Available online at www.sciencedirect.com

SCIENCE DIRECT®

The Journal of Logic and Algebraic Programming 67 (2006) 87–113

THE JOURNAL OF LOGIC AND ALGEBRAIC PROGRAMMING

www.elsevier.com/locate/jlap

An institution of modal logics for coalgebras

Corina Cîrstea [1]

School of Electronics and Computer Science, University of Southampton, Southampton SO16 7PX, UK

Abstract

This paper presents a modular framework for the specification of certain inductively-defined coalgebraic types. Modal logics for coalgebras of polynomial endofunctors on the category of sets have been studied in [M. Rößiger, Coalgebras and modal logic, in: H. Reichel (Ed.), Coalgebraic Methods in Computer Science, Electronic Notes in Theoretical Computer Science, vol. 33, Elsevier Science, 2000, pp. 299–320; B. Jacobs, Many-sorted coalgebraic modal logic: a model-theoretic study, Theoretical Informatics and Applications 35(1) (2001) 31–59]. These logics are here generalised to endofunctors on categories of sorted sets, in order to allow collections of inter-related types to be specified simultaneously. The inductive nature of the coalgebraic types considered is then used to formalise semantic relationships between different types, and to define translations between the associated logics. The resulting logical framework is shown to be an institution, whose specifications and specification morphisms admit final and respectively cofree models.

Keywords: Coalgebras; Modal logic; Institutions

1. Introduction

During the last decade, coalgebras have been used to model various kinds of state-based, dynamical systems, including transition systems, automata, and object systems [6,5,7,18]. The emphasis in such modelling is on the observable properties of system states, with the indistinguishability of states by observations being captured by coalgebraic bisimulation.

The use of coalgebras as models for systems [18] generalises the use of transition systems as models for processes [17], with coalgebraic bisimulation generalising the standard, process-theoretic notion of bisimulation. And since, in the case of transition systems, bisimulation-invariant properties of processes can be expressively captured within modal

E-mail address: cc2@ecs.soton.ac.uk

URL: http://www.ecs.soton.ac.uk/~cc2/

[1] Research partially supported by St. John's College, Oxford.

1567-8326/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jlap.2005.09.004

logic[2] [4], when seeking to formalise bisimulation-invariant properties of system states it is natural to consider logics which are modal in nature.

Various such logics have been studied in recent years [14,9,16,8,15]. The idea underlying these logics is to generalise the use of standard modal operators to quantify over successor states in Kripke structures, to arbitrary coalgebraic structures. Coalgebras generalise Kripke structures by replacing the accessibility relations between their states with arbitrarily complex ways of observing the states of a system. Correspondingly, modal logics for coalgebras generalise standard modal logic by employing type-specific modal operators that arise from particular ways of observing the system states in one step. And while such logics differ in the kinds of coalgebraic types they apply to, as well as in the level of abstraction of the modal operators they employ, they all capture bisimulation, i.e. the logical equivalence relation between the states of coalgebras coincides with the bisimilarity relation[3].

The present paper is concerned with logics for inductively-defined coalgebraic types, as considered in [16,8]. The definition of these logics exploits the inductive nature of the underlying types in order to derive a concrete modal language for each particular type. However, the approach in [16,8] only considers coalgebras with one sort. Moreover, different, but semantically-related coalgebraic types give rise to different, but not yet formally related modal logics. The aim of this paper is to define a specification framework wherein the logics associated to semantically-related types can themselves be related. Such a framework would provide support for modular specification, as well as for specification reuse.

The modal logics introduced in [16] have, as models, coalgebras of certain endofunctors on the category of sets. These endofunctors are constructed from constant and identity functors, using products, coproducts, certain exponentials and powersets. Here we consider similar endofunctors on categories of sorted sets, with sorts being used to denote coalgebraic types, and with the components of the endofunctors in question defining the particular (and possibly interdependent) structures associated to these types. Moreover, we use natural transformations arising from the structure of particular endofunctors to capture semantic relationships between different coalgebraic types. Such natural transformations are shown to induce translations between the classes of coalgebras associated to these types, as well as translations between the modal languages induced by these types. Moreover, the satisfaction of modal formulae by coalgebras is shown to be preserved and reflected by these translations. That is, the resulting framework constitutes an *institution* [3]. The morphisms of this institution capture both refinement and encapsulation relations between coalgebraic types, as illustrated by several examples. Furthermore, coalgebraic specifications are shown to admit final coalgebras, whereas coalgebraic specification morphisms come equipped with cofree coalgebra constructions.

The paper is structured as follows. Section 2 establishes the notation for subsequent sections, and recalls some basic coalgebraic concepts. Section 3 generalises the approach in [16,8] to categories of sorted sets, and at the same time defines semantic relationships between the coalgebraic types of interest. Section 4 defines translations between the logics associated to semantically-related types, and shows that the resulting framework is an institution. Section 5 investigates the semantic constructions associated to this institution. Finally, Section 6 summarises the results presented.

[2] The formulae of modal logic are in fact interpreted over *Kripke structures*; these are annotated transition systems, with the annotations specifying the atomic propositions which hold in particular states.

[3] Restrictions similar to those used in standard modal logic (i.e. image finiteness) are required to obtain expressiveness for these logics.

2. Preliminaries

Given a category C, its collection of objects will be denoted $|\mathsf{C}|$, and its collection of arrows will be denoted $\|\mathsf{C}\|$. Also, the identity map on an object C will be denoted 1_C, while the equality relation on C will be denoted $\langle \Delta_C, 1_C, 1_C \rangle$.[4] Binary products (coproducts) will be denoted $X \times Y$ ($X + Y$), with canonical projections (injections) $\pi_1 : X \times Y \to X$ and $\pi_2 : X \times Y \to Y$ ($\kappa_1 : X \to X + Y$ and $\kappa_2 : Y \to X + Y$). Exponentials will be denoted X^Y, the induced evaluation map will be denoted $eval_{X,Y} : X^Y \times Y \to X$, and the curried version of a function $f : X \times Y \to Z$ will be denoted $f^* : X \to Z^Y$.

The opposite category of a category C will be denoted C^{op}. The category of categories and functors will be denoted Cat. Also, for conciseness of presentation, functor applications $\mathsf{F}(X)$ will be written $\mathsf{F}X$, while functor compositions $\mathsf{F} \circ \mathsf{G}$ will sometimes be written FG. Given categories C and D, the category of functors from C to D and natural transformations between them will be denoted $[\mathsf{C}, \mathsf{D}]$.

Throughout the paper, Set will denote the category of sets and functions, and $1 = \{*\}$ will denote a one-element set. The *identity functor* on Set will be denoted $\mathsf{Id} : \mathsf{Set} \to \mathsf{Set}$, while the *powerset functor*, taking a set to the set of its subsets, and a function to its direct image, will be denoted $\mathscr{P} : \mathsf{Set} \to \mathsf{Set}$. For a regular cardinal κ, the *κ-bounded powerset functor*, taking a set to the set of its subsets of cardinality smaller that κ, will be denoted $\mathscr{P}_\kappa : \mathsf{Set} \to \mathsf{Set}$. In particular, the *finite powerset functor* $\mathscr{P}_\omega$ takes a set to the set of its finite subsets.

Given a set S (of *sorts*), Set^S will denote the category of *S-sorted sets and S-sorted functions*: its objects are given by S-indexed families $C = (C_s)_{s \in S}$ of sets, while arrows from C to D are given by S-indexed families $f = (f_s)_{s \in S}$ of functions, with $f_s : C_s \to D_s$ for $s \in S$. For $s \in S$, the *s-projection functor*, taking S-sorted sets/functions to their s-sorted component, will be denoted $\Pi_s : \mathsf{Set}^S \to \mathsf{Set}$. (If $S = \{s\}$, we identify $\Pi_s : \mathsf{Set}^{\{s\}} \to \mathsf{Set}$ with $\mathsf{Id} : \mathsf{Set} \to \mathsf{Set}$.)

Now let $\mathsf{T} : \mathsf{C} \to \mathsf{C}$ be an endofunctor.[5] A *T-coalgebra* is a pair $\langle C, \gamma \rangle$ with $C \in |\mathsf{C}|$ (the *carrier* of the coalgebra) and $(\gamma : C \to \mathsf{T}C) \in \|\mathsf{C}\|$[6] (the *coalgebra map*). Also, a *T-coalgebra homomorphism* between T-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ is a C-arrow $f : C \to D$ additionally satisfying $\mathsf{T}f \circ \gamma = \delta \circ f$. The category of T-coalgebras and T-coalgebra homomorphisms is denoted $\mathsf{Coalg}(\mathsf{T})$.

In what follows, we will only be concerned with *weak pullback preserving endofunctors*[7] $\mathsf{T} : \mathsf{C} \to \mathsf{C}$. For such endofunctors, notions of *T-subcoalgebra* and *T-bisimulation* are defined as follows.

For a T-coalgebra $\langle C, \gamma \rangle$, a *$\mathsf{T}$-subcoalgebra of $\langle C, \gamma \rangle$* is given by a T-coalgebra $\langle D, \delta \rangle$ together with a T-coalgebra homomorphism $m : \langle D, \delta \rangle \to \langle C, \gamma \rangle$, with $m : D \to C$ a C-monomorphism.[8] The category whose arrows are T-subcoalgebras of $\langle C, \gamma \rangle$, and whose arrows from $\langle \langle D, \delta \rangle, m \rangle$ to $\langle \langle D', \delta' \rangle, m' \rangle$ are T-coalgebra homomorphisms $f : \langle D, \delta \rangle \to \langle D', \delta' \rangle$ additionally satisfying $m' \circ f = m$, is denoted $\mathsf{SubCoalg}(\langle C, \gamma \rangle)$. Also, given a

[4] The category-theoretic definition of relations as monic spans is used here.

[5] For $\mathsf{C} = \mathsf{Set}^S$, such an endofunctor can be used to specify the structure associated to an S-indexed collection of coalgebraic types.

[6] Each such γ provides a particular interpretation of the structure specified by T.

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

[8] The preservation of weak pullbacks (and hence of weak kernel pairs) by T results in the T-subcoalgebras of $\langle C, \gamma \rangle$ being in one-to-one correspondence with the $\mathsf{Coalg}(\mathsf{T})$-subobjects of $\langle C, \gamma \rangle$.

C-monomorphism $\iota : X \to C$, the full subcategory of $\mathsf{SubCoalg}(\langle C, \gamma\rangle)$ whose objects $\langle\langle D, \delta\rangle, m\rangle$ are such that $m : D \to C$ factors through ι[9] is denoted $\mathsf{SubCoalg}(\langle C, \gamma\rangle, \iota)$.

Given T-coalgebras $\langle C, \gamma\rangle$ and $\langle D, \delta\rangle$, a T-*bisimulation* between them is a relation[10] $\langle R, \pi_1, \pi_2\rangle$ between C and D, with R carrying a T-coalgebra structure $\rho : R \to \mathsf{T}R$, making $\pi_1 : R \to C$ 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 bisimulations) is called T-*bisimilarity* and is denoted $\sim$.

3. Modal logics for Kripke polynomial endofunctors on categories of sorted sets

Modal logics for an inductively-defined class of endofunctors on Set have been studied in [16,8]. This section generalises the approach in [16,8] to endofunctors on categories of sorted sets.

In order to facilitate the definition of a modular specification framework in the next section, the components of such endofunctors are regarded as objects of a category whose arrows, arising naturally from the structure of the functors, capture semantic relationships between coalgebraic types.

Definition 1. Let S denote a set (of sorts). The **category of Kripke polynomial functors on** Set^S, denoted KP_S, is the subcategory of $[\mathsf{Set}^S, \mathsf{Set}]$ defined by

- the objects of KP_S are generated by the following grammar:

$$\mathsf{F} ::= D \mid \Pi_s \mid \mathsf{F}_1 \times \mathsf{F}_2 \mid \mathsf{F}_1 + \mathsf{F}_2 \mid \mathsf{F}^D \mid \mathscr{P} \circ \mathsf{F}$$

  with $D : \mathsf{Set}^S \to \mathsf{Set}$ denoting the constant functor $X \mapsto D$, for $D \neq \emptyset$, and with $\mathsf{F}^D : \mathsf{Set}^S \to \mathsf{Set}$ denoting the functor $X \mapsto (\mathsf{F}X)^D$, for D finite and non-empty;
- the arrows of KP_S are generated from:
  (i) natural transformations $\alpha : D' \Rightarrow D$ induced by functions $\alpha : D' \to D$ with $\alpha^{-1}(d)$ finite for all $d \in D$, and
  (ii) constant natural transformations $d : \mathsf{F} \Rightarrow D$ (given by $d_X(f) = d$ for all $f \in \mathsf{F}X$), for $d \in D$,

  by closing KP_S under products, coproducts, exponentials and powersets.

The restrictions concerning the finiteness of the sets $\alpha^{-1}(d)$ with $d \in D$, and of the sets D appearing as exponents in the definition of Kripke polynomial functors will be required later in the paper, namely when defining translations between the logics induced by different functors (Definition 25).

Replacing the closure under powersets in Definition 1 with closure under finite powersets yields a notion of **finite Kripke polynomial functor on** Set^S. Most of the results in this paper are formulated for Kripke polynomial functors[11], however, they also hold for finite Kripke polynomial functors.

[9] That is, $m = \iota \circ n$ for some $n : D \to X$.

[10] In Set, relations are given by subsets of the cartesian product $C \times D$. In Set^S, relations are given by S-indexed families $(R_s)_{s\in S}$, with R_s a subset of $C_s \times D_s$ for $s \in S$.

[11] Exceptions to this are Lemma 19 and Proposition 20, which *only* hold for finite Kripke polynomial functors.

Remark 2. An immediate consequence of the definition of KP_S is the existence, in this category, of arrows of form:

- $\pi_i : \mathsf{F}_1 \times \mathsf{F}_2 \Rightarrow \mathsf{F}_i$ with $i \in \{1, 2\}$, whenever $\mathsf{F}_i \in |\mathsf{KP}_S|$ for $i = 1, 2$,
- $\langle \eta_1, \eta_2 \rangle : \mathsf{F} \Rightarrow \mathsf{F}_1 \times \mathsf{F}_2$ whenever $(\eta_i : \mathsf{F} \Rightarrow \mathsf{F}_i) \in \| \mathsf{KP}_S \|$ for $i = 1, 2$,
- $\kappa_i : \mathsf{F}_i \Rightarrow \mathsf{F}_1 + \mathsf{F}_2$ with $i \in \{1, 2\}$, whenever $\mathsf{F}_i \in |\mathsf{KP}_S|$ for $i = 1, 2$,
- $[\eta_1, \eta_2] : \mathsf{F}_1 + \mathsf{F}_2 \Rightarrow \mathsf{F}$ whenever $(\eta_i : \mathsf{F}_i \Rightarrow \mathsf{F}) \in \| \mathsf{KP}_S \|$ for $i = 1, 2$,
- $eval_{\mathsf{F},D} : \mathsf{F}^D \times D \Rightarrow \mathsf{F}$ whenever $\mathsf{F}, D \in |\mathsf{KP}_S|$ with D a constant functor induced by some finite, non-empty D,
- $\eta^* : \mathsf{F}' \Rightarrow \mathsf{F}^D$ whenever $(\eta : \mathsf{F}' \times D \Rightarrow \mathsf{F}) \in \| \mathsf{KP}_S \|$ with D a constant functor induced by some finite, non-empty D,
- $\mathscr{P}(\eta) : \mathscr{P} \circ \mathsf{F} \Rightarrow \mathscr{P} \circ \mathsf{F}'$ whenever $(\eta : \mathsf{F} \Rightarrow \mathsf{F}') \in \| \mathsf{KP}_S \|$,

subject to the following equalities:

(1) $\pi_i \circ \langle \eta_1, \eta_2 \rangle = \eta_i$ for $i = 1, 2$.
(2) $[\eta_1, \eta_2] \circ \kappa_i = \eta_i$ for $i = 1, 2$.
(3) $eval_{\mathsf{F},D} \circ (\eta^* \times 1_D) = \eta$.

