results from the theory of basic modal logic that have been generalised to the coalgebraic level such as van Benthem’s characterisation of modal logic as the bisimulation invariant fragment of first-order logic [90], a Lindström theorem [64] and the Goldblatt-Thomasson theorem [63].

We only mentioned logics for coalgebras over the category of sets. There are, however, many interesting examples of logics for coalgebras over other base categories, eg. the category of measure spaces [69], the category of Stone spaces [56]. One more area that we skipped concerns logics that reason about traces [43] that seem to be of a particular interest in connection with probabilistic fixpoint logics [23]. To put in all in a nutshell, coalgebraic logic is a very active research field and this short note can only offer the reader a first impression of its basics.

Acknowledgments.

We thank our colleagues in the field for many lively discussions and the editors of this volume for their continued patience.

References

- [1] S. Abramsky. Domain Theory in Logical Form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.
- [2] J. Adámek. A logic of coequations. In C.-H. L. Ong, editor, *CSL*, volume 3634 of *Lecture Notes in Computer Science*, pages 70–86. Springer, 2005.
- [3] J. Adámek and V. Trnková. *Automata and Algebras in Categories*. Kluwer Academic Publishers, 1990.
- [4] C. Areces and R. Goldblatt, editors. *Advances in Modal Logic 7, papers from the seventh conference on "Advances in Modal Logic," held in Nancy (France) in September 2008*. College Publications, 2008.
- [5] C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*. Elsevier, 2006.
- [6] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [7] M. Barr. Relational Algebras. In *Reports of the Midwest Category Seminar IV*, volume 137 of *Lecture Notes in Mathematics*, pages 39–55, 1970.
- [8] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114:299–315, 1993.
- [9] F. Bartels, A. Sokolova, and E. P. de Vink. A hierarchy of probabilistic system types. *Theoretical Computer Science*, 327(1-2):3–22, 2004.

- [10] L. Beklemishev, V. Goranko, and V. Shehtman, editors. *Advances in Modal Logic 8, papers from the eighth conference on "Advances in Modal logic," held in Moscow (Russia) in October 2002*. College Publications, 2010.
- [11] J. van Benthem. Two simple incomplete modal logics. *Theoria*, 44:25–37, 1978.
- [12] M. Bílková, A. Palmigiano, and Y. Venema. Proof systems for the coalgebraic cover modality. In Areces and Goldblatt [4], pages 1–21.
- [13] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [14] M. Bonsangue and A. Kurz. Duality for logics for transition systems. In V. Sassone, editor, *Proc. FoSSaCS 2005*, number 3441 in Lect. Notes in Comp. Sci., pages 455–469, 2005.
- [15] M. Bonsangue and A. Kurz. Pi-calculus in logical form. In *Proc. 22nd IEEE Symposium on Logic in Computer Science*, pages 303–312, 2007.
- [16] G. Boolos and G. Sambin. An incomplete system of modal logic. *Journal of Philosophical Logic*, 14(4):351–358, 1985.
- [17] J. Bradfield and C. Stirling. Modal μ -calculi. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *The Handbook of Modal Logic*, pages 721–756. Elsevier, 2006.
- [18] F. D. Caro. Graded modalities II (canonical models). *Studia Logica*, 47:1–10, 1988.
- [19] B. ten Cate, W. Conradie, M. Marx, and Y. Venema. Definitorially complete description logics. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *KR*, pages 79–89. AAAI Press, 2006.
- [20] A. Chandra and L. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.
- [21] B. Chellas. *Modal Logic*. Cambridge University Press, 1980.
- [22] C. Cîrstea. A compositional approach to defining logics for coalgebras. *Theoretical Computer Science*, 327:45–69, 2004.
- [23] C. Cîrstea. Generic infinite traces and path-based coalgebraic temporal logics. In B. Jacobs, M. Niqui, J. Rutten, and A. Silva, editors, *Proceedings of the Tenth Workshop on Coalgebraic Methods in Computer Science (CMCS 2010)*, volume 264 of *ENTCS*, pages 83–103, 2010.
- [24] C. Cîrstea, C. Kupke, and D. Pattinson. EXPTIME tableaux for the coalgebraic μ -calculus. In E. Grädel and R. Kahle, editors, *Proc. CSL 2009*, number 5771 in Lect. Notes in Comp. Sci., pages 179–193, 2009.
- [25] C. Cîrstea and D. Pattinson. Modular proof systems for coalgebraic logics. *Theoretical Computer Science*, 388:83–108, 2007.
- [26] M. Cresswell:1984:IDM. An incomplete decidable modal logic. *Journal of Symbolic Logic*, 49(2):520–527, 1984.
- [27] G. d’Agostino and M. Hollenberg. Logical questions concerning the μ -calculus: Interpolation, lyndon and loś-theorem. *The Journal of Symbolic Logic*, 65(1):310–332, 2000.
- [28] G. D’Agostino and A. Visser. Finality regained: A coalgebraic study of Scott-sets and multisets. *Arch. Math. Logic*, 41:267–298, 2002.

