Electronic Notes in Theoretical Computer Science 82 No. 1 (2003)
URL: http://www.elsevier.nl/locate/entcs/volume82.html 18 pages

On Expressivity and Compositionality in Logics for Coalgebras

Corina Cîrstea [1,2]

Computing Laboratory
University of Oxford
Oxford, UK

Abstract

This paper attempts to unify some of the existing approaches to defining modal logics for coalgebras, from the point of view of constructing the languages employed by these logics. An abstract framework for defining languages for coalgebras from so-called language constructors, corresponding to one-step unfoldings of the coalgebraic structure, is introduced, and a method for deriving expressive languages for coalgebras from suitable choices for the language constructors is described. Moreover, it is shown that the derivation of such languages by means of language constructors is well-behaved w.r.t. various forms of composition between coalgebraic types.

1 Introduction

Existing modal logics for coalgebras can be classified into three categories, depending on the types of coalgebraic structures they refer to, as well as on the degree of abstraction of the modal operators they employ. The first category consists of logics which are generic in the types of coalgebraic structures they are able to capture, and whose associated languages are derived directly from the coalgebraic types under consideration [7,2]. While both natural and expressive, these logics employ modal operators of an abstract nature, and as a result are difficult to use for actual specification. Moreover, these logics lack compositionality as far as the languages they employ are concerned, in that the languages induced by functor compositions are not directly derivable from the languages induced by the functors being composed. The second category of logics concerns inductively-defined classes of coalgebraic structures [9,5]. The specific nature of the types considered here is reflected in the associated languages, which employ concrete modal operators derived from the inductive

[1] Research supported by St. John's College, Oxford
[2] Email: corina@comlab.ox.ac.uk

definitions of the underlying types. While restrictive from the point of view of the coalgebraic types they cover, these logics are intrinsically compositional as far as the definition of the corresponding languages and of their semantics is concerned. Finally, the third category of logics aims to combine some of the benefits of the previous two categories, by providing reasonably concrete languages for arbitrarily general coalgebraic structures [8]. However, this is achieved at the expense of losing the naturality of the logics: rather than being determined by the coalgebraic types under consideration, the languages employed by such logics are based on (semantically-defined) modal operators which have to be provided explicitly. Thus, the structure of the underlying types is not, in general, reflected in the resulting languages. Furthermore, additional constraints on the collection of modal operators are necessary to guarantee that the resulting logics are expressive, and these constraints are not well-behaved w.r.t. type composition – it is not, in general, possible to derive expressive logics for compositions of coalgebraic types from expressive logics for the types being composed. This is simply because the class of endofunctors for which expressive logics of this kind exist is not closed under composition. An example in this sense is provided by (coalgebras of) the functor $\mathcal{P} \circ \mathcal{P}$, with $\mathcal{P}$ denoting the powerset functor – while expressive logics exist for both $\mathcal{P}$ and $\mathcal{P} \circ \mathcal{P}$ (see e.g. [7] or [9]), an expressive logic of the kind considered in [8] exists for $\mathcal{P}$, but not for $\mathcal{P} \circ \mathcal{P}$.

The aim of this work is to investigate the existence of generic logics for coalgebras, which are both expressive and compositional w.r.t. the underlying types. Our approach is based on a generalisation of the technique used in [9] to derive languages for inductively-defined endofunctors, to arbitrary endofunctors. We use an abstract notion of language constructor, corresponding to a one-step unfolding of the coalgebraic structure, to capture one inductive step in the definition of a language for coalgebras; and through repeated applications of language constructors (to a propositional language to begin with), we derive languages able to formalise properties involving arbitrary unfoldings of the coalgebraic structure. The definition of these languages resembles the approach in [8], in that it uses transfinite induction along the final sequence of an endofunctor. If the language constructor underlying such a definition preserves expressivity (in a sense made precise in what follows), an expressive language for coalgebras is eventually obtained. Furthermore, combining expressivity preserving language constructors for different coalgebraic types yields expressive languages for (coalgebras of) various forms of composition between those types. All the previously-mentioned approaches to defining modal logics for coalgebras are covered by the resulting approach.

The paper is structured as follows. Section 2 recalls some coalgebraic concepts which are used in subsequent sections, and at the same time outlines two existing approaches to defining modal logics for coalgebras. Section 3 introduces the notion of *language constructor* for an endofunctor, and shows how instances of this notion are retrieved in existing modal logics for coalgebras.

Section 4 defines *languages for coalgebras of endofunctors*, and uses transfinite induction to derive such languages from language constructors. Suitable choices for the language constructors are shown to yield expressive languages for coalgebras. Section 5 shows how to derive expressive logics for various forms of functor composition from expressive logics for the functors being composed. Finally, Section 6 outlines some possible directions for future work.

2 Preliminaries

The setting we shall be working in is that of coalgebras of endofunctors on Set. Given such an endofunctor $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, a T-*coalgebra* is given by a pair $\langle C, \gamma \rangle$ with C a set (the *carrier* of the coalgebra) and $\gamma : C \to \mathsf{T}C$ a function (the *coalgebra map*). Also, a T-*coalgebra homomorphism* between T-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ is given by a function $f : C \to D$ additionally satisfying $\mathsf{T}f \circ \gamma = \delta \circ f$. The category of T-coalgebras and T-coalgebra homomorphisms is denoted $\mathsf{Coalg}(\mathsf{T})$.

Given T-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$, two states $c \in C$ and $d \in D$ are called T-*behaviourally equivalent* if there exist a T-coalgebra $\langle E, \eta \rangle$ and T-coalgebra homomorphisms $f : \langle C, \gamma \rangle \to \langle E, \eta \rangle$ and $g : \langle D, \delta \rangle \to \langle E, \eta \rangle$ with $f(c) = g(d)$. In the presence of a final T-coalgebra, T-behavioural equivalence is given by equality under the unique homomorphisms into the final coalgebra (see e.g. [8, Theorem 3.4]).

A T-*bisimulation* between T-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ is a relation $\langle R, \pi_1, \pi_2 \rangle$ on $C \times D$, with R carrying a (not necessarily unique) T-coalgebra structure $\rho : R \to \mathsf{T}R$ that makes $\pi_1 : R \to C$ and $\pi_2 : R \to D$ T-coalgebra homomorphisms. The largest T-bisimulation between $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ (obtained as the union of all such T-bisimulations) is called T-*bisimilarity*.

If two states are T-bisimilar, then they are also T-behaviourally equivalent. And if, in addition, T preserves *weak pullbacks* [3], then the converse is also true.

The class of weak pullback preserving endofunctors is sufficiently general to account for most known examples of coalgebraic types. And although preservation of weak pullbacks is not required by our approach, the fact that a given endofunctor preserves weak pullbacks is an advantage, in that the logics we obtain in this case are expressive not only w.r.t. behavioural equivalence but also w.r.t. bisimilarity.

Preservation of weak pullbacks will, however, be required by one particular instance of our approach. The next observation will prove useful in that case.

Remark 2.1 Weak pullback preserving endofunctors also preserve weak limits of W-shaped diagrams. This follows from weak limits for such diagrams being obtained from weak pullbacks for their left and right (V-shaped) subdiagrams, by subsequently constructing another weak pullback.

[3] Weak pullbacks are defined similarly to standard pullbacks, except that the mediating arrows are not required to be unique.

Bisimilarity and behavioural equivalence are two slightly different ways of capturing the observational indistinguishability of states. Additional observational equivalence relations between the states of coalgebras can be defined via the final sequence of the endofunctor in question.

Definition 2.2 ([10]) Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$. The **final sequence of** T is an ordinal-indexed sequence of sets (Z_α) together with a family $(p^\alpha_\beta)_{\beta \leq \alpha}$ of functions $p^\alpha_\beta : Z_\alpha \to Z_\beta$, satisfying:

- $Z_{\alpha+1} = \mathsf{T} Z_\alpha$
- $p^{\alpha+1}_{\beta+1} = \mathsf{T} p^\alpha_\beta$ for $\beta \leq \alpha$
- $p^\alpha_\alpha = 1_{Z_\alpha}$
- $p^\alpha_\gamma = p^\beta_\gamma \circ p^\alpha_\beta$ for $\gamma \leq \beta \leq \alpha$
- if α is a limit ordinal, the cone $Z_\alpha, (p^\alpha_\beta)_{\beta<\alpha}$ for $(p^\beta_\gamma)_{\gamma \leq \beta < \alpha}$ is limiting.

The final sequence of T is uniquely defined by the above conditions. In particular, $Z_0 = 1$, with $1 = \{0\}$ denoting a final object in Set.

Remark 2.3 The final sequence of T can be used to construct a final T-coalgebra. Specifically, if the final sequence of T *stabilises at* α (that is, if $p^{\alpha+1}_\alpha$ is an isomorphism), then Z_α is the carrier of a final T-coalgebra (see [3, Theorem 1.3], or [1, Theorem 5]). Various constraints on T can be used to ensure that its final sequence stabilises at a specific α. In particular, if T is ω^{op}-continuous, its final sequence stabilises at ω. Also, if T is κ-accessible, with κ a regular cardinal, its final sequence stabilises at $\kappa{\cdot}2$ (see [10, Theorem 10]).

