Predicate Logic for Functors and Monads

Bart Jacobs

Institute for Computing and Information Sciences, Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands.
Email: bart@cs.ru.nl URL: http://www.cs.ru.nl/~bart

May 10, 2010

Abstract. This paper starts from the elementary observation that what is usually called a predicate lifting in coalgebraic modal logic is in fact an endomap of indexed categories. This leads to a systematic review of basic results in predicate logic for functors and monads, involving induction and coinduction principles for functors and compositional modal operators for monads.

1 Introduction

Functors form the starting point in the categorical study of datatypes (usually as algebras) and state-based systems (as coalgebras). Sometimes the functors involved are actually monads, capturing some form of computational effect. Logic is needed to reason about datatypes, mainly via induction and equational logic, and about state-based systems, mainly via coinduction and (coalgebraic) modal logic. Hence the logical study of datatypes and state-based systems requires a systematic logical view on functors and monads.

Part of this view is provided in [4], where induction and coinduction principles for datatypes are described via a lifting of functors from categories of types to categories of predicates and relations (using the language of fibrations), see also [9]. The lifting used in [4] is defined by induction on the structure of polynomial functors.

Subsequently, in coalgebraic modal logic such liftings have been studied more systematically as certain natural transformations (see *e.g.* [11] or [2]), which are called predicate (or relation) liftings. They typically have components of the form $\sigma_X : \wp(X) \to \wp(F(X))$, and give a way of "lifting" predicates on X to predicates on $F(X)$. As said, in simple cases this can be done by induction on the structure of F. The basic observation in this paper is that such a pair (F, σ) forms an endomap in a category of indexed categories. Once this perspective is accepted, many (partially) known constructions in the logic of datatypes and state-based computation become instances of general constructions on indexed categories. For instance, taking the category of (co)algebras of such an endomap (F, σ) in the (2-categorical) setting of indexed categories yields a new indexed category of predicates on the (co)algebras of F.

Within this indexed perspective the notion of "monad with predicate lifting" arises naturally. It is . . . a monad in the 2-category of indexed categories. Taking for instance the *indexed* Kleisli category of such a monad with predicate lifting (T, σ) yields an indexed category of predicates to reason about the Kleisli category of T. In particular, this indexed Kleili category involves modal operators (as substitution functors) that are suitably compositional: $\Box_{g\circ f}(Q) = \Box_f \Box_g(Q)$. This says: "$Q$ holds after g after f" is the same as "(Q holds after g) holds after f".

Indexed categories—or, more or less equivalently, fibrations—give a systematic categorical account of predicate logics, in which predicates are indexed by (or fibred over) types. Here we only use the basics, and will not go deeply into technicalities: the focus in this paper lies on developing logics for functors and monads, and not on the 2-categorial/indexed niceties. The standard reference is [10], but see also [7], for connections with logic and type theory. The general framework of indexed categories that we use here is not needed to handle the main examples. They are set-theoretic and can also be described in a more concrete manner. However, the general framework is useful precisely because it yields abstract transparancy: it clarifies which aspects are essential.

This paper has a simple structure. Section 2 recalls the basics of indexed categories as categorical models of predicate logic, and introduces the 2-categorical structure. Section 3 looks at functors with liftings, and basically describes the essentials of [4] but parametrised by a predicate lifting. Section 4 applies the indexed framework to monads and develops their logic with modal operators for Kleisli categories. It is only in Example 5 that the general approach of this paper becomes more concrete and is applied to specific modal logics, including dynamic logic.

2 Basics of indexed categories

For a set X, the predicates on X are given by the powerset of subsets of X. Its Boolean algebra structure captures the logical operations on these predicates. Each function $f: X \to Y$ yields a way of turning predicates on Y into predicates on X, namely via pullback or substitution: for $Q \subseteq Y$ we have $f^{-1}(Q) \subseteq X$ given as $f^{-1}(Q) = \{x \in X \mid f(x) \in Q\}$. This logic can thus be organised as a functor of the form $\wp: \mathbf{Sets}^{\mathrm{op}} \to \mathbf{BA}$, where **BA** is the category of Boolean algebras. Notice that we write $\wp(X)$, instead of $\mathcal{P}(X)$, when we wish to consider subsets as predicates on X. Technically, $\wp$ is a contravariant functor $\mathbf{Sets}^{\mathrm{op}} \to \mathbf{BA}$, whereas $\mathcal{P}$ is used as a covariant functor $\mathbf{Sets} \to \mathbf{Sets}$, and a monad.

More generally, a predicate logic for a category **C** is given by a functor of the form $\Phi: \mathbf{C}^{\mathrm{op}} \to \mathbf{PoSets}$. For $X \in \mathbf{C}$ in the base category one calls $\Phi(X)$ the fibre (category) over X. It contains the predicates on X, with order $P \leq Q$ given by implication. The fibres $\Phi(X)$ may have more algebraic structure, like in the Boolean algebra example on **Sets**. One can drop the requirement that the fibres $\Phi(X)$ are posets and just require that they are categories. A map $P \to Q$ between two predicates is then understood as a proof of Q from P. This follows the "propositions-as-types" and "proofs-as-terms" paradigm. Each map $f: X \to Y$ in **C** yields a "substitution" functor $\Phi(f): \Phi(Y) \to \Phi(X)$ between the fibres—in reversed direction. Often, when the context is clear, this substitution functor $\Phi(f)$ is simply written as f^*.

In what follows there is no technical reason to restrict fibres to posets. Hence we shall be working with functor of the form $\mathbf{C}^{\mathrm{op}} \to \mathbf{Cat}$. They are called *indexed categories* [10]. Nevertheless we shall think of them as providing a predicate logic on **C**. For the record we list some examples.

Example 1. We already mentioned the indexed category of subsets $\wp\colon \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Cat}$. More generally, for a category $\mathbf{C}$ with pullbacks (of monos) there is the indexed category of subobjects $\mathit{Sub}\colon \mathbf{C}^{\mathrm{op}} \to \mathbf{Cat}$ that sends an object $X \in \mathbf{C}$ to its poset (category) $\mathit{Sub}(X)$ of subobjects of X. Substitution is given by pullback. This subobject indexed category captures the logic of subobjects of $\mathbf{C}$, for instance when $\mathbf{C}$ is a topos. For some special cases, like for metric spaces or for Hilbert spaces, one restricts to the *closed* subobjects, so that one gets an indexed category of the form $\mathit{ClSub}\colon \mathbf{Hilb}^{\mathrm{op}} \to \mathbf{Cat}$.

For an arbitrary category $\mathbf{C}$ and set I, let $\mathbf{C}^I$ be the category of I-indexed families in $\mathbf{C}$. Its objects are families $(X_i)_{i\in I}$ of objects $X_i \in \mathbf{C}$, and its morphisms $(X_i)_{i\in I} \to (Y_i)_{i\in I}$ are families $(f_i)_{i\in I}$ of maps $f_i\colon X_i \to Y_i$ in $\mathbf{C}$. The mapping $I \mapsto \mathbf{C}^I$ then yields an indexed category $\mathbf{Sets}^{\mathrm{op}} \to \mathbf{Cat}$.

We recall (*e.g.* from [7]) the so-called Grothendieck construction $\int \Phi$ for an indexed category $\Phi\colon \mathbf{A}^{\mathrm{op}} \to \mathbf{Cat}$; it is also known as the category of elements, or as the category of predicates of Φ. It yields a split fibration $\int \Phi \to \mathbf{A}$, but we do not rely in an essential way on the theory of fibrations. The category $\int \Phi$ has pairs (X, P) as objects, where $X \in \mathbf{A}$ and $P \in \Phi(X)$. A morphism $(X, P) \to (Y, Q)$ in $\int \Phi$ consists of a pair of maps $f\colon X \to Y$ in $\mathbf{A}$ with a "proof" $p\colon X \to f^*(Q)$ in $\Phi(X)$. This yields a category with composition $(g, q) \circ (f, p) = (g \circ f, f^*(q) \circ p)$. This category $\int \Phi$ can be understood as the "total" category of all predicates on objects in $\mathbf{A}$. It forms a fibration over $\mathbf{A}$ with (split) cartesian maps of the form (f, id).

