Coalgebras, Stone Duality, Modal Logic

Alexander Kurz

University of Leicester

9th August 2004

The course aims at introducing some basic facts about coalgebras (Sections 1.1, 2.1, 3.1) and then showing how logics for coalgebras can by obtained by putting a duality of algebras and coalgebras on top of a Stone duality (Section 4). I make no attempt to introduce Stone duality here but rather use it to get interesting dualities between algebras and coalgebras. The first part also contains some 'additional material', namely, a comparison of different notions of coalgebras (for a functor, for a signature, for a comonad; Section 1.2), a comparison of different notions of bisimulation (Section 2.2), and the final coalgebra sequence (Section 3.2).

As the summer school is affiliated with the conference on Category Theory and Computer Science, choice of material and presentation assume some basic category theory. I also didn't want to repeat too much from what is available in the excellent introductions [27, 45, 21]. The present notes can also be regarded as a sequel to [33].

The contents of the notes is limited by the available 5 times 45 minutes. The biggest omission is probably more on coinduction which would easily fill such a course by itself. A similar remark applies to Stone duality.

Contents

1 Coalgebras

1.1 Basic Definitions and Examples

Definition 1.1 (Coalgebras). Given a category $\mathcal{X}$, called the base category, and a functor $T : \mathcal{X} \to \mathcal{X}$, a T-**coalgebra** (X, ξ) is given by an arrow $\xi : X \to TX$ in $\mathcal{X}$. A morphism between two coalgebras $f : (X, \xi) \to (X', \xi')$ is an arrow f in $\mathcal{X}$ such that $\xi' \circ f = Tf \circ \xi$:

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

The category of coalgebras and morphisms is denoted by $\mathsf{Coalg}(T)$. We write $\mathsf{Alg}(T)$ for $\mathsf{Coalg}(T^{\mathrm{op}})^{\mathrm{op}}$ and call its objects algebras for the functor $T^{\mathrm{op}} : \mathcal{X}^{\mathrm{op}} \to \mathcal{X}^{\mathrm{op}}$.

Exercise 1.2. Spell out the definition of $\mathsf{Alg}(T)$. Note that, up to dual isomorphism, there is no difference between coalgebras over $\mathcal{X}$ and algebras over $\mathcal{X}^{\mathrm{op}}$.

Notation 1.3. 0,1,2 denote sets of the respective cardinality. For $K \in \mathcal{X}$ the constant functor $X \mapsto K$, $f \mapsto \mathrm{id}_K$ is denoted by K and the identity functor by Id. We freely use structure available in $\mathcal{X}$. For example, Set has products, coproducts and is cartesian closed, ie, given functors F, G and a constant functor K, we also have functors $F \times G$, $F + G$, K^F, F^K defined pointwise.[1] A special case is the contravariant powerset functor 2^{Id}. Its action on functions corresponds to inverse image. The covariant powerset is denoted by $\mathcal{P}$.

Example 1.4. Let $\mathcal{X} = \mathsf{Set}$ be the category of sets and functions.

1. **(Streams)** Let $TX = O \times X$ and $Tf = \mathrm{id}_D \times f$. A coalgebra

$$X \xrightarrow{\xi} O \times X$$

   can be understood as a process which, started in some state $x \in X$, produces a list of outputs $(o_0, o_1, \ldots)$ determined by $\xi(x_n) = (o_n, x_{n+1})$. The infinite list (or stream) $(o_0, o_1, \ldots)$ is called the behaviour of the state x, a notion formalised in Section 2.

2. **(Deterministic Automata)** Let $TX = (2 \times X)^I$. A coalgebra

$$X \xrightarrow{\xi} (2 \times X)^I$$

   is a deterministic automaton which determines for each state $x \in X$ and each input $i \in I$ the pair $(b, x') = \xi(x)(i)$ where $b \in \{0, 1\}$ indicates whether x is an accepting state and x' is the successor state of x.

[1] Eg $(F \times G)(X) = FX \times GX$, $(F \times G)(f) = Ff \times Gf$, etc. In particular, (Ff^K) is post-composition with Ff and K^{Ff} is pre-composition, ie, $(Ff^K)(g) = Ff \circ g$ and $K^{Ff}(h) = h \circ Ff$ where $f : X \to Y$, $g : K \to FX$, $h : FX \to K$.

3. **(Partial Functions, 'Exceptions', 'Classes')** If one wants to allow the state transition map for deterministic automata to be partial, then this can be accounted for by letting $TX = 2 \times (1 + X)^I$. The idea of a 'method' as mapping an 'object' x and input i to either an 'exception' $e \in E$ or an output and a successor state can be formalised as a coalgebra
$$X \longrightarrow (E + O \times X)^I.$$
A 'class' consisting of several methods $m_1, \ldots m_n$ would then correspond to a coalgebra $\langle m_1, \ldots m_n \rangle : X \to (E_1 + O_1 \times X)^{I_1} \times \ldots (E_n + O_n \times X)^{I_n}$.

4. **(Polynomial Endofunctors)** The above examples of T (restricting inputs I to finite sets) are particular instances of polynomial endofunctors which are built according to
$$T ::= Id \mid K \mid T \times T \mid T + T.$$
T-algebras are the usual algebras given by a signature (see below). An algebra structure $TA \to A$ describes how to constructed elements of A using terms built from a finite number of operations. A coalgebra structure $X \to TX$ describes how to deconstruct (or observe) states of X using terms built from a possibly infinite number of operations.

5. **(Relations, Kripke Frames)** A Kripke frame is a set with a relation (X, R), $R \subseteq X \times X$. The difference to relations is in the notion of morphism. A morphism $f : (X, R) \to (X', R')$ between Kripke frames is not only relation-preserving $(xRy \;\Rightarrow\; f(x)R'f(y))$ but also satisfies the backward condition[2] $f(x)R'y' \;\Rightarrow\; \exists y \,.\, xRy \;\&\; f(y) = y'$. This notion of morphism can be elegantly captured by considering relations (X, R) as coalgebras
$$X \xrightarrow{\xi} \mathcal{P}X$$
where $\mathcal{P}X$ is the covariant powerset functor[3] and $\xi(x) = \{y \mid xRy\}$ is the set of successors of x. In the following , we will identify Kripke frames with $\mathcal{P}$-coalgebras.

6. **(Labelled Transition Systems)** Transition systems with labels from a set L are coalgebras for the functors $\mathcal{P}(L \times -)$ or, equivalently, $\mathcal{P}(-)^L$. The morphisms are again precisely those functions whose graph is a bisimulation (exercise).

7. **(Probabilistic Transition Systems)** Probabilistic transition systems in which for each state and label there is either no successor or a probability distribution of successors are coalgebras
$$X \to (1 + D_\omega(X))^L$$

[2]Beware, this is not equivalent to $f(x)R'f(y) \;\Rightarrow\; xRy$.
[3]$\mathcal{P}X$ is the set of subsets of X and $\mathcal{P}f$ is the direct image map.

where

$$D_\omega(X) = \{\mu : X \to \mathbb{R}_0^+ \mid \{x \mid \mu(x) \neq 0\} \text{ is finite and } \sum_{x \in X} \mu(x) = 1\}$$

$$D_\omega(f)(\mu) = \lambda y. \sum_{x \in f^{-1}(y)} \mu(x)$$

8. **(Hypersystems, Neighbourhood Frames, Topological Spaces)** Coalgebras

$$X \to 2^{2^X}$$

are called hypersystems in [45]. There is a one-to-one correspondence between coalgebras $\xi : X \to 2^{2^X}$ and maps $\check{\xi} : 2^X \to 2^X$, ie, coalgebras for a signature consisting of one (2,2)-ary operation symbol (see next subsection); this—and the examples below—suggest that $(2^{2^{(-)}})$-coalgebras are one of the fundamental examples of coalgebras. The functor 2^{2^-} is also of interest as an example of a functor that does not preserve weak pullbacks (see Exercise 5.4).

Of independent interest are certain covarieties[4] of hypersystems. For example, the class of coalgebras $\xi : X \to 2^{2^X}$ such that $\xi(x)$ is closed under supersets is known in modal logic as *neighbourhood frames* (see eg the recent [23]). If $\xi(x)$ is, moreover, required to be closed under finite intersections, then one obtains *normal neighbourhood frames.* In terms of $\check{\xi}$, the two conditions above are that $\check{\xi}$ is monotone and preserves finite intersections. If we add the requirement $\check{\xi} \subseteq \check{\xi} \circ \check{\xi}$, then $\check{\xi}$ is an interior operator and we obtain the covariety of *topological spaces and open and continuous maps.* If we impose on $\check{\xi}$ to (only) preserve infinite intersections, then one obtains Kripke frames.

9. **(Nesting Initial Algebras and Final Coalgebras)** If $H : \mathsf{Set} \times \mathsf{Set} \to \mathsf{Set}$ is a functor such that $H(-, A)$ has an initial algebra for each A, denoted by $\mu Y.H(Y, A)$, then we denote by $\mu Y.H(Y, X)$ the induced endofunctor on set. Dually, we denote by $\nu Y.H(Y, X)$ the endofunctor that maps X to the final coalgebra of the functor $H(-, X) : \mathsf{Set} \to \mathsf{Set}$. See eg Hensel and Jacobs [24] for more.

Example 1.5. Some examples of coalgebras over other base categories than Set.

1. Let $\mathcal{X}$ be the category of sets with inclusions as arrows. A functor is a monotone operator. A coalgebra $X \subseteq TX$ is a post-fixed point.

2. The functors of examples (1-4) use products, coproducts and exponents[5]. Analogous functors exist therefore on any other category than Set providing that structure.

[4] A covariety is a full subcategory that is closed under subcoalgebras, homomorphic images and coproducts. They are precisely the (co)equationally definable subcategories, see [35] for more information.

[5] The exponent X^I can be generalised in two ways: If I is an object of the category one needs cartesian closure; if I is a set one just needs I-fold products.

Analogies of the powerset and the probabilistic functor are available on a number of topological spaces and domains.

3. Often algebraic and coalgebraic operations interact. For example, in process algebra, one has the coalgebraic structure given by labelled transition systems and algebraic structure as eg prefixing, choice, parallel composition. Algebraic and coalgebraic structure is typically expected to interact in way that guarantees that coalgebraic bisimulation (see next section) is a congruence wrt the algebraic operations.[6] Following Turi and Plotkin [50], these situations can often be described by *bialgebras* $FX \xrightarrow{\varphi} X \xrightarrow{\gamma} GX$ which satisfy $\gamma \circ \varphi = G\varphi \circ \lambda_X \circ F\gamma$ where $\lambda : FG \to GF$ is a natural transformation (a 'distributive law') describing the interaction of the algebraic operations given by F and the coalgebraic ones given by G. One can then lift G to an endofunctor G' on $\mathsf{Alg}(F)$ and show that the category of bialgebras is isomorphic to the category of G'-coalgebras over $\mathsf{Alg}(F)$.

4. Some of the interesting base categories in semantics as eg preorders, generalised ultrametric spaces, generalised metric spaces can be considered as enriched categories. See eg Worrell [53] for examples of coalgebras over enriched categories.

5. If one wants to consider datatypes together with appropriate logics, it makes sense to consider coalgebras for functors that act on fibred categories, see Hermida and Jacobs [25] (and also Section 2.2).

1.2 Other Notions of (Co)Algebras

This section is intended to give some background on other ways to define algebras and coalgebras. In particular, we will compare the notions of (co)algebras for a functor, for a signature and for a (co)monad. We will also indicate why algebras over set are usually given by a signature and coalgebras over set by a functor. This section may be somewhat dense at places but will not be needed later on.

