2006

# Coalgebras and Their Logics

Alexander Kurz
*Chapman University*, akurz@chapman.edu



Recommended Citation
A. Kurz, “Coalgebras and their logics,” ACM SIGACT News, vol. 37, no. 2, p. 57, Jun. 2006. DOI: 10.1145/1140612.1140628

# SIGACT News Logic Column 15

Riccardo Pucella
Northeastern University
Boston, MA 02115 USA
riccardo@ccs.neu.edu

Some comments about the last Logic Column, on nominal logic. Pierre Lescanne points out that the terminology "de Bruijn levels" was introduced in the paper *Explicit Substitutions with de Bruijn's Levels*, by Pierre Lescanne and Jocelyne Rouyer-Degli, presented at the 1995 RTA conference. He also points out that Stoy diagrams were probably invented by Stoy, but appear in work by Bourbaki as early as 1939 (published in 1954). Merci, Pierre.

That article also initiated what is bound to be an interesting discussion. The critique of higher-order abstract syntax in that article prompted Karl Crary and Robert Harper to prepare a response to the leveled criticisms. The response should appear in an upcoming Column.

In this issue, Alexander Kurz describes recent work on the topic of specifying properties of transition systems. It turns out that by giving a suitably abstract description of transition systems as coalgebras, we can derive logics for capturing properties of these transition systems in a rather elegant way. I will let you read the details below.

I am always looking for contributions. If you have any suggestion concerning the content of the Logic Column, or if you would like to contribute by writing a column yourself, feel free to get in touch with me.

---

# Coalgebras and Their Logics[1]

Alexander Kurz
University of Leicester, UK

## 1 Introduction

Transition systems pervade much of computer science. This article outlines the beginnings of a general theory of specification languages for transition systems. More specifically, transition systems are generalised to coalgebras. Specification languages together with their proof systems, in the following called (logical or modal) calculi, are presented by the associated classes of algebras (e.g., classical propositional logic by Boolean algebras). Stone duality will be used to relate the logics and their coalgebraic semantics. The relationship between these notions can be summarised as

$$\begin{tikzcd} \text{systems} \arrow[r, dash] \arrow[d, dotted, dash] & \text{coalgebras} \arrow[d, dash, "\text{Stone duality}"] \\ \text{logic} \arrow[r, dash] & \text{algebras} \end{tikzcd} \tag{1}$$

Let us have a closer look at the role of Stone duality, which relates the class $\mathcal{A}$ of algebras of a propositional logic to the class $\mathcal{X}$ representing the carriers of the transition systems (i.e., coalgebras). The duality is provided by two operations $P$ and $S$

$$\begin{tikzcd} \mathcal{X} \arrow[r, bend left, "P"] & \mathcal{A} \arrow[l, bend left, "S"] \end{tikzcd}, \tag{2}$$

where $P$ maps a carrier $X$ to its propositional theory and $S$ maps an algebra $A$ to its 'canonical model' ($PX$ and $SA$ are also called the *dual* of $X$ and $A$, respectively; the reason for the terminology will become clear later). The situation in Diagram 2 describes a perfect match of logic and semantics if both models and theories can be reconstructed (up to isomorphism) from their dual, formally, if $X \cong SPX$ and $A \cong PSA$ for all $X \in \mathcal{X}$ and $A \in \mathcal{A}$.

As we will explain below, the type of a class of coalgebras is an operation $T : \mathcal{X} \to \mathcal{X}$. This suggests that, in the same way as a logic for $\mathcal{X}$ is given by the algebras in $\mathcal{A}$, a logic for $T$-coalgebras is given by the algebras for the corresponding operation $L$ on $\mathcal{A}$. The Stone duality (2) then lifts to $L$-algebras and $T$-coalgebras

$$\begin{tikzcd} \mathsf{Coalg}(T) \arrow[r, bend left, "\tilde{P}"] \arrow[d] & \mathsf{Alg}(L) \arrow[l, bend left, "\tilde{S}"] \arrow[d] \\ \mathcal{X} \arrow[loop left, "T"] \arrow[r, bend left, "P"] & \mathcal{A} \arrow[l, bend left, "S"] \arrow[loop right, "L"] \end{tikzcd} \tag{3}$$

The logical interpretation of the upper duality is that the logic corresponding to $\mathsf{Alg}(L)$ is (strongly) complete and characterises bisimilarity wrt the transition system $\mathsf{Coalg}(T)$. This will be explained in Section 3. Section 2 discusses one example of the dualities in Diagram 2 in detail and briefly sketches the more general picture. The remainder of the introduction is devoted to a more detailed exposition of the ideas above. Section 4 summarises what can be gained from this approach and outlines some research directions. Definitions of the few notions from category theory we need are collected in an appendix.

## 1.1 Systems as Coalgebras

In its simplest form, a transition system consists of a set $X$ and a relation $R \subseteq X \times X$. Denoting by $\mathcal{P}$ the operation mapping a set to its powerset, a transition system can equivalently be described by a map

$$X \xrightarrow{\xi} \mathcal{P}X$$

where $\xi(x) = \{y|xRy\}$ is the set of successors of $x$. The structure $(X, \xi)$ is a $\mathcal{P}$-*coalgebra.*

This description of transition systems is very flexible. Table 1 gives some examples of $T$-coalgebras $(X, \xi)$ for suitable mappings $T : \mathsf{Set} \to \mathsf{Set}$. The set $C$ appearing in some functors is a constant

| | $TX$ | $X \xrightarrow{\xi} TX$ |
|---|---|---|
| 1. | $C \times X$ | streams over $C$ |
| 2. | $C \times X + 1$ | finite or infinite lists over $C$ |
| 3. | $2 \times X^C$ | deterministic automaton with input alphabet $C$ |
| 4. | $\mathcal{P}(C \times X) \cong (\mathcal{P}X)^C$ | $C$-labelled transition system |
| 5. | $(1 + \mathcal{D}X)^C$ | probabilistic transition systems |
| 6. | $2^{2^X}$ | predicate transformer |

Table 1: Examples of Coalgebras

parameter denoting an input or output alphabet. The models we have in mind are often systems with a distinguished initial state (which we also call *pointed* coalgebras or *processes*). (1) A coalgebra $X \to C \times X$ with specified initial state $x_0$ is a process outputting the infinite stream $(\mathit{head}(x_0), \mathit{head}(\mathit{tail}(x_0)), \ldots)$ of elements of $C$, where $\mathit{head} : X \to C$ and $\mathit{tail} : X \to X$ refer to the two components of $X \to C \times X$. (2) Here 1 denotes a one-element set, which allows a process to stop; hence, in addition to streams, one now also allows behaviours given by finite lists. (3) The 2 in $2 \times X^C$ denotes a two-element set and the $X \to 2$ part of the coalgebra expresses whether a state is accepting or not; $X^C$ denotes the set of functions from $C$ to $X$ and $X \to X^C$ calculates the successor state from a current state and an input from $C$. (4) Comparing with (1) and (3), $\mathcal{P}(C \times X)$ suggests to think of the labels in $C$ as outputs and $(\mathcal{P}X)^C$ of the labels as inputs, but both are isomorphic. (5) The distribution functor $\mathcal{D}X$ maps $X$ to the set of discrete probability distributions. (6) Coalgebras $X \to 2^{2^X}$ are (in bijective correspondence to) predicate transformers $2^X \to 2^X$.

**Bisimilarity** The crucial observation that makes the coalgebraic point of view useful is that all of these type constructors $T$ are functors and that, therefore, $T$-coalgebras come equipped with a canonical notion of behavioural equivalence or bisimilarity.

