Modalities in the Stone age: *A comparison of coalgebraic logics*

Alexander Kurz[1,3]

Computer Science Departament
University of Leicester
Leicester, UK

Raul Leal[2,4]

ILLC
Universiteit van Amsterdam
Amsterdam, The Netherlands

Abstract

Coalgebra develops a general theory of transition systems, parametric in a functor T; the functor T specifies the possible one-step behaviours of the system. A fundamental question in this area is how to obtain, for an arbitrary functor T, a logic for T-coalgebras. We compare two existing proposals, Moss's coalgebraic logic and the logic of all predicate liftings, by providing one-step translations between them, extending the results in [34] by making systematic use of Stone duality. Our main contribution then is a novel coalgebraic logic, which can be seen as an equational axiomatization of Moss's logic. The three logics are equivalent for a natural but restricted class of functors. We give examples showing that the logics fall apart in general. Finally, we argue that the quest for a generic logic for T-coalgebras is still open in the general case.

Keywords: coalgebra, coalgebraic logic, Stone duality, predicate liftings, Moss-modality, nabla-modality

1 Introduction

When Aczel ([2, Chapter 7,8], [3]) introduced the idea of coalgebras for a functor T as a generalisation of transition systems, he made three crucial observations: (i) coalgebras come with a canonical notion of observational or *behavioural equivalence* (induced by the functor T); (ii) this notion of behavioural equivalence generalizes the notion of *bisimilarity* from computer science and modal logic; (iii) any 'domain equation' $X \cong TX$ has a canonical solution, namely the *final* coalgebra, which is fully abstract wrt behavioural equivalence.

[1] Thanks to the anonymous referees for valuable comments and suggestions.

[2] The research of this author has been made possible by VICI grant 639.073.501 of the Netherlands Organization for Scientific Research (NWO).

[3] Email: ak155@mcs.le.ac.uk

[4] Email: R.A.LealRodriguez@uva.nl

This idea of a type of dynamic systems being represented by a functor T and an individual system being a T-coalgebra, led Rutten [40] to the theory of universal coalgebra which, parametrized by T, applies in a uniform way to a large class of different types of systems. In particular, final semantics and the associated proof principle of coinduction (which are dual to initial algebra semantics and induction) find their natural place here.

These ideas have been proved very successful. Coalgebras encompass such diverse systems as, for example, labelled transition systems [2], deterministic automata [39], π-calculus processes [18], HD-automata [17], stochastic systems [15], neighborhood frames [19].

Very early on in this endeavour the following question arose. If universal coalgebra can cover a wide range of models of computation uniformly and parametric in the type-functor T, can the same be done for logics for coalgebras? The first positive answer was given by Moss [36]. His fascinating idea was, roughly, to take T itself as constructing a modality. More precisely, if $\mathcal{M}$ is the set of formulas of his language and $\alpha \in T\mathcal{M}$ then $\nabla\alpha \in \mathcal{M}$.

In the case of the power-set functor $\mathcal{P}$, this modality, denoted as ∇, can be defined using the standard box and diamond: With $\alpha \in \mathcal{PM}$ a set of formulas, the formula $\nabla\alpha$ can be seen as an abbreviation $\nabla\alpha = \Box \bigvee \alpha \wedge \bigwedge \Diamond\alpha$, where $\Diamond\alpha$ denotes the set $\{\Diamond a \mid a \in \alpha\}$.

Independently of Moss's work, Janin and Walukiewicz [22] already observed that the connectives ∇ and $\vee$ may replace the connectives $\Box, \Diamond, \wedge, \vee$. This observation, which is closely linked to fundamental automata-theoretic constructions, lies at the heart of the theory of the modal μ-calculus, and has many applications, see for instance [12,41]. Kupke & Venema [28] generalized the link between fix-point logics and automata theory to the coalgebraic level of generality by showing that many fundamental results in automata theory are really theorems of universal coalgebra.

Moss's connective ∇ is not easily studied with standard methods of modal logic. Subsequently [29] proposed a standard modal logic for a restricted class of coalgebras and Pattinson [37] discovered how to describe modal logics for coalgebras in general via *predicate liftings*. The logic $\mathcal{L}$ of all predicate liftings was first investigated by Schröder [42] and Klin [25].

The second author's [34] started a systematic investigation of the relationship of Moss's logic $\mathcal{M}$ and the logic $\mathcal{L}$ of all predicate liftings. In particular, [34] introduced a special notion of predicate liftings, the *singleton liftings*, and observed that 1) they generate all other predicate liftings and 2) they can be translated into Moss's logic for all Kripke polynomial functors.

We continue this line of research and summarize the contributions of this paper as follows:

- Coalgebraic logics can extend different underlying propositional logics. We investigate how this choice influences translations between Moss's logic and logic with predicate liftings.
- If the underlying logic is classical, i.e. based on Boolean algebras, we
  - improve on the result of [34] by showing that all singleton liftings for any functor T can be translated into Moss's logic, establishing a one-step translation $\mathcal{L} \longrightarrow \mathcal{M}$,
  - give a simple description of a one-step translation of $\mathcal{M}$ to $\mathcal{L}$,
  - show that all expressive coalgebraic logics for a finitary functor that preserves finite sets are mutually translatable.
- We show that Moss's logic can be given a more standard equational (or modal) logic

style by replacing the modal operator ∇ by a set of conventional modal operators. This is based on the well-known fact that any set-functor T has a presentation by operations and equations [6].

2 Notation and Preliminaries

In this section we will introduce the terminology and notation to be used in the paper. We assume the reader to be familiar with the basics of category theory and classical propositional logic. Familiarity with transition systems and modal logic will be helpful.

2.1 Categories

We write Set for the category of sets, functions and usual composition; we identify the natural number n with the set $\{0, 1, \ldots, n-1\}$. BA denotes the category of Boolean algebras and Boolean homomorphisms and usual composition. BA_ω is the category of finite Boolean algebras and all Boolean homomorphisms between them, and Set_ω is the category of finite sets and all functions. The category of distributive lattices and lattice homomorphisms is denoted DL.

2.2 Functors

In the following, we fix our notation for the functors that will appear.

(i) We use $\mathcal{Q} : \mathsf{Set} \longrightarrow \mathsf{Set}^{op}$ for the contra-variant power set functor. This functor maps a set X to its power set and a function $f : X \longrightarrow Y$ to its inverse image. $\mathcal{Q}$ is intended to remind us of 2, because of $\mathcal{Q}X = 2^X$.

(ii) $\mathcal{P} : \mathsf{Set} \longrightarrow \mathsf{Set}$ denotes the covariant power set functor. This functor maps a set to its power set and $f : X \longrightarrow Y$ to the function $\mathcal{P}f : \mathcal{P}X \longrightarrow \mathcal{P}Y$ which maps a subset of X to its direct image under f.

(iii) Given a fixed set A, we write $(-)^A$ for the exponential functor. This functor maps a set X to the set of functions from A to X, denoted by X^A. A function $f : X \longrightarrow Y$ is mapped to the function $f^A : X^A \longrightarrow Y^A$ mapping $h \in X^A$ to $f \circ h$; if there is no risk for confusion, we write $f \circ -$ for f^A.

(iv) We write $\mathcal{B}_\mathbb{N} : \mathsf{Set} \longrightarrow \mathsf{Set}$ for the finite multiset functor: It maps a set X to $\mathcal{B}_\mathbb{N}X$ which consists of all maps ('bags') $B : X \longrightarrow \mathbb{N}$ with finite support; for $f : X \longrightarrow Y$, the function $\mathcal{B}_\mathbb{N}(f)$ maps $B : X \longrightarrow \mathbb{N}$ to the function $\mathcal{B}_\mathbb{N}(f)(B) : Y \longrightarrow \mathbb{N}$ given by $y \mapsto \sum_{x \in f^{-1}(\{y\})} B(x)$.

(v) The finite distribution functor $\mathcal{D}$ follows the same idea of the finite multiset functor: A set X is mapped to $\mathcal{D}X$, which is the set of probability distributions, i.e. functions $\mu : X \longrightarrow [0,1]$ such that $\sum_{x \in X} \mu(x) = 1$, with finite support. Similarly, $\mathcal{D}_\leq$ denotes the subdistribution functor, which maps X to $\{\mu : X \longrightarrow [0,1] \mid \mu$ has finite support and $\Sigma_{x \in X} \mu(x) \leq 1\}$; on functions, both functors act like $\mathcal{B}_\mathbb{N}$.

Given a set endofunctor T and a set X, we keep the following conventions: We use φ, ψ for subsets of X. The letters α, β are used for the elements of $T(X)$. We write A for subsets of $T(X)$, i.e. elements of $\mathcal{P}T(X)$ or $\mathcal{Q}T(X)$. Finally, we use Φ for entities in $T\mathcal{P}X$ or in $T\mathcal{Q}(X)$.

The following class of functors will be an important source of examples.

Definition 2.1 A *Kripke polynomial functor* [38], or KPF for short, is built according to the following grammar

$$T ::= Id \mid K_C \mid (-)^A \mid \mathcal{P} \mid T+T \mid T\times T \mid T\circ T$$

where Id is the identity functor, K_C is the constant functor that maps all sets to the finite set C, $(-)^A$ is the exponential functor for a finite set A, i.e. X^A is the set of functions from A to X; and $\mathcal{P}$ is the covariant powerset functor. Functors that are built without using $\mathcal{P}$ are called *polynomial functors*.

Some of the results of the paper will only hold for functors T that **preserve finite sets**, i.e. map finite sets to finite sets. Notice all Kripke polynomial functors, as defined above, preserve finite sets whereas the multiset functor and finite distributions functor do not.

The following definition will be particularly useful in connection with Moss's logic, see Section 3.3.1.

Definition 2.2 A functor $T : \mathsf{Set} \longrightarrow \mathsf{Set}$ is *standard* if T preserves inclusions and the equalizer $0 \longrightarrow 1 \rightrightarrows 2$. Under these assumptions we can define the finitary version of T by $T_\omega X = \bigcup\{TY \mid Y \subseteq X, Y \text{ finite}\}$. A standard functor is said to be *finitary* iff $T = T_\omega$.

For example, $\mathcal{P}$ is standard and $\mathcal{P}_\omega X$ is the set of finite subsets of X. An important property of finitary functors, is that they preserve directed or, equivalently, filtered colimits. In fact, in a general category this is used as the definition of finitary functor (see e.g. [4]).

In all our investigations we can always assume that T is standard without loss of generality. Indeed, given any T we can define $T'X = TX$ for $X \neq 0$ and $T0$ as the equaliser $T0 \longrightarrow T1 \rightrightarrows T2$. Further, given T' we can find a naturally isomorphic T'' that preserves inclusions. The details can be found in [6], but the important point for us is that the categories of T-coalgebras and T''-coalgebras are (concretely) isomorphic.

As an illustration of the previous situation consider the functor $(-)^2$. This functor is not standard because functions can only be equal if their codomains are equal. However, $(-)^2$ is isomorphic to $Id\times Id$ which is standard. A similar remark applies to $\mathcal{B}, \mathcal{D}, \mathcal{D}_{\leq}$. For example, $\mathcal{D}$ becomes standard if we replace the $\mu : X \longrightarrow [0,1]$ by $\{(x,\mu(x)) \mid \mu(x) \neq 0\}$.

2.3 Relation Lifting

We write composition of relations $R \subseteq X\times Y$, $R' \subseteq Y \times Z$ as $R;R'$ and the converse of a relation as R^o.

Definition 2.3 Given a binary relation $R \subseteq X \times Y$ with projections $X \xleftarrow{p_1} R \xrightarrow{p_2} Y$, the *relation lifting* $\overline{T}(R) \subseteq TX \times TY$ of R is the set

$$\overline{T}(R) = \{(t,t') \in TX\times TY \mid (\exists r \in TR)(Tp_1(r) = t \text{ and } Tp_2(r) = t'\}.$$

We identify functions with their graphs. Using this we can show $\overline{T}(R) = (Tp_1)^o; Tp_2$. Moreover, $\overline{T}(R)$ can also be characterised as the image of $TR \xrightarrow{\langle Tp_1,Tp_2\rangle} TX\times TY$. Here are some concrete examples.

Example 2.4 (i) In the case of $T = Id$, for every relation R we have $\overline{T}(R) = R$.

(ii) For $T = \mathcal{P}$, the lifting of a relation $R \subseteq X \times Y$ is the set

$$\overline{T}(R) = \Big\{(\varphi, \psi) \in \mathcal{P}(X) \times \mathcal{P}(Y) \mid (\forall x \in \varphi)(\exists y \in \psi)(xRy) \wedge (\forall x \in \psi)(\exists y \in \varphi)(xRy)\Big\}$$

Relationlifting is closely related to bisimulation. A binary relation B between Kripke frames (X, R_0) and (Y, R_1) is a bisimulation iff $(R_0[x], R_1[y]) \in \overline{\mathcal{P}}(B)$ for all $(x, y) \in B$; where $R[x]$ denoted the set of R-successors of x.

(iii) Using the finite distribution functor the lifting of a relation $R \subseteq X \times Y$ can be described as follows: Recall the illustration after Definition 2.2. A distribution $\mu : X \longrightarrow [0, 1]$, with finite support, can be seen as a finite list $\{(x_i, p_i) \mid i \in n\}$; the idea is to only consider states that have non zero probability; with this in mind, we read the pair (x_i, p_i) as $\mu(x_i) = p_i$. Using this perspective, we see that $\{(x_i, p_i) \mid i \in n\}\overline{D}(R)\{(y_j, q_j) \mid j \in m\}$ holds iff there exists $(r_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$, $r_{ij} \in [0, 1]$ such that $\neg(x_i R y_j) \Rightarrow (r_{ij} = 0)$ and $\sum_i r_{ij} = q_j$ and $\sum_j r_{ij} = p_i$. As in the previous item, relation lifting is related to bisimulation; in [14] a presentation like the one above is used to describe bisimulation of probabilistic systems.

The process described in Definition 2.3 determines a function $\overline{T}$ mapping relations to relations. In case T preserves weak pullbacks, $\overline{T}$ is a functor $\overline{T} : \mathsf{Rel} \longrightarrow \mathsf{Rel}$, where Rel is the category with sets as objects and relations as arrows. It is known that T preserves weak pullbacks iff $\overline{T} : \mathsf{Rel} \longrightarrow \mathsf{Rel}$ is a functor iff $\overline{T}(R \circ S) = \overline{T}(R) \circ \overline{T}(S)$. A proof of this fact appears in [7] although it is not explicitly stated there.

Proposition 2.5 *If T preserves weak-pullbacks, then*

(i) $T(\{-\}); T(\in^o) = \mathrm{id}$ *where* $\{-\}_X : X \longrightarrow \mathcal{Q}X$, $x \mapsto \{x\}$ *and* $\in_X \subseteq X \times \mathcal{Q}X$ *is the membership relation,*

(ii) *the map* $\boldsymbol{\nabla} : T\mathcal{Q} \longrightarrow \mathcal{Q}T$, $\Phi \mapsto \{\alpha \in TX \mid \alpha\ \overline{T}(\in_X)\ \Phi\}$ *is natural.*

The first item is immediate from $\overline{T}$ preserving composition; this property plays a crucial role in [27]. The second item is essentially the observation that Moss's logic is invariant under bisimilarity [36].

2.4 Stone Duality

From the general theory of Stone duality [23,13,43], we mainly need that Set_ω and BA_ω are dually equivalent categories. In detail, the contravariant powerset functor $\mathcal{Q} : \mathsf{Set} \longrightarrow \mathsf{Set}^{op}$ can be seen as a functor $P : \mathsf{Set} \longrightarrow \mathsf{BA}^{op}$ to Boolean algebras. It has a right adjoint $S : \mathsf{BA}^{op} \longrightarrow \mathsf{Set}$, which maps a Boolean algebra to its set of ultrafilters (an ultrafilter is a maximal consistent propositional theory). On maps, both functors map a function f to its inverse image f^{-1}. Now, restricting P and S to Set_ω and BA_ω, the adjunction becomes an equivalence. The following well-known proposition will be needed.

Proposition 2.6 (i) *Every Boolean algebra homomorphism $h : PX \longrightarrow PY$, where X is finite, is the inverse image of a function $f : Y \longrightarrow X$.*

(ii) *The free Boolean algebra on the finite set n is given by $P\mathcal{Q}n$.*

To recall the argument: Since h is a homomorphism, it preserves finite meets. Because X is finite, h actually preserves all meets in PX. Therefore, since PX is complete, h has a left adjoint g; since $g(b)$ is a join of atoms (singletons) and h preserves joins and $b \leq h(g(b))$, it follows that $g(b)$ is an atom if b is an atom; thus we can restrict g to a map $f : Y \longrightarrow X$ and since every element of PY is a join of atoms, we have that $g = \mathcal{P}f$ is direct image, which implies $h = f^{-1}$.

2.5 Varieties

BA is a (one-sorted) *variety* in the sense that it is described by operations of finite arity and equations. Every variety $\mathcal{A}$ comes equipped with a forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$, which has a left-adjoint $F : \mathsf{Set} \longrightarrow \mathcal{A}$. Every algebra $A \in \mathcal{A}$ is a colimit $Fn_i \longrightarrow A$ of finitely generated free algebras in a canonical way [5]. In fact, $A \cong \mathsf{colim}(I \downarrow A \longrightarrow \mathcal{A}_0 \xrightarrow{I} \mathcal{A})$ where $I : \mathcal{A}_0 \hookrightarrow \mathcal{A}$ is the inclusion of the full subcategory $\mathcal{A}_0$ of finitely generated free algebras. Thus, to define a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$, it is enough to describe L on $\mathcal{A}_0$ and to extend to general $A \in \mathcal{A}$ via colimits, that is, $LA \cong \mathsf{colim}(I \downarrow A \longrightarrow \mathcal{A}_0 \xrightarrow{L} \mathcal{A})$. This colimit is preserved by U and thus calculated as in Set.

Definition 2.7 We say that a functor L on a variety $\mathcal{A}$ is determined by finitely generated free algebras if $LA \cong \mathsf{colim}(I \downarrow A \longrightarrow \mathcal{A}_0 \xrightarrow{L} \mathcal{A})$.

A functor is determined by finitely generated free algebras iff it preserves, so-called, sifted colimits [5]. It was proved in [32] that a functor preserves sifted colimits iff it can be described by operations and equations [11]. We will see examples of such presentations in Sections 3.2, 3.3.2 and 5.

2.6 Coalgebras

In this section we introduce coalgebras and bisimilarity.

Definition 2.8 The category $\mathsf{Coalg}(T)$ of *coalgebras* for a functor T on a category $\mathcal{X}$ has as objects arrows $\xi : X \longrightarrow TX$ in $\mathcal{X}$ and morphisms $f : (X, \xi) \longrightarrow (X', \xi')$ are arrows $f : X \longrightarrow X'$ such that $Tf \circ \xi = \xi' \circ f$.

Coalgebras are generalized transition systems. The states of the system are the elements of the set X, the type of transitions are described by the functor T and the transitions of the system are given by the function $\xi : X \longrightarrow TX$.

Example 2.9 (i) Coalgebras for $1 + Id$ are transition systems with termination. In a coalgebra $\xi : X \longrightarrow 1 + X$ we say that $x \in X$ terminates if $\xi(x) \in 1$; this is written $x \not\rightarrow$. If $\xi(x) = y$ then we write $x \longrightarrow y$ and say that there is a transition from x to y.

(ii) Coalgebras for $2 \times (-)^A$ are deterministic automata on the alphabet A. A coalgebra $\xi : X \longrightarrow 2 \times X^A$ is described by two functions $\xi_1 : X \longrightarrow 2$ and $\xi_2 : X \longrightarrow X^A$. The former function provides the accepting states of the automaton, the latter function describe the transition of the system, i.e., if $\xi_2(x)(a) = y$ we write $x \xrightarrow{a} y$ and read "there is a transition a from x to y".

(iii) Coalgebras for the covariant power set functor are Kripke frames, also known as non-deterministic (unlabelled) transitions systems [2]. For this, recall that a function $\xi : X$

$\longrightarrow \mathcal{P}(X)$ can be seen as a binary relation R_ξ, on X, defined as $xR_\xi y$ iff $y \in \xi(x)$. If $y \in \xi(x)$ we write $x \longrightarrow y$ and read "there is a transition from x to y".