1.2.1 Algebras for a Signature over Set

Usually, algebras are given wrt a signature Σ which consists of operation symbols $\sigma \in \Sigma$ with associated arities n_σ. A Σ-algebra consists of a set A and an interpretation $\sigma_A : A^{n_\sigma} \to A$ of each operation symbol σ. A morphism between Σ-algebras is a function $f : A \to A'$ such that $f(\sigma_A(\langle a_i \rangle_{i<n_\sigma})) = \sigma_{A'}(\langle f(a_i) \rangle_{i<n_\sigma})$. We write $\mathsf{SAlg}(\Sigma)$ for the category of algebras for the signature Σ. We may also want to consider equations $(t = s) \in E$ over variables V. An algebra A satisfies the equation $t = s$ if for all 'valuations of variables' $v : V \to A$ the extension $v^\#$ from variables to terms satisfies $v^\#(t) = v^\#(s)$. The category of algebras given by a signature and equations is denoted by $\mathsf{SAlg}(\Sigma, V, E)$, or shorter, $\mathsf{SAlg}(\Sigma, E)$.

[6] Intuitively, this says that adding the algebraic operations does not allow to distinguish more states than with the coalgebraic structure alone.

Remark (on matters of size). In the definitions above, we want to allow arities to be cardinals and the collection of operation symbols of a given arity to be a (small) set. Σ, V and E, however, may be proper classes. One reason for admitting classes is to treat structures such as complete semilattices, complete atomic Boolean algebras, etc using algebraic methods. Another reason is that the signature and equations associated to a functor $T : \mathsf{Set} \to \mathsf{Set}$ below may require proper classes.

That algebras for a signature are algebras for a functor is the content of the next proposition the proof of which is straightforward.

Proposition 1.6. *If Σ is a (small) set then $\mathsf{SAlg}(\Sigma) \cong \mathsf{Alg}(T)$ for $TA = \coprod_{\sigma\in\Sigma} A^{n_\sigma}$.*

Conversely, we can associate to any functor $T : \mathsf{Set} \to \mathsf{Set}$ a signature and equations. But before doing this we present a generalisation of algebras for a signature to arbitrary base categories.

1.2.2 Algebras for a Signature over a Base Category

Following earlier work by Lawvere, Linton [36] generalised algebras for operations and equations from algebras over Set to algebras over an arbitrary base category $\mathcal{X}$. The idea is to replace arities by objects in $\mathcal{X}$ and operations $A^n \to A$ by operations $\mathcal{X}(X, A) \to \mathcal{X}(Y, A)$ or, more suggestively, $A^X \to A^Y$ (where $\mathcal{X}$ replaces Set, X replaces n and Y replaces 1).

In detail, following Rosický [42], arities are pairs (X, Y) of objects in $\mathcal{X}$ and a signature Σ consists of a set of operation symbols for each arity. A Σ-algebra is given by an object $A \in \mathcal{X}$ and an arrow $\sigma_A : A^X \to A^Y$ for each $\sigma_A \in \Sigma$. Terms are defined inductively by (i) x_f is a (X, Y)-ary term for all arrows $f : Y \to X$ in $\mathcal{X}$, (ii) σ is a term for all $\sigma \in \Sigma$, (iii) if t is an (Z, Y)-ary term and s a (X, Z)-ary term then $s \cdot t$ is a (X, Y)-ary term. A (X, Y)-ary term t induces on an algebra A a map $t_A : A^X \to A^Y$ (where x_f is interpreted by $\mathcal{X}(f, A) : A^X \to A^Y$). An equation is a pair of (X, Y)-ary terms (t, s) and an algebra satisfies the equation if $t_A = s_A$. Given such a signature Σ and equations E, we write $\mathsf{SAlg}(\Sigma, E)$ as before for the corresponding category of algebras.

We can show now in full generality a converse of the above proposition saying that (co)algebras for a functor are (co)algebras for a signature and equations.

Proposition 1.7 (Reiterman [40]). *Let T be an endofunctor on $\mathcal{X}$. There are signature and equations such that $\mathsf{SAlg}(\Sigma, E) \cong \mathsf{Alg}(T)$.*

Proof. For each object $X \in \mathcal{X}$ there is an (X, TX)-ary operation symbol σ^X. For each $f : Y \to X$ there is an equation

$$x_{Tf} \cdot \sigma^X = \sigma^Y \cdot x_f. \tag{1}$$

This defines the category $\mathsf{SAlg}(\Sigma, E)$. A T-algebra A with structure $\alpha : TA \to A$ determines a (Σ, E)-algebra which interprets operations as $\sigma^X_A : A^X \to A^{TX}, f \mapsto \alpha \circ Tf$. Conversely, a (Σ, E)-algebra A determines the T-algebra $\sigma^A_A(\mathrm{id}_A) : TA \to A$. To see that this

T-algebra determines the original Σ-algebra A, we have to check that $\sigma_A^A(\mathrm{id}_A) \circ Tf = \sigma_A^X(f)$, $f : X \to A$. This follows from A satisfying the equation $x_{Tf} \cdot \sigma^A = \sigma^X \cdot x_f$ which is interpreted on A as

$$\begin{tikzcd} A^A \arrow[r, "\sigma_A^A"] \arrow[d, "A^f"'] & A^{TA} \arrow[d, "A^{Tf}"] \\ A^X \arrow[r, "\sigma_A^X"] & A^{TX} \end{tikzcd}$$

□

Remark 1.8.

1. In general, the proof only works because we have so many operation symbols, one for each object in $\mathcal{X}$ (to obtain a T-algebra from a (Σ, E)-algebra A we needed an (A, TA)-ary operation). But if $\mathcal{X}$ is lfp (as eg Set) and T is ω-accessible (as eg polynomial functors or $\mathcal{P}_\omega$), then on can restrict arities (X, TX) to objects X that are finitely presentable. The T-algebra corresponding to $A \in \mathsf{SAlg}(\Sigma, E)$ is then obtained as follows. A is a filtered colimit $c_i : X_i \to A$ of finitely presentable objects X_i. Since T preserves filtered colimits, $TX_i \to TA$ is a filtered colimit as well. Since $\sigma_A^X(c_i) : TX_i \to A$ is a cocone, there is a unique mediating arrow $TA \to A$.

2. It is instructive to specialise the construction of the proof for the case $\mathcal{X} = \mathsf{Set}$. An operation $A^X \to A^{TX}$ is then the same as TX-many operations $A^X \to A$, that is, we obtain a signature whose set of X-ary operation symbols is TX. Denote by $q_A : \coprod_{X \in \mathcal{X}} TX \times A^X \to TA$ the natural transformation sending $(\sigma, v) \in TX \times A^X$ to $Tv(\sigma)$. The relationship between T-algebras and σ-algebras can now be described as follows. Given a T-algebra α, the corresponding (Σ, E)-algebra is described by $s = \alpha \circ q_A$ as in

$$\begin{tikzcd} \coprod_{X \in \mathcal{X}} TX \times A^X \arrow[rr, "q_A"] \arrow[dr, "s"'] & & TA \arrow[dl, "\alpha"] \\ & A & \end{tikzcd}$$

A pair $(\sigma, v) \in TX \times A^X$ can be viewed as an X-ary term that takes his arguments from A. The equations (1) become, for a chosen A, $(\sigma, v \circ f) = (Tf(\sigma), v)$, $v : X \to A$; these equations generate an equivalence relation on $\coprod_{X \in \mathcal{X}} TX \times A^X$ which is the kernel of q_A. This shows that, conversely, given a Σ-algebra A with structure s, A corresponds to a T-algebra iff s factors through q_A.

3. As a consequence of the above, if T is a finitary functor on Set then $\mathsf{Alg}(T) \cong \mathsf{Alg}(\Sigma, E)$ where the n-ary operation symbols are the elements of Tn, $n < \omega$, the variables are given by a countable set V and the equations are those $(\sigma, f) = (\sigma', f')$ for which

$$\begin{tikzcd} \coprod_{n < \omega} Tn \times V^n \arrow[rr, "q_V"] & & TV \end{tikzcd}$$

$q_V(\sigma, f) = q_V(\sigma', f')$, see [8][Chapter III.3]. There is also a converse, stating that if the equations only involve terms 'of depth 1' then $\mathsf{Alg}(\Sigma, E)$ is the categories of algebras for a functor (namely the polynomial functor of Proposition 1.6 quotiented by the equations), see [8].

1.2.3 Coalgebras for a Signature

Since coalgebras over $\mathcal{X}$ are (dually isomorphic to) algebras over $\mathcal{X}^{\mathrm{op}}$, the above concept of algebras over a base category gives rise to a notion of coalgebras for a signature over a base category. Spelling this out for $\mathcal{X} = \mathsf{Set}$, operations have arities which are pairs (X, Y) of sets. Operations are interpreted on a carrier set A by functions $\mathsf{Set}(A, X) \to \mathsf{Set}(A, Y)$ or, as a shorthand, $X^A \to Y^A$. For example, hypersystems (as well as neighbourhood frames, topological spaces, Kripke frames, see Example 1.4) are coalgebras for a single $(2, 2)$-ary operation. More generally, operations $X^A \to Y^A$ are modal operators that transform X-valued predicates into Y-valued predicates. These 'predicate transformers' are in fact modal operators in the sense that they respect the notion of behavioural equivalence presented in the next section. Coalgebras for operations and equations were investigated by Davis in the 70s [18] and more recently in [35].

Remark 1.9. One might be inclined to say that algebras for a functor on Set are of limited interest since they are not more powerful than algebras for operations and equations. But then, why are coalgebras usually described by functors and not by operations and equations?

Part of the answer might involve the following observation. The functors that describe algebras over Set are often finitary. In that case, as follows from Remark 1.8, one can find a nice signature, in the sense that the arities can be chosen to be finite. This depends on the fact that every set is the union (or filtered colimit) of finite sets. Unfortunately, the dual property cannot be proved in (but is consistent with) ZFC, see [7][A.5].

1.2.4 (Co)Algebras for a (Co)Monad

If the signature of a category $\mathsf{SAlg}(\Sigma, E)$ is a proper class, then $U : \mathsf{SAlg}(\Sigma, E) \to \mathsf{Set}$ may fail to have a left adjoint (ie free algebras need not exist in $\mathsf{SAlg}(\Sigma, E)$). Often one is interested in this additional property. Categories of algebras given by operations and equations and having free algebras can be characterised as categories of algebras for a monad.

Recall from Mac Lane [37] that a monad (M, η, μ) consists of an endofunctor on a category $\mathcal{X}$ and two natural transformations $\eta : Id \to M$, $\mu : MM \to M$ satisfying $\mu \circ M\eta = \mu \circ \eta_M = Id_M$ and $\mu \circ \mu_M = \mu \circ M\mu$. The category $\mathsf{MAlg}(M)$ of algebras for the monad M is the full subcategory of algebras $\alpha : MA \to A$ for the functor M satisfying $\alpha \circ \eta_A = \mathrm{id}_A$ and $\alpha \circ \mu_A = \alpha \circ M\alpha$.[7] A functor $U : \mathcal{A} \to \mathsf{Set}$ is called *monadic* if it

[7] It follows from the equations that $\mu_A : MMA \to MA$ is the free algebra over A with 'insertion of generators' given by $\eta_A : A \to MA$.

has a left adjoint F with counit ε and the functor $\mathcal{A} \to \mathsf{MAlg}(UF), A \mapsto (UA, U\varepsilon_A)$ is an isomorphism.