Let us explain this important point in more detail. To say that an operation $T : \mathsf{Set} \to \mathsf{Set}$ is a functor means that $T$ is not only defined on sets but also on functions, mapping $f : X \to Y$ to $Tf : TX \to TY$. Moreover, $T$ is required to preserve identities $\mathrm{id}_X : X \to X$, and composition, $T(f \circ g) = Tf \circ Tg$. This allows us to define a morphism of coalgebras $(X, \xi) \to (Y, \nu)$ as a map $f : X \to Y$ such that $Tf \circ \xi = \nu \circ f$:

$$\begin{tikzcd} X \arrow[r, "\xi"] \arrow[d, "f"'] & TX \arrow[d, "Tf"] \\ Y \arrow[r, "\nu"] & TY \end{tikzcd} \tag{4}$$

Coalgebraic *bisimilarity*, or *behavioural equivalence*, denoted $\simeq$, is now the smallest equivalence relation that is invariant under all morphisms: Define $\simeq$ to be the smallest equivalence relation containing all pairs $x \simeq f(x)$ where $f$ ranges over coalgebra morphisms and $x$ over the domain of $f$.[2]

[2]This definition is equivalent to the following one. Two states $x, y$ in two coalgebras are bisimilar iff there are two

Coming back to our introductory example, $\mathcal{P}$ is a functor if we let $\mathcal{P}f : \mathcal{P}X \to \mathcal{P}Y$ map a subset of $X$ to its direct image under $f$. It is now an instructive exercise to show that coalgebraic bisimilarity agrees with the standard notion of bisimilarity for (unlabelled) transition systems. For this, one shows that a map is a coalgebra morphism iff its graph is a bisimulation between the two transition systems.

In examples of Table 1 we have the following. (1,2) Two processes are bisimilar iff their outputs are the same. (3) Two states are bisimilar iff they accept the same language. (4) Here coalgebraic bisimilarity is the one known from modal logic or process algebra and similarly for (5) and (6).

**Final Coalgebras** It is often possible to characterise bisimilarity by a single coalgebra, the so-called final coalgebra. Formally, a coalgebra is *final* if from any other coalgebra there is a unique morphism into the final coalgebras. It follows that two states in two coalgebras are bisimilar iff they are identified by the unique morphisms into the final coalgebra. In other words, the final coalgebra, if it exists, provides a canonical representative for each class of bisimilar states.

In many cases, apart from characterising bisimilarity, final coalgebras are interesting objects in their own right. Recognising these structures as final coalgebras allows to reason about them, sometimes to great advantage, using coinduction instead of induction. We will not pursue this issue any further here but only mention some examples. In Table 1 the final coalgebra is in (1) the coalgebra of streams over $C$, in (2) the coalgebra of finite and infinite lists over $C$, in (3) the coalgebra of all languages (with successors given by language derivative). In our leading example of $\mathcal{P}$-coalgebras the final coalgebra is the universe of non-well founded sets.

## 1.2 Modal Logic

Let us now turn to logics for coalgebras. One would want such logics to respect bisimilarity, that is, formulae should not distinguish bisimilar states. We call such logics modal because it can be argued that invariance under bisimilarity is the main feature of modal logic. For example, a theorem of van Benthem states that modal logic is precisely the fragment of first order logic that is invariant under bisimilarity. Moreover, either by strengthening the logic allowing for infinite conjunctions or be restricting the semantics to finitely branching transition systems, modal logic characterises bisimilarity in the sense that for each two non-bisimilar states there is a formula distinguishing them.

Let us first look at the usual modal logic for unlabelled transition systems, ie, $\mathcal{P}$-coalgebras. It consists of classical propositional logic extended by one unary operator $\Box$. The interpretation of $\Box$ is that of a restricted universal quantifier, more precisely, $\Box\varphi$ holds in state $x$ iff $\varphi$ holds in all successors of $x$. We write this as

$$[\![\Box\varphi]\!]_{(X,R)} = \{x \mid xRy \;\Rightarrow\; y \in [\![\varphi]\!]_{(X,R)}\} \tag{5}$$

Having seen a logic for $\mathcal{P}$-coalgebras, can we generalise this to arbitrary functors $T$?

Note first that, semantically, a modal operator transforms predicates into predicates. So we could say that a modal operator is a suitable operation

$$2^X \to 2^X$$

coalgebra morphisms $f, g$ such that $f(x) = g(y)$.

where $2^X$ denotes again the set of functions $X \to 2$, or equivalently, the set of subsets of $X$. But we also want to capture that $\Box$ says something about the immediate successors of a state, that is, about a single transition step. We therefore identify (one-step) modalities $\Box$ for $T$ with so-called *predicate liftings*[3]

$$\boxbox_X : 2^X \to 2^{TX} \tag{6}$$

which give the semantics of a modal operator $\Box$ wrt a coalgebra $(X, \xi)$ via

$$2^X \xleftarrow{\xi^{-1}} 2^{TX} \xleftarrow{\boxbox_X} 2^X \tag{7}$$

that is, dropping the subscripts,

$$[\![\Box\varphi]\!] = \xi^{-1} \circ \boxbox[\![\varphi]\!]. \tag{8}$$

To recover (5) as a special case of (8), one defines the corresponding predicate lifting as $\boxbox Y = \{Z \subseteq X \mid Z \subseteq Y\}$ for $Y \in 2^X$.

How do we guarantee that modal logics for coalgebras given by predicate liftings are invariant under bisimilarity? Simply by requiring that predicate liftings $\boxbox_X : 2^X \to 2^{TX}$ are natural transformations. Spelling out the definition of a natural transformation this means that in the following diagram the right-hand square commutes (for all $f : X \to Y$)

$$\begin{tikzcd}
2^X & 2^{TX} \arrow[l, "\xi^{-1}"'] & 2^X \arrow[l, "\boxbox_X"'] \\
2^Y \arrow[u, "f^{-1}"'] & 2^{TY} \arrow[l, "\nu^{-1}"'] \arrow[u, "(Tf)^{-1}"'] & 2^Y \arrow[l, "\boxbox_Y"'] \arrow[u, "f^{-1}"']
\end{tikzcd} \tag{9}$$

If, moreover, $f : (X, \xi) \to (Y, \nu)$ is a coalgebra morphism, then also the left-hand square commutes and, therefore, the outer rectangle as well. The proof of invariance of the logic under bisimilarity (p. 3) is now a routine induction on the structure of the formulae, where the case of modal operators $\Box$ is taken care of by Diagram (9).

## 1.3 Logics as Algebras

After having explained the basic notions of coalgebras and their logics, in the remainder of the article, I will sketch a deeper analysis of the situation. It is based on the insight that logics for coalgebras are in fact algebras and, moreover, that a logic perfectly captures the coalgebras if the algebras and coalgebras are related by Stone duality.

Observe that the modal logics discussed above come in two stages. First, for any coalgebra $(X, \xi)$ we have the Boolean algebra $2^X$, which corresponds to propositional logic. This logic is then extended by modal operators $2^X \to 2^X$. Traditionally, these algebras, called Boolean algebras with operators or modal algebras, are thought of as given by a carrier $A$ plus boolean operators $\bot, \neg, \wedge, \vee$ plus (possibly more than one) modal operator $\Box$. For example, the modal logic for $\mathcal{P}$-coalgebras can be given by one unary modal operator $\Box$ that preserves top (true) and conjunction

$$\Box\top = \top \qquad\qquad \Box(a \wedge b) = \Box a \wedge \Box b \tag{10}$$

This example shows clearly the relationship between algebras and modal calculi. On the one hand, (10) is the equational definition of the class of modal algebras. On the other hand, (10) plus

[3]'Predicate lifting' because $\boxbox$ lifts a predicate on $X$ to a predicate on $TX$.

equational logic provides a calculus for modal logic. Since modal logics are more commonly given by Hilbert calculi, we indicate briefly that both calculi are equivalent in a rather straightforward way.

The usual Hilbert calculus of the modal logic for $\mathcal{P}$-coalgebras, denoted $\mathbf{K}$, has as axioms all propositional tautologies and $\Box(p \to q) \to (\Box p \to \Box q)$; rules are modus ponens, substitution, and necessitation 'from $\varphi$ derive $\Box\varphi$'. To compare $\mathbf{K}$ with the equational calculus given by (10), we write $\vdash_{\mathbf{K}} \varphi$ and $\vdash_{\mathsf{EL}} \varphi = \psi$ for formulae derivable in $\mathbf{K}$ and equations derivable in equational logic. One then shows that $\vdash_{\mathbf{K}} \varphi \iff \vdash_{\mathsf{EL}} \varphi = \top$ and $\vdash_{\mathsf{EL}} \varphi = \psi \iff \vdash_{\mathbf{K}} \varphi \leftrightarrow \psi$. For example, the necessitation rule is simulated on the equational side by the congruence rule of equational logic and the first of the axioms (10).