Remark 2.4 The elements of the final sequence provide approximations of notions of observable behaviour. Given a T-coalgebra $\langle C, \gamma \rangle$, one can define an ordinal-indexed sequence of functions (γ_α), with $\gamma_\alpha : C \to Z_\alpha$, as follows:

- $\gamma_\alpha = \mathsf{T}\gamma_\beta \circ \gamma$, if $\alpha = \beta + 1$;
- γ_α is the unique function satisfying $p^\alpha_\beta \circ \gamma_\alpha = \gamma_\beta$ for each $\beta < \alpha$, if α is a limit ordinal.

The functions γ_α take states of the coalgebra to their partial observable behaviours, as defined by Z_α.

Remark 2.5 A notion of observational equivalence between states of coalgebras can then be defined as equality of certain partial observable behaviours. Specifically, if α is a regular cardinal and $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ are T-coalgebras, then two states $c \in C$ and $d \in D$ are called *α-observationally equivalent* if $\gamma_\alpha(c) = \delta_\alpha(d)$. Taking $\alpha = \omega$ yields a notion of observational equivalence which only takes into account the finitary behaviour of states [6]. Also, if T is κ-accessible, taking $\alpha = \kappa$ yields a notion of observational equivalence which is the same as behavioural equivalence (see [8, Theorem 3.4]).

We now recall two existing approaches to deriving modal logics for coalgebras of endofunctors on Set.

Definition 2.6 ([7]) Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ denote a κ-accessible, weak pullback preserving endofunctor. The **language $\mathcal{L}_{\mathsf{T}}$ of (T-)coalgebraic logic** is the carrier of the initial algebra of the functor $X \mapsto \mathcal{P}X + \mathsf{T}X$. We write $\bigwedge : \mathcal{P}\mathcal{L}_{\mathsf{T}} \to \mathcal{L}_{\mathsf{T}}$ and respectively $\nabla : \mathsf{T}\mathcal{L}_{\mathsf{T}} \to \mathcal{L}_{\mathsf{T}}$ for the two coproduct injections arising from the definition of $\mathcal{L}_{\mathsf{T}}$.

Given a T-coalgebra $\langle C, \gamma\rangle$, the **satisfaction relation** $\models$ between elements of C and formulae of $\mathcal{L}_{\mathsf{T}}$ is defined inductively as follows:

- $c \models \bigwedge \Phi$ iff $c \models \varphi$ for all $\varphi \in \Phi$
- $c \models \nabla\psi$ iff $\gamma(c)\,(\mathsf{T}{\models})\,\psi$ [4]

for $c \in C$, $\Phi \in \mathcal{P}\mathcal{L}_{\mathsf{T}}$ and $\psi \in \mathsf{T}\mathcal{L}_{\mathsf{T}}$.

The language of coalgebraic logic is sufficiently expressive to characterise the elements of final coalgebras, but at the same time sufficiently weak not to distinguish between bisimilar states (see [7], and also [8, Section 5] for an alternative proof of this statement).

Definition 2.7 ([8]) Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$. A **predicate lifting for** T is a natural transformation $\lambda : \hat{\mathcal{P}} \Rightarrow \hat{\mathcal{P}} \circ \mathsf{T}$ (with $\hat{\mathcal{P}} : \mathsf{Set} \to \mathsf{Set}$ denoting the contravariant powerset functor).

Now let Λ denote a set of predicate liftings for T, let σ denote a regular cardinal, and let $\mathcal{P}_\sigma : \mathsf{Set} \to \mathsf{Set}$ denote the functor taking a set X to the set of subsets of X of cardinality smaller than σ. The **(modal) language $\mathcal{L}_\sigma(\Lambda)$** is defined inductively by:

$$\varphi ::= \bigwedge \Phi \mid \neg\varphi \mid [\lambda]\varphi\,, \quad \Phi \in \mathcal{P}_\sigma(\mathcal{L}_\sigma(\Lambda))\,, \quad \varphi \in \mathcal{L}_\sigma(\Lambda)\,, \quad \lambda \in \Lambda$$

In addition, one defines $\bigvee \Phi ::= \neg \bigwedge_{\varphi\in\Phi} \neg\varphi$ for $\Phi \in \mathcal{P}_\sigma(\mathcal{L}_\sigma(\Lambda))$, and $\langle\lambda\rangle\varphi ::= \neg[\lambda]\neg\varphi$ for $\lambda \in \Lambda$ and $\varphi \in \mathcal{L}_\sigma(\Lambda)$.

Given a T-coalgebra $\langle C, \gamma\rangle$, the **satisfaction relation** $\models$ between elements of C and formulae of $\mathcal{L}_\sigma(\Lambda)$ is defined by structural induction on formulae:

- $c \models \bigwedge \Phi$ iff $c \models \varphi$ for all $\varphi \in \Phi$
- $c \models \neg\varphi$ iff $c \not\models \varphi$
- $c \models [\lambda]\varphi$ iff $\gamma_C(c) \in \lambda_C([\![\varphi]\!]_\gamma)$, with $[\![\varphi]\!]_\gamma$ being given by $\{c \in C \mid c \models \varphi\}$.

A set of predicate liftings Λ is said to be **separating** if, for any set X, the map $t \in \mathsf{T}X \mapsto \{\, \lambda_X(Y) \mid \lambda \in \Lambda, Y \in \mathcal{P}X, \lambda_X(Y) \ni t \,\}$ is monic.

It is shown in [8] that the language $\mathcal{L}_\sigma(\Lambda)$ is *adequate* (i.e. behavioural equivalence implies logical equivalence); and if, in addition, T is κ-accessible and Λ is separating, then there exists a cardinal σ, depending only on κ and on $\mathsf{card}(\Lambda)$, such that $\mathcal{L}_\sigma(\Lambda)$ is also *expressive* (i.e. logical equivalence implies behavioural equivalence).

[4] As noted in [7], $\mathsf{T}{\models}$ induces a relation on $\mathsf{T}C \times \mathsf{T}\mathcal{L}_{\mathsf{T}}$, defined via $\mathsf{T}\pi_1$ and $\mathsf{T}\pi_2$ (with $\pi_1 : {\models} \to C$ and $\pi_2 : {\models} \to \mathcal{L}_{\mathsf{T}}$ defining the relation $\models$). Given $t \in \mathsf{T}C$ and $\psi \in \mathsf{T}\mathcal{L}_{\mathsf{T}}$, we write $t\,(\mathsf{T}{\models})\,\psi$ if there exists $w \in \mathsf{T}{\models}$ such that $(\mathsf{T}\pi_1)(w) = t$ and $(\mathsf{T}\pi_2)(w) = \psi$.

3 Language Constructors

We now fix a regular cardinal σ. In what follows, we shall consider languages which are closed under conjunctions of cardinality smaller than σ, as well as under negation. Such languages will be regarded as algebras of the functor $\mathsf{B}_\sigma = \mathcal{P}_\sigma + \mathsf{Id} : \mathsf{Set} \to \mathsf{Set}$, with the two components of the algebra maps taking sets of formulae Φ of cardinality smaller than σ to their conjunction $\bigwedge \Phi$, and respectively single formulae φ to their negation $\neg\varphi$. The free B_σ-algebra over a set A will be denoted $A^{\wedge,\neg}$, while the unique extension of a function $f : A \to B$ to a B_σ-algebra homomorphism will be denoted $f^{\wedge,\neg} : A^{\wedge,\neg} \to B^{\wedge,\neg}$. Also, given a function $g : A \to C$, with A a set and C (the carrier of) a B_σ-algebra, the B_σ-algebra homomorphism arising from the freeness of $A^{\wedge,\neg}$ will be denoted $g^\# : A^{\wedge,\neg} \to C$.

We now define languages whose formulae are interpreted over given sets.

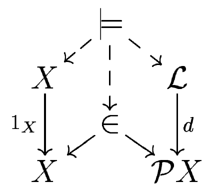
Definition 3.1 Let X be a set. An **X-language** is a pair $\langle \mathcal{L}, d \rangle$ with $\mathcal{L}$ a set carrying B_σ-structure and $d : \mathcal{L} \to \mathcal{P}X$ a function which preserves the B_σ-structure[5]. A **map** between X-languages $\langle \mathcal{L}, d \rangle$ and $\langle \mathcal{L}', d' \rangle$ is a function $l : \mathcal{L} \to \mathcal{L}'$, itself preserving the B_σ-structure, such that $d' \circ l = d$.

In particular, a 1-language is given by a B_σ-algebra $\mathcal{L}$ together with a B_σ-algebra homomorphism $d : \mathcal{L} \to \mathcal{P}1$.