The following theorem is formulated over Set but also holds for arbitrary base categories [36] (and hence for coalgebras over Set) using Linton's notion of a signature (see above).

Theorem 1.10 (Linton [36]). *Monadic categories over* Set *coincide with categories* $\mathsf{SAlg}(\Sigma, E)$ *that have free algebras.*

Sketch. Given a left adjoint F of $U : \mathsf{SAlg}(\Sigma, E) \to \mathsf{Set}$ one appeals to Beck's theorem [37] to show that $\mathsf{SAlg}(\Sigma, E)$ is isomorphic to $\mathsf{MAlg}(UF)$. Conversely, given a monad M one finds a signature and equations as in the proof of Proposition 1.7, representing the algebras for the functor M by operations and equations. One then adds appropriate equations that enforce the laws $\alpha \circ \eta_A = \mathrm{id}_A$ and $\alpha \circ \mu_A = \alpha \circ M\alpha$. $\square$

1.3 Further Remarks

Functors of Mixed Variance It seems essential for the theory of coalgebras for a functor that the functor be an endofunctor. This excludes, in particular, functors of mixed variance. Consider a functor $F : \mathsf{Set} \times \mathsf{Set}^{\mathrm{op}} \to \mathsf{Set}$. The natural notion of morphism from a 'coalgebra' $\xi : X \to F(X, X)$ to $\xi' : X' \to F(X', X')$ is a map $f : X \to X'$ such that $F(f, \mathrm{id}) \circ \xi = F(\mathrm{id}, f) \circ \xi' \circ f$. These coalgebras have been of interest in the study of so-called 'binary methods' (cf. Example 1.4.3) but there mathematical properties are quite different. See Tews [46, 47, 48].

Covariety Theorems Birkhoff's variety theorem states that a class of algebras for a signature is closed under images, quotients and products iff it is equationally definable. Dualising this theorem, the main question is what should replace equational logic. Different answers are offered by [20, 41, 32, 19, 9, 6, 35].

2 Behavioural Equivalence

We consider different notions of bisimulation for coalgebras and compare them in the case of coalgebras over sets.

2.1 Basic Definitions and Examples

The notion of behavioural equivalence is most useful as an equivalence between states of systems rather than only systems. This is one of the reasons to assume a base category $\mathcal{X} = \mathsf{Set}$. Another is that, otherwise, the comparison with alternative formulations in the next subsection became rather complicated.

Definition 2.1 (Behavioural Equivalence). Given two coalgebras (X, ξ), (X', ξ') and two states $x \in X, x' \in X'$ we say that x, x' are behaviourally equivalent if there is a coalgebra (Q, κ) and there are coalgebra morphisms f, f'

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\xi"] & Q \arrow[d, "\kappa"] & X' \arrow[l, "f'"'] \arrow[d, "\xi'"] \\ TX & TQ & TX' \end{tikzcd}$$

such that $f(x) = f'(x')$.

Behavioural equivalence is transitive because $\mathsf{Coalg}(T)$ has pushouts.

Remark 2.2. The idea of the above definition is that two states are behavioural equivalent iff they can be related by coalgebra morphisms. This is emphasised by the following two alternative formulations.

1. Behavioural equivalence is the smallest equivalence relation containing the pairs

$$((X, \xi, x),\ (X', \xi', f(x)) \quad \text{for all } f : (X, \xi) \to (X', \xi') \text{ and all } x \in X.$$

2. The equivalence classes of behavioural equivalence are the components of the category of elements of the forgetful functor.

3. This shows that the notion of behavioural equivalence can be applied to any set-valued functor $\mathcal{A} \to \mathsf{Set}$. But it is typically category of coalgebras where this notion is interesting. It is trivial, for example, in categories of presheaves or, more generally, many-sorted algebras (because of the existence of a trivial terminal object).

The quotient wrt behavioural equivalence is a coalgebra itself.

Proposition 2.3. *Consider $U : \mathsf{Coalg}(T) \to \mathsf{Set}$. For any coalgebra (X, ξ) the quotient $X \to Q$ of X wrt behavioural equivalence is a coalgebra. This quotient is 'maximal' in the sense that every surjective coalgebra-morphism $Q \to Y$ is an isomorphism.*

Proof (Sketch). We use that Set is cocomplete and that U creates colimits (see Exercise 5.1. First note, for $x, y \in X$, that $x \simeq y$ iff there is a coalgebra $B_{x,y}$ and a coalgebra-morphisms $f : X \to B_{x,y}$, such that $f(x) = f(y)$. The quotient of X wrt behavioural equivalence is now the colimit of the $X \to B_{x,y}$. If $e : Q \to Y$ is a surjective (=epi) coalgebra-morphism then Y is a quotient of X and there must be a coalgebra morphism s such that $s \circ e = id_Q$. Hence e is injective and therefore an isomorphism. □

Note that two states x, y in two different coalgebras are behavioural equivalent iff they are equivalent considered as states of the coproduct of the coalgebras (which is disjoint union). It therefore suffices to consider behavioural equivalence as a relation on a given coalgebra.

Example 2.4. In the following we select from Example 1.4 and give examples of $x \simeq y$ for two states in the same coalgebra $\xi : X \to TX$ for different functors T.

1. **(Streams)** $x \simeq y$ iff the stream produced by x is the same as the one produced by y.

2. **(Deterministic Automata)** $x \simeq y$ iff the language accepted in x is the same as the language accepted in y.

3. **(Relations, Kripke Frames, Labelled Transition Systems)** $x \simeq y$ iff they are bisimilar in the usual sense. That is, given $\xi : X \to \mathcal{P}(A \times X)$ and writing $x \xrightarrow{a} y$ for $(a, y) \in \xi(x)$, it holds $x \simeq y$ iff

$$x \simeq y \;\&\; x \xrightarrow{a} x' \;\Rightarrow\; \exists y' \,.\, y \xrightarrow{a} y' \;\&\; x' \simeq y'$$
$$x \simeq y \;\&\; y \xrightarrow{a} y' \;\Rightarrow\; \exists x' \,.\, x \xrightarrow{a} x' \;\&\; x' \simeq y'$$

For the proof of 1, using Proposition 2.3, it is enough to show that (a) $x \simeq y$ implies that the two streams produced by x and y are the same and (b) the set of streams produced by elements of X carries a coalgebra structure that makes it into a quotient of X (exercise!). Similarly for 2. For 3, 'only if' follows from the fact that coalgebra morphisms are bisimulations, 'if' follows from the fact that the two projections from the bisimulation to X are coalgebra morphisms (see the definition of bisimulation in the next section).

2.2 Other Notions of Bisimulation

In the following, we will consider other formalisations of bisimilarity and see 3 different ways of capturing the notion of bisimulation (rather than just bisimilarity[8]). We will also see that these notions essentially agree for coalgebras over sets.

[8] But one can define a notion of bisimulation that corresponds to behavioural equivalence, see Exercise 5.3.

All three notions of bisimulation considered below, can be motivated from the well-known observation that bisimilarity is a coinductively defined relation, in the sense that it is the largest fixed point[9] of a monotone operator. Let us look, as an example, at (unlabelled) transition systems, that is, $T = \mathcal{P}$. Then, given (X, ξ), bisimilarity on X is the largest fixed point of the operator

$$\Phi(R) = \{(x, y) \in X \times X \mid \forall x' \in \xi(x).\exists y' \in \xi(y).x'Ry' \ \ \& \ \ \forall y' \in \xi(y).\exists x' \in \xi(x).x'Ry'\}$$

In order to generalise this from transition systems to coalgebras for an arbitrary functor, we need to separate the part of the definition of Φ that uses T from the part that uses (X, ξ). We can write $\Phi(R)$ as

$$\xi^*\hat{\mathcal{P}}(R)$$

where $\xi^* = (\xi \times \xi)^{-1}$ and

$$\hat{\mathcal{P}}(R) = \{(A, B) \in \mathcal{P}X \times \mathcal{P}X \mid \forall a \in A.\exists b \in B.aRb \ \ \& \ \ \forall b \in B.\exists a \in A.aRb\} \qquad (2)$$

2.2.1 Bisimulation

Aczel and Mendler defined bisimulation in [3] as follows. $R \subseteq X \times X'$ is a *bisimulation* between coalgebras (X, ξ) and (X', ξ') if one can find a coalgebra structure ϱ on R such that the projections $X \leftarrow R \rightarrow X'$ become coalgebra morphisms.

$$\begin{array}{ccc} X & R & X' \\ {\scriptstyle \xi} & {\scriptstyle \varrho} & {\scriptstyle \xi'} \\ TX & TR & TX' \end{array}$$

x, x' are called bisimilar iff there is a bisimulation R such that xRx'. Note that ϱ need not be unique (eg for $T = \mathcal{P}$ (exercise)) and is not part of the structure of a bisimulation.

Example 2.5. Consider two coalgebras $\langle head, tail \rangle : X \to D \times X$, $\langle head', tail' \rangle : X' \to D \times X'$. Then R is a bisimulation iff

$$\begin{aligned} x\ R\ x' &\Rightarrow head(x) = head'(x') \\ x\ R\ x' &\Rightarrow tail(x)\ R\ tail'(x') \end{aligned}$$

It is not immediate that bisimilarity is an equivalence relation and it depends on T preserving weak pullbacks.[10]

Comparing bisimilarity with behavioural equivalence we would expect that bisimilarity on a coalgebra (X, ξ) and behavioural equivalence are the same, in other words, that bisimilarity is the kernel pair of the quotient wrt behavioural equivalence. In order to

[9]Here, inductively defined refers to smallest fixed point and coinductively to largest fixed point.

[10]The composition of two bisimulations is a bisimulation if ([45]) and only if ([22, 21]) T preserves weak pullbacks.

get the not necessarily unique arrow ϱ one needs to assume that T maps pullbacks to weak pullbacks (or, equivalently in any category with pullbacks, that T preserves weak pullbacks). The proof of the following proposition is straight forward. [11]

Proposition 2.6. *If* $T : \mathsf{Set} \to \mathsf{Set}$ *preserves weak pullbacks, then bisimilarity and behavioural equivalence coincide.*

Proof. That bisimilarity implies behavioural equivalence is immediate from the respective definitions. For the converse, consider

$$\begin{tikzcd}
 & R & & & TR & \\
X & & X & TX & & TX \\
 & Q & & & TQ &
\end{tikzcd}$$

where $X \to Q$ is the quotient wrt behavioural equivalence and R is the pullback. Since TR is a weak pullback, the required structure on R exists. □

2.2.2 Bisimulation via Relators

Let us write Rel for the category that has sets as objects and relations as arrows.[12] We denote by $(-)^\circ$ the operation that maps a relation $R : A \to B$ to its converse $R^\circ : B \to A$. Intuitively, a relator extends a functor from Set to Rel. This idea has been formalised in different ways. We follow [15]. The proofs taken from this paper are sketched in the exercises.

Note that every arrow in Set, called a map in this context , appears in Rel as its graph. We write f, g for maps in Rel and denote the projections of a relation $R : A \to B$ by $r_1 : R \to A$ and $r_2 : R \to B$. A **relator** Γ is a graph homomorphism $\mathsf{Rel} \to \mathsf{Rel}$ such that (a) $\Gamma(\mathrm{id}) = \mathrm{id}$, (b) $fR \subseteq R'g \;\Rightarrow\; (\Gamma f)(\Gamma R) \subseteq (\Gamma R')(\Gamma g)$,[13] (c) Γ preserves maps.