Up to now we have only seen standard material from modal logic. It will now be shown that modal algebras are algebras for a functor. Since algebras for a functor are, in a precise sense, dual to coalgebras this will allow us to relate modal algebras (and hence modal calculi) in a uniform way to their coalgebraic semantics.

We start by observing that the two-stage process of building a modal algebra can be made more explicit by saying that a modal algebra is a Boolean algebra $A$ with a finite-meet preserving map $\Box : A \rightharpoonup A$. From a technical point of view, it is inconvenient that $A$ is a Boolean algebra but $\Box$ is only a meet-semi-lattice morphism, which does not preserve all of the Boolean structure. This is easily rectified: Modal algebras are in one-to-one correspondence to algebras for the functor $L$ where

$$LA \text{ is the free Boolean algebra over } A \text{ considered as a meet-semi-lattice.} \tag{11}$$

This means that $LA$ is determined by the property that for each finite-meet preserving function $A \rightharpoonup B$ there is a unique Boolean algebra morphism $LA \to B$ such that

$$\begin{tikzcd} LA \arrow[r] & B \\ A \arrow[u] \arrow[ur] & \end{tikzcd} \tag{12}$$

commutes. It follows that Boolean algebra morphisms $LA \to A$ are in one-to-one correspondence with semi-lattice morphisms $A \rightharpoonup A$.

$LA \to A$ is an *algebra for the functor* $L$. This notion of an algebra for a functor dualises the notion of a coalgebra, the arrows going in opposite directions: into the carrier for algebras and out of the carrier for coalgebras (the appendix gives a more formal statement of this duality).

Let us summarise the relationship between logics and algebras in our example. Boolean algebras correspond to classical propositional logic. A functor $L$ specifies an extension of propositional logic with modal operators. The algebras for this modal logic are the algebras for the functor $L$.

| | |
|---|---|
| class $\mathsf{BA}$ of Boolean algebras | classical propositional logic |
| functor $L : \mathsf{BA} \to \mathsf{BA}$ | modal operators + axioms |
| class $\mathsf{Alg}(L)$ of modal algebras | modal logic |

### 1.4 Relating Algebras and Coalgebras via Stone Duality

We have explained so far the two horizontal lines of the picture (1), namely systems as coalgebras and logics as algebras. We now come to Stone duality, relating the two.

We start by remarking that finitary modal logic does not describe $\mathcal{P}$-coalgebras perfectly in the following sense. First, finitary logics cannot characterise bisimilarity, that is, they are not strong enough to distinguish all non-bisimilar states. Furthermore, there are consistent modal logics that are incomplete in the sense that there are no $\mathcal{P}$-coalgebras satisfying them.[4] There are two ways to rectify this mismatch.

The first is based on the observation that $2^X$ is not only a Boolean algebra but also has infinitary intersections, or algebraically speaking, $2^X$ is a complete atomic Boolean algebra. This suggests that a perfect description of transition system requires infinitary propositional logic. This is well-known in process algebra: For infinitely branching transition systems Hennessy-Milner logic only characterises bisimilarity if one allows infinite conjunctions.

Alternatively, instead of strengthening the logic by infinitary constructs, one can modify the semantics to take the weaker expressivity of the logic into account: One equips transition systems with a notion of 'admissible' or 'observable' predicate. For this, one usually lets carriers consist not of sets but topological spaces $(X, \mathcal{O}X)$. The topology $\mathcal{O}X$ is a subset of $2^X$ encoding which predicates on $X$ can be expressed by the logic.[5]

In the first case, the algebras in Diagram 2 are complete atomic Boolean algebras, so Diagram 2 becomes

$$\begin{tikzcd} \mathsf{Set} \arrow[r, bend left, "P"] & \mathsf{CABA} \arrow[l, bend left, "S"] \end{tikzcd},$$

where $PX = 2^X$ and $S$ maps an algebra to its set of atoms.[6]

In the second case, the algebras are Boolean algebras. The corresponding spaces are known as Stone spaces and Diagram 2 becomes

$$\begin{tikzcd} \mathsf{Stone} \arrow[r, bend left, "P"] & \mathsf{BA} \arrow[l, bend left, "S"] \end{tikzcd}.$$

Both diagrams are dual equivalences, so from an abstract point of view they share exactly the same (or rather dual) properties. But the categories on the right-hand side are categories of algebras that come with an equational logic. Lifting such a basic duality to a duality of coalgebras and modal algebras, as indicated in Diagram 3, will provide modal logics for coalgebras.

### 1.5 Notes

References to Stone duality will be given in the next section. The standard reference for systems and coalgebras is Rutten [61]. The duality of algebras/coalgebras and induction/coinduction are

[4] This phenomenon appears if the proposition letters of the modal axioms are interpreted as ranging over all subsets of the carrier of the model (Kripke frame semantics). It does not happen if proposition letters receive a fixed interpretation (Kripke model semantics).

[5] For example, consider the modal logic **K** and a $\mathcal{P}$-coalgebra that is a tree with initial state $x_0$ having branches of any bounded length and one infinite branch. The subset of states reachable in bounded branches is not admissible. This corresponds to the fact that having an infinite branch is not expressible in the finitary logic **K**.

[6] $a$ is an atom if $\bot < a$ and $\bot < b \leq a \;\Rightarrow\; b = a$.

explained in detail in the tutorial by Jacobs and Rutten [32]. Further introductions are provided by the course notes of Gumm [27], Pattinson [55], and Kurz [46] and the forthcoming book by Jacobs [29].

**Coalgebras** Motivated by Milner's CCS (4 in Table 1), Aczel [5] introduced the idea of coalgebras for a functor $T$ as a generalisation of transition systems. He also made three crucial observations: (1) coalgebras come with a canonical notion of bisimilarity; (2) this notion generalises the notion from computer science and modal logic; (3) any 'domain equation' $X \cong TX$ has a canonical solution (in sets or classes), namely the final coalgebra, which is fully abstract wrt behavioural equivalence.

This idea of a type of dynamic systems being represented by a functor $T$ and an individual system being an $T$-coalgebra, led Rutten [61] to the theory of universal coalgebra which, parameterised by $T$, applies in a *uniform* way to a large class of different types of systems. In particular, final semantics and the associated proof principle of coinduction (which are dual to initial algebra semantics and induction) find their natural place here.

The following references provide details on the examples in Table 1. Stream coalgebras have been studied by Rutten in a number of papers, see e.g. [63]. For the example of deterministic automata as coalgebras see Rutten [60]. Probabilistic transition system as coalgebras go back to Rutten and de Vink [21]. Coalgebras for the double contravariant powerset functor are investigated in Kupke and Hansen [28].

The idea of systems as coalgebras and the paradigm of final semantics—together with its associated principles of coinduction—has been applied to such different topics as, for example, automata theory [60], combinatorics [62], control theory [39], denotational semantics of $\pi$-calculus [23, 66], process calculi and GSOS-formats [69, 8, 37], probabilistic transition systems [9, 18, 51], component-based software development [6, 7], and the solution of recursive program schemes [49]. Modelling classes in object-oriented programming as coalgebras [57, 30] led to new verification tools (LOOP-Tool [70], CCSL [59], CoCasl [52]) which also incorporate reasoning with modal logics based on the research on coalgebras and modal logic described below.

**Coalgebras and Modal Logic** For background on modal logic the reader is referred to Blackburn, de Rijke, Venema [10] (Thm 2.68 shows that modal logic is the bisimulation invariant fragment of first-order logic, Chapter 5 is on modal algebras, Thm 4.49 gives an example of an incomplete modal logic). Further material can be found in Venema [72]. Modal algebras as algebras for a functor and their duality to coalgebras for a functor was first presented in Abramsky [1].

