The Temporal Logic of Coalgebras via Galois Algebras

B.P.F. Jacobs

Computing Science Institute/

CSI-R9906 April 1999

Computing Science Institute Nijmegen
Faculty of Mathematics and Informatics
Catholic University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands

The Temporal Logic of Coalgebras via Galois Algebras

Bart Jacobs*

Department of Computer Science, University of Nijmegen,
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands.
Email: bart@cs.kun.nl URL: http://www.cs.kun.nl/~bart

Abstract

This paper introduces a temporal logic for coalgebras. Nexttime and lasttime operators are defined for a coalgebra, acting on predicates on the state space. They give rise to what is called a Galois algebra. Galois algebras form models of temporal logics like *Linear Temporal Logic* (LTL) and *Computation Tree Logic* (CTL). The mapping from coalgebras to Galois algebras turns out to be functorial, yielding indexed categorical structures. This construction gives many examples, for coalgebras of polynomial functors on sets. More generally, it will be shown how "fuzzy" predicates on metric spaces, and predicates on presheaves, yield indexed Galois algebras, in basically the same coalgebraic manner.

Keywords: Temporal logic, coalgebra, Galois connection, fuzzy predicate, presheaf
Classification: 68Q60, 03G05, 03G25, 03G30 (AMS'91); D.2.4, F.3.1, F.4.1 (CR'98).

1 Introduction

This paper combines the areas of coalgebra and of temporal logic. Coalgebras are simple mathematical structures (similar, but dual, to algebras), underlying state-based dynamical systems [JR97, Rut99], including automata, transition systems and classes in object-oriented languages. Temporal logic is a logic which is particularly suitable for reasoning about (reactive) state-based systems, as argued for example in [Pnu77, Pnu81], via its nexttime and lasttime operators. Hence one expects a connection. It is probably Moss [Mos99] who was the first to realise that the shape of a coalgebra (as given by its interface functor) determines a logical modal language. His emphasis lies on characterisation results, capturing bisimilarity as validity for the same formulas. This line is followed in [Röß99b, Röß99a, Kur98]. Here the emphasis lies on the temporal aspects of a coalgebra, in particular on the associated nexttime and lasttime operators. Moreover, this is basically a semantical study, leaving proof-theoretic aspects for future work.

We will give a sketch of the underlying developments, leaving some notions (at this stage) without precise definition. Let $\alpha: X \to T(X)$ be a coalgebra. One can

*Research Fellow of the Royal Netherlands Academy of Arts and Sciences.

think of it as a particular transition system (in particular when T is powerset $\mathcal{P}$), with the set X as its state-space (*i.e.* set of states). There is the following familiar definition of bisimilarity ($\underline{\leftrightarrow}_\alpha$) with respect to α. For $x, y \in X$,

$$x \underline{\leftrightarrow}_\alpha y \iff \exists R \subseteq X \times X.\, R \text{ is an } \alpha\text{-bisimulation, and } R(x, y).$$

This introduces bisimilarity as the greatest bisimulation. It contains those pairs of states which are observationally indistinguishable.

Besides bisimulation, invariance is very important in the theory of coalgebras (and in system theory in general). An invariant is a predicate on the state space which is maintained by all operations. The following definition is probably less familiar. For an arbitrary predicate $P \subseteq X$, a new predicate $\Box P \subseteq X$ is defined as:

$$\Box P(x) \iff \exists Q \subseteq X.\, Q \text{ is an } \alpha\text{-invariant, } Q \subseteq P, \text{ and } Q(x).$$

It is not hard to see that $\Box P$ is the greatest invariant contained in P. It may be read as: "henceforth P", that is, "in all future states, P holds". We like to write $\underset{\Rightarrow}{\alpha} P$ for $\Box P$. There is a related operation $\underset{\Leftarrow}{\alpha}$ on predicates on X, given by:

$$\underset{\Leftarrow}{\alpha} P(x) \iff \forall Q \subseteq X.\, Q \text{ is an } \alpha\text{-invariant and } P \subseteq Q \text{ implies } Q(x).$$

Then $\underset{\Leftarrow}{\alpha} P$ is the least invariant containing P. It may be read as: "in some earlier state, P holds". There is the following fundamental Galois connection: $\underset{\Leftarrow}{\alpha} P_1 \subseteq P_2 \Leftrightarrow P_1 \subseteq \underset{\Rightarrow}{\alpha} P_2$.

The definitions of $\underset{\Rightarrow}{\alpha} P$ and $\underset{\Leftarrow}{\alpha} P$ occur in [Rut99] (as $[P]$ and $\langle P \rangle$ respectively) and in [Jac97] (as $\underline{P}$ and $\overline{P}$). In [Jac97, before Proposition 3.8] the connections with temporal logic are mentioned, but not elaborated. Also the single-step, future and past operators $\underset{\rightarrow}{\alpha}$ and $\underset{\leftarrow}{\alpha}$ occur there. The full impact of these operators becomes apparent when they are identified as giving examples of "Galois algebras", introduced in [Kar98]. These Galois algebras are simple structures consisting of a complete Boolean algebra carrying a Galois connection (in the spirit of [JT51]). The latter is interpreted as the connection between lasttime and nexttime operators. It is shown in [Kar98] that all axioms and rules of Computation Tree Logic (CTL) [MP92, Eme90, Gol92] are valid in Galois algebras, and that all axioms and rules of Linear Temporal Logic (LTL) are valid in Galois algebras satisfying certain linearity conditions. Several examples of Galois algebras are given in [Kar98], but no systematic construction is presented. The main contribution of this paper lies in establishing a connection between coalgebras and temporal logic (at a semantical level), by showing that each coalgebra (of a suitable functor T) gives rise to a Galois algebra. Technically, this mapping is functorial, and gives rise to "coalgebra-indexed Galois algebras" of the form

$$\mathrm{CoAlg}(T)^{\mathrm{op}} \xrightarrow{\qquad\qquad} \mathbf{GA}$$

where $\mathrm{CoAlg}(T)$ is the category of coalgebras of the functor T, and where $\mathbf{GA}$ is the category of Galois algebras (see Theorem 6.1 below). Further, it will be shown how familiar models of temporal logic given by fuzzy predicates and presheaves [GM88] exhibit the same underlying structure of coalgebra-indexed Galois algebras. Probably, the contribution of this paper lies not so much in the results that are obtained, but more in the integration of fields.

One area of direct application of the definitions and results in this paper is specification and verification for classes in object-oriented languages, based on coalgebras, see [Rei95, Jac96, Jac97, HHJT98, JvdBH$^+$98, Hen99]. When a class is seen as a coalgebra of its interface functor, then the definitions in this paper give tailor-made temporal operators for the class, incorporating appropriate clauses for all the methods of the class. This makes it possible to formulate and prove properties about future and past states of an object of the class. In particular, safety and liveness properties of classes can be expressed, and also refinements (as in [Jac97]) can be formulated via $\Box$. A suitable logical language with temporal operators for such coalgebraic specifications will be described elsewhere (together with a comparison with alternative approaches [SSC95] based on Kripke structures). This is a topic on its own. Here we will simply give an illustration, see Example 3.7 (iii).

A crucial aspect of the connection between coalgebra and temporal logic that is unveiled here is that it is parametric in the functor (or interface) and coalgebra involved. This means that the definitions of the temporal operators can be instantiated with different functors (and coalgebras) and thus give different logics. This opens a new perspective, in which for example the operators from LTL and CTL arise from the same pattern, see Example 3.7. Also this opens up new research questions. One of the more interesting ones involves the possibility of model checking [CE81, McM93] for coalgebras, in suitably parametrised form.

The paper is organised as follows. It starts with a preliminary section providing some order-theoretic background information, and also introducing the definition of Galois algebras. The next section 3 shows how coalgebras of so-called polynomial functors on sets give rise to Galois algebras, making crucial use of "predicate lifting". Section 4 forms an intermezzo, showing how the temporal operators can also be defined pathwise, as in [Mos99, Rößß99b, Röß99a, Kur98], and give rise to Galois algebras as well. Then, Section 5 elaborates on Galois algebras. Most of this comes directly from [Kar98] (with our own notations and proofs), except for the part dealing with strict and affine lifting. The final section 6 describes the main result (Theorem 6.1) and elaborates on the examples of fuzzy predicates and presheaves. These examples are mere illustrations which do not contribute to the general theory. Especially the last example requires some categorical sophistication. But the first sections (2 – 5) do not really require experience in category theory.

2 Preliminaries

We start with a brief overview of the notions and notations that will be used. At the end we will briefly introduce Galois algebras. They are studied further in Section 5. We do not include an introduction to coalgebra, and refer to [JR97, Rut99] instead.

Some basic constructions on sets will be used, like product, coproduct and exponent. The product of two sets X, Y will be written as $X \times Y$, with projection functions $X \xleftarrow{\pi} X \times Y \xrightarrow{\pi'} Y$. The coproduct, or disjoint union, of X, Y is $X + Y$, with coprojection (or injection) functions $X \xrightarrow{\kappa} X + Y \xleftarrow{\kappa'} Y$. And the exponent, or function space, is X^Y, with evaluation function $X^Y \times Y \xrightarrow{\mathrm{ev}} X$. The empty product is a singleton set, typically written as $1 = \{*\}$.

Posets play an important rôle in the sequel. Finite meets in a poset will be written as $\top, \wedge$, and finite joins as $\bot, \vee$. A poset X is complete if each subset $S \subseteq X$ has a join $\bigvee S \in X$. It is well-known that each subset S then also has a meet $\bigwedge S \in X$, given as $\bigvee\{x \in X \mid \forall y \in S. x \leq y\}$.

A function $f\colon X \to Y$ between posets X, Y is monotone if $x \leq x' \Rightarrow f(x) \leq f(x')$ for all $x, x' \in X$. Such a function $f\colon X \to Y$ is said to have a right (or upper) adjoint if there is a function $g\colon Y \to X$ in the reverse direction such that $f(x) \leq y \Leftrightarrow x \leq g(y)$ for all $x \in X, y \in Y$. Such a situation forms a Galois connection (or an adjunction between poset categories), and will often be denoted by $f \dashv g$. Then f is also called a left (or lower) adjoint of g. If X, Y are complete posets, then $f\colon X \to Y$ has a right adjoint if and only if f preserves all joins. The right adjoint is then $g(y) = \bigvee\{z \in X \mid f(z) \leq y\}$, see *e.g.* [Joh82, I, Theorem 4.2] or [MSS85]. Similarly, a function $g\colon Y \to X$ is a right adjoint (*i.e.* has a left adjoint) if and only if g preserves all meets.

Each monotone function $f\colon X \to X$ on a complete poset X has both a least fixed point $\mu f \in X$ and a greatest fixed point $\nu f \in X$, see *e.g.* [DP90, Chapter 4]. These can be described explicitly as:

$$\mu f = \bigwedge\{x \in X \mid f(x) \leq x\} \qquad \text{and} \qquad \nu f = \bigvee\{x \in X \mid x \leq f(x)\}.$$

A Heyting algebra is a poset X with finite meets and joins such that for each element $x \in X$, the function $x \wedge (-)\colon X \to X$ has a right adjoint $x \supset (-)$, also called implication. A complete Heyting algebra is a Heyting algebra which is complete as a poset. A complete poset is thus a complete Heyting algebra if and only if the following distributivity $x \wedge (\bigvee S) = \bigvee_{s \in S}(x \wedge s)$ holds. The canonical example of a complete Heyting algebra is the poset $\mathcal{O}(X)$ of open subsets of a topological space X.

In a Heyting algebra one can define negation $\neg x$ as $x \supset \bot$. Then $x \leq \neg\neg x$. The Heyting algebra is called a Boolean algebra if $\neg\neg x \leq x$, for all x. The canonical example of a complete Boolean algebra is the poset $\mathcal{P}(X)$ of subsets of an arbitrary set X. Such subsets $P \subseteq X$ are also called predicates on X, and membership $x \in P$ is therefore also written as $P(x)$.

2.1. Definition ([Kar98]). A Galois algebra is a complete Boolean algebra B together with a "nexttime" function $B \to B$ that preserves all meets.

The nexttime operator $B \to B$ in the definition is written as $\widetilde{\oplus}$ in [Kar98], but here we shall write it as $\underset{\rightarrow}{\bullet}$. Since this operation preserves all meets, it has a left adjoint $\underset{\leftarrow}{\bullet}$, given by $\underset{\leftarrow}{\bullet}\, y = \bigwedge\{z \in B \mid y \leq \underset{\rightarrow}{\bullet}\, z\}$, so that $\underset{\leftarrow}{\bullet}\, y \leq x \Leftrightarrow y \leq \underset{\rightarrow}{\bullet}\, x$. If $\underset{\rightarrow}{\bullet}\, x$ is 'nexttime x', then $\underset{\leftarrow}{\bullet}\, y$ is 'lasttime y'.

3 Polynomial functors on Sets

In this section we introduce a collection of special functors[1] on the category **Sets** of sets and functions, containing so-called polynomial functors. They can be extended to predicates, in what is called predicate lifting [HJ98, Jac97]. It forms a crucial technique for the construction of nexttime and lasttime operators for each coalgebra of a polynomial functor. These operators yield a Galois algebra on the complete Boolean algebra of subsets of the state space of the coalgebra. This will be illustrated in several examples.

[1] Briefly, in this context a functor T is a mapping $X \mapsto T(X)$ of sets to sets, which also works on functions, written as $(X \xrightarrow{f} Y) \mapsto (T(X) \xrightarrow{T(f)} T(Y))$, in such a way that identities and composites are preserved. How polynomial functors work on functions is "obvious" and left to the reader.