In particular, KP_S contains arrows of form:

- $\eta_1 \times \eta_2 : \mathsf{F}_1 \times \mathsf{F}_2 \Rightarrow \mathsf{F}'_1 \times \mathsf{F}'_2$ (given by $\langle \eta_1 \circ \pi_1, \eta_2 \circ \pi_2 \rangle$) whenever $(\eta_i : \mathsf{F}_i \Rightarrow \mathsf{F}'_i) \in \| \mathsf{KP}_S \|$ for $i = 1, 2$.
- $\eta_1 + \eta_2 : \mathsf{F}_1 + \mathsf{F}_2 \Rightarrow \mathsf{F}'_1 + \mathsf{F}'_2$ (given by $[\kappa_1 \circ \eta_1, \kappa_2 \circ \eta_2]$) whenever $(\eta_i : \mathsf{F}_i \Rightarrow \mathsf{F}'_i) \in \| \mathsf{KP}_S \|$ for $i = 1, 2$.
- $\eta^D : \mathsf{F}'^D \Rightarrow \mathsf{F}^D$ (given by $(\eta \circ eval_{F',D})^*$) whenever $(\eta : \mathsf{F}' \Rightarrow \mathsf{F}) \in \| \mathsf{KP}_S \|$ and $D \in |\mathsf{KP}_S|$ with D a constant functor induced by some finite, non-empty D.
- $\mathsf{F}^\alpha : \mathsf{F}^D \Rightarrow \mathsf{F}^{D'}$ (given by $(eval_{\mathsf{F},D} \circ (1_{\mathsf{F}^D} \times \alpha))^*$) whenever $\mathsf{F} \in |\mathsf{KP}_S|$ and $(\alpha : D' \Rightarrow D) \in \| \mathsf{KP}_S \|$ with D, D' being constant functors induced by some finite, non-empty D, D'.

The notion of Kripke polynomial endofunctor on Set, as defined in [8], now generalises to categories of sorted sets as follows.

Definition 3. Let S denote a set (of sorts). A **Kripke polynomial endofunctor on** Set^S is an endofunctor $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ such that $\mathsf{T}_s \in |\mathsf{KP}_S|$ for each $s \in S$.

Kripke polynomial endofunctors on Set^S specify the structure associated to an S-indexed collection of (interdependent) coalgebraic types. Any occurrence of the projection functor $\Pi_s : \mathsf{Set}^S \to \mathsf{Set}$ in the definition of $\mathsf{T}_{s'}$, with $s, s' \in S$, specifies a dependence of the type denoted by s' on the type denoted by s.

For $S \simeq 1$, the objects of the category KP_S are precisely the Kripke polynomial endofunctors on Set, as defined in [8]. The emphasis in [8] is, however, on a different aspect of Kripke polynomial endofunctors, namely on the *syntactic* dependencies between these endofunctors, with the notion of *ingredient* being used to capture such a dependency. An endofunctor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is an *ingredient* [8] of a Kripke polynomial endofunctor $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ in case the inductive definition of T incorporates that of F, i.e. in case F "occurs" in the definition of T. In contrast, the arrows of the category KP_1 (and indeed, KP_S, for an arbitrary S) capture *semantic* relationships between (the components of) Kripke polynomial endofunctors: any natural transformation $\eta : \mathsf{F} \Rightarrow \mathsf{G}$ also induces a mapping

from F-coalgebras to G-coalgebras, which takes an F-coalgebra $\langle C, \gamma\rangle$, with $\gamma : C \to \mathsf{F}C$, to the G-coalgebra $\langle C, \gamma'\rangle$, with $\gamma' : C \to \mathsf{G}C$ being given by $\eta_C \circ \gamma$. This observation will be exploited in Section 4, where an institution of many-sorted coalgebraic modal logics will be defined.

Example 4. Let A be a set (of labels), and let $\mathsf{T}_{\mathtt{LTS}} : \mathsf{Set} \to \mathsf{Set}$ be given by

$$\mathsf{T}_{\mathtt{LTS}} = \mathscr{P}^A = (\mathscr{P} \circ \mathsf{Id})^A$$

Then, any $\mathsf{T}_{\mathtt{LTS}}$-coalgebra $\langle S, next\rangle$ defines an A-labelled transition system with states S and transition relations $R_a \subseteq S \times S$ given by

$$s\ R_a\ t \ \text{ iff } \ next(s)(a) \ni t$$

for $s, t \in S$ and $a \in A$. Conversely, any A-labelled transition system defines a $\mathsf{T}_{\mathtt{LTS}}$-coalgebra.

Example 5. Lists whose elements belong to a set E can be specified using the endofunctor $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ given by

$$\mathsf{T}_{\mathtt{LIST}} = (1 + E) \times (1 + \mathsf{Id})$$

Then, a $\mathsf{T}_{\mathtt{LIST}}$-coalgebra defines a set L (of lists) together with a pair of functions $\langle hd, tl\rangle : L \to (1 + E) \times (1 + L)$ (defining the head and the tail of each list). Not any such coalgebra, however, provides a meaningful implementation of lists; for this, one has to additionally require that the head and tail of a list are either both undefined (i.e. equal to $\kappa_1(*)$) or both defined.

Example 6. A specification of finite lists which exploits the previous specification of lists can be given using the endofunctor $\mathsf{T}_{\mathtt{FLIST}} : \mathsf{Set} \to \mathsf{Set}$ defined by

$$\mathsf{T}_{\mathtt{FLIST}} = \mathsf{T}_{\mathtt{LIST}} \times \mathbb{N}$$

In addition to the structure required of $\mathsf{T}_{\mathtt{LIST}}$-coalgebras, $\mathsf{T}_{\mathtt{FLIST}}$-coalgebras also have to provide a function $len : L \to \mathbb{N}$ (defining the length of each list). Again, not any coalgebra of this endofunctor provides a meaningful implementation of finite lists; for this, one has to also require that the length of a list is consistent with the information provided by the tail operation.

Example 7. A specification of arrays of size m can be obtained by reusing the specification of lists given in Example 5. Specifically, one can consider the endofunctor $\mathsf{T} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}}$ whose two components are given by

$$\begin{aligned}\mathsf{T}_{\mathtt{mList}} &= (\mathsf{T}_{\mathtt{LIST}}\Pi_{\mathtt{mList}} \times \{0, \ldots, m\}) \times (1 + E)^{\{1,\ldots,m\}}\\ \mathsf{T}_{\mathtt{Array}} &= \Pi_{\mathtt{mList}} \times E^{\{1,\ldots,m\}}\end{aligned}$$

These components correspond to the types of lists of length not exceeding m, and respectively of arrays of size m (with lists of length m being used to implement such

arrays). In defining $\mathsf{T}_{\mathtt{mList}}$, the projection functor $\Pi_{\mathtt{mList}} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}$ is used to extract the $\mathtt{mList}$-sorted component of the (two-sorted) argument of T, to which the endofunctor specifying lists is then applied. The result of this application is $(1 + E) \times (1 + \Pi_{\mathtt{mList}})$. $\mathsf{T}_{\mathtt{mList}}$ also specifies a length, between 0 and m, for each list, as well as an observer for extracting the element situated in a given position (between 1 and m) in a list, in case such an element exists. Next, $\mathsf{T}_{\mathtt{Array}}$ specifies a list used to implement an array (again, by using $\Pi_{\mathtt{mList}}$ to extract the $\mathtt{mList}$-sorted component of the argument of T), as well as an observer for extracting the element with a given index in the array. A complete specification of arrays of size m will have to include a specification of lists as outlined in Example 5 (subject to a suitable translation), as well as to further constrain the length operation on lists, the additional list observer and the array observer.

In [8], the notion of ingredient is used to associate a modal language to each Kripke polynomial endofunctor on Set (by means of structural induction). Such modal languages are subsequently interpreted over coalgebras of the underlying Kripke polynomial endofunctors. The next definition generalises the notion of modal formula introduced in [8] to Kripke polynomial endofunctors on Set^S. Instantiating it to Kripke polynomial endofunctors on Set yields a definition equivalent to the one in [8], but which does not make use of ingredient functors. Similarly to [8], the resulting modal languages will later be interpreted over coalgebras of Kripke polynomial endofunctors on Set^S.

Definition 8. Let $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ denote a Kripke polynomial endofunctor. For $\mathsf{F} \in |\mathsf{KP}_S|$, the set $\mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ of **modal formulae over** T **of type** F is defined inductively (on the structure of F) as follows:

- $\bot \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$
- $(\varphi \to \psi) \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ if $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ and $\psi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$
- $d \in \mathsf{Form}_{\mathsf{T}}(D)$ if $d \in D$
- $[\mathsf{next}_s]\varphi \in \mathsf{Form}_{\mathsf{T}}(\Pi_s)$ if $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{T}_s)$, with $s \in S$
- $[\pi_i]\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F}_1 \times \mathsf{F}_2)$ if $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F}_i)$, with $i \in \{1, 2\}$
- $[\kappa_i]\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F}_1 + \mathsf{F}_2)$ if $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F}_i)$, with $i \in \{1, 2\}$
- $[ev(d)]\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F}^D)$ if $d \in D$ and $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$
- $[\mathscr{P}]\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathscr{P} \circ \mathsf{F})$ if $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$

Also, for $s \in S$, the set $\mathsf{SForm}(\mathsf{T})_s$ of **state formulae over** T **of type** s is given by $\mathsf{Form}_{\mathsf{T}}(\Pi_s)$.

If T is an endofunctor on Set and F is an ingredient of T (see [8]), then modal formulae over T of type F are essentially the same as modal formulae of *sort* F, as defined in [8] (w.r.t. T).[12] The above definition, however, differs from the one in [8] in that it makes the coalgebraic type of interest explicit. This will later allow us to consider semantic

[12] The modal logic defined in [8] is also qualified as *many-sorted*. However, in [8], sorts are used to refer to the ingredients of an endofunctor on Set, whereas here, many-sortedness is a feature of the underlying category, with sorts being used to denote the types of interest.

relationships between different coalgebraic types, and to lift such relationships to a logical level.

Remark 9. For a Kripke polynomial endofunctor $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$, one can also define:

- $\top ::= \bot \to \bot \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$
- $\neg\varphi ::= \varphi \to \bot \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$
- $\varphi \vee \psi ::= \neg\varphi \to \psi \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$
- $\varphi \wedge \psi ::= \neg(\varphi \to \neg\psi) \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$
- $\langle \mathsf{next}_s \rangle \varphi ::= \neg[\mathsf{next}_s]\neg\varphi \in \mathsf{Form}_\mathsf{T}(\Pi_s)$ for $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{T}_s)$ with $s \in S$
- $\langle \pi_i \rangle \varphi ::= \neg[\pi_i]\neg\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F}_1 \times \mathsf{F}_2)$ for $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F}_i)$ with $i \in \{1, 2\}$
- $\langle \kappa_i \rangle \varphi ::= \neg[\kappa_i]\neg\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F}_1 + \mathsf{F}_2)$ for $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F}_i)$ with $i \in \{1, 2\}$
- $\langle ev(d) \rangle \varphi ::= \neg[ev(d)]\neg\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F}^D)$ for $d \in D$ and $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$
- $\langle \mathscr{P} \rangle \varphi ::= \neg[\mathscr{P}]\neg\varphi \in \mathsf{Form}_\mathsf{T}(\mathscr{P} \circ \mathsf{F})$ for $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$

(Similar modal operators are defined in [16] in a one-sorted setting.)

Example 10. Let $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ be as in Example 5. Then, one can successively infer:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(1)}{\langle\kappa_1\rangle\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(1 + E)}}{[\pi_1]\langle\kappa_1\rangle\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(\mathsf{T}_{\mathtt{LIST}})}}{[\mathsf{next}][\pi_1]\langle\kappa_1\rangle\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(\mathsf{Id})}}{\langle\kappa_2\rangle[\mathsf{next}][\pi_1]\langle\kappa_1\rangle\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(1 + \mathsf{Id})}}{[\pi_2]\langle\kappa_2\rangle[\mathsf{next}][\pi_1]\langle\kappa_1\rangle\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(\mathsf{T}_{\mathtt{LIST}})}}{[\mathsf{next}][\pi_2]\langle\kappa_2\rangle[\mathsf{next}][\pi_1]\langle\kappa_1\rangle\top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(\mathsf{Id})}$$

Example 11. Let $\mathsf{T} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}}$ be as in Example 7. Then, one can successively infer:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{m \in \mathsf{Form}_\mathsf{T}(\{0, \ldots, m\})}{[\pi_2]m \in \mathsf{Form}_\mathsf{T}(\mathsf{T}_{\mathtt{LIST}}\Pi_{\mathtt{mList}} \times \{0, \ldots, m\})}}{[\pi_1][\pi_2]m \in \mathsf{Form}_\mathsf{T}(\mathsf{T}_{\mathtt{mList}})}}{[\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m \in \mathsf{Form}_\mathsf{T}(\Pi_{\mathtt{mList}})}}{[\pi_1][\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m \in \mathsf{Form}_\mathsf{T}(\mathsf{T}_{\mathtt{Array}})}}{[\mathsf{next}_{\mathtt{Array}}][\pi_1][\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m \in \mathsf{Form}_\mathsf{T}(\Pi_{\mathtt{Array}})}$$

The formulae which interest us are the state formulae, defined as formulae of projection type (i.e. Π_s with $s \in S$). They refer to the states of coalgebras, and are to be interpreted as subsets of the carriers of coalgebras. The definition of such interpretations follows the structure of the corresponding components (i.e. T_s with $s \in S$).