Remark 3.2 An X-language $\langle \mathcal{L}, d \rangle$ induces a *satisfaction relation* $\models \subseteq X \times \mathcal{L}$ given by:

$$x \models \varphi \;\text{ iff }\; x \in d(\varphi) \ , \quad \text{for } x \in X \text{ and } \varphi \in \mathcal{L}$$

Equivalently, the cone defined by $\models$ over the diagram defined by 1_X, d and the two projections defining the membership relation is (weakly) limiting:



A map between X-languages defines a B_σ-structure preserving as well as denotation preserving translation between the given languages. The category of X-languages and maps between them is denoted X-Lang.

Proposition 3.3 *X-Lang has colimits.*

Proof (Sketch). The $\mathcal{L}$-component of the colimit in X-Lang of a diagram $\mathcal{D}$ is given by the colimit in $\mathsf{Alg}(\mathsf{B}_\sigma)$ of the diagram relating the $\mathcal{L}$-components of languages in $\mathcal{D}$. The d-component of the colimit is obtained by exploiting the couniversality of the $\mathcal{L}$-component. □

[5] The set $\mathcal{P}X$ can be naturally endowed with B_σ-algebra structure, namely by interpreting $\bigwedge$ as intersection and $\neg$ as complement.

In particular, the $\mathcal{L}$-component of an initial object in X-Lang contains $\top$ (defined as $\bigwedge \emptyset$) and $\bot$ (defined as $\neg\top$), for any set X.

The mapping $X \mapsto X$-Lang can be extended to a contravariant functor $\mathsf{L} : \mathsf{Set} \to \mathsf{Cat}$ by letting, for $f : X' \to X$, $\mathsf{L}(f) : X$-$\mathsf{Lang} \to X'$-Lang be given by $\mathsf{L}(f)(\langle \mathcal{L}, d \rangle) = \langle \mathcal{L}, \hat{\mathcal{P}} f \circ d \rangle$ [6] and $\mathsf{L}(f)(l) = l$ for $l : \langle \mathcal{L}, d \rangle \to \langle \mathcal{L}', d' \rangle$.

More generally, relationships between languages for different sets can be captured using the notion of *cofibration* [7]. Let Lang denote the category whose objects are given by pairs $\langle X, \langle \mathcal{L}, d \rangle \rangle$ with X a set and $\langle \mathcal{L}, d \rangle \in |X\text{-}\mathsf{Lang}|$, and whose arrows from $\langle X, \langle \mathcal{L}, d \rangle \rangle$ to $\langle X', \langle \mathcal{L}', d' \rangle \rangle$ are given by pairs $\langle f, l \rangle$ with $f : X' \to X$ a function and $l : \mathcal{L} \to \mathcal{L}'$ a function preserving the B_σ-structure, such that $\hat{\mathcal{P}} f \circ d = d' \circ l$:

$$\begin{array}{ccc} \mathcal{L} & \xrightarrow{\;l\;} & \mathcal{L}' \\ {\scriptstyle d}\downarrow & & \downarrow{\scriptstyle d'} \\ \mathcal{P}X & \xrightarrow[\hat{\mathcal{P}} f]{} & \mathcal{P}X' \end{array}$$

Also, let $\mathsf{E} : \mathsf{Lang} \to \mathsf{Set}^{\mathrm{op}}$ denote the functor taking $\langle X, \langle \mathcal{L}, d \rangle \rangle$ to X and $\langle f, l \rangle : \langle X, \langle \mathcal{L}, d \rangle \rangle \to \langle X', \langle \mathcal{L}', d' \rangle \rangle$ to f^{op}.

Proposition 3.4 E *is a cofibration.*

Proof (Sketch). The coreindexing functor $f_* : X$-$\mathsf{Lang} \to X'$-Lang induced by a function $f : X' \to X$ takes $\langle X, \langle \mathcal{L}, d \rangle \rangle$ to $\langle X', \langle \mathcal{L}, \hat{\mathcal{P}} f \circ d \rangle \rangle$. □

Proposition 3.5 Lang *has colimits.*

Proof (Sketch). Colimits in Lang are constructed from limits in Set and colimits in the corresponding cofibres (see Proposition 3.3). The fact that the coreindexing functors preserve colimits is also used. □

Definition 3.6 An X-language $\langle \mathcal{L}, d \rangle$ is called **expressive** if there exists a function $i : X \to \mathcal{L}$ such that $d \circ i = \{\_\}_X$:

$$\begin{array}{ccc} & & \mathcal{L} \\ & {\scriptstyle i}\nearrow & \downarrow{\scriptstyle d} \\ X & \xrightarrow[\{\_\}_X]{} & \mathcal{P}X \end{array}$$

with the natural transformation $\{\_\} : \mathsf{Id} \Rightarrow \mathcal{P}$ being given by $\{\_\}_S(s) = \{s\}$ for $s \in S$ and $S \in |\mathsf{Set}|$.

Remark 3.7 Since $\{\_\}_X$ is injective, any function i satisfying the condition in Definition 3.6 is itself injective. Also, if $\langle \mathcal{L}, d \rangle$ is expressive and $x, y \in X$ are logically equivalent, then $x = y$. For, in this case, $y \models i(x)$, and hence $y \in \{x\}$.

The notion of language constructor which we now introduce aims to capture *one inductive step* in the definition of languages for T-coalgebras.

[6] Note that, for $f : X' \to X$, the function $\hat{\mathcal{P}} f : \mathcal{P}X \to \mathcal{P}X'$ preserves the B_σ-structure.
[7] See [4] for an introduction to the theory of fibrations.

Definition 3.8 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$. A **language constructor for** T is a fibred functor[8] $\mathcal{F} : \mathsf{Lang} \to \mathsf{Lang}$ over T^{op}. Thus, $\mathsf{E} \circ \mathcal{F} = \mathsf{T}^{\mathrm{op}} \circ \mathsf{E}$:

$$\begin{array}{ccc} \mathsf{Lang} & \xrightarrow{\mathcal{F}} & \mathsf{Lang} \\ {\scriptstyle \mathsf{E}}\downarrow & & \downarrow{\scriptstyle \mathsf{E}} \\ \mathsf{Set}^{\mathrm{op}} & \xrightarrow[\mathsf{T}^{\mathrm{op}}]{} & \mathsf{Set}^{\mathrm{op}} \end{array}$$

That is, language constructors for T take X-languages to $\mathsf{T}X$-languages. Furthermore, language constructors preserve relationships between languages, as captured by arrows in Lang.

The existence of coproducts in the categories X-Lang with $X \in |\mathsf{Set}|$ (see Proposition 3.3) and Lang (see Proposition 3.5) makes it possible to define a *join operator* $\biguplus$ on language constructors.

Definition 3.9 Let $(\mathcal{F}_i)_{i\in I}$ denote a family of language constructors for T. Then, the language constructor $\biguplus_{i\in I} \mathcal{F}_i : \mathsf{Lang} \to \mathsf{Lang}$ for T takes an X-language $\langle \mathcal{L}, d\rangle$ to the $\mathsf{T}X$-language $\coprod_{i\in I} \mathcal{F}_i\langle \mathcal{L}, d\rangle$[9]. The action of $\biguplus_{i\in I} \mathcal{F}_i$ on arrows in Lang is determined by the couniversality of coproducts in Lang.

One way of defining a language constructor for T is to consider, for an X-language $\mathcal{L}$, the least $\mathsf{T}X$-language containing $\mathsf{T}\mathcal{L}$. This language constructor mirrors the construction of the language of coalgebraic logic, as given in Definition 2.6.

Example 3.10 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ denote a κ-accessible, weak pullback preserving endofunctor. A language constructor $\mathcal{F}_{\mathsf{T}}$ for T is given by the functor taking $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to $\langle \mathsf{T}X, \langle \mathcal{L}', d'\rangle\rangle$, with $\mathcal{L}' = (\mathsf{T}\mathcal{L})^{\wedge,\neg}$ and $d' = (\epsilon_X \circ \mathsf{T}d)^{\#}$:

$$\begin{array}{c} \mathcal{L} \\ \downarrow{\scriptstyle d} \\ \mathcal{P}X \end{array} \qquad \begin{array}{ccc} \mathsf{T}\mathcal{L} & & \\ \downarrow{\scriptstyle \mathsf{T}d} & & \\ \mathsf{T}\mathcal{P}X & \xrightarrow{\epsilon_X} & \mathcal{P}\mathsf{T}X \end{array} \qquad \begin{array}{c} \mathcal{L}' \\ \downarrow{\scriptstyle d'} \\ \mathcal{P}\mathsf{T}X \end{array}$$

where the natural transformation $\epsilon : \mathsf{T} \circ \hat{\mathcal{P}} \Rightarrow \hat{\mathcal{P}} \circ \mathsf{T}$ is given by:

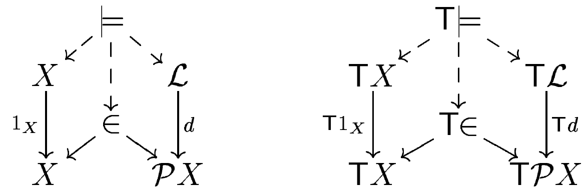
$$(1) \qquad \epsilon_X(Y) = \{\, t \in \mathsf{T}X \mid t\ (\mathsf{T}{\in})\ Y \,\} \text{ for } X \in |\mathsf{Set}| \text{ and } Y \in \mathsf{T}\mathcal{P}X.$$

With the above definition, the satisfaction relation $\models \subseteq \mathsf{T}X \times \mathcal{L}'$ induced by $\langle \mathcal{L}', d'\rangle$ (see Remark 3.2) coincides with $(\mathsf{T}{\models})^{\wedge,\neg}$, where $\models \subseteq X \times \mathcal{L}$ denotes the satisfaction relation induced by $\langle \mathcal{L}, d\rangle$, and where $(\mathsf{T}{\models})^{\wedge,\neg}$ denotes the natural extension of $\mathsf{T}{\models}$ to formulae containing conjunctions and negations. To see this, note that the preservation by T of weak limits of W-shaped diagrams (see Remark 2.1) results in the cone defined by $\mathsf{T}{\models}$ over the diagram defined by $\mathsf{T}1_X$, $\mathsf{T}d$ and the images under T of the two projections defining

[8] See [4] for a definition of fibred functors.