(iv) Slight variations of the previous examples allow us to add labels to transitions of states. Coalgebras for $\mathcal{P}^A$ are labelled transition systems. Equally important are non-deterministic automata which can be seen as coalgebras for $2 \times \mathcal{P}^A$.

(v) Coalgebras for the finite distribution functor are discrete time Markov chains [8], also known as probabilistic transition systems. This can be seen as follows. Given a coalgebra $\xi : X \longrightarrow \mathcal{D}(X)$ and a state $x \in X$, we obtain a probability distribution $\xi_x = \xi(x) : X \longrightarrow [0,1]$. If $\xi_x(y) = p$, we write $x \xrightarrow{p} y$ and read "the probability of having a transition from x to y is p".

(vi) Coalgebras for the finite multiset functor are directed graphs with $\mathbb{N}$-weighted edges, often referred as multigraphs [44]. The idea follows the same spirit used in the example of distributions.

(vii) $\mathcal{QQ}$-coalgebras are known as neighborhood frames in modal logic and are investigated as coalgebras in [19]. A coalgebra $\xi : X \longrightarrow \mathcal{QQ}(X)$ can be interpreted as a two player game where a move in state x_1 consists of the first player choosing a set $S \in \xi(x_1)$ and the second player then the successor-state $x_2 \in S$.

The traditional notion of bisimilarity can be captured coalgebraically as follows.

Definition 2.10 Two states $x_i, (i = 1, 2)$, in two coalgebras (X_i, ξ_i) are T-*bisimilar*, or T-*behaviourally equivalent*, if there is a coalgebra (Z, ζ) and there are coalgebra morphisms $f_i : (X_i, \xi_i) \longrightarrow (Z, \zeta)$ such that $f_1(x_1) = f_2(x_2)$.

Going back to Example 2.9, one finds that this notion of bisimilarity coincides with the standard notions found in computer science. In detail: in Example 2.9 two states are bisimilar iff in (i), they do precisely the same number of steps before terminating; in (ii), they accept the same language [39]; in (iii-vii), they are bisimilar in the sense of process algebra and modal logic [2,40,14,19].

Remark 2.11 A bisimulation between two coalgebras (X_1, ξ_1) and (X_2, ξ_2) is a relation $B \subseteq X_1 \times X_2$ such that there is a coalgebra $B \longrightarrow TB$ making the two projections $B \longrightarrow X_i$ into coalgebra morphisms. In case the functor T preserves weak pullbacks, to say that there is a bisimulation relating x_1, x_2 is the same [40] as to say that x_1, x_2 are bisimilar according to Definition 2.10. In case T does not preserve weak-pullbacks, the notion of bisimulation is problematic but the notion of bisimilarity still works fine [30].

3 A Brief Survey of Coalgebraic Logic

In this section we will briefly introduce logics for coalgebras and describe how they can be treated parametric in the type-functor T. We will start with a general abstract framework based on Stone duality and then show how it relates to concrete logics. The main part of the section then discusses in detail the two most important examples of coalgebraic logics, namely Moss's logic and the logic given by all predicate liftings, before we turn in Section 4 for a detailed comparison of the two.

3.1 *Coalgebraic Logic: The abstract functorial framework*

For most of the paper, we are interested in Set-coalgebras and (finitary) logics which extend Boolean propositional logic; this is where our journey begins. More explicitly, we are in the following situation:

$$L \circlearrowright \mathsf{BA} \underset{S}{\overset{P}{\leftrightarrows}} \mathsf{Set} \circlearrowleft T \tag{1}$$

In the above picture, S and P are contravariant functors as described in Section 2.4. A coalgebraic logic has two components, syntax and semantics. Syntax is given by the functor L; semantics is a mean to relate L-algebras to T-coalgebras. The following definition formulates this more precisely.

Definition 3.1 A (Boolean) logic for T-coalgebras is a functor L determined by finitely generated free algebras (Definition 2.7) together with a natural transformation

$$\delta : LP \longrightarrow PT. \tag{2}$$

Using δ we can associate to a T-coalgebra $\xi : X \longrightarrow TX$ its dual L-algebra

$$\widehat{P}(\xi) = LPX \xrightarrow{\delta_X} PTX \xrightarrow{P(\xi)} PX. \tag{3}$$

The logic is given by the initial L-algebra $LI \longrightarrow I$, and the semantics by the unique arrow

$$[\![-]\!]_{(X,\xi)} : I \longrightarrow \widehat{P}(\xi) \tag{4}$$

A formula $\varphi \in I$ is then mapped to a set $[\![\varphi]\!]_{(X,\xi)} \subseteq X$. If $x \in [\![\varphi]\!]_{(X,\xi)}$, we say that x *satisfies* φ and write $x \Vdash_\xi \varphi$.

Remark 3.2 (i) The requirement that the functor L is determined by finitely generated free algebras ensures that free L-algebras exist and that L can be described by modal operators of finite arity. A proof of this can be found in [32], but we will see examples in the next subsections.

(ii) It is important to understand that L only describes how to add one layer of modalities: If A consists of Boolean formulas, then LA consists of modal formulas in which each formula $a \in A$ is under the scope of precisely one modal operator. The initial L-algebra is obtained by iterating this construction and contains modal formulas of arbitrary depth. Moreover, L can take into account not only the syntax, but also the axiomatisation of the logic, as revealed in Equation (7) below. To capture these by a functor, it is essential to consider L on BA and not simply on Set.

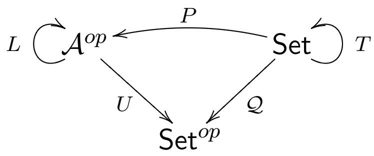
The functorial approach to modal logic makes it possible to directly relate the type constructors T on the semantic side with the 'logic constructors' L on the logical side. We will see examples of this in Section 3.3; there, the functor L will be defined directly from T.

Another advantage of the functorial approach is that (L, δ) gives us an abstract syntax-free description of the logic. This will be exploited, for example, in Sections 4.4 and 4.5 to define translations. Another illustration of this is given by the following definition where η and ε refer to the units of the adjunction in Diagram (1).

Definition 3.3 Let (L,δ) be a Boolean logic for T-coalgebras. It is *one-step complete*, if δ is injective. It is *one-step expressive*, if $TS \stackrel{\eta_{TS}}{\longrightarrow} SPTS \stackrel{S\delta_S}{\longrightarrow} SLPS \stackrel{SL\varepsilon}{\longrightarrow} SL$ is injective.

Remark 3.4 That one-step completeness implies completeness was shown in [26]. **Completeness** here means that if two elements φ_1, φ_2 of the initial L-algebra have the same extension in all coalgebras, then $\varphi_1 = \varphi_2$. That one-step expressiveness implies expressiveness for finitary T was shown in [42,25,21]. Here **expressiveness** means that any two non-bisimilar states of any two coalgebras are distinguished by some formula. If T preserves finite sets, then (L,δ) is one-step complete and expressive iff δ is an iso on finite sets.

Yet another advantage of the functorial approach is that Diagram (1) immediately suggests important generalisations. For example, in the following sections, to construct certain counterexamples, we will need to replace BA by other categories corresponding to other logics. For example, the category of distributive lattices which corresponds to the positive fragment of propositional logic; also the category of sets will be used to describe the modalities that need no extra structure to be translated. In all these examples, it is essential that powersets are algebras. To make this precise, we replace BA by any category $\mathcal{A}$ that comes with a 'forgetful' functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$ and a functor $P : \mathsf{Set} \longrightarrow \mathcal{A}^{op}$ such that $UP = \mathcal{Q}$ (in fact' $U^{op}P = \mathcal{Q}$ would be a more precise notation). As before a coalgebraic logic is then a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$ together with a natural transformation $\delta : LP \longrightarrow PT$. The situation is depicted in the following diagram



(5)

and formalised in

Definition 3.5 A category $\mathcal{A}$ is said to be a category *with powerset algebras* if (i) it is a concrete category over Set; (ii) the forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$ is monadic [35] with left adjoint $F : \mathsf{Set} \longrightarrow \mathcal{A}$; (iii) there exists a functor $P : \mathsf{Set} \longrightarrow \mathcal{A}^{op}$ such that $U^{op}P = \mathcal{Q}$.

We require U to be monadic in order to guarantee that, under mild conditions, initial algebras for L exist. In case $\mathcal{A} = \mathsf{BA}$ the following proposition specialises to Diagram (1). The category $\mathcal{A}$ is called the **base category of the coalgebraic logic**.

Proposition 3.6 *Under the assumptions of Definition 3.5, we have that $P : \mathsf{Set} \longrightarrow \mathcal{A}^{op}$ has a right adjoint $S : \mathcal{A}^{op} \longrightarrow \mathsf{Set}$ given by $A \mapsto \mathcal{A}(A, P1)$.*

Proof. The unit $\eta_X : X \longrightarrow SPX$ is given by $x \mapsto P(i_x)$ where we write $i_x : 1 \longrightarrow X$ for the map picking x. To show that P is left-adjoint, given $f : X \longrightarrow SA$, we need to find an appropriate morphism $A \longrightarrow PX$ in $\mathcal{A}$. Note that since U is monadic, U^{op} creates colimits; together with $U^{op}P = \mathcal{Q}$ and the fact that $\mathcal{Q}$ preserves colimits, this implies that P preserves colimits. Now, f is a family $f_x : A \longrightarrow P1$, that is, a map $A \longrightarrow \prod_{x\in X} P1 \cong P(\coprod_{x\in X} 1) \cong PX$. $\square$

3.2 Coalgebraic logic: First concrete examples

In this section we explain that the usual notion of a logic as given by connectives and axioms agrees with our notion of a logic (L, δ) given by a functor. We will show how these logics fit into the framework of the previous section.

The first step is to recognise that the basic propositional logic corresponds to a category of algebras. For example, classical propositional logic corresponds to BA and classical propositional logic without negation to DL. We can think of the algebras of the category as propositional, algebraic, theories and of morphisms as truth preserving translations between theories.

The second step consists of adding modal connectives to the basic propositional logic. For example, adding a unary $\Box$ to classical propositional logic, one might expect algebras $A \xrightarrow{\Box^A} A$. This is the standard approach in modal logic [10], but it has the drawback that although A is a BA there is no reason why $\Box^A$ should be a BA-morphism. Here is a little trick we can use: We define a functor $\mathsf{BA} \longrightarrow \mathsf{BA}$, call it $\bar{L}$, such that BA-morphisms $\bar{L}A \longrightarrow A$ are in 1-1 correspondence with maps $UA \longrightarrow UA$:

Example 3.7 Consider $\mathcal{A} = \mathsf{BA}$. Define $\bar{L}A$ to be the free Boolean algebra generated by $\Box a$, $a \in A$. Note that the $\Box a$'s are just formal symbols and we have

$$\bar{L} \cong FU. \tag{6}$$

Next, we observe that certain axioms of a special form can be incorporated into the definition of the functor. In particular, the axioms defining the basic modal logic $\mathbf{K}$ (see e.g. [10]) are of this form:

Example 3.8 Continuing Example 3.7, define $L : \mathsf{BA} \longrightarrow \mathsf{BA}$ to map an algebra A to the algebra LA generated by $\Box a$, $a \in A$, and quotiented by the relation stipulating that $\Box$ preserves finite meets, that is,

$$\Box\top = \top \qquad\qquad \Box(a \wedge b) = \Box a \wedge \Box b \tag{7}$$

It follows from the definition that BA-morphisms $LA \longrightarrow A$ are in 1-1 correspondence with meet-preserving maps $A \longrightarrow A$ and, therefore, that $\mathsf{Alg}(L)$ is isomorphic to the category of modal algebras [10].

The next step is that we can describe the semantics of such a logic without referring to Kripke frames, but directly in terms of the functor T. This is what allows us to generalise the relationship between algebras and their relational semantics to arbitrary functors.

Example 3.9 Continuing Example 3.8, consider $T = \mathcal{P}$. We define the semantics δ_X : $LPX \longrightarrow P\mathcal{P}X$ by, for $a \in PX$,

$$\Box a \mapsto \{b \in \mathcal{P}X \mid b \subseteq a\}. \tag{8}$$

Now we detail how to obtain from Equation (8), by using Equations (3) and (4), the usual semantics of $\Box$.

First define a language $\mathcal{L}$ by the grammar

$$\varphi ::= \top \mid \bot \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Box\varphi,$$

Then the initial $\bar{L}$-algebra is the smallest set closed under the operations $\wedge, \neg, \Box$ modulo the axioms for Boolean algebras; in other words, the initial $\bar{L}$ algebra is the term algebra over the language $\mathcal{L}$ modulo the usual axioms for Boolean algebras. The initial L-algebra is obtained from further quotienting by the modal axioms in Equation (7).

According to Equations (3) and (4), the interpretation of a formula is defined by initiality as in the following diagram

$$\begin{array}{ccccc} L(I) & & \xrightarrow{\quad\Box\quad} & & I \\ {\scriptstyle L([\![-]\!]_{(X,\xi)})}\downarrow & & & & \downarrow{\scriptstyle [\![-]\!]_{(X,\xi)}} \\ LP(X) & \xrightarrow[\delta_X]{} & PT(X) & \xrightarrow[\xi^{-1}]{} & P(X) \end{array} \tag{9}$$

which means that $[\![\Box\varphi]\!]_{(X,\xi)} = \xi^{-1}(\delta_X(L[\![\varphi]\!]_{(X,\xi)}))$. Now we can compute (eliding the subscript (X,ξ))

$$\begin{array}{llr} s \Vdash \Box\varphi & \text{iff } s \in [\![\Box\varphi]\!] & \text{(Definition } \Vdash) \\ & \text{iff } s \in \xi^{-1}(\delta_X(L[\![\varphi]\!])) & \text{(Diagram (9))} \\ & \text{iff } \xi(s) \in \delta_X(L[\![\varphi]\!]) & \text{(Definition } \xi^{-1}) \\ & \text{iff } \xi(s) \subseteq [\![\varphi]\!] & \text{(Equation (8))} \\ & \text{iff } (\forall x)(x \in \xi(s) \Rightarrow x \in [\![\varphi]\!]) & \text{(Definition } \Vdash) \\ & \text{iff } (\forall x)(x \in \xi(s) \Rightarrow x \Vdash \varphi) & \end{array}$$

which gives the usual semantics of $\Box$ in terms of a satisfaction relation $\Vdash$.

In the same vein we can also present logics over DL instead of BA as e.g. positive modal logic, which was introduced in [16], but also appeared in [24,1].

Example 3.10 Consider $T = \mathcal{P}$ as in Example 3.9 but now let $\mathcal{A} =$ DL. Then *positive modal logic* is given by the functor L : DL $\longrightarrow$ DL that maps a distributive lattice A to distributive lattice LA generated by $\Box a$ and $\Diamond a$ for all $a \in A$, and quotiented by the relations stipulating that $\Box$ preserves finite meets, $\Diamond$ preserves finite joins, and

$$\Box a \wedge \Diamond b \leq \Diamond(a \wedge b) \qquad \Box(a \vee b) \leq \Diamond a \vee \Box b \tag{10}$$

$\delta_X : LPX \longrightarrow P\mathcal{P}X$ is defined by, for $a \in PX$,

$$\Diamond a \mapsto \{b \in \mathcal{P}X \mid b \cap a \neq \emptyset\}, \tag{11}$$

the clause for $\Box a$ being the same as in Example 3.9. The above construction L is a variation of the Plotkin power domain or the Vietories locale, see e.g. [43].

Summary: From the concrete to the abstract Elaborating a bit further the previous example, the traditional definition of languages as "the smallest set closed under..." is a description of an initial algebra for some functor. More explicitly, it is the description of the functor providing the signature of the language. For example, if the signature would have operators $\{\Box_j | j \in J\}$, then $\bar{L} = \coprod_{j\in J}(-)^{ar(j)}$, where $ar(j)$ is the arity of operator $\Box_j$. The carrier set of the initial algebra, called I, would then be the smallest set closed under the Boolean operations and the operators $\Box_j$. The natural transformation δ would then describe the interpretation of each of the operators. Additional axioms for the modalities can be obtained as quotients of $\bar{L}$ which are equivalent to quotients of I. More examples will be discussed in Section 3.3 below.

It remains the question, which axioms can be incorporated into a functor. Consider adding a number of connectives, called modal operators, to a basic propositional logic such as BA. We say that an equation is of **rank-1**, if all variables are under the scope of precisely one modal operator. Without going into the technical details here, we note that for all equations of rank 1 we can quotient the functor by the axioms as in Equations (7) and (10). It is important to notice that this is not a restriction of coalgebraic logic as such: First use a rank-1 logic to describe properties of all T-coalgebras; then further non-rank-1 axioms can be used to single out T-coalgebras with particular properties.

From the abstract to the concrete Conversely, for any finitary functor on BA, or, more generally, any functor on a variety $\mathcal{A}$ determined by its action on finitely generated free algebras, we can find a presentation by modal operators and axioms of rank 1; see Remark 4.4 and [32] for more details.

3.3 Two generic Coalgebraic Languages

We have seen that coalgebras come equipped with a generic notion of bisimilarity (Definition 2.10). In the same spirit, the quest for the generic modal language to describe coalgebraic systems has played a key role in the development of coalgebraic logic. Two major currents have been successfully used for specifications and descriptions: Moss's logic and the logic of all predicate liftings. Both proposals can be elegantly presented within the framework of coalgebraic logics via Stone duality presented in Section 3.1.

3.3.1 Moss's Logic

Moss's logic was the first proposal of a coalgebraic logic parametric in the type-functor T. It requires T to preserve weak-pullbacks (Section 2.3); examples of such functors include all KPFs and composition of those with $\mathcal{B}_{\mathbb{N}}$ and $\mathcal{D}$, but not the functor $\mathcal{QQ}$. We follow the general approach of Diagrams (1) and (5).

Definition 3.11 Let $\mathcal{A}$ be a category with power set algebras, and let $T : \mathsf{Set} \longrightarrow \mathsf{Set}$ be a weak pullback preserving functor. *Moss's logic for T on $\mathcal{A}$* is given by the functor

$$FT_\omega U = M_T : \mathcal{A} \longrightarrow \mathcal{A}.$$

If there is no risk for confusion, we will simply write M.

Remark 3.12 (i) As discussed in Section 3.2, we can describe Moss's logic concretely as follows. For each $\alpha \in T_\omega UA$ we have a generator, written as $\nabla\alpha$. Then Moss's language $\mathcal{M}_T$ is the smallest set closed under Boolean operations and under the formation rule 'if $\alpha \in T_\omega(\mathcal{M}_T)$ then $\nabla\alpha \in \mathcal{M}_T$' (we will often drop the subscript

T). Quotienting $\mathcal{M}_T$ by Boolean axioms yields the carrier of the initial M_T-algebra (compare with Example 3.7).

(ii) In the original version [36], Moss showed that his coalgebraic logic characterizes bisimilarity of T-coalgebras. However, T may permit unbounded branching, e.g. $T = \mathcal{P}$, therefore a general result requires infinitary conjunctions in the logic (but does not need negation). Here our interests are different: We want to specify properties of coalgebras using only finitary, but all, Boolean connectives; accordingly, we will work with the finitary version T_ω of T. One consequence of this is that our modal formulas have then only finite depth.

To define the semantics $M_T P \longrightarrow PT$ as in Equation (8), it is enough to give a natural transformation $T_\omega UP \longrightarrow UPT$. Since $\mathcal{Q} = UP$, this can be written $T_\omega \mathcal{Q} \longrightarrow \mathcal{Q}T$, see Proposition 2.5(ii).

Definition 3.13 The semantics $M_T P \longrightarrow PT$ of Moss's logic is induced by $\boldsymbol{\nabla} : T_\omega \mathcal{Q} \longrightarrow \mathcal{Q}T$ mapping $\Phi \in T_\omega \mathcal{Q}X$ to

$$\boldsymbol{\nabla}(\Phi) = \{\alpha \in TX \mid \alpha\ \overline{T}(\in_X)\ \Phi\}, \tag{12}$$

where $\overline{T}(\in_X)$ is the relation lifting of $\in_X$ (Section 2.3).