It follows that Γ is monotone ($R \subseteq R' \;\Rightarrow\; \Gamma R \subseteq \Gamma R'$) and that Γ induces a functor $\Gamma_\sharp : \mathsf{Set} \to \mathsf{Set}$ (ie Γ preserves composition of maps). Conversely, each functor $T : \mathsf{Set} \to \mathsf{Set}$ extends to a relator $\hat{T}$ given by $\hat{T}A = TA$ and and $\hat{T}R = (Tr_2)(Tr_1)^\circ$. Equivalently, given $R : A \to B$, $\hat{T}R$ is given by the image factorisation

$$TR \longrightarrow \hat{T}R \hookrightarrow TA \times TB$$

[11] Recall that the kernel pair of a morphism $X \to Y$ is the pullback of $X \to Y \leftarrow X$. A weak pullback is defined like a pullback but the mediating map need not be unique.

[12] The homsets of Rel are partially ordered and this order is respected by composition of relations, so Rel is a 2-category, or, more specifically, a Poset-enriched category.

[13] $fR \subseteq R'g$ iff $f \times g$ restricts to R iff $\{(f(x), g(x)) \mid (x, y) \in R\} \subseteq R'$ iff $R \subseteq (f \times g)^{-1}(R')$. Assuming monotonicity, (b) is equivalent to $\Gamma((f \times g)^{-1}(R')) \subseteq (\Gamma f \times \Gamma g)^{-1}(\Gamma R')$ where '$\subseteq$' can be replaced by an equality if Γ preserves composition.

of $TR \to T(A \times B) \to TA \times TB$.

As an example, $\hat{T}$ has been explicitly given for $T = \mathcal{P}$ in (2).

Given a T-coalgebra (X, ξ), one can now define a *bisimulation* on X to be a post fixed-point (and bisimilarity on X to be the largest fixed point) of the monotone operator $R \mapsto (\xi \times \xi)^{-1}(\hat{T}(R))$. Equivalently, a bisimulation is a subset $R \subseteq X \times X$ such that there is a (necessarily unique) map $\hat{\varrho}$ such that

$$\begin{array}{ccc} X & R & X' \\ \xi & \hat{\varrho} & \xi' \\ TX & \hat{T}R & TX' \end{array}$$

commutes.

The following theorem is not needed to compare the different notions of bisimilarity but it shows the role of 'weak pullback preservation' in the context of relators. It is a special case of a theorem of [15] where it is developed for arbitrary regular categories instead of Set. Note that for a relator to be a (2-)functor is the same as to preserve composition of relations.

Theorem 2.7 ([15, 4.3]). *Let T be a functor on Set and Γ a relator.*

1. *$\hat{T}$ is a functor iff T preserves weak pullbacks.*
2. *Γ is a functor then $\Gamma = \widehat{(\Gamma_\sharp)}$.*

The next proposition is the analogue of Proposition 2.6. But now, preservation of weak pullback is used for the other direction.

Proposition 2.8. *Given a coalgebra (X, ξ) for a weak pullback preserving functor T, behavioural equivalence on X is the largest fixed point of the monotone operator $R \mapsto (\xi \times \xi)^{-1}(\hat{T}(R))$.*

Proof. It is immediate from the respective definitions that Aczel-Mendler bisimulation on X is a post-fixed point of $R \mapsto (\xi \times \xi)^{-1}(\hat{T}(R))$. To show that, conversely, the largest fixed-point of $R \mapsto (\xi \times \xi)^{-1}(\hat{T}(R))$ is contained in the behavioural equivalence on X, consider the diagram of the proof of Proposition 2.6 with $\hat{T}R$ instead of TR. Recall that R was defined as kernel pair of $X \to Q$. We find an arrow $f : \hat{T}R \to P$ from $\hat{T}R$ to the the pullback P of $TX \to TQ \leftarrow TX$. Since T preserves weak pullbacks, the arrow $TR \to P$ has a half-inverse g, exhibiting R as an Aczel-Mendler bisimulation via $gf : R \to TR$. $\square$

2.2.3 Bisimulation via Relation Lifting

Another approach to bisimulation (or other coinductively defined relations) is to replace the carrier sets X of coalgebras (X, ξ) by pairs (X, R) where R is a relation on X and then lift the functor T from sets to sets with relations. To be specific, consider the category BPred of 'binary predicates' the objects of which are pairs (X, R), $R \subseteq X \times X$ and arrows are functions $X \to X'$ such that $xRy \;\Rightarrow\; f(x)R'f(y)$. A lifting of T is a functor $\bar{T}$ such that (p is first projection)

$$\begin{tikzcd} \mathsf{BPred} \arrow[r, "\bar{T}"] \arrow[d, "p"'] & \mathsf{BPred} \arrow[d, "p"] \\ \mathsf{Set} \arrow[r, "T"'] & \mathsf{Set} \end{tikzcd}$$

meaning that $\bar{T}(X, R) = (\bar{T}_0(X, R), \bar{T}_1(X, R))$ satisfies $\bar{T}_0(X, R) = TX$.[14]

It is convenient to abbreviate $(\bar{T}_0(X, R), \bar{T}_1(X, R))$ by $(T(X), \bar{T}(R))$.

There may be different choices for $\bar{T}$ giving rise to different liftings of T, but there is a canonical one, namely the one which is obtained from $TR \to T(X \times X) \to TX \times TX$ as the image $TR \to \bar{T}(X, R) \hookrightarrow TX \times TX$.

For example, with $T = \mathcal{P}$, $\bar{T}(R)$ is the $\hat{\mathcal{P}}(R)$ given in (2).

One can now define a *bisimulation* on a coalgebra (X, ξ) to be a relation R such that $((X, R), \xi)$ is a $\bar{T}$-coalgebra. Moreover, if T has a final coalgebra (Z, ζ), then $\bar{T}$ has a final coalgebra and it is given by $((Z, =), \zeta)$, in accordance with the fact that behavioural equivalence is the identity relation on the final coalgebra (see the next section).

It is immediate from the definitions that the relation lifting $\bar{T}$ and the relator $\hat{T}$ give rise to the same notion of bisimulation.

The above is only a particular instance of a much more general approach using fibrations, see Hermida and Jacobs [25]. A general theorem describing how final T-coalgebras are lifted to final $\bar{T}$-coalgebras is given in Hensel and Jacobs [24].

2.3 Further Remarks

A Comparison We have seen that, if the functor preserves weak pullbacks, all notions of bisimilarity or behavioural equivalence coincide. If the functor does not preserve weak pullbacks, the notions of bisimulation fall apart and do not work nicely anymore. For example, bisimilarity may fail to be an equivalence relation. But behavioural equivalence still works fine. So the point of view I would take, given the examples I am aware of, is the following. Given a functor $T : \mathsf{Set} \to \mathsf{Set}$, behavioural equivalence is the correct formalisation of the intuitive notion of bisimilarity; the other notions work only in case the functor preserves weak pullbacks.

[14] Moreover, $\bar{T}$ is often required to be fibred which amounts to $\bar{T}((f \times g)^{-1}(R')) = (\bar{T}f \times \bar{T}g)^{-1}(\bar{T}R')$.

Having said that, I want to emphasise that this judgement relies on choosing the base category Set. The case of functors on categories other than Set seems not to be much investigated. See Worrell [53] for work generalising relators for coalgebras over an enriched category and Plotkin [39] for some remarks on coalgebras over cpos.

Relators and Simulations The term 'relator' seems to be due to the Thijs [49]. His notion of monotonic relator or weak relator [10] insists on relators preserving composition but, more importantly, does not require that maps are preserved. This allows to treat simulations instead of bisimulations, see [10, 16]. A related approach to simulations based on relation liftings is proposed in [26].

Other Process Equivalences and Preorders The test suite approach by Klin [29, 30] shows how the fibrational approach can be used to characterise a large number of equivalences and preorders on processes other than (bi)simulation. Instead of BPred, it uses a category of 'test suites' which has objects (X, θ) where θ is a collection of subsets of X and, similar to topological spaces, $f : X \to X'$ is a morphism if $f^{-1}(a') \in \theta$ for all $a' \in \theta'$. As for relation liftings, one uses the lifted functor to define a monotone operator. The specialisation preorder of the largest fixed point is then the defined preorder. Klin develops a methodology how to define the lifted functors in order to obtain specific preorders of interest.

3 Final Coalgebras

3.1 Basic Definitions and Examples

Definition 3.1. An object A in a category $\mathcal{A}$ is called *final* or *terminal* if for any object B in $\mathcal{A}$ there is a unique arrow $B \to A$.

As any limit, a final coalgebra is determined uniquely up to isomorphism.

The final T-coalgebras can be considered as solutions of the 'domain equation' $X \cong TX$:

Proposition 3.2 (Lambek's lemma). *If $\zeta : Z \to TZ$ is a final coalgebra, then ζ is an isomorphism.*

Elements of the final coalgebra can be understood as equivalence classes of behaviourally equivalent states.

Proposition 3.3. *$x \simeq y$ iff x and y are mapped to the same element of the final coalgebra.*

Example 3.4. We select again from Example 1.4.

1. **(Largest Fixed Points)** Let $\mathcal{X}$ be a category of sets with inclusions as morphisms. Then the final T-coalgebra is the largest fixed point of T.
2. **(Streams)** If $TX = D \times X$, then the final coalgebra is $\langle head, tail\rangle : D^\omega \to D \times D^\omega$ where $head(l) = l(0)$ gives the first element of the infinite list l and $tail(l) = \lambda n \in \omega.l(n+1)$.
3. **(The Automaton of all Languages)** The set of all languages can be equipped with a transition structure that makes it into a final coalgebra. A language L is an accepting state if it contains the empty word. And L makes an a-transition to the language $L_a = \{w \mid aw \in L\}$. See Rutten [43].
4. **(Non well-founded sets)** Aczel's [2] universe of non-well founded sets is the final coalgebra for the covariant powerset functor (of course, due to Lambek's lemma, the carrier of this final coalgebra cannot be a set but is a proper class).

Coinduction

Final Coalgebras give rise to the principle of coinduction. Since we know that for any coalgebra $\xi : X \longrightarrow TX$ there is a *unique morphism* into the final coalgebra (Z, ζ), we can define a function $f : X \to Z$ just by giving an appropriate structure ξ:

$$\begin{tikzcd} X \arrow[r, "\xi"] \arrow[d, "f"'] & TX \arrow[d, "Tf"] \\ Z \arrow[r, "\zeta"'] & TZ \end{tikzcd}$$

We say that a function $f : X \to Z$ is defined by coinduction if it arises in such a way from a $\xi : X \to TX$.

For example, let us define the operation zipping two streams (recall Example 3.4). That is, we are looking for a function

$$zip : D^\omega \times D^\omega \to D^\omega$$

such that

$$head(zip(l_1, l_2)) = head(l_1) \tag{3}$$
$$tail(zip(l_1, l_2)) = zip(l_2, tail(l_1)) \tag{4}$$

Exercise 3.5. Show that *zip* is defined by coinduction via

$$\begin{aligned} \xi : D^\omega \times D^\omega &\to D \times D^\omega \times D^\omega \\ \langle l_1, l_2 \rangle &\mapsto \langle head(l_1), \langle l_2, tail(l_1) \rangle \rangle, \end{aligned}$$

More precisely, show that, for an arbitrary $zip : D^\omega \times D^\omega \to D^\omega$, *zip* is a morphism $(D^\omega \times D^\omega, \xi) \to (D^\omega, \zeta)$ iff it satisfies 3 and 4.

