Logics Admitting Final Semantics

Alexander Kurz

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
e-mail: kurz@cwi.nl

Abstract. A logic for coalgebras is said to admit final semantics iff—up to some technical requirements—all definable classes contain a fully abstract final coalgebra. It is shown that a logic admits final semantics iff the formulas of the logic are preserved under coproducts (disjoint unions) and quotients (homomorphic images).

Introduction

In the last few years it became clear that a great variety of state-based dynamical systems, like transition systems, automata, process calculi can be captured uniformly as coalgebras, see [24, 7] for an introduction to coalgebras and [6, 8, 20, 3] for recent developments. One of the important features of coalgebras is that under rather weak conditions, categories of coalgebras have final (or cofree) coalgebras. This allows to give final semantics to systems and to use coinduction as a proof and definition principle.

In the view of coalgebras as systems, logics for coalgebras are specification languages for systems. Examples of different approaches to logics for coalgebras include [17, 14, 22, 10, 4, 18]. These examples show that—due to the generality provided by functors as signatures—there is no uniform syntax for specification languages for coalgebras.

The purpose of this paper is not to develop a new logical syntax for coalgebras (although we make the proposal to use modal logics with a global diamond). Rather, we want to take an abstract approach. To this end, we consider as a logic for coalgebras any pair $(\mathcal{L}, \models)$ consisting of a class of formulas $\mathcal{L}$ and a satisfaction relation $\models$ between coalgebras and formulas, subject to the condition that definable classes are closed under isomorphism. We then ask the question whether we can characterise those logics for coalgebras which admit final semantics.

The definition of a logic admitting final semantics as well as the proof of our characterisation theorem follow the work of Mahr and Makowsky [15] and Tarlecki [26] who characterised logics for algebras admitting initial semantics.

The first section covers preliminaries, the second gives a characterisation of logics admitting final semantics. The third section points out that logics admitting final semantics may be quite stronger than those cited as examples above and makes two suggestions that can be used to strengthen (these and others) logics in a way that they still admit final semantics.

1 Preliminaries

We first review coalgebras and final semantics and then briefly discuss logics for coalgebras. For more details on coalgebras and modal logic we refer to [24] and [2], respectively.

1.1 Coalgebras and Final Semantics

A *coalgebra* is given wrt. a base category $\mathcal{X}$ and an endofunctor (also called *signature*) $\Sigma : \mathcal{X} \to \mathcal{X}$. A Σ-coalgebra (X, ξ) consists of a carrier $X \in \mathcal{X}$ and an arrow $\xi : X \to \Sigma X$. Σ-coalgebras form a category $\mathsf{Coalg}(\Sigma)$ where a coalgebra morphism $f : (X, \xi) \to (X', \xi')$ is an arrow $f : X \to X'$ in $\mathcal{X}$ such that $\Sigma f \circ \xi = \xi' \circ f$. In the following we assume $\mathcal{X} = \mathsf{Set}$, the category of sets and functions. Given a coalgebra (X, ξ), we call the elements of X *states* and ξ the *(transition) structure*. We sometimes denote a coalgebra (X, ξ) by its structure ξ.

We mention only the following paradigmatic example in which coalgebras appear as transition systems (or as Kripke frames, in the terminology of modal logic).

Example 1 (Kripke frames). Consider the functor $\Sigma X = \mathcal{P}_\omega X$ where $\mathcal{P}_\omega$ denotes the finite powerset.[1] Then $\mathcal{P}_\omega$-coalgebras $\xi : X \to \mathcal{P}X$ are image-finite (ie., finitely branching) *Kripke frames*: For $x \in X$, $\xi(x)$ is the set of successors of x. Morphisms are functional bisimulations, also known as p-morphisms or bounded morphisms.

A Σ-coalgebra Z is *final* iff for all $\mathsf{X} \in \mathsf{Coalg}(\Sigma)$ there is a unique morphism $!_\mathsf{X} : \mathsf{X} \to \mathsf{Z}$. The interest in the final coalgebra comes from the following definition of behavioural equivalence. Given two coalgebras $\mathsf{X} = (X, \xi)$, $\mathsf{Y} = (Y, \nu)$ one says that $x \in X$ and $y \in Y$ are behaviourally equivalent, written $(\mathsf{X}, x) \sim (\mathsf{Y}, y)$, iff $!_\mathsf{X}(x) = !_\mathsf{Y}(y)$.

We call a pair (X, x) a **process** and x its initial state. Every element of the final coalgebra represents a class of behaviourally equivalent processes. We call the elements of the final coalgebra **behaviours** and $!_\mathsf{X}(x)$ the behaviour of (X, x). The **final semantics** of a coalgebra X is given by the unique morphism $!_\mathsf{X} : \mathsf{X} \to \mathsf{Z}$ (assigning to each process in X its behaviour).