- [29] S. Demri and D. Lugiez. Presburger modal logic is only PSPACE-complete. In U. Furbach and N. Shankar, editors, *Proc. IJCAR 2006*, volume 4130 of *Lect. Notes in Artificial Intelligence*, pages 541–556. Springer, 2006.
- [30] R. Fagin and J. Halpern. Reasoning about knowledge and probability. *Journal of the ACM*, 41:340–367, 1994.
- [31] K. Fine. In so many possible worlds. *Notre Dame J. Formal Logic*, 13:516–520, 1972.
- [32] G. Fontaine, R. Leal, and Y. Venema. Automata for coalgebras: An approach using predicate liftings. In S. Abramsky, C. Gavaille, C. Kirchner, F. Meyer auf der Heide, and P. G. Spirakis, editors, *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 381–392. Springer, 2010.
- [33] D. Friggens and R. Goldblatt. A modal proof theory for final polynomial coalgebras. *Theoretical Computer Science*, 360(1-3):1–22, 2006.
- [34] R. Goldblatt. What is the coalgebraic analogue of Birkhoff’s variety theorem? *Theoretical Computer Science*, 266(1-2):853–886, 2001.
- [35] R. Goldblatt. Equational logic of polynomial coalgebras. In P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyashev, editors, *Advances in Modal Logic*, pages 149–184. King’s College Publications, 2003.
- [36] R. Goldblatt. Final coalgebras and the Hennessy-Milner property. *Annals of Pure and Applied Logic*, 138(1-3):77–93, 2006.
- [37] Robert Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1992. Second Edition.
- [38] R. Goré, C. Kupke, D. Pattinson, and L. Schröder. Global caching for coalgebraic description logics. In J. Giesl and R. Hähnle, editors, *IJCAR*, volume 6173 of *Lecture Notes in Computer Science*, pages 46–60. Springer, 2010.
- [39] R. Goré and L. Nguyen. EXPTIME tableaux for ALC using sound global caching. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A.-Y. Turhan, and S. Tessaris, editors, *Proc. Description Logics 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [40] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
- [41] H. H. Hansen, C. Kupke, and E. Pacuit. Neighbourhood structures: Bisimilarity and basic model theory. *Logical Methods in Computer Science*, 5(2), 2009.
- [42] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
- [43] I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Logical Methods in Computer Science*, 3(4), 2007.
- [44] A. Heifetz and P. Mongin. Probabilistic logic for type spaces. *Games and Economic Behavior*, 35:31–53, 2001.