The polynomial functors are defined as the least collection of functors $\mathbf{Sets} \to \mathbf{Sets}$ containing:

(i) the identity functor $\mathbf{Sets} \to \mathbf{Sets}$, and constant functors K_A, given by $X \mapsto A$, for an arbitrary set A;

(ii) the product $X \mapsto T_1(X) \times T_2(X)$ and the coproduct $X \mapsto T_1(X) + T_2(X)$ of polynomial functors $T_1, T_2 \colon \mathbf{Sets} \to \mathbf{Sets}$;

(iii) the (constant) exponent $X \mapsto T(X)^A$, for an arbitrary set A, and the covariant powerset $X \mapsto \mathcal{P}(T(X))$, of a polynomial functor $T \colon \mathbf{Sets} \to \mathbf{Sets}$.

Typical polynomial functors are $X \mapsto 1 + (A \times X)$, where 1 is a singleton set $\{*\}$, and $X \mapsto \mathcal{P}(B \times X)$. Coalgebras $S \to 1 + (A \times S)$ give rise to finite and infinite sequences of elements of A, and coalgebras $S \to \mathcal{P}(B \times S)$ capture transition systems with labels in B.

In the sequel it is convenient to use the coproduct $\coprod_f$ and product $\prod_f$ functions $\mathcal{P}(A) \to \mathcal{P}(B)$ along a function $f \colon A \to B$. These are given by

$$\begin{aligned}
\textstyle\coprod_f(P) &= \{y \in B \mid \exists x \in A.\, f(x) = y \wedge x \in P\} \\
&= \{f(x) \mid x \in P\} \qquad \text{(the image of } P \text{ under } f) \\
\textstyle\prod_f(P) &= \{y \in B \mid \forall x \in A.\, f(x) = y \Rightarrow x \in P\}.
\end{aligned}$$

There are the standard adjunctions $\coprod_f \dashv f^* \dashv \prod_f$, where $f^* \colon \mathcal{P}(B) \to \mathcal{P}(A)$ is the "substitution" or "inverse image" function, given by $Q \mapsto \{a \in A \mid f(a) \in Q\}$. Using this notation will give a hint for the upgrade of the next result to a more general setting where similar adjunctions exist.

3.1. Definition (See [HJ98, Jac97]). Let $T \colon \mathbf{Sets} \to \mathbf{Sets}$ be a polynomial functor as described above, and let X be an arbitrary set.

(i) The predicate lifting function $(-)^T \colon \mathcal{P}(X) \to \mathcal{P}(T(X))$ is defined by induction on the structure of T. For $P \subseteq X$, one gets $P^T \subseteq T(X)$ as:

$$\begin{aligned}
P^{\mathrm{id}} &= P \\
P^{\mathrm{K}_A} &= A \\
P^{T_1 \times T_2} &= \pi^*(P^{T_1}) \cap \pi'^*(P^{T_2}) \\
&= \{(z_1, z_2) \mid z_1 \in P^{T_1} \wedge z_2 \in P^{T_2}\} \\
P^{T_1 + T_2} &= \textstyle\coprod_\kappa(P^{T_1}) \cup \coprod_{\kappa'}(P^{T_2}) \\
&= \textstyle\prod_\kappa(P^{T_1}) \cap \prod_{\kappa'}(P^{T_2}) \\
&= \{\kappa z_1 \mid z_1 \in P^{T_1}\} \cup \{\kappa' z_2 \mid z_2 \in P^{T_2}\} \\
P^{T^A} &= \textstyle\prod_\pi \mathrm{ev}^*(P^T) \qquad \text{where } \mathrm{ev} \colon T(X)^A \times A \to T(X) \\
&\qquad\qquad\qquad\quad\ \text{is the evaluation function} \\
&= \{f \mid \forall a \in A.\, f(a) \in P^T\} \\
P^{\mathcal{P}(T)} &= \textstyle\prod_\pi(\epsilon_{T(X)} \supset \pi'^*(P^T)) \qquad \text{where } \epsilon_{T(X)} \subseteq \mathcal{P}(T(X)) \times T(X) \\
&\qquad\qquad\qquad\qquad\qquad\ \text{is the membership relation} \\
&= \{U \mid \forall z \in T(X).\, z \in U \Rightarrow z \in P^T\}.
\end{aligned}$$

(ii) A left adjoint $(-)_T \colon \mathcal{P}(T(X)) \to \mathcal{P}(X)$ to $(-)^T$ can also be desribed explicitly:

$$\begin{array}{rcl}
Q_{\mathrm{id}} & = & Q \\
Q_{\mathrm{K}_A} & = & \emptyset \\
Q_{T_1 \times T_2} & = & (\coprod_{\pi}(Q))_{T_1} \cup (\coprod_{\pi'}(Q))_{T_2} \\
 & = & (\{z_1 \mid \exists z_2 . (z_1, z_2) \in Q\})_{T_1} \cup (\{z_2 \mid \exists z_1 . (z_1, z_2) \in Q\})_{T_2} \\
Q_{T_1 + T_2} & = & (\kappa^*(Q))_{T_1} \cup (\kappa'^*(Q))_{T_2} \\
 & = & (\{z_1 \mid \kappa z_1 \in Q\})_{T_1} \cup (\{z_2 \mid \kappa' z_2 \in Q\})_{T_2} \\
Q_{T^A} & = & (\coprod_{\mathrm{ev}} \pi^*(Q))_T \\
 & = & (\{f(a) \mid a \in A \wedge f \in Q\})_T \\
Q_{\mathcal{P}(T)} & = & (\coprod_{\pi'}(\epsilon_{T(X)} \wedge \pi^*(Q)))_T \\
 & = & (\bigcup Q)_T .
\end{array}$$

By induction on the structure of T one checks that $P^T \subseteq T(X)$ as described in the definition can also be defined directly as $T(P) \subseteq T(X)$, where P is considered as set itself. The above inductive definition however is convenient, because it gives a good handle on the various cases, and allows us to describe the left adjoint explicitly. It is not hard to see, again by induction on the structure of T, that $Q_T \subseteq P \Leftrightarrow Q \subseteq P^T$. Thus $(-)_T$ is indeed the left adjoint of $(-)^T$.

3.2. Lemma. *Let T be a polynomial functor, and $f\colon X \to Y$ be a function.*

(i) *For a predicate $P \subseteq Y$,*

$$(f^*(P))^T \quad = \quad T(f)^*(P^T).$$

(ii) *And for a predicate $Q \subseteq T(X)$,*

$$\coprod_f(Q_T) \quad = \quad \big(\coprod_{T(f)}(Q)\big)_T.$$

Proof. (i) By induction on the structure of T.

(ii) Because for an arbitrary predicate $P \subseteq X$,

$$\begin{array}{rcll}
\coprod_f(Q_T) \subseteq P & \Leftrightarrow & Q \subseteq (f^*(P))^T & \text{by the adjunctions} \\
 & \Leftrightarrow & Q \subseteq T(f)^*(P^T) & \text{by (i)} \\
 & \Leftrightarrow & (\coprod_{T(f)}(Q))_T \subseteq P. &
\end{array}$$

□

Next we turn to coalgebras of polynomial functors, see [JR97] for more information.

3.3. Definition. Let $T\colon \mathbf{Sets} \to \mathbf{Sets}$ be a polynomial functor, and $\alpha\colon X \to T(X)$ be a T-coalgebra, with a predicate $P \subseteq X$ on its state space (or carrier) X.

(i) Define a new predicate $\underset{\rightarrow}{\alpha}\, P \subseteq X$ as:

$$\begin{array}{rcl}
\underset{\rightarrow}{\alpha}\, P & \overset{\mathrm{def}}{=} & \alpha^*(P^T) \\
 & = & \{x \in X \mid \alpha(x) \in P^T\}.
\end{array}$$

Intuitively, $\underset{\rightarrow}{\alpha}\, P$ contains those states all whose direct successor states (*w.r.t.* α), if any, satisfy P. It is the weak nexttime operator from temporal logic. The corresponding strong nexttime operator is $\neg \underset{\rightarrow}{\alpha} \neg$, see Example 3.7.

(ii) Call $P \subseteq X$ an α-invariant if $P \subseteq \underset{\rightarrow}{\alpha} P$. This means that α maintains P.

(iii) We further write

$$\begin{array}{rcl} \underset{\leftarrow}{\alpha} P & \overset{\text{def}}{=} & \left(\coprod_\alpha(P)\right)_T \\ & = & \left(\{\alpha(x) \mid x \in P\}\right)_T. \end{array}$$

The predicate $\underset{\leftarrow}{\alpha} P$ contains those states which are direct successors of states in P. It is the strong lasttime operator (involving an existential quantifier), with the corresponding weak version given by $\neg \underset{\leftarrow}{\alpha} \neg$, see Example 3.7.

It is not hard to see that $\underset{\rightarrow}{\alpha}(-)$ and $\underset{\leftarrow}{\alpha}(-)$ form a Galois connection: $\underset{\leftarrow}{\alpha} P_1 \subseteq P_2 \Leftrightarrow P_1 \subseteq \underset{\rightarrow}{\alpha} P_2$. Hence, as an alternative formulation, a predicate P is an invariant if and only if $\underset{\leftarrow}{\alpha} P \subseteq P$.

3.4. Proposition. *For a coalgebra $X \xrightarrow{\alpha} T(X)$ of a polynomial functor $T\colon$ **Sets** $\rightarrow$ **Sets**, the set $\mathcal{P}(X)$ of predicates on its state space forms a Galois algebra, with (weak) nexttime operator $\underset{\rightarrow}{\alpha}\colon \mathcal{P}(X) \rightarrow \mathcal{P}(X)$.* □

Note that our construction of the nexttime $\underset{\rightarrow}{\cdot}$ and lasttime $\underset{\leftarrow}{\cdot}$ operations is very general, because it works for an arbitrary coalgebra of an arbitrary polynomial functor. It thus applies to all (coalgebraic) systems whose interface forms a polynomial functor.

The weak lasttime operator $\neg \underset{\leftarrow}{\alpha} \neg\colon \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ also forms a Galois algebra on $\mathcal{P}(X)$—with left adjoint $\neg \underset{\rightarrow}{\alpha} \neg$—but we shall take $\underset{\rightarrow}{\alpha}$ as basic operation. Similarly, the derived henceforth operator $\underset{\Rightarrow}{\alpha}$ gives a Galois algebra, as shown in the next result.

3.5. Definition (Least and greatest invariants). In the context of the previous definition, we write

$$\begin{array}{lll} \underset{\Rightarrow}{\alpha} P & \text{for} & \text{the greatest fixed point of } S \mapsto P \wedge \underset{\rightarrow}{\alpha} S \\ \underset{\Leftarrow}{\alpha} P & \text{for} & \text{the least fixed point of } S \mapsto P \vee \underset{\leftarrow}{\alpha} S. \end{array}$$

One reads $\underset{\Rightarrow}{\alpha} P$ as "henceforth P", *i.e.* as: P holds now and in all successor states (*w.r.t.* α). Similarly, one can read $\underset{\Leftarrow}{\alpha} P$ as "P sometime earlier", *i.e.* as: P holds now or at some predecessor state.

It is easy to see that $\underset{\Rightarrow}{\alpha} P$ is the greatest invariant contained in P, and that $\underset{\Leftarrow}{\alpha} P$ is the least invariant containing P. The latter predicate contains all states which are reachable from P. By construction we have a new Galois connection: $\underset{\Leftarrow}{\alpha} P_1 \subseteq P_2 \Leftrightarrow P_1 \subseteq \underset{\Rightarrow}{\alpha} P_2$.

Alternative notation for $\underset{\rightarrow}{\alpha} P$ is $\mathsf{X}_\alpha P$ [Eme90], or $\mathsf{O}_\alpha P$ [Gol92], or $[\alpha]P$ [KT90] (in the style of dynamic logic). Similarly, one may write $\underset{\Rightarrow}{\alpha} P$ as $\mathsf{G}_\alpha P$ or as $\Box_\alpha P$. And $\neg \underset{\Rightarrow}{\alpha} \neg P$ is also written as $\mathsf{F}_\alpha P$, or $\Diamond_\alpha P$.

Before presenting examples, we consider the interaction of the future modalities with homomorphisms of coalgebras.

3.6. Lemma. *Consider two coalgebras $X \xrightarrow{\alpha} T(X)$ and $Y \xrightarrow{\beta} T(Y)$ of a polynomial functor T, with a homomorphism $f\colon X \to Y$ between them—so that $\beta \circ f = T(f) \circ \alpha$. Then,*

(i) $f^{*}(\underset{\rightarrow}{\beta}\, P) = \underset{\rightarrow}{\alpha}\, f^{*}(P)$;

(ii) *Q is an α-invariant* $\Rightarrow \coprod_f(Q)$ *is a β-invariant;*

(iii) $f^{*}(\underset{\Rightarrow}{\beta}\, P) = \underset{\Rightarrow}{\alpha}\, f^{*}(P)$.

Proof. (i) Since

$$\begin{array}{rcll}
f^{*}(\underset{\rightarrow}{\beta}\, P) & = & f^{*}\beta^{*}(P^{T}) & \\
& = & (\beta \circ f)^{*}(P^{T}) & \\
& = & (T(f) \circ \alpha)^{*}(P^{T}) & \text{because } f \text{ is a homomorphism} \\
& = & \alpha^{*}(T(f)^{*}(P^{T})) & \\
& = & \alpha^{*}((f^{*}(P))^{T}) & \text{by Lemma 3.2 (i)} \\
& = & \underset{\rightarrow}{\alpha}\, f^{*}(P). &
\end{array}$$

(ii) Assume Q is an α-invariant, *i.e.* $(\coprod_\alpha(Q))_T = \underset{\leftarrow}{\alpha}\, Q \subseteq Q$. Then $\coprod_f(Q)$ is a β-invariant, because:

$$\begin{array}{rcll}
\underset{\leftarrow}{\beta}\,(\coprod_f(Q)) & = & (\coprod_\beta(\coprod_f(Q)))_T & \\
& = & (\coprod_{\beta \circ f}(Q))_T & \\
& = & (\coprod_{T(f) \circ \alpha}(Q))_T & \\
& = & (\coprod_{T(f)}(\coprod_\alpha(Q)))_T & \\
& = & \coprod_f((\coprod_\alpha(Q))_T) & \text{by Lemma 3.2 (ii)} \\
& \subseteq & \coprod_f(Q) & \text{since } Q \text{ is an } \alpha\text{-invariant.}
\end{array}$$

(iii) By construction, $\underset{\Rightarrow}{\alpha}\, f^{*}(P)$ is the greatest α-invariant contained in $f^{*}(P)$. We show that $f^{*}(\underset{\Rightarrow}{\beta}\, P)$ also satisfies this characterisation.