Example 2 (Kripke frames, cont'd). Given two image-finite Kripke frames $\mathsf{X} = (X, \xi)$, $\mathsf{Y} = (Y, \nu)$, and $x \in X$, $y \in Y$ then $(\mathsf{X}, x) \sim (\mathsf{Y}, y)$ iff (X, x) and (Y, y) are bisimilar, that is, iff there is a relation $R \subseteq X \times Y$ with $x\,R\,y$ and

$$
\begin{aligned}
x\,R\,y \ \&\ x' \in \xi(x) \ &\Rightarrow\ \exists y' \in \nu(y) \ \&\ x'\,R\,y', \\
x\,R\,y \ \&\ y' \in \nu(y) \ &\Rightarrow\ \exists x' \in \xi(x) \ \&\ x'\,R\,y'.
\end{aligned}
$$

[1] That is, $\mathcal{P}_\omega(X) = \{A \subseteq X : A \text{ finite}\}$. On functions $\mathcal{P}$ is defined as follows: given $f : X \to Y$, $\mathcal{P}_\omega f = \lambda A \in \mathcal{P}X.\{f(a) : a \in A\}$.

The notion of a final coalgebra can be extended to incorporate additional observations of the states as follows. Let C be a set, called a set of colours, and $\mathsf{X} = (X, \xi)$ be a coalgebra. A mapping $v : X \to C$ is called a colouring of the states. X together with v gives rise to a $(\Sigma \times C)$-coalgebra $\langle \xi, v\rangle : X \to \Sigma X \times C$. We write (X, v), or $\langle \xi, v\rangle$, for a $(\Sigma \times C)$-coalgebra consisting of a Σ-coalgebra $\mathsf{X} = (X, \xi)$ and a colouring v. Triples $((\mathsf{X}, v), x)$ for $x \in X$ are called **coloured processes** and abbreviated as (X, v, x).

Example 3 (Kripke models). Let $\Sigma = \mathcal{P}_\omega$ and $C = \mathcal{P}\mathbf{P}$ where $\mathbf{P}$ is a set of propositional variables. Then $(\Sigma \times C)$-coalgebras $\langle \xi, v\rangle : X \to \mathcal{P}_\omega X \times \mathcal{P}\mathbf{P}$ are *Kripke models*: For $x \in X$, $\xi(x)$ is the set of successors of x and $v(x)$ is the set of propositions holding in x. As for Kripke frames, $(\Sigma \times C)$-morphism are functional bisimulations (respecting, this time, the valuations of propositional variables).

We can think of the colouring v as allowing additional observations. Accordingly, a notion of behavioural equivalence is of interest that takes into account these additional observations. This is provided by the final $(\Sigma \times C)$-coalgebra $\langle \zeta_C, \varepsilon_C\rangle : Z_C \to \Sigma Z_C \times C$. We call ζ_C the **cofree Σ-coalgebra over C**.

Definition 1 (having cofree coalgebras). *We say that* $\mathsf{Coalg}(\Sigma)$ *has cofree coalgebras iff for all* $C \in \mathsf{Set}$ *a final coalgebra exists in* $\mathsf{Coalg}(\Sigma \times C)$.

Remark 1. The standard way to establish that for a given functor Σ the category $\mathsf{Coalg}(\Sigma)$ has a final coalgebra is to show that Σ is bounded (see [24]). In that case $\Sigma \times C$ is also bounded and $\mathsf{Coalg}(\Sigma)$ has cofree coalgebras as well. Since the class of bounded functors seems to include the signatures which are important in specifying systems,[2] requiring cofree coalgebras is not much stronger than requiring only a final coalgebra.

Nevertheless, there are examples of categories $\mathsf{Coalg}(\Sigma)$ which don't have all cofree coalgebras but still a final one. The use of the functor $\mathcal{P}_{ne}$ in the following example was suggested to the author by Falk Bartels.

Example. Let $\mathcal{P}_{ne}$ be the functor mapping a set to the set of its non-empty subsets. $\mathsf{Coalg}(\mathcal{P}_{ne})$ has $(\{*\}, \mathrm{id})$ as a final coalgebra. But $\mathsf{Coalg}(\mathcal{P}_{ne})$ does not have a coalgebra cofree over a two element set 2. This follows from the fact that a final $(\mathcal{P}_{ne} \times 2)$-coalgebra can not exist due to cardinality reasons (same argument as the one showing that $\mathsf{Coalg}(\mathcal{P})$ has no final coalgebra).

We conclude this subsection with two more definitions needed later. First, we note that coalgebras for signatures $\Sigma \times C$ and $\Sigma \times D$ are related as follows.

Definition 2 (the functor $\bar\lambda$). *Given a mapping $\lambda : C \to D$ we write $\bar\lambda$ for the functor*

$$\begin{aligned} \bar\lambda : \mathsf{Coalg}(\Sigma \times C) &\to \mathsf{Coalg}(\Sigma \times D) \\ \langle \xi, v\rangle &\mapsto \langle \xi, \lambda \circ v\rangle \end{aligned}$$

[2] $\mathsf{Coalg}(\mathcal{P})$, the category of coalgebras for the powerset functor, does not have a final coalgebra. But $\mathsf{Coalg}(\mathcal{P}_\kappa)$, where the cardinality of the subsets is restricted to be smaller than some cardinal κ, has cofree coalgebras.

where $\xi : X \to \Sigma X$ and $v : X \to C$. On morphisms, $\bar{\lambda}$ is given by $\bar{\lambda}(f) = f$.