We also recall that an indexed category $\Phi\colon \mathbf{A}^{\mathrm{op}} \to \mathbf{Cat}$ has indexed finite (co)products if each fibre $\Phi(X)$ has finite (co)products and substitution functors preserve them. When the fibres $\Phi(X)$ are posets, these products and coproducts form conjunctions and disjunctions of predicates. If Φ has an indexed final object—each fibre $\Phi(X)$ has a final object $1(X) \in \Phi(X)$ and these final object are preserved by reindexing—then one can gets a "truth" functor $1\colon \mathbf{A} \to \int \Phi$. Comprehension for (the logic of) Φ is then given by a right adjoint $\{-\}\colon \int \Phi \to \mathbf{A}$ to 1.

Universal and existential quantification are handled by adjoints to (special) substitution functors. In most general form it involves left and right adjoints to substitution, as in:

$$\textstyle\coprod_f \dashv f^* \dashv \prod_f \tag{1}$$

subject to the so-called Beck-Chevalley condition that regulates proper interaction between quantification and substitution (see [7] for details). From the pure logical perspective adjoints (1) are too much: only adjoints to weakening functors π_1^* and to contraction functors δ^* are enough, where the projections $\pi_1\colon X \times Y \to X$ and diagonals $\delta\colon X \to X \times X$ stem from Cartesian structure in the base category $\mathbf{A}$. The left adjoint to contraction δ^* provides the logic with equality: for $X \in \mathbf{A}$ one writes $\mathrm{Eq}(X) = \coprod_\delta(1X) \in \Phi(X \times X)$ for the equality relation on X.

Example 2. We sketch some of the logical structure introduced above for the indexed category $\wp\colon \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Cat}$ of subsets on sets. Obviously, each fibre $\wp(X)$ has finite products and coproducts, given by meets and joins. These are preserved by substitution functors, since $f^{-1}(U \cap V) = f^{-1}(U) \cap f^{-1}(V)$ *etc.* For each function $f\colon X \to Y$

one has adjoints as in (1) given on a predicate $P \subseteq X$ as:

$$\begin{aligned} \textstyle\coprod_f(P) &= \{y \in Y \mid \exists x \in X.\, f(x) = y \wedge x \in P\} \\ \textstyle\prod_f(P) &= \{y \in Y \mid \forall x \in X.\, f(x) = y \Rightarrow x \in P\}. \end{aligned}$$

Then indeed:

$$\textstyle\coprod_f(P) \subseteq Q \Longleftrightarrow P \subseteq f^{-1}(Q) \quad \text{and} \quad f^{-1}(Q) \subseteq P \Longleftrightarrow Q \subseteq \prod_f(P).$$

The category $\int \wp$ has pairs (X, P) were $P \subseteq X$ as objects. A morphism $f\colon (P \subseteq X) \to (Q \subseteq Y)$ is a function $f\colon X \to Y$ satisfying $f(x) \in Q$ for all $x \in P$. The "truth" functor $1\colon \mathbf{Sets} \to \int \wp$ maps a set X to the "truth" predicate $1(X) = (X \subseteq X) \in \int \wp$. It has a right adjoint $\{-\}\colon \int \wp \to \mathbf{Sets}$ that sends a predicate $(P \subseteq X)$ to its extent P, considered as a set itself.

The same constructions are present in a subobject indexed category *Sub*: $\mathbf{C}^{\mathrm{op}} \to \mathbf{Cat}$ when $\mathbf{C}$ is a topos.

2.1 Indexed categories form 2-categories

In the remainder of this section we briefly consider morphisms between indexed categories (forming 1-cells), and also maps between these morphisms (forming 2-cells). A morphism

$$\Big(\mathbf{B}^{\mathrm{op}} \xrightarrow{\Phi} \mathbf{Cat}\Big) \xrightarrow{\quad (F,\sigma) \quad} \Big(\mathbf{C}^{\mathrm{op}} \xrightarrow{\Psi} \mathbf{Cat}\Big) \tag{2}$$

between two indexed consists of a functor $F\colon \mathbf{B} \to \mathbf{C}$ between the base categories together with a natural transformation $\sigma\colon \Phi \Rightarrow \Psi F$. One can compose such maps as $(G, \tau) \circ (F, \sigma) = (G \circ F, \tau F \circ \sigma)$.

These maps (2) between indexed categories can themselves be seen as objects (or 1-cells), between which morphisms (or 2-cells) may be defined. They are sometimes called *modifications*. Such a 2-cell consists of a pair (α, β) in:

$$\Big(\mathbf{B}^{\mathrm{op}} \xrightarrow{\Phi} \mathbf{Cat}\Big) \underset{(G,\tau)}{\overset{(F,\sigma)}{\underset{\longrightarrow}{\overset{\longrightarrow}{\Downarrow (\alpha, \beta)}}}} \Big(\mathbf{C}^{\mathrm{op}} \xrightarrow{\Psi} \mathbf{Cat}\Big) \tag{3}$$

where α is a natural transformation $\alpha\colon F \Rightarrow G$ and $\beta = (\beta_X)_{X \in \mathbf{C}}$ is a collection of natural transformations $\beta_X\colon \sigma_X \Rightarrow (\Psi(\alpha_X) \circ \tau_X)$ in:

$$\begin{array}{ccc} \Phi(X) & \xrightarrow{\quad \sigma_X \quad} & \Psi(FX) \\ & \Downarrow \beta_X & \\ {\scriptstyle \tau_X} \searrow & \Psi(GX) & \nearrow {\scriptstyle \Psi(\alpha_X)} \end{array}$$

commuting with substitution: for $f\colon X \to Y$,

$$
\begin{array}{ccc}
\sigma_X \Phi(f) & \xrightarrow{\beta_X \Phi(f)} & \Psi(\alpha_X)\tau_X \Phi(f) \\
\| & & \| \text{ (naturality of } \tau) \\
\text{(naturality of } \sigma) \; \| & & \Psi(\alpha_X)\Psi(Gf)\tau_Y \\
\| & & \| \text{ (naturality of } \alpha \text{, under } \Psi) \\
\Psi(Ff)\sigma_Y & \xrightarrow{\Psi(Ff)\beta_Y} & \Psi(Ff)\Psi(\alpha_Y)\tau_Y
\end{array}
$$

For two such modifications:

$$(F, \sigma) \xrightarrow{(\alpha,\beta)} (G, \tau) \xrightarrow{(\gamma,\delta)} (H, \rho)$$

their composition is defined as $(\gamma \circ \alpha, \delta \bullet \beta)$, where $(\delta \bullet \beta)_X = \Psi(\alpha_X)\delta_X \circ \beta_X \colon \sigma_X \Rightarrow \Psi(\alpha_X)\tau_X \Rightarrow \Psi(\alpha_X)\Psi(\gamma_X)\rho_X = \Psi((\gamma \circ \alpha)_X)\rho_X$. It is not hard to see that this commutes again with substitution. In this way we obtain a 2-category **IndCat** of indexed categories.

For a general account of the theory of indexed categories, see [10]. It contains a more general notion than we have sketched above, involving a "pseudo-functor" instead of a functor $\mathbf{C}^{\mathrm{op}} \to \mathbf{Cat}$. This additional generality is not needed in the current applications and so it will not be used. An alternative description in terms of fibrations, in combination with logic and type theory, may be found in [7].

3 Predicate liftings for functors

The (categorical) theory of datatypes and state-based systems takes the notion of endofunctor $\mathbf{A} \to \mathbf{A}$ on a category as starting point. Here we wish to extend this approach in a systematic manner to include logic. Hence it is natural to replace 'category' by 'indexed category' and 'endomap on a category' by 'endomap on an indexed category'. Thus, the basic notion of this paper consists of an indexed category with an endomap. This may be either just an ordinary endomap (this section), or a monad (in a 2-categorical sense, see the next section).

