Editorial

Coalgebra and Logic: A Brief Overview

1 Introduction

Introduced to computer science by Aczel in the late 1980s, coalgebras provide modern mathematical foundations of the theory of evolving systems. The coalgebraic viewpoint combines wide applicability and mathematical simplicity: since every set functor determines its own type of coalgebra, notions, properties and results of state-based systems can be uniformly explained just in terms of properties of their type functors. This applies to systems as diverse as streams, probabilistic transition systems, automata, Kripke structures and neighbourhood frames.

For instance, the general theory of coalgebra makes uniformly available, for each functor $T$, canonical notions of coalgebraic homomorphism and of bisimilarity between states of coalgebras. These notions correspond to the natural ones that were independently developed in each specific setting. Bisimilarity can be classified by the final coalgebra, while the latter structure also gives rise to the proof and definition principles of coinduction and corecursion. These are as fundamental for infinite structures (such as the behaviour of a non-terminating system) as induction is for finitely built structures (such as the natural numbers, or the terms of an algebra). In fact, algebra and coalgebra, as well as induction and coinduction, are dual notions, in a precise category theoretic sense. This duality provides a deep connection between finite terms and infinitely ongoing processes.

Logic enters the picture if we want to design specification languages and derivation systems, in order to describe and reason about the kind of behaviour modelled by coalgebras. The basic observation that Kripke frames are coalgebras, combined with an analysis of the intricate connections between bisimilarity and modal logic, led to the following idea, pioneered by Barwise and Moss. Coalgebraic languages, uniformly defined using the functor $T$ as the only syntactic parameter, can be regarded as generalizing modal logic as the logic of transition systems, bisimulation invariance being their main desideratum. The earlier mentioned algebra/coalgebra duality is again fundamental, especially when we look at complete derivation systems for logical languages of the appropriate type. Indeed, the duality between Lindenbaum algebras and canonical models, traditionally all-important in modal completeness proofs, extends category-theoretically to the duality between initial $L$-algebras and final $T$-coalgebras (where $L$ is the type functor of the modal logic of the $T$-coalgebras).

In the next two sections we will try to explain the above in more detail, after which Section 4 provides a number of reasons to be interested in the coalgebraic perspective. In Section 5, we introduce the present special issue of the *Journal of Logic and Computation*; our discussion includes a short account of the contributed papers. For obvious reasons of space limitations, this introduction cannot serve as either a proper introduction to the area of coalgebra or a complete guide to the literature. We have therefore decided not to give any references here, but instead provide a bibliography of readily accessible introductory texts on coalgebras.

2 Basic notions in coalgebra

*Coalgebras:* Intuitively, the functor $T$ specifies the one-step behaviours a system could possibly engage in. To give a precise but preliminary definition: for any operation $T$ on sets, we say that a $T$-coalgebra is a set $X$ equipped with a function $\xi: X \to TX$. For example, consider $TX = 2 \times X^2$...
where 2 represents some two-element set and Σ an arbitrary set called the input alphabet. Then ξ determines for each state $x \in X$ two things: an element of 2, specifying whether $x$ is accepting or not, and a function $\Sigma \rightarrow X$, specifying for each input letter from $\Sigma$ a successor of $x$ in $X$. So we see that $T$-coalgebras are nothing but deterministic automata.

A natural idea is now to vary $T$ and investigate what happens. Consider $T = \mathcal{P}$ where $\mathcal{P}X = \{Y \mid Y \subseteq X\}$. Then a map $X \rightarrow TX$ can be seen as a binary relation on $X$, thus providing us with a Kripke frame. To turn Kripke frames into Kripke models, in addition we label the states with sets of atomic propositions (taken from a set $\text{AtProp}$), giving rise to $TX = 2^{\text{AtProp}} \times \mathcal{P}X$. Labelling instead the transitions (with letters from the set $\Sigma$) corresponds to $TX = \mathcal{P}(\Sigma \times X)$. Similarly, when $T$ is the distribution functor $\mathcal{P}$, the $T$-coalgebras model probabilistic transition systems.