(a) $f^{*}(\underset{\Rightarrow}{\beta}\, P)$ is contained in $f^{*}(P)$ since $\underset{\Rightarrow}{\beta}\, P$ is contained in P.

(b) $f^{*}(\underset{\Rightarrow}{\beta}\, P)$ is an α-invariant, by (i): $f^{*}(\underset{\Rightarrow}{\beta}\, P) \subseteq f^{*}(\underset{\rightarrow}{\beta}\, \underset{\Rightarrow}{\beta}\, P) = \underset{\rightarrow}{\alpha}\, f^{*}(\underset{\Rightarrow}{\beta}\, P)$.

(c) $f^{*}(\underset{\Rightarrow}{\beta}\, P)$ is the greatest α-invariant contained in $f^{*}(P)$: if $Q \subseteq f^{*}(P)$ is an α-invariant, then $\coprod_f(Q) \subseteq P$ is a β-invariant by (ii). Hence $\coprod_f(Q) \subseteq \underset{\Rightarrow}{\beta}\, P$, and thus $Q \subseteq f^{*}(\underset{\Rightarrow}{\beta}\, P)$. □

3.7. Example. First we investigate the temporal operators associated with the functors for infinite sequences and also for transition systems. These give the familiar operators of Linear Temporal Logic (LTL) and Computation Tree Logic (CTL), see [Eme90, MP92]. Then, in the last point, we sketch an example of temporal operators in coalgebraic specification.

(i) Consider the functor $T(X) = B \times X$ on the category **Sets**, for an arbitary set B. For predicates $P \subseteq X$ and $Q \subseteq T(X)$ we have, following Definition 3.1,

$$P^{T} = \{(b, x) \in T(X) \mid b \in B \wedge x \in P\} \quad \text{and} \quad Q_T = \{x \in X \mid \exists b \in B.\, (b, x) \in Q\}$$

For an arbitary coalgebra $\gamma\colon X \to T(X)$ we get

$$\begin{array}{rcl} \underset{\rightarrow}{\gamma} P & = & \{x \in X \mid \gamma(x) \in P^T\} \\ & = & \{x \in X \mid \pi'\gamma(x) \in P\} \end{array} \qquad \text{and} \qquad \begin{array}{rcl} \underset{\leftarrow}{\gamma} P & = & \big(\{\gamma(x) \mid x \in P\}\big)_T \\ & = & \{\pi'\gamma(x) \mid x \in P\}. \end{array}$$

So that $P \subseteq X$ is an invariant if and only if $x \in P \Rightarrow \pi'\gamma(x) \in P$, for all $x \in X$. It is easy to see that the greatest invariant $\underset{\Rightarrow}{\gamma} P$ contained in P is $\{x \mid \forall m.\, (\pi'\gamma)^m(x) \in P\}$. Similarly, the least invariant $\underset{\Leftarrow}{\gamma} P$ containing P is $\{(\pi'\gamma)^m(x) \mid x \in P, m \in \mathbb{N}\}$.

We now consider a concrete instantiation: we take $B = \{0,1\}$ and write B^ω for the set of bit streams, consisting of infinite sequences $(b_n)_{n\in\mathbb{N}}$ of bits $b_n \in \{0,1\}$. We consider B^ω with the (terminal) T-coalgebra $\beta = \langle \mathsf{hd}, \mathsf{tl}\rangle\colon B^\omega \to B \times B^\omega$ consisting of head and tail function given by

$$\mathsf{hd}((b_n)) = b_0 \qquad \text{and} \qquad \mathsf{tl}((b_n)) = (b_{n+1}).$$

Let $P \subseteq B^\omega$ be the predicate given by $(b_n) \in P \Leftrightarrow b_{10} = 0$. Then, for example,

$$\begin{array}{rcl} (b_n) \in \underset{\rightarrow}{\beta} P & \Leftrightarrow & (b_{n+1}) \in P \\ & \Leftrightarrow & b_{11} = 0 \\ (b_n) \in \underset{\leftarrow}{\beta} P & \Leftrightarrow & \exists (a_n) \in P.\, (a_{n+1}) = (b_n) \\ & \Leftrightarrow & b_9 = 0 \\ (b_n) \in \neg \underset{\rightarrow}{\beta} \neg P & \Leftrightarrow & \neg((b_{n+1}) \in \neg P) \\ & \Leftrightarrow & (b_{n+1}) \in P \\ & \Leftrightarrow & b_{11} = 0. \end{array}$$

This shows that $\underset{\leftarrow}{\beta} \neq \neg \underset{\rightarrow}{\beta} \neg$. But $\underset{\rightarrow}{\beta}$ coincides with $\neg \underset{\rightarrow}{\beta} \neg$ due to a special property of the interface functor T, namely that it has an "affine" lifting, as investigated in the second part of Section 5.

Further,

$$\begin{array}{rcl} (b_n) \in \underset{\Rightarrow}{\beta} P & \Leftrightarrow & \forall m.\, \mathsf{tl}^m((b_n)) \in P \\ & \Leftrightarrow & \forall m.\, (b_{n+m}) \in P \\ & \Leftrightarrow & \forall n \geq 10.\, b_n = 0 \\ (b_n) \in \neg \underset{\Rightarrow}{\beta} \neg P & \Leftrightarrow & \neg \forall m.\, \mathsf{tl}^m((b_n)) \notin P \\ & \Leftrightarrow & \exists m.\, (b_{n+m}) \in P \\ & \Leftrightarrow & \exists n \geq 10.\, b_n = 0 \\ (b_n) \in \underset{\Leftarrow}{\beta} P & \Leftrightarrow & \exists (a_n).\, \exists m.\, (a_n) \in P \wedge \mathsf{tl}^m((a_n)) = (b_n) \\ & \Leftrightarrow & \exists (a_n).\, \exists m.\, a_{10} = 0 \wedge (a_{n+m}) = (b_n) \\ & \Leftrightarrow & \mathsf{true} \\ (b_n) \in \neg \underset{\Leftarrow}{\beta} \neg P & \Leftrightarrow & \neg \exists (a_n).\, \exists m.\, (a_n) \notin P \wedge \mathsf{tl}^m((a_n)) = (b_n) \\ & \Leftrightarrow & \forall (a_n).\, \forall m.\, ((a_{n+m}) = (b_n) \Rightarrow a_{10} = 0) \\ & \Leftrightarrow & \forall n \leq 10.\, b_n = 0. \end{array}$$

Next consider the T-coalgebra $\alpha = \langle \mathsf{hd}, \mathsf{tl} \circ \mathsf{tl}\rangle\colon B^\omega \to B \times B^\omega$, together with the function $\mathsf{evens}\colon B^\omega \to B^\omega$ given by $(b_n) \mapsto (b_{2n})$. Then evens is a homomorphism from α to β.

For the above predicate P we have:

$$\begin{array}{rcl}
(b_n) \in \mathsf{evens}^*(P) & \Leftrightarrow & (b_{2n}) \in P \\
& \Leftrightarrow & b_{20} = 0 \\
(b_n) \in \underset{\leftarrow}{\alpha}\, \mathsf{evens}^*(P) & \Leftrightarrow & \exists (a_n) \in \mathsf{evens}^*(P).\, (a_{n+1}) = (b_n) \\
& \Leftrightarrow & b_{19} = 0 \\
(b_n) \in \mathsf{evens}^*(\underset{\leftarrow}{\beta}\, P) & \Leftrightarrow & (b_{2n}) \in \underset{\leftarrow}{\beta}\, P \\
& \Leftrightarrow & b_{18} = 0
\end{array}$$

This shows that Lemma 3.6 (i) does not hold for $\underset{\leftarrow}{\cdot}$ instead of $\underset{\rightarrow}{\cdot}$. This point is stressed in [GM88].

(ii) In order to see the difference between the weak and strong versions of the nexttime $\underset{\rightarrow}{\cdot}$ and lasttime $\underset{\leftarrow}{\cdot}$ operators we take a look at the functor $T(X) = \mathcal{P}(A \times X)$ on **Sets**, which forms an interface for transition systems with labels from A. For predicates $P \subseteq X$ we have, following Definition 3.1,

$$\begin{array}{rcl}
P^T & = & P^{\mathcal{P}(\mathrm{K}_A \times id)} \\
& = & \{U \in T(X) \mid \forall z \in U.\, z \in P^{\mathrm{K}_A \times id}\} \\
& = & \{U \in T(X) \mid \forall z \in U.\, \pi z \in P^{\mathrm{K}_A} \wedge \pi' z \in P^{id}\} \\
& = & \{U \in T(X) \mid \forall (a, x) \in U.\, a \in A \wedge x \in P\} \\
& = & \{U \in T(X) \mid \forall (a, x) \in U.\, x \in P\}.
\end{array}$$

Similarly, for $Q \subseteq \mathcal{P}(A \times X)$,

$$Q_T \quad = \quad \{x \in X \mid \exists a \in A.\, \exists U \in Q.\, (a, x) \in U\}$$

Consider now a T-coalgebra $\alpha\colon X \to \mathcal{P}(A \times X)$. For elements $x, x' \in X$ and $a \in A$ one often writes $x \xrightarrow{a} x'$ for $(a, x') \in \alpha(x)$, and one says that x can do an a-step to x'. The associated nexttime and lasttime operators are, on $P \subseteq X$,

$$\begin{array}{rcll}
\underset{\rightarrow}{\alpha}\, P & = & \{x \in X \mid \alpha(x) \in P^T\} & \\
& = & \{x \in X \mid \forall x' \in X.\, \forall a \in A.\, x \xrightarrow{a} x' \Rightarrow x' \in P\} & \text{(weak nexttime)} \\
\underset{\leftarrow}{\alpha}\, P & = & \big(\{\alpha(x) \mid x \in P\}\big)_T & \\
& = & \{x' \in X \mid \exists x \in X.\, \exists a \in A.\, x \xrightarrow{a} x' \wedge x \in P\} & \text{(strong lasttime)} \\
\neg \underset{\rightarrow}{\alpha} \neg P & = & \{x \in X \mid \exists x' \in X.\, \exists a \in A.\, x \xrightarrow{a} x' \wedge x' \in P\} & \text{(strong nexttime)} \\
\neg \underset{\leftarrow}{\alpha} \neg P & = & \{x' \in X \mid \forall x \in X.\, \forall a \in A.\, x \xrightarrow{a} x' \Rightarrow x \in P\} & \text{(weak lasttime)}
\end{array}$$

(iii) In coalgebraic specification (see [Rei95, Jac96]) one specifies object-oriented systems via coalgebraic operations and initial states, satisfying certain assertions. Typically in these assertions, one uses bisimilarity $\underline{\leftrightarrow}$ instead of equality $=$ on states. We shall present an example of a coalgebraic specification of a stack, using some (hopefully) self-explanatory notation.

```
Stack[$A$ : TYPE] : CLASSSPEC
  METHOD
    size: Self $\longrightarrow \mathbb{Z}$
    push: Self $\times A \longrightarrow$ Self
    pop: Self $\longrightarrow 1 + (A \times \mathsf{Self})$
  ASSERTION
    $\forall x \in \mathsf{Self}.\, \forall a \in A.\, \mathsf{size}(\mathsf{push}(x,a)) = \mathsf{size}(x) + 1$
    $\forall x \in \mathsf{Self}.$ CASES $\mathsf{pop}(x)$ OF
              $\kappa u$ : $\mathsf{size}(x) = 0$
              $\kappa' v$ : $\mathsf{size}(\pi v) = \mathsf{size}(x) - 1$
            ENDCASES
    $\forall x \in \mathsf{Self}.\, \forall a \in A.$ CASES $\mathsf{pop}(\mathsf{push}(x,a))$ OF
                      $\kappa u$ : false
                      $\kappa' v$ : $\pi v = a \wedge \pi' v \,\underline{\leftrightarrow}\, x$
                    ENDCASES
  CONSTRUCTOR
    new_stack: Self
  CREATION
    $\mathsf{pop}(\mathsf{new\_stack}) = \kappa *$
```

Notice that the last assertion says that $\mathsf{pop}(\mathsf{push}(x,a))$ is always of the form $\kappa'(a,y)$ with y bisimilar to x. This says that after a push, the pop operation returns and removes the most recently pushed element from the stack (leaving a stack which is indistinguishable from the stack before the push). The pop method can also fail, by returning $\kappa *$ in the 1-component of its codomain $1 + (A \times \mathsf{Self})$. This indicates that the stack is empty.