Example 3.14 (i) In the case of the identity functor Id, we have that $\boldsymbol{\nabla} : Id\mathcal{Q} \longrightarrow \mathcal{Q}Id$ is the identity and Moss's logic is just that of deterministic transition systems ($\nabla\varphi \equiv \Box\varphi \equiv \Diamond\varphi$). Explicitly, a state x in a coalgebra ξ satisfies $\nabla\varphi$ iff $\xi(x) \in [\![\varphi]\!]$.

(ii) In the case of a constant functor K_C, we have that $\boldsymbol{\nabla} : K_C\mathcal{Q} \longrightarrow \mathcal{Q}K_C$ maps an element $d \in C$ to the set $\{d\}$. A state x in a coalgebra ξ satisfies ∇d iff $\xi(x) = d$.

(iii) Consider the functor $A \times (-)$ for some fixed set A. Given $\alpha \in A \times X$ and $\Phi \in A \times \mathcal{Q}(X)$ we have

$$\alpha \in \boldsymbol{\nabla}(\Phi) \text{ iff } \pi_1(\alpha) = \pi_1(\Phi) \text{ and } \pi_2(\alpha) \in \pi_2(\Phi).$$

For example, let $a, b \in A$ and consider the system $\circ \xrightarrow{a} \bullet$. In this system, state $\circ$ does not satisfy $\nabla(b, \top)$. In fact, $\circ$ can only satisfy modal formulas of the form $\nabla(a, \varphi)$, where φ is a formula valid on $\bullet$.

(iv) In the case of the covariant power set functor, we have that $\boldsymbol{\nabla}$ is given by

$$\alpha \in \boldsymbol{\nabla}(\Phi) \text{ iff } (\forall\varphi \in \Phi\,.\,\exists x \in \alpha\,.\,x \in \varphi) \text{ and } (\forall x \in \alpha\,.\,\exists\varphi \in \Phi\,.\,x \in \varphi).$$

It is well-known (and not difficult to check) that in this case Moss's logic (over BA or DL) is equivalent to classical modal logic, that is, there are *translations* in both directions:

$$\nabla\alpha = \Box \bigvee \alpha \wedge \bigwedge \Diamond\alpha$$
$$\Box\varphi = \nabla\{\varphi\} \vee \nabla\emptyset \quad \text{and} \quad \Diamond\varphi = \nabla\{\varphi, \top\}$$

Hence, Moss's logic for $\mathcal{P}$ is equivalent to standard modal logic.

(v) To describe $\boldsymbol{\nabla}$ in the case of the finite distribution functor recall, Example 2.4, that $b \in \mathcal{D}(X)$ and $B \in \mathcal{D}(\mathcal{Q}X)$ we can be presented as finite sequences $b = (x_i, p_i)_{1 \le i \le n}$ for some $x_i \in X, p_i \in [0,1], p_i > 0, n \in \mathbb{N}$; and $B = (\varphi_j, q_j)_{1\leq j \leq m}$ for $\varphi_j \in \mathcal{Q}X, q_j \in [0,1], q_j > 0, m \in \mathbb{N}$. The relation $b\,\overline{\mathcal{D}}(\in_X)\,B$ can be then described as follows: $b\,\overline{\mathcal{D}}(\in_X)\,B$ iff there are $(r_{ij})_{1\leq i\leq n, 1\leq j\leq m}, r_{ij} \in [0,1]$ such that $x_i \notin \varphi_j \;\Rightarrow\; r_{ij} = 0$ and $\sum_i r_{ij} = q_j$ and $\sum_j r_{ij} = p_i$.

For example, a state x in a coalgebra ξ satisfies $\nabla\{(\varphi, q), (\top, 1-q)\}$ iff the probability of going to a successor satisfying φ is larger or equal to q. That is, ∇ (together with Boolean operators) can express the modal operators of probability logic [20].

(vi) In the case of the finite multiset functor we have the same description, just replacing $[0,1]$ by $\mathbb{N}$. For example, a state x in a coalgebra ξ satisfies
- $\nabla\{(\top, n)\}$ iff x has exactly n successors;
- $\nabla\{(\varphi, m), (\top, n)\}$ iff x has at least m successors satisfying φ and exactly $m+n$ successors in total.

In fact, each ∇-formula specifies the total number of successors; this means that the usual graded modalities can therefore not be expressed.

3.3.2 *The Logic of All Predicate Liftings*

Whereas Moss's logic has an unusual syntax and semantics, the logics presented in this section are a direct generalisation of the modal logics of Examples 3.7 and 3.10. The main point of this section is to illustrate that for any $T : \mathsf{Set} \longrightarrow \mathsf{Set}$ there is a canonical way of extracting the modal operators and their semantics from T. We will assume that the basic propositional logic corresponds to a variety (Section 2.5) with powerset algebras (Definition 3.5). We will make us of the fact that every algebra is the colimit of finitely generated free ones.

Definition 3.15 The functor $L_T : \mathcal{A} \longrightarrow \mathcal{A}$ is defined on finitely generated free algebras Fn as $L_T Fn = PT\mathcal{Q}n$ and extended to arbitrary $A \in \mathcal{A}$ via colimits.

Given the construction of L_T, to define the semantics $\delta : L_T P \longrightarrow PT$ it is enough to first describe it on finitely generated free algebras.

Definition 3.16 The semantics $\delta_T : L_T P \longrightarrow PT$ is given by considering PX as a colimit $c_i : Fn_i \longrightarrow PX$, which is, by construction, preserved by L_T. More explicitly, $(\delta_T)_X$ is the unique arrow making the following diagram

$$\begin{array}{ccc} L_T PX & \overset{(\delta_T)_X}{\dashrightarrow} & PTX \\ {\scriptstyle L_T c_i}\uparrow & & \uparrow{\scriptstyle PT\hat{c}_i} \\ L_T Fn_i & \underset{id}{\longrightarrow} & PT\mathcal{Q}n_i \end{array} \qquad (13)$$

commute for each i; in the previous diagram, $\hat{c}_i$ comes from applying the sequence of isomorphisms $\mathcal{A}(Fn_i, PX) \cong \mathsf{Set}(n_i, UPX) \cong \mathsf{Set}(n_i, \mathcal{Q}X) \cong \mathsf{Set}(X, \mathcal{Q}n_i)$ to c_i.

In case $\mathcal{A} = \mathsf{BA}$ and $T = \mathcal{P}$ the previous definition describes the functor in Example 3.7. This follows from the fact that both the LPX in Example 3.7 and the $L_{\mathcal{P}}PX$ above are isomorphic, via the corresponding δ, to $P\mathcal{P}X$ on finite X. However, $L_{\mathcal{P}}$ hides more modal operators and for a general T we need all of them to describe L_T concretely. This concrete description of L_T is based on the observation that the carrier set of $L_T F(n)$, i.e. $UL_TF(n)$, can be described as follows:

$$UL_TFn = UPT\mathcal{Q}n = \mathcal{Q}T\mathcal{Q}(n); \tag{14}$$

the first equality spells out the definition of L_T on finitely generated free algebras, the second equality uses $\mathcal{Q} = UP$. An element of $\mathcal{Q}T\mathcal{Q}n$, i.e. a map $T(2^n) \longrightarrow 2$, is called an n-ary **predicate lifting** of T. Those are the modal operators that we wanted to unravel. Since $\mathcal{Q}(X) = \mathsf{Set}(X, 2)$ the reader will recognise that these predicate liftings of arity n are precisely the natural transformation $\mathcal{Q}^n \longrightarrow \mathcal{Q}T$. This is an incarnation of the Yoneda lemma; here is the formal statement and proof.

Proposition 3.17 *There is a natural isomorphism (natural in n and $\mathcal{Q}T$)*

$$Y_{(n,T)} : \mathcal{Q}T\mathcal{Q}(n) \longrightarrow \mathsf{Nat}(\mathcal{Q}^n, \mathcal{Q}T). \tag{15}$$

Proof. Recall that $\mathcal{Q}^n X = \mathsf{Hom}(X, \mathcal{Q}n)$. We define a bijection between $\mathcal{Q}T(\mathcal{Q}n)$ and natural transformations $\mathcal{Q}^n X \longrightarrow \mathcal{Q}TX$ as follows: any $p \in \mathcal{Q}T(\mathcal{Q}n)$ gives a natural transformation $Y(p) : \mathcal{Q}^n X \longrightarrow \mathcal{Q}TX$ that maps $v : X \longrightarrow \mathcal{Q}n$ to $\mathcal{Q}Tv(p)$. Conversely, for each $\lambda_X : \mathcal{Q}^n X \longrightarrow \mathcal{Q}TX$ we have $\lambda_{\mathcal{Q}n}(\mathrm{id}_{\mathcal{Q}(n)}) \in \mathcal{Q}T(\mathcal{Q}n)$. □

This also shows that the predicate liftings introduced here are indeed precisely those introduced in [37,42].

Remark 3.18 Using Proposition 3.6, we see that $L_TFn = PT\mathcal{Q}n$ in Definition 3.15 could also be written as $L_TFn = PTSFn$, showing that Definition 3.15 is in agreement with [32,31].

We now proceed to give a concrete description of $\delta_T : L_TP \longrightarrow PT$.

Express PX as the canonical colimit $c_\varphi : Fn_\varphi \longrightarrow PX$ where φ ranges over maps $\{n \longrightarrow \mathcal{Q}X \mid n < \omega\}$ and n_φ denotes the domain of φ and $c_\varphi : Fn_\varphi \longrightarrow PX$ is the transpose of $\varphi : n_\varphi \longrightarrow UPX$. Since U preserves the colimit, we can calculate UL_TPX as a colimit in Set, that is, UL_TPX is a quotient of the φ-indexed disjoint union of $UL_TFn_\varphi = \mathcal{Q}T\mathcal{Q}n_\varphi$. In other words, every element in UL_TPX is of the form $UL_Tc_\varphi(\lambda)$ for some $\lambda \in \mathcal{Q}T\mathcal{Q}n_\varphi$. Let us write $\lambda(\varphi)$ for $UL_Tc_\varphi(\lambda)$. Then Diagram (13) gives us:

$$\begin{aligned} (\delta_T)_X : L_TPX &\longrightarrow PTX \\ \lambda(\varphi) &\mapsto TX \xrightarrow{T(\chi_\varphi)} T\mathcal{Q}n \xrightarrow{\lambda} 2 \end{aligned} \tag{16}$$

where $\chi_\varphi : X \longrightarrow \mathcal{Q}n$ is the transpose of $\varphi : n \longrightarrow \mathcal{Q}X$.

Equation (16) and Proposition 3.17 explain the term ‘predicate lifting’: λ lifts a list φ of predicates on X to a predicate $\lambda \circ T\chi_\varphi$ on TX. We summarise all this in the next definition.

Definition 3.19 Given a functor $T : \mathsf{Set} \longrightarrow \mathsf{Set}$, an n-ary predicate lifting is a natural transformation $\mathcal{Q}^n X \longrightarrow \mathcal{Q}TX$ or, equivalently, it is an element of $\mathcal{Q}T\mathcal{Q}(n)$.

Proposition 3.17 depicts the procedure to convert natural transformations $\mathcal{Q}^n \longrightarrow \mathcal{Q}T$ into subsets of $T(2^n)$ and viceversa. More explicitly, this is done as follows: Given a set

$P \subseteq T(2^n)$ we define a predicate lifting $\lambda_P : \mathcal{Q}^n \longrightarrow \mathcal{Q}T$ which maps a sequence $\varphi : n \longrightarrow \mathcal{Q}X$ to the set

$$(\lambda_P)_X(\varphi) = \{t \in TX \mid T(\chi_\varphi)(t) \in P\} \tag{17}$$

where $\chi_\varphi : X \longrightarrow 2^n$ is the transpose of φ. We now present some concrete examples of predicate liftings.

Example 3.20 (i) Let $T = K_C$ be a constant functor with value C. Any subset P of C defines a predicate lifting $\lambda_P : \mathcal{Q} \longrightarrow \mathcal{Q}K_C$; it is has constant value P.

(ii) The previous example can be modified to provide propositional information. For this we consider the functor $\mathcal{P}(Q) \times T$, where Q is a fixed set of proposition letters. The semantics of the proposition letter $q \in Q$ is given by the predicate liftings $\lambda^q_X(\varphi) = \{(U, \alpha) \in \mathcal{P}(Q) \times T(X) \mid q \in U\}$, and $\lambda^{\neg q}_X(\varphi) = \{(U, \alpha) \in \mathcal{P}(Q) \times T(X) \mid q \notin U\}$. These predicate liftings are associated with the sets $U_q \times T(X)$ and $U_{\neg p} \times TX$ respectively; we write U_q for the set of subsets of Q containing q and $U_{\neg q}$ for its complement.

(iii) Let T be the covariant power set functor and let $2 = \{\bot, \top\}$. The existential modality $\Diamond$ can be presented using an homonymous predicate lifting $\Diamond : \mathcal{Q} \longrightarrow \mathcal{QP}$, with the followings components a set $\varphi \subseteq X$ is mapped to $\Diamond_X(\varphi) = \{\psi \subseteq X \mid \varphi \cap \psi \neq \emptyset\}$. Using (15), we can see that this corresponds to the set $\{\{\top\}, \{\top, \bot\}\}$. Similarly, the universal modality $\Box$ can be presented as a predicate lifting which transforms a set $\varphi \subseteq X$ into $\Box_X(\varphi) = \{\psi \subseteq X \mid \psi \subseteq \varphi\}$ (compare this with Equation (8) and the examples neighbouring it). Using Equation (15), this predicate lifting is associated to the set $\{\emptyset, \{\top\}\}$.

(iv) Consider neighborhood functor, i.e. $\mathcal{QQ}$. The standard modalities, used in game logic and coalition logic, can be seen as predicate liftings. For example, the universal modality transforms a set $\varphi \subseteq X$ in to the set $\Box_X(\varphi) = \{N \in \mathcal{QQ}(X) \mid \varphi \in N\}$. This modality is associated with the (ultra)-filter generated by $\{\top\}$.

(v) Consider the multiset functor $\mathcal{B}_\mathbb{N}$ and let k be a natural number. A graded modality can be seen as a predicate lifting for this functor; a set $\varphi \subseteq X$ is mapped $\lambda^k_X(\varphi) = \{B : X \longrightarrow \mathbb{N} \mid \sum_{x \in \varphi} B(x) \geq k\}$. In this case $x \Vdash_\xi \lambda^k \varphi$ holds iff x has at least k many successors satisfying φ. Using Equation (15) we obtain that λ^k corresponds to the set $\{B : 2 \longrightarrow \mathbb{N} \mid B(\top) \geq k\}$; in other words it is associated with the set $[k, \infty)$. In general, a predicate lifting for $\mathcal{B}_\mathbb{N}$ can be described by two subsets of $\mathbb{N}$; one describing the target of $\top$ and other describing the target of $\bot$.

(vi) Let T be the finite distribution functor. The modality $\Diamond_p \varphi$ specifies a probability of at least p for the event of going to a successor satisfying φ. It can be described by the predicate lifting $\mathcal{Q}X \longrightarrow \mathcal{Q}TX$, $\varphi \mapsto \{d \in \mathcal{D}X \mid \mu_d(\varphi) \geq p\}$, where $\mu_d(\varphi) = \sum_{x \in \varphi} d(x)$ is the measure associated with d. By Equation (15), this predicate lifting corresponds to a subset of $\mathcal{D}(2)$. Since we can describe a probability distribution $d : 2 \longrightarrow [0, 1]$ by its value on $\top$ ($d(\bot) = 1 - d(\top)$), we can see that unary predicate liftings correspond to subsets of $[0, 1]$; more precisely, $P \subseteq [0, 1]$ corresponds to the set of distributions $d : 2 \longrightarrow [0, 1]$ such that $d(\top) \in P$. In particular, $\Diamond_p$ corresponds to the set $[p, 1]$. Similarly, the predicate lifting $\Box_p = \neg\Diamond_p\neg$ correspond to the set $(1 - p, 1]$; more explicitly, $\Box_p$ maps a set φ to the set $\{d \in \mathcal{D}X \mid \mu_d(\varphi) > 1 - p\}$. Another common modality in probability logic is given by $\Diamond^p = \Diamond_{1-p}\neg$; this modality corresponds to the predicate lifting associated with $[0, p]$. These modalities give the usual language from [20]. In general, the predicate lifting associated to an interval $(q, q') \subseteq [0, 1]$ maps a set $\varphi \subseteq X$ to the set of probability distributions over X that assign a probability between q and q' to the set φ. More explicitly, a state in a coalgebra ξ satisfies $\lambda_{(q,q')}\varphi$ iff the probability of executing a transition to a state satisfying φ is between q and q'.

Remark 3.21 It is often the case that not all predicate liftings are needed to generate L_T. For example, over BA we have that $L_{\mathcal{P}}$ is generated by a single $\Box$ as in Example 3.7 and over DL we have that $L_{\mathcal{P}}$ is generated by $\Box$ and $\Diamond$ as in Example 3.10.

In the light of the above remark, we introduce notation used in the following for logics generated by some set Λ of predicate liftings.

Definition 3.22 (i) Let $\Lambda = (\Lambda_n)_{n<\omega}$ be a family of sets of predicate liftings $\Lambda_n \subseteq \mathcal{Q}T\mathcal{Q}n$. The functor $\bar{L}_\Lambda : \mathcal{A} \longrightarrow \mathcal{A}$ is defined as $F(\coprod_{n<\omega} \coprod_{\lambda\in\Lambda_n} U^n_\lambda)$. The semantics $\bar{L}_\Lambda PX \longrightarrow PTX$ is given, for each $\lambda \in \Lambda$, via Equation (16).

(ii) The language $\mathcal{L}_\Lambda$ is the smallest language closed under propositional connectives and under the rule: $n < \omega, 1 \leq i \leq n, \varphi_i \in \mathcal{L}_T, \lambda \in \Lambda_n \quad \Rightarrow \quad \lambda(\varphi_1, \dots \varphi_n) \in \mathcal{L}_T$. As before, quotienting $\mathcal{L}_\Lambda$ by the equations defining the variety $\mathcal{A}$ yields the (carrier of the) initial $\bar{L}_\Lambda$-algebra.

(iii) The language $\mathcal{L}_T$, or just $\mathcal{L}$, is $\mathcal{L}_\Lambda$ where Λ consists of all predicate liftings.

The functor $\bar{L}_\Lambda$ in the previous definition can also be seen as follows: For each $\lambda \in \Lambda$ consider the functor $L_\lambda = F(U^{ar(\lambda)})$; the semantics is given via Equation (16). Compare this with Example 3.7. Take the coproduct of all those. Since F is a left adjoint it preserves coproducts, i.e. it can be moved outside; this gives the functor $\bar{L}_\Lambda$ described in the previous definition.

4 Translating coalgebraic logics

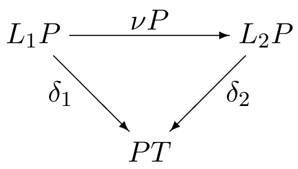
In this section we will investigate under what circumstances we can find a translation from the ∇-logic $\mathcal{M}$ into the logic of all predicate liftings $\mathcal{L}$ and vice versa. The main result states that both logics are equivalent, that is, can be translated into each other, in case the functor T preserves weak pullbacks and finite sets; and the basic propositional logic is Boolean. Recall that the first condition is needed because otherwise Moss's logic is not defined. Examples 4.17(i) and 4.25 explain why the other conditions conditions are needed.

Let us emphasise that we are not interested in showing only that every formula in $\mathcal{L}$ has an equivalent formula in $\mathcal{M}$ (and v.v.). Rather we want an inductive definition of the translation, which respects the one-step nature of the logics (see Remarks 3.2 and 4.2). This stronger property of translations is captured by natural transformations $\bar{L} \longrightarrow M$ and $M \longrightarrow L$.