The notion of coalgebra is general enough to encompass structures that are not usually perceived as relational structures or transition systems. For $TX = 2^2$, a coalgebra $\xi$ rather resembles a topological space (mapping a point to its collection of neighbourhoods), for $TX = (\mathcal{P}X)^2$ one obtains Chellas’s conditional frames.

**Coalgebra morphisms:** In all of the examples above, $T$ can be extended so that it not only maps sets to sets but also functions to functions. In other words, $T$ is a functor on the category $\text{Set}$ of sets and functions. This allows us to define a coalgebra (homo)morphism $(X, \xi) \rightarrow (X', \xi')$ as a map $f : X \rightarrow X'$ respecting $T$-structure, that is, $Tf \circ \xi = \xi' \circ f$:

$$
\begin{array}{ccc}
X & \xrightarrow{f} & X' \\
\downarrow & & \downarrow \\
TX & \xrightarrow{\xi} & TX'
\end{array}
$$

To check on the examples, let us note that in case of $TX = 2 \times X^2$, coalgebra morphisms are automata morphisms and in case of $T = \mathcal{P}$, coalgebra morphisms are the bounded morphisms (or $p$-morphisms) from modal logic. In other examples as well, coalgebra morphisms coincide with the natural notions independently developed in each specific setting.

**Behavioural equivalence and bisimulations:** If we think of $T$ as the interface specifying the observations that can be made of a system, then coalgebra morphisms allow to identify states but have to preserve the observations. One therefore defines two states in a coalgebra to be *behaviourally equivalent* if they can be identified by some coalgebra morphism. In the case of deterministic automata, this means that the two states induce the same accepted language. In case of Kripke frames or models this amounts to bisimilarity.

One can also define bisimulations. A bisimulation between two coalgebras $(X, \xi)$ and $(X', \xi')$ is a relation $R \subseteq X \times X'$ such that there is a map $R \rightarrow TR$ making the two projections $R \rightarrow X$ and $R \rightarrow X'$ into coalgebra morphisms. If $T$ satisfies a technical condition (preservation of weak pullbacks), then two states are related by a bisimulation iff they are behaviourally equivalent; otherwise behavioural equivalence seems to be the right notion.

**Final coalgebra:** The final coalgebra classifies behavioural equivalence: if $(Z, \zeta)$ is final, then there is, by definition, a unique morphism $!: (X, \xi) \rightarrow (Z, \zeta)$. From this it easily follows that $!(x) = !(y)$ if $x$ and $y$ are behaviourally equivalent.

The final coalgebra is always a solution of the ‘domain equation’ $Z \cong TZ$. If $T$ is a set functor, this solution always exists. ($Z$ may not be a set but a proper class; but for most applications this does not matter as $Z$ still classifies behaviours.) This shows that, in many cases, domain equations can be solved over sets, i.e. no additional structure (such as complete partial orders) is needed.
Final coalgebras are not only useful to reason about behavioural equivalence: in many cases, the final coalgebra carries some additional structure that is meaningful in the specific setting of the coalgebraic type. For instance, in the case of deterministic automata over an alphabet $\Sigma$, the final coalgebra can be based on the set of all languages over $\Sigma$.

**Coinduction:** In the same (or, rather, dual) way as induction is associated with initial (or free) algebras, a final coalgebra gives rise to the proof and definition principle of co-induction.

For example, (infinite) streams over the natural numbers form a final coalgebra $\mathbb{N}^\omega \to \mathbb{N} \times \mathbb{N}^\omega$, mapping a stream $\sigma$ to $(\sigma(0), \sigma')$, where $\sigma'$ is the stream given by $\sigma'(n) = \sigma(n+1)$. Then $(\sigma + \tau)(0) = \sigma(0) + \tau(0)$ and $(\sigma + \tau)' = \sigma' + \tau'$ is a coinductive (but not inductive) definition of the function which inputs two streams $\sigma$ and $\tau$ and adds them componentwise.