[9] This coproduct is constructed in $\mathsf{T}X$-Lang.

the membership relation being weakly limiting:

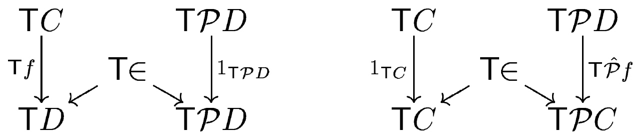


Hence, $t\ (\mathsf{T}{\models})\ \psi$ is equivalent to $t\ (\mathsf{T}{\in})\ (\mathsf{T}d)(\psi)$, which, in turn, is equivalent to $t \in d'(\psi)$, for any $t \in \mathsf{T}X$ and any $\psi \in \mathsf{T}\mathcal{L}$. The particular definition of the denotation map d' was driven precisely by the need to ensure that the satisfaction relations induced by d and d' are related as above. As a result, $\mathcal{F}_{\mathsf{T}}$ captures one step in the definition of the language used in [7], the only difference being that here negation is also present.

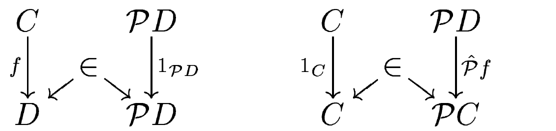
Before defining the action of $\mathcal{F}_{\mathsf{T}}$ on arrows in Lang, we need to verify that ϵ as defined by (1) is, indeed, natural. For this, let $f : C \to D \in \| \mathsf{Set} \|$. Then, the naturality of ϵ w.r.t. f reduces to:

$$(\mathsf{T}f)(t)\ (\mathsf{T}{\in})\ Y \ \text{ iff } \ t\ (\mathsf{T}{\in})\ (\mathsf{T}\hat{\mathcal{P}}f)(Y)$$

for any $t \in \mathsf{T}C$ and any $Y \in \mathsf{T}\mathcal{P}D$. This, in turn, follows from the limiting cones of the following diagrams defining the same relation on $\mathsf{T}C \times \mathsf{T}\mathcal{P}D$:

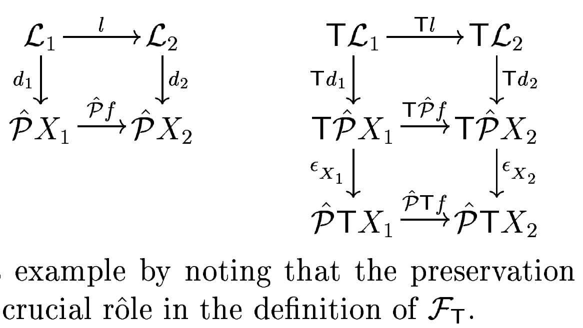


The previous statement follows e.g. from the existence of weakly limiting cones for the two diagrams, cones which, in addition, coincide on the arrows into $\mathsf{T}C$ and $\mathsf{T}\mathcal{P}D$ respectively. The last statement is a consequence of the existence of limiting cones with a similar property for the following two diagrams:



and of the fact that T takes limits of W-shaped diagrams to weak limits of the images under T of those diagrams (see Remark 2.1).

We can now define the action of $\mathcal{F}_{\mathsf{T}}$ on arrows in Lang. Specifically, an arrow $\langle f, l\rangle : \langle X_1, \langle \mathcal{L}_1, d_1\rangle\rangle \to \langle X_2, \langle \mathcal{L}_2, d_2\rangle\rangle$ is taken by $\mathcal{F}_{\mathsf{T}}$ to $\langle \mathsf{T}f, (\mathsf{T}l)^{\wedge,\neg}\rangle$: $\langle \mathsf{T}X_1, \langle (\mathsf{T}\mathcal{L}_1)^{\wedge,\neg}, (\epsilon_{X_1} \circ \mathsf{T}d_1)^{\#}\rangle\rangle \to \langle \mathsf{T}X_2, \langle (\mathsf{T}\mathcal{L}_2)^{\wedge,\neg}, (\epsilon_{X_2} \circ \mathsf{T}d_2)^{\#}\rangle\rangle$:



We conclude this example by noting that the preservation of weak pullbacks by T played a crucial rôle in the definition of $\mathcal{F}_{\mathsf{T}}$.

If some information about the structure specified by T is available, e.g. in the form of a set of predicate liftings for T, then language constructors for T can be derived based on this information.

Example 3.11 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, and let Λ denote a set of predicate liftings for T. A language constructor $\mathcal{F}_\Lambda$ for T is given by the functor taking $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to $\langle \mathsf{T}X, \langle \mathcal{L}', d'\rangle\rangle$, where $\mathcal{L}' = \{ [\lambda]\varphi \mid \lambda \in \Lambda,\ \varphi \in \mathcal{L} \}^{\wedge,\neg}$, and where $d' : \mathcal{L}' \to \mathcal{P}\mathsf{T}X$ is given by $d'([\lambda]\varphi) = \lambda_X(d(\varphi))$, $d'(\bigwedge \Phi) = \bigcap_{\varphi\in\Phi} d'(\varphi)$, and $d'(\neg\varphi) = \overline{d'(\varphi)}$ [10]. The action of the language constructor on an arrow $\langle f, l\rangle : \langle X_1, \langle \mathcal{L}_1, d_1\rangle\rangle \to \langle X_2, \langle \mathcal{L}_2, d_2\rangle\rangle$ in Lang is given by $\langle \mathsf{T}f, l'\rangle : \langle \mathsf{T}X_1, \langle \mathcal{L}_1', d_1'\rangle\rangle \to \langle \mathsf{T}X_2, \langle \mathcal{L}_2', d_2'\rangle\rangle$, with $l' : \mathcal{L}_1' \to \mathcal{L}_2'$ being given by $l'([\lambda]\varphi) = [\lambda]l(\varphi)$, $l'(\bigwedge \Phi) = \bigwedge_{\varphi\in\Phi} l'(\varphi)$, and $l'(\neg\varphi) = \neg l'(\varphi)$.

Alternatively, one can define a language constructor $\mathcal{F}_\lambda$ for T for each $\lambda \in \Lambda$ (by taking $\Lambda = \{\lambda\}$ in the above), and then define $\mathcal{F}_\Lambda$ as $\biguplus_{\lambda\in\Lambda} \mathcal{F}_\lambda$. The resulting language constructor is, up to a natural isomorphism, the same as the previously-defined one.

In the case of inductively-defined endofunctors, as considered e.g. in [9,5], language constructors can be derived from the structure of the endofunctors.