Finally, a coalgebra is said to be fully abstract iff it has no proper quotient. In case that a final coalgebra exists, this is equivalent to being a subcoalgebra of the final coalgebra.

Definition 3 (fully abstract final coalgebras). X *is a final coalgebra in* $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma)$ *iff* $\mathsf{X} \in \mathcal{B}$ *and for all* $\mathsf{Y} \in \mathcal{B}$ *there is a unique morphism* $\mathsf{Y} \to \mathsf{X}$. *Assuming that* $\mathsf{Coalg}(\Sigma)$ *has a final coalgebra, we call* X *fully abstract iff* X *is a subcoalgebra of the final Σ-coalgebra.*

1.2 Logics for Coalgebras

Recalling Definition 2, we begin with

Definition 4 (logic for coalgebras). *Let* $\Sigma : \mathsf{Set} \to \mathsf{Set}$ *be a functor. A logic for Σ-coalgebras* $\mathcal{L} = (\mathcal{L}_C, \models_C)_{C \in \mathsf{Set}}$ *consists of classes* $\mathcal{L}_C$ *and satisfaction relations* $\models_C \subseteq \mathsf{Coalg}(\Sigma \times C) \times \mathcal{L}_C$ *for all* $C \in \mathsf{Set}$ *and translations of formulas* $\lambda^* : \mathcal{L}_D \to \mathcal{L}_C$ *for all mappings* $\lambda : C \to D$. *This data has to satisfy for all* $\mathsf{X} \in \mathsf{Coalg}(\Sigma \times C)$ *and* $\varphi \in \mathcal{L}_D$

$$\mathsf{X} \models \lambda^*(\varphi) \quad \iff \quad \bar{\lambda}(\mathsf{X}) \models \varphi. \tag{1}$$

Moreover, we require $(\lambda_2 \circ \lambda_1)^* = \lambda_1^* \circ \lambda_2^*$, $(\mathrm{id}_C)^* = \mathrm{id}_{\mathcal{L}_C}$, *and* $\forall \varphi \in \mathcal{L}_C : \mathsf{X} \models \varphi \Leftrightarrow \mathsf{X}' \models \varphi$ *for isomorphic $(\Sigma \times C)$-coalgebras* $\mathsf{X} \cong \mathsf{X}'$.

A class $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma \times C)$ *is called $\mathcal{L}$-definable, or also $\mathcal{L}_C$-definable, iff there is* $\Phi \subseteq \mathcal{L}_C$ *such that* $\mathcal{B} = \{\mathsf{X} \in \mathsf{Coalg}(\Sigma \times C) : \mathsf{X} \models_C \varphi$ *for all* $\varphi \in \Phi\}$.

Remark 2. 1. The simpler notion of a logic for Σ-coalgebras as a pair $(\mathcal{L}, \models)$ where $\mathcal{L}$ is a class and $\models$ is a relation $\models \subseteq \mathsf{Coalg}(\Sigma) \times \mathcal{L}$ is a special case. Indeed, $(\mathcal{L}, \models)$ can be considered as a logic $(\mathcal{L}_C, \models_C)_{C \in \mathsf{Set}}$ as follows. Let $\mathcal{L}_C = \mathcal{L}$, $\langle \xi, v \rangle \models_C \varphi \;\Leftrightarrow\; \xi \models \varphi$, and $\lambda^*(\varphi) = \varphi$ for all $C, D \in \mathsf{Set}$, $\xi : X \to \Sigma X$, $v : X \to C$, $\lambda : C \to D$, $\varphi \in \mathcal{L}$. Conversely, any logic for Σ-coalgebras $(\mathcal{L}_C, \models_C)_{C \in \mathsf{Set}}$ gives rise to the pair $(\mathcal{L}, \models)$ defined as $\mathcal{L} = \mathcal{L}_1$, $\models = \models_1$ where 1 is some one-element set.

2. Condition (1) ensures that if a class $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma \times C)$ is $\mathcal{L}$-definable then $\bar{\lambda}^{-1}(\mathcal{B})$ is $\mathcal{L}$-definable as well.
3. The condition that $(-)^*$ be functorial ensures that $C \cong D$ implies that $\mathcal{L}_C \cong \mathcal{L}_D$ and that $(\mathcal{L}_C, \models_C)$ and $(\mathcal{L}_D, \models_D)$ are equivalent logics. It plays no role in the sequel.