For another example do the following

Exercise 3.6. Find a function $\xi : D^\omega \to D \times D^\omega$ showing that

$$head(even(l)) = head(l) \tag{5}$$
$$tail(even(l)) = even(tail(tail(l))) \tag{6}$$

is a coinductive definition.

We can also use coinduction as a proof principle. It is based on the following

Proposition 3.7 (coinduction proof principle). *Behavioural equivalence is equality on the final coalgebra.*

It follows that, in order to show that two elements of a final coalgebra are equal, *it is enough to show that there is a bisimulation* relating them. This is called 'proof by coinduction'.

For an example, recall the functions *zip* and *even* and define $odd(x) = even(tail(x))$. We want to show

$$zip(even(x), odd(x)) = x.$$

It is not difficult to guess the bisimulation

$$R = \{\langle zip(even(x), odd(x)) \, , \, x \rangle \, , \; x \in D^\omega\}.$$

It remains to check the two clauses of Example 2.5 (exercise!).

Remark 3.8. Coinduction becomes really interesting only if we also consider algebraic operations. For example, if D is a ring, we can define operations like addition and multiplication of streams coinductively. In fact, one can go much further and Rutten [44] developed a coinductive calculus of streams to solve so-called behavioural differential equations. Another area is process algebra where operations on processes are defined in the style of SOS, ie, coinductively, see Turi and Plotkin [50] and, for a recent account, Bartels [12].

3.2 The Final Coalgebra Sequence

The *final coalgebra sequence*, or *final sequence*, or *terminal sequence*, can be pictured as

$$1 \qquad T1 \qquad \cdots \qquad T^n 1 \qquad \cdots \qquad T^\omega 1 \qquad T(T^\omega 1) \qquad \cdots$$

We write T_n for $T^n 1$ and define, for a sufficiently complete category $\mathcal{X}$, the final sequence of T as an ordinal indexed sequence of sets (T_n) together with a family $(p^n_m)_{m \le n}$ of arrows $p^n_m : T_n \to T_m$ for all ordinals $m \le n$ such that

- $T_{n+1} = TT_n$ and $p^{n+1}_{m+1} = Tp^n_m$ for all $m \le n$
- $p^n_n = \mathrm{id}_{T_n}$ and $p^n_k = p^m_k \circ p^n_m$ for $k \le m \le n$.
- The cone $(T_n, (p^n_m))_{m<n}$ is limiting whenever n is a limit ordinal.

The final sequence has many applications. It can be used to prove the existence of final coalgebras. It allows to reduce coinduction to induction along the final sequence. It can be used to define a metric on T_ω. T_n is also the natural semantic domain for formulae of modal logic of depth n.

The fundamental observation here is that any coalgebra $\xi : X \to TX$ gives rise to a cone over the final sequence

$$\begin{array}{l} X \\ \xi_0 \quad \xi_1 \qquad \xi_n \qquad \xi_\omega \\ 1 \qquad T1 \qquad \cdots \qquad T^n 1 \qquad \cdots \qquad T^\omega 1 \qquad \cdots \end{array}$$

where $\xi_n : C \to T^n 1$ is $T\xi_m \circ \xi$ if $n = m + 1$ is a successor ordinal and ξ_n is the unique map satisfying $\xi_m = p^n_m \circ \xi_n$ for all $m < n$ if n is a limit ordinal.

Example 3.9. If $TX = D \times X$, then the final sequence 'terminates' after ω steps since $T_\omega = D^\omega$ is the final coalgebra (cf. Example 3.4).[15] The finitary approximants are $T_n = D^n$ and $\xi_n(x)$ gives the list of the first n outputs, ie, forgets from the behaviour of x all but the first n steps (cf. Examples 1.4, 2.4).

[15]More precisely, the inverse of $T(T_\omega) \to T_\omega$ is the final coalgebra.

3.2.1 Approximating Final Coalgebras

The example suggests that the T_n should be considered as approximating the final coalgebra; and the elements of T_n as behaviours up to n steps. Indeed, we have the following

Proposition 3.10. *If, for some ordinal n, the arrow $p_n^{n+1} : T(T_n) \to T_n$ is an isomoprhism, then the inverse $(p_n^{n+1})^{-1}$ is a final coalgebra.*

Proof. That, given a coalgebra (X, ξ), ξ_n is a coalgebra morphism follows from $\xi_n = p_n^{n+1} \circ T\xi_n \circ \xi$. For uniqueness suppose $f : X \to T^n 1$ is a coalgebra morphism and let $f_m = p_m^n \circ f$. One shows that $f_m = \xi_m$ for all $m \leq n$. The step for a successor ordinal is $f_{m+1} = p_{m+1}^n \circ f = p_{m+1}^n \circ p_n^{n+1} \circ Tf \circ \xi = p_{m+1}^{n+1} \circ Tf \circ \xi = T(p_m^n) \circ Tf \circ \xi = Tf_m \circ \xi$. □

This theorem also has a converse.

Theorem 3.11 (Adámek and Koubek [5]). *Let $\mathcal{X}$ be cocomplete and cowellpowered.*[16] *If the final coalgebra exists, then the final sequence terminates.*

Remark 3.12 (Existence of Final Coalgebras). Sufficient conditions for the final coalgebra (over Set) to exist are that T is bounded [45] or that T is accessible [7]. Both notions are in fact equivalent [6].

Remark 3.13 (Reducing Coinduction to Induction). Given two states x, y in a coalgebra (X, ξ), in order to establish that they are behaviourally equivalent, we usually employ a proof by coinduction (ie we find an appropriate bisimulation). But we can also use induction along the final sequence to establish $\xi_n(x) = \xi_n(y)$ for all ordinals. In fact, that is what one often does to establish soundness of proofs by coinduction. For example, going back to streams, one way to proof that the existence of a bisimulation (Example 2.5) relating x and y implies $(head(tail^n(x))_{n<\omega} = (head(tail^n(y))_{n<\omega}$ (ie behavioural equivalence) is by induction on $n < \omega$. The fact that the induction has to cover only natural numbers, corresponds to the final sequence terminating at ω.

To summarise, we can consider the elements of the final sequence $T^n 1$ as approximants to the final coalgebra. This makes sense even if the final coalgebra does not exist.

Example 3.14. If $TX = \mathcal{P}X$, then the final sequence does not terminate. $\mathcal{P}^\omega 1$ is known as the final coalgebra for the compact powerset on complete ultrametric spaces or also for the convex powerdomain on Stone spaces. In order to obtain an explicit description of $\mathcal{P}^\omega 1$ (see Worrell [52] for the full story), observe that an element of $\mathcal{P}^n 1$ can be considered as a tree of depth n where x is a child of y if $x \in y$. The projections p_m^n cut a tree at depth m and then quotient it so that it depicts again a set. $\mathcal{P}^\omega 1$ contains all trees 'that can be built using an infinite sequence of such trees (t_n) of finite depth'. It is not difficult to see that $\mathcal{P}^\omega 1$ has an infinitely branching tree.

[16] Cowellpowered means that any object X there is, up to isomorphism, only a set of epis with domain X.

Now consider the finitary powerset functor $\mathcal{P}_\omega$. Obviously, $\mathcal{P}_\omega^\omega 1 = \mathcal{P}^\omega 1$. Since we have indicated in the example above that $\mathcal{P}^\omega 1$ has infinitely branching trees, we cannot expect $\mathcal{P}^\omega 1$ to carry a $\mathcal{P}_\omega$-coalgebra structure.[17] But on the other hand, since $\mathcal{P}_\omega$ is finitary, we know that a final coalgebra exists and appears in the final sequence. In fact, one needs another ω iterations to cut out, at each step, the infinitely branching nodes. This is the significance of the following theorem.

Theorem 3.15 (Worrell [52]). *For a finitary $T : \mathsf{Set} \to \mathsf{Set}$ the final sequence terminates after $\omega + \omega$ steps.*

3.2.2 The Metric Induced by the Final Sequence

Consider a functor $T : \mathsf{Set} \to \mathsf{Set}$. Every cone (X, ξ_n) on the finitary part of the final sequence induces a (pseudo)-metric on X, namely $d(x, y) = 2^{-n}$ where n is the smallest number such that $\xi_n(x) \neq \xi_n(y)$. In particular, $T^\omega 1$ is a (ultra)metric space.

The following theorem exhibits $T^\omega 1$ as a metric completion. We write $T^n 0$ for the elements of the initial sequence which is defined dually to the final sequence.

Theorem 3.16 (Barr [11]). *If $T0 \neq 0$ and T preserves monos then the canonical map $T^\omega 0 \to T^\omega 1$ is injective and $T^\omega 1$ is the Cauchy-completion of $T^\omega 0$.*

For example, in the theory of the infinitary lambda calculus one usually defines sets of infinite terms as metric completions of finite terms. Using the theorem above, one can show that these definitions are equivalent to certain coinductive definitions.

The topology on $T^\omega 1$ can also be used to study finitary logics for coalgebras. For simplicity, let us say that a finitary logic for coalgebras consists of a set of formulae and each formula denotes a subset of some $T^n 1$, $n < \omega$.[18] For example, assuming that all $T^n 1$ are finite, one can then show that the logic is compact iff the functor T weakly preserves limits of ω-chains [34].

[17] But, since $\mathcal{P}^{\omega+1} 1 \to \mathcal{P}^\omega 1$ is surjective, it has a right-inverse which is a $\mathcal{P}$-coalgebra structure on $\mathcal{P}^\omega 1$.
[18] Then we can say that $(X, \xi), x \models \varphi$ iff $\xi_n(x)$ in the denotation of φ.

4 Stone Duality and Logics for Coalgebras

4.1 Stone Duality

The main reference for Stone duality is Johnstone's book on Stone Spaces [28] which also provides detailed historical information. The handbook article [1] covers the topic from the point of view of domain theory. Three introductory textbooks on the subject, all of which I find very helpful and complementing each other are: Vickers [51], Davey and Priestley [17], Brink and Rewitzky [14]. I will just give a very brief sketch which, hopefully, is sufficient for what follows.

A **topological space** $(X, \mathcal{O})$ is a set X together with a collection $\mathcal{O}$ of subsets of X closed under finite intersections and arbitrary unions. Elements $a \in \mathcal{O}$ are called *open* sets. A function $(X, \mathcal{O}) \to (X', \mathcal{O}')$ is *continuous* if f^{-1} preserves opens, that is, restricts to a map $\mathcal{O}' \to \mathcal{O}$. Topological spaces and continuous maps form the category Top. Note that f^{-1} preserves finite intersections and arbitrary unions.[19]

Abstracting from the set of points X and axiomatising the algebraic properties of a topology $\mathcal{O}$, one arrives at the following notion. A **frame** A is a distributive lattice (with bottom $\bot$ and top $\top$) with infinite joins satisfying the infinite distributive law

$$a \wedge \bigvee C = \bigvee \{a \wedge c \mid c \in C\}$$

for all $a \in A$ and all subsets $C \subseteq A$. Frames with functions preserving arbitrary joins and finite meets form the category Frm. Frm has free algebras, in other words, the forgetful functor from Frm to Set mapping each frame to its underlying set is monadic.

There is an obvious contravariant functor

$$\begin{aligned} P : \mathsf{Top} &\longrightarrow \mathsf{Frm} \\ (X, \mathcal{O}) &\mapsto \mathcal{O} \\ f &\mapsto f^{-1} \end{aligned}$$

denoted P since we think of P as associating to a space X the set of 'admissible' predicates over X. If $(X, \mathcal{O})$ is a discrete[20] topological space, then P is the contravariant powerset functor.