Since bisimilarity is identity on the final coalgebra, one can now prove equality between streams by exhibiting a bisimulation. This is a powerful principle, as the task of proving equality of two complete infinite behaviours is reduced to checking one-step behaviours only.

**Duality:** Although algebra and coalgebra are dual concepts, the theory of coalgebras cannot be obtained from universal algebra simply using categorical duality: the point is that coalgebras over $\text{Set}$ are not dual to algebras over $\text{Set}$, but to algebras over the opposite category, $\text{Set}^{\text{op}}$. (More precisely, for a functor $T : \mathcal{C} \to \mathcal{C}$, the category of $T$-coalgebras over $\mathcal{C}$ is dual to the category of algebras for the functor $T^{\text{op}} : \mathcal{C}^{\text{op}} \to \mathcal{C}^{\text{op}}$.) As a consequence, there are significant differences between universal coalgebra and universal algebra. Nevertheless, category theoretic duality provides a powerful heuristic. For example, the well-known isomorphism theorems and Birkhoff’s and Reiterman’s HSP theorems have coalgebraic analogues.

## 3 Coalgebraic logic

Given a functor $T$, what are good formal languages to describe $T$-coalgebraic behaviour? Does there exist such a language for any functor $T$? And if so, can we find sound and complete reasoning systems for these languages? Can the theory of logics for coalgebras be developed uniformly in the parameter $T$? Is it possible to treat complicated functors composed of simpler ones in a modular way? Such questions arise naturally from the desire to formally describe and reason about behaviour.

The paradigmatic example of a coalgebra is $X \to TX$. In this case we know a good logic, namely modal logic. There are different, but closely related ways to generalize from there to coalgebras.

**Moss’ coalgebraic logic:** was the first suggestion. Here the set of formulas $\mathcal{L}$ is closed under the following clause introducing a modal operator $\Box$: if $\alpha \in T(\mathcal{L})$ then $\Box \alpha \in \mathcal{L}$. In the case $T = \mathcal{P}$ one can write, for instance, $\Box [a_1, a_2]$ where $a_i$ are formulas. $\Box [a_1, a_2]$ will be equivalent to $\alpha(a_1 \lor a_2) \land \Box a_1 \land \Box a_2$. In general, the semantics of $\mathcal{V}$ w.r.t. a Kripke frame $(X, \xi)$ is given by $x \Vdash \alpha$ iff $(\xi(x), \alpha) \in \mathcal{T}(\mathcal{V})$, where $\mathcal{T}(\mathcal{V}) \subseteq TX \times \mathcal{L}$ is the (Egli–Milner) lifting of the binary relation $\models \subseteq X \times \mathcal{L}$. This approach generalizes well to any functor $T : \text{Set} \to \text{Set}$ preserving weak pullbacks. (The point is that $T$ preserves weak pullbacks iff its lifting $\mathcal{T}$ preserves composition of relations.)

**Predicate liftings:** Let us start with a modal logic having one unary modal operator $\Box$. If the logic is not required to satisfy further conditions such as monotonicity or normality, the appropriate semantic structure are neighbourhood frames, that is, coalgebras $\xi : X \to 2^X$. Thus, each natural transformation $TX \to 2^X$ gives rise to a modal operator. Such a natural transformation is called a predicate lifting as it can also be written as $2^X \to 2^{TX}$ (predicates over $X$ are lifted to predicates over $TX$). To summarize, for each collection $\lambda$ of $n$-ary predicate liftings $(2^X)^n \to 2^{TX}$ we obtain a modal logic for $T$-coalgebras with the corresponding $n$-ary modal operators.
Editorial