Definition 12. Let $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ denote a Kripke polynomial endofunctor, and let $\langle C, \gamma \rangle$ denote a T-coalgebra. For $\mathsf{F} \in |\mathsf{KP}_S|$, the **interpretation**[13] $[\![\varphi]\!]^\gamma_{\mathsf{F}} \in \mathscr{P}(\mathsf{F}C)$ of a modal formula $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ in the coalgebra $\langle C, \gamma \rangle$ is defined inductively (on the structure of φ and F) as follows:

- $[\![\bot]\!]^\gamma_{\mathsf{F}} = \emptyset$
- $[\![\varphi \to \psi]\!]^\gamma_{\mathsf{F}} = \overline{[\![\varphi]\!]^\gamma_{\mathsf{F}}} \cup [\![\psi]\!]^\gamma_{\mathsf{F}}$[14]
- $[\![d]\!]^\gamma_D = \{d\}$ with $d \in D$
- $[\![[\mathsf{next}_s]\varphi]\!]^\gamma_{\Pi_s} = \gamma_s^{-1}([\![\varphi]\!]^\gamma_{\mathsf{T}_s})$ with $s \in S$
- $[\![[\pi_i]\varphi]\!]^\gamma_{\mathsf{F}_1 \times \mathsf{F}_2} = \pi_i^{-1}([\![\varphi]\!]^\gamma_{\mathsf{F}_i})$ with $i \in \{1, 2\}$
- $[\![[\kappa_i]\varphi]\!]^\gamma_{\mathsf{F}_1 + \mathsf{F}_2} = \kappa_i([\![\varphi]\!]^\gamma_{\mathsf{F}_i}) \cup \kappa_j(\mathsf{F}_j C)$ with $i \in \{1, 2\}$ and $\{j\} = \{1, 2\} \setminus \{i\}$
- $[\![[ev(d)]\varphi]\!]^\gamma_{\mathsf{F}D} = \{ f : D \to \mathsf{F}C \mid f(d) \in [\![\varphi]\!]^\gamma_{\mathsf{F}} \}$ with $d \in D$
- $[\![[\mathscr{P}]\varphi]\!]^\gamma_{\mathscr{P}\circ\mathsf{F}} = \mathscr{P}([\![\varphi]\!]^\gamma_{\mathsf{F}})$

An element $c \in \mathsf{F}C$ is said to **satisfy** a modal formula $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ (written $c \models \varphi$) if and only if $c \in [\![\varphi]\!]^\gamma_{\mathsf{F}}$. Also, the coalgebra $\langle C, \gamma \rangle$ is said to **satisfy** the modal formula φ (written $\langle C, \gamma \rangle \models \varphi$) if and only if $[\![\varphi]\!]^\gamma_{\mathsf{F}} = \mathsf{F}C$. In particular, given $s \in S$, an element $c \in C_s$ is said to **satisfy** a state formula $\varphi \in \mathsf{SForm}(\mathsf{T})_s$ if and only if $c \in [\![\varphi]\!]^\gamma_{\Pi_s}$, while the coalgebra $\langle C, \gamma \rangle$ is said to **satisfy** the state formula φ if and only if $[\![\varphi]\!]^\gamma_{\Pi_s} = C_s$. Two modal formulae $\varphi, \psi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ are said to be **semantically equivalent** (written $\varphi \equiv \psi$) if and only if $[\![\varphi]\!]^\gamma_{\mathsf{F}} = [\![\psi]\!]^\gamma_{\mathsf{F}}$ for any T-coalgebra $\langle C, \gamma \rangle$.

(The above definition generalises a similar definition in [8] to Kripke polynomial endofunctors on sorted sets.)

Remark 13. The following are consequences of Remark 9 and Definition 12:

- $[\![\top]\!]^\gamma_{\mathsf{F}} = \mathsf{F}C$
- $[\![\neg\varphi]\!]^\gamma_{\mathsf{F}} = \overline{[\![\varphi]\!]^\gamma_{\mathsf{F}}}$
- $[\![\varphi \vee \psi]\!]^\gamma_{\mathsf{F}} = [\![\varphi]\!]^\gamma_{\mathsf{F}} \cup [\![\psi]\!]^\gamma_{\mathsf{F}}$
- $[\![\varphi \wedge \psi]\!]^\gamma_{\mathsf{F}} = [\![\varphi]\!]^\gamma_{\mathsf{F}} \cap [\![\psi]\!]^\gamma_{\mathsf{F}}$
- $[\![\langle \mathsf{next}_s \rangle \varphi]\!]^\gamma_{\Pi_s} = [\![[\mathsf{next}_s]\varphi]\!]^\gamma_{\Pi_s}$
- $[\![\langle \pi_i \rangle \varphi]\!]^\gamma_{\mathsf{F}_1 \times \mathsf{F}_2} = [\![[\pi_i]\varphi]\!]^\gamma_{\mathsf{F}_1 \times \mathsf{F}_2}$
- $[\![\langle ev(d) \rangle \varphi]\!]^\gamma_{\mathsf{F}D} = [\![[ev(d)]\varphi]\!]^\gamma_{\mathsf{F}D}$
- $[\![\langle \kappa_i \rangle \varphi]\!]^\gamma_{\mathsf{F}_1 + \mathsf{F}_2} = \kappa_i([\![\varphi]\!]^\gamma_{\mathsf{F}_1}) \subsetneq \kappa_i([\![\varphi]\!]^\gamma_{\mathsf{F}_1}) \cup \kappa_j(\mathsf{F}_2 C) = [\![[\kappa_i]\varphi]\!]^\gamma_{\mathsf{F}_1 + \mathsf{F}_2}$
- $[\![\langle \mathscr{P} \rangle \varphi]\!]^\gamma_{\mathscr{P}\circ\mathsf{F}} = \{ X \subseteq \mathsf{F}C \mid X \cap [\![\varphi]\!]^\gamma_{\mathsf{F}} \neq \emptyset \} \neq \mathscr{P}([\![\varphi]\!]^\gamma_{\mathsf{F}}) = [\![[\mathscr{P}]\varphi]\!]^\gamma_{\mathscr{P}\circ\mathsf{F}}$

Example 14. Let $\mathsf{T}_{\mathtt{LTS}} : \mathsf{Set} \to \mathsf{Set}$ be as in Example 4. Then, the formulae of Hennessy-Milner logic are essentially a subset of $\mathsf{SForm}(\mathsf{T}_{\mathtt{LTS}})$. For, one can successively infer:

[13] The definition of $[\![\varphi]\!]^\gamma_{\mathsf{F}}$ also depends on T. However, to keep the notation as simple as possible, this dependency is not reflected in the notation.

[14] For $X \in \mathscr{P}(\mathsf{F}C)$, $\overline{X}$ is given by $\mathsf{F}C \setminus X$.

$$\dfrac{\dfrac{\dfrac{\varphi \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LTS}}}(\mathsf{Id})}{[\mathscr{P}]\varphi \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LTS}}}(\mathscr{P} \circ \mathsf{Id})}}{[ev(a)][\mathscr{P}]\varphi \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LTS}}}((\mathscr{P} \circ \mathsf{Id})^A)}}{[\mathsf{next}][ev(a)][\mathscr{P}]\varphi \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LTS}}}(\mathsf{Id})}$$

Thus, the modal operators of Hennessy-Milner logic can be recovered by letting $\langle a \rangle ::= [\mathsf{next}][ev(a)][\mathscr{P}]$ for $a \in A$. Moreover, the interpretations of state formulae of form $\langle a \rangle \varphi$ over $\mathsf{T}_{\mathtt{LTS}}$-coalgebras coincide with the standard interpretations of such formulae over the corresponding transition systems:

$$s \models \langle a \rangle \varphi \;\Leftrightarrow\; \exists\, t \in next(s)(a).\, t \models \varphi$$

for any $\mathsf{T}_{\mathtt{LTS}}$-coalgebra $next : S \to \mathscr{P}(S)^A$, $s \in S$ and $a \in A$.

Example 15. Let $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ be as in Example 10. Also, let $\gamma = \langle hd, tl \rangle : L \to (1 + E) \times (1 + L)$ denote a $\mathsf{T}_{\mathtt{LIST}}$-coalgebra. One can then successively infer:

$$\dfrac{\dfrac{\dfrac{[\![\top]\!]^{\gamma}_{1} = 1}{[\![\langle \kappa_1 \rangle \top]\!]^{\gamma}_{1+L} = \kappa_1(1)}}{[\![[\pi_2]\langle \kappa_1 \rangle \top]\!]^{\gamma}_{\mathsf{T}_{\mathtt{LIST}}} = (1 + E) \times \kappa_1(1)}}{[\![[\mathsf{next}][\pi_2]\langle \kappa_1 \rangle \top]\!]^{\gamma}_{\mathsf{Id}} = \gamma^{-1}((1 + E) \times \kappa_1(1)) \;= tl^{-1}(\kappa_1(1))}$$

Thus, the formula $[\mathsf{next}][\pi_2]\langle \kappa_1 \rangle \top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(\mathsf{Id})$ holds in precisely those states $l \in L$ on which the tail operation $tl : L \to 1 + L$ yields an undefined result. Similarly, the formula $[\mathsf{next}][\pi_1]\langle \kappa_1 \rangle \top \in \mathsf{Form}_{\mathsf{T}_{\mathtt{LIST}}}(\mathsf{Id})$ holds in precisely those states $l \in L$ on which the head operation $hd : L \to 1 + E$ yields an undefined result. The following modal formula now completes the specification of lists:

$$[\mathsf{next}][\pi_1]\langle \kappa_1 \rangle \top \leftrightarrow [\mathsf{next}][\pi_2]\langle \kappa_1 \rangle \top$$

After renaming $[\mathsf{next}][\pi_1]\langle \kappa_1 \rangle$ to `<hdU>` and $[\mathsf{next}][\pi_2]\langle \kappa_1 \rangle$ to `<tlU>`, this formula becomes:

$$\texttt{<hdU>}\top \leftrightarrow \texttt{<tlU>}\top$$

where

$$l \models \texttt{<hdU>}\varphi_1 \;\Leftrightarrow\; \exists\, s\,.\, hd(l) = \kappa_1(s) \text{ and } s \models \varphi_1$$
$$l \models \texttt{<tlU>}\varphi_1 \;\Leftrightarrow\; \exists\, s\,.\, tl(l) = \kappa_1(s) \text{ and } s \models \varphi_1$$

for any $\mathsf{T}_{\mathtt{LIST}}$-coalgebra $\langle hd, tl \rangle : L \to (1 + E) \times (1 + L)$ and any $l \in L$. Thus, the specification of lists formalises the observation that the head and tail of a list are either both undefined or both defined.

Example 16. Let $\mathsf{T} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}}$ be as in Example 11. Also, let $\gamma = (\gamma_{\mathtt{mList}}, \gamma_{\mathtt{Array}}) : (L, A) \to (\mathsf{T}_{\mathtt{mList}}(L, A), \mathsf{T}_{\mathtt{Array}}(L, A))$ denote a T-coalgebra. One can then successively infer:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[\![m]\!]^{\gamma}_{\{0,\ldots,m\}} = \{m\}}}{[\![[\pi_2]m]\!]^{\gamma}_{\mathsf{T}_{\mathtt{LIST}}\Pi_{\mathtt{mList}}\times\{0,\ldots,m\}} = \pi_2^{-1}(\{m\})}}{[\![[\pi_1][\pi_2]m]\!]^{\gamma}_{\mathsf{T}_{\mathtt{mList}}} = \pi_1^{-1}(\pi_2^{-1}(\{m\}))}}{[\![[\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m]\!]^{\gamma}_{\Pi_{\mathtt{mList}}} = \gamma^{-1}_{\mathtt{mList}}(\pi_1^{-1}(\pi_2^{-1}(\{m\})))}}{[\![[\pi_1][\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m]\!]^{\gamma}_{\mathsf{T}_{\mathtt{Array}}} = \pi_1^{-1}(\gamma^{-1}_{\mathtt{mList}}(\pi_1^{-1}(\pi_2^{-1}(\{m\}))))}}{[\![[\mathsf{next}_{\mathtt{Array}}][\pi_1][\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m]\!]^{\gamma}_{\Pi_{\mathtt{Array}}} = \gamma^{-1}_{\mathtt{Array}}(\pi_1^{-1}(\gamma^{-1}_{\mathtt{mList}}(\pi_1^{-1}(\pi_2^{-1}(\{m\})))))}$$

That is, a state $a \in A$ satisfies the formula $[\mathsf{next}_{\mathtt{Array}}][\pi_1][\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m$ precisely when $\pi_2(\pi_1(\gamma_{\mathtt{mList}}(\pi_1(\gamma_{\mathtt{Array}}(a))))) = m$, i.e. precisely when the length of the list used to implement the array a is m.

It is shown in [16] that the modal logics defined there for *finite* Kripke polynomial endofunctors on Set *capture bisimulation*, that is, the logical equivalence relation between states coincides with the bisimilarity relation. The proof of this result uses an alternative definition of the notion of bisimulation induced by Kripke polynomial endofunctors on Set. Both the result and its proof generalise to Kripke polynomial endofunctors on Set^S. The remainder of this section briefly sketches this generalisation.

Definition 17. Let $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ denote a Kripke polynomial endofunctor, and let $\langle C, \gamma\rangle$ and $\langle D, \delta\rangle$ be T-coalgebras. For $s \in S$, two states $c \in C_s$ and $d \in D_s$ are said to be **logically equivalent** (written $c \approx d$) if for all $\varphi \in \mathsf{SForm}(\mathsf{T})_s$, $c \models \varphi$ if and only if $d \models \varphi$.

Definition 18. Let $C, D \in |\mathsf{Set}^S|$, and let R denote a relation between C and D. For $\mathsf{F} : \mathsf{Set}^S \to \mathsf{Set}$ a finite Kripke polynomial functor, the **F-lifting of** R[15], denoted R_{F}, is a relation between $\mathsf{F}C$ and $\mathsf{F}D$ defined inductively by

- $R_D = \Delta_D$ for $D \in |\mathsf{Set}|$ finite and non-empty
- $R_{\Pi_s} = R_s$ for $s \in S$
- $R_{\mathsf{F}_1\times\mathsf{F}_2} = \{\, \langle s, t\rangle \mid \pi_i(s)\, R_{\mathsf{F}_i}\, \pi_i(t) \text{ for } i = 1, 2\,\}$
- $R_{\mathsf{F}_1+\mathsf{F}_2} = \{\, \langle \kappa_1(s), \kappa_1(t)\rangle \mid s\, R_{\mathsf{F}_1}\, t\,\} \cup \{\, \langle \kappa_2(s), \kappa_2(t)\rangle \mid s\, R_{\mathsf{F}_2}\, t\,\}$
- $R_{\mathsf{F}^D} = \{\, \langle f, g\rangle \mid f(d)\, R_{\mathsf{F}}\, g(d) \text{ for all } d \in D\,\}$
- $\mathsf{R}_{\mathscr{P}_\omega(\mathsf{F})} = \{\, \langle S, T\rangle \mid \forall s \in S\, \exists t \in T\, .\, s\, R_{\mathsf{F}}\, t \text{ and } \forall t \in T\, \exists s \in S\, .\, s\, R_{\mathsf{F}}\, t\,\}$

Now given a finite Kripke polynomial endofunctor $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$, the **$\mathsf{T}$-lifting of** R, denoted R_{T}, is the (S-sorted) relation between $\mathsf{T}C$ and $\mathsf{T}D$ whose components are given by $(R_{\mathsf{T}})_s = R_{\mathsf{T}_s}$ for $s \in S$.