Research into coalgebras and modal logic started with Moss [50]. The logic of [50] is uniform[7] in the functor $T$, but it does not provide the linguistic means to decompose the structure of $T$ which is needed to allow for a flexible specification language. To address this issue, [47, 58] (independently) proposed to restrict attention to specific classes of functors and presented a suitable, but ad hoc, modal logic. This work was generalised by Jacobs [31]. Pattinson showed that these languages with their ad hoc modalities arise from modal operators given by predicate liftings. He gives conditions under which logics given by predicate liftings are sound and complete [54] and expressive [56]. Schröder [64] and Klin [38] show that for any finitary functor $T$ on $\mathsf{Set}$ there is a modal logic given by predicate liftings that characterises bisimilarity.

From a semantical point of view, modal logic can be considered as dual to equational logic [45, 44]. [48] goes further and shows that coalgebras can be specified—in the same (or dual) way as algebras—by operations and equations; moreover, the dual of the algebraic operations turn out to

[7]The restrictions are that $T$ is on $\mathsf{Set}$ and has to preserve weak pullbacks.

be bisimilarity preserving predicate transformers, that is, modal formulae. The results following from this approach work for all functors but the logics need to be strong enough to express all possible behaviours. This needs, in general, infinite conjunctions in the logics. To study finitary logics, Jacobs [31] covers some ground towards a duality for coalgebras/generalised BAOs and Goldblatt [26] develops a notion of ultrapower for coalgebras. Both approaches are restricted again to specific classes of functors. In this paper we argue that, based on Stone duality, it is possible to develop a uniform account.

# 2 Stone Duality

We will treat Stone duality for Boolean algebras as an illustrative example and then remark on how it generalises to other cases.

## 2.1 The Representation Theorem for Boolean Algebras

The axioms of a Boolean algebra relating $\bot, \neg, \wedge, \vee$ are the abstract essence of the set-theoretic operations of empty set, complement, intersection and union. But how can one show that the axioms of Boolean algebra are indeed complete? We have to exhibit, for each non-derivable equation, an algebra of subsets violating that equation.

Suppose $\varphi = \psi$ is not derivable from the axioms of Boolean algebra. By completeness of equational logic, there is a Boolean algebra $A$ such that $A \not\models \varphi = \psi$. To conclude that there is a Boolean algebra of subsets (a field of sets) that refutes $\varphi = \psi$ it is enough to find a set $SA$ and an injective Boolean algebra morphism

$$A \to PSA$$

where $P$ denotes here the operation mapping a set to the Boolean algebra of its subsets. Indeed, if $A \not\models \varphi = \psi$, then by injectivity $PSA \not\models \varphi = \psi$, yielding a counterexample for $\varphi = \psi$ in an algebra where all the Boolean operations are interpreted by their set-theoretic counterparts.

How does one get the points of the space $SA$? Similarly to defining real numbers as certain collections of intervals, a point will be a certain collection of elements of $A$, or, equivalently, a function $A \to 2$. Which of these functions should be points? Observing that 2 is not only a set, but also a Boolean algebra 2, we define $SA = \mathsf{BA}(A, 2)$ where the notation $\mathsf{BA}(A, B)$ denotes the set of Boolean algebra morphisms $A \to B$. Detailing the definition of an algebra morphism, it is straightforward to verify that the requirement that $p : A \to 2$ be an algebra morphism says that $p$ is a maximal and consistent collection of elements of $A$. With the canonical map $A \to PSA$, we can now state Stone's representation theorem for Boolean algebras. Note that with the definition below, the statement that a point $p$ satisfies the predicate $a$ expresses itself as $p \in \hat{a}$.

**Theorem 2.1.** *The map*

$$\hat{(\cdot)} : A \longrightarrow PSA \tag{13}$$

$$a \;\mapsto\; \hat{a} = \{p \in SA \mid p(a) = 1\} \tag{14}$$

*is an injective Boolean algebra morphism.*

## 2.2 Stone Duality for Boolean Algebras

The representation theorem works by associating a space to an algebra (via $S$) and then, vice versa, an algebra to a space (via $P$). What precisely are the spaces that correspond to algebras?

In a first instance, we can say that a space $(X, A)$ consists of a set $X$ and a Boolean algebra of subsets $A \subseteq 2^X$ such that (1) any two different points in $X$ are separated by elements of $A$ and (2) $(X, A)$ is compact, that is, every collection $\mathcal{C}$ of elements of $A$ with the finite intersection property [8] has non-empty intersection. The two properties capture that the points of the space are determined by the algebra in the following sense. (1) says that there are not more points than can be separated by predicates and (2) that there are enough points to realise every consistent collection of predicates from $A$.

Further, one notices that a space $(X, A)$ can be considered as the topological space $(X, \mathcal{O}X)$ with $\mathcal{O}X$ being the topology generated by $A$, that is, the closure of $A$ under arbitrary unions. One recovers the Boolean algebra $A$ from $\mathcal{O}X$ as the collection of all compact opens. Since in a compact Hausdorff space a subset is compact iff it is closed, one can replace compact open by clopen (which is brief for closed and open). To summarise:

**Definition 2.2.** A Stone space is a topological space that (1) is $T_0$, (2) compact, and (3) the clopens are a basis for the topology.

Stone spaces with continuous maps form the category $\mathsf{Stone}$. From the representation theorem and the definition of $\mathsf{Stone}$ we obtain two operations $S$ and $P$

$$\begin{tikzcd} \mathsf{Stone} \arrow[r, bend left, "P"] & \mathsf{BA} \arrow[l, bend left, "S"] \end{tikzcd}$$

where $SA$ is the topological space with points $\mathsf{BA}(A, 2)$ and the topology generated by $\{\hat{a} \mid a \in A\}$ as in (14); $PX = \mathsf{Stone}(X, 2)$ is now the Boolean algebra of clopens (instead of the full powerset). We speak of a duality here because both operations are functors that act on morphism by reversing the arrows, namely, mapping a morphism $f$ to inverse image $f^{-1}$. Moreover, $\mathsf{Stone}$ and $\mathsf{BA}$ are dually equivalent, that is we have isomorphisms

$$A \cong PSA \tag{15}$$

$$X \cong SPX \tag{16}$$

(16) is injective because $X$ is $T_0$ and surjective because $X$ is compact. (15) is surjective by construction and injective by the Representation Theorem 2.1. To summarise:

**Theorem 2.3.** *The categories* $\mathsf{BA}$ *and* $\mathsf{Stone}$ *are dually equivalent.*

From our presentation, one could get the impression that topologies come in here accidentally and the logical content of the duality is completely contained in the representation theorem. I would reply the following. First, Stone spaces arise here from logical considerations but they are of independent interest. A well-known example is the Cantor middle-third space. In fact, all complete ultrametric spaces are Stone spaces. Second, the dual equivalence is nice to have; for example, we then have that the dual of an initial algebra is the final coalgebra; this will be used in the next section to show that that modal logics characterise bisimilarity. Third, topologies often have an interesting computational perspective arising from the idea that observable properties are closed under arbitrary unions but not intersections [65, 73, 22]. Finally, the topological perspective suggests and unifies many generalisations, some of which we briefly review now.

[8] $\mathcal{C}$ has the finite-intersection property if all finite subset of $\mathcal{C}$ have non-empty intersection.