Definition 1. *Let $\Phi\colon \mathbf{A}^{op} \to \mathbf{Cat}$ be an indexed category. A* functor with predicate lifting *for Φ is an endomap of indexed categories $\Phi \to \Phi$. It is given by an endofunctor $F\colon \mathbf{A} \to \mathbf{A}$ together with a natural transformation $\sigma\colon \Phi \Rightarrow \Phi F$; the latter is often called the predicate lifting.*

The predicate lifting σ will be called truth preserving *if each functor $\sigma_X\colon \Phi(X) \to \Phi(FX)$ preserves final objects, if any.*

For instance, the endofunctor $F = (A \times -)$ on **Sets** becomes a functor with predicate lifting for the subset logic $\wp\colon \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Cat}$ via the natural transformation:

$$\wp(X) \longrightarrow \wp(A \times X) \quad \text{given by} \quad Q \longmapsto \{(a, x) \in A \times X \mid x \in Q\}.$$

Similarly, the obvious predicate lifting for the functor $F = 1 + (A \times -)$ is:

$$\wp(X) \longrightarrow \wp(1 + A \times X) \quad \text{given by} \quad Q \longmapsto \{*\} \cup \{(a, x) \in A \times X \mid x \in Q\},$$

where $*$ is the (sole) element of the singleton set 1. Both these predicate liftings are obviously truth preserving.

We shall later see, in Example 4.(1), that there may be multiple liftings σ for the same functor F.

The first result about functors with predicate liftings is that they can be lifted to categories of predicates. We shall formulate it for endomaps $\Phi \to \Phi$ like in the above definition, but it can be extended to maps $\Phi \to \Psi$ between different indexed categories.

Lemma 1. *Let $\Phi: \boldsymbol{A}^{op} \to \boldsymbol{Cat}$ be an indexed category. A map of indexed categories $(F, \sigma): \Phi \to \Phi$ gives rise to a lifting of the functor $F: \boldsymbol{A} \to \boldsymbol{A}$ to a (split) fibred functor $\overline{F}$, in a commuting diagram:*

$$\begin{array}{ccc} \int \Phi & \xrightarrow{\overline{F}} & \int \Phi \\ \downarrow & & \downarrow \\ \boldsymbol{A} & \xrightarrow{F} & \boldsymbol{A} \end{array} \tag{4}$$

via:

$$\overline{F}(X, P) = (F(X), \sigma_X(P)) \qquad \textit{and} \qquad \overline{F}(f, p) = (F(f), \sigma(p)).$$

The latter is well-defined since for $p: P \to f^*(Q)$ in $\Phi(X)$ we get $\sigma(p): \sigma(P) \to \sigma(f^*(Q)) = F(f)^*(\sigma(Q))$ in $\Psi(FX)$ by naturality of σ.

If the indexed category Φ has indexed terminal objects, and $\sigma: \Phi \to \Psi F$ preserves, then the functor $\overline{F}: \int \Phi \to \int \Psi$ commutes with truth functors. $\square$

Functoriality of $\overline{F}$ is an easy exercise. Notice that the dependence on σ in the notation $\overline{F}$ for lifting is left implicit.

Given this lifting $\overline{F}$ one can consider algebras or coalgebras of the functor F, but also of the lifted functor $\overline{F}$. Algebras of F may be understood as datatypes, and algebras of $\overline{F}$ as suitably structured predicates on these datatypes: commutation of the diagram (4) yields obvious functors:

$$\begin{array}{ccc} \mathit{Alg}(\overline{F}) & \qquad\qquad & \mathit{CoAlg}(\overline{F}) \\ \downarrow & & \downarrow \\ \mathit{Alg}(F) & & \mathit{CoAlg}(F) \end{array}$$

In the algebraic case a truth functor $1: \mathbf{A} \to \int \Phi$ yields a functor $\mathit{Alg}(F) \to \mathit{Alg}(\overline{F})$, also written as 1, by:

$$\Big(FX \xrightarrow{a} X\Big) \longmapsto \Big(\overline{F}(1X) \cong 1(FX) \xrightarrow{1(a)} 1(X)\Big)$$

where we assume that the predicate lifting is truth preserving. Following [4] we say that the logic Φ *admits induction* for $(F, \sigma): \Phi \to \Phi$ if this functor $1: \mathit{Alg}(F) \to \mathit{Alg}(\overline{F})$ preserves initial objects. One of the main results in [4] is that this is the case if Φ has comprehension $\{-\}$.

Example 3. Consider the functor $F = 1 + A \times (-)$ on **Sets**. As already mentioned, it comes with an obvious predicate lifting $\sigma_X : \wp(X) \to \wp(FX)$ given by $\sigma_X(Q) = \{*\} \cup \{(a, x) \mid Q(x)\}$, which is truth preserving. The initial algebra of F is the set $A^\star$ of finite lists of elements of A, with structure map:

$$F(A^\star) = 1 + A \times A^\star \xrightarrow[\cong]{[\mathsf{nil}, \mathsf{cons}]} A^\star$$

The lifting $\overline{F}: \int \wp \to \int \wp$ sends a predicate $(Q \subseteq X)$ to the predicate $\sigma(Q) \subseteq F(X)$. In particular, it sends the truth predicate $(A^* \subseteq A^*)$ to the truth predicate $(F(A^\star) \subseteq F(A^*))$, which is the initial $\overline{F}$-algebra in $\int \wp$. This means the following. Suppose we have a predicate $(Q \subseteq X)$ and an $\overline{F}$-algebra on $(Q \subseteq X)$, consisting of an F-algebra $[f, g]: 1 + A \times X \to X$ on X that satisfies $f(*) \in Q$ and $x \in Q \Rightarrow g(a, x) \in Q$, for each $a \in A$. By initiality of $A^\star$ there is the unique map of algebras $h: A^\star \to X$ given by $h(\mathsf{nil}) = f(*)$ and $h(\mathsf{cons}(a, \alpha)) = g(a, h(\alpha))$. The initiality claim about $1(A^\star) = (A^\star \subseteq A^*)$ then says that h is a morphism $1(A^\star) \to (Q \subseteq X)$, *i.e.* that $h(\alpha) \in Q$ for all $\alpha \in A^\star$.

In the coalgebraic case one does not reason with predicates but with (binary) relations. For an indexed category $\Phi: \mathbf{A}^{\mathrm{op}} \to \mathbf{Cat}$ we shall use the *ad hoc* notation Φ^2 for the indexed category $\Phi^2: \mathbf{A}^{\mathrm{op}} \to \mathbf{Cat}$ of relations in Φ given by:

$$\Phi^2(X) = \Phi(X \times X) \quad \text{and} \quad \Phi^2(f) = \Phi(f \times f).$$

The fibre $\Phi^2(X)$ thus contains the binary relations on X in Φ.

Definition 2. *A* functor with relation lifting *for an indexed category* $\Phi: \mathbf{A}^{op} \to \mathbf{Cat}$ *is an endomap of indexed categories* $(F, \sigma): \Phi^2 \to \Phi^2$. *It is given by an endofunctor* $F: \mathbf{A} \to \mathbf{A}$ *together with a relation lifting* $\sigma: \Phi^2 \Rightarrow \Phi^2 F$.

The relation lifting σ *will be called* equality preserving *if each functor* $\sigma_X: \Phi(X \times X) \to \Phi(FX \times FX)$ *preserves the equality relation.*

Such a functor (F, σ) with relation lifting gives rise to a lifted functor $\overline{F}: \int \Phi^2 \to \int \Phi^2$ like in Lemma 1. If σ preserves equality, then $\overline{F}(\mathrm{Eq}(X)) \cong \mathrm{Eq}(F(X))$, so that equality forms a functor $\mathrm{Eq}: \mathit{CoAlg}(F) \to \mathit{CoAlg}(\overline{F})$. One then says, like in [4], that $(F, \sigma): \Phi^2 \Rightarrow \Phi^2$ *admits coinduction* if this functor $\mathrm{Eq}: \mathit{CoAlg}(F) \to \mathit{CoAlg}(\overline{F})$ preserves final objects. This happens in the presence of quotients, see [4] for more information.

Remark 1. So-called polynomial functors are built up inductively from the identity, constants, products and coproducts. For such functors there is a canonical way to define a predicate lifting, by induction on the structure. It can be defined quite generally, for indexed categories with some basic structure (especially indexed (co)products), see [4].

Even more generally, for non-necessarily polynomial functors, there is a canonical lifting in case the indexed category has comprehension, see [4, Remark 2.13]. This lifting is used in [9] to generically formulate induction principles for many datatypes.