4.1 One-step translations

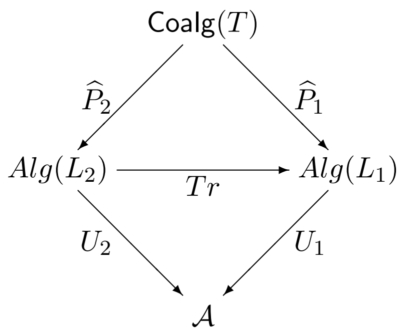
We start by defining translations between coalgebraic logics. Our notion of coalgebraic logic assumes a category $\mathcal{A}$ of power-set algebras, a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$ and a natural transformation $\delta : LP \longrightarrow PT$, as explained in Section 3.1.

Definition 4.1 Given two coalgebraic logics (L_1, δ_1) and (L_2, δ_2), a natural transformation $\nu : L_1 \longrightarrow L_2$ is a *one-step translation* if it commutes with the semantics:



A one-step translation can be understood as an inductive definition of a translation between the associated logics. Indeed, given any L_2-algebra, $L_2 A \longrightarrow A$, we obtain an L_1-algebra $L_1 A \xrightarrow{\nu_A} L_2 A \longrightarrow A$; moreover, since ν is a natural transformation any morphism $f : A \longrightarrow A'$ of L_2 algebras is also a morphism between the corresponding L_1-algebras. Denote by $L_i I_i \longrightarrow I_i$ the initial L_i-algebras. Using this observation, we find, by initially of I_1, an inductively defined morphism of L_1-algebras $I_1 \longrightarrow I_2$ which translates formulas in I_1 to formulas in I_2. Notice that it is important that ν is natural because this allows us to map a morphism of L_2-algebras $I_2 \longrightarrow A$ to a morphism of L_1-algebras.

Consequently, a one-step translation from L_1 to L_2 induces a functor (translation functor) $Tr : Alg(L_2) \longrightarrow Alg(L_1)$ such that the following diagram



commutes, where $\widehat{P}_i$ is the functor described in equation (3), page 8. The commutativity of the lower triangle is used to define the translation, i.e. the function $I_1 \longrightarrow I_2$. The upper triangle is used to show that this translation preserves the interpretation of formulas.

Remark 4.2 We do not want to define translations as morphism between the free monads generated by L_1 and L_2. Such a more general notion would allow us, for example, to express an L_1-formula $\Box_1 \varphi_1$ as a combination of L_2-formulas with nested modal operators such as e.g. $\Box_2 \Diamond_2 \varphi_2$. But it would not solve the problem of Examples 4.17(i) and 4.25 where translations fail to exist.

The next example illustrates one-step translations using the well known equivalences for the power set functor.

Example 4.3 Let (M, ∇) be Moss's logic for $\mathcal{P}$ (Example 3.14) and let $(L_{(\Box,\Diamond)}, \Box + \Diamond)$ be the basic modal logic for $\mathcal{P}$ (Example 3.7, Definition 3.22). We write $U : \mathsf{BA} \longrightarrow \mathsf{Set}$ for the forgetful functor and F for its left adjoint; let α be a boolean algebra with carrier set A.

(i) In the particular case of $\Diamond$ we can define a one-step translation $\nu_\Diamond : FU \longrightarrow F\mathcal{P}_\omega U$ by just presenting a natural transformation $\tau_\Diamond : U \longrightarrow \mathcal{P}_\omega U$ and then extending it freely, i.e applying F to it. We define $\tau_\Diamond$ as follows: an element $a \in U(\alpha)$ is mapped to $\tau_\Diamond(a) = \{a, \top\}$. The free extension procedure is particular to $\Diamond$ see Section 4.2 for more on this.

(ii) The usual translation of $\Box$ is given by a natural transformation $\nu_\Box : FU \longrightarrow F\mathcal{P}_\omega U$. In this case, using the properties of free algebras, we can defined the translation by presenting a natural transformation $\tau : U \longrightarrow UF\mathcal{P}_\omega U$. An element $a \in U(\alpha)$ is mapped to $\tau(a) = \nabla\{a\} \vee \nabla\bot$.

(iii) To translate ∇ we ought to define a natural transformation $\nu_\nabla : F\mathcal{P}_\omega U \longrightarrow F(U_\Box + U_\Diamond)$, here we write $U_\Box + U_\Diamond$ to indicate that one factor deals with $\Box$ and the other with $\Diamond$. One more time, using properties of free algebras it is enough to define a natural transformation $\tau : \mathcal{P}_\omega U \longrightarrow UF(U_\Box + U_\Diamond)$. Let φ be an element in $\mathcal{P}_\omega U(\alpha)$. Since φ is finite there are elements in $U_\Box(\alpha)$ and $FU_\Diamond(\alpha)$ corresponding to $\bigvee \varphi$ and $\bigwedge_{a\in\varphi} \Diamond a$ respectively. We now define τ as expected, i.e. $\tau(\varphi) = \Box \bigvee \varphi \wedge \bigwedge_{a\in\varphi} \Diamond a$.

Remark 4.4 [From abstract to concrete] Another illustration of one-step translations is a concrete presentation of a coalgebraic logic.

The first observation is that any coalgebraic logic (L, δ) can be translated into the language of all predicate liftings. To see this, first recall that $L_T F(n) = PT\mathcal{Q}(n)$ (Definition 3.15). In any category of power set algebras we have $SF(n) = \mathcal{Q}(n)$ (Proposition 3.6); hence $L_T F(n) = PTSF(n)$. Now notice that $\delta : LP \longrightarrow PT$ has an adjoint transpose $\delta^\sharp : L \longrightarrow PTS$. From this, we obtain

$$LF(n) \xrightarrow{\delta^\sharp_{F(n)}} PTSF(n) = L_T F(n)$$

Since both L and L_T are are determined by their action on finitely generated free algebras, the natural transformation above can be extended into a natural transformation $L \longrightarrow L_T$ which is in fact a one step translation.

The second observation is that we can do slightly better and present the predicate liftings needed explicitly. Notice that each $p \in ULF(n)$ induces, by Yoneda, a natural transformation $E(p, -) : U^n \longrightarrow UL$ from which we can obtain a predicate lifting as the following composite

$$(UP)^n \xrightarrow{E_P(p,-)} ULP \xrightarrow{U(\delta)} UPT.$$

These are the concrete modalities that can generally use to present (L, δ). In this paper we only develop the case $(M, \boldsymbol{\nabla})$, see Section 5.2. More details on such presentations can be found in [11,32].

4.2 *Translating predicate liftings*

We are looking for a natural transformation $\bar{L}_\Lambda \longrightarrow M$ (see Definitions 3.22 and 3.11). As explained after Definition 3.22, this can be done considering one predicate lifting at a time. In order to tailor the desired translations, we will first introduce the concept of translators for predicate liftings (Definition 4.5). Unfortunately, not all predicate liftings have translators (Example 4.6). However, all singleton liftings (Definition 4.9) have translators and in fact every predicate lifting is a union of singleton liftings (Proposition 4.12).

Definition 4.5 A translator for an n-ary predicate lifting λ is a natural transformation $\tau : \mathcal{Q}^n \longrightarrow T\mathcal{Q}$ such that

$$\begin{array}{ccc} \mathcal{Q}^n & \xrightarrow{\;\;\tau\;\;} & T_\omega \mathcal{Q} \\ & {\scriptstyle\lambda}\searrow \qquad \swarrow{\scriptstyle\boldsymbol{\nabla}} & \\ & \mathcal{Q}T & \end{array} \tag{18}$$

We illustrate the concept with some examples.

Example 4.6 The following are examples of translators.

(i) Consider the predicate lifting associated with the existential modality $\Diamond$ of the covariant power set functor (Example 3.20). The following natural transformation is a translator for $\Diamond$; we define $\tau_X : \mathcal{Q}X \longrightarrow \mathcal{P}_\omega \mathcal{Q}X$ mapping an element $\varphi \subseteq X$ to $\tau_X(\varphi) = \{\varphi, X\}$. Compare this with the equivalence $\Diamond\varphi = \nabla\{\varphi, \top\}$ discussed in Example 3.14. More illustrations can be seen in Example 4.10 below.

(ii) Consider the usual probability modality $\Diamond_p$, i.e. "the probability of ... is at least p". This predicate lifting has a translator $\tau_p : \mathcal{Q} \longrightarrow \mathcal{D}\mathcal{Q}$ defined as follows: A set $\varphi \subseteq X$ is mapped to to the probability distribution $D_p^\varphi : \mathcal{Q}(X) \longrightarrow [0,1]$ which assigns p to the set φ and $1-p$ to the set X. Compare this with the description in Example 3.14.

(iii) We can use the same idea of the previous item to translate the probability modality $\Diamond^p$, i.e. "the probability of ... is at most p" (Example 3.20). The natural transformation $\tau^p : \mathcal{Q} \longrightarrow \mathcal{D}\mathcal{Q}$ which maps a set φ to the probability distribution, $D_\varphi^p : \mathcal{Q}(X) \longrightarrow [0,1]$, assigning $1-p$ to the set $\neg\varphi$ and p to the set X, is a translator for $\Diamond^p$.

Remark 4.7 Using relation lifting we can describe translators as follows: a natural transformation $\tau : \mathcal{Q}^n \longrightarrow T_\omega \mathcal{Q}$ is a translator for λ iff for every $\varphi : n \longrightarrow \mathcal{Q}(X)$ and every $t \in T(X)$ the following holds

$$(t, \tau(\varphi)) \in \overline{T}(\in_X) \text{ iff } t \in \lambda(\varphi).$$

The idea of a translator is to define a one-step translation tr via

$$tr(\lambda\varphi) = \nabla\tau(tr(\varphi)). \tag{19}$$

Unfortunately not all predicate liftings have translators. This means that not all predicate liftings can be translated using only ∇ without propositional connectives. The following example illustrates this.

Example 4.8 The following predicate liftings fail to have translators.

(i) Let K_C be a constant functor where C has at least two distinct elements c_1, c_2. Using Proposition 3.17 (see also Example 3.20), predicate liftings correspond to subsets of C. The predicate lifting λ_E corresponding to $E = \{c_1, c_2\}$ does not have a translator. This is because the components of a natural transformation $\tau : \mathcal{Q} \longrightarrow K_C$ ought to be constant functions, hence the cardinality of $\boldsymbol{\nabla}\tau(X)$ is always 1, but $\lambda_E X = E$. Nevertheless, notice that the formula $\nabla c_1 \vee \nabla c_2$ translates the predicate lifting λ_E.

(ii) Consider de graded modality λ^k, at least k successors, for the finite multiset functor. Recall from Example 3.14 that each ∇ formula for $\mathcal{B}_\mathbb{N}$ specifies the total number of successors. Since λ^k does not declare an specific number of successors, we conclude that λ^k can not have a translator.

(iii) Let $\Diamond_{>p}$ be a modality for the finite distribution functor corresponding to the set $(p, 1]$, i.e. $\Diamond_{>p}(\varphi) = \{d \in \mathcal{D}(X) \mid \mu_d(\varphi) > p\}$, where $\mu_d(\varphi) = \sum_{x\in\varphi} d(x)$. Each of these modalities fail to have a translator. The reason for this is that each natural transformation $\tau : \mathcal{Q} \longrightarrow \mathcal{DQ}$ specifies a probability for each set φ, as an element of $\mathcal{Q}(X)$, say q. Consequently, Example 3.14, $\nabla\tau(\varphi)$ can only contain probability distributions d such that $\sum_{x\in\varphi} d(x) = q$. Hence no single natural transformation can factor $\Diamond_{>p}$ via $\boldsymbol{\nabla}$. In particular the modality $\Box_p$, the dual to $\Diamond_p$ in Example 3.20, does not have a translator because it corresponds to the set $(1 - p, 1]$. Nevertheless, $\Box_p$ can be translated into Moss language using negations because $\Diamond_p$ is translatable, see previous example.

First, we need to know a big enough class of predicate liftings that do have translators.

Definition 4.9 ([34]) An n-ary predicate lifting λ is called a *singleton predicate lifting*, or a *singleton lifting* for short, if it is associated (via Proposition 3.17) with a single element $p \in T(2^n)$, i.e. if the following holds: Given $\varphi : n \longrightarrow 2^X$

$$\lambda_X(\varphi) = \{t \in TX \mid T(\chi_\varphi)(t) = p\}, \tag{20}$$

where $\chi_\varphi : X \longrightarrow 2^n$ is the transpose of φ. If λ is a singleton lifting, we write it λ_p or just p, where p is the associated element of $T(2^n)$.

Example 4.10 (i) If T is a constant functor with value C, then the singleton liftings for T are associated with elements $c \in C$. The X-component of a singleton lifting λ_c is the function $\lambda_c : \mathcal{Q}X \longrightarrow \mathcal{Q}K_C$ with constant value $\{c\}$.

(ii) If T is the identity functor and we assume $2 = \{\top, \bot\}$, then there are two singleton liftings of arity 1 for Id. The X-component of $\lambda_\top$ is the identity. Similarly, the X-component of $\lambda_\bot$ is the function $(\lambda_\bot)_X : \mathcal{Q}X \longrightarrow \mathcal{Q}X$ mapping a set $\varphi \subseteq X$ to $\lambda_\bot(\varphi) = \neg_X\varphi$ to its complement.

(iii) Let $T = 1 + Id$. Consider the set $\{*\} \subseteq 1 + 2$, where $* \in 1$. The associated singleton lifting $\lambda_* : \mathcal{Q} \longrightarrow \mathcal{Q}(1 + Id)$ maps a set $\varphi \subseteq X$ to $\{*\}$. This modality indicates termination, i.e. $x \Vdash_\xi \lambda_*\varphi$ iff a transition from x leads the system to halt. The other singleton liftings for T are similar to those of Id.

(iv) The covariant power set functor has four singleton liftings of arity 1, explicitly these are associated with $\mathcal{P}(2) = \{\emptyset, \{\top\}, \{\bot\}, \{\top, \bot\}\}$. Given a set $\varphi \subseteq X$, the action of these predicate liftings is (we drop the subscripts X):

$$\lambda_{\{\top\}}(\varphi) = \{U \in \mathcal{P}X \mid \emptyset \neq U \subseteq \varphi\}; \quad \lambda_{\{\bot\}}(\varphi) = \{U \in \mathcal{P}X \mid \emptyset \neq U \subseteq \neg_X\varphi\};$$
$$\lambda_\emptyset(\varphi) = \{\emptyset\}; \quad \lambda_{\{\top,\bot\}}(\varphi) = \{U \in \mathcal{P}X \mid U \cap \neg_X\varphi \neq \emptyset \neq U \cap \varphi\};$$

Note that they all have translators, corresponding to $\nabla\{\varphi\}$, $\nabla\{\neg_X\varphi\}$, $\nabla\emptyset$, $\nabla\{\varphi, \neg_X\varphi\}$, respectively.

(v) If T is the finite multiset functor, a singleton lifting is given by a pair of natural numbers (n, m). Its X component, $(n, m) : \mathcal{Q}X \longrightarrow \mathcal{Q}\mathcal{B}_\mathbb{N}X$, maps a set $\varphi \subseteq X$ to the set of bags over X with $n + m$ elements, n of which are in φ and m are in the complement of φ. Such a predicate lifting has a translator as it corresponds to $\nabla\{(\varphi, n), (\neg_X\varphi, m)\}$, in the notation of Example 3.14.

(vi) If T is the finite distribution functor, a singleton lifting probability distribution $d : 2 \longrightarrow [0,1]$. Since we require $d(\top) + d(\bot) = 1$, a singleton lifting for the finite distribution is then determined by is given by a real number $q \in [0,1]$. Recall Example 3.20, the X-component of λ_q. maps a set $\varphi \subseteq X$ to the set of probability distributions over X that assign probability q to the set φ. Such predicate liftings have translators as they correspond to $\nabla\{(\varphi, q), (\neg_X\varphi, 1-q)\}$, in the notation of Example 3.14; compare this formula with the one in the mentioned example.

We now fix some notation for the language of singleton liftings.

Notation 4.11 *The set of, finitary, singleton liftings is denoted by Λ_s; we write $\bar{L}_s$ for the corresponding functor (Definition 3.22).*

The second author's [34] started the study of singleton liftings because: (i) In the case of KPFs they can be presented inductively over the complexity of the functor, and (ii) they generate all the other predicate liftings. This is more formally stated in the next proposition.

Proposition 4.12 ([34]) *If λ is an n-ary predicate lifting associated with a set $P \subseteq T(2^n)$, then for every set X and every n-sequence $\varphi : n \longrightarrow \mathcal{Q}X$ we have: $\lambda_X(\varphi) = \bigcup_{p \in P} (\lambda_p)_X(\varphi)$. In other words, every n-ary predicate lifting can be obtained as a (possibly infinite) join of singleton predicate liftings.*

Proof. The proof is an application of Proposition 3.17. Recall Equation 17, on page 16, and Definition 4.9. Using those we can show that the action of λ, over an n-sequence $\varphi : n \longrightarrow \mathcal{Q}X$, can be described as follows

$$\begin{aligned}(\lambda_P)_X(\varphi) &= \{t \in TX \mid T(\chi_\varphi)(t) \in P\} \\ &= \bigcup_{p \in P} \{t \in TS \mid T(\chi_\varphi)(t) = p\} = \bigcup_{p \in P} (\lambda_p)_X(\varphi).\end{aligned}$$

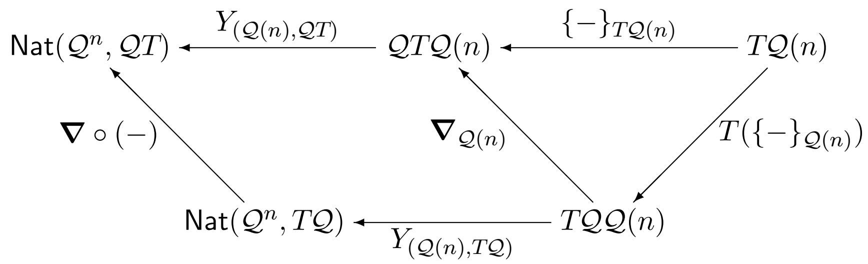
□

Example 4.13 Going back to Example 3.20, the predicate lifting for $\Box$ is $\lambda_{\{\emptyset,\{\top\}\}}$. It does not have a translator but is the union $\lambda_\emptyset \cup \lambda_{\{\top\}}$ of two singleton liftings, which have a translator by Example 4.10. Similarly, the predicate lifting for $\Diamond$ is $\lambda_{\{\{\top,\bot\},\{\bot\}\}} = \lambda_{\{\top,\bot\}} \cup \lambda_{\{\bot\}}$. Incidentally, $\Diamond$ does have a translator, see Example 4.6.

The starting point of the present paper was the discovery that singleton liftings always have translators.

Theorem 4.14 *Let T be a weak pullback preserving functor. Then each singleton lifting λ_p has a translator. Moreover, the translator is associated with $T(\{-\}_{\mathcal{Q}})(p)$.*

Proof. Consider the following diagram



In the diagram, Y denotes the isomorphism given by Yoneda Lemma. Since T preserves weak pullbacks $\boldsymbol{\nabla}$ is natural (Proposition 2.5); therefore, due to the Yoneda Lemma 3.17, the parallelogram on the left commutes. The triangle on the right commutes by Proposition 2.5.

The commutativity of the diagram implies that the natural transformation associated with $T(\{-\}_{\mathcal{Q}(n)})(p)$ is a translator for λ_p. To see this call $\tau_p : \mathcal{Q}^n \longrightarrow T\mathcal{Q}$ the natural transformation associated with $T(\{-\}_{\mathcal{Q}(n)})(p)$. An element p, in $T\mathcal{Q}(n)$, is mapped by the lower edge of the diagram to $\tau_p \circ \boldsymbol{\nabla}$ whereas the upper edge maps it to λ_p. Since the diagram commutes we have $\lambda_p = \boldsymbol{\nabla} \circ \tau_p$ as we wanted to show. □

Remark 4.15 In the previous theorem we used T instead of T_ω; the reader may worry that we don't obtain a translator as in Definition 4.5. This is not a problem because T and T_ω coincide on finite sets and we are only considering predicate liftings of finite arity, i.e. elements (subsets) of $T\mathcal{Q}(n)$ for some finite n. More formally, for a finite n, we use the following chain of isomorphisms/equalities:

$$\mathsf{Nat}(\mathcal{Q}^n, T\mathcal{Q}) \cong T\mathcal{Q}\mathcal{Q}(n) = T_\omega\mathcal{Q}\mathcal{Q}(n) \cong \mathsf{Nat}(\mathcal{Q}^n, T_\omega\mathcal{Q}).$$

The reason to restrict to singleton liftings of finite arity is that we only consider the finitary version of Moss's logic (Definition 3.11). If we define Moss's logic using T instead of T_ω, the previous theorem holds for singleton liftings of possibly infinite arity.