Example 4 (Hennessy-Milner logic). Hennessy-Milner logic is a typical example of a logic for $\mathcal{P}_\omega$-coalgebras $(\mathcal{L}, \models)$ (in the sense of Remark 2.1). Formulas in $\mathcal{L}$ are built from the propositional constant $\bot$ (falsum), boolean operators, and a modal operator $\Box$. Given a formula φ and a process (X, ξ, x), one has $(X, \xi, x) \models \Box\varphi \;\Leftrightarrow\; (X, \xi, x') \models \varphi$ for all $x' \in \xi(x)$. And $(X, \xi) \models \varphi$ iff $(X, \xi, x) \models \varphi$ for all $x \in X$.

Example 5. We extend Hennessy-Milner logic to a logic $(\mathcal{L}_C, \models_C)_{C \in \mathsf{Set}}$ whose formulas involve colours. For each $C \in \mathsf{Set}$, let $\mathcal{L}_C$ be the logic with formulas built from propositional constants $c \in C$, infinitary disjunctions, boolean operators, and a modal operator $\Box$. Define the semantics as in Example 4 with the additional clause $(X, \xi, x) \models c \;\Leftrightarrow\; \pi_2(\xi(x)) = c$ where (X, ξ) is a $(\mathcal{P}_\omega \times C)$-coalgebra and π_2 is the projection $\mathcal{P}_\omega X \times C \to C$. For all $\lambda : C \to D$, let $\lambda^* : \mathcal{L}_D \to \mathcal{L}_C$ be the map replacing each occurrence of $d \in D$ by $\bigvee\{\lambda^{-1}(d)\}$. Note that the disjunction may be infinitary.

The notions of a formula being *preserved under subcoalgebras, quotients, coproducts*, respectively, are defined as usual.[3] Similarly, we say that a class $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma)$ is *closed under domains of quotients* iff for $B \in \mathcal{B}$ and $A \to B$ a surjective coalgebra morphism we have $A \in \mathcal{B}$. Note that the formulas of a logic are preserved under ... iff every definable class of coalgebras is closed under ... .

Formulas of Hennessy-Milner logic are preserved under subcoalgebras, quotients, coproducts, and domains of quotients. The same holds for the logics of the above cited papers [17, 14, 22, 10, 4, 18].

Of interest for us are also the notions of covariety and quasi-covariety which dualise the corresponding notions from algebra. Behavioural covarieties[4] dualise ground varieties.

Definition 5 ((quasi-)covariety, behavioural covariety). *A quasi-covariety is a class of coalgebras closed under coproducts and quotients. A covariety is a quasi-covariety closed under subcoalgebras. A behavioural covariety is a covariety closed under domains of quotients.*

We will use the following fact about quasi-covarieties.

Proposition 1. *Let $\mathsf{Coalg}(\Sigma)$ have cofree coalgebras. Then each quasi-covariety in $\mathsf{Coalg}(\Sigma \times C)$ has a fully abstract final coalgebra.*

Proof. This follows from the fact (see eg. [13], Proposition 2.3) that each quasi-covariety $\mathcal{B}$ is an injective-coreflective subcategory, that is, for all $\mathsf{X} \in \mathsf{Coalg}(\Sigma \times C)$ there is $\mathsf{X}' \in \mathcal{B}$ and an injective morphism $r : \mathsf{X}' \hookrightarrow \mathsf{X}$ such that for all $\mathsf{Y} \in \mathcal{B}$ and all $f : \mathsf{Y} \to \mathsf{X}$ there is a unique $g : \mathsf{Y} \to \mathsf{X}'$ such that $r \circ g = f$. Since, by assumption, $\mathsf{Coalg}(\Sigma \times C)$ has a final coalgebra Z, the fully abstract final coalgebra in $\mathcal{B}$ is given by the coreflection $r : \mathsf{Z}' \hookrightarrow \mathsf{Z}$.

In contrast to algebra where already a weak logic as equational logic allows to define any variety, finitary logics for coalgebras are in general not even expressive

[3] In modal logic terminology one would rather speak of preservation under generated subframes, bounded images, and disjoint unions, respectively.

[4] The name 'behavioural' covariety is due to the fact that a behavioural covariety $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma)$ is closed under behavioural equivalence in the sense that, given $\mathsf{X} \in \mathcal{B}$ and $\mathsf{Y} \in \mathsf{Coalg}(\Sigma)$ such that $!_{\mathsf{X}}(X) = !_{\mathsf{Y}}(Y)$, then $\mathsf{Y} \in \mathcal{B}$ (where X, Y are the carriers of X, Y respectively). Behavioural covarieties are studied eg. in [5, 21, 1].

enough to define all behavioural covarieties. Nevertheless, and this will be used in the proof of our main theorem, any logic for coalgebras is the fragment of an expressive logic, as explained below.