For the standard subset logic indexed category $\wp\colon \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Cat}$ it looks as follows. Given an arbitrary endofunctor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ define for $Q \subseteq X$,

$$\sigma(Q) = \{u \in F(X) \mid \exists u' \in F(Q).\, F(Q \hookrightarrow X)(u') = u\}.$$

This yields a natural transformation $\sigma\colon \wp \Rightarrow \wp F$ in case the functor F preserves weak pullbacks.

3.1 Lifted functor algebras form an indexed category

We have considered algebras and coalgebras of lifted functors (like in Lemma 1) as predicates. This will be (further) justified below, by showing that these categories actually arise via the $\int$ construction of suitably defined indexed categories.

Definition 3. *Let (F, σ) be a functor with predicate lifting, i.e. an endomap $\Phi \to \Phi$, where $\Phi\colon \mathbf{A}^{op} \to \mathbf{Cat}$. Define new indexed categories*

$$\mathrm{Alg}(F)^{op} \xrightarrow{\mathrm{Alg}(F,\sigma)} \mathbf{Cat} \qquad \text{and} \qquad \mathrm{CoAlg}(F)^{op} \xrightarrow{\mathrm{CoAlg}(F,\sigma)} \mathbf{Cat}$$

as follows.

1. *Let $\mathrm{Alg}(F,\sigma)\big(FX \stackrel{a}{\to} X\big)$ be the fibre category which has pairs $P \in \Phi(X)$ with $\alpha\colon \sigma(P) \to a^*(P)$ as objects. A morphism $(\sigma(P) \stackrel{\alpha}{\to} a^*(P)) \longrightarrow (\sigma(Q) \stackrel{\beta}{\to} a^*(Q))$ in this fibre is a map $h\colon P \to Q$ in $\Phi(X)$ making the following square in $\Phi(FX)$ commute.*

$$\begin{array}{ccc} \sigma(P) & \xrightarrow{\sigma(h)} & \sigma(Q) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ a^*(P) & \xrightarrow{a^*(h)} & a^*(Q) \end{array}$$

   For a homomorphism of algebras $(FX \stackrel{a}{\to} X) \stackrel{f}{\longrightarrow} (FY \stackrel{b}{\to} Y)$ we define a substitution functor between such fibre categories:

$$\begin{array}{ccc} \mathrm{Alg}(F,\sigma)\big(FY \xrightarrow{b} Y\big) & \xrightarrow{\mathrm{Alg}(F,\sigma)(f)} & \mathrm{Alg}(F,\sigma)\big(FX \xrightarrow{a} X\big) \\ (Q,\alpha) & \longmapsto & (f^*(Q), F(f)^*(\alpha)) \\ h & \longmapsto & f^*(h) \end{array}$$

   This is well-defined, since if $\alpha\colon \sigma(P) \to a^*(P)$, then $F(f)^*(\alpha)$ yields a map:

$$\sigma(f^*(Q)) = F(f)^*(\sigma(P)) \xrightarrow{F(f)^*(\alpha)} F(f)^*(a^*(P)) = b^*(f^*(P))$$

2. *In the coalgebraic case one simply defines $\mathrm{CoAlg}(F,\sigma)\big(X \stackrel{c}{\to} FX\big) = \mathrm{CoAlg}(c^*\sigma)$, so that the fibre category is the category of coalgebras of the functor $c^*\sigma\colon \Phi(X) \to$*

$\Phi(FX) \to \Phi(X)$. A morphism $f\colon (X \xrightarrow{c} FX) \longrightarrow (Y \xrightarrow{d} FY)$ in $\mathbf{CoAlg}(F)$ gives rise to a substitution functor

$$\begin{array}{ccl}
\mathbf{CoAlg}(d^*\sigma) & \xrightarrow{\mathbf{CoAlg}(F,\sigma)(f)} & \mathbf{CoAlg}(c^*\sigma) \\
\big(Q \xrightarrow{\beta} d^*(\sigma(Q))\big) & \longmapsto & \big(f^*(Q) \xrightarrow{f^*(\beta)} f^*(d^*(\sigma(Q)))\big) \\
 & & \qquad\qquad\qquad \| \\
 & & \qquad c^*(F(f)^*(\sigma(Q))) = c^*(\sigma(f^*(Q)))
\end{array}$$

The main observation is that the two previous constructions, namely talking predicates of (indexed) (co)algebras of functors with predicate lifting and taking (co)algebras of lifted functors coincide.

Proposition 1 (Predicates on (co)algebras are (co)algebras too). *In the context of the previous lemma there are equalities of categories:*

$$\begin{array}{ccccc}
 & & \circlearrowright \overline{F} & & \\
\int \mathbf{Alg}(F,\sigma) = \mathbf{Alg}(\overline{F}) & \longrightarrow & \int \Phi & \longleftarrow & \mathbf{CoAlg}(\overline{F}) = \int \mathbf{CoAlg}(F,\sigma) \\
\downarrow & & \downarrow & & \downarrow \\
\mathbf{Alg}(F) & \longrightarrow & \mathbf{A} & \longleftarrow & \mathbf{CoAlg}(F) \\
 & & F \circlearrowright & &
\end{array}$$

Proof. We shall do the algebra case. First note that on the one hand objects in the category of predicates $\int \mathbf{Alg}(F,\sigma)$ are triples consisting of an algebra $a\colon FX \to X$ together with a predicate $P \in \Phi(X)$ with a map $\alpha\colon \sigma(P) \to a^*(P)$. On the other hand, objects in the category of algebras $\mathbf{Alg}(\overline{F})$ are given by a predicate $P \in \Phi(X)$ together with an algebra map $\overline{F}(X,P) = (F(X), \sigma(P)) \to (X,P)$. The latter consists of an algebra $a\colon F(X) \to X$ together with a map $\alpha\colon \sigma(P) \to a^*(P)$. Hence we have the same data. $\square$

The functor $c^*\sigma$ in the coalgebraic part of this definition may be written as $\Box_c$ because it describes the 'nexttime' operator from modal logic. Intuitively, $\Box_c(Q)$ means: nexttime, after applying the coalgebra c, Q holds. A coalgebra $Q \to \Box_c(Q)$ is an invariant: a predicate that, once it holds, continues to hold after applications of c.

In presence of sums $\coprod$ in the indexed category Φ—as in (1)—in the algebraic case we can also describe the fibre category $\mathbf{Alg}(F,\sigma)(FX \xrightarrow{a} X)$ as a category of algebras, namely of the functor $\coprod_a \sigma\colon \Phi(X) \to \Phi(X)$. There is no established terminology for this functor, but one can read it as 'previously'. An algebra $\coprod_a(\sigma(P)) \to P$ makes P into what is sometimes called an 'inductive' predicate, since it forms the assumption in an induction statement, namely that P is closed under application of the algebra a.

The previous proposition also applies to functors with a relation lifting $(F,\sigma)\colon \Phi^2 \to \Phi^2$. The category $\int \mathbf{CoAlg}(F,\sigma) = \mathbf{CoAlg}(F)$ then contains as objects coalgebras $c\colon X \to F(X)$ together with a relation $R \in \Phi(X \times X)$ with a map $R \to (c \times c)^*(\sigma(R))$. This map shows that R is closed underlying applying c, and thus forms a bisimulation.

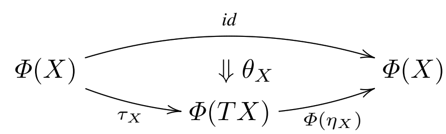
4 Liftings for monads

We now extend the approach from the previous section from functors to monads. We do not explicitly describe the comonad case because it is covered by duality.

Definition 4. *A* monad with a predicate lifting *for an indexed category is defined as a monad in the 2-category* ***IndCat****. More explicitly, for an indexed category* $\Phi\colon \mathbf{A}^{op} \to \mathbf{Cat}$ *it consists of a 1-cell* $(T, \tau)\colon \Phi \to \Phi$ *together with 2-cells* $(\eta, \theta)\colon (Id, id) \Rightarrow (T, \tau)$ *and* $(\mu, \nu)\colon (T, \tau)^2 \Rightarrow (T, \tau)$ *satisfying the familiar monad equations.*

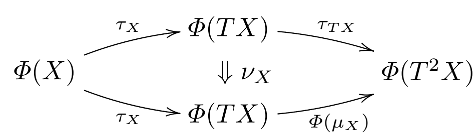
Even more explicitly, this means that we have the following data.