There is also a way to associate a space to each frame. Similarly to defining points on the real line via collections of intervals, one can define a point of a frame A via a collection of elements of A. More precisely, define the set of points of A as the set of frame-morphisms

$$Pt(A) = \mathsf{Frm}(A, \mathbb{2})$$

[19] $f^{-1} : \mathcal{P}X' \to \mathcal{P}X$ preserves complements and arbitrary intersections and unions.

[20] That is, $\mathcal{O} = \mathcal{P}X$.

where $\mathbb{2}$ is the two element frame (consisting of $\bot, \top$). $Pt(A)$ carries a natural topology $\mathcal{O}_{Pt(A)}$, namely the one generated by the sets, for each $a \in A$,

$$\{p \in Pt(A) \mid p(a) = \top\}$$

Here, one should think of a point $p : A \to \mathbb{2}$ as a characteristic function and read $p(a) = \top$ as '$a \in p$'. This gives rise to a contravariant functor

$$\begin{aligned} S : \mathsf{Frm} &\longrightarrow \mathsf{Top} \\ A &\mapsto (Pt(A), \mathcal{O}_{Pt(A)}) \\ f &\mapsto \lambda p \in Pt(A) \,.\, p \circ f \end{aligned}$$

denoted S since we think of SA as the *s*pace consisting of the points, or *s*tates, associated with A.

Theorem 4.1. *The contravariant functors P, S are adjoint on the right, that is, there is a bijection*

$$\mathsf{Top}(X, SA) \cong \mathsf{Frm}(A, PX)$$

natural in X and A.

This duality restricts to a duality on many interesting subcategories, see Abramsky and Jung [1]. We just mention the one example that will be referred to in the sequel.

Example 4.2. A topological space is a Stone space if it is Hausdorff, compact, and has a basis of clopens.[21] Stone is the category of Stone spaces and continuous maps. The category BA of Boolean algebras fully embeds into the category of frames by mapping a Boolean algebra A to the free frame over A.

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

P' maps a Stone space X to the set of all clopens $P(X, 2)$ where 2 is here the two element (discrete) space. S' maps a Boolean algebra A to the space that has as a carrier the set of 'ultrafilters' $\mathsf{BA}(A, \mathbb{2})$.

4.2 Modal Logic - The Coalgebraic Perspective

Systems are given by coalgebras for a functor T. We will show that adequate logics for these systems are given by the functor L that is dual to T.

[21]A basis is a collection of opens that is closed under finite intersections and, moreover, every open is a union of basic ones.

For the general framework we assume that we have a duality

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

which arises from the duality of Top and Frm by restricting to subcategories $\mathcal{X}$ and $\mathcal{A}$. Moreover, we will need that $\mathcal{A}$ is a category of algebras for a signature and equations over Set that has free algebras (ie, $\mathcal{A}$ is monadic over Set, see Theorem 1.10).

Definition 4.3 (Dual functor). L is dual to T if there is a natural isomorphism $\delta : LP \to PT$.

δ allows us to consider the collection of predicates on a coalgebra as an L-algebra. That is, we can extend the picture from the previous section lifting the functors P and S to an equivalence of algebras and coalgebras.[22] We assume that $U : \mathcal{A} \to \mathsf{Set}$ is monadic and denote by F its left adjoint.

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

The category $\mathsf{Alg}(L)$ provides us with a canonical set of propositions, namely the initial algebra, or more generally, if we take variables into account, the free algebras.

Definition 4.4 ($\mathsf{Prop}(T, V)$). Assume that $U : \mathsf{Alg}(L) \to \mathsf{Set}$ has a left adjoint. We write $\mathsf{Prop}(T, V)$ for the (carrier of the) algebra free over (the set of variables) V and call the elements of $\mathsf{Prop}(T, V)$ propositions over V.

The algebraic semantics is defined in the usual way. Recall that there is a bijection between functions $V \to UA$ and morphisms $\mathsf{Prop}(T, V) \to A$.

Definition 4.5 (algebraic semantics of $\mathsf{Prop}(T, V)$). The algebraic semantics $\varphi^{A,h}$ of $\varphi \in \mathsf{Prop}(T, V)$ wrt an algebra $A \in \mathsf{Alg}(L)$ and a valuation of variables $h : V \to UA$ is

$$\varphi^{A,h} = h^{\#}(\varphi)$$

where $h^{\#} : \mathsf{Prop}(T, V) \to A$ is the unique extension of h. A formula φ holds in an algebra A, denoted $A \models \varphi = \top$, if $h(\varphi) = \top$ for all $h : FV \to A$.

---

[22] Explicitly, on objects, the lifted P and S are given as

$$\begin{aligned} P(X, \xi) &= LPX \xrightarrow{\delta_X} PTX \xrightarrow{P\xi} PX \\ S(A, \alpha) &= SA \xrightarrow{S\alpha} SLA \cong SLPSA \xrightarrow{(S\delta S)_A} SPTSA \cong TSA \end{aligned}$$

The semantics of a proposition in a coalgebra is an element of the algebra of propositions of the coalgebra.

Definition 4.6 (coalgebraic semantics of $\mathsf{Prop}(T, V)$). The semantics $[\![\varphi]\!]_{(X,\xi)}$ of a formula $\varphi \in \mathsf{Prop}(T, V)$ wrt a coalgebra $(X, \xi) \in \mathsf{Coalg}(T)$ and a valuation $h : V \to PX$ is given by

$$[\![\varphi]\!]_{(X,\xi,h)} = \varphi^{P(X,\xi),h}$$

Remark 4.7. 'valuations' $h : V \to UPX$ are in one-to-one correspondence with 'colourings' $X \to \prod_V 2$.

Proposition 4.8 (invariance under behavioural equivalence). *The propositions in* $\mathsf{Prop}(T, V)$ *are invariant under behavioural equivalence.*

Proof. We have to show that, given a coalgebra morphism $f : (X, \xi) \to (X', \xi')$ and a valuation $v' : V \to PX'$ and $x \in X$, that

$$x \in [\![\varphi]\!]_{(X,\xi,v)} \;\Leftrightarrow\; f(x) \in [\![\varphi]\!]_{(X',\xi',v')}.$$

where $v = f^{-1} \circ v'$. This follows immediately from

$$\begin{tikzcd}
 & P(X,\xi) \\
\mathsf{Prop}(T,V) \arrow[ur, "{[\![-]\!]_{(X,\xi,v)}}"] \arrow[dr, "{[\![-]\!]_{(X',\xi',v')}}"'] & \\
 & P(X',\xi') \arrow[uu, "{Pf=Pf=f^{-1}}"']
\end{tikzcd}$$

which commutes due to the universal property of the free algebra $\mathsf{Prop}(T, V)$. $\square$

Proposition 4.9 (equivalence of algebraic and coalgebraic semantics).
$\mathsf{Alg}(L) \models \varphi = \top \;\Leftrightarrow\; \mathsf{Coalg}(T) \models \varphi$

Proof. ' $\Rightarrow$ ': For any $(X, \xi) \in \mathsf{Coalg}(T)$, we have $\mathsf{Alg}(L) \models \varphi = \top \;\Rightarrow\; P(X, \xi) \models \varphi = \top \;\Rightarrow\; (X, \xi) \models \varphi$.
' $\Leftarrow$ ': Suppose $\not\models_{\mathsf{Alg}(L)} \varphi = \top$, ie, $\varphi \neq \top$ in $\mathsf{Prop}(T, V)$. To establish the existence of a counter-model for φ, we have to find a coalgebra (X, ξ) and an injective morphism

$$\mathsf{Prop}(T, V) \xrightarrow{v} P(X, \xi)$$

(then $v(\varphi) \neq \top$, hence $(X, \xi, v) \not\models \varphi$). We can choose for (X, ξ) the coalgebra cofree over $PF(V)$,[23] or equivalently, the dual of $\mathsf{Prop}(T, V)$. $\square$

[23]This is the final coalgebra for $V = \emptyset$

Proposition 4.10 (expressiveness). *If two elements x, x' of two coalgebras $(X, \xi), (X', \xi')$ are not behavioural equivalent, then there is $\varphi \in \mathsf{Prop}(T, \emptyset)$ separating x and x', that is, $x \models \varphi \ \Leftrightarrow\ x' \not\models \varphi$.*

Proof. Without loss of generality, let us assume that x, x' are elements of the final coalgebra (Z, ζ). Now the proposition follows from

$$\mathsf{Prop}(T, \emptyset) \to P(Z, \zeta)$$

being surjective (and PZ being a T_0-space). $\square$

Remark 4.11. The reader may have noticed that, despite the duality, the treatment of this section is not symmetric. For example, we emphasised the natural transformation $LP \to PT$ and not $SL \to TS$ and we used two different arguments for the two directions of the equivalence of algebraic and coalgebraic semantics. One of the reasons for doing so, is that our arguments can be carried through in cases where we don't have a duality. Consider, for example, the case of classical modal logic where the coalgebras (Kripke frames) are over Set but the logic (classical, propositional) is over BA. We still have the functors P, S but they do not form an adjunction since the obvious map $(X, \xi) \to SP(X, \xi)$ is not a coalgebra morphism. And L and T are not dual. But there is still an injective natural transformation $\delta : LP \to PT$. Reworking the arguments above, one shows that Propositions 4.8 and 4.9 still hold (but not Proposition 4.10). In particular, to establish $\mathsf{Coalg}(T) \models \varphi \ \Rightarrow\ \mathsf{Alg}(L) \models \varphi = \top$, we can still use $S(\mathsf{Prop}(T, V))$ as a canonical counter model[24] (although it is not a final coalgebra).

Another possibility to obtain completeness for Set from a completeness for $\mathcal{X}$ is to restrict attention to those functors on Set that have a corresponding functor on $\mathcal{X}$ and then argue that every $\mathcal{X}$-coalgebra is also Set-coalgebra.

4.3 Presenting Algebras - Equational Logic for Coalgebras

We called $\mathsf{Prop}(T, V)$ a logic since it has logical operations inherited from the algebraic structure and a relational semantics. What is still missing is an explicit inductive construction of the set of formulae and a notion of derivability. The elements of $\mathsf{Prop}(T, V)$ will turn out to be equivalence classes of formulae under inter-derivability. Conversely, the formulae together with the notion of derivability can be seen as a concrete representation of $\mathsf{Prop}(T, V)$.

The technique to achieve such a presentation uses that algebras can be represented by 'generators and relations', or, more precisely, that every algebra is the quotient of an algebra generated freely by a set of generators and quotiented by a congruence relation. The generators then give rise to the modal operators and the relations to the axioms of the logic.

[24] Indeed, $S(\mathsf{Prop}(T, V))$ is known as the canonical model in modal logic.

4.3.1 Presenting Algebras by Generators and Relations

Suppose we have a forgetful functor $U : \mathcal{A} \to \mathsf{Set}$ that has a left-adjoint F. Then the counit

$$\varepsilon_A : FUA \to A$$

gives us a canonical (albeit not economical) presentation of A, namely generated by the elements of UA and quotiented by the kernel $\varrho_A = \{(t,s) \mid \varepsilon_A(t) = \varepsilon_A(s)\}$ of ε_A. These presentations are useful to describe operations on algebras. For example, whereas the product of algebras is simply given by the cartesian product of the carriers, the coproduct of algebras is more involved.

