Electronic Notes in Theoretical Computer Science 65 No. 1 (2002)
URL: `http://www.elsevier.nl/locate/entcs/volume65.html` 17 pages

Institutionalising Many-Sorted Coalgebraic Modal Logic

Corina Cîrstea [1,2]

Computing Laboratory
University of Oxford
Oxford, UK

Abstract

[4] describes a modal logic for coalgebras of certain polynomial endofunctors on Set. This logic is here generalised to endofunctors on categories of sorted sets. The structure of the endofunctors considered is then exploited in order to define ways of moving from (coalgebras of) one endofunctor to (coalgebras of) another, and to equip them with translations between the associated modal languages. Furthermore, the resulting translations are shown to preserve and reflect the satisfaction of modal formulae by coalgebras.

1 Introduction

The use of coalgebras in modelling state-based, dynamical systems [9] generalises the use of transition systems as operational models for processes [8], with the notion of bisimulation playing an important rôle in coalgebraic approaches. Various kinds of modal logics can be used to reason about coalgebraic structures [6,5,7,4], in the same way as standard modal logic can be used to reason about transition system structures (see e.g. [3]). These logics *capture bisimulation*, in that logical equivalence of states coincides with the bisimulation relation. However, these logics depend on the particular endofunctors used to define the coalgebraic structures of interest, and different, but related endofunctors give rise to different, but not yet formally related modal logics. The aim of this paper is to provide an (institutional) framework for relating the modal logics associated to a particular class of endofunctors, namely those considered in [4]. (A similar, but more abstract such framework is described in [1, Section 2]. The framework introduced here complies with the one in [1].)

---

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

[2] Email: `corina.cirstea@comlab.ox.ac.uk`

[4] (see also [7]) describes a modal logic for coalgebras of *(finite) Kripke polynomial endofunctors*, that is, endofunctors on Set constructed from constant and identity functors using products, coproducts, exponentials with constant exponent and (finite) powersets. The approach in [4] is here taken further, on the one hand by generalising it to endofunctors on categories of sorted sets, and on the other by providing support for modular specification. The previously-mentioned generalisation is useful in situations where there is more than one type of interest, with sorts being used to name these types, and with the components of the endofunctors in question defining the (possibly interrelated) structures associated to these types. After defining Kripke polynomial endofunctors and their associated logics in the setting of categories of sorted sets, natural transformations arising from the structure of such endofunctors are used to define ways of moving from one Kripke polynomial endofunctor to another. Such natural transformations induce functors between the categories of coalgebras associated to their domains and respectively their codomains, as well as translations between the modal languages associated to their codomains and respectively their domains. Moreover, the satisfaction of modal formulae by coalgebras is preserved and reflected along these natural transformations. That is, the resulting framework has the property of being an *institution* [2]. The morphisms of this institution capture both refinement and encapsulation relations between coalgebraic types, as illustrated by several examples. The previously-mentioned property of the satisfaction relation allows specifications and their logical consequences to be carried along morphisms between coalgebraic types.

The paper is structured as follows. Section 2 extends the approach in [4] to endofunctors on categories of sorted sets. Section 3 defines ways of moving from one endofunctor to another which preserve and reflect the satisfaction of modal formulae by coalgebras. Section 4 summarises the results presented.

2 Coalgebraic Modal Logics for Kripke Polynomial Endofunctors on Categories of Sorted Sets

This section presents a generalisation of the coalgebraic modal logic introduced in [4] 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 dependencies between coalgebraic types.

Definition 2.1 Let S denote a set (of sorts)[3]. The **category of Kripke polynomial functors on** Set^S, denoted KP_S, is the least subcategory of

---

[3] Given a set S, the category Set^S *of S-sorted sets and S-sorted mappings* has objects given by families $A = (A_s)_{s \in S}$ with $A_s \in |\mathsf{Set}|$ for $s \in S$, and arrows from A to B given by families $(f_s)_{s \in S}$ with $(f_s : A_s \to B_s) \in \| \mathsf{Set} \|$ for $s \in S$.

$[\mathsf{Set}^S, \mathsf{Set}]$ such that:

- KP_S includes the subcategory of $[\mathsf{Set}^S, \mathsf{Set}]$ whose objects are constant functors $X \xmapsto{D} D$ for $X \in |\mathsf{Set}^S|$ with $D \in |\mathsf{Set}|$ finite and non-empty, and whose arrows are natural transformations $\alpha : D \Rightarrow D'$ with $(\alpha : D \to D') \in \|\mathsf{Set}\|$, and with D and D' finite and non-empty;
- $|\mathsf{KP}_S|$ contains the *projection functors* $\Pi_s : \mathsf{Set}^S \to \mathsf{Set}$ (taking S-sorted sets/functions to their s-component), for $s \in S$;
- KP_S is closed under binary products and coproducts;
- KP_S is closed under exponentials of form F^D, with $D \in |\mathsf{KP}_S|$ a constant functor;
- KP_S is closed under powersets;
- $\| \mathsf{KP}_S \|$ contains all natural transformations of form $d : \mathsf{F} \Rightarrow D$ (each of whose components is a constant function yielding d as result) with $\mathsf{F}, D \in |\mathsf{KP}_S|$, D a constant functor and $d \in D$[4].

Remark 2.2 Replacing the closure under powersets in Definition 2.1 with closure under finite powersets yields a notion of *finite Kripke polynomial functor on* Set^S. All the results in this paper are formulated for Kripke polynomial functors, however, they also hold for finite Kripke polynomial functors.

Remark 2.3 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 $i = 1, 2$
- $\eta^* : \mathsf{F}' \Rightarrow \mathsf{F}^D$ whenever $(\eta : \mathsf{F}' \times D \Rightarrow \mathsf{F}) \in \| \mathsf{KP}_S \|$ with D a constant functor
- $eval_{\mathsf{F},D} : \mathsf{F}^D \times D \Rightarrow \mathsf{F}$ whenever $\mathsf{F}, D \in |\mathsf{KP}_S|$ with D a constant functor
- $\mathcal{P}(\eta) : \mathcal{P}(\mathsf{F}) \Rightarrow \mathcal{P}(\mathsf{F}')$ whenever $(\eta : \mathsf{F} \Rightarrow \mathsf{F}') \in \| \mathsf{KP}_S \|$

subject to the following equalities:

(i) $\pi_i \circ \langle \eta_1, \eta_2 \rangle = \eta_i$ for $i = 1, 2$

(ii) $[\eta_1, \eta_2] \circ \kappa_i = \eta_i$ for $i = 1, 2$

(iii) $eval_{\mathsf{F},D} \circ (\eta^* \times 1_D) = \eta$

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 $i = 1, 2$

---

[4] These natural transformations will be needed in the treatment of exponentials.