Since all singleton liftings have translators, in order to translate $\mathcal{L} \longrightarrow \mathcal{M}$, it remains to make sure that (i) the "formula" $\tau(tr(\varphi))$ in Equation (19) is expressible in the logic and that (ii) all predicate liftings can be expressed using singleton liftings and basic propositional operations. Both (i) and (ii) depend on which basic propositional logic one chooses.

4.3 The basic propositional logic matters

In this subsection we investigate how the choice of the basic propositional logic affects the existence of a translation $\mathcal{L} \longrightarrow \mathcal{M}$. Recall that a translator τ will be used to inductively define a translation $tr(\lambda\varphi) = \nabla\tau(tr(\varphi))$. In order to ensure that $\tau(tr(\varphi))$ is expressible in the base logic given by a category $\mathcal{A}$, we need that the translator τ is what we call an $\mathcal{A}$-logical translator.

Definition 4.16 Let λ be an n-ary predicate lifting, $\mathcal{A}$ a category with power-set algebras, and $U : \mathcal{A} \longrightarrow \mathsf{Set}$ the forgetful functor. An *$\mathcal{A}$-logical translator* τ for λ is a natural transformation $\tau : U^n \longrightarrow T_\omega U$ such that τ_P is a translator for λ (recall that $UP = \mathcal{Q}$).

We often call an $\mathcal{A}$-logical translator a *logical translator* or an $\mathcal{A}$-translator. We say that the logical translator τ extends the translator τ_P. A predicate lifting λ is said to be $\mathcal{A}$-translatable if there exists an $\mathcal{A}$-translator for λ.

The slogan to remember here is: a logical translator is a translator for which we can replace $\mathcal{Q}$ by U (the forgetful functor of $\mathcal{A}$). Here are some illustrations of logical translators. The first item shows that we can not always replace $\mathcal{Q}$ by U, in other words, not all translators can be extended.

Example 4.17 (i) Consider $\mathcal{A} = \mathsf{DL}$ and $T = Id$ and the predicate lifting $\lambda_\perp : \mathcal{Q} \longrightarrow \mathcal{Q}$ given by complementation. In this case $\boldsymbol{\nabla}_{Id} : Id\mathcal{Q} \longrightarrow \mathcal{Q}Id$ is the identity. From this we see that complementation $\neg : \mathcal{Q} \longrightarrow \mathcal{Q}$ is a translator for $\lambda_\perp$. Since the base category of the coalgebraic logics is distributive lattices, all the operators in $\mathcal{M}_{Id}$ are monotone, therefore all the definable predicate liftings are monotone, which implies that negation is not definable. In other words, we cannot translate $\lambda_\perp$ into $\mathcal{M}_{Id}$. To summarise in the terminology of the previous definition, $\tau = \neg$ does not extend to a DL-translator (but, of course, it does extend to a BA-translator).

(ii) Consider the predicate lifting associated with the existential modality $\Diamond$ as in Example 4.6. We define a BA-translator τ as follows: Given a Boolean algebra $\mathfrak{A}$, with carrier A, the function $\tau_{\mathfrak{A}} : A \longrightarrow \mathcal{P}A$ maps an element $x \in A$ to $\tau_{\mathfrak{A}}(x) = \{x, \top\}$; τ induces the following translation $tr(\Diamond\varphi) = \nabla\{tr(\varphi), \top\}$. This is also a DL-translator but not a Set-translator.

(iii) Consider the probabilistic modality $\Diamond_p$. We define a DL-translator $\tau : U \longrightarrow \mathcal{D}U$ as follows: let $\mathfrak{A}$ be a distributive lattice with carrier set A. The $\mathfrak{A}$ component of τ maps $a \in A$ to the probability distribution $D^a_p : A \longrightarrow [0,1]$ assigning probability p to a and $1-p$ to $\top$. Compare with Example 4.6.

(iv) Consider the probabilistic modality $\Diamond^p$. We define a BA-translator $\tau : U \longrightarrow \mathcal{D}U$ as follows: let $\mathfrak{A}$ be a boolean algebra with carrier set A. The $\mathfrak{A}$ component of τ maps $a \in A$ to the probability distribution $D^p_a : A \longrightarrow [0,1]$ assigning probability p to $\neg a$ and $1-p$ to $\top$. Clearly, this can not be regarded as a DL-transaltor. Compare with Example 4.6.

(v) Consider the natural transformation $\eta : Id \longrightarrow \mathcal{P}$ which maps an element x to $\{x\}$. If we precompose η with $\mathcal{Q}$ we obtain an natural transformation $\tau_{\mathcal{Q}} : \mathcal{Q} \longrightarrow \mathcal{P}\mathcal{Q}$ which maps a set $\varphi \subseteq X$ to $\{\varphi\}$. This is a BA-translator for the predicate lifting $\lambda_\top : \mathcal{Q} \longrightarrow \mathcal{Q}\mathcal{P}$ which maps a set $\varphi \subseteq X$ to the its set of non empty subsets. The translator $\tau_{\mathcal{Q}}$ induces the following translation $tr(\lambda_\top\varphi) = \nabla\{\varphi\}$. Notice that this translator is a $\mathcal{A}$-translator for any category $\mathcal{A}$ of power set algebras.

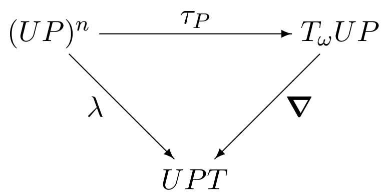
(vi) Generalizing the previous example, we can ask which predicate liftings have $\mathcal{A}$-translators for all categories $\mathcal{A}$ of power-set algebras. These are precisely what we call the Moss liftings (Diagram (24) on page 32), see Remark 5.16.

The main property of logical translators, as suggested by the previous examples, is that they produce one-step translations:

Lemma 4.18 *Every logical translator induces a one-step translation.*

Proof. Let $U : \mathcal{A} \longrightarrow$ Set be a category with power set algebras. And let $\tau : U^n \longrightarrow T_\omega U$ be an $\mathcal{A}$-logical translator for a predicate lifting λ of arity n. We want to define a one-step translation $L_\lambda \longrightarrow M_T$; recall that $L_\lambda = F(U^n)$ and $M_T = FT_\omega U$; the semantics are given by the F-adjoints of λ and $\boldsymbol{\nabla}$, respectively. We write $\lambda^\#$ and $\boldsymbol{\nabla}^\#$ for these adjoints.

The one step translation is given by $F(\tau)$. Since τ natural so is $F(\tau)$. It is only left to show that it commutes with the semantics. By definition of logical translator, τ_P is a translator for λ, this means that the following diagram



commutes, recall $UP = \mathcal{Q}$. Now by properties of adjoints we can move U to the left and obtain $\nabla^\# F(\tau_P) = \lambda^\#$. In other words $F(\tau)$ is a one step translation. $\square$

The next proposition shows how the previous argument can be extended to sets of predicate liftings (recall L_Λ from Definition 3.22).

Proposition 4.19 *Let Λ be a set of predicate liftings, each of which has a logical translator. Then we can find a one-step translation $\bar{L}_\Lambda \longrightarrow M$.*

Proof. Recall that $\bar{L}_\Lambda = \coprod_{\lambda \in \Lambda} FU^{n_\lambda}$ and $M = FT_\omega U$, where n_λ is the arity of λ. By assumption for each $\lambda \in \Lambda$ there is a logical translator $\tau_\lambda : U^{n_\lambda} \longrightarrow T_\omega U$. Using the universal property of coproducts we combine those into a map $\Upsilon : \coprod_{\lambda \in \Lambda} U^{n_\lambda} \longrightarrow T_\omega U$, the image of this map under F is the required translation $\bar{L}_\Lambda \longrightarrow M$. In order to see this recall that since F is a left adjoint, of U, it preserves colimits i.e $F(\coprod_{\lambda \in \Lambda} U^{n_\lambda}) \cong \coprod_{\lambda \in \Lambda} FU^{n_\lambda}$; hence $F(\Upsilon) = \coprod_{\lambda_\tau} F(\tau_\lambda)$. Since each $F(\tau_\lambda)$ is a one step translation, previous lemma, so is $F(\Upsilon)$. $\square$

To summarise, all $\mathcal{A}$-logical translators give rise to one-step translations. To obtain a translation from a translator we need to extend it to a logical translator. Such an extension is not always possible; the possibility of extending a translator rest on the properties of the category $\mathcal{A}$. We are now going to show that all translators do extend to BA-logical translators.

4.4 Translating Boolean Coalgebraic Logics: From $\mathcal{L}$ to $\mathcal{M}$

In this section, we will produce a one-step translation (Definition 4.1) from the logic of all predicate liftings $\mathcal{L}_T$ (Definition 3.22) to Moss's logic $\mathcal{M}_T$ (Definition 3.12). The main technical result is that that translators (Definition 4.5) can always be extended to BA-translators (Definition 4.16).

Lemma 4.20 *Every translator $\tau : \mathcal{Q}^n \longrightarrow T_\omega \mathcal{Q}$ can be extended to a BA-translator, i.e. a natural transformation $U^n \longrightarrow T_\omega U$, where $U : \mathsf{BA} \longrightarrow \mathsf{Set}$ is the forgetful functor.*

Proof. Recall from Section 2.4: (i) every Boolean algebra is the directed colimit of finite Boolean algebras, (ii) every finite Boolean is (isomorphic) to a power set algebra, (iii) every algebra Boolean morphism $A \longrightarrow B$ with A finite and $B = PY$, for some possibly infinite Y, arises from the inverse image of a function between sets, (iv) $Fn = P\mathcal{Q}n$ for finite sets n.

Let $\tau : \mathcal{Q}^n :\longrightarrow T_\omega \mathcal{Q}$ be a translator for a predicate lifting λ. We want to show than we can extend τ to all boolean algebras, i.e. we want to replace $\mathcal{Q}$ by $U : \mathsf{BA} \longrightarrow \mathsf{Set}$. In order to do this we first restrict τ to finite sets; call this restriction τ_ω. Because of (ii), in $\tau_\omega : \mathcal{Q}^n \longrightarrow T_\omega \mathcal{Q}$ we can replace $\mathcal{Q}$ by $U_\omega : \mathsf{BA}_\omega \longrightarrow \mathsf{Set}$, the restriction of the forgetful U to finite

Boolean algebras; because of (iii) this restriction is in fact natural in U_ω. More explicitly, using (ii) and (iii) we can restrict τ to a natural transformation $\tau_\omega : U_\omega^n \longrightarrow T_\omega U_\omega$. Because of (i), we can extend τ_ω to all Boolean algebras, i.e. to a natural transformation $\tau' : U \longrightarrow T_\omega U$; this is the logical translator we are looking for.

It is only left to check that $\tau'_P = \tau$. By definition, $\tau'_{P(X)} = \tau_X$ on finite X, we now show that this is also the case for any power set algebra. Let $P(Y)$ be a power set algebra and let $F(n_i) \xrightarrow{h_i} P(Y)$ be a diagram expressing $P(Y)$ as a colimit; this can be done because of (i); in fact this holds for any finitary equational theory.

Consider the following diagram

$$\begin{array}{ccccc} PY & \quad & UPY = \mathcal{Q}Y & \xrightarrow{\quad ? \quad} & T_\omega \mathcal{Q}Y \\ {\scriptstyle h_i}\uparrow & & {\scriptstyle Uh_i}\uparrow & & \uparrow{\scriptstyle TUh_i} \\ Fn_i & & UFn_i = \mathcal{Q}\mathcal{Q}n_i & \xrightarrow{\tau'_{PX} = \tau_X} & T_\omega \mathcal{Q}\mathcal{Q}n \end{array}$$

the equalities in the lower row hold because of (iv). Since T_ω is finitary and U is a left adjoint, they preserves the colimit $F(n_i) \xrightarrow{h_i} P(Y)$. From this, the equality $\tau'_P = \tau$ will follow once we show that putting $\tau'_{P(Y)}$ or τ_Y in the upper row of the diagram makes it commute. Indeed, for $\tau'_{P(Y)}$ this holds by definition. And for τ_Y, by (iii), we have that each h_i is f_i^{-1} for some function $f_i : Y \longrightarrow \mathcal{Q}n_i$; since τ is natural in $\mathcal{Q}$, this means that τ_Y also makes the diagram commute. This concludes the proof.

□

Remark 4.21 The previous lemma strongly depends on the category BA. Its extensions to other categories (of power set) algebras is a topic for further research. For example, for the category of distributive lattices we should modify the notion of predicate lifting or the proof will not work, the reason being that there are finite distributive lattices which are not power set algebras.

An immediate corollary is that we can translate singleton liftings.

Corollary 4.22 *Every singleton lifting for a weak pullback preserving functor T, can be translated into Moss logic for T on* BA.

Proof. Let λ be singleton lifting. By Theorem 4.14 it has a translator τ. By the previous theorem, τ can be extended to a BA-translator. By Lemma 4.18 this induces a one-step translation, i.e. λ can be translated. □

The following translations illustrate the previous corollary.

Example 4.23 (i) The translations in Example 4.17 are instantiations of the previous theorem.

(ii) Let λ_* the predicate lifting that indicates termination (Example 4.10 item (iii)). The constant natural transformation $\tau : A \longrightarrow 1 + A$ into 1 is a translator for λ_*. This is in fact an $\mathcal{A}$-translator for any category $\mathcal{A}$ of power set algebras. The induced translation is $tr(\lambda_*\varphi) = \nabla *$, where $* \in \mathcal{M}_T$. Notice that $*$ is a formula in $\mathcal{M}_T$ and does not depend on the underlying propositional logic (see Remark 3.12).

(iii) Let (n, m) be singleton lifting for the finite multiset functor (Example 4.10). We define a BA-translator for (n, m) as follows: Given a Boolean algebra $\mathfrak{A}$, with carrier

A, the function $\tau_{\mathfrak{A}} : A \longrightarrow \mathcal{B}_{\mathbb{N}}A$ maps an element $x \in A$ to the bag: $B_{(x,n,m)} : A \longrightarrow \mathbb{N}$

$B_{(x,n,m)}(x) = n, B_{(x,n,m)}(\neg_{\mathfrak{A}} x) = m$ and $B_{(x,n,m)}(a) = 0$ for any other element.

This logical translator induces the following translation $t((n,m)a) = \nabla B_{(t(a),n,m)}$.

(iv) Let $q \in [0,1]$ be a singleton lifting for the distribution functor. We define a BA-translator as follows: $\tau : A \longrightarrow \mathcal{D}A$ maps an element a to the probability distribution $\mu_a : A \longrightarrow [0,1]$ which assigns q to a, $1-q$ to $\neg a$, and 0 to any other element. As shown in Example 4.10, item (vi), the induced translation is $tr(\lambda_q\varphi) = \nabla\{(\varphi, q), (\neg_X\varphi, 1-q)\}$

We can do even better and show that all predicate liftings can be translated provided that the functor also preserves finite sets.

Theorem 4.24 *If T preserves finite sets and weak pullbacks, there is a one-step translation $\bar{L}_T \longrightarrow M_T$.*

Proof. Let $\bar{L}_s$ be the functor given as in Definition 3.22, but using only singleton liftings. Because T preserves finite sets, every predicate lifting can be expressed as a finite join of singleton liftings (Proposition 4.12), hence we have an isomorphism $\bar{L} \cong \bar{L}_s$. Now let λ be a singleton lifting and let τ be the corresponding translator (Theorem 4.14). Obtain a one-step translation $L_\lambda \longrightarrow M_T$ as in the previous corollary. Doing this for each singleton lifting and combining all of these logical translators, as in Proposition 4.19, we obtain a translation $\bar{L}_s \longrightarrow M_T$. □

Note that Examples 4.17(i), and 4.8 show that in order to translate all predicate liftings, we need classical propositional logic. Weak pullback preservation is needed because otherwise Moss's language is not defined. The following example shows that the condition of T preserving finite sets can not be dropped.

Example 4.25 Let T be the constant functor with value $\mathbb{N}$, let $E \subseteq \mathbb{N}$ be the set of even numbers. If we are working over BA, the predicate lifting λ_E can not be translated into Moss's language over BA. Consider the coalgebra $N = (\mathbb{N}, 1_{\mathbb{N}})$ and the formula $\lambda_E\top$. On the one hand, this formula defines the set of even numbers, i.e. $[\![\lambda_E\top]\!] = E$. On the other hand, we can check that using Moss's language we can only define finite and cofinite sets; therefore we conclude that the predicate lifting λ_E can not be translated.

The following translations illustrate the previous theorem.

Example 4.26 (i) The predicate lifting for $\Box$ is $\lambda_{\{\emptyset,\{\top\}\}}$. It does not have a translator but is the union $\lambda_\emptyset \cup \lambda_{\{\top\}}$ of two singleton liftings, which have a translator, see Example 4.10. In this case, the induced translation is the usual one i.e. $tr(\Box\varphi) = \nabla\emptyset \vee \nabla\{\varphi\}$.

(ii) In the case of the existential modality, the predicate lifting for $\Diamond$ is $\lambda_{\{\{\top,\bot\},\{\bot\}\}} = \lambda_{\{\top,\bot\}} \cup \lambda_{\{\bot\}}$. Incidentally, $\Diamond$ does have a translator, see Example 4.6, which induces the usual translation $tr(\Diamond\varphi) = \nabla\{\top, \varphi\}$. However, we could also translate $\Diamond$ using the translators for $\lambda_{\{\top,\bot\}}$ and $\lambda_{\{\bot\}}$; in such perspective we have

$$\begin{aligned} tr(\Diamond\varphi) &= tr(\lambda_{\{\top,\bot\}}\varphi) \vee tr(\lambda_{\{\bot\}}\varphi) \\ &= \nabla\{\varphi, \neg\varphi\} \vee \nabla\{\neg\varphi\}. \end{aligned}$$

It can be checked, by long direct computations, that this is indeed equivalent to the usual translation.

(iii) Even though we can translate singleton liftings for $\mathcal{B}_{\mathbb{N}}$ and $\mathcal{D}$, see above, we can not use the previous theorem to conclude that the standard logics for these functorsare translatable into Moss logic because these functors do not preserve finite sets. In case of $\mathcal{D}$, Example 4.17 shows that sometimes we can. The case of $\mathcal{B}_{\mathbb{N}}$ shows that this might also fail, see Examples 3.14 and 4.8.

4.5 Translating Boolean Coalgebraic Logics: From $\mathcal{M}$ to $\mathcal{L}$

Our next step is to find a translation $M_T \to L_T$. Note that we do not expect a natural transformation $M_T \to \bar{L}_T$ because each ∇-formula corresponds to many different but equivalent formulas of $\mathcal{L}_T$ (see also the next section). So we make use of the fact that L_T is a quotient of $\bar{L}_T$.

Theorem 4.27 *For all weak pullback preserving functors T there exists a one-step translation $M_T \to L_T$.*

Proof. Recall that for finite n we have $F(n) = P\mathcal{Q}(n)$ and $L_T(F(n)) = PT\mathcal{Q}(n)$. From this, we can see that the semantics of Moss's logic $\boldsymbol{\nabla} : T_\omega UP \to UPT$ on $\mathcal{Q}(n)$ can be written $\boldsymbol{\nabla}_{\mathcal{Q}n} : T_\omega UFn \to UL_TFn$. Since U is a right adjoint and definition of M_T this yields $M_TFn \to L_TFn$. Since both M_T and L_T are determined by their action on finitely generated free algebras, this gives the desired translation $M_T \to L_T$. □

Again, the theorem is specific to BA. In particular, both translations $\bar{L}_T \to M_T$ and $M_T \to L_T$ made use of the fact that in case of BA we have $Fn = P\mathcal{Q}n$ for finite n.