The first thing to note is that the interface of the operations is captured by a polynomial functor, namely:

$$T(X) = \mathbb{Z} \times X^A \times (1 + (A \times X))$$

A coalgebra $X \to T(X)$ of this functor combines the three methods size, push, pop in a single function. For a predicate $P \subseteq X$, we get the predicate lifting $P^T \subseteq T(X)$ consisting of:

$$P^T = \{(n, f, z) \mid \forall a \in A.\, P(f(a)) \wedge \forall a \in A.\, \forall y \in X.\, z = \kappa'(a,y) \Rightarrow P(y)\}.$$

Thus, for a coalgebra $c = \langle \mathsf{size}, \mathsf{push}, \mathsf{pop} \rangle$ the predicate $\underset{\rightarrow}{c}P$ for 'nexttime P' is defined by:

$$\begin{aligned}
\underset{\rightarrow}{c}P(x) \quad &\Leftrightarrow \quad P^T(c(x)) \\
&\Leftrightarrow \quad \forall a \in A.\, P(\mathsf{push}(x,a)) \wedge \forall a \in A.\, \forall y \in X.\, \mathsf{pop}(x) = \kappa'(a,y) \Rightarrow P(y).
\end{aligned}$$

It holds for x if P holds at each (immediate) successor state, obtained by either push or pop.

In order to say something interesting about this stack, we use the following auxiliary iterate function. For arbitrary $f\colon X \to 1 + (A \times X)$ and $n \in \mathbb{N}$, we introduce

$$X \xrightarrow{\mathsf{iterate}(f,n)} 1 + (A \times X)$$

as:

$$\begin{array}{rcl}
\mathsf{iterate}(f, 0)(x) & = & f(x) \\
\mathsf{iterate}(f, n+1)(x) & = & \text{CASES } f(x) \text{ OF} \\
 & & \quad \kappa u : \kappa * \\
 & & \quad \kappa' v : \mathsf{iterate}(f, n)(\pi' v) \\
 & & \text{ENDCASES}
\end{array}$$

Consider now the following specific predicate Q.

$$Q(x) \Leftrightarrow \mathsf{size}(x) \geq 0 \wedge \mathsf{iterate}(\mathsf{pop}, \mathsf{size}(x)) = \kappa * .$$

We claim that for each T-coalgebra $c = \langle \mathsf{size}, \mathsf{push}, \mathsf{pop} \rangle$ satisfying the assertions in the class specification Stack, the predicate Q is an invariant, *i.e.* $Q \subseteq \underset{\rightarrow}{c}\, Q$. Then, writing $\Box(c)$ for the weak henceforth operator $\underset{\Rightarrow}{c}$ and $\Diamond(c)$ for strong one $\neg \underset{\Rightarrow}{c} \neg$, we can prove:

$$\Box(c)\big(\lambda x.\, \mathsf{size}(x) = 0 \Rightarrow \mathsf{pop}(x) = \kappa * \big)(\mathsf{new\_stack}). \tag{1}$$

This says that for all reachable states, pop fails if (and only if) the size is zero. For the proof we use that there is an invariant P (namely Q) with $P(x) \Rightarrow (\mathsf{size}(x) = 0 \Rightarrow \mathsf{pop}(x) = \kappa *)$ and $P(\mathsf{new\_stack})$. This proves the result because $\Box$ yields the greatest invariant.

Similarly, one has:

$$\Box(c)\Big(\lambda x.\, \mathsf{size}(x) \geq 0 \wedge \Diamond(c)\big(\lambda y.\, \mathsf{size}(y) = 0\big)(x)\Big)(\mathsf{new\_stack}). \tag{2}$$

This statement says that for all reachable states x the size is positive, and for some future state y of x the size is zero. It can be proved via a slightly stronger invariant Q' given as

$$Q'(x) \Leftrightarrow Q(x) \wedge \forall n \in \mathbb{N}.\, n < \mathsf{size}(x) \Rightarrow \mathsf{iterate}(\mathsf{pop}, n) \neq \kappa * .$$

The tool described in [HHJT98] translates class specifications as above into logical theories for a back-end proof tool (like PVS [ORSvH95] or Isabelle [Pau94]). It extracts the interface functor from a class specification, generates the associated lifting, and thereby also the definitions of invariant and bisimulation. Additionally, it generates tailor-made $\Box(c) = \underset{\Rightarrow}{c}$ and $\Diamond(c) = \neg \underset{\Rightarrow}{c} \neg$ operators for the class/coalgebra c. This allows us to formulate and prove results like above in the back-end proof tool. Actually, the above statements (1) and (2) have been proved in PVS.

3.8. Remark. In the end one can ask what is so special about polynomial functors to make the construction of Galois algebras work, for example, in order to generalise the approach. For an arbitrary functor $T : \mathbb{B} \to \mathbb{B}$ one can require that T preserves arbitrary[2] weak pullbacks. The structure one needs in $\mathbb{B}$ is that pullbacks of monos exist and that the posets $\mathrm{Sub}(X)$ of objects $X \in \mathbb{B}$ are complete Boolean (or Heyting, see Defintion 5.5) algebras and that the induced substitution (pullback) functors f^* preserve meets (or equivalently have left adjoints $\coprod_f$). One can then define predicate lifting $(-)^T : \mathrm{Sub}(X) \to \mathrm{Sub}(T(X))$ simply by $(A \rightarrowtail X) \mapsto (T(A) \rightarrowtail T(X))$.

[2] This means weak pullbacks of arbitrary set-indexed collections of morphisms with a common codomain. This is an essentially stronger requirement than preservation of weak pullbacks of just two morphisms, as shown in [Gum99].

This operation preserves all meets (because T preserves arbitrary weak pullbacks), and thus has a left adjoint, say $(-)_T$. In this way one gets nexttime $\alpha^*(P^T)$ and lasttime $(\coprod_\alpha(P))_T$ operations for a coalgebra $\alpha\colon X \to T(X)$ as above, and thus a Galois algebra on the poset $\mathrm{Sub}(X)$ of subobjects of the state space X.

Allthough this generalises the construction of Galois algebras for polynomial functors, it does not cover examples like in Subsection 6.1 where fuzzy predicates are used, which are not subobjects. There, the more general notion of indexed category or fibration (see [Jac99]) is needed. This goes beyond the scope of the present paper.

We have concentrated on polynomial functors because they include many important examples, and because their lifting (with left adjoint) can be described by induction on the structure of the functor. This makes it possible to mechanise the lifting, and generate appropriate notions of invariant, and thereby nexttime and lasttime operators, as described in Example 3.7 (iii).

4 Galois algebras from nexttime and lasttime along paths

Sofar we have seen the nexttime operator $\underset{\rightarrow}{\alpha} P$, containing those states *all* of whose successor states (*w.r.t.* α) satisfy P. Consider for example, a coalgebra $\alpha\colon X \to X \times X$ of the functor $T(X) = X \times X$. Then $\underset{\rightarrow}{\alpha} P(x)$ means that both $P(\pi\alpha(x))$ and $P(\pi'\alpha(x))$ hold. In this section we shall introduce nexttime and lasttime operators with respect to paths, so that we can use "nexttime P along the first path", holding on x if $P(\pi\alpha(x))$. Such operators have been studied previously in [Mos99, Röß99b, Röß99a, Kur98]. Here we introduce them in a slightly different manner, via operations on predicates (like in Definition 3.1), and we show that they also give rise to Galois algebras.

For a polynomial functor T, let $\mathsf{Inputs}(T)$ be the set of sets A occuring as exponent $(-)^A$ in T. Thus, $\mathsf{Inputs}(\mathrm{id}) = \mathsf{Inputs}(\mathrm{K}_A) = \emptyset$, $\mathsf{Inputs}(T_1 \times T_1) = \mathsf{Inputs}(T_1 + T_2) = \mathsf{Inputs}(T_1) \cup \mathsf{Inputs}(T_2)$, $\mathsf{Inputs}(T^A) = \{A\} \cup \mathsf{Inputs}(T)$, and $\mathsf{Inputs}(\mathcal{P}T) = \mathsf{Inputs}(T)$.

The set $\mathsf{PathSymbols}(T)$ contains the symbols out of which paths will be built. It is defined as $\{\pi, \pi', \kappa, \kappa'\} \cup \bigcup_{A \in \mathsf{Inputs}(T)} \{\mathrm{ev}(a) \mid a \in A\}$.

Next, the set $\mathsf{Paths}(T)$ is the subset of the set $\mathsf{PathSymbols}(T)^\star$ of lists of path symbols that is defined as follows.

$$\begin{array}{rcll}
\mathsf{Paths}(\mathrm{id}) & = & \{\langle\rangle\} & \quad \text{(where } \langle\rangle \text{ is the empty list)} \\
\mathsf{Paths}(\mathrm{K}_A) & = & \emptyset & \\
\mathsf{Paths}(T_1 \times T_2) & = & \{\pi \cdot p \mid p \in \mathsf{Paths}(T_1)\} \cup \{\pi' \cdot p \mid p \in \mathsf{Paths}(T_2)\} & \\
\mathsf{Paths}(T_1 + T_2) & = & \{\kappa \cdot p \mid p \in \mathsf{Paths}(T_1)\} \cup \{\kappa' \cdot p \mid p \in \mathsf{Paths}(T_2)\} & \\
\mathsf{Paths}(T^A) & = & \{\mathrm{ev}(a) \cdot p \mid a \in A, p \in \mathsf{Paths}(T)\} & \\
\mathsf{Paths}(\mathcal{P}T) & = & \mathsf{Paths}(T). &
\end{array}$$

where we have used $\cdot$ as shorthand for the cons operation which adds an element to a list.

For a set X with predicates $P \subseteq X$ and $Q \subseteq T(X)$, we define for a path

$p \in \mathsf{Paths}(T)$ new subsets $P^p \subseteq T(X)$ and $Q_p \subseteq X$ by induction on T:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
P^{\langle\rangle} & = & P & Q_{\langle\rangle} & = & Q \\
P^{\pi\cdot p} & = & \pi^*(P^p) & Q_{\pi\cdot p} & = & (\coprod_{\pi}(Q))_p \\
P^{\pi'\cdot p} & = & \pi'^*(P^p) & Q_{\pi'\cdot p} & = & (\coprod_{\pi'}(Q))_p \\
P^{\kappa\cdot p} & = & \prod_{\kappa}(P^p) & Q_{\kappa\cdot p} & = & (\kappa^*(Q))_p \\
P^{\kappa'\cdot p} & = & \prod_{\kappa'}(P^p) & Q_{\kappa'\cdot p} & = & (\kappa'^*(Q))_p \\
P^{\mathrm{ev}(a)\cdot p} & = & \{f \mid f(a) \in P^p\} & Q_{\mathrm{ev}(a)\cdot p} & = & (\{f(a) \mid f \in Q\})_p \\
P^p & = & \{U \mid \forall u \in U.\, u \in P^p\} & Q_p & = & (\bigcup Q)_p.
\end{array}$$

Just like there is a Galois connection $(-)_T \dashv (-)^T$ for the lifting of the previous section, there is a Galois connection for the lifting with respect to paths. The proof is by induction on the length of paths.

4.1. Lemma. *For a path p of a polynomial functor, the above operation $(-)_p$ is left adjoint to $(-)^p$, that is, $Q_p \subseteq P \Leftrightarrow Q \subseteq P^p$.* □

4.2. Definition. Consider a coalgebra $\alpha\colon X \to T(X)$ of a polynomial functor T. For a path $p \in \mathsf{Paths}(T)$ we define two operators $\mathcal{P}(X) \to \mathcal{P}(X)$, namely $\underrightarrow{p\alpha}$ for 'nexttime along p' and $\underleftarrow{p\alpha}$ for 'lasttime along p' (*w.r.t.* α). They are defined on $P \subseteq X$ as

$$\underrightarrow{p\alpha}P \stackrel{\mathrm{def}}{=} \alpha^*(P^p) \qquad \text{and} \qquad \underleftarrow{p\alpha}P \stackrel{\mathrm{def}}{=} (\coprod_{\alpha}(P))_p.$$

4.3. Example. Fix a set A, and consider the (polynomial) functor

$$T(X) = A + ((A \times X) + (A \times (X \times X)))$$

describing an interface for finite and infinite A-labeled trees with at each node either only a label, or a label with one successor tree, or a label with two successor trees.

The set $\mathsf{Paths}(T)$ contains the following three elements.

$$\left\{\begin{array}{l} \kappa'\kappa\pi' \\ \kappa'\kappa'\pi'\pi \\ \kappa'\kappa'\pi'\pi' \end{array}\right. \qquad \text{which will be written as} \qquad \left\{\begin{array}{l} \mathsf{sub} = \kappa'\kappa\pi' \\ \mathsf{leftsub} = \kappa'\kappa'\pi'\pi \\ \mathsf{rightsub} = \kappa'\kappa'\pi'\pi' \end{array}\right.$$

These paths point to specific state positions in the interface functor:

$$T(X) \;=\; A \;+\; ((A \;\times\; \underset{\overset{\uparrow}{\mathsf{sub}}}{X}) + \;(A \;\times (\overset{\overset{\mathsf{leftsub}}{\downarrow}}{X} \;\times\; \underset{\overset{\uparrow}{\mathsf{rightsub}}}{X})))$$

For predicates $P \subseteq X$ and $Q \subseteq T(X)$, the associated liftings are:

$$\begin{array}{rcl}
P^{\mathsf{sub}} & = & \{z \in T(X) \mid \forall x \in X.\, \forall a \in A.\, z = \kappa'(\kappa(a, x)) \Rightarrow P(x)\} \\
Q_{\mathsf{sub}} & = & \{x \in X \mid \exists a \in A.\, \kappa'(\kappa(a, x)) \in Q\} \\
P^{\mathsf{leftsub}} & = & \{z \in T(X) \mid \forall x_1, x_2 \in X.\, \forall a \in A.\, z = \kappa'(\kappa'(a, (x_1, x_2))) \Rightarrow P(x_1)\} \\
Q_{\mathsf{leftsub}} & = & \{x \in X \mid \exists a \in A.\, \exists y \in X.\, \kappa'(\kappa'(a, (x, y))) \in Q\} \\
P^{\mathsf{rightsub}} & = & \{z \in T(X) \mid \forall x_1, x_2 \in X.\, \forall a \in A.\, z = \kappa'(\kappa'(a, (x_1, x_2))) \Rightarrow P(x_2)\} \\
Q_{\mathsf{rightsub}} & = & \{x \in X \mid \exists a \in A.\, \exists y \in X.\, \kappa'(\kappa'(a, (y, x))) \in Q\}
\end{array}$$

Now assume we have a coalgebra $\alpha\colon X \to T(X)$. It gives for each tree $x \in X$ the label and successors (if any) of x. Then, for example,

$$\begin{array}{rcl}
x \in \underrightarrow{\mathsf{leftsub}\alpha}P & \Leftrightarrow & x \in \alpha^*(P^{\mathsf{leftsub}}) \\
 & \Leftrightarrow & \forall x_1, x_2 \in X.\, \forall a \in A.\, \alpha(x) = \kappa'(\kappa'(a, (x_1, x_2)) \Rightarrow P(x_1) \\
 & \Leftrightarrow & \text{if } x \text{ has two successor trees, then } P \text{ holds for the left one} \\
x \in \underleftarrow{\mathsf{leftsub}\alpha}P & \Leftrightarrow & x \in (\coprod_{\alpha}(P))_{\mathsf{leftsub}} \\
 & \Leftrightarrow & \exists a \in A.\, \exists x_1, x_2 \in X.\, \alpha(x_1) = \kappa'(\kappa'(a, (x, x_2))) \wedge x_1 \in Q \\
 & \Leftrightarrow & x \text{ is a left successor of a tree satisfying } P.
\end{array}$$

Also the path modalities form a Galois algebra.