Example 4.12 (Coproduct of Algebras). Assume that $U : \mathcal{A} \to \mathsf{Set}$ is monadic. Then the coproduct $A_1 + A_2$ is generated from the elements of A_1 and A_2 and then quotiented by the smallest congruence containing all relations already present in A_1 and in A_2

$$A_1 + A_2 \;\cong\; F(UA_1 + UA_2)/\varrho_{A_1} \cup \varrho_{A_2}.$$

Proof (Sketch): Let $q : F(UA_1 + UA_1) \to F(UA_1 + UA_2)/\varrho_{A_1} \cup \varrho_{A_2}$ be the quotient wrt to the smallest congruence generated by $\varrho_{A_1} \cup \varrho_{A_2}$ and consider the insertions of generators $\kappa_i : UA_i \to UF(UA_1 + UA_2)$. The $Uq \circ \kappa_i$ are not only maps but also homomorphisms since the quotient q has just been defined so as to make $Uq \circ \kappa_i$ preserve all the equations ϱ_{A_i} that hold between elements of A_i. The universal propety of the coproduct follows from the universal property (freeness) of $F(UA_1 + UA_2)$ and the fact that any homomorphism $A_i \to B$ has to preserve the equations ϱ_{A_i} that hold between elements of A_i.

As an illustration of this construction consider the product of topological spaces $(X_1, \mathcal{O}_1) \times (X_2, \mathcal{O}_2) = (X_1 \times X_2, \mathcal{O}_1 + \mathcal{O}_2)$ where $\mathcal{O}_1 + \mathcal{O}_2$ is the topology on $X_1 \times X_2$ generated by subsets $a_1 \times X_2$, $X_1 \times a_2$, $a_i \in \mathcal{O}_i$. $\mathcal{O}_1 + \mathcal{O}_2$ is indeed a coproduct of frames and can be presented, as in the example above, by generators $\kappa_i(a) = (i,a)$, $i = 1,2$, $a \in \mathcal{O}_i$, and relations $(i, a \wedge a') = (i,a) \wedge (i,a')$, $(i, \bigvee a_j) = \bigvee (i, a_j)$.

Example 4.13 (Modal Algebras). A modal algebra, or Boolean algebra with operator (BAO), is the algebraic structure required to interpret (classical) modal logic which consists of propositional logic plus a unary modal operator $\Box$ preserving finite conjunctions. Modal algebras are therefore algebras for the functor $\mathcal{V} : \mathsf{BA} \to \mathsf{BA}$, where $\mathcal{V}A$ is defined by generators $\Box a$, $a \in A$, and relations $\Box\top = \top$, $\Box(a \wedge a') = \Box a \wedge \Box a'$.

The fact that $\mathcal{V}$ is on BA (and not on Set) takes care of the propositional part of modal logic. Also observe that the definition of $\mathcal{V}A$ can be phrased more abstractly by saying that the insertion of generators $\Box : UA \to U\mathcal{V}A$ is a universal meet preserving function, that is,

$$\begin{tikzcd} U\mathcal{V}A \arrow[r, "Uf^{\#}"] & UB \\ UA \arrow[u, "\Box"] \arrow[ur, "f"'] & \end{tikzcd}$$

for all $B \in \mathsf{BA}$ and all meet preserving functions $f : UA \to UB$ there is a unique Boolean algebra morphism $f^\# : \mathcal{V}A \to B$ with $Uf^\#(\Box a) = f(a)$.[25] From this observation it is straightforward to show that $\mathsf{Alg}(\mathcal{V})$ is indeed isomorphic to the category of BAOs as usually defined, see Exercise 5.5.

Definition 4.14 (Presentation). Let $U : \mathcal{A} \to \mathsf{Set}$ be monadic with left adjoint F. A *presentation* $\langle G, R\rangle$ consists of a set of 'generators' G and a set of 'relations' $R \subseteq UFG \times UFG$. A morphism $f : FG \to B$ *satisfies the relations* R if $(t, s) \in R \Rightarrow Uf(t) = Uf(s)$. An algebra A is *presented* by $\langle G, R\rangle$ if

$$\begin{tikzcd} FG \arrow[d, "in"'] \arrow[dr, "f"] & \\ A \arrow[r, "f^+"'] & B \end{tikzcd}$$

- A comes with an insertion of generators $in : G \to UA$ (or, equivalently, $in : FG \to A$) satisfying the relations R,
- for all $B \in \mathcal{A}$ and all $f : FG \to B$ satisfying the relations R there is a unique $f^+ : A \to B$ with $f^+ \circ in = f$.

Proposition 4.15. *Every presentation presents an algebra.*

Proof. The proof relies on the fact that, as a category monadic over Set, $\mathcal{A}$ is cocomplete (eg [13][II.4.3.5]). The object presented by $\langle G, R\rangle$ is given by the coequaliser

$$\begin{tikzcd} FR \arrow[r, shift left] \arrow[r, shift right] & FG \arrow[r, "q"] & A. \end{tikzcd}$$

(More concretely, q is the quotient wrt the smallest congruence containing R.[26]) $\square$

4.3.2 Presenting Functors by Operations and Equations

The examples above show how the functors $- + - : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ and $\mathcal{V} : \mathsf{BA} \to \mathsf{BA}$ can be described using generators and relations. These descriptions are uniform, or parametric, in the arguments. In the following we will make this uniformity explicit. The aim is to show how generators give rise to modal operators and relations to modal axioms.

Definition 4.16. A *presentation of a funcor* $L : \mathcal{A} \to \mathcal{A}$ *on a monadic category* $\mathcal{A}$ *by operations and equations* consists of

1. a signature Σ of operations $\sigma \in \Sigma$ with arities n_σ which we also describe by a functor $G : \mathsf{BA} \to \mathsf{Set}$, $A \mapsto \coprod_\Sigma \prod_{n_\sigma} UA$

[25] Or still more abstractly, $\mathcal{V}A$ is the free Boolean algebra over the meet-semilattice A.

[26] The coequaliser q can be obtained by factoring the family (called a 'source' in [4]) of all morphisms $s_i : FG \to A_i$ with $s_i \circ \pi_1 = s_i \circ \pi_2$ into $m_i \circ q$ where q is epi and the m_i are 'jointly-mono'. In terms of kernels, that factorisation means $\mathrm{Ker}(q) = \bigcap\{\mathrm{Ker}(s_i)\}$, ie q is the quotient wrt to the smallest congruence containing R.

2. a set of variables V and set of equations $E \subseteq UFGV \times UFGV$.

A natural transformation $f : FG \to M$ *satisfies the equations* E if for all $A \in \mathcal{A}$ and all $v : FV \to A$ it holds $(t, s) \in E \;\Rightarrow\; (f_A \circ FGv)(t) = (f_A \circ FGv)(s)$. A functor L is *presented* by $\langle G, V, E\rangle$ if

$$\begin{tikzcd} FGFV \arrow[d, "FGv"'] & \\ FGA \arrow[d, "in_A"'] \arrow[dr, "f_A"] & \\ LA \arrow[r, "f_A^+"'] & MA \end{tikzcd}$$

- L comes with a natural transformation, called insertion of generators, $in : G \to UL$ (or, equivalently, $in : FG \to L$) satisfying the equations E,
- for any functor $M : \mathsf{BA} \to \mathsf{BA}$ and natural transformation $f : FG \to M$ satisfying the equations in E there is a unique natural transformation $f^+ : L \to M$ such that $f^+ \circ in = f$.

Proposition 4.17. *Each presentation presents a functor.*

Proof. Given a presentation $\langle G, V, E\rangle$ we define the functor L on objects A as

$$\begin{tikzcd} FE \arrow[r, shift left] \arrow[r, shift right] & FGFV \arrow[r, shift left] \arrow[r, shift right] & FGA \arrow[r, "in_A"] & LA \end{tikzcd}$$

where q is the 'multiple coequaliser' wrt to all pairs $(\pi_1^{\#} \circ FGv, \pi_2^{\#} \circ FGv)$, $v : FV \to A$. The universal property of LA gives the action of L on morphisms and the naturality of in_A. □

Remark. A weaker notion of presenting a functor by generators and relations can be obtained from Definition 4.14 by, essentially, just promoting R and G to functors $\mathsf{BA} \to \mathsf{Set}$. The proposition then still holds. The reason for insisting on the stronger notion is the following theorem.

Since $\mathcal{A}$ is monadic over Set, we can assume that we a have a presentation of $\mathcal{A}$ as a category of algebras $\mathsf{Alg}(\Sigma_{\mathcal{A}}, E_{\mathcal{A}})$ given by a signature $\Sigma_{\mathcal{A}}$ and equations $E_{\mathcal{A}}$. Furthermore, a presentation $\langle G, V, E\rangle$ of a functor L gives rise to a signature Σ_L and equations E_L as follows. Σ_L is simply the signature describing G (see Definition 4.16.1). E_L is the set of equations on $(\Sigma_{\mathcal{A}} + \Sigma_L)$-terms obtained from the set of equations E (see Definition 4.16.2). thus giving us an equational logic for the functor L-algebras.

That $E_{\mathcal{A}} + E_L$ is a sound and complete logic for L-algebras is the content of the next theorem.

Theorem 4.18. $\mathsf{Alg}(\Sigma_{\mathcal{A}} + \Sigma_L, E_{\mathcal{A}} + E_L)$ *is isomorphic to* $\mathsf{Alg}(L)$.

4.4 Modal Logics for Coalgebras

In this section, we translate back the equational logic to a modal logic. To keep things simple we restrict attention to the duality BA and Stone which corresponds to adding a modal logic to classical propositional logic. Using the implication there is an obvious translation between algebraic equations and modal formulae. An equation $t = s$ corresponds to the formula $t \leftrightarrow s$ and a formula φ corresponds to the equation $\varphi = \top$. This correspondence also extends from equations/formulae to proofs in equational logic/modal logic. This gives us the syntax and a calculus of a modal logic. The coalgebraic semantics is determined by the duality of the functors L and T. To summarise:

Definition 4.19 (Modal logic for T-coalgebras). Let T be a functor on Stone, L its dual on BA and $(\Sigma_{\mathsf{BA}} + \Sigma_L, E_{\mathsf{BA}} + E_L)$ be the signature and equations obtained from a presentation of L by operations and equations (Definition 4.16). Operations in Σ_{BA} are called propositional connectives and operations in Σ_L are called modal operators. Following established notation, we often write modal operators $\sigma \in \Sigma_L$ as $[\sigma]$.

Formulae The set of formulae over a set *Propvar* of propositional variables is the smallest set containing *Propvar* and closed under operations in Σ_{BA} and Σ_L.

Axiom Schemes All substitution instances of the equations in E_{BA} and E_L.

Calculus A proof system for propositional logic plus, for each n-ary modal operator $\sigma \in \Sigma_L$, the rule

$$\frac{\varphi_1 \leftrightarrow \psi_1 \quad \ldots \quad \varphi_n \leftrightarrow \psi_n}{\sigma(\varphi_1, \ldots, \varphi_n) \leftrightarrow \sigma(\psi_1, \ldots, \psi_n)}$$

Semantics Given a coalgebra (X, ξ) and a valuation v, the semantics $[\![\varphi]\!]_{(X,\xi,v)}$ of a formula is defined inductively on the structure of formulae. Propositional variables and connectives are clear. For an n-ary modal operator $\sigma \in \Sigma_L$ its semantics is given by

$$(UPX)^n \longrightarrow GUPX \xrightarrow{in_A} ULPX \xrightarrow{\delta_X} UPTX \xrightarrow{P\xi} UPX$$

mapping $[\![\varphi_1]\!]_{(X,\xi,v)}, \ldots [\![\varphi_n]\!]_{(X,\xi,v)}$ to $[\![[\sigma](\varphi_1, \ldots \varphi_n)_{(X,\xi,v)}]\!]$.