On the other hand, Theorem 4.27 is a particular instance of a more general Lindström like theorem showing that (L_T, δ_T) is the most expressive Boolean logic for T-coalgebras; recall Remark 3.4 (see also [33] for more on coalgebraic Lindström theorems).

Theorem 4.28 *Assume that T preserves finite sets and that (L, δ) is a Boolean logic for T-coalgebras. Then (L_T, δ_T) is at least as expressive as (L, δ), that is, there is a one-step translation $\tau : L \to L_T$. Moreover, if L is one-step complete and expressive, then τ is an isomorphism.*

Proof. It follows from the assumptions that $(\delta_T)_X : L_TP(X) \to PT(X)$ is an isomorphism on finite sets X. Take now the following composite $LP(X) \xrightarrow{\delta} PT(X) \xrightarrow{\delta_T^{-1}} L_TP(X)$ on finite X. As in the proof of Theorem 4.27, this determines a translation $L \to L_T$ on finitely generated free Boolean algebras and hence on all Boolean algebras. □

5 Equational Coalgebraic Logic

Moss's logic is based on the non-standard modality $\nabla\alpha \in \mathcal{M}$. Our aim in this section is to use our translation techniques developed in the previous sections to present Moss's logic using only standard modalities, i.e. predicate liftings of finite arity. We also show how the axiomatization of Moss's logic from [27] gives rise to a standard equational axiomatization.

One advantage of such an equational version of Moss's logic is that one can reuse known logical methods. For example, in a logic given by predicate liftings, the subformulas of a formula $\lambda(\varphi_1, \dots \varphi_n)$ are the φ_i. But what should be the subformulas of $\nabla\alpha$, if all we know about α is that $\alpha \in T_\omega(\mathcal{M}_T)$? Or how to state that ∇ is monotone? Or what does congruence mean? All these questions can be answered, see e.g. [27], but it requires some adhoc technical work, which is avoided in the equational presentation.

Another way to compare the work in this section with [36,27] is that we give an implementation of Moss's modality using the datatype of lists. For example, in the case of $T = \mathcal{P}$ we write $\nabla\{\varphi_1, \dots \varphi_n\}$, applying ∇ to a *set* of formulas. Instead, we can represent $\nabla\{\varphi_1, \dots \varphi_n\}$ by a list $[\varphi_1, \dots \varphi_n]$, or, in other words, by a standard n-ary modal operator applied to its arguments $\varphi_1, \dots \varphi_n$. This can be done for any set-functor T, as we are going to recall now.

5.1 Presentations of Functors

Our tool to provide a standard axiomatisation of $\mathcal{M}_T$ are translators. The key idea is to use (logical) translators which can be obtained using presentations of T.

Definition 5.1 A *finitary* presentation of a functor T is a polynomial functor Σ together with a surjective natural transformation

$$\Sigma X = \coprod_{n<\omega} \Sigma_n \times X^n \xrightarrow{E_X} TX. \tag{21}$$

Such a quotient is also called a **presentation $\langle \Sigma, E \rangle$ of T by operations and equations**. Σ_n is called the set of operations of arity n and the equations defining T are the kernel of E_X (for some countably infinite set of 'variables' X) (for more on set-functors and their presentations see Adámek and Trnková [6]).

A functor has a finitary presentation iff the functor is finitary, in which case a presentation can be obtained as follows.

Example 5.2 Every finitary functor T has a **canonical presentation**

$$E : \coprod Tn \times X^n \longrightarrow TX.$$

For $p \in Tn$ and $a : n \longrightarrow X$, we define $E(p, a) = (Ta)(p)$.

We fix some terminology before proceeding.

Definition 5.3 Given a presentation $\langle \Sigma, E \rangle$ we say that (p, a) represents $\alpha \in TX$, or that (p, a) is a representative of α, if $E(p, a) = \alpha$.

The canonical presentation is usually not the most "natural" one. This is illustrated in the following example.

Example 5.4 The finite powerset functor $\mathcal{P}_\omega$ has the canonical presentation

$$\coprod_{n<\omega} \mathcal{P}_\omega(n) \times X^n \longrightarrow \mathcal{P}_\omega X$$

and a standard presentation

$$\mathsf{List} X = \coprod_{n<\omega} X^n \longrightarrow \mathcal{P}_\omega X.$$

The canonical presentation maps an element $(p,a) \in \mathcal{P}(n) \times X^n$ to the set $\bigcup\{a_i \mid i \in p\}$, i.e. restricts the list a to the components in the set p. The standard presentation maps a list $a : n \longrightarrow X$ to $\bigcup\{a_i \mid i \in n\}$. In both cases, two elements in $\mathsf{List}X$ and $\coprod \mathcal{P}(n) \times X^n$, respectively, are identified if they define the same subset of X.

Notice that we can identify the list $[x_1, \dots x_n]$ of the standard presentation of $\mathcal{P}$ with $(\{1, \dots n\}, [x_1, \dots x_n])$ of the canonical presentation of $\mathcal{P}$. This is an instance of a more general observation: Any presentation is a restriction of the canonical presentation. The next lemma makes this precise.

Lemma 5.5 *Consider a presentation $\langle \Sigma, E\rangle$ of T. There are canonical maps $s_n : \Sigma_n \longrightarrow Tn$ such that $E_X(p,a) = Ta(s(p))$ for all $p \in \Sigma_n$ and all $a : n \longrightarrow X$.*

Proof. Consider $E_n : \coprod_{k<\omega} \Sigma_k \times n^k \longrightarrow Tn$. For $p \in \Sigma_n$, define $s_n(p) = E_n(p, \mathrm{id}_n)$. Since E_X is natural in X, we have $E_X(p,a) = Ta(E_n(p, \mathrm{id}_n))$, which proves the claim.□

In other words, given any presentation $\langle \Sigma, E\rangle$ of T, we can identify an operation in Σ_n with the corresponding operation in Tn of the canonical presentation.

To emphasise the equational axiomatisation given by a presentation of T we introduce the following notation.

Notation 5.6 *Given $(p,a), (q,b) \in \Sigma_n \times X^n$, we write $p(a)$ for (p,a) and $q(b)$ for (q,b) and*

$$p(a) \approx_T q(b) \quad \textit{iff} \quad E_X(p,a) = E_X(q,b) \text{ (i.e. iff } Ta(p) = Tb(q) \text{)}.$$

This to emphasise that p and q denote operators acting on lists. Note that $\approx_T$ depends on the given presentation of T, so in case of danger of confusion we might write $\approx_{\langle\Sigma,E\rangle}$ instead.

Example 5.7 (i) For $T = Id$, the identity is itself a presentation of T; in this case $\approx_{Id}$ is equality.

(ii) For the functor $T = 1 + Id$ the canonical presentation maps a pair $(p,a) \in (1+n) \times X^n$ to $* \in 1 + X$ in case $p \in 1$ or to a_p, the evaluation of a in p, in any other case. The congruence relation is then $p(a) \approx_T q(b)$ iff $p = q = *$ or $a_p = b_q$.

(iii) In the case of the canonical presentation for $\mathcal{P}$ the relation $\approx_T$ can be described as follows: for a pair of elements $(p,a) \in \mathcal{P}(n) \times X^n$ and $(p,q) \in \mathcal{P}(m) \times X^m$ we have $p(a) \approx_T q(b)$ iff $\{a_i \mid i \in p\} = \{q_j \mid j \in q\}$.

(iv) In the case of the List-presentation of $\mathcal{P}$, the relation $\approx_{\mathsf{List}}$ has the following characterization: for $a \in X^n$ and $b \in X^m$ we have $a \approx_{\mathsf{List}} b$ iff $\{a_i \mid i \in n\} = \{b_j \mid j \in m\}$.

(v) For $T = \mathcal{B}_{\mathbb{N}}$ the canonical presentation can be described as follows: a pair $(p,a) \in \mathcal{B}_{\mathbb{N}}(n) \times X^n$ is mapped to the bag $b : X \longrightarrow \mathbb{N}$ mapping x to $\sum_{\{i|a_i=x\}} p_i$. The relation $\approx_T$ can be described as in Example 3.14. More explicitly, $p(a) \approx_T q(b)$ for $p : n \longrightarrow \mathbb{N}$, $q : m \longrightarrow \mathbb{N}$ iff there is a matrix $(r_{ij})_{1\le i\le n, 1\le j\le m}$ such that $a_i \neq b_j \Rightarrow r_{ij} = 0$ and $\sum_i r_{ij} = q_j$ and $\sum_j r_{ij} = p_i$. For example, $[3,2](x,y) \approx_T [2,1,1,1](x,y,x,y)$. The case of probability distributions is similar.

The following application of Lemma 5.5 will be useful later.

Proposition 5.8 *Let $\langle \Sigma, E \rangle$ be a presentation of T and assume the following diagram commutes*

$$\begin{array}{ccc} n & \xrightarrow{\quad a \quad} & X \\ & {\scriptstyle f}\searrow \quad \nearrow{\scriptstyle b} & \\ & m & \end{array}$$

If $p \in \Sigma_n$ and $Tf(p) \in \Sigma_m$ then $p(a) \approx_{\langle \Sigma, E \rangle} Tf(p)(b)$.

One of the uses of presentations is that we can compute relation liftings (Section 2.3) "hiding" the functor T. This is formalised in the following technical lemma, which is a key stone for our development of equational coalgebraic logic.

Lemma 5.9 *Let $\langle \Sigma, E \rangle$ be a presentation for a finitary endofunctor T on* Set *and let R be a relation between X and Y. For every $\alpha \in TX$ and $\beta \in TY$ the following conditions are equivalent:*

- $\alpha \, \overline{T}(R) \, \beta$.
- *There exists $k < \omega$, $r \in \Sigma_k$, $a : k \longrightarrow X$, and $b : k \longrightarrow Y$ such that $E_X(r, a) = \alpha$, $E_Y(r, b) = \beta$, and $(\forall i \in k)(a_i \, R \, b_i)$.*

More informally, we read the lemma as

$$t_x \, \overline{T}(R) \, t_y \;\text{ iff }\; t_x \approx_T r(a_1, \ldots a_k) \text{ and } t_y \approx_T r(b_1, \ldots b_k) \text{ and } a_i R b_i \tag{22}$$

for some k-ary operation r.

As we said before, depending on the functor T, relation liftings can be quite complicated, see e.g. Example 3.14. But for polynomial functors relation lifting is just the same relation componentwise, plus equality on the operations. The importance of Equation (22) is that it presents the relation lifting for T componentwise, i.e. in the form of a relation lifting for a polynomial functor, modulo the equational theory $\approx_T$. In other words, Equation (22) "hides" T using the equational theory. Polynomial functors can in fact be characterised as those functors which have a presentation such that $\approx_{\langle \Sigma, E \rangle}$ is equality.

Proof. The proof of the lemma is straightforward from contemplating the following commuting diagram

$$\begin{array}{ccccc} \Sigma X & \xleftarrow{\Sigma(\pi_X)} & \Sigma R & \xrightarrow{\Sigma(\pi_Y)} & \Sigma Y \\ {\scriptstyle E_X}\downarrow & & \downarrow{\scriptstyle E_R} & & \downarrow{\scriptstyle E_Y} \\ TX & \xleftarrow[T(\pi_X)]{} & TR & \xrightarrow[T(\pi_Y)]{} & TY \end{array} \tag{23}$$

and taking into account that E_R is surjective. More explicitly, from the definition of relation lifting (Definition 2.3) we have that $\alpha \overline{T}(R) \beta$ iff there exists $t \in T(R)$ such that $T(\pi_1)(t) = \alpha$ and $T(\pi_2)(t) = \beta$. Since E_R is surjective, this is possible if and only if there exists $(r, c) \in \Sigma(R)$ such that $E_R(r, c) = t$. Since Σ is polynomial functor, there exists $k \leq \omega$ such that $(r, c) \in \Sigma_k \times R^k$. These are the k and r required in the statement of the lemma. The functions a and b are obtained by composing of $k \xrightarrow{c} R \hookrightarrow X \times Y$ with the respective projections; this is equivalent to $(\forall i \in k)(a_i \, R \, b_i)$. The commutativity of the diagram says that $E_X(r, a) = \alpha$ and $E_Y(r, b) = \beta$. □

5.2 The logic $\mathcal{K}_T$

We now use presentations of T to define a modal logic which is equivalent to Moss's logic $\mathcal{M}_T$ but built from standard modal operators.

Definition 5.10 Given a presentation $\langle \Sigma, E\rangle$ of T_ω, each $p \in \Sigma_n$ gives rise to an n-ary predicate lifting λ^p, with a translator as in Definition 4.5, as shown in the next diagram:

$$\begin{array}{ccc} \mathcal{Q}^n & \xrightarrow{E_\mathcal{Q}(p,-)} & T_\omega \mathcal{Q} \\ & {\scriptstyle \lambda^p} \searrow \quad \swarrow {\scriptstyle \boldsymbol{\nabla}} & \\ & \mathcal{Q}T & \end{array} \tag{24}$$

We call a predicate lifting arising in this way a $\langle \Sigma, E\rangle$-**Moss lifting**, or simply a Moss lifting. By Lemma 5.5 the set of Moss liftings for a presentation $\langle \Sigma, E\rangle$ can be identified with a subset of $\coprod_{n<\omega} T_\omega(n)$.

Remark 5.11 We know from Proposition 3.17 that n-ary predicate liftings are in 1-1 correspondence with maps $T(2^n) \longrightarrow 2$. The Moss liftings λ^p among those are the ones which are given by $\lambda^p(t) = \boldsymbol{\nabla}_n(t)(p)$ where we recall $\boldsymbol{\nabla}_n : T(2^n) \longrightarrow 2^{Tn}$.

Example 5.12 (i) Let $T = 1 + Id$ (deterministic transition systems with termination). In the case of the canonical presentation for each arity n there is a Moss lifting λ^*_n, which indicates termination; this lifting corresponds to the unique element of 1. All other Moss liftings of arity n correspond to the elements of n. For $p \in n$, the Moss lifting λ^p maps a sequence $\varphi : n \longrightarrow \mathcal{Q}X$ to the set φ_p. Using the identity presentation $1 + Id \longrightarrow T$, we see that one constant and one unary predicate lifting suffice to describe T-coalgebras.

(ii) Let $T = \mathcal{P}$ (non-deterministic transition systems). Moss liftings of arity n for the canonical presentation are associated with subsets of n. Let p be one of those subsets. The Moss lifting λ^p maps a sequence $\varphi : n \longrightarrow \mathcal{Q}X$ to the set

$$\begin{aligned} \lambda^p(\varphi) &= \{\alpha \in \mathcal{P}X \mid (\forall x \in \alpha)(\exists i \in p)(x \in \varphi_i) \wedge (\forall i \in p)(\exists x \in \alpha)(x \in \varphi_i)\} \\ &= \{\alpha \in \mathcal{P}X \mid \alpha \subseteq \bigcup_{i \in p} \varphi_i \wedge (\forall i \in p)(\alpha \cap \varphi_i \neq \emptyset)\}. \end{aligned}$$

(iii) Let $T = \mathcal{P}$ and let $\langle \Sigma, E\rangle$ be the List-presentation. For each arity there is only one Moss lifting which in this case we write **[n]**. The Moss lifting **[n]** maps a sequence $\varphi : n \longrightarrow \mathcal{Q}X$ to the set

$$\mathbf{[n]}(\varphi) = \{\alpha \in \mathcal{P}X \mid \alpha \subseteq \bigcup_{i \in n} \varphi_i \wedge (\forall i \in n)(\alpha \cap \varphi_i \neq \emptyset)\}$$

(iv) Let T be the finite multiset functor. A Moss liftings of arity n corresponds to a bag $p : n \longrightarrow \mathbb{N}$. The associated predicate lifting maps $\varphi : n \longrightarrow \mathcal{Q}X$ to a multiset over $\mathcal{Q}X$ (Example 5.7(v)) followed by an application of $\boldsymbol{\nabla}$ (Example 3.14).

Definition 5.13 Given a presentation $\langle \Sigma, E\rangle$ of T_ω, the logic $\mathcal{K}_T^{\langle \Sigma, E\rangle}$ is the logic (Definition 3.22) given by the set of predicate liftings λ^p, $p \in \Sigma_n$. We write $\mathcal{K}_T$ if the presentation is clear from the context. The corresponding functor is denoted by $K_T : \mathsf{BA} \longrightarrow \mathsf{BA}$.

Remark 5.14 The functor K_T is isomorphic to $F\Sigma U$. Indeed, from Definition 3.22) we have $K_T = F(\coprod_{n<\omega} \coprod_{\Sigma_n} U^n)$. Since we are working on Set we have $\coprod_{\Sigma_n} U^n \cong \Sigma_n \times U^n$, hence $\coprod_{n<\omega} \coprod_{\Sigma_n} U^n = \Sigma U$, and then $K_T \cong F\Sigma U$ as predicted. In fact this work for any category of power set algebras.

Another use of presentations is that we can now translate Moss logic.

Proposition 5.15 ([34]) *For every formula in $\mathcal{M}_T$ there exists an equivalent formula in $\mathcal{K}_T$. More explicitly, for every $\varphi \in \mathcal{M}_T$ there exists $\psi \in \mathcal{K}_T$ such that $[\![\varphi]\!]_{(X,\xi)} = [\![\psi]\!]_{(X,\xi)}$ for every coalgebra (X, ξ).*

Moss liftings play a special role among the predicate liftings discussed in the previous section.

Remark 5.16 The translators $E_Q(p, -)$ for a Moss liftings λ^p is the restriction of the natural transformation $E(p, -) : Id^n \longrightarrow T$. In our terminology, $E(p, -)$ is a Set-logical translator extending $E_Q(p, -)$ (Definition 4.16). We can also restrict $E(p, -)$ with any functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$; this exhibit an $\mathcal{A}$-logical for λ^p for any category $\mathcal{A}$ with powerset algebras (Definition 3.5). The argument also works backwards, i.e. if a predicate lifting has an $\mathcal{A}$-translator for any category of powerset algebras it is a Moss lifting, because it should then have a Set-logical translator. In summary, Moss liftings are the only predicate liftings that can be translated independently of the underlying propositional logic. For this reason they may be called **totally-translatable**.

Another important property of Moss liftings is that they are monotone:

Proposition 5.17 *Let $\lambda^p : \mathcal{Q}^n \longrightarrow \mathcal{Q}T$ be a Moss lifting; let $\varphi, \psi : n \longrightarrow \mathcal{Q}X$ be sequences of sets. If $(\forall i)(\varphi_i \subseteq \psi_i)$ then $\lambda^p(\varphi) \subseteq \lambda^p(\psi)$.*

Proof. Let $E(p, -)$ be the translator of λ^p. Using Lemma 5.9 we see that $(\forall i)(\varphi_i \subseteq \psi_i)$ implies $E_Q(p, \varphi)\, \overline{T}(\subseteq)\, E_Q(p, \psi)$. Applying $\boldsymbol{\nabla}$ on both sides of the previous inequality will transform $\overline{T}(\subseteq)$ into $\subseteq$; we conclude $\lambda^p(\varphi) \subseteq \lambda^p(\psi)$. □

This has the following important corollary.

Corollary 5.18 *For every finitary weak pullback preserving functor T there exists a set Λ of monotone predicate liftings such that the logic L_Λ is expressive (Remark 3.4). The set Λ is that of Moss liftings.*

Proof. Since T preserves weak pullbacks we can define Moss's logic and in particular we have the Moss liftings. Proposition 5.15 implies that the language of Moss liftings is as expressive as Moss's language. Since Moss's logic is expressive [36] so is $\mathcal{K}_T$. From the previous Proposition (5.17) we know that Moss liftings are monotone. □

Remark 5.19 Finding a monotone set of predicate liftings is important in coalgebraic modal logic, as it opens the possibility of adding fix points operators. The previous proposition solves this problem in the case of weak-pullback preserving functors. As far as we know, the general problem for non-weak pullback preserving functors is still open.