Lemma 19. *Let $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ denote a finite Kripke polynomial endofunctor, and let $\langle C, \gamma\rangle$ and $\langle D, \delta\rangle$ be T-coalgebras. An S-sorted relation R between C and D is a T-bisimulation if and only if, for all $s \in S$ and all $c \in C_s$ and $d \in D_s$, cRd implies $\gamma_s(c) R_{\mathsf{T}_s} \delta_s(d)$.*

[15] Such liftings have also been defined in [7,16,8], in a one-sorted setting.

Proof (*Sketch*). The statement follows easily from the definitions of R_{T_s} and of the notion of bisimulation. □

The result in [16] now generalises to endofunctors on Set^S.

Proposition 20. *Let* $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ *denote a finite Kripke polynomial endofunctor, and let* $\langle C, \gamma\rangle$ *and* $\langle D, \delta\rangle$ *be* T*-coalgebras. Then, given* $c \in C_s$ *and* $d \in D_s$ *with* $s \in S$, $c \approx d$ *if and only if* $c \sim d$. □

Proof (*Sketch*). Similar to the proof of [16, Proposition 4.8]. Specifically, the "if" direction uses Lemma 19 along a structural induction on T_s, whereas the "only if" direction uses structural induction on T_s to define a formula $\varphi \in \mathsf{Form}_{\mathsf{T}}(\Pi_s)$ which holds in c but not in d, whenever $c \not\sim d$. □

4. An institution of modal logics

The arrows of the category KP_S capture semantic relationships between (the components of) Kripke polynomial endofunctors. In the following, such arrows will be shown to induce translations between the logics associated to different endofunctors, in such a way that the satisfaction of modal formulae by coalgebras is preserved and reflected along the induced translations. This approach will provide support for modular specification, by allowing specifications and their *global semantic consequences*[16] to be carried over from simpler coalgebraic types to more complex ones.

Collections of (interdependent) coalgebraic types are specified using *many-sorted cosignatures*, whereas semantic relationships between different such collections are specified using *many-sorted cosignature morphisms*.

Definition 21. A **many-sorted cosignature** is a pair (S, T) with S a set and $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ a Kripke polynomial endofunctor. A **many-sorted cosignature morphism** from (S, T) to (S', T') is a pair (f, η) with $f : S \to S'$ and $\eta : \mathsf{UT}' \Rightarrow \mathsf{TU}$,[17] such that $\Pi_s\eta \in \|\mathsf{KP}_{S'}\|$ for each $s \in S$. The category of many-sorted cosignatures and many-sorted cosignature morphisms is denoted Cosign.

The endofunctor $\mathsf{U} : \mathsf{Set}^{S'} \to \mathsf{Set}^S$ satisfies $\Pi_s\mathsf{U} = \Pi_{f(s)}$ for each $s \in S$. As a result, the natural transformation $\Pi_s\eta$ is of form $\eta_s : \mathsf{T}'_{f(s)} \Rightarrow \mathsf{T}_s\mathsf{U}$, for each $s \in S$. Such a natural transformation specifies a semantic relationship between the coalgebraic structure specified by T for the type denoted by s and the coalgebraic structure specified by T' for the type denoted by $f(s)$. Specifically, if we regard $\mathsf{T}'_{f(s)}$ as defining some additional structure to the structure specified by T_s for the sort s, then η_s shows how the T_s-structure can be retrieved from the $\mathsf{T}'_{f(s)}$-structure.

[16] A formula φ is a *global semantic consequence* of a set Φ of formulae if $\langle C, \gamma\rangle \models \varphi$ holds whenever $\langle C, \gamma\rangle \models \Phi$, for any T-coalgebra $\langle C, \gamma\rangle$. On the other hand, φ is a *local semantic consequence* of Φ if $c \models \Phi$ implies $c \models \varphi$, for any T-coalgebra $\langle C, \gamma\rangle$ and any $c \in C_s$, with $s \in S$ denoting the type of φ.

[17] Here, $\mathsf{U} : \mathsf{Set}^{S'} \to \mathsf{Set}^S$ denotes the functor taking an S'-sorted set C' (S'-sorted function g') to the S-sorted set C (S-sorted function g) given by $C_s = C'_{f(s)}$ ($g_s = g'_{f(s)}$) for $s \in S$.

In case η_s is the identity natural transformation on $\mathsf{T}_s\mathsf{U}$, we say that the type s is *encapsulated along* η. Otherwise, we say that s is *refined along* η.

It follows from Definition 1 (see also Remark 2) that each component $\Pi_s\eta$ of the natural transformation η defining a many-sorted cosignature morphism is constructed from natural transformations of form $1_{\mathsf{F}} : \mathsf{F} \Rightarrow \mathsf{F}$, $\alpha : D' \Rightarrow D$, $d : \mathsf{F} \Rightarrow D$, $\pi_i : \mathsf{F}_1 \times \mathsf{F}_2 \Rightarrow \mathsf{F}_i$, $\kappa_i : \mathsf{F}_i \Rightarrow \mathsf{F}_1 + \mathsf{F}_2$ and $eval_{\mathsf{F},D} : \mathsf{F}^D \times D \Rightarrow \mathsf{F}$, using pairing $\langle \eta_1, \eta_2 \rangle$, co-pairing $[\eta_1, \eta_2]$, currying η^*, direct image $\mathscr{P}(\eta)$ and horizontal composition. This will allow us to use induction to define translations of modal formulae along many-sorted cosignature morphisms.

Example 22. Let $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ and $\mathsf{T}_{\mathtt{FLIST}} : \mathsf{Set} \to \mathsf{Set}$ be as in Examples 5 and respectively 6. Then, the natural transformation $\eta_1 ::= \pi_1 : \mathsf{T}_{\mathtt{FLIST}} \Rightarrow \mathsf{T}_{\mathtt{LIST}}$ defines a cosignature morphism $(1_1, \eta_1) : (1, \mathsf{T}_{\mathtt{LIST}}) \to (1, \mathsf{T}_{\mathtt{FLIST}})$. This cosignature morphism refines the type of lists to finite lists.

Example 23. Let $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ and $\mathsf{T}_{\mathtt{FLIST}} : \mathsf{Set} \to \mathsf{Set}$ be as before, and let $\mathsf{T} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}}$ be as in Example 7. Also, let $\mathsf{U} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}$ be given by $\Pi_{\mathtt{mList}}$. Then, one can define a cosignature morphism $(f, \eta_2) : (\{\mathtt{FList}\}, \mathsf{T}_{\mathtt{FLIST}}) \to (\{\mathtt{mList}, \mathtt{Array}\}, \mathsf{T})$ by letting $f : \{\mathtt{FList}\} \to \{\mathtt{mList}, \mathtt{Array}\}$ take the sort $\mathtt{FList}$ to the sort $\mathtt{mList}$, and letting $\eta_2 : \mathsf{UT} \Rightarrow \mathsf{T}_{\mathtt{FLIST}}\mathsf{U}$ be given by $(1 \times \iota) \circ \pi_1$:

$$\mathsf{UT} = \mathsf{T}_{\mathtt{mList}} \overset{\pi_1}{\Longrightarrow} \mathsf{T}_{\mathtt{LIST}}\Pi_{\mathtt{mList}} \times \{0, \ldots, m\} \overset{1\times\iota}{\Longrightarrow} \mathsf{T}_{\mathtt{LIST}}\Pi_{\mathtt{mList}} \times \mathbb{N} = \mathsf{T}_{\mathtt{FLIST}}\mathsf{U}$$

where $\iota : \{0, \ldots, m\} \to \mathbb{N}$ is the canonical inclusion.

Remark 24. More general notions of morphisms between coalgebraic signatures have been defined e.g. in [2] or [11]. In [2, Section 3.1], a **cosignature** was defined as a pair (C, T) with C a category and $\mathsf{T} : \mathsf{C} \to \mathsf{C}$ an endofunctor, whereas a **morphism between cosignatures** (C, T) and $(\mathsf{D}, \mathsf{T}')$ was defined as a pair (U, η) with $\mathsf{U} : \mathsf{D} \to \mathsf{C}$ a limit-preserving functor which admits a right adjoint, and with $\eta : \mathsf{UT}' \Rightarrow \mathsf{TU}$ a natural transformation. A similar definition was given in [11, Section 4.2], only there, no restrictions on the functor U were imposed. The notion of many-sorted cosignature morphism considered here is an instance of either of these notions (with C, D, U and η all taking specific forms).

Many-sorted cosignature morphisms $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$ induce *reduct functors* $\mathsf{U}_\eta : \mathsf{Coalg}(\mathsf{T}') \to \mathsf{Coalg}(\mathsf{T})$, with U_η taking a T'-coalgebra $\langle C', \gamma' \rangle$ to the T-coalgebra $\langle \mathsf{U}C', \eta_{C'} \circ \mathsf{U}\gamma' \rangle$. This yields a functor $\mathsf{Coalg} : \mathsf{Cosign} \to \mathsf{Cat}^{\mathsf{op}}$, taking a many-sorted cosignature to its category of coalgebras, and a many-sorted cosignature morphism to the induced reduct functor.

Next, we show that many-sorted cosignature morphisms induce translations of state formulae over their domain to state formulae over their codomain. The definition of such translations mirrors the definition of state formulae over a Kripke polynomial endofunctor: in the same way as defining state formulae over a Kripke polynomial endofunctor T involved first defining modal formulae over T *of arbitrary type* F, and then instantiating F with Π_s, defining a translation of state formulae over T along a many-sorted cosignature morphism $\eta : (S, \mathsf{T}) \to (S', \mathsf{T}')$ will involve first defining translations (w.r.t. η) of modal formulae over T *of arbitrary type* F *along arbitrary natural transformations* $\tau : \mathsf{F}' \Rightarrow \mathsf{FU}$, and then instantiating τ with $1_{\Pi_{f(s)}} : \Pi_{f(s)} \Rightarrow \Pi_s\mathsf{U}$. The resulting translations will, in general, depend not only on the natural transformation τ but also on the underlying natural transformation η.

Consequently, translations along identity natural transformations τ will not leave modal formulae unchanged, unless the underlying η is itself an identity natural transformation.

For a particular natural transformation τ, the definition of the translation along τ (w.r.t. a fixed η) is driven by the need to ensure that the interpretations of formulae are preserved along the translation. This will later allow us to prove that the resulting logical framework is an institution.

Definition 25. Let $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$ denote a many-sorted cosignature morphism. For $\mathsf{F} \in |\mathsf{KP}_S|$, $\mathsf{F}' \in |\mathsf{KP}_{S'}|$ and $(\tau : \mathsf{F}' \Rightarrow \mathsf{FU}) \in \|\mathsf{KP}_{S'}\|$[18], the **translation along τ w.r.t. η** of modal formulae φ over T of type F to modal formulae over T' of type F' is defined inductively (on the structure of φ and τ) as follows:

(1) (a) $\bot \overset{\tau_\eta}{\longmapsto} \bot$

(b) $(\varphi \to \psi) \overset{\tau_\eta}{\longmapsto} (\varphi' \to \psi')$ if $\varphi \overset{\tau_\eta}{\longmapsto} \varphi'$ and $\psi \overset{\tau_\eta}{\longmapsto} \psi'$

(2) If τ is given by an identity natural transformation, the following subcases can be distinguished:

(a) If τ is given by $1_{D\mathsf{U}} : D = D\mathsf{U} \Rightarrow D\mathsf{U}$:

$$d \overset{(1_{D\mathsf{U}})_\eta}{\longmapsto} d$$

(b) If τ is given by $1_{\Pi_{f(s)}} : \Pi_{f(s)} \Rightarrow \Pi_{f(s)} = \Pi_s\mathsf{U}$ with $s \in S$:

$$[\mathsf{next}_s]\varphi \overset{(1_{\Pi_{f(s)}})_\eta}{\longmapsto} [\mathsf{next}_{f(s)}]\varphi' \text{ if } \varphi \overset{(\eta_s)_\eta}{\longmapsto} \varphi'$$

where $\eta_s : \mathsf{T}'_{f(s)} \Rightarrow \mathsf{T}_s\mathsf{U}$.

(c) If τ is given by $1_{\mathsf{F}_1\mathsf{U}\times\mathsf{F}_2\mathsf{U}} : \mathsf{F}_1\mathsf{U} \times \mathsf{F}_2\mathsf{U} \Rightarrow \mathsf{F}_1\mathsf{U} \times \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 \times \mathsf{F}_2)\mathsf{U}$:

$$[\pi_i]\varphi \overset{(1_{\mathsf{F}_1\mathsf{U}\times\mathsf{F}_2\mathsf{U}})_\eta}{\longmapsto} [\pi_i]\varphi' \text{ if } \varphi \overset{(1_{\mathsf{F}_i\mathsf{U}})_\eta}{\longmapsto} \varphi', \text{ with } i \in \{1, 2\}$$

(d) If τ is given by $1_{\mathsf{F}_1\mathsf{U}+\mathsf{F}_2\mathsf{U}} : \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U} \Rightarrow \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 + \mathsf{F}_2)\mathsf{U}$:

$$[\kappa_i]\varphi \overset{(1_{\mathsf{F}_1\mathsf{U}+\mathsf{F}_2\mathsf{U}})_\eta}{\longmapsto} [\kappa_i]\varphi' \text{ if } \varphi \overset{(1_{\mathsf{F}_i\mathsf{U}})_\eta}{\longmapsto} \varphi', \text{ with } i \in \{1, 2\}$$

(e) If τ is given by $1_{(\mathsf{FU})^D} : (\mathsf{FU})^D \Rightarrow (\mathsf{FU})^D = \mathsf{F}^D\mathsf{U}$:

$$[ev(d)]\varphi \overset{(1_{(\mathsf{FU})^D})_\eta}{\longmapsto} [ev(d)]\varphi' \text{ if } \varphi \overset{(1_{\mathsf{FU}})_\eta}{\longmapsto} \varphi'$$

(f) If τ is given by $1_{\mathscr{P}\mathsf{FU}} : \mathscr{P} \circ (\mathsf{FU}) \Rightarrow \mathscr{P} \circ (\mathsf{FU}) = (\mathscr{P} \circ \mathsf{F})\mathsf{U}$:

$$[\mathscr{P}]\varphi \overset{(1_{\mathscr{P}\mathsf{FU}})_\eta}{\longmapsto} [\mathscr{P}]\varphi' \text{ if } \varphi \overset{(1_{\mathsf{FU}})_\eta}{\longmapsto} \varphi'$$

(3) (a) If τ is given by $\alpha : D' \Rightarrow D = D\mathsf{U}$[19]:

$$d \overset{\alpha_\eta}{\longmapsto} \bigvee_{\alpha(d')=d} d'$$

(b) If τ is given by $d : \mathsf{F} \Rightarrow D = D\mathsf{U}$:

$$d' \overset{d_\eta}{\longmapsto} \begin{cases} \top & \text{if } d' = d \\ \bot & \text{if } d' \neq d \end{cases}$$

(c) If τ is given by $\pi_i : \mathsf{F}'_1 \times \mathsf{F}'_2 \Rightarrow \mathsf{F}_i\mathsf{U}$ with $i \in \{1, 2\}$ and with $\mathsf{F}'_i = \mathsf{F}_i\mathsf{U}$:

$$\varphi \overset{(\pi_i)_\eta}{\longmapsto} [\pi_i]\varphi' \text{ if } \varphi \overset{(1_{\mathsf{F}_i\mathsf{U}})_\eta}{\longmapsto} \varphi'$$

[18] Note that $\mathsf{F} \in |\mathsf{KP}_S|$ implies $\mathsf{FU} \in |\mathsf{KP}_{S'}|$. This follows from $\Pi_s\mathsf{U} = \Pi_{f(s)}$ for any $s \in S$.