Remark 3.12 Given a set A, a language constructor $\mathcal{F}_A$ for the constant functor $X \mapsto A$ takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to $\langle A^{\wedge,\neg}, \langle A, (\{\_\}_A)^\#\rangle\rangle$. Also, a language constructor $\mathcal{F}_{\mathsf{Id}}$ for Id takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to itself. Finally, language constructors $\mathcal{F}_1 \otimes \mathcal{F}_2$, $\mathcal{F}_1 \oplus \mathcal{F}_2$, $(\mathcal{F}_1)^A$ and $\mathcal{P}\mathcal{F}_1$ for $\mathsf{F}_1 \times \mathsf{F}_2$, $\mathsf{F}_1 + \mathsf{F}_2$, $(\mathsf{F}_1)^A$ and $\mathcal{P} \circ \mathsf{F}_1$ can be derived from language constructors $\mathcal{F}_i$ for F_i, with $i = 1, 2$. Say $\mathcal{F}_i$ takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to $\langle \mathsf{F}_i X, \langle \mathcal{L}_i, d_i\rangle\rangle$. Then, $\mathcal{F}_1 \otimes \mathcal{F}_2$, $\mathcal{F}_1 \oplus \mathcal{F}_2$, $(\mathcal{F}_1)^A$ and $\mathcal{P}\mathcal{F}_1$ are defined as follows:

- $\mathcal{F}_1 \otimes \mathcal{F}_2$ takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to the coproduct of $(\pi_1)_*\langle \mathcal{L}_1, d_1\rangle$ and $(\pi_2)_*\langle \mathcal{L}_2, d_2\rangle$ in $(\mathsf{F}_1 X \times \mathsf{F}_2 X)$-$\mathsf{Lang}$ [11], with $\pi_i : \mathsf{F}_1 X \times \mathsf{F}_2 X \to \mathsf{F}_i X$ for $i = 1, 2$ denoting the product projections. We write $[\pi_i]\varphi_i$ for $\iota_i(\varphi_i)$, where $\iota_i : (\pi_i)_*\langle \mathcal{L}_i, d_i\rangle \to (\pi_1)_*\langle \mathcal{L}_1, d_1\rangle + (\pi_2)_*\langle \mathcal{L}_2, d_2\rangle$ is the ith injection, $\varphi_i \in \mathcal{L}_i$, and $i \in \{1, 2\}$.
- $\mathcal{F}_1 \oplus \mathcal{F}_2$ takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to the coproduct of $\langle (\mathcal{L}_1)^{\wedge,\neg}, ((\mathsf{e}_1)_X \circ d_1)^\#\rangle$ and $\langle (\mathcal{L}_2)^{\wedge,\neg}, ((\mathsf{e}_2)_X \circ d_2)^\#\rangle$ in $(\mathsf{F}_1 X + \mathsf{F}_2 X)$-$\mathsf{Lang}$, where, for $i \in \{1, 2\}$, the natural transformation $\mathsf{e}_i : \mathcal{P} \circ \mathsf{F}_i \Rightarrow \mathcal{P} \circ (\mathsf{F}_1 + \mathsf{F}_2)$ is given by $(e_i)_X(Y) = (\mathcal{P}\iota_i)(Y) \cup (\mathcal{P}\iota_j)(\mathsf{F}_j X)$ for $X \in |\mathsf{Set}|$ and $Y \in \mathcal{P}\mathsf{F}_i X$, with $\{i, j\} = \{1, 2\}$ [12]. We write $[\kappa_i]\varphi_i$ for $\iota_i(\varphi_i)$, where ι_i is the ith injection defining the previously-mentioned coproduct, $\varphi_i \in \mathcal{L}_i$, and $i \in \{1, 2\}$. Also, we write $\langle \kappa_i\rangle\varphi_i$ for $\neg[\kappa_i]\neg\varphi_i$, for $i \in \{1, 2\}$.

---

[10] Hence, $d'(\langle\lambda\rangle\varphi) = \overline{\lambda_X(\overline{d(\varphi)})}$, and $d'(\bigvee \Phi) = \bigcup_{\varphi\in\Phi} d'(\varphi)$.

[11] See Proposition 3.4 for the definitions of $(\pi_1)_*$ and $(\pi_2)_*$.

[12] Note that, since $\langle \mathcal{L}_1, (\mathsf{e}_1)_X \circ d_1\rangle$ and $\langle \mathcal{L}_2, (\mathsf{e}_2)_X \circ d_2\rangle$ do not qualify as languages, it is necessary to perform a closure under $\wedge$ and $\neg$ before constructing the coproduct in $(\mathsf{F}_1 X + \mathsf{F}_2 X)$-$\mathsf{Lang}$.

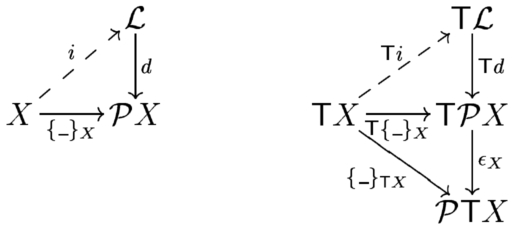
- $(\mathcal{F}_1)^A$ takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to $\langle (\mathsf{F}_1 X)^A, \coprod_{a\in A} (\pi_a)_* \langle \mathcal{L}, d\rangle\rangle$, with $\pi_a : (\mathsf{F}_1 X)^A \to \mathsf{F}_1 X$ taking $f : A \to \mathsf{F}_1 X$ to $f(a)$ for $a \in A$, and with the coproduct being constructed in $(\mathsf{F}_1 X)^A$-Lang. We write $[a]\varphi$ for $\iota_a(\varphi)$, where ι_a is the ath injection into $\coprod_{a\in A} (\pi_a)_* \langle \mathcal{L}, d\rangle$, and $\varphi \in \mathcal{L}_1$.
- $\mathcal{P}\mathcal{F}_1$ takes $\langle X, \langle \mathcal{L}, d\rangle\rangle$ to $\langle \mathcal{P}\mathsf{F}_1 X, \langle (\mathcal{L}_1)^{\wedge,\neg}, (d_1')^{\#}\rangle\rangle$, with $d_1' : \mathcal{L}_1 \to \mathcal{P}\mathcal{P}\mathsf{F}_1 X$ taking $\varphi \in \mathcal{L}_1$ to $\mathcal{P}d_1(\varphi) \in \mathcal{P}\mathcal{P}\mathsf{F}_1 X$ [13]. We write $[\mathcal{P}]\varphi$ for the formula of the resulting language which corresponds to $\varphi \in \mathcal{L}_1$, and $\langle \mathcal{P}\rangle\varphi$ for $\neg[\mathcal{P}]\neg\varphi$.

The preceding definitions mirror the construction of modal languages for *Kripke polynomial endofunctors*, as described in [9,5].

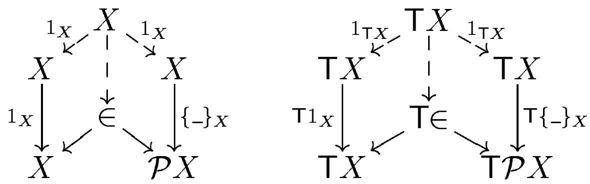
Definition 3.13 A language constructor $\mathcal{F}$ for T **preserves expressivity** if whenever the language $\langle X, \langle \mathcal{L}, d\rangle\rangle$ is expressive, so is the language $\mathcal{F}\langle X, \langle \mathcal{L}, d\rangle\rangle$.

That is, $\mathcal{F}$ preserves expressivity if whenever one starts with a language which is characterising for a set X, by applying $\mathcal{F}$ one obtains a language which is characterising for the set $\mathsf{T}X$.

Example 3.14 The language constructor defined in Example 3.10 preserves expressivity. For, if the left triangle below commutes, so does the top-right triangle.



Also, the bottom-right triangle commutes. For, $t' \in \epsilon_X((\mathsf{T}\{\_\}_X)(t))$ translates to t' $(\mathsf{T}{\in})$ $(\mathsf{T}\{\_\}_X)(t)$. But the fact that the left diagram below is weakly limiting together with the preservation by T of weak limits of W-shaped diagrams (see Remark 2.1) result in the right diagram below also being weakly limiting:



Thus, t' $(\mathsf{T}{\in})$ $(\mathsf{T}\{\_\}_X)(t)$ is equivalent to $t' = t$. Hence, $\epsilon_X((\mathsf{T}\{\_\}_X)(t)) = \{t\}$.

Example 3.15 If $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ is κ-accessible and Λ is a separating set of predicate liftings for T, then the language constructor $\mathcal{F}_\Lambda$ defined in Example 3.11 preserves expressivity (for a suitable choice of σ). For, if $i : X \to \mathcal{L}$

[13] Again, since $\langle \mathcal{L}_1, d_1'\rangle$ does not qualify as a language, a closure under $\wedge$ and $\neg$ has to be performed.

satisfies the condition in Definition 3.6, one can define $i' : \mathsf{T}X \to \mathcal{L}'$ by:

$$i'(t) = \bigwedge_{\substack{\lambda \in \Lambda \\ Y \in \mathcal{P}X \\ \lambda_X(Y) \ni t}} [\lambda]\varphi_Y \;\wedge\; \bigwedge_{\substack{\lambda \in \Lambda \\ Y \in \mathcal{P}X \\ \overline{\lambda_X(\overline{Y})} \ni t}} \langle\lambda\rangle\varphi_Y \,, \quad t \in \mathsf{T}X$$

with φ_Y being given by $\bigvee_{y \in Y} i(y)$ for any $Y \in \mathcal{P}X$ [14] [15]. It then follows immediately from the definition of d' and from the fact that $d(\varphi_Y) = \bigcup_{y\in Y} d(i(y)) = \bigcup_{y \in Y} \{y\} = Y$ for any $Y \in \mathcal{P}X$, that $t \in d'(i'(t))$. Now assume $t' \neq t$. Then, by Λ being saturated, one of the following is true:

(i) there exist $\lambda \in \Lambda$ and $Y \in \mathcal{P}X$ such that $t \in \lambda_X(Y)$ but $t' \notin \lambda_X(Y)$;

(ii) there exist $\lambda \in \Lambda$ and $Y \in \mathcal{P}X$ such that $t' \in \lambda_X(Y)$ but $t \notin \lambda_X(Y)$.