4.4. Lemma. *For a coalgebra $\alpha\colon X \to T(X)$ of a polynomial functor T and a path $p \in \mathsf{Paths}(T)$ one has $\underleftarrow{p\alpha}Q \subseteq P \Leftrightarrow Q \subseteq \underrightarrow{p\alpha}P$, so that $\mathcal{P}(X)$ with $\underrightarrow{p\alpha}$ is a Galois algebra.* □

For these path modalities $\underrightarrow{p\alpha}$ and $\underleftarrow{p\alpha}$ one can define corresponding strong $\neg\underrightarrow{p\alpha}\neg$ and weak $\neg\underleftarrow{p\alpha}\neg$ versions. But also:

4.5. Lemma. *For a coalgebra $\alpha\colon X \to T(X)$ and a path $p \in \mathsf{Paths}(T)$ we define new operators $\underset{\Longrightarrow}{p\alpha}$ and $\underset{\Longleftarrow}{p\alpha}$ with type $\mathcal{P}(X) \to \mathcal{P}(X)$ as fixed points: for $P \subseteq X$,*

$$\begin{array}{lll}
\underset{\Longrightarrow}{p\alpha}P & \textit{is} & \textit{the greatest fixed point of } S \mapsto P \wedge \underrightarrow{p\alpha}S \\
\underset{\Longrightarrow}{p\alpha}P & \textit{is} & \textit{the least fixed point of } S \mapsto P \vee \underleftarrow{p\alpha}S
\end{array}$$

Then $\underset{\Longleftarrow}{p\alpha} \dashv \underset{\Longrightarrow}{p\alpha}$, so that $\mathcal{P}(X)$ with $\underset{\Longrightarrow}{p\alpha}$ is also a Galois algebra (for each path p). □

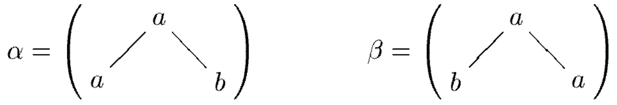
To conclude this section on operators along paths, we briefly discuss logics. For a polynomial functor $T\colon \mathbf{Sets} \to \mathbf{Sets}$ one can define two logics, differing in the modal operators that they have. In one "single-modal" logic $\mathcal{L}(T)_s$ there is only a single nexttime operator $\varphi \mapsto \underrightarrow{\bullet}\varphi$ and in the other "multi-modal" logic $\mathcal{L}(T)_m$ there is a nexttime operator $\varphi \mapsto \underrightarrow{p}\varphi$ for each path p of T. The atomic propositions of both logics are given by the elements $a \in T(1)$, where 1 is a singleton set $\{*\}$. And the propositional connectives are negation $\neg$ and conjunction $\wedge$ (say). Each proposition φ of $\mathcal{L}(T)_s$ or $\mathcal{L}(T)_m$ gives rise to a subset $[\![\,\varphi\,]\!]^{\alpha} \subseteq X$ in the following way.

$$\begin{array}{rcl}
[\![a]\!]^{\alpha} & = & \{x \in X \mid (T(!) \circ \alpha)(x) = a\} \\
 & & \qquad \text{(where } a \in T(1)\text{, and ! is the unique function } X \to 1) \\
[\![\neg\varphi]\!]^{\alpha} & = & \{x \in X \mid x \notin [\![\varphi]\!]^{\alpha}\} \\
[\![\varphi_1 \wedge \varphi_2]\!]^{\alpha} & = & [\![\varphi_1]\!]^{\alpha} \cap [\![\varphi_2]\!]^{\alpha} \\
[\![\underset{\rightarrow}{\bullet}\varphi]\!]^{\alpha} & = & \underset{\rightarrow}{\alpha}[\![\varphi]\!]^{\alpha} \qquad \text{for } \varphi \in \mathcal{L}(T)_s \\
[\![\underset{\rightarrow}{p}\varphi]\!]^{\alpha} & = & \underset{\rightarrow}{p\alpha}[\![\varphi]\!]^{\alpha} \qquad \text{for } \varphi \in \mathcal{L}(T)_m
\end{array}$$

Alternative notation for validity of φ at x, *i.e.* for $[\![\varphi]\!]^{\alpha}(x)$, is $\alpha, x \models \varphi$.

An important result of [Röß99a] is that for two elements $x \in X, y \in Y$ in the state spaces of coalgebras $\alpha: X \to T(X)$ and $\beta: Y \to T(Y)$ are bisimilar if and only if they satisfy the same formulas from the multi-modal logic $\mathcal{L}(T)_m$. This result is proved under certain restrictions: the sets A whose constant functors K_A occur in T should all be finite, and the powerset functor should not occur in T. The result is proved[3] via a terminal coalgebra construction out of maximally consistent sets of formulas. Similar such characterisation results occur in [Mos99, Röß99b, Kur98].

This logical characterisation of bisimilarity does not hold for the single-modal logic $\mathcal{L}(T)_s$. The following two coalgebras present a counter example, for the tree functor $T(X) = A + ((A \times X) + (A \times (X \times X)))$ from Example 4.3. We take $A = \{a, b\}$ and $X = \{0, 1, 2\}$ with the following two coalgebras $\alpha, \beta: X \to T(X)$.



$$\begin{array}{ll}
\alpha(0) = \kappa'(\kappa'(a, (1, 2))) & \qquad \beta(0) = \kappa'(\kappa'(a, (1, 2))) \\
\alpha(1) = \kappa a, \ \alpha(2) = \kappa b & \qquad \beta(1) = \kappa b, \ \beta(2) = \kappa a
\end{array}$$

Clearly, $0 \; {}_{\alpha}\!\underline{\leftrightarrow}_{\beta} \; 0$ does not hold, where ${}_{\alpha}\!\underline{\leftrightarrow}_{\beta}$ denotes bisimilarity *w.r.t.* α and β. But the same propositions from the single modal logic $\mathcal{L}(T)_s$ hold at 0: the same atomic propositions hold there, and, for a predicate $P \subseteq X$, the nextstep operators $\underset{\rightarrow}{\alpha}$ and $\underset{\rightarrow}{\beta}$ say that some observation can be made at both successor trees at the same time. But since the labels are different, there are no such observations.

The "pathwise" modalities $\underset{\rightarrow}{p\alpha}$ thus give a more refined expressivity than the "broad" modality $\underset{\rightarrow}{\alpha}$. But the latter is more useful for expressing safety and liveness properties, involving all possible successor states in a single operator, see Example 3.7 (iii).

5 Galois Algebras

Now that we have seen several examples of Galois algebras—arising from next-time operators for coalgebras—it is time to have a closer look at these structures. This section repeats several basic facts from [Kar98], and adds certain results (like Lemmas 5.3, 5.4) which relate specifically to coalgebras and polynomial functors.

[3] In recent, as yet unpublished, work of Rößiger these size restrictions are removed, and the finite powerset functors is included in this result.

Definition 2.1 already introduced Galois algebras B as complete Boolean algebras with a nexttime operator $\underset{\rightarrow}{\bullet} : B \to B$ preserving all meets. The resulting "last time" left adjoint was written as $\underset{\leftarrow}{\bullet}$. It should be noted that $\underset{\rightarrow}{\bullet}$ is the weak lasttime operator, and $\underset{\leftarrow}{\bullet}$ the strong lasttime. To emphasise this aspect we sometimes write $\underset{\rightarrow_{\mathrm{w}}}{\bullet}$ for $\underset{\rightarrow}{\bullet}$ and $\underset{{}_{\mathrm{s}}\leftarrow}{\bullet}$ for $\underset{\leftarrow}{\bullet}$. For a general Galois algebra we write a bullet $\bullet$ where we put a particular (coalgebraic) operation in concrete examples.

We first recall how other temporal operators can be defined, using fixed points.

Notation	Meaning	Definition
$\underset{\rightarrow_{\mathrm{s}}}{\bullet} b$	at some next step b	$\neg(\underset{\rightarrow}{\bullet} \neg b)$
$\underset{{}_{\mathrm{w}}\leftarrow}{\bullet} b$	at each previous step b	$\neg(\underset{\leftarrow}{\bullet} \neg b)$
$\underset{\Rightarrow}{\bullet} b$	always in the future b	$\nu x.\, (b \wedge \underset{\rightarrow}{\bullet} x)$
$\underset{\Leftarrow}{\bullet} b$	sometime in the past b	$\mu x.\, (b \vee \underset{\leftarrow}{\bullet} x)$
$b\, \mathcal{U}\, c$	b until c	$\mu x.\, (c \vee (b \wedge \underset{\rightarrow_{\mathrm{s}}}{\bullet} x))$
$b\, \mathcal{S}\, c$	b since c	$\mu x.\, (c \vee (b \wedge \underset{{}_{\mathrm{s}}\leftarrow}{\bullet} x))$

We use different notation from [Kar98]. The following table gives an overview.

Here	$\underset{\rightarrow}{\bullet} = \underset{\rightarrow_{\mathrm{w}}}{\bullet}$	$\underset{\leftarrow}{\bullet} = \underset{{}_{\mathrm{s}}\leftarrow}{\bullet}$	$\underset{\rightarrow_{\mathrm{s}}}{\bullet}$	$\underset{{}_{\mathrm{w}}\leftarrow}{\bullet}$	$\underset{\Rightarrow}{\bullet}$	$\underset{\Leftarrow}{\bullet}$
von Karger [Kar98]	$\tilde{\oplus}$	$\ominus$	$\oplus$	$\tilde{\ominus}$	$\boxplus$	⟠

What makes Galois algebras appealing is that their defining requirements are very simple, but have strong consequences. For example, all axioms and rules of CTL are valid in Galois algebras, see [Kar98, 7.2].

As an example, we consider what is usually called the induction[4] rule of temporal logic, formulated inside an arbitrary Galois algebra:

$$b \wedge \underset{\Rightarrow}{\bullet}(b \supset \underset{\rightarrow}{\bullet} b) \;\leq\; \underset{\Rightarrow}{\bullet} b$$

where $b \supset c = \neg b \vee c$ is implication in a Boolean algebra. We call an element b in a Galois algebra an invariant if $b \leq \underset{\rightarrow}{\bullet} b$, or equivalently, if $\underset{\leftarrow}{\bullet} b \leq b$. Then it is easy to see that $\underset{\Rightarrow}{\bullet} b$ is the greatest invariant below b. Hence it suffices to show that the left-hand-side $b \wedge \underset{\Rightarrow}{\bullet}(b \supset \underset{\rightarrow}{\bullet} b)$ is also below b and is also an invariant. The first point is immediate, and for the second we calculate:

$$\begin{array}{rcll}
b \wedge \underset{\Rightarrow}{\bullet}(b \supset \underset{\rightarrow}{\bullet} b) & = & b \wedge (b \supset \underset{\rightarrow}{\bullet} b) \wedge \underset{\rightarrow}{\bullet}\underset{\Rightarrow}{\bullet}(b \supset \underset{\rightarrow}{\bullet} b) & \text{since } \underset{\Rightarrow}{\bullet} \text{ is a fixed point} \\
& \leq & \underset{\rightarrow}{\bullet} b \wedge \underset{\rightarrow}{\bullet}\underset{\Rightarrow}{\bullet}(b \supset \underset{\rightarrow}{\bullet} b) & \\
& = & \underset{\rightarrow}{\bullet}(b \wedge \underset{\Rightarrow}{\bullet}(b \supset \underset{\rightarrow}{\bullet} b)) & \text{since } \underset{\rightarrow}{\bullet} \text{ is a right adjoint.}
\end{array}$$

Certain additional requirements can be imposed on Galois algebras so that all axioms and rules from linear temporal logic (LTL) are valid, see [Kar98, 5.3 and 6].

[4] To use 'induction' for a rule which crucially depends on a greatest fixed is a misnomer; it is better called 'coinduction' rule.

Here we concentrate on the following points relating to forward- or right-linearity. These requirements will be relevant for coalgebras.

5.1. Definition. A Galois algebra will be called strict if $\underset{\rightarrow_{\mathrm{w}}}{\bullet} \bot = \bot$. And it will be called affine (right-linear in [Kar98]) if $\underset{\rightarrow_{\mathrm{s}}}{\bullet} b \leq \underset{\rightarrow_{\mathrm{w}}}{\bullet} b$, for all elements b.