1. *A functor* $T\colon \mathbf{A} \to \mathbf{A}$ *with a predicate lifting* $\tau\colon \Phi \Rightarrow \Phi T$*;*
2. *A natural transformation* $\eta\colon id_{\mathbf{A}} \Rightarrow T$ *and a collection* $\theta = (\theta_X)$ *of natural transformations, where these* $\theta_X\colon id_{\Phi(X)} \Rightarrow \Phi(\eta_X)\tau_X$ *in*



must commute with substitution;

3. *A natural transformation* $\mu\colon T^2 \Rightarrow T$ *and a collection* $\nu = (\nu_X)$ *with* $\nu_X\colon \tau_{TX}\tau_X \Rightarrow \Phi(\mu_X)\tau_X$ *in*



commuting with substitution;

4. *Equations:*

$$\mu_X \circ \eta_{TX} = id_{TX} \quad \textit{and} \quad \Phi(\eta_{TX})\nu_X \circ \theta_{TX}\tau_X = id : \tau_X \Longrightarrow \tau_X$$

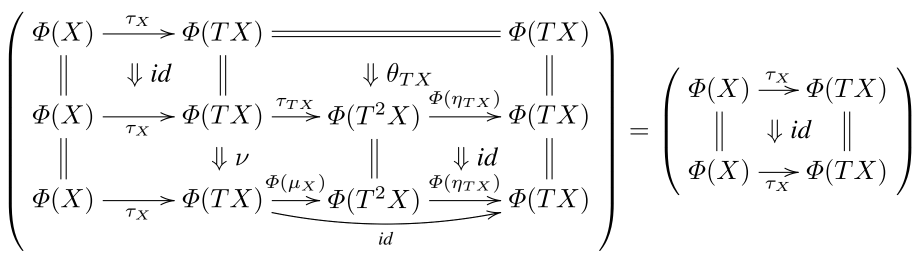
and:

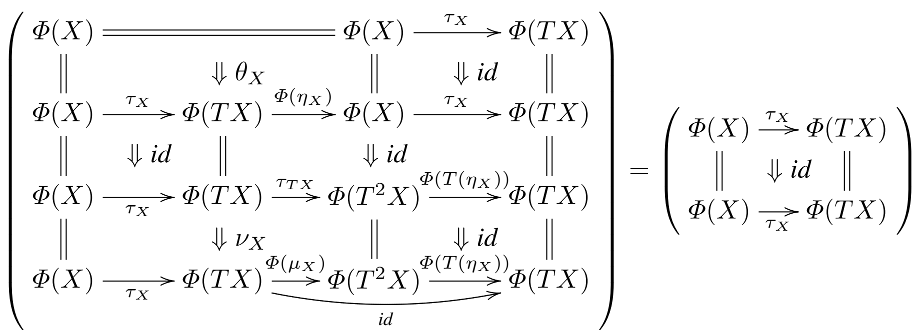
$$\mu_X \circ T(\eta_X) = id_{TX} \quad \textit{and} \quad \Phi(T(\eta_X))\nu_X \circ \tau_X\theta_X = id : \tau_X \Longrightarrow \tau_X$$

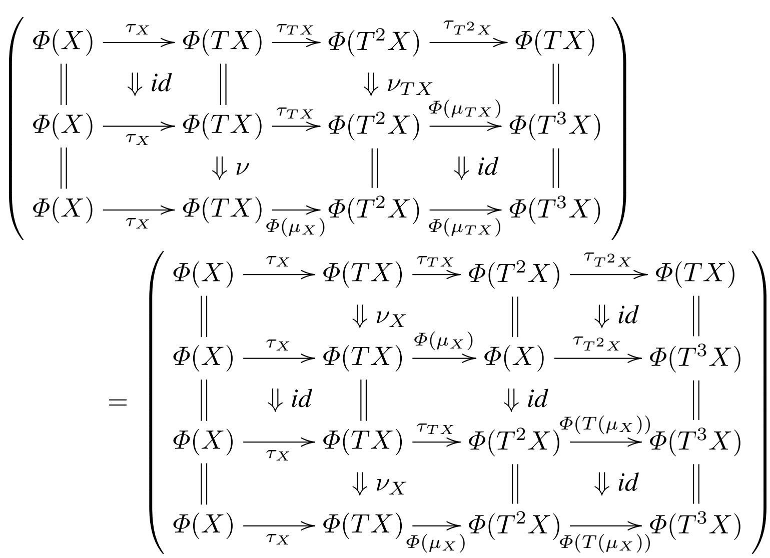
and:

$$\mu_X \circ \mu_{TX} = \mu_X \circ T(\mu_X) \quad \textit{and} \quad \begin{aligned} &\Phi(\mu_{TX})\nu_X \circ \nu_{TX}\tau_X \\ &\qquad = \Phi(T(\mu_X))\nu_X \circ \tau_{T^2(X)}\nu_X. \end{aligned}$$

The equations on the left express that $T\colon \mathbf{A} \to \mathbf{A}$ *with* η, μ *is a monad in the ordinary sense. The equations on the right are a bit more complicated. Diagrammatically, they* *look as follows.*







Finally, we call such a monad with predicate lifting split *when the θ's and ν's are all identities. In that case these right-hand-side equations—and thus the above diagrams—trivially hold.*

Example 4. We shall consider several examples, and also a non-example, for the standard subset logic $\wp\colon \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Cat}$.

1. Recall that we write $\mathcal{P}$ for the powerset monad **Sets** $\to$ **Sets**, whose unit is given by singletons and multiplication by unions. This monad $\mathcal{P}$ can be provided in a standard way with a split predicate lifting for the indexed category $\wp$. It involves a natural transformation with components:

$$\begin{array}{rcl} \wp(X) & \xrightarrow{\quad\pi_X\quad} & \wp(\mathcal{P}(X)) \\ (Q \subseteq X) & \longmapsto & \{U \in \mathcal{P}(X) \mid U \subseteq Q\} \end{array}$$

It is important to distinguish the contravariance of $\wp$ and the covariance of $\mathcal{P}$ in order to see that π is natural: for each function $f\colon X \to Y$ one has a commuting diagram:

$$\begin{array}{ccc} \wp(X) & \xrightarrow{\quad\pi_X\quad} & \wp(\mathcal{P}(X)) \\ {\scriptstyle \wp(f)=f^{-1}}\uparrow & & \uparrow{\scriptstyle \wp(\mathcal{P}(f))=\mathcal{P}(f)^{-1}} \\ \wp(Y) & \xrightarrow{\quad\pi_Y\quad} & \wp(\mathcal{P}(Y)) \end{array}$$

since for $Q \subseteq Y$ one has:

$$\begin{aligned} \big(\mathcal{P}(f)^{-1} \circ \pi_Y\big)(Q) &= \{U \in \mathcal{P}(X) \mid \mathcal{P}(f)(U) \in \pi_Y(Q)\} \\ &= \{U \in \mathcal{P}(X) \mid \{f(x) \mid x \in U\} \subseteq Q\} \\ &= \{U \in \mathcal{P}(X) \mid \forall x \in U.\, f(x) \in Q\} \\ &= \{U \in \mathcal{P}(X) \mid U \subseteq f^{-1}(Q)\} \\ &= \big(\pi_X \circ f^{-1}\big)(Q). \end{aligned}$$

This endomap $(\mathcal{P}, \pi)\colon \wp \to \wp$ is a monad with split predicate lifting since we have equalities (instead of inclusions):

$$\begin{aligned} \big(\wp(\eta) \circ \pi_X\big)(Q) &= \{x \in X \mid \eta(x) \in \pi_X(Q)\} \\ &= \{x \in X \mid \{x\} \subseteq Q\} \\ &= Q \\ \big(\wp(\mu) \circ \pi_X\big)(Q) &= \{A \in \mathcal{P}^2(X) \mid \mu(A) \in \pi_X(Q)\} \\ &= \{A \in \mathcal{P}^2(X) \mid \textstyle\bigcup A \subseteq Q\} \\ &= \{A \in \mathcal{P}^2(X) \mid \forall U \in A.\, U \subseteq Q\} \\ &= \{A \in \mathcal{P}^2(X) \mid A \subseteq \pi_X(Q)\} \\ &= \big(\pi_{\mathcal{P}(X)} \circ \pi_X\big)(Q) \end{aligned}$$