Depending on which of these holds, either $[\lambda]\varphi_Y$ or $\langle\lambda\rangle\varphi_{\overline{Y}}$ does not hold in t', while $t \in \lambda_X(Y)$ and respectively $t \in \overline{\lambda_X(Y)}$ holds. Hence, $t' \notin d'(i'(t))$. This concludes the proof of the fact that $\mathcal{F}_\Lambda$ preserves expressivity.

Example 3.16 A slightly less general setting than the one in Example 3.11 is provided by sets of predicate liftings Λ subject to the additional constraint that $\lambda_X : \mathcal{P}X \to \mathcal{P}\mathsf{T}X$ preserves intersections (and hence has a left adjoint $\lambda_X^* : \mathcal{P}\mathsf{T}X \to \mathcal{P}X$) for any $X \in |\mathsf{Set}|$ and any $\lambda \in \Lambda$. Such natural transformations are known to arise from natural transformations $\mu : \mathsf{T} \Rightarrow \mathcal{P}$ (see e.g. [8, Proposition 6.3]). For $t \in \mathsf{T}X$, the elements of $\lambda_X^*(\{t\})$ can be regarded as T-successors of t. In this case, $i' : \mathsf{T}X \to \mathcal{L}'$ can alternatively be defined by:

$$i'(t) = \bigwedge_{\lambda\in\Lambda} \Big([\lambda]\big(\bigvee_{x\in\lambda_X^*(\{t\})} \varphi_x\big) \;\wedge\; \bigwedge_{x\in\lambda_X^*(\{t\})} (\langle\lambda\rangle\varphi_x) \Big) \,, \quad t \in \mathsf{T}X$$

with φ_x being given by $i(x)$ [16]. Then, replacing the requirement that Λ is separating with the (slightly stronger) condition that, for any $X \in |\mathsf{Set}|$, $t_1 \neq t_2$ implies $\lambda_X^*(\{t_1\}) \neq \lambda_X^*(\{t_2\})$ for some $\lambda \in \Lambda$, one obtains an alternative proof of the fact that $\mathcal{F}_\Lambda$ preserves expressivity (in this more restricted setting). For, the definitions of d', d and λ_X^* yield $t \in d'(i'(t))$. Also, for $t' \neq t$, the condition on Λ yields $\lambda \in \Lambda$ and $x' \in X$ such that either $x' \in \lambda_X^*(\{t\}) \setminus \lambda_X^*(\{t'\})$ or $x' \in \lambda_X^*(\{t'\}) \setminus \lambda_X^*(\{t\})$. Depending on which of these holds, either $t' \notin d'(\langle\lambda\rangle\varphi_{x'})$ (as $d(\varphi_{x'}) = \{x'\}$ and $x' \notin \lambda_X^*(\{t'\})$) while $x' \in \lambda_X^*(\{t\})$, or $t' \notin d'([\lambda](\bigvee_{x\in\lambda_X^*(\{t\})} \varphi_x))$ (as $x' \notin d(\bigvee_{x\in\lambda_X^*(\{t\})} \varphi_x) = \lambda_X^*(\{t\})$). Hence, $t' \notin d'(i'(t))$. This definition of i' provides simpler characterising formulae.

[14] It is shown in [8, Section 7] that both the disjunctions defining the $\varphi_Y s$ and the two conjunctions defining $i'(t)$ can be brought down to a size which does not exceed some fixed σ, with σ depending only on κ and on $\mathsf{card}(\Lambda)$.

[15] Note that the closure of $\mathcal{L}$ under $\bigwedge$ and $\neg$ (and hence also under $\bigvee$) gives $\varphi_Y \in \mathcal{L}$, and therefore $i'(t) \in \mathcal{L}'$.

[16] Note the resemblance between $i'(t)$ and the characterising formulae of infinitary modal logic, as defined e.g. in [7, Section 2.1].

Finally, the following result holds for the language constructors considered in Remark 3.12.

Proposition 3.17 *$\mathcal{F}_A$ and $\mathcal{F}_{\mathsf{Id}}$ preserve expressivity. Also, if $\mathcal{F}_1$ and $\mathcal{F}_2$ preserve expressivity, then so do $\mathcal{F}_1 \otimes \mathcal{F}_2$, $\mathcal{F}_1 \oplus \mathcal{F}_2$, $(\mathcal{F}_1)^A$ and $\mathcal{P}\mathcal{F}_1$.*

Proof (Sketch). The first statement follows immediately from the definitions of $\mathcal{F}_A$ and $\mathcal{F}_{\mathsf{Id}}$. Now let $i_1 : \mathsf{F}_1 X \to \mathcal{L}_1$ and $i_2 : \mathsf{F}_2 X \to \mathcal{L}_2$ satisfy the condition in Definition 3.6, and define:

- $i_\otimes : \mathsf{F}_1 X \times \mathsf{F}_2 X \to (\mathcal{L}_1 + \mathcal{L}_2)^{\wedge,\neg}, \;\; (f_1, f_2) \;\mapsto\; [\pi_1] i_1(f_1) \wedge [\pi_2] i_2(f_2)$
- $i_\oplus : \mathsf{F}_1 X + \mathsf{F}_2 X \to (\mathcal{L}_1 + \mathcal{L}_2)^{\wedge,\neg}, \;\; \iota_j(f_j) \;\mapsto\; [\kappa_j] i_j(f_j) \wedge [\kappa_l] \bot, \;\; \{j, l\} = \{1, 2\}$
- $i_A : (\mathsf{F}_1 X)^A \to (\coprod_{a \in A} \mathcal{L}_1)^{\wedge,\neg}, \; f \mapsto \bigwedge_{a \in A} [a] i_1(f(a))$
- $i_\mathcal{P} : \mathcal{P}\mathsf{F}_1 X \to (\mathcal{L}_1)^{\wedge,\neg}, \;\; Y \;\mapsto\; [\mathcal{P}](\bigvee_{f_1 \in Y} i_1(f_1)) \wedge \bigwedge_{f_1 \in Y} \langle \mathcal{P} \rangle i_1(f_1)$

Some straightforward calculations show that these functions also satisfy the condition in Definition 3.6. □

4 Expressive Languages for Coalgebras

We are now ready to define languages for coalgebras of endofunctors.

Definition 4.1 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$. Also, let $\mathsf{U} : \mathsf{Coalg}(\mathsf{T}) \to \mathsf{Set}$ denote the functor taking T-coalgebras to their carrier. A **language for T-coalgebras** is a pair $\langle \mathcal{L}, \mathsf{d} \rangle$ with $\mathcal{L}$ a set carrying B_σ-structure and $\mathsf{d} : \mathcal{L} \Rightarrow \hat{\mathcal{P}} \circ \mathsf{U}$ a natural transformation[17] such that $\mathsf{d}_\gamma : \mathcal{L} \to \mathcal{P}C$ preserves the B_σ-structure for each T-coalgebra $\langle C, \gamma \rangle$. Given a T-coalgebra $\langle C, \gamma \rangle$ together with $c \in C$ and $\varphi \in \mathcal{L}$, one writes $c \models \varphi$ for $c \in \mathsf{d}_\gamma(\varphi)$. A **map** between languages $\langle \mathcal{L}, \mathsf{d} \rangle$ and $\langle \mathcal{L}', \mathsf{d}' \rangle$ for T-coalgebras is a function $l : \mathcal{L} \to \mathcal{L}'$, itself preserving the B_σ-structure, such that $\mathsf{d}'_\gamma \circ l = \mathsf{d}_\gamma$ for each T-coalgebra $\langle C, \gamma \rangle$. The category of languages for T-coalgebras and maps between them is denoted $\mathsf{CLang}(\mathsf{T})$.

Given a language $\langle \mathcal{L}, \mathsf{d} \rangle$ for T-coalgebras, the naturality of d amounts to the denotations of formulae being reflected by T-coalgebra homomorphisms. As a result, the denotations of formulae are invariant under behavioural equivalence: given T-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$, if $c \in C$ and $d \in D$ are behaviourally equivalent, and hence identified by some T-coalgebra homomorphisms $f : \langle C, \gamma \rangle \to \langle E, \eta \rangle$ and $g : \langle D, \delta \rangle \to \langle E, \eta \rangle$, then $c \in \mathsf{d}_\gamma(\varphi)$ if and only if $f(c) = g(d) \in \mathsf{d}_\eta(\varphi)$ if and only if $d \in \mathsf{d}_\delta(\varphi)$, for any $\varphi \in \mathcal{L}$. In the terminology of [8], any language for T-coalgebras is adequate. Furthermore, in the presence of a final T-coalgebra $\langle Z, \zeta \rangle$, $\langle \mathcal{L}, \mathsf{d} \rangle$ is fully determined by d_ζ: if $!_\gamma : \langle C, \gamma \rangle \to \langle Z, \zeta \rangle$ denotes the unique T-coalgebra homomorphism from a T-coalgebra $\langle C, \gamma \rangle$ to the final one, then $\mathsf{d}_\gamma = \hat{\mathcal{P}}\mathsf{U}!_\gamma \circ \mathsf{d}_\zeta$.