- [45] E. Hemaspaandra. Modal satisfiability is in deterministic linear space. In P. Clote and H. Schwichtenberg, editors, *Proc. CSL 2000*, volume 1862 of *Lecture Notes in Computer Science*, pages 332–342. Springer, 2000.
- [46] M. Hennessy and R. Milner. Algebraic Laws for Nondeterminism and Concurrency. *Journal of the ACM*, 32:137–161, 1985.
- [47] M. Huth, N. Piterman, and D. Wagner. Weak p-automata: Acceptors of Markov chains. In *th International Conference on Quantitative Evaluation of SysTems*, pages 161–170. IEEE Press, 2010.
- [48] B. Jacobs. Introduction to coalgebra. Towards mathematics of states and observations. Two-thirds of a book in preparation. Available from the author’s webpage.
- [49] B. Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications*, 35(1):31–59, 2001.
- [50] B. Jacobs and A. Sokolova. Exemplaric expressivity of modal logics. *Journal of Logic and Computation*, 20(5), 2010.
- [51] D. Janin and I. Walukiewicz. Automata for the Modal μ -Calculus and Related Results. In J. Wiedermann and P. Hájek, editors, *MFCS*, volume 969 of *LNCS*, pages 552–562, 1995.
- [52] C. Kissig and Y. Venema. Complementation of coalgebra automata. In Kurz et al. [61], pages 81–96.
- [53] S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, pages 83–94, 1963.
- [54] C. Kupke, A. Kurz, and D. Pattinson. Algebraic semantics for coalgebraic logics. In J. Adamek and S. Milius, editors, *Coalgebraic Methods in Computer Science (CMCS 2004)*, volume 106 of *Electr. Notes in Theoret. Comp. Sci.*, pages 219–241. Elsevier, 2004.
- [55] C. Kupke, A. Kurz, and Y. Venema. Completeness for the coalgebraic cover modality. *Under submission*.
- [56] C. Kupke, A. Kurz, and Y. Venema. Stone coalgebras. *Theoretical Computer Science*, 327(1-2):109–134, 2004.
- [57] C. Kupke, A. Kurz, and Y. Venema. Completeness of the finitary Moss Logic. In Areces and Goldblatt [4], pages 193–217.
- [58] C. Kupke and R. A. Leal. Characterising behavioural equivalence: Three sides of one coin. In Kurz et al. [61], pages 97–112.
- [59] C. Kupke and Y. Venema. Coalgebraic Automata Theory: Basic Results. *Logical Methods in Computer Science*, 4(4), 2008.
- [60] A. Kurz and R. Leal. Equational Coalgebraic Logic. *Electronic Notes in Theoretical Computer Science*, 249:333–356, 2009.

- [61] A. Kurz, M. Lenisa, and A. Tarlecki, editors. *Algebra and Coalgebra in Computer Science, Third International Conference, CALCO 2009, Udine, Italy, September 7-10, 2009. Proceedings*, volume 5728 of *Lecture Notes in Computer Science*, 2009.
- [62] A. Kurz and D. Petrisan. Presenting functors on many-sorted varieties and applications. *Information and Computation*, 208(12):1421–1446, 2010.
- [63] A. Kurz and J. Rosický. The Goldblatt-Thomason theorem for coalgebras. In T. Mossakowski, U. Montanari, and M. Haverdaen, editors, *CALCO*, volume 4624 of *Lecture Notes in Computer Science*, pages 342–355. Springer, 2007.
- [64] A. Kurz and Y. Venema. Coalgebraic Lindström theorems. In Beklemishev et al. [10], pages 292–309.
- [65] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3), 1977.
- [66] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Trans. Software Eng.*, 3(2):125–143, 1977.
- [67] R. Leal. Predicate liftings versus nabla modalities. *Electronic Notes in Theoretical Computer Science*, 203(5):195–220, 2008.
- [68] R. Montague. Universal grammar. *Theoria*, 36:373–398, 1970.
- [69] L. Moss and I. Viglizzo. Final coalgebras for functors on measurable spaces. *Information and Computation*, 204(4):610–636, 2006.
- [70] L.S. Moss. Coalgebraic Logic. *Annals of Pure and Applied Logic*, 96(1–3):277–317, 1999.
- [71] L.S. Moss. Erratum to “Coalgebraic Logic”. *Annals of Pure and Applied Logic*, 99(1–3):241–259, 1999.
- [72] R. Myers, D. Pattinson, and L. Schröder. Coalgebraic hybrid logic. In L. de Alfaro, editor, *FOSSACS*, volume 5504 of *Lecture Notes in Computer Science*. Springer, 2009.
- [73] E. Pacuit and S. Salame. Majority logic. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Proc. KR 2004*, pages 598–605. AAAI Press, 2004.
- [74] A. Palmigiano and Y. Venema. Nabla Algebras and Chu Spaces. In T. Mossakowski, U. Montanari, and M. Haverdaen, editors, *Proceedings of Calco 2007*, volume 4624 of *LNCS*, pages 394–408, 2007.
- [75] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309:177–193, 2003.
- [76] D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame Journal of Formal Logic*, 45(1):19–33, 2004.
- [77] D. Pattinson and L. Schröder. Beyond rank 1: Algebraic semantics and finite models for coalgebraic logics. In R. Amadio, editor, *Proc. FoSSaCS 2008*, number 4962 in *LNCS*, pages 66–80, 2008.