[19] Here it is essential that the sets $\alpha^{-1}(d)$ with $d \in D$ be finite.

(d) If τ is given by $\langle \tau_1, \tau_2 \rangle : \mathsf{F} \Rightarrow \mathsf{F}_1\mathsf{U} \times \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 \times \mathsf{F}_2)\mathsf{U}$ with $\tau_i : \mathsf{F} \Rightarrow \mathsf{F}_i\mathsf{U}$ for $i = 1, 2$:

$$[\pi_i]\varphi \overset{\langle \tau_1, \tau_2 \rangle_\eta}{\longmapsto} \varphi' \text{ if } \varphi \overset{(\tau_i)_\eta}{\longmapsto} \varphi' , \ i \in \{1, 2\}$$

(e) If τ is given by $\kappa_i : \mathsf{F}_i\mathsf{U} \Rightarrow \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 + \mathsf{F}_2)\mathsf{U}$ with $i \in \{1, 2\}$:

$$[\kappa_j]\varphi \overset{(\kappa_i)_\eta}{\longmapsto} \begin{cases} \varphi' & \text{if } j = i \text{ and } \varphi \overset{(1_{\mathsf{F}_i\mathsf{U}})_\eta}{\longmapsto} \varphi' \\ \top & \text{if } j \neq i \end{cases} , \ j \in \{1, 2\}$$

(f) If τ is given by $[\tau_1, \tau_2] : \mathsf{F}_1 + \mathsf{F}_2 \Rightarrow \mathsf{FU}$ with $\tau_i : \mathsf{F}_i \Rightarrow \mathsf{FU}$ for $i = 1, 2$:

$$\varphi \overset{[\tau_1, \tau_2]_\eta}{\longmapsto} [\kappa_1]\varphi_1 \wedge [\kappa_2]\varphi_2 \text{ if } \varphi \overset{(\tau_i)_\eta}{\longmapsto} \varphi_i \text{ for } i = 1, 2$$

(g) If τ is given by $eval_{\mathsf{FU}, D} : (\mathsf{FU})^D \times D \Rightarrow \mathsf{FU}$[20]:

$$\varphi \overset{(eval_{\mathsf{FU}, D})_\eta}{\longmapsto} \bigwedge_{d \in D} ([\pi_2]d \to [\pi_1][ev(d)]\varphi') \text{ if } \varphi \overset{(1_{\mathsf{FU}})_\eta}{\longmapsto} \varphi'$$

(h) If τ is given by $\zeta^* : \mathsf{F}' \Rightarrow (\mathsf{FU})^D = \mathsf{F}^D\mathsf{U}$ with $\zeta : \mathsf{F}' \times D \Rightarrow \mathsf{FU}$:

$$[ev(d)]\varphi \overset{(\zeta^*)_\eta}{\longmapsto} \varphi' \text{ if } \varphi \overset{\zeta_\eta}{\longmapsto} \varphi_1 \overset{\langle 1_{\mathsf{F}'} d \rangle 1_{\mathsf{T}'}}{\longmapsto} \varphi'$$

(i) If τ is given by $\mathscr{P}(\zeta) : \mathscr{P} \circ \mathsf{F}' \Rightarrow \mathscr{P} \circ (\mathsf{FU}) = (\mathscr{P} \circ \mathsf{F})\mathsf{U}$ with $\zeta : \mathsf{F}' \Rightarrow \mathsf{FU}$:

$$[\mathscr{P}]\varphi \overset{\mathscr{P}(\zeta)_\eta}{\longmapsto} [\mathscr{P}]\varphi' \text{ if } \varphi \overset{\zeta_\eta}{\longmapsto} \varphi'$$

(4) If τ is given by $\tau_1 \circ \tau_2 : \mathsf{F}' \Rightarrow \mathsf{FU}$, with $\tau_1 : \mathsf{F}_1 \Rightarrow \mathsf{FU}$ and $\tau_2 : \mathsf{F}' \Rightarrow \mathsf{F}_1$ in $\| \mathsf{KP}_{S'} \|$, and if τ_η has not yet been defined[21]:

$$\varphi \overset{(\tau_1 \circ \tau_2)_\eta}{\longmapsto} \varphi' \text{ if } \varphi \overset{(\tau_1)_\eta}{\longmapsto} \varphi_1 \text{ and } \varphi_1 \overset{(\tau_2)_{1_{\mathsf{T}'}}}{\longmapsto} \varphi'$$

Also, for $s \in S$, the **translation along** η of state formulae over T of type s to state formulae over T' of type $f(s)$, denoted $\eta_s : \mathsf{SForm}(\mathsf{T})_s \to \mathsf{SForm}(\mathsf{T}')_{f(s)}$, is given by $(1_{\Pi_{f(s)}})_\eta : \mathsf{Form}_{\mathsf{T}}(\Pi_s) \to \mathsf{Form}_{\mathsf{T}'}(\Pi_{f(s)})$ (where $1_{\Pi_{f(s)}} : \Pi_{f(s)} \Rightarrow \Pi_s\mathsf{U}$).

Thus, the boolean structure of formulae is always preserved by the translations (by (1) of Definition 25). In addition, translations between similar[22] types also preserve the modal structure of formulae (by (2) of Definition 25). Finally, in defining the translations induced by non-identity natural transformations τ (in (3) of Definition 25), all possible shapes for the formula being translated have to be considered. In particular:

- the translation of a formula of type F_1 along $\pi_1 : \mathsf{F}_1\mathsf{U} \times \mathsf{F}'_2 \Rightarrow \mathsf{F}_1\mathsf{U}$ requires the first component of a state satisfying it to satisfy the translation of the given formula along $1_{\mathsf{F}_1\mathsf{U}}$;
- the translation of a formula of type $\mathsf{F}_1 + \mathsf{F}_2$ along $\kappa_1 : \mathsf{F}_1\mathsf{U} \Rightarrow (\mathsf{F}_1 + \mathsf{F}_2)\mathsf{U}$ depends on which coproduct component the given formula refers to: if it refers to the first coproduct component, its translation requires whatever the original formula required of states coming from the first coproduct component, but translated along $1_{\mathsf{F}_1\mathsf{U}}$; and if the formula refers to the second coproduct component, its translation does not require anything;

[20] Here it is essential that the set D be finite.

[21] This condition ensures that τ_η is only defined *once*, by preventing the definition of τ_η to be based on equalities of form $\tau = \pi_1 \circ \langle \tau, \zeta \rangle$, $\tau = [\tau, \zeta] \circ \kappa_1$ or $\tau = eval_{\mathsf{F}, D} \circ (\tau^* \times 1_D)$.

[22] Similarity here refers to FU and F'.

- the translation of a formula of form $[ev(d)]\varphi$ along $\zeta^* : \mathsf{F}' \Rightarrow \mathsf{F}^D\mathsf{U}$ is obtained by first translating φ along $\zeta : \mathsf{F}' \times D \Rightarrow \mathsf{FU}$ to φ_1, and then "extracting" from φ_1 a formula φ' of type F', which holds in f' precisely when φ_1 holds in $\langle f', d\rangle$;
- the translation of a formula φ along $eval_{\mathsf{FU},D} : (\mathsf{FU})^D \times D \Rightarrow \mathsf{FU}$ holds in $\langle f, d\rangle$ precisely when the translation of φ along 1_{FU} holds in $f(d)$.

Finally, (4) of Definition 25 defines translations along compositions of natural transformations in terms of the translations along the natural transformations being composed. The next two results ensure the correctness of Definition 25.

Proposition 26. *Let* $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$ *denote a many-sorted cosignature morphism, and let* $(\tau : \mathsf{F}' \Rightarrow \mathsf{F}) \in \| \mathsf{KP}_S \|$[23]. *Then,* $(\tau_{\mathsf{U}})_{1_{\mathsf{T}'}} \circ (1_{\mathsf{FU}})_\eta = (\tau_{\mathsf{U}})_\eta = (1_{\mathsf{F}'\mathsf{U}})_\eta \circ \tau_{1_{\mathsf{T}}}$:

$$\begin{array}{ccc} \mathsf{Form}_{\mathsf{T}}(\mathsf{F}) & \xrightarrow{(1_{\mathsf{FU}})_\eta} & \mathsf{Form}_{\mathsf{T}'}(\mathsf{FU}) \\ {\scriptstyle \tau_{1_{\mathsf{T}}}}\downarrow & \searrow^{(\tau_{\mathsf{U}})_\eta} & \downarrow{\scriptstyle (\tau_{\mathsf{U}})_{1_{\mathsf{T}'}}} \\ \mathsf{Form}_{\mathsf{T}}(\mathsf{F}') & \xrightarrow[(1_{\mathsf{F}'\mathsf{U}})_\eta]{} & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}'\mathsf{U}) \end{array}$$

Proof (*Sketch*). The statement follows by structural induction on τ. □

Corollary 27. *Let* $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$ *denote a many-sorted cosignature morphism, and let* $(\tau_1 : \mathsf{F}_1 \Rightarrow \mathsf{F}) \in \| \mathsf{KP}_S \|$[24] *and* $(\tau_2 : \mathsf{F}' \Rightarrow \mathsf{F}_1\mathsf{U}) \in \| \mathsf{KP}_{S'} \|$ *be such that* $(\tau_{1\mathsf{U}} \circ \tau_2)_\eta$ *is defined in terms of* $(\tau_{1\mathsf{U}})_\eta$ *and* $(\tau_2)_{1_{\mathsf{T}'}}$ *using* (4) *of Definition* 25. *Then,* $(\tau_2)_{1_{\mathsf{T}'}} \circ (\tau_{1\mathsf{U}})_\eta = (\tau_{1\mathsf{U}} \circ \tau_2)_\eta = (\tau_2)_\eta \circ (\tau_1)_{1_{\mathsf{T}}}$:

$$\begin{array}{ccc} \mathsf{Form}_{\mathsf{T}}(\mathsf{F}) & \xrightarrow{(\tau_{1\mathsf{U}})_\eta} & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}_1\mathsf{U}) \\ {\scriptstyle (\tau_1)_{1_{\mathsf{T}}}}\downarrow & \searrow^{(\tau_{1\mathsf{U}} \circ \tau_2)_\eta} & \downarrow{\scriptstyle (\tau_2)_{1_{\mathsf{T}'}}} \\ \mathsf{Form}_{\mathsf{T}}(\mathsf{F}_1) & \xrightarrow[(\tau_2)_\eta]{} & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}') \end{array}$$

Proof (*Sketch*). Definition 25 and Proposition 26 are used. □

Remark 28. The following are consequences of Definition 25 and Corollary 27:

- $[\pi_i]\varphi \overset{(\tau_1 \times \tau_2)_\eta}{\longmapsto} [\pi_i]\varphi'$ if $\varphi \overset{(\tau_i)_\eta}{\longmapsto} \varphi'$
- $[\kappa_i]\varphi \overset{(\tau_1 + \tau_2)_\eta}{\longmapsto} ([\kappa_i]\varphi' \wedge [\kappa_j]\top) \equiv [\kappa_i]\varphi'$ if $\varphi \overset{(\tau_i)_\eta}{\longmapsto} \varphi'$, $j = \{1, 2\} \setminus \{i\}$
- $[ev(d)]\varphi \overset{(\tau^D)_\eta}{\longmapsto} \psi \equiv [ev(d)]\varphi'$ if $\varphi \overset{\tau_\eta}{\longmapsto} \varphi'$
- $[ev(d)]\varphi \overset{((\mathsf{FU})^\alpha)_\eta}{\longmapsto} \xi \equiv [ev(\alpha(d))]\varphi'$ if $\varphi \overset{(1_{\mathsf{FU}})_\eta}{\longmapsto} \varphi'$

[23] Hence, $(\tau_{\mathsf{U}} : \mathsf{F}'\mathsf{U} \Rightarrow \mathsf{FU}) \in \| \mathsf{KP}_{S'} \|$.
[24] Hence, $(\tau_{1\mathsf{U}} : \mathsf{F}_1\mathsf{U} \Rightarrow \mathsf{FU}) \in \| \mathsf{KP}_{S'} \|$.

where the natural transformations $\tau_1 \times \tau_2 : \mathsf{F}'_1 \times \mathsf{F}'_2 \Rightarrow \mathsf{F}_1\mathsf{U} \times \mathsf{F}_2\mathsf{U}$, $\tau_1 + \tau_2 : \mathsf{F}'_1 + \mathsf{F}'_2 \Rightarrow \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U}$, $\tau^D : \mathsf{F}'^D \Rightarrow (\mathsf{FU})^D$ and $(\mathsf{FU})^\alpha : (\mathsf{FU})^{D'} \Rightarrow (\mathsf{FU})^D$ are as in Remark 2.

Moreover, the translation of formulae along cosignature morphisms is compatible with the equalities (2)–(2) in Remark 2, in a sense made precise below.

Proposition 29. *Let $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$ denote a many-sorted cosignature morphism. Then, the following hold up to semantic equivalence*:

(1) $\langle \tau_1, \tau_2 \rangle_\eta \circ (\pi_i)_{1_\mathsf{T}} = (\tau_i)_\eta$ *for* $(\tau_i : \mathsf{F} \Rightarrow \mathsf{F}_i\mathsf{U}) \in \| \mathsf{KP}_{S'} \|$, $i = 1, 2$*:*

$$\mathsf{Form}_\mathsf{T}(\mathsf{F}_i) \xrightarrow{(\pi_i)_{1_\mathsf{T}}} \mathsf{Form}_\mathsf{T}(\mathsf{F}_1 \times \mathsf{F}_2) \xrightarrow{\langle \tau_1, \tau_2 \rangle_\eta} \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}), \quad (\tau_i)_\eta$$

(2) $(\kappa_i)_{1_{\mathsf{T}'}} \circ [\tau_1, \tau_2]_\eta = (\tau_i)_\eta$ *for* $(\tau_i : \mathsf{F}_i \Rightarrow \mathsf{FU}) \in \| \mathsf{KP}_{S'} \|$, $i = 1, 2$*:*

$$\mathsf{Form}_\mathsf{T}(\mathsf{F}) \xrightarrow{[\tau_1, \tau_2]_\eta} \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}_1 + \mathsf{F}_2) \xrightarrow{(\kappa_i)_{1_{\mathsf{T}'}}} \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}_i), \quad (\tau_i)_\eta$$

(3) $(\tau^* \times 1_D)_\eta \circ (eval_{\mathsf{F},D})_{1_\mathsf{T}} = \tau_\eta$ *for* $(\tau : \mathsf{F}' \times D \Rightarrow \mathsf{FU}) \in \| \mathsf{KP}_{S'} \|$*:*

$$\mathsf{Form}_\mathsf{T}(\mathsf{F}) \xrightarrow{(eval_{\mathsf{F},D})_{1_\mathsf{T}}} \mathsf{Form}_\mathsf{T}(\mathsf{F}^D \times D) \xrightarrow{(\tau^* \times 1_D)_\eta} \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}' \times D), \quad \tau_\eta$$

Proof (*Sketch*). The statement follows directly from Definition 25.[25] □

In practice, translating a particular formula involves a number of applications of the rules in Definition 25. Typically, each occurrence of $[\mathsf{next}_s]$ in the formula being translated triggers an application of the rule (2b), followed by a number of applications of rules in (3) and a number of applications of rules in (2).