[17] Here $\mathcal{L}$ is also used to denote the constant functor $X \mapsto \mathcal{L}$.

Remark 4.2 One can also define a category CLang of languages for coalgebras, whose objects are given by pairs $\langle \mathsf{T}, \langle \mathcal{L}, \mathsf{d} \rangle \rangle$ with $\langle \mathcal{L}, \mathsf{d} \rangle \in |\mathsf{CLang}(\mathsf{T})|$, and whose arrows from $\langle \mathsf{T}, \langle \mathcal{L}, \mathsf{d} \rangle \rangle$ to $\langle \mathsf{T}', \langle \mathcal{L}', \mathsf{d}' \rangle \rangle$ are given by pairs $\langle \eta, l \rangle$, with $\eta : \mathsf{T}' \Rightarrow \mathsf{T}$ and $l : \mathcal{L} \to \mathcal{L}'$ being such that $\hat{\mathcal{P}}\eta \circ \mathsf{d} = \mathsf{d}' \circ l$. Moreover, one can show that the functor taking $\langle \mathsf{T}, \langle \mathcal{L}, \mathsf{d} \rangle \rangle$ to T and $\langle \eta, l \rangle$ to η^{op} is a cofibration. The coreindexing functors $\eta_* : \mathsf{CLang}(\mathsf{T}) \to \mathsf{CLang}(\mathsf{T}')$ provide canonical translations of languages for T-coalgebras into languages for T'-coalgebras.

Following [8], we define expressivity of a language for coalgebras as being the ability of the language to capture behavioural equivalence.

Definition 4.3 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$. A language $\langle \mathcal{L}, \mathsf{d} \rangle$ for T-coalgebras is called **expressive** if, for any T-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ and any $c \in C$ and $d \in D$, ($c \models \varphi$ if and only if $d \models \varphi$, for any $\varphi \in \mathcal{L}$) implies c and d are T-behaviourally equivalent.

Remark 4.4 For each regular cardinal α, one can derive a language $\langle \mathcal{L}, \mathsf{d} \rangle$ for T-coalgebras from a Z_α-language $\langle \mathcal{L}, d \rangle$ by letting $\mathsf{d}_\gamma = \hat{\mathcal{P}}\gamma_\alpha \circ d$:

$$\begin{array}{ccc} \mathcal{L} & & \\ {\scriptstyle d}\downarrow & & \\ \mathcal{P}Z_\alpha & \xrightarrow{\hat{\mathcal{P}}\gamma_\alpha} & \mathcal{P}C \end{array}$$

for each T-coalgebra $\langle C, \gamma \rangle$, with the maps $\gamma_\alpha : C \to Z_\alpha$ being as in Remark 2.4. The languages which interest us are those obtained by taking $\alpha = \kappa$, where T is κ-accessible. For, in this case, if the Z_κ-language $\langle \mathcal{L}, d \rangle$ is expressive, then so is the induced language $\langle \mathcal{L}, \mathsf{d} \rangle$.

Proposition 4.5 *Let* $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ *denote a* κ*-accessible endofunctor, and let* $\langle \mathcal{L}, d \rangle$ *denote an expressive* Z_κ*-language. Then, the induced language* $\langle \mathcal{L}, \mathsf{d} \rangle$ *for* T*-coalgebras is also expressive.*

Proof. Let $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ denote T-coalgebras, and let $c \in C$ and $d \in D$ be such that $c \models \varphi$ if and only if $d \models \varphi$, for any $\varphi \in \mathcal{L}$. By the definition of $\langle \mathcal{L}, \mathsf{d} \rangle$, $c \models \varphi$ holds precisely when $\gamma_\kappa(c) \in d(\varphi)$. Hence, $c \models i(\gamma_\kappa(c))$, with $i : Z_\kappa \to \mathcal{L}$ being as in Definition 3.6. But then $d \models i(\gamma_\kappa(c))$, or equivalently, $\delta_\kappa(d) \in d(i(\gamma_\kappa(c))) = \{\gamma_\kappa(c)\}$. Thus, $\delta_\kappa(d) = \gamma_\kappa(c)$. It then follows from the definition of Z_κ together with Remark 2.5 that c and d are behaviourally equivalent. □

The remainder of this section is devoted to deriving an expressive Z_κ-language, and hence an expressive language for T-coalgebras, in the case when T is κ-accessible. By building on the construction of the final sequence of T, we define an ordinal-indexed sequence of languages, whose κth element induces an expressive language for T-coalgebras.

Definition 4.6 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, and let $\mathcal{F}$ denote a language constructor for T. The **language sequence induced by** $\mathcal{F}$ is given by the initial sequence[18] of $\mathcal{F} : \mathsf{Lang} \to \mathsf{Lang}$.

That is, the language sequence induced by $\mathcal{F}$ is an ordinal-indexed sequence of languages $(\langle Z_\alpha, \langle \mathcal{L}_\alpha, d_\alpha \rangle \rangle)$, together with a family $(\langle p^\alpha_\beta, \iota^\alpha_\beta \rangle)_{\beta \leq \alpha}$ of maps $\langle p^\alpha_\beta, \iota^\alpha_\beta \rangle : \langle Z_\beta, \langle \mathcal{L}_\beta, d_\beta \rangle \rangle \to \langle Z_\alpha, \langle \mathcal{L}_\alpha, d_\alpha \rangle \rangle$ between languages, satisfying:

- $\langle Z_{\alpha+1}, \langle \mathcal{L}_{\alpha+1}, d_{\alpha+1} \rangle \rangle = \mathcal{F} \langle Z_\alpha, \langle \mathcal{L}_\alpha, d_\alpha \rangle \rangle$
- $\langle p^{\alpha+1}_{\beta+1}, \iota^{\alpha+1}_{\beta+1} \rangle = \mathcal{F} \langle p^\alpha_\beta, \iota^\alpha_\beta \rangle$ for $\beta \leq \alpha$
- $\langle p^\alpha_\alpha, \iota^\alpha_\alpha \rangle = 1_{\langle \mathcal{L}_\alpha, d_\alpha \rangle}$
- $\langle p^\alpha_\gamma, \iota^\alpha_\gamma \rangle = \langle p^\alpha_\beta, \iota^\alpha_\beta \rangle \circ \langle p^\beta_\gamma, \iota^\beta_\gamma \rangle$ for $\gamma \leq \beta \leq \alpha$
- if α is a limit ordinal, then the cocone $\langle Z_\alpha, \langle \mathcal{L}_\alpha, d_\alpha \rangle \rangle$, $(\langle p^\alpha_\beta, \iota^\alpha_\beta \rangle)_{\beta < \alpha}$ for $(\langle p^\beta_\gamma, \iota^\beta_\gamma \rangle)_{\gamma \leq \beta < \alpha}$ is colimiting[19].

Then, it immediately follows that the Set-sequence underlying the language sequence induced by $\mathcal{F}$ coincides with the final sequence of T.

Proposition 4.7 *Let* $(\langle Z_\alpha, \langle \mathcal{L}_\alpha, d_\alpha \rangle \rangle), (\langle p^\alpha_\beta, \iota^\alpha_\beta \rangle)_{\beta \leq \alpha}$ *denote the language sequence induced by* $\mathcal{F}$*. Then,* $(Z_\alpha), (p^\alpha_\beta)_{\beta \leq \alpha}$ *is the final sequence of* T.

Proof (Sketch). The fact that $\mathcal{F}$ is fibred over T yields $Z_{\alpha+1} = \mathsf{T} Z_\alpha$ for any α, as well as $p^{\alpha+1}_{\beta+1} = \mathsf{T} p^\alpha_\beta$ for any $\beta \leq \alpha$. Also, the fact that colimits of diagrams in Lang are constructed using the limits of the underlying diagrams in Set (see Proposition 3.5) results in the cone $Z_\alpha, (p^\alpha_\beta)_{\beta < \alpha}$ for $(p^\beta_\gamma)_{\gamma \leq \beta < \alpha}$ being limiting. □

Remark 4.8 The existence of maps $\langle p^\alpha_\beta, \iota^\alpha_\beta \rangle : \langle Z_\beta, \langle \mathcal{L}_\beta, d_\beta \rangle \rangle \to \langle Z_\alpha, \langle \mathcal{L}_\alpha, d_\alpha \rangle \rangle$ amounts to the commutativity of diagrams of form:

$$\begin{CD} \mathcal{L}_\beta @>{\iota^\alpha_\beta}>> \mathcal{L}_\alpha \\ @V{d_\beta}VV @VV{d_\alpha}V \\ \mathcal{P} Z_\beta @>>{\hat{\mathcal{P}} p^\alpha_\beta}> \mathcal{P} Z_\alpha \end{CD}$$

with $\beta \leq \alpha$.

Our main result concerns the expressivity of the languages belonging to the language sequence induced by a language constructor.