### 2.3 A Sketch of the General Picture

The variations of Stone duality relevant for the present purposes fit the following picture. We start with a class $\mathcal{A}$ of distributive lattices and $\mathcal{X}$ of topological spaces (assumed to be $T_0$). Think of algebras $A \in \mathcal{A}$ as propositional theories and of spaces $X \in \mathcal{X}$ as models of propositional theories with the opens (or compact opens for finitary logics) interpreting the propositions. There is an operation $P : \mathcal{X} \to \mathcal{A}$, mapping a space to its algebra of predicates. And an operation $S : \mathcal{A} \to \mathcal{X}$ mapping an algebra to its 'canonical model'.

$$\begin{tikzcd} \mathcal{X} \arrow[r, bend left, "P"] & \mathcal{A} \arrow[l, bend left, "S"] \end{tikzcd} . \tag{17}$$

Moreover, in the examples of the table below, $PX = \mathcal{X}(X, 2)$ and $SA = \mathcal{A}(A, 2)$ where 2 denotes the appropriate two-element topological space or two-element algebra. We speak of a duality, since $P$ and $S$ work contravariantly on morphisms, mapping a morphism (that is, algebra morphism or continuous map) $f$ to $f^{-1}$. Moreover, there are morphisms

$$\varrho_A : A \to PSA \qquad\qquad \sigma_X : X \to SPX$$

and (17) is a dual equivalence if they are bijective. Logically, this means the following. As we have explained in Section 2.1, $\varrho_A$ injective means completeness (and it will, in general, be surjective by definition of $S$ and $P$). $\sigma_X$ is injective means, together with $X$ being $T_0$, that the logic is expressive in the sense that different points are separated by some predicate. If $\sigma_X$ is not surjective, then $SPX$ has points not available in $X$; thus the logic is not strong enough to make these additional points inconsistent.

We conclude with a table of some relevant examples.

| $\mathcal{X}$ | $\mathcal{A}$ | spaces/algebras | propositional logic |
|---|---|---|---|
| Set | CABA | sets/complete atomic Boolean algebras | infinitary classical |
| Stone | BA | Stone spaces/Boolean algebras | classical |
| Spec | DL | spectral spaces/bounded distributive lattices | negation free |
| Poset | CDL | posets/complete distributive lattices | infinitary negation free |
| Sob | Frm | sober spaces/frames | geometric |

In the two last examples, $\varrho_A$ is injective for free algebras $A$ but not for all algebras. Logically, this corresponds to having completeness but not strong completeness. This also happens for propositional logic with countable conjunctions.

### 2.4 Notes

Three introductory textbooks on Stone duality are Vickers [73], Davey and Priestley [19], Brink and Rewitzky [16].

Stone duality was introduced by Stone [67, 68]. The main reference for Stone duality is Johnstone's book on Stone Spaces [33] which also provides detailed historical information. The handbook article [4] covers the topic from the point of view of domain theory. Both texts also provide many more examples of Stone dualities. Topological dualities beyond sober spaces, e.g., for completely distributive lattices and posets, are treated by Bonsangue et al [11, 15]. The representation theorem for propositional logic with countable conjunctions can be found in Karp [36]. For applications of complete ultrametric spaces to control flow semantics see de Bakker and de Vink [20].

# 3 Logics of Coalgebras

The previous section discussed dual equivalences (17) between categories $\mathcal{X}$ of topological spaces and categories $\mathcal{A}$ of distributive lattices. In this section, we extend this picture to $T$-coalgebras. Starting with a diagram as in (17) and a functor $T$ on $\mathcal{X}$, we dualise $T$ to a functor $L$ on $\mathcal{A}$. The duality of $\mathcal{X}/\mathcal{A}$ and $T/L$ lifts to a duality of coalgebras and algebras.

$$\begin{tikzcd} \mathsf{Coalg}(T) \arrow[r, shift left, "\tilde{P}"] \arrow[d] & \mathsf{Alg}(L) \arrow[l, shift left, "\tilde{S}"] \arrow[d] \\ \mathcal{X} \arrow[loop left, "T"] \arrow[r, bend left, "P"] & \mathcal{A} \arrow[l, bend left, "S"] \arrow[loop right, "L"] \end{tikzcd} \tag{18}$$

And in the same way as the duality of $\mathcal{X}$ and $\mathcal{A}$ describes a logic for $\mathcal{X}$, so the duality of $\mathsf{Coalg}(T)$ and $\mathsf{Alg}(L)$ describes a logic for $T$-coalgebras.

## 3.1 Abstract Logics: Using the Duality

Given a duality as in (17) and a functor $T$ on $\mathcal{X}$, then $PTS$ is the dual of $T$ on $\mathcal{A}$. In fact, we will need a bit more liberty and say that $L$ is dual to $T$ if $L$ is isomorphic to $PTS$. Or, equivalently, $L$ is *dual* to $T$ if there is a natural isomorphism

$$\delta_X : LPX \to PTX \tag{19}$$

Using $\delta$ we can associate to a $T$-coalgebra $(X, \xi)$ its dual $L$-algebra

$$\tilde{P}(X,\xi) \;=\; LPX \xrightarrow{\delta_X} PTX \xrightarrow{P\xi} PX$$

and similarly for $S$.

In algebraic logic, logics are described by operations and equations, and then properties of a logic are studied by investigating the variety of the algebras for the given operations and equations. A basic construction is that of the Lindenbaum algebra. Given a logic $L$, the Lindenbaum algebra $A_L$ is obtained from quotienting the set of all terms by the smallest congruence derived from the equations. Thus, the elements of the Lindenbaum algebra $A_L$ can be seen as 'abstract propositions', or propositions up to interderivability. Among all algebras in the variety, the Lindenbaum algebra is determined by the following property: for any algebra $A$ there is a unique morphism $A_L \to A$, that is, $A_L$ is the initial algebra. We turn this into a definition.

**Definition 3.1.** Denote by $A_L$ the initial $L$-algebra. The elements of $A_L$ are called propositions. The semantics $[\![\varphi]\!]_{(X,\xi)}$ of a proposition $\varphi$ wrt a coalgebra $(X, \xi) \in \mathsf{Coalg}(T)$ is given by the image of $\varphi$ under

$$A_L \longrightarrow \tilde{P}(X,\xi)$$

We write $\mathsf{Coalg}(T) \models (\varphi = \psi)$ if for all coalgebras $(X, \xi)$ the equation $\varphi = \psi$ is satisfied in $\tilde{P}(X, \xi)$.

We remark that Theorem 3.7 will explain precisely in what sense the initial $L$-algebra is a Lindenbaum algebra.

**Theorem 3.2.** *Propositions are invariant under bisimilarity.*

*Proof.* Recalling the definition of bisimilarity (p. 3), we have to show, given a coalgebra morphism $f : (X, \xi) \to (X', \xi')$ and $x \in X$, that $x \in [\![\varphi]\!]_{(X,\xi)} \Leftrightarrow f(x) \in [\![\varphi]\!]_{(X',\xi')}$. This follows directly from the fact that the diagram

$$\begin{tikzcd}
 & \tilde{P}(X,\xi) \\
A_L \arrow[ur, "{[\![-]\!]_{(X,\xi)}}"] \arrow[dr, "{[\![-]\!]_{(X',\xi')}}"'] & \\
 & \tilde{P}(X',\xi') \arrow[uu, "\tilde{P}f = Pf = f^{-1}"']
\end{tikzcd}$$

commutes due to $A_L$ being initial. □

The essence of completeness wrt to the coalgebraic semantics is:

**Theorem 3.3.** $\mathsf{Alg}(L) \models (\varphi = \psi) \quad \Leftrightarrow \quad \mathsf{Coalg}(T) \models (\varphi = \psi)$.

*Proof.* ‘ $\Rightarrow$ ’ (soundness) is immediate from the definitions. ‘ $\Leftarrow$ ’ (completeness) works as in Theorem 2.1. Suppose $A_L \not\models \varphi = \psi$. By injectivity of $A_L \to \tilde{P}\tilde{S}A_L$ we have $\tilde{P}\tilde{S}A_L \not\models \varphi = \psi$. That is, the coalgebra $\tilde{S}A_L$ does not satisfy $\varphi = \psi$. □

We remark that, as apparent from the proof, it is the representation of the initial (or, more generally, free algebras) which gives completeness. Since we have a dual equivalence, all algebras can be represented and we obtain strong completeness (completeness wrt a set of assumptions).