- $\eta^D : \mathsf{F'}^D \Rightarrow \mathsf{F}^D$ (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
- $\mathsf{F}^\alpha : \mathsf{F}^{D'} \Rightarrow \mathsf{F}^D$ (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' constant functors.

The notion of *Kripke polynomial endofunctor* (see [4]) now generalises to categories of sorted sets as follows.

Definition 2.4 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 $s \in S$.

The objects of the category KP_1, with 1 denoting a one-element set, are precisely the Kripke polynomial endofunctors as defined in [4] (see also [7]). [4] also defines a category, denoted KPF, whose objects are the Kripke polynomial endofunctors on Set and whose arrows are *paths* between such endofunctors, with a path from F to $\mathsf{F'}$ corresponding to $\mathsf{F'}$ being used in the definition of (or being an *ingredient* of) F. While arrows in the category KPF capture *structural* dependencies between Kripke polynomial endofunctors on Set, arrows in the category KP_1 (and indeed, KP_S, for an arbitrary S) capture *semantic* dependencies between (the components of) Kripke polynomial endofunctors, in that coalgebras corresponding to their codomains can be extracted from coalgebras corresponding to their domains[5]. The former category is used in [4] to define modal formulae over Kripke polynomial endofunctors (by means of structural induction). The next definition generalises the notion of modal formula introduced in [4] to Kripke polynomial endofunctors on sorted sets. Instantiating it to Kripke polynomial endofunctors on Set yields a definition equivalent to the one in [4], but which does not make use of the notion of ingredient functor.

Definition 2.5 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 $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$
- $[\mathcal{P}]\varphi \in \mathsf{Form}_\mathsf{T}(\mathcal{P}(\mathsf{F}))$ if $\varphi \in \mathsf{Form}_\mathsf{T}(\mathsf{F})$.

---

[5] This observation will be exploited in Section 3 in order to obtain an institution of many-sorted coalgebraic modal logics.

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

Remark 2.6 If T is an endofunctor on Set and F is an ingredient of T (see [4]), then modal formulae over T of type F are essentially the same as modal formulae of *sort* F, as defined in [4] (w.r.t. T) [6].

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

Definition 2.7 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** $[\![\varphi]\!]^{\gamma}_{\mathsf{F}} \in \mathcal{P}(\mathsf{F}C)$ 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:

- $[\![\bot]\!]^{\gamma}_{\mathsf{F}} = \emptyset$
- $[\![\varphi \to \psi]\!]^{\gamma}_{\mathsf{F}} = \overline{[\![\varphi]\!]^{\gamma}_{\mathsf{F}}} \cup [\![\psi]\!]^{\gamma}_{\mathsf{F}}$ (where, for $X \in \mathcal{P}(\mathsf{F}C)$, $\overline{X}$ is given by $\mathsf{F}C \setminus X$)
- $[\![d]\!]^{\gamma}_{D} = \{d\}$ for $d \in D$
- $[\![\mathsf{next}_s\, \varphi]\!]^{\gamma}_{\Pi_s} = \gamma_s^{-1}([\![\varphi]\!]^{\gamma}_{\mathsf{T}_s})$ 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 $\{j\} = \{1, 2\} \setminus \{i\}$
- $[\![[ev(d)]\varphi]\!]^{\gamma}_{\mathsf{F}^D} = \{ f : D \to \mathsf{F}C \mid f(d) \in [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \}$ for $d \in D$
- $[\![[\mathcal{P}]\varphi]\!]^{\gamma}_{\mathcal{P}(\mathsf{F})} = \mathcal{P}([\![\varphi]\!]^{\gamma}_{\mathsf{F}})$

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 $\varphi \in \mathsf{Form}_{\mathsf{T}}(\Pi_s)$ 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 $[\![\varphi]\!]^{\gamma}_{\Pi_s} = C_s$.

Remark 2.8 The above definition generalises a similar definition in [4] to Kripke polynomial endofunctors on sorted sets.

Remark 2.9 The following are consequences of Definition 2.7: $[\![\top]\!]^{\gamma}_{\mathsf{F}} = C$, $[\![\neg\varphi]\!]^{\gamma}_{\mathsf{F}} = \overline{[\![\varphi]\!]^{\gamma}_{\mathsf{F}}}$, $[\![\varphi \vee \psi]\!]^{\gamma}_{\mathsf{F}} = [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \cup [\![\psi]\!]^{\gamma}_{\mathsf{F}}$ and $[\![\varphi \wedge \psi]\!]^{\gamma}_{\mathsf{F}} = [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \cap [\![\psi]\!]^{\gamma}_{\mathsf{F}}$, where $\top$, $\neg\varphi$, $\varphi \vee \psi$ and $\varphi \wedge \psi$ are given by $\bot \to \bot$, $\bot \to \varphi$, $\neg\varphi \to \psi$ and respectively $\neg(\varphi \to \neg\psi)$.

Definition 2.10 Let $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$ denote a Kripke polynomial endofunctor, and let $\mathsf{F} \in |\mathsf{KP}_S|$. The modal formulae $\varphi, \psi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$ are said to be

---

[6] The modal logic defined in [4] is also qualified as *many-sorted*. However, in [4], 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.

semantically equivalent (written $\varphi \equiv \psi$) if and only if $[\![\varphi]\!]^{\gamma}_{\mathsf{F}} = [\![\psi]\!]^{\gamma}_{\mathsf{F}}$ for any T-coalgebra $\langle C, \gamma \rangle$.

Remark 2.11 For a Kripke polynomial endofunctor $\mathsf{T} : \mathsf{Set}^S \to \mathsf{Set}^S$, one can also define:

- $\langle \mathsf{next}_s \rangle \varphi ::= \neg \mathsf{next}_s \neg \varphi \in \mathsf{Form}_{\mathsf{T}}(\Pi_s)$ 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 $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$
- $\langle \mathcal{P} \rangle \varphi ::= \neg [\mathcal{P}] \neg \varphi \in \mathsf{Form}_{\mathsf{T}}(\mathcal{P}(\mathsf{F}))$ for $\varphi \in \mathsf{Form}_{\mathsf{T}}(\mathsf{F})$