The predicate lifting π can be combined with negation $\neg\colon \wp(X) \to \wp(X)$. It leads to another split predicate lifting $\neg\pi\neg$ for the powerset monad $\mathcal{P}$. Explicitly, it is given by:

$$\begin{aligned} (\neg\pi\neg)_X(Q) \;=\; \neg\pi_X(\neg Q) &= \{U \in \mathcal{P}(X) \mid \text{not: } U \subseteq \neg Q\} \\ &= \{U \in \mathcal{P}(X) \mid U \cap Q \neq \emptyset\}. \end{aligned}$$

2. The ultrafilter monad $\mathcal{UF}$: **Sets** $\rightarrow$ **Sets** is given by:

$$\mathcal{UF}(X) = \{A \subseteq \mathcal{P}(X) \mid A \text{ is an ultrafilter}\}.$$

It is relevant in topology, for instance because its Eilenberg-Moore algebras are precisely the compact Hausdorff spaces (Manes' theorem, see *e.g.* [8, III, 2.4]). This ultrafilter monad carries a predicate lifting:

$$\begin{array}{ccc} \wp(X) & \xrightarrow{\varphi_X} & \wp(\mathcal{UF}(X)) \\ (Q \subseteq X) & \longmapsto & \{A \in \mathcal{UF}(X) \mid Q \in A\} \end{array}$$

Like before this forms a split predicate lifting.

3. If $M = (M, 1, \cdot)$ is a monoid, then $M \times (-)$: **Sets** $\rightarrow$ **Sets** is a monad. We can apply the powerset, and get another monad $\mathcal{P}(M \times -)$: **Sets** $\rightarrow$ **Sets** with unit η and multiplication μ given on $x \in X$ and $A \subseteq M \times \mathcal{P}(M \times X)$ by:

$$\begin{aligned} \eta(x) &= \{(1, x)\} \\ \mu(A) &= \{(m, x) \mid \exists (m_1, U) \in A.\, \exists m_2.\, (m_2, x) \in U \wedge m = m_2 \cdot m_1\}. \end{aligned}$$

There is an associated predicate lifting

$$\begin{array}{ccc} \wp(X) & \xrightarrow{\varphi_X} & \wp(\mathcal{P}(M \times X)) \\ (Q \subseteq X) & \longmapsto & \{U \in \mathcal{P}(M \times X) \mid \forall (m, x) \in U.\, Q(x)\}. \end{array}$$

It interacts appropriately with the unit and multiplication of the monad $\mathcal{P}(M \times -)$, making the latter a monad with predicate lifting.

4. Next we consider the distribution monad $\mathcal{D}$ on **Sets**, with

$$\mathcal{D}(X) = \{\varphi\colon X \rightarrow [0, 1] \mid \textit{supp}(\varphi) \text{ is finite, and } \textstyle\sum_{x \in X} \varphi(x) = 1\}.$$

Here one write $\textit{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$. It is convenient to write φ as a formal convex sum $\varphi = r_1 x_1 + \cdots + r_n x_n$ where $\textit{supp}(\varphi) = \{x_1, \ldots, x_n\}$ and $r_i = \varphi(x_i)$ is the probability of x_i. One can then write $\mathcal{D}(f)$, for $f\colon X \rightarrow Y$, as $\mathcal{D}(f)(\sum_i r_i x_i) = \sum_i r_i f(x_i)$. The unit for $\mathcal{D}$ is $\eta(x) = 1x$, and the multiplication is $\mu(\Phi) = \lambda x.\, \sum_\varphi \Phi(\varphi) \cdot \varphi(x)$.
There are several ways in which to associate a lifting with the distribution monad $\mathcal{D}$, but not all of them are predicate liftings. We start with a lifting that does work:

$$\begin{array}{ccc} \wp(X) & \xrightarrow{\tau_X} & \wp(\mathcal{D}(X)) \\ (Q \subseteq X) & \longmapsto & \{\varphi \in \mathcal{D}(X) \mid \textit{supp}(\varphi) \subseteq Q\}. \end{array}$$

Hence this lifting expresses that all elements with non-empty probability of occurring should be in P. It does not take the value of the probability into account, but only distinguishes zero/non-zero. This forms a split predicate lifting.

There is also a more refined way in which one does take probability values into account. Then one associates multiple liftings $\tau(q)$ with $\mathcal{D}$, one for each $q \in \mathbb{Q} \cap [0, 1]$, namely:

$$\begin{array}{ccc} \wp(X) & \xrightarrow{\quad\tau(q)_X\quad} & \wp(\mathcal{D}(X)) \\ (Q \subseteq X) & \longmapsto & \{\varphi \in \mathcal{D}(X) \mid q \leq \sum_{x \in Q} \varphi(x)\}. \end{array}$$

Each $\tau(q)$ is a natural transformation, but the pairs $(\mathcal{D}, \tau(q))$ do not give rise to a monad with predicate liftings. The reason is that in the multiplication case one does have to allow variation in the q, within a single equation, as illustrated in:

$$\begin{aligned} \big(\tau(q)_{\mathcal{D}(X)} \circ \tau(r)_X\big)(Q). &= \{\Phi \in \mathcal{D}^2(X) \mid q \leq \textstyle\sum_{\varphi \in \tau(r)_X(Q)} \Phi(\varphi)\} \\ &\subseteq \{\Phi \in \mathcal{D}^2(X) \mid q \cdot r \leq \textstyle\sum_{\varphi} \Phi(\varphi) \cdot (\sum_{x \in Q} \varphi(x))\} \\ &= \{\Phi \in \mathcal{D}^2(X) \mid q \cdot r \leq \textstyle\sum_{x \in Q} \sum_{\varphi} \Phi(\varphi) \cdot \varphi(x)\} \\ &= \{\Phi \in \mathcal{D}^2(X) \mid q \cdot r \leq \textstyle\sum_{x \in Q} \mu(\Phi)(x)\} \\ &= \{\Phi \in \mathcal{D}^2(X) \mid \mu(\Phi) \in \tau(q \cdot r)_X(Q)\} \\ &= \big(\wp(\mu) \circ \tau(q \cdot r)_X\big)(Q) \end{aligned}$$

Hence in this case the $(\mathcal{D}, \tau(q))$ form a *functor with predicate lifting* but not a *monad with predicate lifting*.

Monads with predicate liftings, like functors, can be lifted.

Lemma 2. *Let $(T, \tau): \Phi \to \Phi$ be a monad with predicate lifting, for $\Phi: \mathbf{A}^{op} \to \mathbf{Cat}$. It can be lifted to $\overline{T}: \int \Phi \to \int \Phi$, like in Lemma 1. This yields a monad again, and a map of monads in a square:*

$$\begin{array}{ccc} \int \Phi & \xrightarrow{\quad\overline{T}\quad} & \int \Phi \\ \downarrow & & \downarrow \\ \mathbf{A} & \xrightarrow{\quad T\quad} & \mathbf{A} \end{array} \qquad \textit{and thus functors} \qquad \begin{array}{c} \mathcal{K}\ell(\overline{T}) \\ \downarrow \\ \mathcal{K}\ell(T) \end{array} \qquad \begin{array}{c} \mathit{Alg}(\overline{T}) \\ \downarrow \\ \mathit{Alg}(T) \end{array}$$

The lifting $\overline{T}$ is split in case the predicate lifting is split, whence the name.

The notation *Alg* is used here for the category of Eilenberg-Moore algebras of the monad $\overline{T}$—and not for the algebras of a functor, like in the previous section. We use $\mathcal{K}\ell(T)$ to denote the Kleisli category of the monad T.

Proof. One defines a unit $\overline{\eta}_{(X,P)} = (\eta_X, \theta_{X,P}): (X, P) \to \overline{T}(X, P) = (T(X), \tau_X(P))$. It is well-defined because $\eta_X: X \to T(X)$ in $\mathbf{A}$ and $\theta_{X,P}: P \to \eta_X^*(\tau_X(P))$, as required. Similarly, one defines multiplication for $\overline{T}$ as:

$$(T^2(X), \tau_{TX}(\tau_X(P))) = \overline{T}^2(X, P) \xrightarrow{\overline{\mu}_{(X,P)} = (\mu_X, \nu_{X,P})} \overline{T}(X, P) = (T(X), \tau_X(P)).$$

The monad equations for $\overline{T}$ boil down to the equations listed in Definition 4.(4). $\square$

Our next step, like in the previous section is to define an indexed category whose categories of predicate are $\mathcal{K}\ell(\overline{T})$ and $\mathcal{A}lg(\overline{T})$. In the Kleisli case this works for monads with *split* predicate lifting.