Theorem 4.9 *Let* $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$*, and let* $\mathcal{F} : \mathsf{Lang} \to \mathsf{Lang}$ *denote a language constructor for* T *which preserves expressivity. Then, the* Z_α*-language* $\langle \mathcal{L}_\alpha, d_\alpha \rangle$ *is expressive, for any* $\alpha \leq \sigma$.

Proof (Sketch). We use transfinite induction to prove the above statement.

If $\alpha = \beta + 1$, the fact that $\langle \mathcal{L}_\beta, d_\beta \rangle$ is expressive together with the fact that $\mathcal{F}$ preserves expressivity result in $\langle \mathcal{L}_\alpha, d_\alpha \rangle = \mathcal{F} \langle \mathcal{L}_\beta, d_\beta \rangle$ being expressive.

---

[18] The initial sequence of an endofunctor is defined dually to its final sequence (see Definition 2.2).

[19] Recall from Proposition 3.5 that Lang has colimits.

If α is a limit ordinal, one can define $i_\alpha : Z_\alpha \to \mathcal{L}_\alpha$ by:

$$i_\alpha(x) = \bigwedge_{\beta<\alpha} \iota_\beta^\alpha(i_\beta(p_\beta^\alpha(x)))\ ,\ x \in Z_\alpha$$

once each $i_\beta : Z_\beta \to \mathcal{L}_\beta$ with $\beta < \alpha$ has been defined[20] [21]. Then, the fact that $d_\alpha \circ i_\alpha = \{\_\}_{Z_\alpha}$ follows from the expressivity of $\langle \mathcal{L}_\beta, d_\beta \rangle$ for each $\beta < \alpha$, together with Remark 4.8, the preservation by d_α of the B_σ-structure, and the universal property of $Z_\alpha, (p_\beta^\alpha)_{\beta<\alpha}$. □

Taking $\langle \mathcal{L}, d \rangle = \langle \mathcal{L}_\kappa, d_\kappa \rangle$ in Remark 4.4 yields a language for T-coalgebras.

Definition 4.10 Let $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ denote a κ-accessible endofunctor, and let $\mathcal{F} : \mathsf{Lang} \to \mathsf{Lang}$ denote a language constructor for T. The **language induced by** $\mathcal{F}$ is given by $\langle \mathcal{L}_\kappa, \mathsf{d}_\kappa \rangle$.

Example 4.11 The language induced by the language constructor in Example 3.10 is the language of coalgebraic logic [7] enriched with negation.

Example 4.12 The language induced by the language constructor in Example 3.11 coincides with the language used in [8].

Example 4.13 If K is a Kripke polynomial endofunctor, and if $\mathcal{F}_\mathsf{K}$ is defined by induction on the structure of K using the rules in Remark 3.12, then the language induced by $\mathcal{F}_\mathsf{K}$ coincides with the language used in [9,5].

By combining Theorem 4.9 and Proposition 4.5 we obtain the following result (which holds for any choice of σ such that $\sigma \geq \kappa$).

Corollary 4.14 *Let* $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ *denote a* κ*-accessible endofunctor, and let* $\mathcal{F} : \mathsf{Lang} \to \mathsf{Lang}$ *denote a language constructor for* T *which preserves expressivity. Then, the language induced by* $\mathcal{F}$ *is expressive.*

5 Compositionality

The question of deriving expressive languages for (coalgebras of) functor compositions from expressive languages for (coalgebras of) the functors being composed has not, to our knowledge, been treated systematically in existing approaches to defining modal logics for coalgebras[22]. The present section provides a general solution to this question, based on combining language constructors for different endofunctors.

Proposition 5.1 *If* $\mathcal{F}_1$ *is a language constructor for* T_1 *and* $\mathcal{F}_2$ *is a language constructor for* T_2*, then* $\mathcal{F}_2 \circ \mathcal{F}_1$ *is a language constructor for* $\mathsf{T}_2 \circ \mathsf{T}_1$*. Furthermore, if* $\mathcal{F}_1$ *and* $\mathcal{F}_2$ *preserve expressivity, then so does* $\mathcal{F}_2 \circ \mathcal{F}_1$*.*

---

[20] In particular, $i_0 \equiv \top$.
[21] Note that, since $\alpha \leq \sigma$, the size of the conjunction used to define i_α does not exceed σ.
[22] An exception to this is perhaps the approach in [9,5], where compositionality is implicit in the definition of the languages.

Proof. The first part of the statement follows immediately from the commutativity of the left and right squares of the following diagram:

$$\begin{array}{ccccc} \mathsf{Lang} & \xrightarrow{\mathcal{F}_1} & \mathsf{Lang} & \xrightarrow{\mathcal{F}_2} & \mathsf{Lang} \\ {\scriptstyle \mathsf{E}}\downarrow & & \downarrow{\scriptstyle \mathsf{E}} & & \downarrow{\scriptstyle \mathsf{E}} \\ \mathsf{Set}^{\mathrm{op}} & \xrightarrow[\mathsf{T}_1^{\mathrm{op}}]{} & \mathsf{Set}^{\mathrm{op}} & \xrightarrow[\mathsf{T}_2^{\mathrm{op}}]{} & \mathsf{Set}^{\mathrm{op}} \end{array}$$

The second part of the statement is a direct consequence of the definition of expressivity preserving language constructors. □

Corollary 5.2 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be as in Proposition 5.1. If $\mathcal{F}_1$ and $\mathcal{F}_2$ preserve expressivity, then the language induced by $\mathcal{F}_2 \circ \mathcal{F}_1$ is expressive.*

Example 5.3 By combining an expressivity preserving language constructor $\mathcal{F}$ for $\mathcal{P}$ with itself, one obtains an expressive language for $\mathcal{P} \circ \mathcal{P}$-coalgebras. Possible values for $\mathcal{F}$ are obtained: (a) by taking $\mathsf{T} = \mathcal{P}$ in Example 3.10; (b) by considering the single predicate lifting $\lambda : \hat{\mathcal{P}} \Rightarrow \hat{\mathcal{P}} \circ \mathcal{P}$ given by $\lambda_C(X) = \mathcal{P}X$ for $X \in \mathcal{P}C$ in Example 3.11; and (c) by considering the language constructor $\mathcal{P}\mathcal{F}_{\mathsf{Id}}$ defined in Remark 3.12.

Remark 5.4 If $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}$ are the language constructors for T_1, T_2 and $\mathsf{T}_2 \circ \mathsf{T}_1$ obtained using the approach described in Example 3.10, then the language for $\mathsf{T}_2 \circ \mathsf{T}_1$-coalgebras induced by $\mathcal{F}_2 \circ \mathcal{F}_1$ is equivalent to (though larger than) the language induced by $\mathcal{F}$.

Another form of compositionality concerns endofunctors of form $\mathsf{F}_1 \times \mathsf{F}_2$, $\mathsf{F}_1 + \mathsf{F}_2$, $(\mathsf{F}_1)^A$ or $\mathcal{P} \circ \mathsf{F}_1$, as considered in Remark 3.12 and Proposition 3.17. In this case, Theorem 4.9 yields the following.

Corollary 5.5 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be as in Proposition 3.17. Then, the languages induced by $\mathcal{F}_A$, $\mathcal{F}_{\mathsf{Id}}$, $\mathcal{F}_1 \otimes \mathcal{F}_2$, $\mathcal{F}_1 \oplus \mathcal{F}_2$, $(\mathcal{F}_1)^A$ and $\mathcal{P}\mathcal{F}_1$ are expressive.*

6 Future Work

We plan to study relationships between language constructors for arbitrary endofunctors on Set, as suggested (independently) by Definition 3.9 and Remark 3.12. Such relationships could be formalised within a category of language constructors indexed by endofunctors, a category which we expect to be cofibred over $[\mathsf{Set}, \mathsf{Set}]$. Colimits in this category are also worth investigating. Another possible direction for future work is to generalise our approach to endofunctors on arbitrary categories.

Acknowledgement

Thanks are due to Alexandru Baltag for comments on a draft of this paper, and to an anonymous referee for suggestions on simplifying the results presented.

References

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

[2] A. Baltag. A logic for coalgebraic simulation. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.

[3] M. Barr. Algebraically compact functors. *Journal of Pure and Applied Algebra*, 82(3):211–231, 1992.

[4] C.A. Hermida. *Fibrations, Logical Predicates and Indeterminates.* PhD thesis, University of Edinburgh, 1993.

[5] B. Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications*, 35:31–59, 2001.

[6] A. Kurz and D. Pattinson. Definability, canonical models, compactness for finitary coalgebraic modal logic. In L.S. Moss, editor, *Coalgebraic Methods in Computer Science*, volume 65.1 of *Electronic Notes in Theoretical Computer Science*, 2002.

[7] L.S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.

[8] D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. Technical report, LMU München, 2002.

[9] M. Rößiger. Coalgebras and modal logic. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.

[10] J. Worrell. On the final sequence of an accessible set functor. To appear in Theoretical Computer Science.