(The above operators are generalisations of the operators in [7] to categories of sorted sets.) Then, an immediate consequence of Definition 2.7 is that the pairs of modal formulae $\mathsf{next}_s\, \varphi$ and $\langle \mathsf{next}_s \rangle \varphi$, $[\pi_i]\varphi$ and $\langle \pi_i \rangle \varphi$, and respectively $[ev(d)]\varphi$ and $\langle ev(d) \rangle \varphi$ are semantically equivalent. The same, however, can not be said about the pairs $[\kappa_i]\varphi$ and $\langle \kappa_i \rangle \varphi$, and respectively $[\mathcal{P}]\varphi$ and $\langle \mathcal{P} \rangle \varphi$, as, for instance, $[\![[\kappa_1]\varphi]\!]^{\gamma}_{\mathsf{F}_1+\mathsf{F}_2} = \kappa_1([\![\varphi]\!]^{\gamma}_{\mathsf{F}_1}) \cup \kappa_2(\mathsf{F}_2 C) \supsetneqq \kappa_1([\![\varphi]\!]^{\gamma}_{\mathsf{F}_1}) = [\![\langle \kappa_1 \rangle \varphi]\!]^{\gamma}_{\mathsf{F}_1+\mathsf{F}_2}$, whereas $[\![[\mathcal{P}]\varphi]\!]^{\gamma}_{\mathcal{P}(\mathsf{F})} = \mathcal{P}([\![\varphi]\!]^{\gamma}_{\mathsf{F}}) \neq \{\, X \subseteq \mathsf{F}C \mid X \cap [\![\varphi]\!]^{\gamma}_{\mathsf{F}} \neq \emptyset \,\} = [\![\langle \mathcal{P} \rangle \varphi]\!]^{\gamma}_{\mathcal{P}(\mathsf{F})}$.

Example 2.12 Unlabelled transition systems are specified using the endofunctor $\mathsf{T}_{\mathsf{TS}} : \mathsf{Set} \to \mathsf{Set}$ given by $\mathsf{T}_{\mathsf{TS}} = \mathcal{P}(\mathsf{Id})$.

Example 2.13 Given $A \in |\mathsf{Set}|$, A-labelled transition systems are specified using the endofunctor $\mathsf{T}_{\mathsf{LTS}} : \mathsf{Set} \to \mathsf{Set}$ given by $\mathsf{T}_{\mathsf{LTS}} = \mathcal{P}(A \times \mathsf{Id})$.

Example 2.14 Unlabelled transition systems of finite depth are specified using the endofunctor $\mathsf{T}_{\mathsf{FTS}} : \mathsf{Set} \to \mathsf{Set}$ given by $\mathsf{T}_{\mathsf{FTS}} = \mathcal{P}(\mathsf{Id}) \times \mathbb{N}$, together with the modal formulae:

$$\begin{aligned} \mathsf{next}\,[\pi_2]0 &\leftrightarrow \mathsf{next}\,[\pi_1][\mathcal{P}]\bot \\ \mathsf{next}\,[\pi_2](n+1) &\leftrightarrow \mathsf{next}\,[\pi_1](\langle \mathcal{P} \rangle \mathsf{next}\,[\pi_2]n \wedge [\mathcal{P}]\mathsf{next}\,[\pi_2](0 \vee \ldots \vee n)),\ \ n \in \mathbb{N} \end{aligned}$$

Renaming $\mathsf{next}\,[\pi_1][\mathcal{P}]$ to `[succ]`, $\mathsf{next}\,[\pi_1]\langle \mathcal{P} \rangle$ to `<succ>`, and $\mathsf{next}\,[\pi_2]$ to `[depth]`, and using the distributivity of $\mathsf{next}\,[\pi_1]$ over $\wedge$ w.r.t. semantic equivalence [7], we obtain the following equivalent specification of unlabelled transition systems of finite depth:

$$\begin{aligned} \texttt{[depth]}\,0 &\leftrightarrow \texttt{[succ]}\bot \\ \texttt{[depth]}(n+1) &\leftrightarrow \texttt{<succ>[depth]}\,n \wedge \texttt{[succ]}\,\texttt{[depth]}(0 \vee \ldots \vee n),\ \ n \in \mathbb{N} \end{aligned}$$

where:

$$\begin{aligned} c \models \texttt{[succ]}\varphi &\Leftrightarrow (\forall\, c')\ (c' \in \texttt{succ}_C(c) \Rightarrow c' \models \varphi) \\ c \models \texttt{<succ>}\varphi &\Leftrightarrow (\exists\, c')\ (c' \in \texttt{succ}_C(c) \text{ and } c' \models \varphi) \\ c \models \texttt{[depth]}\varphi_{\mathbb{N}} &\Leftrightarrow (\forall\, n)\ (\texttt{depth}_C(c) = n \Rightarrow n \models \varphi_{\mathbb{N}}) \end{aligned}$$

---

[7] The distributivity of each of next and $[\pi_1]$ over $\wedge$ is a consequence of [4, Lemmas 3.3 and 4.3], but also follows directly from Definition 2.7 and Remark 2.9.

for any $\mathsf{T}_{\mathtt{FTS}}$-coalgebra $C = \langle C, \langle \mathtt{succ}_C, \mathtt{depth}_C \rangle\rangle$ and any $c \in C$. Thus, the above formulae formalise the statement that a rooted transition system has depth 0 precisely when its root has no successors, and has depth $n+1$ precisely when its root has a successor of depth n, and the depth of any of its successors does not exceed n.

Example 2.15 Lists whose elements belong to a set E are specified using the endofunctor $\mathsf{T}_{\mathtt{LIST}} : \mathsf{Set} \to \mathsf{Set}$ given by $\mathsf{T}_{\mathtt{LIST}} = (1 + E) \times (1 + \mathsf{Id})$ (with 1 denoting a one-element set), together with the modal formula:

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

After renaming $\mathsf{next}\,[\pi_1]\langle\kappa_1\rangle$ and $\mathsf{next}\,[\pi_2]\langle\kappa_1\rangle$ to `<headF>` and respectively `<tailF>`, the above modal formula becomes:

$$\mathtt{<headF>}\top \leftrightarrow \mathtt{<tailF>}\top$$

where:

$$\begin{aligned} c \models \mathtt{<headF>}\varphi_1 &\Leftrightarrow (\exists\, s)\ (\mathtt{head}_C(c) = \kappa_1(s) \text{ and } s \models \varphi_1) \\ c \models \mathtt{<tailF>}\varphi_1 &\Leftrightarrow (\exists\, s)\ (\mathtt{tail}_C(c) = \kappa_1(s) \text{ and } s \models \varphi_1) \end{aligned}$$

for any $\mathsf{T}_{\mathtt{LIST}}$-coalgebra $C = \langle C, \langle \mathtt{head}_C, \mathtt{tail}_C \rangle\rangle$ and any $c \in C$. Thus, the specification of lists formalises the observation that a list has no head if and only if it has no tail.

3 An Institution of Coalgebraic Modal Logics

The arrows of the category KP_S capture semantic dependencies between (the components of) Kripke polynomial endofunctors. In the following, such arrows will be used to define ways of moving from one Kripke polynomial endofunctor to another which preserve and reflect the satisfaction of modal formulae by coalgebras. Such an approach provides support for modular specification, as it allows specifications and their (global) semantic consequences to be carried over from less complex coalgebraic types to more complex ones. For instance, this will allow us to obtain a specification of *labelled* transition systems of finite depth by simply translating the specification of unlabelled transition systems of finite depth in Example 2.14 along a natural transformation which adds labels to the type structure. And moreover, anything that was proved previously about unlabelled transition systems of finite depth remains true when translated to labelled transition systems of finite depth.