5.3 Well-based and Basic Presentations

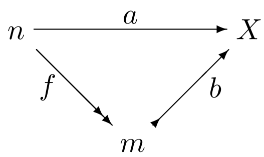
Now that we have introduced the logical system $\mathcal{K}_T$ for T-coalgebras, we would like to present a sound and complete axiomatization for it. To this end, we will need a more careful analysis of representations of an element in $T(X)$. The main point is the following.

Roughly speaking, for any finitary functor T and any $\alpha \in TX$ there is a smallest *finite* set $n \hookrightarrow X$ such hat $\alpha \in Tn$. We call this set n the base of α; basis plays a crucial role in the completeness proof of the axiomatisation of $\mathcal{M}_T$ in [27]. In order to replace $\mathcal{M}_T$ by $\mathcal{K}_T$ smoothly, we need that if T is presented by $\langle \Sigma, E \rangle$ and $\alpha \in TX$ has base n, then there is an n-ary operation symbol $p \in \Sigma_n$, called a basic operation, and an injective $a \in X^n$ such that (p, a) represents α. Such presentations will be called well based and this section studies their basic properties.

Recall that by Lemma 5.5, we can assume that all presentations $\langle \Sigma, E \rangle$ are given by specifying a subset $\Sigma_n \subseteq Tn$ with $\Sigma_n \times X^n \longrightarrow TX$ given by $(p, a) \mapsto E_X(p, a) = Ta(p)$. The next proposition shows that in fact we can also assume a to be injective; this will come in handy to simplify our proofs.

Proposition 5.20 *For each $(p, a) \in Tn \times X^n$, there exists (q, b) such that $q(b) \approx_T p(a)$ and b is injective.*

Proof. Assume $a : n \longrightarrow X$, factor as follows:



where f is onto and b is injective. Let $q = T(f)(p)$, by Proposition 5.8 we conclude $q(b) \approx_T p(a)$. □

Example 5.21 To illustrate the construction in the previous proof, consider the canonical presentation of List. The list $[x, x] \in \mathsf{List}(X)$ has a representative $[0, 1](x, x) \in \mathsf{List}(2) \times X^2$. We can factor it through $2 \longrightarrow 1$ to obtain the representative $[0, 0](x) \in \mathsf{List}(1) \times X^1$.

The next definition will allow us to avoid redundant representatives.

Definition 5.22 We define the category $\mathsf{IElem}(\alpha)$ of 'injective representatives' of $\alpha \in TX$ as follows: The objects of $\mathsf{IElem}(\alpha)$ are given by

$$\mathsf{IElem}_o(\alpha) = \bigcup_{n \in \mathbb{N}} \{(p, a) \in \Sigma_n \times X^n \mid (p, a) \text{ represents } \alpha, a \text{ injective}\}.$$

A morphism $f : (p, a) \longrightarrow (q, b)$, where $(p, a) \in Tn \times X^n$ and $(q, b) \in Tm \times X^m$, is a function $f : n \longrightarrow m$ such that $a = bf$ and $Tf(p) = q$.

We call (p, a) a *basic* representative of α if (p, a) is initial in $\mathsf{IElem}(\alpha)$, that is, $\forall (q, b) \in \mathsf{IElem}(\alpha)\ .\ \exists f : \mathrm{dom}(a) \longrightarrow \mathrm{dom}(b)\ .\ Tf(p) = q\ \&\ a = bf$. f is unique since b is injective.

A presentation $\langle \Sigma, E \rangle$ is *injective* if $\mathsf{IElem}_o(\alpha)$ is always inhabited. It is *well based* if every $\alpha \in T(X)$ has a basic representative.

Example 5.23 (i) The standard presentation and the canonical presentation for $\mathcal{P}$ are well based.

(ii) We call the standard presentation of List the one given by the identity $\coprod_{n<\omega} X^n \longrightarrow$ List. It is not well based as, for example, the list $[x, x]$ has no injective representative.

Thus not all presentations are well-based; however, canonical presentation are well based.

Proposition 5.24 *Canonical presentations are well based.*

Proof. Consider $(p, a : m \longrightarrow X), (q, b : n \longrightarrow X)$ in $\mathsf{IElem}(\alpha)$. Let $(f : k \longrightarrow m, g : k \longrightarrow n)$ be a pullback of a and b. Since T is standard (Remark 2.2), the following diagram

$$\begin{array}{ccc} Tk & \xrightarrow{Tg} & Tn \\ {\scriptstyle Tf}\downarrow & & \downarrow{\scriptstyle Tb} \\ Tm & \xrightarrow[Ta]{} & TX \end{array}$$

is a pullback. Therefore there exists $r \in Tk$ such that $Tf(r) = p, Tg(r) = q$.

Now let in the above be m the smallest number such that there is $(p, a : m \longrightarrow X)$ with $E(p, a) = \alpha$. Since b is injective so is f and by the choice of m we have that f must be iso. Hence we obtain $g \circ f^{-1} : m \longrightarrow n$ with $a = bgf^{-1}$ and $T(g \circ f^{-1})(p) = q$, in fact this is the only function with those two properties. In other words, every $\alpha \in TX$ is represented by a basic element in the canonical presentation. □

Remark 5.25 We actually proved a stronger statement: A presentation $\langle \Sigma, E \rangle$ is well based if 1) every $\alpha \in TX$ has a representative (p, a) with a injective and 2) $\langle \Sigma, E \rangle$ is stable under pullbacks. Here we say that $\langle \Sigma, E \rangle$ is **stable under pullbacks** if whenever

$$\begin{array}{ccc} k & \xrightarrow{b'} & n \\ {\scriptstyle a'}\downarrow & & \downarrow{\scriptstyle a} \\ m & \xrightarrow[b]{} & X \end{array}$$

is a pullback and $p(a) \approx_T q(b)$, and $r \in Tk$ is such that $Ta'(r) = q$ and $Tb'(r) = p$ then $r \in \Sigma_k$.

The next proposition shows that whether (p, a) is basic or not does not depend on a.

Proposition 5.26 *The following are equivalent.*

(i) *There exists X and injective $a : n \longrightarrow X$ such that (p, a) is a basic representative.*

(ii) *(p, id_n) is a basic representative.*

(iii) *(p, a) is a basic representative for all X and all injective $a : n \longrightarrow X$.*

Proof. (iii)⇒(ii)⇒(i) is trivial. For (i)⇒(ii), consider some set X and an injective $a : n \longrightarrow X$ such that (p, a) is basic for $\langle \Sigma, E \rangle$. Suppose that $(q, b) \in Tm \times n^m$ with injective $b : m \longrightarrow n$ represents the same element as (p, id_n), this element exists because $\langle \Sigma, E \rangle$

is injective. Then $(q, a \circ b)$ represents the same element as (p, a), hence we find an $f : n \to m$, which is the required arrow from (p, id) to (q, b); it is unique because (p, a) is basic.

For (ii)⇒(iii), assume (p, id_n) is basic and consider some injective $a : n \to X$. Suppose $(q, b) \in Tm \times X^m$ with injective $b : m \to X$ represents the same element as (p, a), such representation exists because $\langle \Sigma, E\rangle$ is injective. Let $b' : k \to n$ be the pullback of b along a, let a' be the pullback of a along b and denote by r the element of Tk such that $Tb'(r) = p$ and $Ta'(r) = q$. Notice that (r, b') represents the same element as (p, id_n). Since the latter is basic for $\langle \Sigma, E\rangle$ we have $k = n$ and b' iso. It follows that $a' \circ b'^{-1}$ is the arrow from (p, a) to (q, b) required to show that (p, a) is basic. □

Any two basic representations $(p, a : n \to X), (q, b : m \to X)$ of an element $\alpha \in TX$ are isomorphic in $\mathsf{IElem}(\alpha)$. In particular, a and b define the same subset $\mathrm{Im}(a) = \mathrm{Im}(b)$ of X. This justifies the following definition

Definition 5.27 Consider the canonical presentation $\langle \Sigma, E\rangle$ and a basic $(p, a) \in \Sigma X \times X^n$ with $E(p, a) = \alpha$. We call $\mathrm{Im}(a)$ the base of α and write $\mathit{Base}(\alpha) = \mathrm{Im}(a)$.

Example 5.28 We consider canonical presentations.

(i) For $T = \mathcal{P}$, there is exactly one basic operation in each $\mathcal{P}(n)$, namely the full set n.

(ii) For $T = \mathsf{List}$, the basic operations in $\mathsf{List}(n)$ are those lists that contain all elements of n (note that there are infinitely many basic operations of arity n for each $n > 0$). For example, $[0, 0]$ is a basic operation of arity 1 and $[0, 0](x)$ is the list we would usually write as $[x, x]$. Since List preserves inclusions, we have that $[0, 0]$ is also an operation of arity 2, but it is not a basic operation of arity 2.

The usual definition of a base, which is also the one used in [27], can be formulated for standard functors as follows: $\mathit{Base}(\alpha)$ is the smallest set $Y \subseteq X$ such that $\alpha \in TY$. The two definitions are equivalent.

The next remark summarises what we will need below for our soundness and completeness results.

Remark 5.29 If a presentation $\langle \Sigma, E\rangle$ is well based then for every X and every $\alpha \in TX$ there exists $(p, a) \in \Sigma_n \times X^n$ such that (i) (p, a) represents α and (ii) $\mathrm{Im}(a) = \mathit{Base}(\alpha)$.

Another property that is sometimes nice to have is that all operations are basic.

Definition 5.30 A presentation $\langle \Sigma, E\rangle$ is basic if all $p \in \Sigma_n$ are basic.

The next proposition shows that basic presentations are not rare.

Proposition 5.31 *Every well based presentation $\langle \Sigma, E\rangle$ contains a basic presentation $\langle \Sigma', E'\rangle$.*

Proof. We take

$$\Sigma'_n = \{p \in \Sigma_n \mid p \text{ is basic}\}. \tag{25}$$

The natural transformation

$$E' : \coprod \Sigma'_n \times (-)^n \to T$$

is defined by restricting E to Σ'_n. To show that E' is onto, pick $\alpha \in TX$ and let $(q, b : k \to X)$ be a representative of α. Since $\langle \Sigma, E\rangle$ is well based we can assume that q is basic and b is injective, but then, by definition, $q \in \Sigma'_k$. E' is well-based and basic by construction. □

Example 5.32 If we apply the procedure described above to the canonical presentation for $\mathcal{P}$ we obtain the standard presentation (see Example 5.4).

5.4 A complete equational proof system for $\mathcal{K}$

We present a proof system for $\mathcal{K}_T$. The soundness and completeness proof is based in a translation argument. We will translate the system for the ∇-logic presented in [27]; since this system is complete and sound [27], our system will be as well. The system in [27] is:

$$(\nabla 1)\quad \bigwedge\{\nabla\alpha \mid \alpha \in A\} \leq \bigvee\{\nabla T(\bigwedge)\Phi \mid \Phi \in SRD(A)\}.$$

$$(\nabla 2)\quad \nabla T(\bigvee)\Phi \leq \bigvee\{\nabla\alpha \mid \alpha\overline{T}(\in)\Phi\}. \qquad (\nabla 3)\quad \text{From } \alpha\,\overline{T}(\leq)\,\beta \text{ infer} \vdash_\nabla \nabla\alpha \leq \nabla\beta$$

$$\text{where}\ \ \alpha \in T_\omega\mathcal{M},\ A \in \mathcal{P}_\omega T_\omega\mathcal{M},\ \Phi \in T_\omega\mathcal{P}_\omega\mathcal{M}$$

Space forces us to refer to [27] for details. Intuitively, $(\nabla 1)$ eliminates conjunctions, $(\nabla 2)$ distributes disjunctions over the ∇ and $(\nabla 3)$ is a congruence rule. But note that these intuitions are not expressed in standard logical concepts, e.g. $(\nabla 1)$ involves applying T to the map $\bigwedge : \mathcal{P}_\omega\mathcal{M} \to \mathcal{M}$ and the congruence rule uses relation lifting instead of simply substituting terms into operation symbols. This can be avoided by moving from $\mathcal{M}_T$ to $\mathcal{K}_T$, as we show in the following. We clarify some notation. The set $SRD(A)$ is the set of slim redistributions of A; we defined and discuss $SRD(A)$ in the next section.

5.4.1 Slim redistributions

In this section we will define and illustrate the concept of slim redistributions in [27]. Our explanation is based on that given in [9]. The reader might want to skip this section and refer to it when needed.

Slim redistributions will play a key role in Lemma 5.43 and consequently in the proof of our soundness and completeness result (Theorem 5.42).

Definition 5.33 A set $\Phi \in T\mathcal{Q}(X)$ is a redistribution of a set $A \in \mathcal{Q}T_\omega(X)$ if $A \subseteq \boldsymbol{\nabla}\Phi$. More explicitly, $(\forall\alpha \in A)(\alpha\,\overline{T}(\in_X)\,\Phi)$. A redistribution Φ is *slim* if $\Phi \in T_\omega\mathcal{P}_\omega(\bigcup_{\alpha\in A} Base(\alpha))$. The set of slim redistributions of A is denoted by $SRD(A)$.

Remark 5.34 In [27] slim redistributions are only defined for finite subsets, this restriction is not essential for the completeness proof. The important property is that the set of slim redistributions of a finite set is finite, which follows from our definition.

Before explaining the intuition of slim redistribution we present an illustrative example.

Example 5.35 In the case of $\mathcal{P}$, a set $\Phi \in \mathcal{P}\mathcal{Q}X$ is a slim redistribution of a set $A \in \mathcal{Q}\mathcal{P}X$ iff $\bigcup A = \bigcup\Phi$ and for every $\varphi \in \Phi$ and $\alpha \in A$ we have $\alpha \cap \varphi \neq \emptyset$.

The previous example illustrates that slim redistributions are in some sense minimal covers. More explicitly, each $\alpha \in A$ is cover by Φ, i.e. $\alpha \subseteq \bigcup\Phi$. Moreover, since $\alpha \cap \varphi \neq \emptyset$ for each $\varphi \in \Phi$, we can say that in fact Φ is a "minimal" cover of all the elements of A. The following example, taken from [9], illustrates this further.

Example 5.36 A slim redistribution for a set $A \in \mathcal{Q}\mathcal{P}(\mathcal{M}_\mathcal{P})$ arises semantically as follows. Fix a $\mathcal{P}$-coalgebra (X, ξ) and a state $x_0 \in X$. Define, for any successor x of x_0, the set $\varphi_x := \{\alpha \in \bigcup A \mid x \in [\![\alpha]\!]_{(X,\xi)}\}$. Then let Φ_{x_0} be the set $\{\varphi_x \mid x \in \xi(x_0)\}$. It can now be shown that $x_0 \in [\![\bigwedge\{\nabla\alpha \mid \alpha \in A\}]\!]$ iff $\Phi_{x_0} \in SRD(A)$.

As a final example we describe redistributions for the distribution functor.

Example 5.37 Recall Example 3.14. Fix $\Phi \in \mathcal{DQ}(X)$ and a set $A \in \mathcal{QD}(X)$. Recall that Φ can be thought as a sequence $(\varphi_j, q_j)_{1\leq j\leq m}$ for $\varphi_j \in \mathcal{Q}X, q_j \in [0,1], q_j > 0, m \in \mathbb{N}$. In similar fashion each $a \in A$ can be seen as a sequence $(a_i, p_i)_{1\leq i\leq n}$ for some $a_i \in X, p_i \in [0,1], p_i > 0, n \in \mathbb{N}$. Now we can see that Φ is a redistribution of A if for each $a \in A$ there exists a matrix $(r^a_{ij})_{1\leq i\leq n, 1\leq j\leq m}$, $r_{ij} \in [0,1]$ such that $a_i \notin \varphi_j \;\Rightarrow\; r^a_{ij} = 0$ and $\sum_i r^a_{ij} = q_j$ and $\sum_j r^a_{ij} = p_i$. The redistribution Φ would be slim if each of φ_i is a finite set.

5.4.2 *Towards a completeness proof*

The key concept behind (∇1) is that of slim redistribution. Using Lemma 5.9 we can present (slim) redistributions in terms of presentations.

Definition 5.38 Let $\langle\Sigma, E\rangle$ be a presentation of a functor T. A Σ-*redistribution* of a set $A \subseteq \Sigma X$ is an element $(q, \psi) \in \Sigma_n \times (\mathcal{Q}X)^n$ such that: for each $(p, a) \in A$ there exists $k < \omega$, $r \in \Sigma_k$, $b : k \longrightarrow X$ and $\varphi : k \longrightarrow \mathcal{Q}X$ such that

$$r(b) \approx_T p(a) \,\wedge\, r(\varphi) \approx_T q(\psi) \,\wedge\, (\forall i)(b_i \in \varphi_i). \tag{26}$$

Let $|A| = \{a_i \mid (p,a) \in A\}$. A Σ-redistribution (q,ψ) is *slim* if (1) $n \leq 2^{|A|}$ and (2) $\bigcup_{i\in n} \psi_i \subseteq |A|$. The set of slim redistributions of A is denoted $\Sigma RD(A)$.

Whereas the concept of a Σ-redistribution is a translation of the concept of redistribution in Definition 5.33, the concept of slim Σ-redistribution is not a translation of the concept of slim redistribution as in Definition 5.33. The equivalence of the two notions will be the key step to prove completeness; we will use well based presentations for that.

We now explain the intuition behind slim Σ-redistributions. Recall that given a presentation $\langle\Sigma, E\rangle$ we can think of the elements $p(a) \in \Sigma_n \times X^n$ as algebraic terms, in the algebraic language given by Σ, using variables $\{a_i \,|\, i \in n\}$. With this in mind, $E_X(p,a)$ is then an equivalence class of terms. A Σ-redistribution of a set $A \subseteq \Sigma X$ is a term $q(\psi) \in \Sigma\mathcal{Q}X$, i.e. a term that uses as variables sets of variables in X, that allows us to rewrite each of the terms in A modulo $\approx_T$. Now such a redistribution will be slim if we don't need to use more than $2^{|A|}$ sets to do this rewriting, condition (1), and we don't use variables that were not present in $|A|$, condition (2). In case T preserves finite sets, "slim" makes sure that $\Sigma RD(A)$ is finite if A finite.

Example 5.39 Recall Example 5.7.

(i) Consider the identity presentation of $1 + Id$. The Σ-redistributions of a set $A \subseteq 1+X$ can be described in the following cases: i) If $A = \{*\}$ then then only redistribution is $*$ itself, is in fact slim. ii) If $* \notin A$ then a redistribution is any super set of A; the only slim redistribution is A itself. iii) In any other case, i.e. $\{*, x\} \subseteq A$ for some $x \in X$, the set of redistributions is empty.

(ii) For the List presentation of $\mathcal{P}$ the redistributions of a set $A \subseteq \mathsf{List}(X)$ are all the lists $\Psi = [\psi_1, \ldots, \psi_n]$ of subsets of $\mathcal{Q}(X)$ such that for each $[a_1, \ldots, a_m] \in A$ we have

$\{a_1, \ldots, a_m\} \subseteq \bigcup \psi_i$. The redistribution Φ is slim if $\bigcup \psi_i \subseteq \{a_i \mid a \in A\}$ and $n \leq |\{a_i \mid a \in A\}|$.

(iii) In the case of the canonical presentation of $\mathcal{P}$, the redistributions of a set $A \subseteq \Sigma X$ are the pairs (q, ψ) such that for each $(p, a) \in A$ we have $\{a_i \mid i \in p\} \subseteq \bigcup\{\psi_j \mid j \in q\}$. The redistribution Φ is slim if $\bigcup \psi_i \subseteq \{a_i \mid a \in A\}$ and $n \leq |\{a_i \mid a \in A\}|$.

(iv) In the case of the canonical presentation for $\mathcal{B}_\mathbb{N}$ we have: The redistributions of a set $A \subseteq \Sigma X$ are the pairs $(p, \psi) \in \mathcal{B}_\mathbb{N}(n) \times \mathcal{Q}(X)^n$ such that for each $(p, a) \in A$ there exits a matrix $(r^a_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$, such that $a_i \notin \psi_j \Rightarrow r^a_{ij} = 0$ and $\sum_i r^a_{ij} = q_j$ and $\sum_j r^a_{ij} = p_i$.