**Theorem 3.4.** *The logic characterises bisimilarity.*

*Proof.* Without loss of generality, let us assume that $x, x'$ are two different elements of the final coalgebra $\tilde{S}A_L$. The two points can be distinguished by a proposition since $A_L \to \tilde{P}\tilde{S}A_L$ is surjective and $\tilde{S}A_L$ is a $T_0$-space. □

To summarise, we have seen how to obtain a logic that perfectly describes $T$-coalgebras: Just consider as formulae the elements of the initial $L$-algebra[9] where $L$ is the dual of $T$. We called this logic abstract since it is not explicitly built from modal operators and axioms. The next subsection explains that modal operators and axioms are presentations of the functor $L$.

### 3.2 Concrete Logics: Presenting Algebras and Functors

Ultimately, we are interested in relating logical calculi to transition systems. We have motivated to consider transition systems as coalgebras and used Stone duality to dualise coalgebras to algebras. The particular benefit obtained from using Stone duality is that the algebras thus obtained correspond to logical calculi. Let us take a closer look again at the guiding ideas, which have been:

[9]If $T : \mathsf{Set} \to \mathsf{Set}$ is powerset, then the initial $L$-algebra does not exist for reasons of size. But one can still define a class of formulae using the initial algebra sequence of $L$. We ignore this slight complication for the purposes of exposition.

| | |
|---|---|
| category of algebras $\mathcal{A}$ | propositional logic |
| algebra $A$ in $\mathcal{A}$ | propositional theory |
| functor $L : \mathcal{A} \to \mathcal{A}$ | operations and equations for $T$-coalgebras |
| category $\mathsf{Alg}(L)$ | modal logic for $T$-coalgebras |

These correspondences are justified as follows. The categories $\mathcal{A}$ obtained from Stone duality can be presented by a signature $\Sigma$ of operations and equations $E$ in the sense that $\mathcal{A}$ is (isomorphic to) the class $\mathsf{Alg}(\Sigma, E)$ of algebras for the signature $\Sigma$ satisfying $E$. The presentation $\langle \Sigma, E \rangle$ gives a logical calculus, via equational logic. An algebra $A \in \mathcal{A}$ has a presentation $\langle G, R \rangle$ by generators and relations if $A$ is isomorphic to the quotient of the free algebra over $G$ by the smallest congruence containing $R$. In our context, this means that $A$ is the propositional theory given by variables $G$ and additional axioms $R$, see the example below.

**Presenting functors** We emphasised above the point of view that a propositional logic is a presentation of a category of algebras. Similarly, it is a presentation of $L$ that gives rise to the modal operators and its axioms.

*Example.* The functor $L : \mathsf{BA} \to \mathsf{BA}$ for $\mathcal{P}$-coalgebras from (11) is presented by the unary operator $\Box$ and the equations (10) in the following sense. For each $A \in \mathsf{BA}$, the algebra $LA$ is presented by generators $\{\Box a \mid a \in A\}$ and by relations $\{(\Box\top, \top)\} \cup \{(\Box(a \wedge b), \Box a \wedge \Box b) \mid a, b \in A\}$.

It is not a coincidence that the equations in this example are of a special format: Roughly speaking, they do not allow nesting of modal operators. Such terms are called terms of rank 1:

**Definition 3.5.** Assume $\mathcal{A} \cong \mathsf{Alg}(\Sigma, E)$ and a signature $\Sigma'$ (with operation symbols disjoint from $\Sigma$). A term in $\Sigma + \Sigma'$ is of rank 1 (wrt $\Sigma'$) if it is of the form $t(\Box_i(s_{ij}))$ where $t$ is an $n$-ary term in $\Sigma$ and the $\Box_i$, $0 \le i < n$, are $m_i$-ary operations in $\Sigma'$ and the $s_{ij}$, $0 \le j < m_i$ are terms in $\Sigma$. An equation $t = s$ is of rank 1 if both terms are.

In our example, the equations (10) are of rank 1. In particular: $\top$ is a term of rank 1, because $\top$ is a 0-ary term in the signature $\Sigma$ of Boolean algebras; $\Box(a \wedge b)$ is a term of the form $t(\Box(s))$ where $t$ is a variable and $s$ is $a \wedge b$. Terms like $\Box a \to a$ and $\Box a \to \Box\Box a$ are not of rank 1. We can now define what it means to present a functor by operations and equations.

**Definition 3.6.** Assume $\mathcal{A} \cong \mathsf{Alg}(\Sigma_{\mathcal{A}}, E_{\mathcal{A}})$, a signature $\Sigma_L$ and a set of equations $E_L$ that are of rank 1 (wrt $\Sigma_L$). $\langle \Sigma_L, E_L \rangle$ is a presentation of $L : \mathcal{A} \to \mathcal{A}$ if the algebras $LA$ are presented by $\langle G_A, R_A \rangle$ where $G_A = \{\sigma(a_i) \mid \sigma \in \Sigma_L, a_i \in A\}$ and $R_A$ consists of all substitution instances of equations in $E_{\mathcal{A}} \cup E_L$ obtained by replacing variables with elements from $A$.

Generalising the example above, it now follows that logics given by predicate liftings correspond to functors $L : \mathsf{BA} \to \mathsf{BA}$. Indeed, if $\Sigma$ is a collection of predicate liftings (with arities possibly $> 1$), then $\langle \Sigma, \emptyset \rangle$ presents some functor $L$. Moreover, it is not hard to see that the two semantics of the modal operators given by (8) and Definition 3.1 coincide. This also means that, conversely, any presentation of a functor corresponds to a collection of predicate liftings (given by $\Sigma_L$) plus some additional axioms.

The next theorem links the abstract logics from the previous section with concrete logical calculi. In particular, it shows that the Lindenbaum algebra of the logic given by operations $\Sigma_{\mathcal{A}} + \Sigma_L$ and equations $E_{\mathcal{A}} + E_L$ is the initial $L$-algebra. The proof that every $L$-algebra satisfies the equations $E_L$ requires the restriction to rank 1.

**Theorem 3.7.** *Assume* $\mathcal{A} \cong \mathsf{Alg}(\Sigma_\mathcal{A}, E_\mathcal{A})$ *and* $L : \mathcal{A} \to \mathcal{A}$*. If* $L$ *has a presentation* $\langle \Sigma_L, E_L \rangle$ *then* $\mathsf{Alg}(L) \cong \mathsf{Alg}(\Sigma_\mathcal{A} + \Sigma_L, E_\mathcal{A} + E_L)$*.*

The theorem can be read in two directions. First, starting with $T$, we find a presentation for $L$ and obtain completeness results for modal calculi. Of course, finding a good such presentation for a functor is usually not straightforward. It is therefore of interest to know whether arbitrary functors $L$ do have a presentation. This question has recently received a positive answer for finitary functors on Boolean algebras and sifted colimits preserving functors on arbitrary varieties. I expect that these results can be extended to show that modal calculi exist for all functors $T$ on $\mathsf{Set}$ and related categories.

Second, one can take a logical calculus and study the corresponding presented functor. For example, the infinitary version of the modal calculus **K** presents the dual $L : \mathsf{CABA} \to \mathsf{CABA}$ of $\mathcal{P} : \mathsf{Set} \to \mathsf{Set}$. We obtain the theorem mentioned in the introduction that infinitary modal logic characterises bisimilarity. Moreover, we also get a strong completeness result for the infinitary version of the modal calculus **K**. For **K** itself we obtain the corresponding results for the powerspace (also known as Plotkin power domain or hyperspace) on $\mathsf{Stone}$. Similarly, any modal logic of rank 1 is expressive and strongly complete for some functor of Stone spaces (also this may not be intended semantics). This methodology can be applied to all functors in Table 1 as presentations of their duals are known, with the possible exception of the functors for probabilistic transition systems which deserve some further attention.

### 3.3 Notes

The application of Stone duality to modal logic goes back to Jónsson and Tarski [34, 35] and then Goldblatt [25]. The idea of relating type constructors on algebras (see the $L$ above) and topological spaces (see the $T$ above) is from Abramsky's Domain Theory in Logical Form [3, 2]. Compared to [3, 2], the models we are interested in are not only solutions to recursive domain equations (final coalgebras) but any coalgebras; moreover, their base category need not be a domain but can be a more general topological space. Compared to [25], we use the duality of algebras and coalgebras to lift the Stone duality from Boolean logic to modal logic. Our functors $L$ (or their presentations) are closely related to Cîrstea's language and proof system constructors [17].