Definition 6 (fragment/extension of a logic). *We say that $\mathcal{L}'$ extends the logic for Σ-coalgebras $\mathcal{L}$ and that $\mathcal{L}$ is a fragment of $\mathcal{L}'$ iff $\mathcal{L}'$ is a logic for Σ-coalgebras with $\mathcal{L}_C \subseteq \mathcal{L}'_C$ and $\varphi \in \mathcal{L}_C \;\Rightarrow\; (\forall \mathsf{X} \in \mathsf{Coalg}(\Sigma \times C) : \mathsf{X} \models_C \varphi \;\Leftrightarrow\; \mathsf{X} \models'_C \varphi)$ for all $C \in \mathsf{Set}$.*

Definition 7 (expressive logic). *A logic for Σ-coalgebras $\mathcal{L}$ is expressive iff, for all $C \in \mathsf{Set}$, every behavioural covariety in $\mathsf{Coalg}(\Sigma \times C)$ is $\mathcal{L}$-definable.*

Remark 3. If $\mathsf{Coalg}(\Sigma)$ has cofree coalgebras then any logic $\mathcal{L}$ for Σ-coalgebras has a smallest expressive extension $\mathcal{L}'$. The idea of the construction is simply to add, for each behavioural covariety $\mathcal{B}$, a formula defining $\mathcal{B}$. That this results indeed in a logic in the sense of Definition 4 follows from [11], Theorem 4.12. The extension $\mathcal{L}'$ is the smallest expressive extension in the sense that $\mathcal{L}'$-definable classes are also definable in any other expressive extension of $\mathcal{L}$.

2 Logics Admitting Final Semantics

The notion of a logic admitting final semantics is adapted from Mahr and Makowsky [15] and Tarlecki [26] who characterised logics for algebras admitting initial semantics. For the notion of a class having a fully abstract final coalgebra see Definition 3.

Definition 8 (logic admitting final semantics). *A logic for Σ-coalgebras $\mathcal{L}$ admits final semantics iff $\mathcal{L}$ is a fragment of an expressive logic $\mathcal{L}'$ such that every $\mathcal{L}'$-definable class has a fully abstract final coalgebra.*

Remark 4. 1. Comparing with [15, 26] the analogous requirement would be to demand that $\mathcal{L}$ itself is expressive. This is too strong in our setting since many logics for coalgebras are not expressive. On the other hand all the logics for coalgebras considered in the papers mentioned in the introduction satisfy our weakened requirement.

2. The requirement of full abstractness means that any definable class $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma)$ not only has a final semantics but that the final semantics of $\mathcal{B}$ is 'inherited' from the final semantics of $\mathsf{Coalg}(\Sigma)$, that is, if two processes of $\mathcal{B}$ are identified in the final semantics of $\mathsf{Coalg}(\Sigma)$, then they are also identified in the final semantics of $\mathcal{B}$.

   The following gives an example—based on a similar one due to Tobias Schröder [25]—of a category $\mathcal{B}$ which has a final coalgebra which is not fully abstract.

   Example. Consider $\mathsf{Coalg}(\mathcal{P}_\omega)$ consisting of the finitely branching transition systems. $\mathsf{Coalg}(\mathcal{P}_\omega)$ has cofree coalgebras. Consider the class $\mathcal{B} \subseteq \mathsf{Coalg}(\mathcal{P}_\omega)$

consisting only of the following transition system X

$$\begin{tikzcd}
& x_0 \arrow[dl] \arrow[dr] & & \\
x_1 & & x_2 \arrow[dr] & \\
& & & x_3
\end{tikzcd}$$

It is not difficult to see that X has only one endomorphism, the identity (recall that morphisms are bisimulations). It follows that X is the final coalgebra of $\mathcal{B} = \{\mathsf{X}\}$. But X is not fully abstract: It distinguishes the states x_1 and x_3 which are identified in the final semantics of $\mathsf{Coalg}(\mathcal{P}_\omega)$ (corresponding to the fact that both states are terminating).

We now formulate our main result.

Theorem 1. *Let $\mathsf{Coalg}(\Sigma)$ have cofree coalgebras and let $\mathcal{L}$ be a logic for Σ-coalgebras. Then $\mathcal{L}$ admits final semantics iff the formulas of $\mathcal{L}$ are preserved under coproducts and quotients.*

As in the results on logics admitting initial semantics [15, 26], the proof is based on a theorem by Mal'cev [16] which we state and prove in the following dualised form (cf. [26] Theorem 4.2).

Theorem 2. *Let $\mathsf{Coalg}(\Sigma)$ have cofree coalgebras. Then for a class of coalgebras $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma \times C)$ the following are equivalent.*

1. *For all mappings $\lambda : D \to C$ and all behavioural covarieties $\mathcal{V} \subseteq \mathsf{Coalg}(\Sigma \times D)$ it holds that $\bar{\lambda}^{-1}(\mathcal{B}) \cap \mathcal{V}$ has a fully abstract final $(\Sigma \times D)$-coalgebra.*
2. *$\mathcal{B}$ is closed under coproducts and quotients.*

Proof (of Theorem 2). '$\Leftarrow$': If $\mathcal{B}$ is a quasi-covariety then $\bar{\lambda}^{-1}(\mathcal{B})$ is a quasi-covariety. The intersection of a quasi-covariety with a behavioural covariety is a quasi-covariety. And quasi-covarieties have fully abstract final coalgebras, see Proposition 1.