Example 30. Let $(1_1, \eta_1) : (1, \mathsf{T}_{\mathtt{LIST}}) \to (1, \mathsf{T}_{\mathtt{FLIST}})$ be as in Example 22. The translation of the modal formula defining lists over E (see Example 15) along $(1_1, \eta_1)$ is obtained as follows:

[25] Note that Corollary 27 can not be applied here, since the translations along $(\tau_i)_\eta$ and τ_η have been defined *independently of* the translations along the other cosignature morphisms involved.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\top \overset{(1_1)_{\eta_1}}{\longmapsto} \top}{\langle\kappa_1\rangle\top \overset{(1_{1+E})_{\eta_1}}{\longmapsto} \langle\kappa_1\rangle\top}}{[\pi_1]\langle\kappa_1\rangle\top \overset{(1_{(1+E)\times(1+\mathsf{Id})})_{\eta_1}}{\longmapsto} [\pi_1]\langle\kappa_1\rangle\top}}{[\pi_1]\langle\kappa_1\rangle\top \overset{\eta_{1\eta_1}}{\longmapsto} [\pi_1][\pi_1]\langle\kappa_1\rangle\top}}{[\mathsf{next}][\pi_1]\langle\kappa_1\rangle\top \overset{(1_{\mathsf{Id}})_{\eta_1}}{\longmapsto} [\mathsf{next}][\pi_1][\pi_1]\langle\kappa_1\rangle\top} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\top \overset{(1_1)_{\eta_1}}{\longmapsto} \top}{\langle\kappa_1\rangle\top \overset{(1_{1+\mathsf{Id}})_{\eta_1}}{\longmapsto} \langle\kappa_1\rangle\top}}{[\pi_2]\langle\kappa_1\rangle\top \overset{(1_{(1+E)\times(1+\mathsf{Id})})_{\eta_1}}{\longmapsto} [\pi_2]\langle\kappa_1\rangle\top}}{[\pi_2]\langle\kappa_1\rangle\top \overset{\eta_{1\eta_1}}{\longmapsto} [\pi_1][\pi_2]\langle\kappa_1\rangle\top}}{[\mathsf{next}][\pi_2]\langle\kappa_1\rangle\top \overset{(1_{\mathsf{Id}})_{\eta_1}}{\longmapsto} [\mathsf{next}][\pi_1][\pi_2]\langle\kappa_1\rangle\top}}{[\mathsf{next}][\pi_1]\langle\kappa_1\rangle\top \leftrightarrow [\mathsf{next}][\pi_2]\langle\kappa_1\rangle\top \overset{(1_{\mathsf{Id}})_{\eta_1}}{\longmapsto} [\mathsf{next}][\pi_1][\pi_1]\langle\kappa_1\rangle\top \leftrightarrow [\mathsf{next}][\pi_1][\pi_2]\langle\kappa_1\rangle\top}$$

Any specification of finite lists should include the above formula. In addition, any such specification should require a certain consistency between the length operation and the tail operation. This is captured by the following formulae:

$$[\mathsf{next}][\pi_2]0 \leftrightarrow [\mathsf{next}][\pi_1][\pi_2]\langle\kappa_1\rangle\top$$
$$[\mathsf{next}][\pi_2](n+1) \leftrightarrow [\mathsf{next}][\pi_1][\pi_2]\langle\kappa_2\rangle[\mathsf{next}][\pi_2]n,\ n \in \mathbb{N}$$

formalising the observation that the length of a list is 0 precisely when the tail of the list is undefined, while the length of a list whose tail is defined is obtained by adding 1 to the length of the tail. After renaming $[\mathsf{next}][\pi_1][\pi_1]\langle\kappa_1\rangle$, $[\mathsf{next}][\pi_1][\pi_2]\langle\kappa_1\rangle$, $[\mathsf{next}][\pi_1][\pi_2]\langle\kappa_2\rangle$ and $[\mathsf{next}][\pi_2]$ to $\texttt{<hdU>}$, $\texttt{<tlU>}$, $\texttt{<tlD>}$ and respectively $\texttt{[len]}$, the specification of finite lists becomes:

$$\texttt{<hdU>}\top \leftrightarrow \texttt{<tlU>}\top$$
$$\texttt{[len]}\,0 \leftrightarrow \texttt{<tlU>}\top$$
$$\texttt{[len]}\,(n+1) \leftrightarrow \texttt{<tlD>[len]}\,n,\ n \in \mathbb{N}$$

where

$$\begin{aligned} l \models \texttt{<hdU>}\varphi_1 &\Leftrightarrow \exists s\,.\,hd(l) = \kappa_1(s) \text{ and } s \models \varphi_1 \\ l \models \texttt{<tlU>}\varphi_1 &\Leftrightarrow \exists s\,.\,tl(l) = \kappa_1(s) \text{ and } s \models \varphi_1 \\ l \models \texttt{<tlD>}\varphi_2 &\Leftrightarrow \exists t\,.\,tl(l) = \kappa_2(t) \text{ and } t \models \varphi_2 \\ l \models \texttt{[len]}\,n &\Leftrightarrow len(l) = n \end{aligned}$$

for any $\mathsf{T}_{\texttt{FLIST}}$-coalgebra $\langle\langle hd, tl\rangle, len\rangle : L \to ((1+E)\times(1+L))\times\mathbb{N}$ and any $l \in L$.

Example 31. Let $(f, \eta_2) : (\{\texttt{FList}\}, \mathsf{T}_{\texttt{FLIST}}) \to (\{\texttt{mList}, \texttt{Array}\}, \mathsf{T})$ be as in Example 23. Translating the modal formulae defining finite lists over E (see Example 30) along (f, η_2) yields the formulae:

$$\begin{aligned} [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_1][\pi_1]\langle\kappa_1\rangle\top &\leftrightarrow [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_1\rangle\top \\ [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_2]0 &\leftrightarrow [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_1\rangle\top \\ [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_2](n+1) &\leftrightarrow [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_2\rangle[\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_2]n,\ n < m \\ \bot &\leftrightarrow [\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_2\rangle[\mathsf{next}_{\texttt{mList}}][\pi_1][\pi_2]n,\ n = m \\ \bot &\leftrightarrow \bot,\ n > m \end{aligned}$$

of type $\texttt{mList}$. In particular, the translation of the last formula defining finite lists over E yields three different formulae, depending on the value of n in this formula. For $n \geqslant m$, its left subformula translates to $\bot$ along $(1_{\Pi_{\texttt{mList}}})_{\eta_2}$, since $n+1$ translates to $\bot$ along ι_{η_2}

(with $\iota : \{0, \ldots, m\} \to \mathbb{N}$). For $n > m$, its right subformula also translates to $\bot$ (for similar reasons). For $n = m$, the translated formula states that there are no lists whose tail has length m. Its left and right subformulae are computed as follows:

$$\dfrac{\dfrac{\dfrac{m+1 \overset{\iota_{\eta_2}}{\longmapsto} \bot}{[\pi_2](m+1) \overset{(1\times\iota)_{\eta_2}}{\longmapsto} [\pi_2]\bot \equiv \bot} \quad \bot \overset{[\pi_1]_{1_\mathsf{T}}}{\longmapsto} \bot}{[\pi_2](m+1) \overset{\eta_{2\eta_2}}{\longmapsto} \bot}}{[\mathsf{next}][\pi_2](m+1) \overset{(1\Pi_{\mathtt{mList}})_{\eta_2}}{\longmapsto} [\mathsf{next}_{\mathtt{mList}}]\bot \equiv \bot}$$

$$\dfrac{\dfrac{\dfrac{m \overset{\iota_{\eta_2}}{\longmapsto} m}{[\pi_2]m \overset{(1\times\iota)_{\eta_2}}{\longmapsto} [\pi_2]m} \quad [\pi_2]m \overset{[\pi_1]_{1_\mathsf{T}}}{\longmapsto} [\pi_1][\pi_2]m}{[\pi_2]m \overset{\eta_{2\eta_2}}{\longmapsto} [\pi_1][\pi_2]m}}{[\mathsf{next}][\pi_2]m \overset{\eta_{2\eta_2}}{\longmapsto} [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m}}{\ldots}$$

$$[\mathsf{next}][\pi_1][\pi_2]\langle\kappa_2\rangle[\mathsf{next}][\pi_2]m \overset{\eta_{2\eta_2}}{\longmapsto} [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_2\rangle[\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]m$$

After introducing the following abbreviations:

$$\begin{aligned}
\texttt{<hdU>} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_1][\pi_1]\langle\kappa_1\rangle \\
\texttt{<hdD>} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_1][\pi_1]\langle\kappa_2\rangle \\
\texttt{<tlU>} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_1\rangle \\
\texttt{<tlD>} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_1][\pi_2]\langle\kappa_2\rangle \\
\texttt{[len]} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_1][\pi_2]
\end{aligned}$$

the previous formulae become:

$$\begin{aligned}
\texttt{<hdU>}\top &\leftrightarrow \texttt{<tlU>}\top \\
\texttt{[len]}\,0 &\leftrightarrow \texttt{<tlU>}\top \\
\texttt{[len]}\,(n+1) &\leftrightarrow \texttt{<tlD>[len]}\,n, \quad n < m \\
\bot &\leftrightarrow \texttt{<tlD>[len]}\,n, \quad n = m
\end{aligned}$$

In addition to these formulae, formulae which constrain the list observer and the array observer have to be specified. For this purpose, we introduce some additional abbreviations, namely:

$$\begin{aligned}
\texttt{<elU(}p\texttt{)>} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_2][ev(p)]\langle\kappa_1\rangle \\
\texttt{<elD(}p\texttt{)>} &::= [\mathsf{next}_{\mathtt{mList}}][\pi_2][ev(p)]\langle\kappa_2\rangle \\
\texttt{[list]} &::= [\mathsf{next}_{\mathtt{Array}}][\pi_1] \\
\texttt{[get(}p\texttt{)]} &::= [\mathsf{next}_{\mathtt{Array}}][\pi_2][ev(p)]
\end{aligned}$$

With this notation, the following formulae complete the specification of arrays:

$$\begin{aligned}
\texttt{<elU(1)>}\top &\leftrightarrow \texttt{<hdU>}\top \\
\texttt{<elD(1)>}e &\leftrightarrow \texttt{<hdD>}e, \quad e \in E \\
\texttt{<elU(}p+1\texttt{)>}\top &\leftrightarrow \texttt{<tlU>}\top \vee \texttt{<tlD><elU(}p\texttt{)>}\top, \quad p \in \{1,\ldots,m-1\} \\
\texttt{<elD(}p+1\texttt{)>}e &\leftrightarrow \texttt{<tlD><elD(}p\texttt{)>}e, \quad p \in \{1,\ldots,m-1\},\ e \in E \\
\texttt{[get(}p\texttt{)]}e &\leftrightarrow \texttt{[list]<elD(}p\texttt{)>}e, \quad p \in \{1,\ldots,m\},\ e \in E
\end{aligned}$$

The first four formulae, of type `mList`, specify the list observer in terms of the head and tail operations on lists. The last formula, of type `Array`, specifies the array observer in terms of the previously-defined list observer. What this formula actually states is that, for any position $p \in \{1,\ldots,m\}$, the pth element of an array is given by the pth element of the associated list. As a result, all the lists used to represent arrays are constrained to have length (at least) m. Thus, the formula:

$$\texttt{[list][len]}m$$

(see Example 16) is a global semantic consequence of the array specification.

As mentioned previously, the translation of formulae along cosignature morphisms preserves the interpretations of formulae.

Proposition 32. *Let $(f,\eta) : (S,\mathsf{T}) \to (S',\mathsf{T}')$ denote a many-sorted cosignature morphism, let $\langle C',\gamma'\rangle$ denote a T'-coalgebra, and let $\gamma = \eta_{C'} \circ \mathsf{U}\gamma' : \mathsf{U}C' \to \mathsf{T}\mathsf{U}C'$ denote its T-reduct along η. Then, $\tau_{C'}^{-1}([\![\varphi]\!]_{\mathsf{F}}^{\gamma}) = [\![\tau_\eta(\varphi)]\!]_{\mathsf{F}'}^{\gamma'}$ for any $\mathsf{F} \in |\mathsf{KP}_S|$, $\mathsf{F}' \in |\mathsf{KP}_{S'}|$, $(\tau : \mathsf{F}' \Rightarrow \mathsf{F}\mathsf{U}) \in \|\mathsf{KP}_{S'}\|$ and $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$.*

Proof. The statement follows by structural induction on φ and τ. Only a few cases are considered here. The remaining ones (see Definition 25) are treated similarly.