Stone duality: Think of the category $\text{BA}$ of Boolean algebras as providing a base logic for $\text{Set}$. Expressed categorically: given a set $X$, the powerset $PX$ is a Boolean algebra (and conversely, given a Boolean algebra $B$, we may let $SB$ denote the set of ultrafilters of $B$). To extend this to $T$-coalgebras, a logic for $T$-coalgebras is given by a functor $L: \text{BA} \to \text{BA}$, together with a natural transformation $\delta: LP \to PT$. The connection between all these notions can be nicely depicted in the following diagram.

\[
\begin{array}{ccc}
\text{Coalg}(T) & \xrightarrow{(\cdot)^+} & \text{Alg}(L) \\
\downarrow & & \downarrow \\
T & \xrightarrow{p} & \text{BA} \\
\downarrow & & \downarrow \\
\text{Set} & \xrightarrow{s} & L
\end{array}
\]

Here the coalgebraic logic is incarnated by the functor $(\cdot)^+$ which maps a coalgebra $(X, \xi): X \to TX$ to its complex algebra $(X, \xi)^+$, given as the structure $\left(\text{PX}, (P\xi \circ \delta_X)^\circ: LPX \to PX, \right)$, where $P\xi: PTX \to PX$ is the inverse of the coalgebra map $\xi$.

The relationship of the three approaches is as follows. Denote by $F$ the left-adjoint of the forgetful functor $U: \text{BA} \to \text{Set}$; in other words, $FS$ denotes the free Boolean algebra generated by the set $S$. Now, if we have a set of predicate liftings $\lambda \in \Lambda_1$ with arities $r_\lambda$, then the functor corresponding to the logic of predicate liftings is given by $L\lambda = F \coprod_{\lambda \in \Lambda_1} (UA)^{r_\lambda}$ (with $\coprod$ denoting coproduct). And in the case of Moss’ logic (with Boolean connectives) it is given by $L\lambda = FTUA$. In other words, the latter approach generalizes the first two. In fact, it allows for many other instances as well.

4 Why coalgebra?

We believe that a coalgebraic perspective on evolving state-based systems is of interest, not only to computer scientists, but also to logicians and mathematicians. Below we list a number of reasons why.

- **Uniform** treatment of different types of systems. For example, one can establish that satisfiability of coalgebraic logic is in PSPACE and that complete coalgebraic logics have the finite model property. Furthermore analogues or generalizations of Birkhoff’s HSP theorem and of the theorems of Jónsson–Tarski and Goldblatt–Thomason can be shown. Finally, the uniformity of the metatheory might well translate into software tools that are easier to design, maintain, and to implement (a generic satisfiability solver has already been developed).

- **Modularity**: We have seen examples of different functors $T$. These can be further combined using composition of functors, product and coproduct (choice). Therefore one investigates classes of functors, given by a collection of basic functors and closed under various type constructors. Theorems and algorithms for basic types can then be lifted to arbitrarily complex combinations.

- **One-step analysis**: Coalgebraic analysis of dynamic systems seems is particularly successful where the class of all complete behaviours is determined by the possible one-step behaviours. This is, e.g. the basis of coinduction. It also plays an important role in the uniform metatheory briefly mentioned above and in applications to automata and fixpoint logics.

- **Stone duality** is an important technique in Modal Logic. The algebra-coalgebra duality can be used to extend given Stone dualities and thus widen the scope of this technique. Furthermore, this technique can be used to derive new logics for systems that have a coalgebraic semantics. This has recently been worked out for the $\pi$-calculus.
Applications to theoretical computer science: We list some areas that coalgebras have been applied to: recursive program schemes, $\mu$-calculus, behavioural differential equations with applications to combinatorics or the specification of distributed systems, automata theory, regular languages, Kleene algebras, coinductive methods in theorem proving, structural operational semantics, process algebra, control theory, discrete event structures.