Intuitively, in a strict Galois algebra there is at every stage at least one successor state, whereas in an affine one there is at most one successor state. There are dual requirements about predecessor states, but they will not be considered here. The combination of these requirements gives models of LTL, see [Kar98].

A typical property which holds in an affine Galois algebra is:

$$\underset{\rightarrow}{\bullet}(b \supset c) = \underset{\rightarrow}{\bullet} b \supset \underset{\rightarrow}{\bullet} c \qquad (\text{where } \underset{\rightarrow}{\bullet} = \underset{\rightarrow_{\mathrm{w}}}{\bullet})$$

The direction ($\leq$) holds in arbitrary Galois algebras because $\underset{\rightarrow}{\bullet}$ preserves meets (since it's a right adjoint). The reverse direction ($\geq$) uses the affine inequality $\neg \underset{\rightarrow}{\bullet} x \leq \underset{\rightarrow}{\bullet} \neg x$ in:

$$\begin{array}{rcll}
\underset{\rightarrow}{\bullet}(b \supset c) & = & \underset{\rightarrow}{\bullet}(\neg b \vee c) & \\
& \geq & \underset{\rightarrow}{\bullet} \neg b \vee \underset{\rightarrow}{\bullet} c & \text{by monotonicity of } \underset{\rightarrow}{\bullet} \\
& \geq & \neg \underset{\rightarrow}{\bullet} b \vee \underset{\rightarrow}{\bullet} c & \text{by the affine inequality} \\
& = & \underset{\rightarrow}{\bullet} b \supset \underset{\rightarrow}{\bullet} c. &
\end{array}$$

We consider some examples of strict and affine Galois algebras induced by coalgebras of polynomial functors. We take the functor $T(X) = X \times X$ on **Sets**. For an aribitrary coalgebra $\alpha \colon X \to T(X)$ and a predicate $P \subseteq X$ we have $\underset{\rightarrow_{\mathrm{s}}}{\alpha} P = \{x \mid P(\pi\alpha(x)) \vee P(\pi'\alpha(x))\}$ and $\underset{\rightarrow_{\mathrm{w}}}{\alpha} P = \{x \mid P(\pi\alpha(x)) \wedge P(\pi'\alpha(x))\}$. The Galois algebra induced by α is strict, since $\underset{\rightarrow_{\mathrm{w}}}{\alpha} \emptyset = \emptyset$. But if it is affine or not cannot be stated in general. For instance, consider the state space $\mathbb{N}$ with coalgebras $\beta, \gamma \colon \mathbb{N} \to T(\mathbb{N})$ given by $\beta(x) = \langle x, x \rangle$, and $\gamma(x) = \langle x, x+1 \rangle$. Then $\underset{\rightarrow_{\mathrm{s}}}{\beta} P = \{x \mid P(x)\} = \underset{\rightarrow_{\mathrm{w}}}{\beta} P$, but $\underset{\rightarrow_{\mathrm{s}}}{\gamma} P = \{x \mid P(x) \vee P(x+1)\}$ which is not contained in $\underset{\rightarrow_{\mathrm{w}}}{\gamma} P = \{x \mid P(x) \wedge P(x+1)\}$.

Thus the property of being affine depends on the coalgebra, and not on the functor. But there is a bit more we can say.

5.2. Definition. Consider a polynomial functor $T \colon \mathbf{Sets} \to \mathbf{Sets}$, with predicate lifting functions $(-)^T \colon \mathcal{P}(X) \to \mathcal{P}(T(X))$ as introduced in Definition 3.1.

(i) We say that T has a strict (predicate) lifting if the function $(-)^T$ is strict, *i.e.* preserves least predicates: $\emptyset^T = \emptyset$.

(ii) And we say that T has a (finitely) affine (predicate) lifting if $(-)^T$ preserves non-empty finite supremema. This amounts to $(P_1 \cup P_2)^T = P_1^T \cup P_2^T$ for each pair of predicates $P_1, P_2 \subseteq X$ on a set X.

(Strict and affine functions between complete lattices are considered in [Jac94] as one of the running examples giving categories having tensors with diagonals or with projections, and with exponential operators ! introducing only weakening or only contraction. The issue, like here, is the distinction between at least/most once. See also [Jac93] for examples of models of untyped lambda calculi with variables occuring at least/most once, constructed from strict/affine functions.)

5.3. Lemma. *If T is a polynomial functor with a strict/affine lifting, then the Galois algebra of each coalgebra of T is strict/affine.*

Proof. If T has a strict lifting, then for each coalgebra $\alpha\colon X \to T(X)$ we get

$$\underset{\to_{\mathrm{w}}}{\alpha}\,\emptyset = \alpha^*(\emptyset^T) = \alpha^*(\emptyset) = \emptyset.$$

And if the lifting is affine, then

$$\begin{array}{rcll}
\underset{\to_{\mathrm{s}}}{\alpha}\,P \subseteq \underset{\to_{\mathrm{w}}}{\alpha}\,P & \Leftrightarrow & \neg\alpha^*((\neg P)^T) \subseteq \alpha^*(P^T) & \\
& \Leftrightarrow & \alpha^*((\neg P)^T \cup P^T) = X & \\
& \Leftrightarrow & \alpha^*(X^T) = X & \\
& \Leftrightarrow & \alpha^*(T(X)) = X & \text{since } (-)^T \text{ is a right adjoint} \\
& \Leftrightarrow & \mathsf{true}. & \square
\end{array}$$

Whether a polynomial functor has a strict/affine lifting can be deduced from its structure.

5.4. Lemma. (i) *The identity functor* **Sets** $\to$ **Sets** *has a strict lifting. And if T_1, T_2:* **Sets** $\to$ **Sets** *have a strict lifting, then so have $T_1 + T_2$, T_1^A, $T_1 \times S$, where A is an arbitrary set and S is an arbitrary polynomial functor* **Sets** $\to$ **Sets**.

(ii) *The identity functor* **Sets** $\to$ **Sets** *also has an affine lifting. And if both T_1, T_2:* **Sets** $\to$ **Sets** *have an affine lifting, then so have $T_1 + T_2$, $K_A \times T_1$, where A is an arbitrary set.* □

These strict and affine lifting properties can also be investigated for the pathwise operators from Section 4. But that will not be done here.

Just like Heyting algebras are the intuitionistic versions of Boolean algebras, there are intuitionistic versions of Galois algebras. There, the weak and strong versions of nexttime and lasttime are not interdefinable via negation, and have to be present separately. We shall see examples in Subsection 6.1. Here we merely repeat the definition from [Kar98].

5.5. Definition. An intuitionistic Galois algebra consists of a complete Heyting algebra B with a weak and a strong nexttime operator $\underset{\to_{\mathrm{w}}}{\bullet}, \underset{\to_{\mathrm{s}}}{\bullet}\colon B \to B$, both preserving arbitrary meets, and satisfying the inequalities:

$$\underset{\to_{\mathrm{s}}}{\bullet}\,x \wedge \underset{\to_{\mathrm{w}}}{\bullet}\,y \leq \underset{\to_{\mathrm{s}}}{\bullet}\,(x \wedge y) \qquad \text{and} \qquad \underset{{}_{\mathrm{s}}\leftarrow}{\bullet}\,x \wedge \underset{{}_{\mathrm{w}}\leftarrow}{\bullet}\,y \leq \underset{{}_{\mathrm{s}}\leftarrow}{\bullet}\,(x \wedge y)$$

where $\underset{{}_{\mathrm{s}}\leftarrow}{\bullet} \dashv \underset{\to_{\mathrm{w}}}{\bullet}$ and $\underset{{}_{\mathrm{w}}\leftarrow}{\bullet} \dashv \underset{\to_{\mathrm{s}}}{\bullet}$ are the induced left adjoints, playing the rôle of strong and weak lasttime operators.

We conclude this section by introducing homomorphisms of (intuitionistic) Galois algebras. The canonical examples in the next section will be substitution functions f^*. In the examples they all preserve arbitrary meets and joins, so that is what we shall include in the definition of homomorphism. But possibly in another context, a different requirement is more appropriate.

5.6. Definition. (i) A homomorphism $(B, \underset{\to}{\bullet}) \to (C, \underset{\to}{\bullet})$ between ordinary (non-intuitionistic) Galois algebras is a function $f\colon B \to C$ which preserves all meets and joins, and commutes with the nexttime operations: $f \circ \underset{\to}{\bullet} = \underset{\to}{\bullet} \circ f$. This yields a category, which we shall write as **GA**.

(ii) Similarly, a homomorphism $f\colon (B, \underset{\to_{\mathrm{w}}}{\bullet}, \underset{\to_{\mathrm{s}}}{\bullet}) \to (C, \underset{\to_{\mathrm{w}}}{\bullet}, \underset{\to_{\mathrm{s}}}{\bullet})$ between intuitionistic Galois algebras is a function $f\colon B \to C$ preserving all meets and joins and commuting with both nexttime operations: $f \circ \underset{\to_{\mathrm{w}}}{\bullet} = \underset{\to_{\mathrm{w}}}{\bullet} \circ f$ and $f \circ \underset{\to_{\mathrm{s}}}{\bullet} = \underset{\to_{\mathrm{s}}}{\bullet} \circ f$. This gives a category **iGA**.

Notice that a morphism f of (intuitionistic) Galois algebras maps greatest fixed points νg to greatest fixed points νh, for meet-preserving functions g and h with $h \circ f = f \circ g$. In particular, it will commute with the henceforth operators $\underset{\Rightarrow_{\mathrm{w}}}{\bullet}$ and $\underset{\Rightarrow_{\mathrm{s}}}{\bullet}$ induced by the nexttime operators $\underset{\to_{\mathrm{w}}}{\bullet}$ and $\underset{\to_{\mathrm{s}}}{\bullet}$—as in Lemma 3.6 (iii).

6 Indexed Galois Algebras

In this section we first collect results from the previous sections in a summarising theorem. Subsequently we concentrate on two specific examples, and show how the coalgebraic structures which are of central importance in this paper also exist in some other, possibly unexpected, situations that have been considered in the literature (without the coalgebraic perspective).

Proposition 3.4 and Lemma 3.6 (i) yield the following fundamental result.

6.1. Theorem. *For each polynomial functor $T\colon$* **Sets** $\to$ **Sets** *there is functor*

$$\mathit{CoAlg}(T)^{op} \longrightarrow \boldsymbol{GA}$$

given by

$$\Big(X \xrightarrow{\alpha} T(X)\Big) \longmapsto \Big(\mathcal{P}(X), \underset{\to}{\alpha}\Big) \qquad \textit{and} \qquad f \longmapsto f^{*} \qquad \square$$

This functor forms what may be called an "indexed Galois algebra", providing a predicate logic on coalgebras. It can be seen as arising via composition (or change-of-base, see [Jac99]) along the forgetful functor CoAlg(T) $\to$ **Sets** from the indexed complete Boolean algebra

$$\mathbf{Sets}^{\mathrm{op}} \xrightarrow{\quad\mathcal{P}\quad} \mathbf{cBA}$$

incorporating the standard predicate logic on sets—where **cBA** is a category of complete Boolean algebras.

In this section we shall describe similar examples following this pattern.

6.1 Galois algebras indexed by metric spaces

Let $[0, 1]$ be the unit interval of real numbers. It can be seen as a domain of "fuzzy" truth values, with 0 as false and 1 as true (say). A function of the form $X \to [0, 1]$ can then be considered as a fuzzy predicate, and a function of the form $X \times X \to [0, 1]$ as a fuzzy (transition) relation, describing for example the probability of a transition $x \to x'$. In [Kar98] an intuitionistic Galois algebra is constructed out of such fuzzy predicates, given a transition relation $R\colon X \times X \to [0, 1]$. It involves for a fuzzy predicate $\varphi\colon X \to [0, 1]$ strong nexttime and lasttime operators, defined via the "max-min products":

$$\begin{array}{rcl} \underset{\to_{\mathrm{s}}}{R}\,\varphi & = & \lambda x.\, \max\limits_{y \in X} \min\{R(x, y), \varphi(y)\} \\ \underset{{}_{\mathrm{s}}\leftarrow}{R}\,\varphi & = & \lambda x.\, \max\limits_{y \in X} \min\{R(y, x), \varphi(y)\}. \end{array}$$

Here we shall redescribe this intuitionistic Galois algebra as resulting from the general constructions in this paper. We shall describe these constructions more generally in terms of metric spaces and metric predicates, see also [Law73, Ken90], or [Jac99, Example 4.6.3 (iv)].

First some preliminaries. We consider metric spaces (X, d) where the distance function $d: X \times X \to [0, \infty]$ takes values in the non-negative reals, extended with a top element ∞. As morphisms $(X, d) \to (Y, d)$ between such metric spaces we take "non-expansive" functions $f: X \to Y$, satisfying $d(f(x), f(x')) \leq d(x, x')$, for all $x, x' \in X$. This yields a category $\mathbf{MS}$, with a forgetful functor $\mathbf{MS} \to \mathbf{Sets}$. The latter has a left adjoint which provides an arbitrary set X with the discrete metric: $d(x, x') = \infty$ for $x \neq x'$, and $d(x, x) = 0$. We shall need the tensor product $X \otimes Y$ of two metric space. It has the Cartesian product $X \times Y$ as underlying set, with distance function $d((x, y), (x', y')) = d(x, x') + d(y, y')$. The projection function $\pi: X \times Y \to X$ is then non-expansive, so that we have a tensor with projections $X \otimes Y \to X$, see [Jac94].