- If τ is given by $1_{\Pi_{f(s)}} : \Pi_{f(s)} \Rightarrow \Pi_{f(s)} = \Pi_s\mathsf{U}$ with $s \in S$:

$$\begin{aligned}
[\![[\mathsf{next}_s]\varphi]\!]_{\Pi_s}^{\gamma} &= \gamma_s^{-1}([\![\varphi]\!]_{\mathsf{T}_s}^{\gamma}) = (\gamma'_{f(s)})^{-1}(\eta_{s,C'}^{-1}([\![\varphi]\!]_{\mathsf{T}_s}^{\gamma})) \\
&= (\gamma'_{f(s)})^{-1}([\![(\eta_s)_\eta(\varphi)]\!]_{\mathsf{T}'_{f(s)}}^{\gamma'}) = [\![[\mathsf{next}_{f(s)}](\eta_s)_\eta(\varphi)]\!]_{\Pi_{f(s)}}^{\gamma'} \\
&= [\![(1_{\Pi_{f(s)}})_\eta([\mathsf{next}_s]\varphi)]\!]_{\Pi_{f(s)}}^{\gamma'} = [\![\tau_\eta([\mathsf{next}_s]\varphi)]\!]_{\Pi_{f(s)}}^{\gamma'}
\end{aligned}$$

- If τ is given by $\alpha : D' \Rightarrow D = D\mathsf{U}$:

$$\begin{aligned}
\tau_{C'}^{-1}([\![d]\!]_D^{\gamma}) &= \tau_{C'}^{-1}(\{d\}) = \{\, d' \in D' \mid \alpha(d') = d \,\} = \bigcup_{\alpha(d')=d} \{d'\} \\
&= \bigcup_{\alpha(d')=d} [\![d']\!]_{D'}^{\gamma'} = \Big[\!\Big[\bigvee_{\alpha(d')=d} d' \Big]\!\Big]_{D'}^{\gamma'} = [\![\alpha_\eta(d)]\!]_{D'}^{\gamma'} = [\![\tau_\eta(d)]\!]_{D'}^{\gamma'}
\end{aligned}$$

- If τ is given by $\pi_i : \mathsf{F}'_1 \times \mathsf{F}'_2 \Rightarrow \mathsf{F}_i\mathsf{U}$ with $i \in \{1,2\}$ and with $\mathsf{F}'_i = \mathsf{F}_i\mathsf{U}$:

$$\begin{aligned}
\tau_{C'}^{-1}([\![\varphi]\!]_{\mathsf{F}_i}^{\gamma}) &= \pi_i^{-1}([\![\varphi]\!]_{\mathsf{F}_i}^{\gamma}) = \pi_i^{-1}([\![(1_{\mathsf{F}_i\mathsf{U}})_\eta(\varphi)]\!]_{\mathsf{F}_i\mathsf{U}}^{\gamma'}) \\
&= [\![[\pi_i](1_{\mathsf{F}_i\mathsf{U}})_\eta(\varphi)]\!]_{\mathsf{F}'_1\times\mathsf{F}'_2}^{\gamma'} = [\![(\pi_i)_\eta(\varphi)]\!]_{\mathsf{F}'_1\times\mathsf{F}'_2}^{\gamma'} = [\![\tau_\eta(\varphi)]\!]_{\mathsf{F}'_1\times\mathsf{F}'_2}^{\gamma'}
\end{aligned}$$

- If τ is given by $\kappa_i : \mathsf{F}_i\mathsf{U} \Rightarrow \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 + \mathsf{F}_2)\mathsf{U}$ with $i \in \{1, 2\}$:
  - If $j = i$ and $\{l\} = \{1, 2\} \setminus \{j\}$:

$$\begin{aligned}\tau_{C'}^{-1}([\![[\kappa_j]\varphi]\!]^{\gamma}_{\mathsf{F}_1+\mathsf{F}_2}) &= \kappa_j^{-1}(\kappa_j([\![\varphi]\!]^{\gamma}_{\mathsf{F}_j}) \cup \kappa_l(\mathsf{F}_l\mathsf{U}C')) = [\![\varphi]\!]^{\gamma}_{\mathsf{F}_j}\\ &= [\![(1_{\mathsf{F}_j\mathsf{U}})_\eta(\varphi)]\!]^{\gamma'}_{\mathsf{F}_j\mathsf{U}} = [\![(\kappa_j)_\eta([\kappa_j]\varphi)]\!]^{\gamma'}_{\mathsf{F}_j\mathsf{U}}\\ &= [\![\tau_\eta([\kappa_j]\varphi)]\!]^{\gamma'}_{\mathsf{F}_j\mathsf{U}}\end{aligned}$$

  - If $j \neq i$:

$$\begin{aligned}\tau_{C'}^{-1}([\![[\kappa_j]\varphi]\!]^{\gamma}_{\mathsf{F}_1+\mathsf{F}_2}) &= \kappa_i^{-1}(\kappa_j([\![\varphi]\!]^{\gamma}_{\mathsf{F}_j}) \cup \kappa_i(\mathsf{F}_i\mathsf{U}C')) = \mathsf{F}_i\mathsf{U}C'\\ &= [\![\top]\!]^{\gamma'}_{\mathsf{F}_i\mathsf{U}} = [\![(\kappa_i)_\eta([\kappa_j]\varphi)]\!]^{\gamma'}_{\mathsf{F}_i\mathsf{U}} = [\![\tau_\eta([\kappa_j]\varphi)]\!]^{\gamma'}_{\mathsf{F}_i\mathsf{U}}\end{aligned}$$

- If τ is given by $\zeta^* : \mathsf{F}' \Rightarrow (\mathsf{F}\mathsf{U})^D = \mathsf{F}^D\mathsf{U}$ with $\zeta : \mathsf{F}' \times D \Rightarrow \mathsf{F}\mathsf{U}$:

$$\begin{aligned}\tau_{C'}^{-1}([\![[ev(d)]\varphi]\!]^{\gamma}_{\mathsf{F}^D}) &= (\zeta^*_{C'})^{-1}(\{\, f : D \to \mathsf{F}\mathsf{U}C' \mid f(d) \in [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \,\})\\ &= \{\, f' \in \mathsf{F}'C' \mid \zeta^*_{C'}(f')(d) \in [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \,\}\\ &= \{\, f' \in \mathsf{F}'C' \mid \zeta_{C'}(f', d) \in [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \,\}\\ &= \langle 1_{\mathsf{F}'}, d\rangle^{-1}_{C'}(\zeta_{C'}^{-1}([\![\varphi]\!]^{\gamma}_{\mathsf{F}})) = [\![\langle 1_{\mathsf{F}'}, d\rangle_{1_{\mathsf{T}'}}(\zeta_\eta(\varphi))]\!]^{\gamma'}_{\mathsf{F}'}\\ &= [\![\tau_\eta([ev(d)]\varphi)]\!]^{\gamma'}_{\mathsf{F}'} \qquad \square\end{aligned}$$

Thus, the denotation of a translated formula in a T'-coalgebra is obtained as an inverse image of the denotation of the original formula in the T-reduct of the given coalgebra. In particular, the denotation of a translated state formula in a T'-coalgebra coincides with the denotation of the original formula in the T-reduct of that T'-coalgebra—this follows by taking $\tau = 1_{\Pi_{f(s)}}$ with $s \in S$.

Definition 25 yields a functor $\mathsf{SForm} : \mathsf{Cosign} \to \mathsf{Set}$, taking a many-sorted cosignature to the set of state formulae over it, and a many-sorted cosignature morphism to the induced translation. We are then ready for our main result.

Theorem 33. $(\mathsf{Cosign}, \mathsf{Coalg}, \mathsf{SForm}, \models)$ *is an institution.*

Proof. The property of being an institution amounts to the following equivalence holding for any many-sorted cosignature morphism $\eta : (S, \mathsf{T}) \to (S', \mathsf{T}')$, any T'-coalgebra $\langle C', \gamma'\rangle$, and any formula $\varphi \in \mathsf{SForm}(\mathsf{T})$:

$$\langle C', \gamma'\rangle \models \eta(\varphi) \;\Leftrightarrow\; \mathsf{U}_\eta\langle C', \gamma'\rangle \models \varphi$$

Showing that the above equivalence holds can be reduced to showing that, given η and $\langle C', \gamma'\rangle$, $[\![\varphi]\!]^{\gamma}_{\Pi_s} = [\![\eta_s(\varphi)]\!]^{\gamma'}_{\Pi_{f(s)}}$ holds for any $\varphi \in \mathsf{SForm}(\mathsf{T})_s$ and any $s \in S$ (where $\gamma = \eta_{C'} \circ \mathsf{U}\gamma'$). Then, one can reason as follows: $\langle C', \gamma'\rangle \models \eta_s(\varphi) \;\Leftrightarrow\; [\![\eta_s(\varphi)]\!]^{\gamma'}_{\Pi_{f(s)}} = C'_{f(s)}$ $\Leftrightarrow [\![\varphi]\!]^{\gamma}_{\Pi_s} = (\mathsf{U}C')_s \;\Leftrightarrow\; \mathsf{U}_\eta\langle C', \gamma'\rangle \models \varphi$ for any $\varphi \in \mathsf{SForm}(\mathsf{T})_s$ and any $s \in S$. But the

previous claim follows from Proposition 32, namely by taking $\mathsf{F} = \Pi_s$, $\mathsf{F}' = \Pi_{f(s)}$ and $\tau = 1_{\Pi_{f(s)}}$ for $s \in S$. □

Any institution comes equipped with notions of *specification* and *specification morphism* (see [3]). In our setting, they are as follows.

Definition 34. A **(many-sorted) coalgebraic specification** is given by a tuple (S, T, Φ), with (S, T) a many-sorted cosignature and Φ a set of state formulae over T. A **(many-sorted) coalgebraic specification morphism** from (S, T, Φ) to (S', T', Φ') is given by a many-sorted cosignature morphism $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$, additionally satisfying $\Phi' \models \eta(\varphi)$ for any $\varphi \in \Phi$.[26]

For a coalgebraic specification $\mathsf{Sp} = (S, \mathsf{T}, \Phi)$, the full subcategory of $\mathsf{Coalg}(\mathsf{T})$ whose objects satisfy Φ is denoted $\mathsf{Coalg}(\mathsf{Sp})$. Then, any specification morphism $(f, \eta) : \mathsf{Sp} \to \mathsf{Sp}'$ induces a reduct functor $\mathsf{U}_{(f,\eta)} : \mathsf{Coalg}(\mathsf{Sp}') \to \mathsf{Coalg}(\mathsf{Sp})$: by Theorem 33, the reduct functor induced by the underlying cosignature morphism takes T'-coalgebras satisfying Φ' (and hence $\eta(\Phi)$) to T-coalgebras satisfying Φ.

Related work

We conclude this section by comparing our approach to similar work concerning (institutions of) modal logics induced by predicate liftings.

In [12], the notion of *parameterised signature*, defined as a functor $\Omega : \mathsf{L} \times \mathsf{C} \to \mathsf{C}$, was used to define coalgebraic signatures and their morphisms. Given such a functor Ω, and given $L \in |\mathsf{L}|$, the functor $X \mapsto \Omega(L, X)$ defines a coalgebraic signature $\Omega_L : \mathsf{C} \to \mathsf{C}$. Also, for each $(l : L \to L') \in \| \mathsf{L} \|$, the C-arrows $\Omega(l, 1_C) : \Omega_L(C) \to \Omega_{L'}(C)$ with $C \in |\mathsf{C}|$ define a natural transformation $\hat{l} : \Omega_L \Rightarrow \Omega_{L'}$, and hence a morphism of coalgebraic signatures.

In the case when $\mathsf{C} = \mathsf{Set}$, modal logics induced by $|\mathsf{L}|$-indexed sets of *predicate liftings* were used in [12] formalise bisimulation-invariant properties of states of coalgebras. Given an endofunctor $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, a *predicate lifting for* T [12] is a natural transformation $\lambda : \hat{\mathscr{P}} \Rightarrow \hat{\mathscr{P}} \circ \mathsf{T}$, with $\hat{\mathscr{P}} : \mathsf{Set} \to \mathsf{Set}$ denoting the contravariant powerset functor.[27] The *modal language* $\mathscr{L}(\Lambda)$ induced by a set Λ of predicate liftings then contains a unary modal operator $[\lambda]$ for each $\lambda \in \Lambda$, as well as basic propositional connectives. Its coalgebraic semantics is defined inductively on the structure of formulae:

$$c \models_\gamma [\lambda]\varphi \text{ iff } \gamma(c) \in \lambda_C([\![\varphi]\!]_\gamma)$$

for each T-coalgebra $\langle C, \gamma \rangle$ and $c \in C$. Then, an $|\mathsf{L}|$-indexed family of predicate liftings $(\Lambda_L)_{L \in |\mathsf{L}|}$, with Λ_L containing predicate liftings for Ω_L, is said to be *coherent* if, for any $(l : L \to L') \in \| L \|$ and any $\lambda' \in \Lambda_{L'}$, $\hat{\mathscr{P}}\hat{l} \circ \lambda' \in \Lambda_L$. Any coherent family of predicate liftings $(\Lambda_L)_{L \in |\mathsf{L}|}$ induces a translation $l^* : \mathscr{L}(\Lambda_{L'}) \to \mathscr{L}(\Lambda_L)$ along each L-arrow $l : L \to L'$, with l^* being defined inductively by: $l^*([\lambda']\varphi') = [\hat{\mathscr{P}}\hat{l} \circ \lambda']l^*(\varphi')$ for $\varphi' \in \mathscr{L}(\Lambda_{L'})$. Finally, each pair consisting of a parameterised signature $\Omega : \mathsf{L} \times \mathsf{Set} \to \mathsf{Set}$ and a coherent family of predicate liftings $(\Lambda_L)_{L \in |\mathsf{L}|}$ is shown in [12] to give rise to an institution of modal logics for Ω_L-coalgebras, with L ranging over $|\mathsf{L}|$.

[26] Here, $\models \subseteq \mathscr{P}(\mathsf{SForm}(\mathsf{T}')) \times \mathsf{SForm}(\mathsf{T}')$ denotes global semantic consequence.

[27] $\hat{\mathscr{P}}$ takes a set to the set of its subsets, and a function to its inverse image.

The setting considered in [12] is, in a sense, more general than the one here, as it allows for arbitrary endofunctors on Set (and indeed, on any fixed category C). However, the modal logics considered here are not subsumed by logics induced by predicate liftings – the formulae associated to a Kripke polynomial endofunctor T have a multi-sorted structure, which can not, in general, be derived from a set of predicate liftings. This multi-sorted structure also makes the logics considered here generally more expressive than logics induced by predicate liftings. For instance, by taking $\mathsf{T} = \mathscr{P}_\omega \circ \mathscr{P}_\omega$, an expressive logic for T-coalgebras is obtained using the approach in [16,8], whereas no expressive logic arising from a set of predicate liftings is known to exist.

Nonetheless, our approach to defining translations of modal formulae along many-sorted cosignature morphisms follows the same principles as that of [12], with the notion of cosignature morphism being chosen in such a way that modal operators can be naturally translated along cosignature morphisms. Moreover, the proof of Proposition 32 is similar to that of [12, Theorem 4.7], where induction on the structure of formulae is used to show that the semantics of formulae is preserved by translations along L-arrows.

5. Semantic constructions

We now use final and cofree coalgebras to provide denotations for the specifications and specification morphisms of the institution defined in Section 4.

We begin by showing the existence of final models for coalgebraic specifications. A further restriction on Kripke polynomial endofunctors is required in this sense. Specifically, κ-bounded powerset functors (with κ some regular cardinal) must be used in the definition of Kripke polynomial endofunctors, in order to ensure that the resulting endofunctors are *κ-accessible*,[28] and hence have final coalgebras. The remainder of this section refers to Kripke polynomial endofunctors whose definition involves the bounded powerset functor $\mathscr{P}_\kappa$, rather than the unbounded powerset functor $\mathscr{P}$. For such endofunctors, the existence of final coalgebras can be inferred from results in [13] (see also [1]).

Corollary 35. *Let* (S, T) *denote a many-sorted cosignature. Then*, Coalg(T) *has a final object.*

Example 36. Let $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ be as in Example 5. A final $\mathsf{T}_{\mathtt{LIST}}$-coalgebra has carrier given by $S = (1 + E)^+ \cup (1 + E)^\omega$,[29] and the coalgebra map $\langle hd, tl\rangle : S \to (1 + E) \times (1 + S)$ given by

$$hd(e : s) = e$$
$$tl(e : s) = \begin{cases} \kappa_1(*) & \text{if } s = [] \\ \kappa_2(s) & \text{if } s \neq [] \end{cases}$$

for $e \in 1 + E$ and $s \in (1 + E)^* \cup (1 + E)^\omega$.