Remark 4.20. Pattinson [38] described, for functors T on Set, the semantics of (unary) modal operators via predicate liftings $2^X \to 2^{TX}$. They appear here as $UPX \to GUPX \to UPTX$. See [31] for more information.

Theorem 4.21. *Let T be a functor on* Stone *and $\mathcal{L}$ the logic given by a presentation of the dual functor. Then the formulae of $\mathcal{L}$ are invariant under behavioural equivalence and $\mathcal{L}$ is sound, complete and expressive.*

Proof. We first check that equational deduction is equivalent to deduction in $\mathcal{L}$. It then follows from Theorem 4.18 that $\mathsf{Prop}(T, V)$ is a quotient of the set of $\mathcal{L}$-formulae wrt to the interderivability $\vdash_{\mathcal{L}} \varphi \leftrightarrow \psi$ (the so-called Lindenbaum-Tarski algebra of $\mathcal{L}$). It remains to check that the coalgebraic semantics of $\mathcal{L}$ coincides with the one from Definition 4.18. After having established the relationship between equational and modal logic, soundness and completeness is Proposition 4.9. Invariance under behavioural equivalence and expressiveness are Propositions 4.8 and 4.8, respectively. $\square$

5 Exercises

The following (rather unsystematic selection of) 'exercises' contain further material that has not made it into the short course of 5 lectures.

Structure of $\mathsf{Coalg}(T)$

Colimits in $\mathsf{Coalg}(T)$ are calculated as in the base category. Also, a coalgebra-morphism is epi in $\mathsf{Coalg}(T)$ iff it is epi in the base category. This follows from

Exercise 5.1 (structure of coalgebras). Show that the forgetful functor $U : \mathsf{Coalg}(T) \to \mathcal{X}$ creates colimits. That is, for a diagram $D : \mathcal{I} \to \mathsf{Coalg}(T)$, if $d : UDi \to X$ is a colimiting cocone then there are unique morphisms c_i with $Uc_i = d_i$ and, moreover, c_i is a colimiting cocone in $\mathsf{Coalg}(T)$.

The situation for limits is more complicated. Concerning monos the situation is the following. U preserves and reflects monos if T preserves weak pullbacks. In particular, if $T : \mathsf{Set} \to \mathsf{Set}$ preserves weak pullbacks, then a coalgebra morphism is mono iff it is injective.

Limits can be obtained from the observation that, for coalgebras over set, if (X_i, ξ_i) are subcoalgebras of (X, ξ), then the union $\bigcup X_i$ is a subcoalgebra of (X, ξ). The following exercise treats a special case.

Exercise 5.2. Show that the equaliser of $f, g : A \to B$ in $\mathsf{Coalg}(T)$, $T : \mathsf{Set} \to \mathsf{Set}$, is given by the largest subcoalgebra contained in $\{x \in A \mid f(x) = g(x)\}$.

This can be generalised to other limits and also to other base categories, see eg [33].

Behavioural Equivalence

The following exercise gives a formulation of behavioural equivalence in terms of an operator $\tilde{T}$, resembling the definitions of bisimulation via relators and relation lifting in Section 2.2.

Exercise 5.3 (pre-congruence [3]). Given a relation R on X, define $\tilde{T}R$ as the kernel of Tq where q is the quotient $q : X \to X/R$ of X wrt (the equivalence generated by) R. Call R a *pre-congruence* on a coalgebra (X, ξ) if

$$R \subseteq \xi^{-1}(\tilde{T}(R)).$$

Show that R is pre-congruence on (X, ξ) iff there is (a necessarily unique) ξ_R such that the quotient $q : X \to X/R$ is a coalgebra-morphism $(X, \xi) \to (X/R, \xi_R)$.

Preservation of Weak Pullbacks

The next exercise gives an example of a functor that does not preserve weak pullbacks. A related functor, whose coalgebras are the monotone neighbourhood frames (Example 1.4), has the same property as shown in Hansen and Kupke [23]. A third example, from Aczel and Mendler [3], is given by the functor that maps a set X to $\{(x, y, z) \mid \text{card}(\{x, y, z\}) \leq 2\}$ (it is not difficult to see that the cardinality restriction prohibits the map that should exist into the image of the weak pullback).

Exercise 5.4 (2^{2^-} does not preserve weak pullbacks [45]). Show that the hypersystems functor $T = 2^{2^-}$ does not preserve weak pullbacks. [Hint: The pushout of the constant maps $\text{zero}, \text{one} : 2 \to 2$ is given by the empty set 0. Apply T to this pushout diagram and let P be the pushout of $T(\text{zero})$ and $T(\text{one})$. One has to show that the canonical map $T0 \to P$ is not surjective for which a sufficient condition is that the cardinality of P is larger than 2.]

Modal Logic

The following exercise gives the traditional definition of a BAO and shows that BAOs are algebras for the functor $\mathcal{V}$ (Example 4.13).

Exercise 5.5 (BAOs). A Boolean algebra with operator (BAO) is a Boolean algebra A with a finite preserving operator $l : A \to A$. A BAO-morphism is a Boolean algebra morphism $f : A \to A'$ such that $l' \circ f = f \circ l$. Show that the the category of BAOs is isomorphic to the category $\mathsf{Alg}(\mathcal{V})$ where $\mathcal{V}$ is as in Example 4.13.

References

[1] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science Volume 3*, pages 1–168. Oxford University Press, 1994.

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

[3] P. Aczel and N. P. Mendler. A final coalgebra theorem. In D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigné, editors, *Category Theory and Computer Science*, volume 389 of *LNCS*, pages 357–365. Springer, 1989.

[4] J. Adámek, H. Herrlich, and G. E. Strecker. *Abstract and Concrete Categories.* John Wiley & Sons, 1990.

[5] J. Adámek and V. Koubek. On the greatest fixed point of a set functor. *Theoretical Computer Science*, 150(1):57–75, 1995.

[6] J. Adámek and H.-E. Porst. On varieties and covarieties in a category. *Mathematical Structures in Computer Science.* To appear.

[7] J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Notes Series*. Cambridge University Press, 1994.

[8] J. Adámek and V. Trnková. *Automata and Algebras in Categories*. Kluwer Academic Publishers, 1990.

[9] 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.

[10] A. Baltag. A logic for coalgebraic simulation. In H. Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *ENTCS*, pages 41–60, 2000.

[11] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114(2):299–315, June 1993.

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

[13] F. Borceux. *Handbook of Categorical Algebra*. Cambridge University Press, 1994.

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

[15] A. Carboni, G. Kelly, and R. Wood. A 2-categorical approach to change of base and geometric morphisms I. Technical Report 90-1, Department of Pure Mathematics, University of Sydney, 1990.

[16] C. Cîrstea. On logics for coalgebraic simulation. In J. Adamek, editor, *Coalgebraic Methods in Computer Science (CMCS'04)*, ENTCS, 2004.

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

[18] R. Davis. Quasi-cotripleable categories. *Proceedings of the American Mathematical Society*, 35:43–38, 1972.

[19] R. Goldblatt. What is the coalgebraic analogue of Birkhoff's variety theorem? *Theoretical Computer Science*, 266:853–886, 2001.

[20] H. P. Gumm. Equational and implicational classes of co-algebras. Extended abstract. *RelMiCS'4. The 4th International Seminar on Relational Methods in Logic, Algebra and Computer Science, Warsaw*, 1998.

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

[22] H. P. Gumm and T. Schröder. Coalgebraic structure from weak limit preserving functors. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *ENTCS*, pages 113–134, 2000.

[23] H. H. Hansen and C. Kupke. A coalgebraic perspective on monotone modal logic. In J. Adamek, editor, *Coalgebraic Methods in Computer Science (CMCS'04)*, ENTCS, 2004.

[24] U. Hensel and B. Jacobs. Proof principles for iterated datatypes. In E. Moggi and G. Rosolini, editors, *Category Theory and Computer Science*, volume 1290 of *LNCS*, pages 220–241, 1997.

[25] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145(2):107–152, 1998.

[26] J. Hughes and B. Jacobs. Simulations in coalgebra. *Theoretical Computer Science*. Accepted for publication.

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

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

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

[30] B. Klin and P. Sobocinski. Syntactic formats for free: An abstract approach to process equivalence. In *Proceedings of CONCUR 2003*, volume 2671 of *LNCS*, 2003.

[31] C. Kupke, A. Kurz, and D. Pattinson. Algebraic semantics for coalgebraic logics. In J. Adamek, editor, *Coalgebraic Methods in Computer Science (CMCS'04)*, ENTCS, 2004.

[32] A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.

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

[34] A. Kurz and D. Pattinson. Coalgebraic modal logic of finite rank. *Mathematical Structures in Computer Science*. To appear.

[35] A. Kurz and J. Rosický. Operations and equations for coalgebras. *Mathematical Structures in Computer Science*. To appear.

[36] F. Linton. An outline of functorial semantics. In B. Eckmann, editor, *Seminar on triples and categorical homology theory*, volume 80 of *Lecture Notes in Mathematics*, pages 7–52. Springer, 1969.

[37] S. Mac Lane. *Category Theory for the Working Mathematician.* Springer, 1971.

[38] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Comput. Sci.*, 309(1–3):177–193, 2003.

[39] G. Plotkin. Bialgebraic semantics and recursion (extended abstract). In A. Corradini, M. Lenisa, and U. Montanari, editors, *Coalgebraic Methods in Computer Science (CMCS'01)*, 2001.

[40] J. Reiterman. Algebraic theories and varieties of functor algebras. *Fundamenta Mathematicae*, 118:59–68, 1983.

[41] G. Roşu. Equational axiomatizability for coalgebra. *Theoretical Computer Science*, 260:229–247, 2001.

[42] J. Rosický. On algebraic categories. In *Universal Algebra (Proc. Coll. Esztergom 1977)*, volume 29 of *Colloq. Math. Soc. J. Bolyai*, pages 662–690, 1981.

[43] J. Rutten. Automata and coinduction - an exercise in coalgebra. In D. Sangiorigi and R. de Simone, editors, *Proceedings of CONCUR '98*, volume 1466 of *LNCS*. Springer, 1998.

[44] J. Rutten. Behavioural differential equations: A coinductive calculus of streams, automata, and power series. Report SEN-R0023, CWI, Amsterdam, 2000.

[45] J. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[46] H. Tews. Coalgebras for binary methods: Properties of bisimulations and invariants. *Theoretical Informatics and Applications*, 35:83–111, 2001.

[47] H. Tews. *Coalgebraic Methods for Object-Oriented Specification.* PhD thesis, University of Dresden, 2002.

[48] H. Tews. Predicate and relation lifting for parametric algebraic specifications. In J. Adamek, editor, *Coalgebraic Methods in Computer Science (CMCS'04)*, ENTCS, 2004.

[49] A. Thijs. *Simulation and Fixpoint Semantics.* PhD thesis, Rijksuniversiteit Groningen, 1996.

[50] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. of the LICS'97*, pages 280–305, 1997.

[51] S. J. Vickers. *Topology Via Logic.* Cambridge University Press, 1989.

[52] J. Worrell. Terminal sequences for accessible endofunctors. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'99)*, volume 19 of *ENTCS*, pages 39–53, 1999.

[53] J. Worrell. *On Coalgebras and Final Semantics*. PhD thesis, Oxford University Computing Laboratory, 2000.