The interval $[0, \infty]$ with it usual order forms a complete Heyting algebra, with max as least upper bound and min as greatest lower bound. The implication $r \supset s$ for $r, s \in [0, \infty]$ is given as $r \supset s = \infty$ if $r \leq s$, and $r \supset s = s$ otherwise. Then $\min\{t, r\} \leq s \Leftrightarrow t \leq r \supset s$. And thus for a subset $S \subseteq [0, \infty]$, $\min\{t, \max S\} = \max\{\min\{t, s\} \mid s \in S\}$.

A metric predicate on a metric space (X, d) is a non-expansive function $\varphi: X \to [0, \infty]$. The set $\mathrm{MP}(X, d)$ of metric predicates on (X, d) is a metric space itself, with distance $d(\varphi, \psi) = \max_{x \in X} |\varphi(x) - \psi(x)|$. This set can be ordered pointwise, and thus inherits the complete Heyting algebra structure from $[0, \infty]$. Moreover, the mapping $(X, d) \mapsto \mathrm{MP}(X, d)$ extends to a functor $\mathbf{MS}^{\mathrm{op}} \to \mathbf{cHA}$, since a non-expansive function $f: X \to Y$ yields a substitution function $f^*: \mathrm{MP}(Y, d) \to \mathrm{MP}(X, d)$ by precomposition. We thus get metric-space-indexed complete Heyting algebras. Note that f^* preserves the (pointwise) joins and meets, and thus has both a left adjoint $\coprod_f$ and a right adjoint $\prod_f$. Explicitly, they are given on a metric predicate $\varphi: X \to [0, \infty]$ as $\coprod_f(\varphi)(y) = \max_{x \in X}\{\varphi(x) \mid f(x) = y\}$ and $\prod_f(\varphi)(y) = \min_{x \in X}\{\varphi(x) \mid f(x) = y\}$. Using the exponents one can prove the "Frobenius" equation: $\coprod_f(\varphi \wedge f^*(\psi)) = \coprod_f(\varphi) \wedge \psi$, where $\wedge$ is min. We shall also make use of the "Beck-Chevalley" property for coproducts $\coprod_\pi$ and products $\prod_\pi$ along projections $\pi: X \otimes Z \to X$. These satisfy, for $f: X \to Y$ and $\varphi: Y \otimes Z \to [0, \infty]$,

$$f^*(\textstyle\coprod_\pi(\varphi)) = \coprod_\pi((f \otimes id)^*(\varphi)) \qquad \text{and} \qquad f^*(\prod_\pi(\varphi)) = \prod_\pi((f \otimes id)^*(\varphi)).$$

Next we consider taking metric predicates as a (covariant) functor $\mathrm{MP}: \mathbf{MS} \to \mathbf{MS}$, given on morphisms as $f \mapsto \coprod_f$. It can be seen as a metric analogue of the covariant powerset functor. A coalgebra $\alpha: (X, d) \to \mathrm{MP}(X, d)$ is a "metric fuzzy transition system". As above, one can interpret $\alpha(x)(x') \in [0, \infty]$ as some probability for a transition $x \to x'$, for $x, x' \in X$.

What we need to understand is the lifting of the functor MP to predicates $\varphi: X \to [0, \infty]$. It should yield a new predicate $\varphi^{\mathrm{MP}}: \mathrm{MP}(X, d) \to [0, \infty]$. Basically we follow the set-theoretic formulations for the powerset functor in Definition 3.1. Therefore we first need a metric membership relation ϵ, given as metric predicate $\epsilon: \mathrm{MP}(X, d) \otimes (X, d) \to [0, \infty]$. It is simply given by evaluation:[5] $(\varphi, x) \mapsto \varphi(x)$, and is easily seen to be non-expansive. Now we can can define for metric predicates

[5] The set-theoretic membership relation $\epsilon \hookrightarrow \mathcal{P}(A) \times A$ can equivalently be described as evaluation $\{0, 1\}^A \times A \to \{0, 1\}$, using characteristic functions instead of subsets.

$\varphi\colon X \to [0,\infty]$ and $\psi\colon \mathrm{MP}(X,d) \to [0,\infty]$,

$$\begin{array}{rcll}
\varphi^{\mathrm{MP}} & = & \prod_{\pi}(\epsilon \supset \pi'^{*}(\varphi)) & \text{(see Definition 3.1 (i))} \\
& = & \lambda\chi\colon \mathrm{MP}(X,d).\ \min\limits_{y\in X} \chi(y) \supset \varphi(y) & \\
& = & \lambda\chi\colon \mathrm{MP}(X,d).\ \min\limits_{y\in X}\{\varphi(y) \mid \varphi(y) < \chi(y)\} & \\
\psi_{\mathrm{MP}} & = & \coprod_{\pi'}(\epsilon \wedge \pi^{*}(\psi)) & \text{(see Definition 3.1 (ii))} \\
& = & \lambda x\colon X.\ \max\{\min\{\chi(x), \psi(\chi)\} \mid \chi \in \mathrm{MP}(X,d)\}. &
\end{array}$$

These operations are used for the weak nexttime and the strong lasttime operations. For strong nexttime and weak lasttime we also consider, in analogy with the formulations at the end of Example 3.7 (ii):

$$\begin{array}{rcl}
\coprod_{\pi}(\epsilon \wedge \pi'^{*}(\varphi)) & : & \mathrm{MP}(X,d) \longrightarrow [0,\infty] \\
\prod_{\pi'}(\epsilon \supset \pi^{*}(\psi)) & : & X \longrightarrow [0,\infty]
\end{array}$$

Assuming a coalgebra $\alpha\colon (X,d) \to \mathrm{MP}(X,d)$, this leads to the following weak and strong next- and last-time operators, all with type $\mathrm{MP}(X,d) \to \mathrm{MP}(X,d)$.

$$\begin{array}{rcl}
\underset{\rightarrow_{\mathrm{w}}}{\alpha}\varphi & = & \alpha^{*}(\varphi^{\mathrm{MP}}) \\
& = & \lambda x\colon X.\ \min\limits_{y\in X}\{\varphi(y) \mid \varphi(y) < \alpha(x)(y)\} \\
\underset{{}_{\mathrm{s}}\leftarrow}{\alpha}\varphi & = & (\coprod_{\alpha}(\varphi))_{\mathrm{MP}} \\
& = & \lambda x\colon X.\ \max\limits_{\chi}\min\{\chi(x), \max\limits_{y\in X}\{\varphi(y) \mid \alpha(y) = \chi\}\} \\
& = & \lambda x\colon X.\ \max\limits_{\chi}\max\limits_{y\in X}\{\min\{\chi(x), \varphi(y)\} \mid \alpha(y) = \chi\} \\
& = & \lambda x\colon X.\ \max\limits_{y\in X}\min\{\alpha(y)(x), \varphi(y)\} \\
\underset{\rightarrow_{\mathrm{s}}}{\alpha}\varphi & = & \alpha^{*}(\coprod_{\pi}(\epsilon \wedge \pi'^{*}(\varphi))) \\
& = & \lambda x\colon X.\ \max\limits_{y\in X}\min\{\alpha(x)(y), \varphi(y)\} \\
\underset{{}_{\mathrm{w}}\leftarrow}{\alpha}\varphi & = & \prod_{\pi'}(\epsilon \supset \pi^{*}(\prod_{\alpha}(\varphi))) \\
& = & \lambda x\colon X.\ \min\limits_{\chi}\min\{\chi(x), \min\limits_{y\in X}\{\varphi(y) \mid \alpha(y) = \chi\}\} \\
& = & \lambda x\colon X.\ \min\limits_{y\in X}\min\{\alpha(y)(x), \varphi(y)\}.
\end{array}$$

It is then easy to see that there are Galois connections $\underset{{}_{\mathrm{s}}\leftarrow}{\alpha}\varphi_1 \leq \varphi_2 \Leftrightarrow \varphi_1 \leq \underset{\rightarrow_{\mathrm{w}}}{\alpha}\varphi_2$ and $\underset{\rightarrow_{\mathrm{s}}}{\alpha}\varphi_1 \leq \varphi_2 \Leftrightarrow \varphi_1 \leq \underset{{}_{\mathrm{w}}\leftarrow}{\alpha}\varphi_2$. For example,

$$\begin{array}{rcl}
\underset{\rightarrow_{\mathrm{s}}}{\alpha}\varphi_1 = \alpha^{*}(\coprod_{\pi}(\epsilon \wedge \pi'^{*}(\varphi_1))) \leq \varphi_2 & \Leftrightarrow & \coprod_{\pi}(\epsilon \wedge \pi'^{*}(\varphi_1)) \leq \prod_{\alpha}(\varphi_2) \\
& \Leftrightarrow & \epsilon \wedge \pi'^{*}(\varphi_1) \leq \pi^{*}(\prod_{\alpha}(\varphi_2)) \\
& \Leftrightarrow & \pi'^{*}(\varphi_1) \leq \epsilon \supset \pi^{*}(\prod_{\alpha}(\varphi_2)) \\
& \Leftrightarrow & \varphi_1 \leq \prod_{\pi'}(\epsilon \supset \pi^{*}(\prod_{\alpha}(\varphi_2))) = \underset{{}_{\mathrm{w}}\leftarrow}{\alpha}\varphi_2.
\end{array}$$

Further, the additional requirements for intuitionistic Galois algebras can be proved by abstract “logical” calculation using the Frobenius equation:

$$\begin{array}{rcl}
\underset{\rightarrow_{\mathrm{s}}}{\alpha}\varphi_1 \wedge \underset{\rightarrow_{\mathrm{w}}}{\alpha}\varphi_2 & = & \alpha^*\big(\coprod_\pi(\epsilon \wedge \pi'^*(\varphi_1)) \wedge \prod_\pi(\epsilon \supset \pi'^*(\varphi_2))\big) \\
& = & \alpha^* \coprod_\pi \big(\epsilon \wedge \pi'^*(\varphi_1) \wedge \pi^* \prod_\pi(\epsilon \supset \pi'^*(\varphi_2))\big) \\
& \leq & \alpha^* \coprod_\pi \big(\epsilon \wedge \pi'^*(\varphi_1) \wedge (\epsilon \supset \pi'^*(\varphi_2))\big) \\
& \leq & \alpha^* \coprod_\pi \big(\epsilon \wedge \pi'^*(\varphi_1) \wedge \pi'^*(\varphi_2)\big) \\
& = & \alpha^* \coprod_\pi \big(\epsilon \wedge \pi'^*(\varphi_1 \wedge \varphi_2)\big) \\
& = & \underset{\rightarrow_{\mathrm{s}}}{\alpha}(\varphi_1 \wedge \varphi_2) \\
\underset{{}_{\mathrm{s}}\leftarrow}{\alpha}\varphi_1 \wedge \underset{{}_{\mathrm{w}}\leftarrow}{\alpha}\varphi_2 & = & \coprod_{\pi'}(\epsilon \wedge \pi^* \coprod_\alpha(\varphi_1)) \wedge \prod_{\pi'}(\epsilon \supset \pi^* \prod_\alpha(\varphi_2)) \\
& = & \coprod_{\pi'} \big(\epsilon \wedge \pi^* \coprod_\alpha(\varphi_1) \wedge \pi'^* \prod_{\pi'}(\epsilon \supset \pi^* \prod_\alpha(\varphi_2))\big) \\
& \leq & \coprod_{\pi'} \big(\epsilon \wedge \pi^* \coprod_\alpha(\varphi_1) \wedge (\epsilon \supset \pi^* \prod_\alpha(\varphi_2))\big) \\
& \leq & \coprod_{\pi'} \big(\epsilon \wedge \pi^*(\coprod_\alpha(\varphi_1) \wedge \prod_\alpha(\varphi_2))\big) \\
& = & \coprod_{\pi'} \big(\epsilon \wedge \pi^* \coprod_\alpha(\varphi_1 \wedge \alpha^* \prod_\alpha(\varphi_2))\big) \\
& \leq & \coprod_{\pi'} \big(\epsilon \wedge \pi^* \coprod_\alpha(\varphi_1 \wedge \varphi_2)\big) \\
& = & \underset{{}_{\mathrm{s}}\leftarrow}{\alpha}(\varphi_1 \wedge \varphi_2).
\end{array}$$

Finally, for a non-expansive function $f\colon X \to Y$ forming a coalgebra homomorphism from $\alpha\colon X \to \mathrm{MP}(X)$ to $\beta\colon Y \to \mathrm{MP}(Y)$—so that $\beta \circ f = \mathrm{MP}(f) \circ \alpha$—the induced substitution functor $f^*\colon \mathrm{MP}(Y) \to \mathrm{MP}(X)$ commutes with the weak and strong nexttime operators. For instance,

$$\begin{array}{rcl}
f^*(\underset{\rightarrow_{\mathrm{w}}}{\beta}\varphi) & = & f^*\beta^*(\varphi^{\mathrm{MP}}) \\
& = & \alpha^*\mathrm{MP}(f)^*(\prod_\pi(\epsilon \supset \pi'^*(\varphi))) \\
& = & \alpha^*(\prod_\pi((\mathrm{MP}(f) \otimes id)^*(\epsilon \supset \pi'^*(\varphi)))) \\
& & \qquad \text{by Beck-Chevalley} \\
& = & \alpha^*(\prod_\pi((\mathrm{MP}(f) \otimes id)^*(\epsilon) \supset \pi'^*(\varphi))) \\
& & \qquad \text{since substitution distributes over } \supset \\
& = & \alpha^*(\prod_\pi((\coprod_{id\otimes f}(\epsilon) \supset \pi'^*(\varphi)))) \\
& & \qquad (\mathrm{MP}(f) \otimes id)^*(\epsilon) = \coprod_{id\otimes f}(\epsilon) \text{ follows by direct calculation} \\
& \overset{(*)}{=} & \alpha^*(\prod_\pi \prod_{id\otimes f}(\epsilon \supset (id \otimes f)^*\pi'^*(\varphi))) \\
& = & \alpha^*(\prod_\pi(\epsilon \supset \pi'^* f^*(\varphi))) \\
& = & \underset{\rightarrow_{\mathrm{w}}}{\alpha} f^*(\varphi).
\end{array}$$

In the marked equation $\overset{(*)}{=}$ we have used the law[6] $\coprod_g(\varphi) \supset \psi = \prod_g(\varphi \supset g^*(\psi))$, which follows easily from the Frobenius equation.