Collections of (related) coalgebraic types are specified using *many-sorted cosignatures*, while ways of moving from one such collection to another (larger or more refined one) are specified using *many-sorted cosignature morphisms*.

Definition 3.1 A **many-sorted cosignature** is a tuple (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 tuple (f, η) with $f : S \to S'$ and $\eta : \mathsf{UT}' \Rightarrow \mathsf{TU}$, such that $\Pi_s \eta \in \| \mathsf{KP}_{S'} \|$ for each $s \in S$. (Here $\mathsf{U} : \mathsf{Set}^{S'} \to \mathsf{Set}^S$ denotes the functor taking S'-sorted sets/functions to the S-sorted sets/functions whose s-component is given by the $f(s)$-component of the S'-sorted set/function in question, for $s \in S$.) The category of many-sorted cosignatures and many-sorted cosignature morphisms is denoted Cosign.

Remark 3.2 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 $s \in S$.

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.

We will show in the following that many-sorted cosignature morphisms also 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, translating along identity natural transformations τ will not leave modal formulae unchanged, unless the underlying η is itself an identity natural transformation. Furthermore, identity natural transformations of form $\tau = 1_{\Pi_{f(s)}}$ will play a crucial rôle in defining the above-mentioned translations; it will be these natural transformations which will ultimately ensure moving from modal formulae over T to modal formulae over T'.

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 property of the translations will later allow us to prove that the defining condition of institutions holds in our framework.

Definition 3.3 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 $(\tau : \mathsf{F}' \Rightarrow \mathsf{FU}) \in \| \mathsf{KP}_{\mathsf{S}'} \|$[8], the

---

[8] 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 $s \in S$ (see Remark 3.2).

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:

- (i) (a) $\bot \xmapsto{\tau_\eta} \bot$
  - (b) $(\varphi \to \psi) \xmapsto{\tau_\eta} (\varphi' \to \psi')$ if $\varphi \xmapsto{\tau_\eta} \varphi'$ and $\psi \xmapsto{\tau_\eta} \psi'$
- (ii) If τ is given by an identity natural transformation, the following subcases can be distinguished:
  - (a) If τ is given by $1_{D\mathsf{U}} : D = D\mathsf{U} \Rightarrow D\mathsf{U}$:

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

    $$\mathsf{next}_s\,\varphi \xmapsto{(1_{\Pi_{f(s)}})_\eta} \mathsf{next}_{f(s)}\,\varphi' \text{ if } \varphi \xmapsto{(\eta_s)_\eta} \varphi'$$

    where $\eta_s : \mathsf{T}'_{f(s)} \Rightarrow \mathsf{T}_s\mathsf{U}$.
  - (c) If τ is given by $1_{\mathsf{F}_1\mathsf{U}\times\mathsf{F}_2\mathsf{U}} : \mathsf{F}_1\mathsf{U} \times \mathsf{F}_2\mathsf{U} \Rightarrow \mathsf{F}_1\mathsf{U} \times \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 \times \mathsf{F}_2)\mathsf{U}$:

    $$[\pi_i]\varphi \xmapsto{(1_{\mathsf{F}_1\mathsf{U}\times\mathsf{F}_2\mathsf{U}})_\eta} [\pi_i]\varphi' \text{ if } \varphi \xmapsto{(1_{\mathsf{F}_i\mathsf{U}})_\eta} \varphi' \text{, with } i \in \{1,2\}$$
  - (d) If τ is given by $1_{\mathsf{F}_1\mathsf{U}+\mathsf{F}_2\mathsf{U}} : \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U} \Rightarrow \mathsf{F}_1\mathsf{U} + \mathsf{F}_2\mathsf{U} = (\mathsf{F}_1 + \mathsf{F}_2)\mathsf{U}$:

    $$[\kappa_i]\varphi \xmapsto{(1_{\mathsf{F}_1\mathsf{U}+\mathsf{F}_2\mathsf{U}})_\eta} [\kappa_i]\varphi' \text{ if } \varphi \xmapsto{(1_{\mathsf{F}_i\mathsf{U}})_\eta} \varphi' \text{, with } i \in \{1,2\}$$
  - (e) If τ is given by $1_{(\mathsf{FU})^D} : (\mathsf{FU})^D \Rightarrow (\mathsf{FU})^D = \mathsf{F}^D\mathsf{U}$:

    $$[ev(d)]\varphi \xmapsto{(1_{(\mathsf{FU})^D})_\eta} [ev(d)]\varphi' \text{ if } \varphi \xmapsto{(1_{\mathsf{FU}})_\eta} \varphi'$$
  - (f) If τ is given by $1_{\mathcal{P}(\mathsf{FU})} : \mathcal{P}(\mathsf{FU}) \Rightarrow \mathcal{P}(\mathsf{FU}) = (\mathcal{P}(\mathsf{F}))\mathsf{U}$:

    $$[\mathcal{P}]\varphi \xmapsto{(1_{\mathcal{P}(\mathsf{FU})})_\eta} [\mathcal{P}]\varphi' \text{ if } \varphi \xmapsto{(1_{\mathsf{FU}})_\eta} \varphi'$$
- (iii) (a) If τ is given by $\alpha : D' \Rightarrow D = D\mathsf{U}$:

  $$d \xmapsto{\alpha_\eta} \bigvee_{\alpha(d')=d} d'$$
  - (b) If τ is given by $d : \mathsf{F} \Rightarrow D = D\mathsf{U}$:

    $$d' \xmapsto{d_\eta} \begin{cases} \top & \text{if } d' = d \\ \bot & \text{if } d' \neq d \end{cases}$$
  - (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 $\mathsf{F}'_i = \mathsf{F}_i\mathsf{U}$:

    $$\varphi \xmapsto{(\pi_i)_\eta} [\pi_i]\varphi' \text{ if } \varphi \xmapsto{(1_{\mathsf{F}_i\mathsf{U}})_\eta} \varphi'$$
  - (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 $i = 1, 2$:

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

(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 $i \in \{1,2\}$:

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

(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 $i = 1,2$:

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

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

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

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

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

(Note that here it is essential that the set D be finite.)

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

$$[\mathcal{P}]\varphi \xmapsto{\mathcal{P}(\zeta)_\eta} [\mathcal{P}]\varphi' \text{ if } \varphi \xmapsto{\zeta_\eta} \varphi'$$

(iv) 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[9]:

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

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 $\eta_s : \mathsf{SForm}(\mathsf{T})_s \rightarrow \mathsf{SForm}(\mathsf{T}')_{f(s)}$, is given by $(1_{\Pi_{f(s)}})_\eta : \mathsf{Form}_{\mathsf{T}}(\Pi_s) \rightarrow \mathsf{Form}_{\mathsf{T}'}(\Pi_{f(s)})$ (where $1_{\Pi_{f(s)}} : \Pi_{f(s)} \Rightarrow \Pi_s\mathsf{U}$).