Using slim Σ-redistributions we can translate $(\nabla 1)$ as follows

$$(\Sigma 1) \qquad \bigwedge\{\lambda^p(a) \mid (p, a) \in A\} \leq \bigvee\{\lambda^q(\bigwedge \psi) \mid (q, \psi) \in \Sigma RD(A)\}.$$

where $\bigwedge \psi$ is short for $(\bigwedge \psi_1 \ldots \bigwedge \psi_n)$.

Remark 5.40 Axiom $(\Sigma 1)$ simplifies some, but not all aspects of $(\nabla 1)$. In particular, it does not replace the notion of a redistribution in the sense of [27] by something fundamentally simpler: A ΣRD lives in the upper row of Diagram (23), on page 31, and has been defined so that it matches the notion from [27] living in the lower row. One way to understand our axiomatisation in general, and Axiom $(\Sigma 1)$, and Equation (26) in particular, is as an *implementation* of the axiomatisation in [27]. Indeed, given a set A as in $(\nabla 1)$ or $(\Sigma 1)$, to apply the axiom we need a join over a sufficiently large set of redistributions of A. Equation (26) tells us how to compute this set using the equational theory $\approx_T$. For such computational purposes, one would not work with the canonical representation but rather a smaller one as e.g. given by List for the powerset in Example 5.4.

To translate $(\nabla 2)$ we make

Definition 5.41 A *coredistribution* of an element $(q, \psi) \in \Sigma \mathcal{Q} X$ is an element $(p, a) \in \Sigma X$ satisfying (26) and a injective. The set of coredistributions of (p, ψ) is denoted $CRD(p, \psi)$.

Now $(\nabla 2)$ can be written as follows:

$$(\Sigma 2) \qquad \lambda^p(\bigvee \psi) \leq \bigvee\{\lambda^q(a) \mid (q, a) \in CRD(p, \psi)\}.$$

One advantage of our equational axiomatisation is that the rule $(\nabla 3)$ reduces to the standard congruence rule of equational logic. In summary we have:

Theorem 5.42 *Let $\langle \Sigma, E\rangle$ be a well based presentation of T_ω. The derivation system given by the equational logic for Σ and the axioms $\Sigma 1$ and $\Sigma 2$ on top of a complete equational presentation for classical propositional logic is sound and complete for the logic $\mathcal{K}_T$.*

In order to prove this we need two lemmas relating SRD and $\overline{T}(\in_X)$ to ΣRD and CRD, respectively.

Lemma 5.43 *Let $\langle \Sigma, E\rangle$ be a well based presentation and let $A \subseteq \Sigma X$. For any $\Phi \in T\mathcal{Q}X$ the following conditions are equivalent:*

(i) $\Phi \in SRD(E_X[A])$.

(ii) *There exits* $(q, \psi) \in \Sigma RD(A)$ *such that* $E_{QX}(q, \psi) = \Phi$.

In other words, slim redistributions and Σ*-slim redistributions coincide for well based presentations.*

Proof. Since $\langle \Sigma, E \rangle$ is well based, wlog, we can assume that each $(p, a) \in A$ is a basic representative of $E(p, a)$.

From (i) to (ii): Since $\langle \Sigma, E \rangle$ is well based Φ has a basic presentation (q, ψ); notice that $(q, \psi) \in \Sigma QX$. Since Φ is a redistribution of $E_X[A]$ we have $E_X[A] \subseteq \boldsymbol{\nabla}\Phi$; this implies that for each $(p, a) \in A$ we have $E(p, a)\overline{T}(\in_X)E(q, \psi)$. From this, using Lemma 5.9, we conclude that (q, ψ) is a Σ-redistribution of A. Now we show that it is in fact a slim Σ-redistribution.

Since Φ is a slim redistribution of $E_X[A]$ we have

$$\Phi \in T_\omega \mathcal{P}_\omega \left(\bigcup_{(p,a) \in A} Base(E(p, a)) \right).$$

Since (q, ψ) is a representation of Φ, this equation implies

$$\forall i \in dom(\psi) \quad \psi_i \subseteq \bigcup_{(p,a) \in A} Base(E(p, a)).$$

Since (p, a) is basic we have $Base(E(p, a)) = \{a_i \,|\, i \in dom(a)\}$. From this we conclude

$$\bigcup_{i \in dom(\psi)} \psi_i \subseteq \{a_i \,|\, (p, a) \in A\}.$$

It is only left to bound the arity of ψ, i.e. $dom(\psi) \leq 2^{|A|}$, but this follows because (q, ψ) is basic and then ψ is injective.

From (ii) to (i): Lemma 5.9 and the definition of Σ-redistribution imply that for each $(p, a) \in A$ we have $E(p, a)\overline{T}(\in_X)E(q, \psi)$; that is $E(q, \psi) = \Phi$ is a redistribution of $E_X[A]$. Now we show that it is in fact a slim redistribution in the sense of Definition 5.33. Since (q, ψ) is a slim Σ-redistribution we have $\bigcup_{i \in dom(\psi)} \psi_i \subseteq \{a_i \,|\, (p, a) \in A\}$. Since each (p, a) is a basic presentation the right hand of the inclusion can be replaced by $\bigcup_{(p,a) \in A} Base(E(p, a))$. Clearly each ψ_i is finite, because of that we can assume ψ to be a function as follows:

$$\psi : n \longrightarrow \mathcal{P}_\omega \left(\bigcup_{(p,a) \in A} Base(E(p, a)) \right)$$

In other words, Φ is a slim redistribution of $E_X[A]$. $\square$

In the case of coredistributions, using well based presentations, we have the following result which is immediate from Lemma 5.9.

Lemma 5.44 *Let* $\langle \Sigma, E \rangle$ *be a well based presentation and let* $(q, \psi) \in \Sigma QX$ *for an element* $\alpha \in TX$ *the following are equivalent:*

(i) $\alpha \, \overline{T}(\in_X) \, E_Q(q, \psi)$.

(ii) *There exists* $(p, a) \in CRD(p, \psi)$ *such that* $\alpha = E(p, a)$.

Proof. It is straightforward from Proposition 5.9 that we can choose $(p, a) \in CRD(p, \psi)$, i.e. a injective, holds because the presentation is well based. □

Now we can prove Theorem 5.42.

Proof. [Proof Theorem 5.42] Let tr be the translation of Moss liftings into the ∇-logic obtained from the translators $E(p, -)$ (Diagram 24, page 32). We will show that the axioms $(\Sigma 1)$ and $(\Sigma 2)$ are translated into instances of the axioms $(\nabla 1)$ and $(\nabla 2)$; and vice-versa. This will imply our completeness result because the system $(\nabla 1), (\nabla 2), (\nabla 3)$ is complete. In order to see the translation first notice the following: Let $B : \mathcal{Q} \longrightarrow I$ be one of the Boolean connectives $\bigwedge, \bigvee$ Since E is natural the following diagram commutes

$$\begin{array}{ccc} \Sigma X & \xrightarrow{E_X} & TX \\ {\scriptstyle \Sigma(B)}\uparrow & & \uparrow{\scriptstyle T(B)} \\ \Sigma \mathcal{Q} X & \xrightarrow[E_{\mathcal{Q}X}]{} & T\mathcal{Q}X \end{array} \tag{27}$$

In the following, to make our equations simpler we will avoid X in the subindex of E. The left hand of Axiom $(\Sigma 1)$ is translated as follows:

$$tr\left(\bigwedge\{\lambda^p(a) \mid (p, a) \in A\}\right) = \bigwedge\{\nabla E(p, a) \mid (p, a) \in A\} = \bigwedge\{\nabla \alpha \mid \alpha \in E_X[A]\}$$

It is now enough to show that the right hand is

$$tr\left(\bigvee\{\lambda^q(\bigwedge \psi) \mid (q, \psi) \in \Sigma RD(A)\}\right) = \bigvee\{\nabla T(\bigwedge)\Phi \mid \Phi \in SRD(E_X[A])\}$$

This is done as follows

$$\begin{aligned} tr\left(\bigvee \{\lambda^q(\bigwedge \psi) \mid (q, \psi) \in \Sigma RD(A)\}\right) &= \\ &= \bigvee\{tr(\lambda^q(\bigwedge \psi)) \mid (q, \psi) \in \Sigma RD(A)\} && (tr \text{ is a Boolean morphism}) \\ &= \bigvee\{\nabla E_X(q, \bigwedge \psi) \mid (q, \psi) \in \Sigma RD(A)\} && (\text{Definition of } tr) \\ &= \bigvee\{\nabla E_X \Sigma(\bigwedge)(q, \psi) \mid (q, \psi) \in \Sigma RD(A)\} && (\text{Making } \Sigma \text{ explicit}) \\ &= \bigvee\{\nabla T(\bigwedge) E_Q(q, \psi) \mid (q, \psi) \in \Sigma RD(A)\} && (\text{Diagram } 27) \\ &= \bigvee\{\nabla T(\bigwedge) E_Q(q, \psi) \mid E_{\mathcal{Q}}(q, \psi) \in SRD(E_X[A])\} && (\text{Lemma 5.43 (ii) to (i)}) \\ &= \bigvee\{\nabla T(\bigwedge)\Phi \mid \Phi \in SRD(E_X[A])\} && (\text{Lemma 5.43 (i) to (ii)}) \end{aligned}$$

This concludes the case for $(\Sigma 1)$.

The situation for $(\Sigma 2)$ is as follows: The left hand is translated into

$$tr(\lambda^p(\bigvee \psi)) = \nabla E(p, \bigvee \psi) = \nabla E \Sigma(\bigvee)(p, \psi) = \nabla T(\bigvee) E_Q(p, \psi).$$

Now it is enough to show that the right hand is

$$tr\left(\bigvee\{\lambda^q(a) \mid (q,a) \in CRD(p,\psi)\}\right) = \bigvee\{\nabla\alpha \mid \alpha\,\overline{T}(\in_X)\,E_Q(p,\psi)\}$$

This is seen as follows:

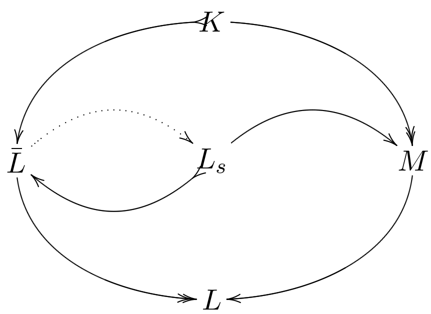
$$\begin{aligned} tr\left(\bigvee\{\lambda^q(a) \mid (q,a) \in CRD(p.\psi)\}\right) &= \\ &= \bigvee\{tr(\lambda^q(a)) \mid (q,a) \in CRD(p,\psi)\} && (tr \text{ is a Boolean morphism}) \\ &= \bigvee\{\nabla E(q,a) \mid (q,a) \in CRD(p,\psi)\} && (\text{Definition of } tr) \\ &= \bigvee\{\nabla\alpha \mid \alpha\,\overline{T}(\in_X)\,E_Q(p,\psi)\} && (\text{Lemma 5.44}) \end{aligned}$$

Now since the system $(\nabla 1), (\nabla 2), (\nabla 3)$ is complete so is the logic $\mathcal{K}_T$.

□

6 Conclusion

In this paper we depicted a general relation between Moss's coalgebraic logic and the logic of all predicate liftings. The following picture summarises the transaltions discussed.



K is the logic given by Moss liftings, $\bar{L}$ is the logic of all finitary predicate liftings, L_s is the logic of all singleton liftings; L is obtained by quotienting $\bar{L}$ with a complete axiomatisation; M is Moss's coalgebraic logic.

A solid arrow means that the translation works for all T for which Moss's logic is defined. A dotted arrow means that T has to preserve finite sets (ie maps finite sets to finite sets). The translations $K \rightarrowtail \bar{L}$, $K \twoheadrightarrow M$, $\bar{L} \twoheadrightarrow L$ are immediate from the definitions, the translations $\bar{L} \longrightarrow M$ and $M \longrightarrow L$ are Theorems 4.24 and 4.27, respectively. Double arrowheads indicate that the translation is onto and can be reversed, albeit not necessarily by a natural transformation as choices of representatives are involved. Arrows with tails indicate that the translation is one-to-one.

In case T preserves finite sets, the diagram above suggests that L is *the* logic for T-coalgebras. If T does not preserve finite sets, the situation is not so clear. For example, in the case of the distribution functor, L has uncountably many formulas and is more expressive than the standard logic for the distribution functor (which happens to be equi-expressive to M in this example). On the other hand, in the case of the multiset-functor

M is less expressive than the standard logic as the modality "in more than n successors" cannot be expressed. So the quest for the coalgebraic logic in general is still open.

In Example 4.25, we showed that not all predicate liftings are translatable if T does not preserve finite sets, even if the underlying logic is classical logic. However, it would be interesting to give a general characterization of the predicate liftings that can be translated into Moss's logic and of those that can not be translated. Our conjecture is: A predicate lifting is translatable into Moss's logic iff it can be presented as a finite disjunction of Moss liftings.

Another issue that we have not studied is related to the computable properties of translators and logical translators. We don't know what is the actual computational cost of a translation using logical translators. Related to this is the computational nature of the set of slim redistributions. It would be interesting to investigate for which T the set of slim redistributions is enumerable or recursive.

Recall that the translations here depend on the category BA. This suggests that it would be worth studying *coalgebraic non-classical logic*. For example, we expect that some of the results of this paper transfer to coalgebraic logics over DL if we also replace the category Set by the category Poset.

References

[1] Samson Abramsky, *Domain theory in logical form*, Ann. Pure Appl. Logic **51** (1991).

[2] Peter Aczel, *Non-well-founded sets*, CSLI, Stanford, 1988.

[3] Peter Aczel and Nax Paul Mendler, *A final coalgebra theorem*, Category Theory and Computer Science, Lecture Notes In Computer Science, vol. 389, 1989.

[4] Jiří Adámek and Jiří Rosický, *Locally presentable and accessible categories*, London Mathematical Society Lecture Notes Series, vol. 189, Cambridge University Press, 1994.

[5] Jiří Adámek, JJiří Rosický, and E. M. Vitale, *Algebraic theories: a categorical introduction to general algebra*, Cambridge Tracts in Mathematics, vol. 184, Cambridge University Press, 2011.

[6] Jiří Adámek and Vera Trnková, *Automata and algebras in categories*, Kluwer, 1990.

[7] Michael Barr, *Relational algebras*, Reports of the Midwest Category Theory Seminar IV, Lecture Notes in Mathematics, vol. 137, Springer Berlin / Heidelberg, 1970.

[8] Falk Bartels, Ana Sokolova, and Erik de Vink, *A hierarchy of probabilistic system types*, Theoretical Computer Science **327** (2004).

[9] Marta Bílková, Alessandra Palmigiano, and Yde Venema, *Proof systems for the coalgebraic cover modality*, Advances in Modal Logic (Carlos Areces and Rob Goldblatt, eds.), King's College Publications, vol. 7, 2008.

[10] Patrick Blackburn, Maarteen de Rijke, and Yde Venema, *Modal logic*, Cambridge Tracts in Theoretical Computer Science, vol. 53, Cambridge University Press, 2002.

[11] Marcello Bonsangue and Alexander Kurz, *Presenting functors by operations and equations*, Lecture Notes in Computer Science, vol. 3921, Springer Berlin / Heidelberg.

[12] Giovanna D'Agostino and Giacomo Lenzi, *On modal μ-calculus with explicit interpolants*, Journal of Applied Logic **338** (2006).

[13] Brian Davey and Hilary Priestley, *Introduction to lattices and order*, Cambridge University Press, 1990.

[14] Erik de Vink and Jan Rutten, *Bisimulation for probabilistic transition systems: a coalgebraic approach*, Proceedings of ICALP'97, Lecture Notes in Computer Science, vol. 1256, 1997.

[15] Ernst-Erich Doberkat, *Stochastic relations*, Chapman & Hall/CRC, 2007.

[16] Micheal Dunn, *Positive modal logic*, Studia Logica **55** (1995), no. 2.

[17] Gianluigi Ferrari, Ugo Montanari, and Marco Pistore, *Minimizing transition systems for name passing calculi: A co-algebraic formulation*, FoSSaCS'02.

[18] Marcelo P. Fiore and Daniele Turi, *Semantics of name and value passing*, Proc. 16th LICS, IEEE Computer Society Press, 2001.

[19] Helle Hansen and Clemens Kupke, *A coalgebraic perspective on monotone modal logic*, Electronic Notes in Theoretical Computer Science, vol. 106, 2004.

[20] Aviad Heifetz and Philippe Mongin, *Probabilistic logic for type spaces*, Games and Economic Behavior **35** (2001).

[21] Bart Jacobs and Ana Sokolova, *Exemplaric expressivity of modal logics*, Journal Logic Computation, To appear.

[22] David Janin and Igor Walukiewicz, *Automata for the modal mu-calculus and related results*, MFCS'95 Proceedings, Springer-Verlag, 1995.

[23] Peter Johnstone, *Stone spaces*, Cambridge University Press, 1982.

[24] ______, *Vietoris locales and localic semilattices*, Continuous Lattices and their Applications, Lecture Notes in Pure and Applied Mathematics, vol. 101, Marcel Dekker, 1985.

[25] Bartek Klin, *The least fibred lifting and the expressivity of coalgebraic modal logic*, Lecture Notes in Computer Science, vol. 3629, Springer Berlin / Heidelberg, 2005.

[26] Clemens Kupke, Alexander Kurz, and Dirk Pattinson, *Algebraic semantics for coalgebraic logics*, Electronic Notes Theoretical Computer Science, vol. 106, 2004.

[27] Clemens Kupke, Alexander Kurz, and Yde Venema, *Completeness of the finitary Moss logic*, Advances in Modal Logic (Carlos Areces and Rob Goldblatt, eds.), King's College Publications, vol. 7, 2008.

[28] Clemens Kupke and Yde Venema, *Coalgebraic automata theory: basic results*, Logical Methods in Computer Science **4** (2008).

[29] Alexander Kurz, *Specifying coalgebras with modal logic*, Electronic Notes in Theoretical Computer Science, vol. 11, Elsevier, 1998.

[30] ______, *Logics for Coalgebras and Applications to Computer Science*, Ph.D. thesis, LMU, 2000.

[31] Alexander Kurz and Jiří Rosický, *The Goldblatt-Thomason-theorem for coalgebras*, CALCO'07, Lecture Notes in Computer Science, vol. 4624, Springer, 2007.

[32] ______, *Strongly complete logics for coalgebras*, (July 2006), Submitted.

[33] Alexander Kurz and Yde Venema, *Coalgebraic Lindström theorems*, Advances in Modal Logic 2010.

[34] Raul Andres Leal, *Predicate liftings versus nabla modalities*, Electronic Notes in Theoretical Computer Science **203** (2008), no. 5.

[35] Saunders Mac Lane, *Category theory for the working mathematician*, Springer, 1971.

[36] Lawrence Moss, *Coalgebraic logic*, Ann. Pure Appl. Logic **96** (1999).

[37] Dirk Pattinson, *Coalgebraic modal logic: Soundness, completeness and decidability of local consequence*, Theoretical Computer Science **309** (2003).

[38] Martin Rößiger, *Coalgebras and modal logic*, Electronic Notes in Theoretical Computer Science, vol. 33, 2000.

[39] Jan Rutten, *Automata and coinduction - an exercise in coalgebra*, CONCUR'98 (D. Sangiorigi and R. de Simone, eds.), Lecture Notes in Computer Science, vol. 1466, 1998.

[40] ______, *Universal coalgebra: A theory of systems*, Theoretical Computer Science **249** (2000).

[41] Luigi Santocanale and Yde Venema, *Completeness for flat modal fixpoint logics*, LPAR (Nachum Dershowitz and Andrei Voronkov, eds.), LNAI Springer, vol. 4790, 2007.

[42] Lutz Schröder, *Expressivity of Coalgebraic Modal Logic: The Limits and Beyond*, Theoretical Computer Science **390** (2008).

[43] Steven J. Vickers, *Topology via logic*, Cambridge Tracts in Theoretical Computer Science, vol. 5, Cambridge University Press, 1989.

[44] Albert Visser and Giovanna D'Agostino, *Finality regained: A coalgebraic study of Scott-sets and multisets*, Arch. Math. Logic **41** (2002).