'$\Rightarrow$': We use that $\mathcal{B}$ is a quasi-covariety if it is an injective-coreflective subcategory (see eg. [13] Proposition 2.3), that is, if for any $(X, \xi) \in \mathsf{Coalg}(\Sigma \times C)$ there is $(X', \xi') \in \mathcal{B}$ and an injective morphism $r : (X', \xi') \hookrightarrow (X, \xi)$ such that for all $(Y, \nu) \in \mathcal{B}$ and all $f : (Y, \nu) \to (X, \xi)$ there is $g : (Y, \nu) \to (X', \xi')$ such that $r \circ g = f$.

Given (X, ξ) we let $D = C \times X$ and $\lambda : C \times X \to C$ the projection. In the following we denote $(\Sigma \times C)$-coalgebras by their structure and $(\Sigma \times C \times X)$-coalgebras by pairs $\langle \nu, v \rangle$ where ν is a $(\Sigma \times C)$-coalgebra and v is a colouring. Let $\mathcal{V} \subseteq \mathsf{Coalg}(\Sigma \times D)$ be the behavioural covariety $\{\langle \nu, v \rangle \mid \exists f : \langle \nu, v \rangle \to \langle \xi, \mathrm{id}_X \rangle\}$ and note that (ξ, id_X) is a fully abstract final coalgebra in $\mathcal{V}$. It follows from our assumption on $\mathcal{B}$ that $\bar{\lambda}^{-1}(\mathcal{B}) \cap \mathcal{V}$ has a final coalgebra $\langle \xi', v' \rangle$ and (by full abstractness) that there is an injective morphism $r : \langle \xi', v' \rangle \to \langle \xi, \mathrm{id}_X \rangle$. To show that $r : \xi' \to \xi$ is the required 'coreflection'-morphism, let $\nu \in \mathcal{B}$ and consider a $(\Sigma \times C)$-coalgebra morphism $f : \nu \to \xi$. Then $f : \langle \nu, f \rangle \to \langle \xi, \mathrm{id}_X \rangle$ is a $(\Sigma \times D)$-coalgebra morphism. Since $\langle \nu, f \rangle \in \bar{\lambda}^{-1}(\mathcal{B}) \cap \mathcal{V}$, there is $g : \langle \nu, f \rangle \to \langle \xi', v' \rangle$ by the finality of $\langle \xi', v' \rangle$. Hence $f = r \circ g$ as required.

Proof (of Theorem 1). '$\Rightarrow$': Let $\mathcal{B} \subseteq \mathsf{Coalg}(\Sigma \times C)$ be $\mathcal{L}$-definable and $\mathcal{L}'$ be an expressive extension of $\mathcal{L}$. It follows that for all $\lambda : D \to C$ and all behavioural covarieties $\mathcal{V} \subseteq \mathsf{Coalg}(\Sigma \times D)$ the class $\bar{\lambda}^{-1}(\mathcal{B}) \cap \mathcal{V}$ is $\mathcal{L}'$-definable and, therefore, has a fully abstract final coalgebra. Now apply Theorem 2.

'$\Leftarrow$': Let $\mathcal{L}''$ be a logic having precisely the behavioural covarieties as definable classes. That $\mathcal{L}''$ is a logic in the sense of Definition 4 follows from Theorem 4.12 in [11]. Define, for all $C \in \mathsf{Set}$, $\mathcal{L}'_C$ as the disjoint union $\mathcal{L}_C + \mathcal{L}''_C$ and $\models'_C = (\models_C \cup \models''_C)$. Then $\mathcal{L}$ is a fragment of the expressive logic $\mathcal{L}'$. Since $\mathcal{L}$-definable classes are quasi-covarieties and $\mathcal{L}''$-definable classes are behavioural covarieties, $\mathcal{L}'$-definable classes are quasi-covarieties. And quasi-covarieties have fully abstract final coalgebras, see Proposition 1.

Remark 5.
1. In '$\Rightarrow$' of the proof of Theorem 1 we see why we need to require that definable classes of an *expressive* extension of $\mathcal{L}$ have fully abstract final coalgebras. For an example showing that it does not suffice to require that $\mathcal{L}$-definable classes have fully abstract final coalgebras, recall Remark 2.1 and consider a logic $(\mathcal{L}, \models)$ that has as the only definable class $\mathcal{B} \subset \mathsf{Coalg}(\Sigma)$ the class consisting of precisely the cofree coalgebras (for, say, $\Sigma = \mathcal{P}_\omega$). Then every $\mathcal{L}_C$-definable class $\subseteq \mathsf{Coalg}(\Sigma \times C)$ has a fully abstract final coalgebra but $\mathcal{B}$ is not closed under quotients.
2. The corresponding result in Tarlecki [26], Theorem 4.4, is proved more generally not only for algebras but for 'abstract algebraic institutions'. Our result can be generalised along the same lines. In fact, a logic for coalgebras as in Definition 4 is a co-institution (see [11]). It suffices therefore to extract the additional requirements needed to prove the theorems above in order to reach a corresponding notion of 'abstract coalgebraic co-institution'. But in contrast to abstract algebraic institutions which subsume not only (standard) algebras but also other structures as eg. partial algebras and continuous algebras, we are not aware of analogous examples in the coalgebraic world that would justify the generalisation of our result to 'abstract coalgebraic co-institutions'.