(i) of Definition 3.3 defines the translations of complex formulae along arbitrary natural transformations τ in terms of the translations of their subformulae along the same natural transformations. (ii) of Definition 3.3 translates modal formulae over T to modal formulae over T', but of a similar kind. This is done by taking $\tau = 1_{\mathsf{F}'}$ and using structural induction on F'. The interesting case here is $\mathsf{F}' = \Pi_{f(s)}$. (iii) of Definition 3.3 translates modal formulae over T to modal formulae over T' along arbitrary natural transformations τ, by considering the various shapes these formulae can take depending on the form of τ. For instance, the translation of a modal 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 any state satisfying it to satisfy the translation of the given formula along $1_{\mathsf{F}_1\mathsf{U}}$. On the other hand, the translation of a modal 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 the

---

[9] This condition ensures that τ_η is only defined *once*, by preventing the definition of τ_η to be based on equalities of form $\tau = \pi_1 \circ \langle\tau,\zeta\rangle$ or $\tau = [\tau,\zeta] \circ \kappa_1$.

formula 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 $1_{\mathsf{F}_1\mathsf{U}}$. If the formula refers to the second coproduct component, its translation does not require anything. The translation of a modal 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, say, φ', and then "extracting" from φ' a formula of type F' which holds in a state f' precisely when φ' holds in the state $\langle f', d\rangle$. Also, the translation of a modal formula φ along $eval_{\mathsf{FU},D} : (\mathsf{FU})^D \times D \Rightarrow \mathsf{FU}$ requires any state $\langle f, d\rangle$ satisfying it to be such that $f(d)$ satisfies the translation of φ along 1_{FU}. Finally, (iv) of Definition 3.3 defines the translations along compositions of natural transformations in terms of the translations along the natural transformations being composed.

The correctness of Definition 3.3 is justified by the following result.

Proposition 3.4 *Let* $(f,\eta) : (S,\mathsf{T}) \to (S',\mathsf{T}')$ *denote a many-sorted cosignature morphism, and let* $(\tau : \mathsf{F}' \Rightarrow \mathsf{F}) \in\| \mathsf{KP}_S \|$ *(hence* $(\tau_{\mathsf{U}} : \mathsf{F}'\mathsf{U} \Rightarrow \mathsf{FU}) \in\| \mathsf{KP}_{S'} \|$*). Then,* $(\tau_{\mathsf{U}})_{1_{\mathsf{T}'}} \circ (1_{\mathsf{FU}})_\eta = (\tau_{\mathsf{U}})_\eta = (1_{\mathsf{F}'\mathsf{U}})_\eta \circ \tau_{1_{\mathsf{T}}}$*:*

$$\begin{tikzcd}
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}) \arrow[r, "(1_{\mathsf{FU}})_\eta"] \arrow[d, "\tau_{1_{\mathsf{T}}}"'] \arrow[dr, "(\tau_{\mathsf{U}})_\eta"] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{FU}) \arrow[d, "(\tau_{\mathsf{U}})_{1_{\mathsf{T}'}}"] \\
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}') \arrow[r, "(1_{\mathsf{F}'\mathsf{U}})_\eta"'] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}'\mathsf{U})
\end{tikzcd}$$

Proof. The statement follows by structural induction on τ. □

Corollary 3.5 *Let* $(f,\eta) : (S,\mathsf{T}) \to (S',\mathsf{T}')$ *denote a many-sorted cosignature morphism, and let* $(\tau_1 : \mathsf{F}_1 \Rightarrow \mathsf{F}) \in\| \mathsf{KP}_S \|$ *(hence* $(\tau_{1\mathsf{U}} : \mathsf{F}_1\mathsf{U} \Rightarrow \mathsf{FU}) \in\| \mathsf{KP}_{S'} \|$*) 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 (iv) of Definition 3.3. Then,* $(\tau_2)_{1_{\mathsf{T}'}} \circ (\tau_{1\mathsf{U}})_\eta = (\tau_{1\mathsf{U}} \circ \tau_2)_\eta = (\tau_2)_\eta \circ (\tau_1)_{1_{\mathsf{T}}}$*:*

$$\begin{tikzcd}
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}) \arrow[r, "(\tau_{1\mathsf{U}})_\eta"] \arrow[d, "(\tau_1)_{1_{\mathsf{T}}}"'] \arrow[dr, "(\tau_{1\mathsf{U}} \circ \tau_2)_\eta"] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}_1\mathsf{U}) \arrow[d, "(\tau_2)_{1_{\mathsf{T}'}}"] \\
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}_1) \arrow[r, "(\tau_2)_\eta"'] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}')
\end{tikzcd}$$

Proof. Definition 3.3 and Proposition 3.4 are used. □

Remark 3.6 The following are consequences of Definition 3.3 and of Corollary 3.5:

- $[\pi_i]\varphi \overset{(\tau_1 \times \tau_2)_\eta}{\longmapsto} [\pi_i]\varphi'$ 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 $\varphi \overset{\tau_\eta}{\longmapsto} \varphi'$
- $[ev(d)]\varphi \overset{((\mathsf{FU})^\alpha)_\eta}{\longmapsto} \psi \equiv [ev(\alpha(d))]\varphi'$ if $\varphi \overset{(1_{\mathsf{FU}})_\eta}{\longmapsto} \varphi'$

(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 2.3.)

The translation of formulae along cosignature morphisms is compatible with the equalities (i)–(iii) in Remark 2.3, in a sense made precise below.

Proposition 3.7 *Let $(f, \eta) : (S, \mathsf{T}) \to (S', \mathsf{T}')$ denote a many-sorted cosignature morphism. Then, the following hold up to semantic equivalence*[10] *:*

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

$$\begin{tikzcd}
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}_i) \arrow[r, "(\pi_i)_{1_\mathsf{T}}"] \arrow[rr, bend right, "(\tau_i)_\eta"'] & \mathsf{Form}_{\mathsf{T}}(\mathsf{F}_1 \times \mathsf{F}_2) \arrow[r, "\langle \tau_1, \tau_2 \rangle_\eta"] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F})
\end{tikzcd}$$

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

$$\begin{tikzcd}
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}) \arrow[r, "{[\tau_1, \tau_2]_\eta}"] \arrow[rr, bend right, "(\tau_i)_\eta"'] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}_1 + \mathsf{F}_2) \arrow[r, "(\kappa_i)_{1_{\mathsf{T}'}}"] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}_i)
\end{tikzcd}$$

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

$$\begin{tikzcd}
\mathsf{Form}_{\mathsf{T}}(\mathsf{F}) \arrow[r, "(eval_{\mathsf{F},D})_{1_\mathsf{T}}"] \arrow[rr, bend right, "\tau_\eta"'] & \mathsf{Form}_{\mathsf{T}}(\mathsf{F}^D \times D) \arrow[r, "(\tau^* \times 1_D)_\eta"] & \mathsf{Form}_{\mathsf{T}'}(\mathsf{F}' \times D)
\end{tikzcd}$$

Proof. The statement follows directly from Definition 3.3. □

Definition 3.3 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.