Thus we have arrived at the following result.

6.2. Proposition. *The metric predicate functor MP:* $\boldsymbol{MS} \to \boldsymbol{MS}$ *on the category of metric spaces yields an indexed intuitionistic Galois algebra of the form:*

$$\mathit{Coalg}(\mathit{MP})^{op} \longrightarrow \boldsymbol{iGA}$$

[6] In logical terms it is $(\exists x.\,\varphi) \supset \psi = \forall x.\,(\varphi \supset \psi)$ with x not free in ψ.

arising by composing the forgetful functor Coalg(MP) $\to$ ***MS*** *with the indexed complete Heyting algebra* $\boldsymbol{MS}^{op} \to \boldsymbol{cHA}$. □

Example 8.6 from [Kar98] is a special case of this result, involving an intuitionistic Galois algebra arising from a transition relation and metric predicates on a *discrete* metric space. Here we have shown that there is much more—both metric and indexed—structure involved.

6.2 Galois algebras indexed by presheaves

In [GM88] a presheaf model for modal logic is presented. It involves nexttime $\underset{\rightarrow}{\cdot}$ and lasttime $\underset{\leftarrow}{\cdot}$ operators, given by cofreely and freely generated presheaves. Here we shall redescribe this model in the present coalgebraic context, as indexed Galois algebras, arising by change-of-base. This shows that the example fits into the general setting of this paper. We shall not introduce much extra structure—like in the previous fuzzy predicate example—but merely unveil the (implicit) coalgebraic content of the presheaf model.

The starting point is a small category $\mathbb{C}$. We shall write C_0 for its set of objects and $C_1 = \{(A, B, f) \mid A, B \in C_0 \wedge f \in \mathbb{C}(A, B)\}$ for its set of morphisms, with domain and codomain maps $\partial_0, \partial_1 : C_1 \to C_0$, given by $\partial_0(A, B, f) = A$ and $\partial_1(A, B, f) = B$. A presheaf on $\mathbb{C}$ is a functor $\mathbb{C}^{\mathrm{op}} \to \mathbf{Sets}$. Presheaves with natural transformation between them yield a category, written as $\mathbf{Sets}^{\mathbb{C}^{\mathrm{op}}}$.

Interestingly, this category of presheaves can also be described as a category of coalgebras, not of a functor but of a comonad. This comonad is given as $\prod_{\partial_1} \partial_0^*$ on the slice category $\mathbf{Sets}/C_0$, see [Joh77, Proposition 2.21], [LM92, V, 7, Theorem 2] (involving the monad redescription, which is equivalent to the present comonad form, using the Eilenberg-Moore Theorem, see also [Jac99, Remark 7.4.2 (iii)]). The objects of the slice category $\mathbf{Sets}/C_0$ can be identified with C_0-indexed collections of sets, written as $(U_A)_{A \in C_0}$. The comonad $T = \prod_{\partial_1} \partial_0^*$ sends such a family $\big(U_A\big)_{A \in C_0}$ to $\big(\prod_{h \in \partial_1^{-1}(A)} U_{\partial_0(h)}\big)_{A \in C_0}$. A presheaf $H : \mathbb{C}^{\mathrm{op}} \to \mathbf{Sets}$, corresponds to a $\prod_{\partial_1} \partial_0^*$-coalgebra on the family $\big(H(A)\big)_{A \in C_0}$. This coalgebra, say c_H, can be described as $(x \in H(A)) \longmapsto (\lambda h : B \to A.\, H(h)(x))$.

There is a standard logic on the slice category $\big(\mathbf{Sets}/C_0\big)^{\mathrm{op}} \to \mathbf{cBA}$ obtained by change-of-base (composition) with the standard logic of sets $\mathbf{Sets}^{\mathrm{op}} \to \mathbf{cBA}$ via the domain functor $\mathrm{dom} : \mathbf{Sets}/C_0 \to \mathbf{Sets}$. It maps a family $(U \to C_0)$ to the Boolean algebra $\mathcal{P}(U)$ of subsets of the domain. Such a subset $P \subseteq U$ can be identified with a family of of subsets $(P_A \subseteq U_A)_{A \in C_0}$. There is a lifting $P^T \subseteq \mathrm{dom}(T(U \to C_0))$. Since the functor T has a left adjoint, namely $\coprod_{\partial_0} \partial_1^*$, it preserves monos, so that P^T is defined as $T(P \hookrightarrow U)$. Basically, this is also what happens in Definition 3.1, see also Remark 3.8. Explicitly, $(P^T)_A = \{\phi \in \prod_{h \in \partial_1^{-1}(A)} U_{\partial_0(h)} \mid \forall h \in \partial_1^{-1}(A).\, \phi(h) \in P_{\partial_0(h)}\}$.

6.3. Lemma. *For a presheaf* $H : \mathbb{C}^{op} \to \mathbf{Sets}$, *considered as* $\prod_{\partial_1} \partial_0^*$*-coalgebra, the induced nexttime operator* $\underset{\rightarrow}{H}$ *is given on* $P = \big(P_A \subseteq H(A)\big)_{A \in C_0}$ *as:*

$$\Big(\{x \mid \forall B \in C_0.\, \forall f \in \mathbb{C}(B, A).\, H(f)(x) \in P_B\} \subseteq H(A)\Big)_{A \in C_0}.$$

And its left adjoint lasttime operator $\underset{\leftarrow}{H}$ *on* P *is:*

$$\Big(\{y \mid \exists B \in C_0.\, \exists g \in \mathbb{C}(A, B).\, \exists x \in H(B).\, y = H(g)(x) \wedge x \in P_B\} \subseteq H(A)\Big)_{A \in C_0}.$$

Proof. Recall that $\underset{\rightarrow}{H}P$ is defined as $c_H^*(P^T)$, where c_H is the above coalgebra $x \mapsto \lambda h.\, H(h)(x)$ associated with H. This yields the formulation used in the lemma. It is easily seen that $\underset{\leftarrow}{H}$ forms a left adjoint. □

As discussed in [GM88], $\underset{\rightarrow}{H}P$ is the maximum subpresheaf of H contained in P, and $\underset{\leftarrow}{H}P$ is the minimum subpresheaf containing P. Hence a predicate P is an invariant if and only if it forms a subpresheaf. The presheaf model is used in [GM88] for a completeness result for modal predicate logic. Summarising our perspective, we get the following redescription of this presheaf model.

6.4. Proposition. *For a small category $\mathbb{C}$, there is a presheaf-indexed Galois algebra:*

$$\left(\mathbf{Sets}^{\mathbb{C}^{op}}\right)^{op} \longrightarrow \boldsymbol{GA}$$

which is obtained by change-of-base along the forgetful functor from a category of (comonad) coalgebras to its underlying (slice) category. □

We conclude this example by adding the following observation.

6.5. Lemma. *In the above presheaf model $\underset{\Rightarrow}{H} = \underset{\rightarrow}{H}$ and $\underset{\Leftarrow}{H} = \underset{\leftarrow}{H}$.*

Proof. It must be shown that $\underset{\rightarrow}{H}P$ is the greatest fixed point of the operation $\big(S_A \subseteq H(A)\big)_{A \in C_0} \longmapsto \big((P_A \cap (\underset{\rightarrow}{H}S)_A) \subseteq H(A)\big)_{A \in C_0}$. This follows from the "transitive" nature of categories. □

Acknowledgements

Thanks to Alexandru Baltag, Martin Rößiger and Hendrik Tews for various discussions on coalgebras and modal logic.

References

[CE81] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logics of Programs*, number 131 in Lect. Notes Comp. Sci., pages 52–71. Springer, Berlin, 1981.

[DP90] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order.* Math. Textbooks. Cambridge Univ. Press, 1990.

[Eme90] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier/MIT Press, 1990.

[GM88] S. Ghilardi and G.C. Meloni. Modal and tense predicate logic: models in presheaves and categorical conceptualization. In F. Borceux, editor, *Categorical Algebra and its Applications*, number 1348 in Lect. Notes Math., pages 130–142. Springer, Berlin, 1988.

[Gol92] R. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes 7, Stanford, 2nd rev. edition, 1992.

[Gum99] H.P. Gumm. Functors for coalgebras. *Algebra Universalis*, 1999. To appear.

[Hen99] U. Hensel. *Definition and Proof Principles for Data and Processes.* PhD thesis, Techn. Univ. Dresden, Germany, 1999.

[HHJT98] U. Hensel, M. Huisman, B. Jacobs, and H. Tews. Reasoning about classes in object-oriented languages: Logical models and tools. In Ch. Hankin, editor, *European Symposium on Programming*, number 1381 in Lect. Notes Comp. Sci., pages 105–121. Springer, Berlin, 1998.

[HJ98] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. & Comp.*, 145:107–152, 1998.

[Jac93] B. Jacobs. Semantics of lambda-I and of other substructure lambda calculi. In M. Bezem and J.F. Groote, editors, *Typed Lambda Calculi and Applications*, number 664 in Lect. Notes Comp. Sci., pages 195–208. Springer, Berlin, 1993.

[Jac94] B. Jacobs. Semantics of weakening and contraction. *Ann. Pure & Appl. Logic*, 69(1):73–106, 1994.

[Jac96] B. Jacobs. Objects and classes, co-algebraically. In B. Freitag, C.B. Jones, C. Lengauer, and H.-J. Schek, editors, *Object-Orientation with Parallelism and Persistence*, pages 83–103. Kluwer Acad. Publ., 1996.

[Jac97] B. Jacobs. Invariants, bisimulations and the correctness of coalgebraic refinements. In M. Johnson, editor, *Algebraic Methodology and Software Technology*, number 1349 in Lect. Notes Comp. Sci., pages 276–291. Springer, Berlin, 1997.

[Jac99] B. Jacobs. *Categorical Logic and Type Theory.* North Holland, Amsterdam, 1999.

[Joh77] P.T. Johnstone. *Topos Theory.* Academic Press, London, 1977.

[Joh82] P.T. Johnstone. *Stone Spaces.* Number 3 in Cambridge Studies in Advanced Mathematics. Cambridge Univ. Press, 1982.

[JR97] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:222–259, 1997.

[JT51] B. Jónsson and A. Tarski. Boolean algebras with operators I. *Amer. Journ. Math.*, 73:891–939, 1951.

[JvdBH$^+$98] B. Jacobs, J. van den Berg, M. Huisman, M. van Berkum, U. Hensel, and H. Tews. Reasoning about classes in Java (preliminary report). In *Object-Oriented Programming, Systems, Languages and Applications*, pages 329–340. ACM Press, 1998.

[Kar98] B. von Karger. Temporal algebra. *Math. Struct. in Comp. Sci.*, 8:277–320, 1998.

[Ken90] R.E. Kent. The metric closure powerspace construction. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Language Semantics*, number 442 in Lect. Notes Comp. Sci., pages 173–199. Springer, Berlin, 1990.

[KT90] D. Kozen and J. Tiuryn. Logics of programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 789–840. Elsevier/MIT Press, 1990.

[Kur98] A. Kurz. Specifying coalgebras with modal logic. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in*

Computer Science, number 11 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 1998.

[Law73] F.W. Lawvere. Metric spaces, generalized logic, and closed categories. *Seminario Matematico e Fisico. Rendiconti di Milano*, 43:135–166, 1973.

[LM92] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic. A First Introduction to Topos Theory.* Springer, New York, 1992.

[McM93] K.L. McMillan. *Symbolic Model Checking.* Kluwer Acad. Publ., 1993.

[Mos99] L.S. Moss. Coalgebraic logic. *Ann. Pure & Appl. Logic*, 1999. To appear.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems.* Springer-Verlag, Berlin, 1992.

[MSS85] A. Melton, D.A. Schmidt, and G.E. Strecker. Galois connections and computer science applications. In D.H. Pitt, S. Abramsky, A. Poigné, and D.E. Rydeheard, editors, *Category Theory and Computer Programming*, number 240 in Lect. Notes Comp. Sci., pages 299–312. Springer, Berlin, 1985.

[ORSvH95] S. Owre, J.M. Rushby, N. Shankar, and F. von Henke. Formal verification for fault-tolerant architectures: Prolegomena to the design of PVS. *IEEE Trans. on Softw. Eng.*, 21(2):107–125, 1995.

[Pau94] L.C. Paulson. *Isabelle: A Generic Theorem Prover.* Number 828 in Lect. Notes Comp. Sci. Springer, Berlin, 1994.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Found. Comp. Sci.*, pages 46–57. IEEE, 1977.

[Pnu81] A. Pnueli. The temporal semantics of concurrent programs. *Theor. Comp. Sci.*, 31:45–60, 1981.

[Rei95] H. Reichel. An approach to object semantics based on terminal coalgebras. *Math. Struct. in Comp. Sci.*, 5:129–152, 1995.

[Röß99a] M. Rößiger. From modal logic to terminal coalgebras. *Theor. Comp. Sci.*, 1999. To appear.

[Röß99b] M. Rößiger. Languages for coalgebras on datafunctors. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, number 19 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 1999.

[Rut99] J. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 1999. To appear.

[SSC95] A. Sernadas, C. Sernadas, and J.F. Costa. Object specification logic. *Journ. of Logic and Computation*, 5(5):603–630, 1995.