Definition 5. *Let $(T,\tau)\colon \Phi \to \Phi$ be a monad with* split *predicate lifting, for an indexed category $\Phi\colon \mathbf{A}^{op} \to \mathbf{Cat}$. We define a new indexed category:*

$$\mathcal{K}\ell(T)^{op} \xrightarrow{\mathcal{K}\ell(T,\tau)} \mathbf{Cat}$$

by setting $\mathcal{K}\ell(T)(X) = \Phi(X)$ and for $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, i.e. for $f\colon X \to T(Y)$ in $\mathbf{A}$, taking as substitution functor:

$$\mathcal{K}\ell(T,\tau)(f) = \Big(\mathcal{K}\ell(T,\tau)(Y) = \Phi(Y) \xrightarrow{\tau_Y} \Phi(TY) \xrightarrow{\Phi(f)} \Phi(X) = \mathcal{K}\ell(T,\tau)(X)\Big).$$

In verifying that $\mathcal{K}\ell(T,\tau)$ is a functor we use that that the predicate lifting is split:

$$\begin{aligned}
\mathcal{K}\ell(T,\tau)(\mathrm{id}_X) &= \eta_X^* \circ \tau_X \\
&= \mathrm{id}_{\Phi(X)} && \text{since the predicate lifting is split} \\
&= \mathrm{id}_{\mathcal{K}\ell(T,\tau)(X)} \\
\mathcal{K}\ell(T,\tau)(g \circ f) &= (\mu \circ T(g) \circ f)^* \circ \tau \\
&= f^* \circ T(g)^* \circ \mu^* \circ \tau \\
&= f^* \circ T(g)^* \circ \tau \circ \tau && \text{idem} \\
&= f^* \circ \tau \circ g^* \circ \tau \\
&= \mathcal{K}\ell(T,\tau)(f) \circ \mathcal{K}\ell(T,\tau)(g).
\end{aligned}$$

This restriction to split predicate liftings is not needed in the case of algebras. We shall not use this construction, but include it for reasons of completeness.

Definition 6. *Let $(T,\tau)\colon \Phi \to \Phi$ be a monad with an arbitrary predicate lifting, for an indexed category $\Phi\colon \mathbf{A}^{op} \to \mathbf{Cat}$. We now define a new indexed category:*

$$\mathrm{Alg}(T)^{op} \xrightarrow{\mathrm{Alg}(T,\tau)} \mathbf{Cat}$$

basically by following Definition 3, except that some additional conditions are imposed on objects in the fibres. The fibre category $\mathrm{Alg}(T,\tau)(TX \xrightarrow{a} X)$ now has as objects maps $\alpha\colon \tau_X(P) \to a^*(P)$ satisfying two conditions:

$$\begin{array}{ccc}
P & = & P \\
{\scriptstyle \theta_{X,P}}\downarrow & & \| \\
\eta_X^*(\tau_X(P)) & \xrightarrow[\eta_X^*(\alpha)]{} & \eta_X^* a^*(P)
\end{array}
\qquad
\begin{array}{ccc}
\tau_{TX}(\tau_X(P)) & \xrightarrow{\tau_{TX}(\alpha)} & \tau_{TX}(a^*(P)) = T(a)^*(\tau_X(P)) \\
{\scriptstyle \nu_{X,P}}\downarrow & & \downarrow{\scriptstyle T(a)^*(\alpha)} \\
\mu_X^*(\tau_X(P)) & \xrightarrow[\mu^*(\alpha)]{} & \mu_X^*(a^*(P)) = T(a)^*(a^*(P))
\end{array}$$

A morphism $h\colon \alpha \to \beta$ in $\mathrm{Alg}(T,\tau)(TX \xrightarrow{a} X)$ is, like in Definition 3, a map satisfying $a^*(h) \circ \alpha = \beta \circ \tau_X(h)$. Substitution is also like in Definition 3.

Proposition 2. *In the context of the previous definitions there are equalities of categories:*

$$\begin{array}{ccccc} & & \overset{\overline{T}}{\circlearrowleft} & & \\ \int \mathcal{K}\ell(T,\tau) = \mathcal{K}\ell(\overline{T}) & \longrightarrow & \int \varPhi & \longleftarrow & \mathcal{A}lg(\overline{T}) = \int \mathcal{A}lg(T,\tau) \\ \downarrow & \text{(splitting assumed)} & \downarrow & & \downarrow \\ \mathcal{K}\ell(T) & \longrightarrow & \boldsymbol{A} & \longleftarrow & \mathcal{A}lg(T) \\ & & \underset{T}{\circlearrowright} & & \end{array}$$

Proof. An object of $\mathcal{K}\ell(\overline{T})$ is an object $(X, P) \in \int \varPhi$, where $X \in \mathbf{A}$ and $P \in \varPhi(X) = \mathcal{K}\ell(T, \tau)(X)$. A morphism $(f, p): (X, P) \to (Y, Q)$ in $\mathcal{K}\ell(\overline{T})$ consists of a map $(f, p): (X, P) \to \overline{T}(X, P)$ in $\int \varPhi$; it involves $f: X \to T(Y)$ in $\mathbf{A}$ and $p: P \to f^*(\tau_Y(Q))$ in $\varPhi(X)$. Hence f is a map $X \to Y$ in the category $\mathcal{K}\ell(T)$, and p is a map $P \to \mathcal{K}\ell(T, \tau)(f)(Q)$. Thus (f, p) is a map in the category of predicates $\int \mathcal{K}\ell(T, \tau)$.

An object of $\mathcal{A}lg(\overline{T})$ consists of a $\overline{T}$-algebra $(a, \alpha): \overline{T}(X, P) \to (X, P)$. It thus consists of maps $a: TX \to X$ in $\mathbf{A}$ and $\alpha: \tau_X(P) \to a^*(P)$ in $\varPhi(X)$ satisfying:

$$\begin{aligned} (a \circ \eta_X, \eta_X^*(\eta) \circ \theta_{X,P}) &= (a, \alpha) \circ \overline{\eta}_{(X,P)} \\ &= (\mathrm{id}, \mathrm{id}) \\ (a \circ \mu_X, \mu_X^*(\alpha) \circ \nu_{X,P}) &= (a, \alpha) \circ \overline{\mu}_{(X,P)} \\ &= (a, \alpha) \circ \overline{T}(a, \alpha) \\ &= (a \circ T(a), T(a)^*(\alpha) \circ \tau_X(\alpha)). \end{aligned}$$

These equations express precisely that a a T-algebra and that α is a map in the fibre category $\mathcal{A}lg(T, \tau)(TX \xrightarrow{a} X)$. $\square$

We conclude with some examples of logics on Kleisli categories, in which it will turn out that substitution functors are compositional modalities.

Example 5. We review some of the cases from Example 4.

1. The Kleisli category $\mathcal{K}\ell(\mathcal{P})$ of the powerset monad is of course the category of sets and relations between them. The split predicate lifting $(\mathcal{P}, \pi): \wp \to \wp$ from Example 4.(1) gives rise to an indexed category $\mathcal{K}\ell(\mathcal{P}, \pi): \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} \to \mathbf{Cat}$. It sends $X \mapsto \wp(X)$ and $f: X \to \mathcal{P}(Y)$ to the substitution functor

$$\wp(Y) \xrightarrow{\mathcal{K}\ell(\mathcal{P},\pi)(f) = \Box_f} \wp(X)$$

described—according to Definition 5—by:

$$\begin{aligned} \mathcal{K}\ell(\mathcal{P}, \pi)(f)(Q \subseteq Y) &= f^{-1}(\pi_Y(Q)) \\ &= \{x \in X \mid f(x) \in \pi_Y(Q)\} \\ &= \{x \in X \mid f(x) \subseteq Q\} \\ &= \Box_f(Q). \end{aligned}$$

Thus $\Box_f(Q)$ is the predicate that says "Q holds everywhere after applying f", *i.e.* "Q holds for all successor states via f". Notice that this modal operator is compositional, in the sense that $\Box_{g \circ f} = \Box_f \Box_g$, where $\circ$ refers to (relation) composition in the Kleisli category. This compositionality follows directly from the functoriality of $\mathcal{K}\ell(\mathcal{P}, \pi)$. These modalities preserve intersections because the predicate lifting π does.