3 Examples of Logics Admitting Final Semantics

As mentioned already, most logics for coalgebras studied so far, only allow for definable classes closed under coproducts, quotients, subcoalgebras, and domains of quotients. On the other hand, as our theorem shows, stronger specification languages with formulas not necessarily preserved under subcoalgebras and domains of quotients may be of interest. In this section we show some ways of how to extend known logics for coalgebras to such stronger ones. Since the ideas from modal logic used in this section are completely standard, we avoid providing full details. They can be found, eg., in [2].

Let us first go back to the example of Hennessy-Milner logic. We can extend the expressive power of Hennessy-Milner logic by *adding propositional variables.*

Example 6 (adding propositional variables). Let us extend the logic of Example 4 by adding propositional variables from a set $\mathbf{P}$. The satisfaction relation is now defined by referring to coloured processes (X, v, x) where $v : X \to \mathcal{P}\mathbf{P}$. For propositional variables $p \in \mathbf{P}$ we have $(\mathsf{X}, v, x) \models p \quad \Leftrightarrow \quad p \in v(x)$. Boolean and modal operators are defined as usual. And $\mathsf{X} \models \varphi$ iff $(\mathsf{X}, v, x) \models \varphi$ for all $v : X \to \mathcal{P}\mathbf{P}$ and all $x \in X$.

Note that the definition of $\mathsf{X} \models \varphi$ involves a quantification over all valuations of propositional variables $v : X \to \mathcal{P}\mathbf{P}$. Since the extension of a propositional variable can be any subset of the carrier of X, adding propositional variables can be described as *allowing a prefix of universally quantified monadic second-order variables* in the formulas (cf. [2], Definition 2.45 and Proposition 3.12).

Typical examples of how adding propositional variables increases expressiveness are the following. Referring to Example 6, the formulas $\Box p \to p$ and $\Box p \to \Box\Box p$ with $p \in \mathbf{P}$ define, respectively, the class of reflexive Kripke frames and the class of transitive Kripke frames. Both classes are not closed under domains of quotients, showing that propositional variables add indeed expressiveness. On the other hand, formulas with propositional variables are still preserved under subcoalgebras, that is, definable classes are covarieties. Conversely, every covariety is definable by an (infinitary) modal logic with propositional variables (see [12]).

We show now how to build logics whose formulas are not necessarily preserved under subcoalgebras. A logic for coalgebras, possibly with propositional variables, can be strengthened by *adding rules.* Given two formulas φ, ψ we call φ/ψ a rule and extend the satisfaction relation via

$$\mathsf{X} \models \varphi/\psi \quad \text{iff} \quad (\mathsf{X}, v) \models \varphi \ \Rightarrow \ (\mathsf{X}, v) \models \psi \quad \text{for all valuations } v : X \to \mathcal{P}\mathbf{P}.$$

This definition dualises the definition of implications for algebras and was studied in [13] where it was shown that—allowing infinitary conjunctions—any quasi-covariety is definable by a logic for coalgebras with rules.

Since rules can be rather unintuitive in writing specifications, *adding a global diamond* instead (suggested to the author by Alexandru Baltag) may be preferable. A global diamond $\mathbf{E}$ (cf. [2], Section 7.1) is a unary modal operator defined via

$$(\mathsf{X}, v, x) \models \mathbf{E}\varphi \quad \text{iff} \quad (\mathsf{X}, v, y) \models \varphi \text{ for some } y \in X.$$

$\mathbf{E}$ is called global because the range of quantification is not confined by the transition structure. Of course, adding a global diamond, we have to restrict occurrences of $\mathbf{E}$ to appear only positively in the formulas (otherwise we would also add the defining power of a global box[5]).

Concerning expressiveness, adding the global diamond is equivalent to adding rules. To sketch the argument: On the one hand, every rule φ/ψ is equivalent to the formula $\neg\psi \to \mathbf{E}\neg\varphi$; on the other hand, formulas containing a global diamond are still preserved under coproducts and quotients and therefore can not be more expressive than rules.

Conclusion

We have shown that a logic for coalgebras admits final semantics iff its formulas are preserved under coproducts and quotients.

On the one hand, this result allows to design specifications languages admitting final semantics, since it is usually not difficult to check whether formulas are preserved under coproducts and quotients. This can be of interest for specification languages for coalgebras like CCSL [23]. CCSL allows the coalgebraic specification of classes of object-oriented programs. A question in this context is to determine the largest fragment of CCSL that ensures that specified classes of objects have a final semantics (final semantics for objects was proposed by Reichel [19] and Jacobs [9]). The value of our result in such a concrete setting needs further exploration.