[28] For an endofunctor $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$, κ-accessibility amounts to the action of T on an S-sorted set C being determined by its action on the S-sorted subsets of C of cardinality smaller than κ.

[29] For a set A, the sets of finite sequences, finite and non-empty sequences, and respectively infinite sequences of elements of A are denoted A^*, A^+ and A^ω.

In order to extend the existence of final models from many-sorted cosignatures to specifications, we need the existence of *largest subcoalgebras* induced by sets of formulae.

Proposition 37. *Let* $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ *denote a* (*weak pullback preserving*) *endofunctor. Then, for any* T*-coalgebra* $\langle C, \gamma \rangle$ *and any* Set^S*-monomorphism* $\iota : X \to C$*, the category* $\mathsf{SubCoalg}(\langle C, \gamma \rangle, \iota)$ *has a final object.*

Proof (*Sketch*). The conclusion follows from the observation that a *factorisation system for sinks*[30] exists for $\mathsf{Coalg}(\mathsf{T})$ – [10, Corollary 1.3.14] states this for $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, but the result can be easily generalised to $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$. Consequently, a final object in $\mathsf{SubCoalg}(\langle C, \gamma \rangle, \iota)$ is obtained as the union of all subcoalgebras of $\langle C, \gamma \rangle$ whose carrier is "contained in X". □

Before applying Proposition 37 to Kripke polynomial endofunctors, we observe that all these endofunctors preserve weak pullbacks: constant and projections functors preserve weak pullbacks, and this property is preserved by products, coproducts, exponentials and bounded powersets.

Theorem 38. *Let* $\mathsf{Sp} = (S, \mathsf{T}, \Phi)$ *denote a coalgebraic specification. Then, the category* $\mathsf{Coalg}(\mathsf{Sp})$ *has a final object.*

Proof (*Sketch*). Let $\langle F, \zeta \rangle$ denote a final T-coalgebra (see Corollary 35), let $X \in |\mathsf{Set}^S|$ be given by $X_s = \{ f \in F_s \mid f \models \Phi_s \}$[31] for $s \in S$, and let $\iota : X \to F$ denote the induced inclusion. Also, let $\langle \langle D, \delta \rangle, m \rangle$ be a final object in $\mathsf{SubCoalg}(\langle C, \gamma \rangle, \iota)$ (see Proposition 37). Then, $\langle D, \delta \rangle$ is final in $\mathsf{Coalg}(\mathsf{Sp})$. □

Example 39. Let $\mathsf{Sp} = (\mathsf{T}_{\mathtt{LIST}}, \Phi)$ denote the coalgebraic specification of lists, as given in Example 15. That is, Φ consists of the following formula:

$$\texttt{<hdU>}\top \leftrightarrow \texttt{<tlU>}\top$$

A state $e : s$ of the final $\mathsf{T}_{\mathtt{LIST}}$-coalgebra (see Example 36) satisfies this formula if $e = \kappa_1(*)$ precisely when $s = []$. Taking the largest subcoalgebra induced by the above formula yields a coalgebra whose carrier is isomorphic to $E^* \cup E^\omega$ (i.e. the set of finite or infinite lists with elements from E).

An important property of institutions is *liberality*. This amounts to the existence of adjoints to the reduct functors induced by specification morphisms. In algebraic specification, left adjoints are of interest, as they yield free algebra constructions. However, in coalgebraic specification, cofree coalgebras are typically used at the semantic level (see e.g. [18,2]). The following generalisation of [2, Theorem 3.1.62] (see also [18, Theorem 17.1]) will allow us to prove the existence of cofree coalgebras induced by coalgebraic specification morphisms.

Proposition 40. *Let* C *and* D *be categories with binary products, and let* $\mathsf{U} : \mathsf{D} \to \mathsf{C}$ *be a functor which preserves binary products, and has a right adjoint* R*. Also, let* $\mathsf{T} : \mathsf{C} \to \mathsf{C}$ *and*

[30] See [10] for a definition.

[31] Here, Φ_s consists of all formulae in Φ which have type s.

$\mathsf{S} : \mathsf{D} \to \mathsf{D}$ *be* (*weak pullback preserving*) *endofunctors, and* $\eta : \mathsf{US} \Rightarrow \mathsf{TU}$ *be a natural transformation, inducing a reduct functor* $\mathsf{U}_\eta : \mathsf{Coalg}(\mathsf{S}) \to \mathsf{Coalg}(\mathsf{T})$. *If the functors* $\mathsf{T} \times C$ *and* $\mathsf{S} \times \mathsf{R}C$ *have final coalgebras for any* C*-object* C, *then* U_η *has a right adjoint*.

Proof (*Sketch*). The proof is similar to that of [2, Theorem 3.1.62]. □

Remark 41. We briefly comment on the relationship with similar results formulated in [12]. In the setting of [12], any parameterised cosignature $\Omega : \mathsf{L} \times \mathsf{C} \to \mathsf{C}$ induces a cofibration $p : \mathsf{E} \to \mathsf{L}$, with the fibre over $L \in |\mathsf{L}|$ being (isomorphic to) $\mathsf{Coalg}(\Omega_L)$. It is shown in [12, Theorem 3.3] that, if cofree Ω_L-coalgebras over C-objects exist, and if equalisers exist in each fibre $\mathsf{Coalg}(\Omega_L)$, then p is also a fibration. This translates to the functors $\mathsf{U}_l : \mathsf{Coalg}(\Omega_L) \to \mathsf{Coalg}(\Omega_{L'})$ induced by L-arrows $l : L \to L'$ having right adjoints. Thus, a result similar to Proposition 40 is obtained in [12], in the case when $\mathsf{U} : \mathsf{D} \to \mathsf{C}$ of Proposition 40 is the identity functor. Moreover, by [12, Corollary 3.7], the requirement that $\mathsf{Coalg}(\Omega_L)$ has equalisers is satisfied for endofunctors $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ which preserve weak pullbacks. Thus, in the case when $\mathsf{C} = \mathsf{D} = \mathsf{Set}$ and $\mathsf{U} = \mathsf{Id}$, the hypotheses of Proposition 40 imply those of [12, Theorem 3.3]—the existence of final $\Omega_L \times C$-coalgebras results in the existence of cofree Ω_L-coalgebras over C-objects. As a result, Proposition 40 is a consequence of [12, Theorem 3.3], in this particular case.

We now prove the existence of cofree coalgebras w.r.t. specification morphisms.

Theorem 42. *Let* $(f, \eta) : (S, \mathsf{T}, \Phi) \to (S', \mathsf{T}', \Phi')$ *denote a coalgebraic specification morphism. The reduct functor* $\mathsf{U}_{(f,\eta)} : \mathsf{Coalg}(S', \mathsf{T}', \Phi') \to \mathsf{Coalg}(S, \mathsf{T}, \Phi)$ *has a right adjoint*.

Proof (*Sketch*). We first use Proposition 40 to obtain a right adjoint to the reduct functor induced by the cosignature morphism $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$. For this, we note that the functor $\mathsf{U} : \mathsf{Set}^{S'} \to \mathsf{Set}^{S}$ induced by $f : S \to S'$ (see Definition 21) preserves binary products and has a right adjoint R.[32] Also, since T and T' are Kripke polynomial endofunctors defined using $\mathscr{P}_\kappa$, so are the endofunctors $\mathsf{T} \times C$ and $\mathsf{T}' \times \mathsf{R}C$, for any $C \in |\mathsf{C}|$. Hence, by Corollary 35, final coalgebras exist for these endofunctors. It then follows by Proposition 40 that the reduct functor induced by the cosignature morphism (f, η) has a right adjoint. Now let $\langle C, \gamma \rangle \in |\mathsf{Coalg}(S, \mathsf{T}, \Phi)|$, let $\langle C', \gamma' \rangle$ denote a cofree (S', T')-coalgebra over $\langle C, \gamma \rangle$ w.r.t. $\mathsf{U}_{(f,\eta)} : \mathsf{Coalg}(S', \mathsf{T}') \to \mathsf{Coalg}(S, \mathsf{T})$, and let $\langle \langle D, \delta \rangle, m \rangle$ denote the largest (S', T')-subcoalgebra of $\langle C', \gamma' \rangle$ which satisfies Φ'.[33] Then, $\langle D, \delta \rangle$ is cofree over $\langle C, \gamma \rangle$ w.r.t. $\mathsf{U}_{(f,\eta)} : \mathsf{Coalg}(S', \mathsf{T}', \Phi') \to \mathsf{Coalg}(S, \mathsf{T}, \Phi)$. □

Example 43. Let `LIST` and `ARRAY` denote the coalgebraic specifications of lists and respectively arrays of size m, as given in Examples 15 and 23, and let $(g, \eta) : (\{\mathtt{List}\}, \mathsf{T}_{\mathtt{LIST}}) \to (\{\mathtt{mList}, \mathtt{Array}\}, \mathsf{T}_{\mathtt{ARRAY}})$ denote the cosignature morphism obtained by composing the cosignature morphisms given in Examples 22 and 23. Then, (g, η) defines a coalgebraic specification morphism from `LIST` to `ARRAY` (since `ARRAY` contains the translations along η of all the formulae in `LIST`). Now let $\langle C, \gamma \rangle$ denote the $\mathsf{T}_{\mathtt{LIST}}$-coalgebra whose carrier is given by E^* (i.e. only the finite lists), and whose coalgebraic structure is given by

[32] R takes an S-sorted set $(C_s)_{s \in S}$ to the S'-sorted set $(\prod_{f(s)=s'} C_s)_{s' \in S'}$.

[33] $\langle \langle D, \delta \rangle, m \rangle$ is constructed as in the proof of Theorem 38.

$$\gamma([]) = \langle \kappa_1(*), \kappa_1(*) \rangle$$
$$\gamma(e : s) = \langle \kappa_2(e), \kappa_2(s) \rangle$$

for $e \in E$ and $s \in E^*$. The cofree $\mathsf{T}_{\mathtt{ARRAY}}$-coalgebra $\langle C', \gamma' \rangle$ over $\langle C, \gamma \rangle$ has the carrier given by $C'_{\mathtt{mList}} = E^0 \cup \ldots \cup E^m$ and $C'_{\mathtt{Array}} = E^m$ (with E^n denoting the set of lists of length n, for $n \in \mathbb{N}$), and the coalgebraic structure given by

$$\gamma'_{\mathtt{mList}}(l) = \gamma_{\mathtt{List}}(l)$$
$$\gamma'_{\mathtt{Array}}([e_1 \ldots e_m]) = \langle [e_1 \ldots e_m], (i \mapsto e_i)_{i=1,\ldots,m} \rangle$$

for $l \in C'_{\mathtt{mList}} \subseteq C_{\mathtt{List}}$ and $[e_1 \ldots e_m] \in E^m$. On the other hand, starting with a LIST-coalgebra with carrier E^n, where $n < m$, yields an ARRAY-coalgebra whose carrier has the Array-sorted component given by the empty set. (There are not enough lists in the LIST-coalgebra to implement arrays of size m.)

6. Conclusions

The main contributions of the paper can be summarised as follows. First, a generalisation of the modal logic described in [16,8] to endofunctors on categories of sorted sets was presented. Moreover, natural transformations arising from the structure of the endofunctors defining coalgebraic types were used to formally capture semantic relationships between these types. Such semantic relationships were subsequently lifted to a logical level, by equipping the underlying natural transformations with translations between the corresponding categories of coalgebras, as well as with translations between the corresponding languages. The resulting framework was shown to be an institution, with final and cofree coalgebras providing suitable denotations for its specifications and specification morphisms.

Acknowlegements

The author is grateful to the two anonymous referees for their valuable comments and suggestions.

References

[1] M. Barr, Terminal coalgebras in well-founded set theory, Theor. Comput. Sci., 114 (2) (1993) 299–315.
[2] C. Cîrstea, Integrating Observations and computations in the specification of state-based, dynamical systems. Ph.D. thesis, University of Oxford, 2000. Available from: <http://www.ecs.soton.ac.uk/~cc2/>.
[3] J. Goguen, R. Burstall, Institutions: abstract model theory for specification and programming, J. ACM 39 (1) (1992) 95–146.
[4] R. Goldblatt, Logics of Time and Computation, CSLI Lecture Notes, vol. 7, Stanford University, 1992.
[5] B. Jacobs, Inheritance and cofree constructions, in: P. Cointe (Ed.), European Conference on Object-Oriented Programming, Lecture Notes in Computer Science, vol. 1098, Springer, 1996, pp. 210–231.
[6] B. Jacobs, Objects and classes, coalgebraically, in: B. Freitag, C.B. Jones, C. Lengauer, H.-J. Schek (Eds.), Object Orientation with Parallelism and Persistence, Kluwer Academic Publishers, 1996, pp. 83–103.
[7] B. Jacobs, Invariants, bisimulations and the correctness of coalgebraic refinements, in: M. Johnson (Ed.), Algebraic Methodology and Software Technology, Lecture Notes in Computer Science, vol. 1349, Springer, 1997, pp. 276–291.

[8] B. Jacobs, Many-sorted coalgebraic modal logic: a model-theoretic study, Theor. Informatics Appl. 35 (1) (2001) 31–59.
[9] A. Kurz, Specifying coalgebras with modal logic, in: B. Jacobs, L. Moss, H. Reichel, J. Rutten (Eds.), Coalgebraic Methods in Computer Science, Electronic Notes in Theoretical Computer Science, vol. 11, Elsevier Science, 1998, pp. 57–71.
[10] A. Kurz, Logics for coalgebras and applications to computer science, Ph.D. thesis, Ludwig Maximilians Universität München, 2000.
[11] A. Kurz, R. Hennicker, On institutions for modular coalgebraic specifications, Theor. Comput. Sci. 280 (2002) 69–103.
[12] A. Kurz, D. Pattinson, Coalgebras and modal logic for parameterised endofunctors, Technical Report SEN-R0040, CWI, 2000.
[13] M. Makkai, R. Paré, Accessible Categories, Contemporary Mathematics, vol. 104, American Mathematical Society, 1990.
[14] L.S. Moss, Coalgebraic logic, Ann. Pure Appl. Logic 96 (1999) 277–317.
[15] D. Pattinson, Expressive logics for coalgebras via terminal sequence induction, Notre Dame J. Formal Logic 45 (1) (2004) 19–33.
[16] M. Rößiger, Coalgebras and modal logic, in: H. Reichel (Ed.), Coalgebraic Methods in Computer Science, Electronic Notes in Theoretical Computer Science, vol. 33, Elsevier Science, 2000, pp. 299–320.
[17] J.J.M.M. Rutten, A calculus of transition systems (towards universal coalgebra), Technical Report CS-R9503, CWI, 1995.
[18] J.J.M.M. Rutten, Universal coalgebra: a theory of systems, Theor. Comput. Sci. 249 (1) (2000) 3–80.