Example 3.8 Given $A \in |\mathsf{Set}|$, A-labelled transition systems of finite depth are specified using the endofunctor $\mathsf{T_{LFTS}} : \mathsf{Set} \to \mathsf{Set}$ given by $\mathsf{T_{LFTS}} = \mathcal{P}(A \times \mathsf{Id}) \times \mathbb{N}$, together with the translations of the modal formulae defining unlabelled transition systems of finite depth (see Example 2.14) along the cosignature morphism defined by the natural transformation $\eta ::= \mathcal{P}(\pi_2) \times \mathbb{N} : \mathcal{P}(A \times \mathsf{Id}) \times \mathbb{N} \Rightarrow \mathcal{P}(\mathsf{Id}) \times \mathbb{N}$. Specifically, these modal formulae are:

$$\mathsf{next}\,[\pi_2]0 \leftrightarrow \mathsf{next}\,[\pi_1][\mathcal{P}][\pi_2]\bot$$
$$\mathsf{next}\,[\pi_2](n+1) \leftrightarrow \mathsf{next}\,[\pi_1](\langle\mathcal{P}\rangle[\pi_2]\mathsf{next}\,[\pi_2]n \wedge [\mathcal{P}][\pi_2]\mathsf{next}\,[\pi_2](0 \vee \ldots \vee n)), \;\; n \in \mathbb{N}$$

or, after renaming $\mathsf{next}\,[\pi_1][\mathcal{P}][\pi_2]$ to `[succ]`, $\mathsf{next}\,[\pi_1]\langle\mathcal{P}\rangle[\pi_2]$ to `<succ>`, and $\mathsf{next}\,[\pi_2]$ to `[depth]`, and using the distributivity of $\mathsf{next}\,[\pi_1]$ over $\wedge$ w.r.t. semantic equivalence:

$$\texttt{[depth]}\,0 \leftrightarrow \texttt{[succ]}\bot$$
$$\texttt{[depth]}(n+1) \leftrightarrow \texttt{<succ>[depth]}n \wedge \texttt{[succ]}\,\texttt{[depth]}(0 \vee \ldots \vee n), \;\; n \in \mathbb{N}$$

---

[10] Note that Corollary 3.5 can not be applied here.

For instance, the fact that the translation of the modal formula $\mathsf{next}\,[\pi_1][\mathcal{P}]\bot$ along the cosignature morphism defined by η is given by $\mathsf{next}\,[\pi_1][\mathcal{P}][\pi_2]\bot$ can be inferred as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\bot \xmapsto{(1_{\mathsf{Id}})_\eta} \bot}{\bot \xmapsto{(\pi_2)_\eta} [\pi_2]\bot}}{[\mathcal{P}]\bot \xmapsto{\mathcal{P}(\pi_2)_\eta} [\mathcal{P}][\pi_2]\bot}}{[\pi_1][\mathcal{P}]\bot \xmapsto{\eta_\eta} [\pi_1][\mathcal{P}][\pi_2]\bot}}{\mathsf{next}\,[\pi_1][\mathcal{P}]\bot \xmapsto{(1_{\mathsf{Id}})_\eta} \mathsf{next}\,[\pi_1][\mathcal{P}][\pi_2]\bot}$$

It is also worth noting that $\mathsf{T}_{\mathtt{LFTS}}$ can be obtained by taking the pushout in Cosign of $\mathcal{P}(\pi_2) : \mathsf{T}_{\mathtt{TS}} \to \mathsf{T}_{\mathtt{LTS}}$ and $\pi_1 : \mathsf{T}_{\mathtt{TS}} \to \mathsf{T}_{\mathtt{FTS}}$. For, the following is a pullback diagram in KP_1:

$$\begin{tikzcd}
 & \mathcal{P}(A \times \mathsf{Id}) \times \mathbb{N} \arrow[dl, Rightarrow, dashed, "\pi_1"'] \arrow[dr, Rightarrow, dashed, "\mathcal{P}(\pi_2)\times 1_{\mathbb{N}}"] & \\
\mathcal{P}(A \times \mathsf{Id}) \arrow[dr, Rightarrow, "\mathcal{P}(\pi_2)"'] & & \mathcal{P}(\mathsf{Id}) \times \mathbb{N} \arrow[dl, Rightarrow, "\pi_1"] \\
 & \mathcal{P}(\mathsf{Id}) &
\end{tikzcd}$$

Example 3.9 Lists whose elements belong to a set E and whose size does not exceed $m \in \mathbb{N}$ are specified using the endofunctor $\mathsf{T}_{\mathtt{mLIST}} : \mathsf{Set} \to \mathsf{Set}$ given by $\mathsf{T}_{\mathtt{mLIST}} = \mathsf{T}_{\mathtt{LIST}} \times \{0, \ldots, m\}$ (with $\mathsf{T}_{\mathtt{LIST}}$ being as in Example 2.15), together with the translation of the modal formula defining lists over E (see Example 2.15) along the cosignature morphism defined by $\eta ::= \pi_1 : \mathsf{T}_{\mathtt{mLIST}} \Rightarrow \mathsf{T}_{\mathtt{LIST}}$, and together with the following modal formulae:

$$\begin{gathered}
\mathsf{next}\,[\pi_2]0 \leftrightarrow \mathsf{next}\,[\pi_1][\pi_2]\langle\kappa_1\rangle\top \\
\mathsf{next}\,[\pi_2](n+1) \leftrightarrow \mathsf{next}\,[\pi_1][\pi_2]\langle\kappa_2\rangle\mathsf{next}\,[\pi_2]n,\ \ n \in \{0, \ldots, m-1\}
\end{gathered} \tag{1}$$

In particular, the translation of the modal formula defining lists over E along the cosignature morphism defined by η is obtained as follows:

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

After renaming $\mathsf{next}\,[\pi_1][\pi_1]\langle\kappa_1\rangle$, $\mathsf{next}\,[\pi_1][\pi_1]\langle\kappa_2\rangle$, $\mathsf{next}\,[\pi_1][\pi_2]\langle\kappa_1\rangle$, $\mathsf{next}\,[\pi_1][\pi_2]\langle\kappa_2\rangle$ and $\mathsf{next}\,[\pi_2]$ to `<headF>`, `<headS>`, `<tailF>`, `<tailS>` and respectively `[size]`, the specification of lists of size at most m becomes:

$$\begin{aligned} \texttt{<headF>}\top &\leftrightarrow \texttt{<tailF>}\top \\ \texttt{[size]}0 &\leftrightarrow \texttt{<tailF>}\top \\ \texttt{[size]}(n+1) &\leftrightarrow \texttt{<tailS>[size]}n,\ n \in \{0,\ldots,m-1\} \end{aligned} \tag{2}$$

Example 3.10 A specification of arrays of size m can be obtained by suitably extending the specification of lists of size not exceeding m given in Example 3.9. Specifically, one can consider the cosignature $(\mathsf{Set}^{\{\texttt{mList},\texttt{Array}\}}, \mathsf{T})$, with the components of the endofunctor T being given by $\mathsf{T}_{\texttt{mList}} = \mathsf{T}_{\texttt{mLIST}}\Pi_1 \times (1+E)^{\{1,\ldots,m\}}$ and respectively $\mathsf{T}_{\texttt{Array}} = \Pi_1 \times E^{\{1,\ldots,m\}}$. (The second component of $\mathsf{T}_{\texttt{mList}}$ specifies a list observer which takes an argument $p \in \{1,\ldots,m\}$ and returns the pth element of the list in case this element exists, or $\bot$ otherwise. Also, the second component of $\mathsf{T}_{\texttt{Array}}$ specifies an array observer which takes an argument $p \in \{1,\ldots,m\}$ and returns the pth element of the array.) The specification of arrays of size m then consists of the following formulae:

$$\begin{aligned} \texttt{<headF>}\top &\leftrightarrow \texttt{<tailF>}\top \\ \texttt{[size]}0 &\leftrightarrow \texttt{<tailF>}\top \\ \texttt{[size]}(n+1) &\leftrightarrow \texttt{<tailS>[size]}n, && n \in \{0,\ldots,m-1\} \\ \texttt{<elemF(1)>}\top &\leftrightarrow \texttt{<headF>}\top \\ \texttt{<elemS(1)>}e &\leftrightarrow \texttt{<headS>}e, && e \in E \\ \texttt{<elemF(}p+1\texttt{)>}\top &\leftrightarrow \texttt{<tailF>}\top \vee \texttt{<tailS><elemF(}p\texttt{)>}\top, && p \in \{1,\ldots,m-1\} \\ \texttt{<elemS(}p+1\texttt{)>}e &\leftrightarrow \texttt{<tailS><elemS(}p\texttt{)>}e, && p \in \{1,\ldots,m-1\}, \\ & && e \in E \end{aligned}$$

of type `mList`, together with the following formula:

$$\texttt{[get(}p\texttt{)]}e \leftrightarrow \texttt{[list]<elemS(}p\texttt{)>}e, \quad p \in \{1,\ldots,m\},\ e \in E \tag{3}$$

o type `Array`, using the following abbreviations:

$$\begin{aligned} \texttt{<headF>} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_1][\pi_1][\pi_1]\langle\kappa_1\rangle \\ \texttt{<headS>} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_1][\pi_1][\pi_1]\langle\kappa_2\rangle \\ \texttt{<tailF>} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_1][\pi_1][\pi_2]\langle\kappa_1\rangle \\ \texttt{<tailS>} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_1][\pi_1][\pi_2]\langle\kappa_2\rangle \\ \texttt{[size]} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_1][\pi_2] \\ \texttt{<elemF(}p\texttt{)>} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_2][ev(p)]\langle\kappa_1\rangle \\ \texttt{<elemS(}p\texttt{)>} &::= \mathsf{next}_{\texttt{mList}}\,[\pi_2][ev(p)]\langle\kappa_2\rangle \\ \texttt{[list]} &::= \mathsf{next}_{\texttt{Array}}\,[\pi_1] \\ \texttt{[get(}p\texttt{)]} &::= \mathsf{next}_{\texttt{Array}}\,[\pi_2][ev(p)] \end{aligned}$$

In particular, the formula in (3) states 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. It is worth noting that this formula actually constrains the lists used to represent arrays to lists of size *exactly* m. The inclusion of the cosignature specifying lists of size at most m into the (two-sorted) cosignature specifying arrays of size m is then captured by a cosignature morphism (f, η), with $f : \{\mathtt{mList}\} \to \{\mathtt{mList}, \mathtt{Array}\}$ being the inclusion function, and with $\eta : \mathsf{U}\mathsf{T} \Rightarrow \mathsf{T}_{\mathtt{mLIST}}\mathsf{U}$ being given by $\pi_1 : \mathsf{T}_{\mathtt{mList}} \Rightarrow \mathsf{T}_{\mathtt{mLIST}}\Pi_1$ (where $\mathsf{U} : \mathsf{Set}^{\{\mathtt{mList},\mathtt{Array}\}} \to \mathsf{Set}$ is given by Π_1). The translation of the modal formulae defining lists of size at most m along this cosignature morphism leaves the formulae in (2) unchanged. Note, however, that the meanings of `<headF>`, `<tailF>`, ... change when moving from $\mathsf{T}_{\mathtt{mLIST}}$ to T, so for instance the formulae in (1) do in fact change when moving from $\mathsf{T}_{\mathtt{mLIST}}$ to T.

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

Proposition 3.11 *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* $\gamma = \eta_{C'} \circ \mathsf{U}\gamma' : \mathsf{U}C' \to \mathsf{T}\mathsf{U}C'$. *Then,* $\tau_{C'}^{-1}(\llbracket\varphi\rrbracket_{\mathsf{F}}^{\gamma}) = \llbracket\tau_\eta(\varphi)\rrbracket_{\mathsf{F}'}^{\gamma'}$ *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 3.3) are treated similarly.