[42] studies coalgebras over Stone spaces to show that they capture the descriptive general frames from modal logic and to present a different view on Jacobs many-sorted coalgebraic modal logic [31]; [53] applies this approach to give a coalgebraic analysis of positive modal logic; [12] proposes to study logics for coalgebras via the dual functor and shows that powerspace can be treated in a uniform way for different categories of topological spaces; [14] introduces the notion of a functor presented by operations and equations; [40] shows that logics given by predicate liftings can be described by functors $L : \mathsf{BA} \to \mathsf{BA}$; [41] studies the relationship between Stone-coalgebras and Set-coalgebras.

## 4 Outlook

The aim of this exposition was to give a principled explanation of coalgebras and their logics. It cannot be denied that it took us some work in Section 2 to set up the necessary machinery. On the other hand, we got paid back with short and easy proofs of Theorems 3.3 and 3.4. Notice that these proofs of completeness and expressiveness do not involve any syntax. The interface between syntax and semantics, so to speak, is provided by the notion of a presentation of a functor. This provides an interesting way to reason about different modal logics in a uniform and syntax independent way.

One of the benefits of setting up the theory of coalgebras and their logics in a way uniform in the functor is compositionality. For example, given presentations for $L_1$ and $L_2$, one obtains a presentation of the composition $L_1 L_2$. This allows us to not only build new types of coalgebras from old ones, but to do the same for their associated logics (as done already in Abramsky [3]). The power of this approach is exemplified by [13] which derives a logic for $\pi$-calculus: Using known results and compositionality, a presentation for the functor of $\pi$-calculus is not difficult to find and we can then apply the general results.

Let us conclude with some further topics.

**The modal logic of a functor** Our original question has been the following. If universal coalgebra is a general theory of systems as proposed by Rutten [61], then what are the logics for coalgebras? More specifically, can the theory of logics for $T$-coalgebras be developed uniformly in the functor $T$? The insight alone that, semantically, modal logic is dual to equational logic [44] does not give a handle on relating coalgebras and their modal calculi. As shown here, this is where Stone duality comes in. The solution to the original problem of associating a logic to a functor $T$ now looks in close reach: It will be shown that, under appropriate conditions, the dual of a functor $T$ has a presentation, which then provides a strongly complete modal logic characterising bisimilarity. This should also allow to generalise Moss's original work [50] and provide his logic with a complete calculus.

**Relating different Stone dualities** Topology-based models arise either, as in this article, to capture the expressivity of logics weaker than infinitary classical logic, or in situations, as in domain theory, where a natural notion of observable predicate is given. In both cases, it would be interesting to be able to treat the topology as a parameter. This would allow us to compare similar models based on different categories of spaces and to study logics which involve two different Stone dualities, e.g., the ones for $\mathsf{BA}$ and $\mathsf{Set}$. Ongoing work is based on the idea to consider both dualities as arising from different completions of one and the same simpler duality.

**Logics with name binding** The work on the logic of $\pi$-calculus [13] suggests that also other logics with name binding and quantifiers can be usefully treated in the presented framework. This needs still to be worked out.

**Coalgebraic modal model theory** In order to better appreciate the relationship between modal logic and coalgebras, it would be good to understand in how far known results in modal logic can be extended to coalgebras. Some work in this direction has been done on the Jónsson-Tarski-theorem and ultrafilter extensions [41]. There are also new questions brought to modal logic from coalgebra, for example, how to best deal with infinite parameters $C$ in Table 1, see Friggens and Goldblatt [24].

**Going beyond rank 1** The original motivation in using Stone dualities was to understand logics of coalgebras for a functor $T$. We have seen that a logic for $T$ only needs axioms of rank 1. From this point of view, rank 1 is no restriction. And, of course, we can deal with axioms of rank $> 1$ in a trivial way: axioms of rank 1 determine a functor $T$ and hence a category $\mathsf{Coalg}(T)$, whereas the other axioms specify a subcategory of $\mathsf{Coalg}(T)$. So the question really is whether axioms not of rank 1 can be treated in a *uniform coalgebraic* way.

**Fixed-point logic** It is straightforward to extend a basic logic derived from $T$ by fixed-points as in $\mu$-calculus. But it is not clear at all whether a Stone duality based approach can help in better understanding fixed-point logics. Venema [71] and Kupke and Venema [43] introduce the notion of coalgebraic fixed point logic and show that $\mu$-calculus interpreted over different data structures such as words and trees can be treated uniformly in a coalgebraic framework.

# A Some Notions of Category Theory

A *category* $\mathcal{C}$ consists of a class of objects and has, for any two objects $A, B$, a set $\mathcal{C}(A, B)$ of arrows (or morphisms) from $A$ to $B$. Furthermore, arrows $f : A \to B, g : B \to C$ have a composition $g \circ f$ and for each object $A$ there is an identity arrow $\mathrm{id}_A$. Examples: The category $\mathsf{Set}$ with sets as objects and functions as arrows; $\mathsf{BA}$ with Boolean algebras and their homomorphisms; further, $\mathsf{Coalg}(T)$ and $\mathsf{Alg}(L)$.

An *isomorphism* is an arrow $f : A \to B$ for which there is a $g : B \to A$ with $f \circ g = \mathrm{id}_B, g \circ f = \mathrm{id}_A$.

A *covariant functor* between two categories $F : \mathcal{C} \to \mathcal{D}$ maps objects to objects and arrows $f : A \to B$ to $Ff : FA \to FB$, preserving identities and composition. Examples: the functors $T$ and $L$.

For each category $\mathcal{C}$ we have the *dual category* $\mathcal{C}^{\mathrm{op}}$ obtained from reversing the arrows. Example: Each functor $F : \mathcal{C} \to \mathcal{C}$ gives rise to a functor $F^{\mathrm{op}} : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}^{\mathrm{op}}$; the duality of algebras and coalgebras can now be stated as $\mathsf{Alg}(F^{\mathrm{op}}) = \mathsf{Coalg}(F)^{\mathrm{op}}$.

A *contravariant functor* $F : \mathcal{C} \to \mathcal{D}$ is a covariant functor $\mathcal{C}^{\mathrm{op}} \to \mathcal{D}$ or, equivalently, $\mathcal{C} \to \mathcal{D}^{\mathrm{op}}$, that is, it reverses the direction of the arrows. Example: $2^{-} : \mathsf{Set} \to \mathsf{BA}$ maps $f : X \to Y$ to $2^f = f^{-1} : 2^Y \to 2^X$.

Given functors $F, G : \mathcal{C} \to \mathcal{D}$, a *natural transformation* $\tau : F \to G$ consists of maps $\tau_A : FA \to GA$, $A$ in $\mathcal{C}$, such that for all $f : A \to A'$ we have $Gf \circ \tau_A = \tau_{A'} \circ f$. Example: the predicate liftings (6).

Given two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$, we say that $\mathcal{C}$ and $\mathcal{D}$ are *equivalent* if there are natural isomorphisms $\tau_C : C \to GFC$ and $\sigma_D : D \to FGD$. $\mathcal{C}$ and $\mathcal{D}$ are *dually equivalent* if $\mathcal{C}^{\mathrm{op}}$ and $\mathcal{D}$ are equivalent.

# References


[1] S. Abramsky. A Cook's Tour of the Finitary Non-Well-Founded Sets. Invited Lecture at BCTCS 1988.

[2] S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92, 1991.

[3] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51, 1991.

[4] S. Abramsky and A. Jung. Domain theory. In *Handbook of Logic in Computer Science*. OUP, 1994.

[5] P. Aczel. *Non-Well-Founded Sets.* CSLI, Stanford, 1988.