On the other hand we have pointed out possibilities to extend weaker logics in a way that they still admit final semantics. Possible strengthenings may allow formulas with (1) prefixes of universally quantified monadic second-variables (propositional variables) and (2) positive occurrences of a first-order existential quantifier (global diamond).

Acknowledgements

I am indebted to Alexandru Baltag for suggesting to consider logics with a global diamond, to Tobias Schröder for his example showing that a Kripke model admitting no non-trivial endomorphisms still can have distinct bisimilar points, and to Falk Bartels for suggesting $\mathcal{P}_{ne}$ when looking for a category of coalgebras with a final but not all cofree coalgebras. I want to thank Till Mossakowski and Hendrik Tews for asking me the question answered in this paper. Finally, I want to thank the anonymous referees for their helpful and interesting comments. The diagram was drawn with Paul Taylor's macro package.

[5] A global box $\mathbf{A}$ (defined as $\neg\mathbf{E}\neg$) would allow to define classes that do not have final coalgebras. Eg., for a propositional variable p, $p \to \mathbf{A}p$ defines the class of $\mathcal{P}_\omega$-coalgebras with at most one element (and this class contains exactly three coalgebras none of which is final).

References

1. S. Awodey and J. Hughes. The coalgebraic dual of Birkhoff's variety theorem. Technical Report CMU-PHIL-109, Carnegie Mellon University, Pittsburgh, PA, 15213, November 2000.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001. See also `http://www.mlbook.org`.
3. A. Corradini, M. Lenisa, and U. Montanari, editors. *Coalgebraic Methods in Computer Science (CMCS'01)*, volume 44-1 of *Electronic Notes in Theoretical Computer Science*, 2001.
4. Robert Goldblatt. A calculus of terms for coalgebras of polynomial functors. In A. Corradini, M. Lenisa, and U. Montanari, editors, *Coalgebraic Methods in Computer Science (CMCS'01)*, volume 44.1 of *ENTCS*. Elsevier, 2001.
5. H. P. Gumm and T. Schröder. Covarieties and complete covarieties. *Theoretical Computer Science*, 260:71–86, 2001.
6. B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors. *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11. Electronic Notes in Theoretical Computer Science, 1998.
7. B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997.
8. B. Jacobs and J. Rutten, editors. *Coalgebraic Methods in Computer Science*, volume 19. Electronic Notes in Theoretical Computer Science, 1999.
9. Bart Jacobs. Objects and classes, co-algebraically. In B. Freitag, C. B. Jones, C. Lengauer, and H.-J. Schek, editors, *Object-Orientation with Parallelism and Persistence*, pages 83–103. Kluwer Acad. Publ., 1996.
10. Bart Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications*, 35(1):31–59, 2001.
11. A. Kurz and D. Pattinson. Coalgebras and modal logics for parameterised endofunctors. Technical Report SEN-R0040, CWI, 2000. `http:www.cwi.nl/~kurz`.
12. Alexander Kurz. A co-variety-theorem for modal logic. In *Advances in Modal Logic 2*. Center for the Study of Language and Information, Stanford University, 2001.
13. Alexander Kurz. Modal rules are co-implications. In A. Corradini, M. Lenisa, and U. Montanari, editors, *Coalgebraic Methods in Computer Science (CMCS'01)*, volume 44.1 of *ENTCS*. Elsevier, 2001.
14. Alexander Kurz. Specifying coalgebras with modal logic. *Theoretical Computer Science*, 260:119–138, 2001.
15. B. Mahr and J. A. Makowsky. Characterizing specification language which admit initial semantics. *Theoretical Computer Science*, 31(1+2):59–60, 1984.
16. A. I. Mal'cev. Quasiprimitive classes of abstract algebras. In *The Metamathematics of Algebraic Systems, Collected Papers: 1936–1967*. North-Holland, 1971. Originally published in Dokl. Akad. Nauk SSSR 108, 187–189, 1956.
17. Lawrence Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.
18. Dirk Pattinson. Semantical principles in the modal logic of coalgebras. In *Proceedings 18th International Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, volume 2010 of *LNCS*, Berlin, 2001. Springer. Also available as technical report at `http://www.informatik.uni-muenchen.de/~pattinso/`.
19. Horst Reichel. An approach to object semantics based on terminal co-algebras. *Mathematical Structures in Computer Science*, 5(2):129–152, June 1995.

20. Horst Reichel, editor. *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.
21. Grigore Roşu. Equational axiomatizability for coalgebra. *Theoretical Computer Science*, 260:229–247, 2001.
22. Martin Rößiger. Coalgebras and modal logic. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 299–320, 2000.
23. J. Rothe, H. Tews, and B. Jacobs. The coalgebraic class specification language CCSL. *Journal of Universal Computer Science*, 7(2):175–193, 2001.
24. J.J.M.M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000. First appeared as technical report CS R 9652, CWI, Amsterdam, 1996.
25. Tobias Schröder, January 2001. Personal Communication.
26. Andrzej Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, 37:269–304, 1986. Preliminary version, University of Edinburgh, Computer Science Department, Report CSR-165-84, 1984.