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

$$\llbracket \mathsf{next}_s\, \varphi \rrbracket_{\Pi_s}^{\gamma} = \gamma_s^{-1}(\llbracket\varphi\rrbracket_{\mathsf{T}_s}^{\gamma}) = (\gamma'_{f(s)})^{-1}(\eta_{s,C'}^{-1}(\llbracket\varphi\rrbracket_{\mathsf{T}_s}^{\gamma})) =$$
$$(\gamma'_{f(s)})^{-1}(\llbracket(\eta_s)_\eta(\varphi)\rrbracket_{\mathsf{T}'_{f(s)}}^{\gamma'}) = \llbracket \mathsf{next}_{f(s)}\, (\eta_s)_\eta(\varphi)\rrbracket_{\Pi_{f(s)}}^{\gamma'} =$$
$$\llbracket(1_{\Pi_{f(s)}})_\eta(\mathsf{next}_s\, \varphi)\rrbracket_{\Pi_{f(s)}}^{\gamma'} = \llbracket\tau_\eta(\mathsf{next}_s\, \varphi)\rrbracket_{\Pi_{f(s)}}^{\gamma'}$$

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

$$\tau_{C'}^{-1}(\llbracket d\rrbracket_D^{\gamma}) = \tau_{C'}^{-1}(\{d\}) = \{\, d' \in D' \mid \alpha(d') = d \,\} = \bigcup_{\alpha(d')=d} \{d'\} =$$
$$\bigcup_{\alpha(d')=d} \llbracket d'\rrbracket_{D'}^{\gamma'} = \llbracket \bigvee_{\alpha(d')=d} d'\rrbracket_{D'}^{\gamma'} = \llbracket\alpha_\eta(d)\rrbracket_{D'}^{\gamma'} = \llbracket\tau_\eta(d)\rrbracket_{D'}^{\gamma'}$$

- 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 $\mathsf{F}'_i = \mathsf{F}_i\mathsf{U}$:

$$\tau_{C'}^{-1}(\llbracket\varphi\rrbracket_{\mathsf{F}_i}^{\gamma}) = \pi_i^{-1}(\llbracket\varphi\rrbracket_{\mathsf{F}_i}^{\gamma}) = \pi_i^{-1}(\llbracket(1_{\mathsf{F}_i\mathsf{U}})_\eta(\varphi)\rrbracket_{\mathsf{F}_i\mathsf{U}}^{\gamma'}) =$$
$$\llbracket[\pi_i](1_{\mathsf{F}_i\mathsf{U}})_\eta(\varphi)\rrbracket_{\mathsf{F}'_1\times\mathsf{F}'_2}^{\gamma'} = \llbracket(\pi_i)_\eta(\varphi)\rrbracket_{\mathsf{F}'_1\times\mathsf{F}'_2}^{\gamma'} = \llbracket\tau_\eta(\varphi)\rrbracket_{\mathsf{F}'_1\times\mathsf{F}'_2}^{\gamma'}$$

- 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 $\{l\} = \{1, 2\} \setminus \{j\}$:

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

  - If $j \neq i$:

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

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

$$\tau_{C'}^{-1}(\llbracket [ev(d)]\varphi \rrbracket^{\gamma}_{\mathsf{F}^D}) = (\zeta^*_{C'})^{-1}(\{\, f : D \to \mathsf{FU}C' \mid f(d) \in \llbracket \varphi \rrbracket^{\gamma}_{\mathsf{F}} \,\}) =$$
$$\{\, f' \in \mathsf{F}'C' \mid \zeta^*_{C'}(f')(d) \in \llbracket \varphi \rrbracket^{\gamma}_{\mathsf{F}} \,\} = \{\, f' \in \mathsf{F}'C' \mid \zeta_{C'}(f', d) \in \llbracket \varphi \rrbracket^{\gamma}_{\mathsf{F}} \,\} =$$
$$\langle 1_{\mathsf{F}'}, d \rangle_{C'}^{-1}(\zeta_{C'}^{-1}(\llbracket \varphi \rrbracket^{\gamma}_{\mathsf{F}})) = \llbracket \langle 1_{\mathsf{F}'}, d \rangle_{1_{\mathsf{T}'}}(\zeta_\eta(\varphi)) \rrbracket^{\gamma'}_{\mathsf{F}'} = \llbracket \tau_\eta([ev(d)]\varphi) \rrbracket^{\gamma'}_{\mathsf{F}'}$$

□

In particular, the interpretations of state formulae in the T-reducts of T'-coalgebras coincide with the interpretations of their translations in the original T'-coalgebras – this follows by taking $\tau = 1_{\Pi_{f(s)}}$ with $s \in S$.

We are now ready for our main result.

Theorem 3.12 $(\mathsf{Cosign}, \mathsf{Coalg}, \mathsf{SForm}, \models)$ *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 $\varphi \in \mathsf{SForm}(\mathsf{T})$:

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

Showing that the above holds can be reduced to showing that, given η and $\langle C', \gamma' \rangle$, $\llbracket \varphi \rrbracket^{\gamma}_{\Pi_s} = \llbracket \eta_s(\varphi) \rrbracket^{\gamma'}_{\Pi_{f(s)}}$ holds for any $\varphi \in \mathsf{SForm}(\mathsf{T})_s$ and any $s \in S$ (where $\gamma = \eta_{C'} \circ \mathsf{U}\gamma'$). For, then one can reason as follows: $\langle C', \gamma' \rangle \models \eta_s(\varphi) \;\Leftrightarrow\; \llbracket \eta_s(\varphi) \rrbracket^{\gamma'}_{\Pi_{f(s)}} = C'_{f(s)} \;\Leftrightarrow\; \llbracket \varphi \rrbracket^{\gamma}_{\Pi_s} = (\mathsf{U}C')_s \;\Leftrightarrow\; \mathsf{U}_\eta \langle C', \gamma' \rangle \models \varphi$ for any $\varphi \in \mathsf{SForm}(\mathsf{T})_s$ and any $s \in S$. But the previous claim follows from Proposition 3.11, namely by taking $\mathsf{F} = \Pi_s$, $\mathsf{F}' = \Pi_{f(s)}$ and $\tau = 1_{\Pi_{f(s)}}$ for $s \in S$. This concludes the proof. □

4 Conclusions

The main contributions of the paper can be summarised as follows. First, a generalisation of the modal logic described in [4] to categories of sorted sets was presented. This generalisation was introduced in such a way as to allow one to formally capture ways of moving from one Kripke polynomial endofunctor to another. Natural transformations arising from the structure of such endofunctors were then used to define a category of cosignatures, whose arrows were equipped with (backward) translations between the corresponding categories of coalgebras, as well as with (forward) translations between the corresponding sets of formulae. Finally, the resulting framework was shown to be an institution, capturing both refinement and encapsulation relations between coalgebraic types.

Acknowledgements

The author would like to thank the anonymous referees for their comments and suggestions.

References

[1] C. Cîrstea. On specification logics for algebra-coalgebra structures: Reconciling reachability and observability. In M. Nielsen and U. Engberg, editors, *Foundations of Software Science and Computation Structures*, volume 2303 of *Lecture Notes in Compute Science*, pages 82–97. Springer, 2002.

[2] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

[3] R. Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1992.

[4] B. Jacobs. Many-sorted coalgebraic modal logic: A model-theoretic study. *Theoretical Informatics and Applications*, 35(1):31–59, 2001.

[5] A. Kurz. Specifying coalgebras with modal logic. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 57–71. Elsevier Science, 1998.

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

[7] M. Rößiger. Coalgebras and modal logic. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 299–320. Elsevier Science, 2000.

[8] J.J.M.M. Rutten. A calculus of transition systems (towards universal coalgebra). Technical Report CS-R9503, CWI, 1995.

[9] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.