[6] F. Arbab and J. Rutten. A coinductive calculus of component connectors. In *WADT'02*, LNCS 2755, 2003.

[7] L. Barbosa. Towards a calculus of software components. *J. Univ. Comp. Sci.*, 9, 2003.

[8] F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats.* PhD thesis, Vrije Universiteit Amsterdam, 2004.

[9] F. Bartels, A. Sokolova, and E. de Vink. A hierarchy of probabilistic system types. *Theoret. Comput. Sci.*, 327, 2004.

[10] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* CUP, 2001.

[11] M. Bonsangue, B. Jacobs, and J. N. Kok. Duality beyond sober spaces: Topological spaces and observation frames. *Theoret. Comput. Sci.*, 151, 1995.

[12] M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *FoSSaCS'05*, LNCS 3441, 2005.

[13] M. Bonsangue and A. Kurz. Pi-calculus in logical form. Draft, 2006.

[14] M. Bonsangue and A. Kurz. Presenting functors by operations and equations. In *FoSSaCS'06*, LNCS 3921, 2006.

[15] M. M. Bonsangue. *Topological Dualities in Semantics.* PhD thesis, Vrije Universiteit Amsterdam, 1996.

[16] C. Brink and I. M. Rewitzky. *A Paradigm for Program Semantics: Power Structures and Duality.* Cambridge University Press, 2001.

[17] C. Cîrstea. On expressivity and compositionality in logics for coalgebras. In *CMCS'03*, ENTCS, 2003.

[18] C. Cîrstea and D. Pattinson. Modular construction of modal logics. In *CONCUR'04*, LNCS 3170, 2004.

[19] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order.* Cambridge University Press, 1990.

[20] J. de Bakker and E. de Vink. *Control Flow Semantics.* MIT Press, 1996.

[21] E. de Vink and J. Rutten. Bisimulation for probabilistic transition systems: A coalgebraic approach. *Theoret. Comput. Sci.*, 221, 1999.

[22] M. Escardó. Synthetic topology of data types and classical spaces. *ENTCS*, 87, 2004.

[23] M. Fiore, E. Moggi, and D. Sangiorgi. A fully-abstract model for the $\pi$-calculus. In *LICS 96*, 1996.

[24] D. Friggens and R. Goldblatt. A modal proof theory for final polynomial coalgebras. *Theoret. Comput. Sci.* To appear.

[25] R. Goldblatt. Metamathematics of modal logic I. *Reports on Mathematical Logic*, 6, 1976.

[26] R. Goldblatt. Observational ultraproducts of polynomial coalgebras. *Ann. Pure Appl. Logic*, 123, 2003.

[27] H. P. Gumm. Elements of the general theory of coalgebras. LUATCS'99, 1999.

[28] H. Hansen and C. Kupke. A coalgebraic perspective on monotone modal logic. In *CMCS'04*, ENTCS 106, 2004.

[29] B. Jacobs. *Introduction to Coalgebra. Towards Mathematics of States and Observations.* In preparation, draft electronically available.

[30] B. Jacobs. Objects and classes, co-algebraically. In *Object-Orientation with Parallelism and Persistence*. Kluwer, 1996.

[31] B. Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theor. Inform. Appl.*, 35, 2001.

[32] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997.

[33] P. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.

[34] B. Jónsson and A. Tarski. Boolean algebras with operators, part 1. *Amer. J. Math.*, 73, 1951.

[35] B. Jónsson and A. Tarski. Boolean algebras with operators, part 2. *American Journal of Mathematics*, 74, 1952.

[36] C. Karp. *Languages with Expressions of Infinite Length*. North-Holland, 1964.

[37] B. Klin. *An Abstract Coalgebraic Approach to Process Equivalence for Well-Behaved Operational Semantics*. PhD thesis, University of Aarhus, 2004.

[38] B. Klin. The least fibred lifting and the expressivity of coalgebraic modal logic. In *CALCO'05*, 2005.

[39] J. Komenda and J. van Schuppen. Decentralized supervisory control with coalgebra. In *ECC'03*, 2003.

[40] C. Kupke, A. Kurz, and D. Pattinson. Algebraic semantics for coalgebraic logics. In *CMCS'04*, ENTCS 106, 2004.

[41] C. Kupke, A. Kurz, and D. Pattinson. Ultrafilter extensions of coalgebras. In *CALCO'05*, LNCS 3629, 2005.

[42] C. Kupke, A. Kurz, and Y. Venema. Stone coalgebras. *Theoret. Comput. Sci.*, 327, 2004.

[43] C. Kupke and Y. Venema. Closure properties of coalgebra automata. In *LICS'05*, 2005.

[44] A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, LMU, 2000.

[45] A. Kurz. A co-variety-theorem for modal logic. In *Advances in Modal Logic 2*. CSLI, 2001.

[46] A. Kurz. *Coalgebras and Modal Logic - Course Notes for ESSLLI 2001*. University of Helsinki, 2001.

[47] A. Kurz. Specifying coalgebras with modal logic. *Theoret. Comput. Sci.*, 260, 2001.

[48] A. Kurz and J. Rosický. Operations and equations for coalgebras. *Math. Structures Comput. Sci.*, 15, 2005.

[49] S. Milius and L. Moss. The category theoretic solution of recursive program schemes. 2006. Draft.

[50] L. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96, 1999.

[51] L. Moss and I. Viglizzo. Final coalgebras for functors on measurable spaces. *Inform. and Comput.* To appear.

[52] T. Mossakowski, H. Reichel, M. Roggenbach, and L. Schröder. Algebraic-coalgebraic specification in CoCASL. In *WADT'02*, LNCS 2755, 2003.

[53] A. Palmigiano. A coalgebraic view on positive modal logic. *Theoret. Comput. Sci.*, 327, 2004.

[54] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoret. Comput. Sci.*, 309, 2003.

[55] D. Pattinson. An introduction to the theory of coalgebras, 2003. Course Notes for NASSLLI 2003.

[56] D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame Journal of Formal Logic*, 45, 2004.

[57] H. Reichel. An approach to object semantics based on terminal co-algebras. *Math. Structures Comput. Sci.*, 5, 1995.

[58] M. Rößiger. From modal logic to terminal coalgebras. *Theoret. Comput. Sci.*, 260, 2001.

[59] J. Rothe, H. Tews, and B. Jacobs. The coalgebraic class specification language CCSL. *J. Univ. Comp. Sci.*, 7, 2001.

[60] J. Rutten. Automata and coinduction - an exercise in coalgebra. In *CONCUR'98*, LNCS 1466, 1998.

[61] J. Rutten. Universal coalgebra: A theory of systems. *Theoret. Comput. Sci.*, 249, 2000.

[62] J. Rutten. Coinductive counting with weighted automata. *J. Autom. Lang. Comb.*, 8, 2003.

[63] J. Rutten. A tutorial on coinductive stream calculus and signal flow graphs. *Theoret. Comput. Sci.*, 343, 2005.

[64] L. Schröder. Expressivity of Coalgebraic Modal Logic: The Limits and Beyond. In *FoSSaCS'05*, LNCS 3441, 2005.

[65] M. Smyth. Topology. In *Handbook of Logic in Computer Science*. OUP, 1993.

[66] I. Stark. A fully-abstract domain model for the $\pi$-calculus. In *LICS 96*, 1996.

[67] M. H. Stone. The theory of representations for boolean algebras. *Trans. Amer. Math. Soc*, 40, 1936.

[68] M. H. Stone. Topological representation of distributive lattices and Brouwerian lattices. *Časopis pěst. mat. fys.*, 67, 1937.

[69] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *LICS'97*, 1997.

[70] J. van den Berg and B. Jacobs. The loop compiler for java and jml. In *TACAS'01*, LNCS 2031, 2001.

[71] Y. Venema. Automata and fixed point logics: a coalgebraic perspective. *Inform. and Comput.* To appear.

[72] Y. Venema. *Handbook of Modal Logic*, chapter Algebras and Coalgebras. To appear. Electronically available.

[73] S. J. Vickers. *Topology Via Logic.* CUP, 1989.