Further connections: Applications in Mathematics and Logic have been developed much less than applications to Computer Science. But Coalgebra as an area naturally overlaps with Universal Algebra, Modal Logic, Algebraic Logic, Duality Theory and Domain Theory. We believe that this presents many opportunities for exciting future research.

5 The present volume

In this section, we briefly describe the means, motives and opportunity that led us to editing this special issue of the *Journal of Logic and Computation* on Coalgebra and Logic.

To start with the motives, in the previous section we provided many reasons to be interested in the area of Coalgebra and Logic. We also believe that the state reached by this arising field is worth being reported, both because of its outlining coherence and beauty and because we hope that our initiative may foster further advances. The opportunity came to us as the coincidence of Yde Venema starting as this journal’s corner editor for the area of Algebraic and Coalgebraic Logic, and the three of us being the organizers of two workshops on ‘Logics for Coalgebras’ and ‘Coalgebraic Logic’, held on 10–11 May and 10–11 August 2007 in Amsterdam and Oxford, respectively. At these workshops we made a first informal round of proposals for this special issue and registered the enthusiastic adhesions of many of the leading researchers active in this arising field. This, finally, provided us with the means for the special issue. More specifically, below we briefly introduce its six contributions.

The contributions

*Presentation of set functors*: a coalgebraic perspective (Adámek, Gumm and Trnková): It is well-known that every accessible set functor $F$ is a quotient of a polynomial one, i.e. $F$ has a presentation by operations and equations. This paper characterizes several important properties of functors by the syntactic form of the equations, for example, it characterizes the accessible weak-pullback preserving functors. The paper might also a good starting point to learn about set-functors in general.

*Vietoris bisimulations* (Bezhanishvili, Fontaine and Venema): The notions of Vietoris bisimulation and bisimilarity between descriptive modal models are introduced. Bisimilarity is shown to coincide with Kripke bisimilarity, behavioural and modal equivalence but not with Aczel–Mendler bisimilarity. As a consequence, the Vietoris functor does not preserve weak pullbacks.

*Exemplaric expressivity of modal logics* (Jacobs and Sokolova): In a number of different settings, some coalgebraic modal logics are shown to be expressive in the sense that states are behaviourally equivalent iff they satisfy the same formulas. The approach is based on the framework that we discussed above as ‘Stone duality’, and focuses on the injectivity of the natural transformation mentioned there. The contribution of the paper, apart from some new/sharper expressivity results, is the wide applicability of the uniform proof method.

*Deduction systems for coalgebras over measurable spaces* (Goldblatt): A theory of infinitary deduction systems is developed for the modal logic of coalgebras for measurable polynomial functors on measurable spaces. Completeness of these systems is shown via a deductive construction...
Editorial

of canonical spaces and coalgebras. The semantic consequence relation for the logic of any measurable polynomial functor is characterized as the least one satisfying Lindenbaum's Lemma.

A note on expressive coalgebraic logics for finitary set functors (Moss): The final coalgebra of an arbitrary finitary set functor can be based on a subset of the $\omega$-th term in the so-called terminal sequence. This construction is used to define an expressive coalgebraic language of which the formulas correspond to the elements of the finitary approximations in the terminal sequence.

Rank-1 modal logics are coalgebraic (Schröder and Pattinson): Rank-1 logics are described by axioms in which each propositional variable is under the scope of precisely one modal operator. The paper shows that all modal logics of rank-1 have a coalgebraic semantics, which is strongly complete with respect to the local consequence relation. Moreover, an interesting application of Coalgebra to Deontic Logic is presented.

Bibliography: Introductions to Coalgebra and Logic


Alexander Kurz
Department of Computer Science, University of Leicester
E-mail: kurz@mcs.le.ac.uk

Alessandra Palmigiano and Yde Venema
Institute for Logic, Language and Computation, University of Amsterdam,
E-mail: A.Palmigiano@uva.nl; Y.Venema@uva.nl