- [78] D. Pattinson and L. Schröder. Cut elimination in coalgebraic logics. *Information and Computation*, 208(12):1447–1468, 2010.
- [79] M. Pauly. A modal logic for coalitional power in games. *J. Logic Comput.*, 12(1):149–166, 2002.
- [80] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, pages 1–35, 1969.
- [81] M. Rößiger. Coalgebras and modal logic. *Electronic Notes in Theoretical Computer Science*, 33, 2000.
- [82] J.J.M.M. Rutten. Relators and metric bisimulations (extended abstract). In B. Jacobs, L. Moss, H. Reichel, and J.J.M.M. Rutten, editors, *Proceedings of the First Workshop on Coalgebraic Methods in Computer Science (CMCS 1998)*, volume 11 of *ENTCS*, pages 1–7, 1998.
- [83] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.
- [84] L. Santocanale and Y. Venema. Uniform interpolation for monotone modal logic. In Beklemishev et al. [10], pages 350–370.
- [85] L. Schröder. A finite model construction for coalgebraic modal logic. *J. Log. Algebr. Program.*, 73(1-2):97–110, 2007.
- [86] L. Schröder. Expressivity of coalgebraic modal logic: the limits and beyond. *Theoretical Computer Science*, 390(2-3):230–247, 2008.
- [87] L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. In *Proc. 21st IEEE Symposium on Logic in Computer Science (LICS 2006)*, pages 231–242, 2006.
- [88] L. Schröder and D. Pattinson. Shallow models for non-iterative modal logics. In A. Dengel, K. Berns, T. Breuel, F. Bomarius, and T. Roth-Berghofer, editors, *Proc. KI 2008*, volume 5243 of *Lecture Notes in Computer Science*, pages 324–331. Springer, 2008.
- [89] L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. *ACM Transactions on Computational Logics*, 10(2), 2009.
- [90] L. Schröder and D. Pattinson. Coalgebraic correspondence theory. In C.-H. L. Ong, editor, *FOSSACS*, volume 6014 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2010.
- [91] L. Schröder and D. Pattinson. Modular algorithms for heterogeneous modal logics via multi-sorted coalgebra. *Mathematical Structures in Computer Science*, to appear.
- [92] L. Schröder, D. Pattinson, and C. Kupke. Nominals for everyone. In C. Boutilier, editor, *Proc. IJCAI 2009*, pages 917–922, 2009. Online proceedings.
- [93] L. Schröder and Y. Venema. Flat coalgebraic fixed point logics. In P. Gastin and F. Laroussinie, editors, *CONCUR*, volume 6269 of *Lecture Notes in Computer Science*, pages 524–538. Springer, 2010.
- [94] D. Schwencke. Coequational logic for accessible functors. *Information and Computation*, 208(12):1469–1489, 2010.

- [95] D. Scott. Advice in modal logic. In Karel Lambert, editor, *Philosophical Problems in Logic*. Reidel, 1970.
- [96] A. Silva, M. M. Bonsangue, and J. J. M. M. Rutten. Non-deterministic Kleene coalgebras. *Logical Methods in Computer Science*, 6(3), 2010.
- [97] S. Staton. Relating coalgebraic notions of bisimulation. In Kurz et al. [61], pages 191–205.
- [98] C. Stirling. Games and modal mu-calculus. In T. Margaria and B. Steffen, editors, *Proc. TACAS 1996*, volume 1055 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 1996.
- [99] V. Trnková. General theory of relational automata. *Fundamenta Informaticae*, 3(2):189–233, 1980.
- [100] A. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Number 43 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1996.
- [101] Moshe Y. Vardi. Why is modal logic so robustly decidable? In Neil Immerman and Phokion G. Kolaitis, editors, *Descriptive Complexity and Finite Models*, volume 31 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 149–184. AMS, 1996.
- [102] Y. Venema. Automata and fixed point logic: A coalgebraic perspective. *Information and Computation*, 204(4):637–678, 2006.
- [103] A. Visser. Bisimulations, model descriptions and propositional quantifiers. Technical Report 161, Department of Philosophy, Utrecht University, 1996.
- [104] I. Walukiewicz. Completeness of Kozen’s Axiomatisation of the Propositional μ -Calculus. *Information and Computation*, 157(1-2):142–182, 2000.
- [105] J. Worrell. On the final sequence of a finitary set functor. *Theoretical Computer Science*, 338(1-3):184–199, 2005.