This is precisely the logic on relations that arises in the context of quantum logic via daggers and kernels, see [5, §§3.1].

The modal operator associated with the predicate lifting $\neg\pi\neg$ is of course the operator $\Diamond$ that expresses "for some successor state ...". Indeed, for $f \colon X \to \mathcal{P}(Y)$ we have:

$$\begin{aligned}
\mathcal{K}\ell(\mathcal{P}, \neg\pi\neg)(f)(Q \subseteq Y) &= f^{-1}((\neg\pi\neg)_Y(Q)) \\
&= \{x \in X \mid f(x) \notin \pi_Y(\neg Q)\} \\
&= \{x \in X \mid f(x) \cap Q \neq \emptyset\} \\
&= \Diamond_f(Q).
\end{aligned}$$

2. The predicate lifting φ for the ultrafilter monad $\mathcal{UF}$ from Example 4.(2) yields a similar compositional modality: for a map $f \colon X \to \mathcal{UF}(Y)$ we get $\Box_f \colon \wp(Y) \to \wp(X)$ by:

$$\begin{aligned}
\Box_f(Q) &= \{x \in X \mid f(x) \in \varphi_Y(Q)\} \\
&= \{x \in X \mid Q \in f(x)\}.
\end{aligned}$$

   This modality preserves all Boolean operations because $\varphi \colon \wp(X) \to \wp(\mathcal{UF}(X))$ does so.

3. The starting point for dynamic logic, see *e.g.* [3], is a set A of actions together with a relation $R_a \subseteq X \times X$, for each $a \in A$. We shall extend these relations to sequences as $R_\alpha \subseteq X \times X$, for $\alpha \in A^\star$, in the obvious way:

$$R_{\langle a_1, \ldots, a_n \rangle} = R_{a_n} \circ \cdots \circ R_{a_1}.$$

   Clearly, $R_{\alpha;\beta} = R_\beta \circ R_\alpha$, where ; is sequential composition of lists, and $\circ$ is relational composition. Each such relation R_α yields a coalgebra $\overline{R}_\alpha \colon X \to \mathcal{P}(A^\star \times X)$, for the monad $\mathcal{P}(A^\star \times -)$ from Example 4.(3), via $\overline{R}_\alpha(x) = \{(\alpha, y) \mid (x, y) \in R_\alpha\}$. It is then not hard to see that:

$$\overline{R}_\alpha \circ \overline{R}_\beta = \overline{R}_{\alpha;\beta}, \tag{5}$$

   where the composition $\circ$ on the left-hand-side is in the Kleisli category of the monad $\mathcal{P}(A^\star \times -)$.

   Our general approach now gives a modality, namely $\Box_\alpha(Q) = \left(\overline{R}_\alpha\right)^*(Q)$, using substitution $(-)^*$ in the "Kleisli" indexed category from Definition 5. It is of course the same as the usual one, when specialised to a singleton sequence:

$$\begin{aligned}\Box_a(Q) &= \big(\overline{R}_a\big)^*(Q)\\ &= \big(\overline{R}_a^{-1}\circ\sigma\big)(Q) \qquad \text{with } \sigma \text{ as in Example 4.(3)}\\ &= \{x \mid \overline{R}_a(x)\in\sigma(Q)\}\\ &= \{x \mid \forall(\alpha,y)\in\overline{R}_a(x).\, Q(y)\}\\ &= \{x \mid \forall y.\, R_a(x,y)\Rightarrow Q(y)\}.\end{aligned}$$

The compositionality property follows from the general framework:

$$\begin{aligned}\Box_\beta(\Box_\alpha(Q)) &= \overline{R}_\beta^*\big(\overline{R}_\alpha^*(Q)\big)\\ &= \big(\overline{R}_\alpha\circ\overline{R}_\beta\big)^*(Q)\\ &\overset{(5)}{=} \big(\overline{R}_{\alpha;\beta}\big)^*(Q)\\ &= \Box_{\alpha;\beta}(Q).\end{aligned}$$

Of course the general framework is a bit of an overkill for proving compositionality in this particular example. The point is rather to illustrate how the general framework covers many known cases and clarifies the underlying structure.

At the end we briefly mention two possible topics for further research.

1. The current approach focuses on compositionality for sequential composition (in Kleisli) categories. It remains open how to integrate other process theoretic operators, involving for instance parallel composition in component calculi like in [1,6]. One expects results about lifting products and coproducts to total categories $\int\Phi$ to be useful here, see *e.g.* [4, Lemma 2.1].
2. The predicate lifting $\pi\colon\wp\Rightarrow\wp\mathcal{P}$ used in Example 4.(1) can be defined in the predicate logic of the indexed category $\wp$ itself. In order to do so we use that the inhabitation relation $\in_X$ can be seen as an object of the fibre category $\wp(\mathcal{P}(X)\times X)$, namely given by $\in_X = \{(U,x)\mid x\in U\}$. Then we can define the predicate lifting π from Example 4.(1) internally via quantification (1) and fibred implication $\Rightarrow$ as:
$$\begin{aligned}\pi_X(Q) &= \textstyle\prod_{\pi_1}\big(\in_X\Rightarrow\pi_2^*(Q)\big) \quad \text{in} \quad \wp(\mathcal{P}(X)),\\ &= \{U\in\mathcal{P}(X)\mid\forall x\in X.\, x\in U\Rightarrow Q(x)\}.\end{aligned}$$
where $\mathcal{P}(X)\xleftarrow{\pi_1}\mathcal{P}(X)\times X\xrightarrow{\pi_2}X$. Indexed categories can thus be used to express when a predicate lifting (and thereby a modal operator) can be defined within the predicate logic itself. This may be applied to any predicate logic Φ, and thus gives a flexible, semantic approach to definability of modal operators in predicate logic, see [12] for a more syntactic approach (and further references).

References

1. L.S. Barbosa. Towards a calculus of state-based software components. *Journ. of Universal Comp. Sci.*, 9(8):891–909, 2003.

2. C. Cîrstea, A. Kurz, D. Pattinson, L. Schröder, and Y. Venema. Modal logics are coalgebraic. *The Computer Journal*, 2009. doi:10.1093/comjnl/bxp004.
3. R. Goldblatt. *Logics of Time and Computation*. CSLI Lecture Notes 7, Stanford, 2nd rev. edition, 1992.
4. C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. & Comp.*, 145:107–152, 1998.
5. C. Heunen and B. Jacobs. Quantum logic in dagger kernel categories. *Order*, 2010. doi:10.1007/s11083-010-9145-5.
6. B. Jacobs I. Hasuo, C. Heunen and A. Sokolova. Coalgebraic components in a many-sorted microcosm. In A. Kurz and A. Tarlecki, editors, *Conference on Algebra and Coalgebra in Computer Science (CALCO 2009)*, number 5728 in Lect. Notes Comp. Sci., pages 64–80. Springer, Berlin, 2009.
7. B. Jacobs. *Categorical Logic and Type Theory*. North Holland, Amsterdam, 1999.
8. P.T. Johnstone. *Stone Spaces*. Number 3 in Cambridge Studies in Advanced Mathematics. Cambridge Univ. Press, 1982.
9. N. Ghani P. Johann and C. Fumex. Fibrational induction rules for initial algebras. Submitted to CSL 2010, 2010.
10. R. Paré and D. Schumacher. Abstract families and the adjoint functor theorems. In P.T. Johnstone and R. Paré, editors, *Indexed Categories and their Applications*, number 661 in Lect. Notes Math., pages 1–125. Springer, Berlin, 1978.
11. D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theor. Comp. Sci.*, 309(1-3):177–193, 2003.
12. L. Schröder and D. Pattinson. Coalgebraic correspondence theory. In L. Ong, editor, *Foundations of Software Science and Computation Structures*, number 6014 in Lect. Notes Comp. Sci., pages 328–342. Springer, Berlin, 2010.