Coalgebra and Modal Logic: a first introduction

Yde Venema*

Abstract

These notes give a brief introduction to the theory of universal coalgebra and coalgebraic modal logic.

*Institute for Logic, Language and Computation, University of Amsterdam, Science Park 107, NL–1098XG Amsterdam. E-mail: y.venema@uva.nl.

Contents

1 Introduction

Starting from concrete examples, this chapter introduces (set-based) coalgebras, together with some of the most important coalgebraic concepts, including coinduction, behavioural equivalence and bisimilarity. We then give a first discussion of the relation between coalgebra and modal logic, and we give some examples of coalgebraic modal logics.

1.1 State-based evolving systems

Example 1.1 Perhaps the simplest example of a computational process is the following black-box machine with two buttons, h and n. If we press the h-button, the machine displays some value from a data set C. No matter how many times we press the h-button, this value remains the same. Each time we push the n-button, however, we may observe a different value by pushing the h-button.

A natural way to formally describe this device is as a set S of internal states (that are not visible to the user), together with two maps

$$\begin{array}{rl} h: & S \to C \\ n: & S \to S, \end{array}$$

where $h(s)$ indicates the observable value at state s, and $n : S \to S$ is a function mapping a state $s \in S$ to its unique next state.

All that we may observe of a state s, is the *stream*

$$h(s) \cdot h(n(s)) \cdot h(n(n(s))) \cdot h(n^3(s)) \cdots$$

of data. This stream will be called the *behaviour* of s in $\mathbb{S} = (S, h, n)$, notation: $\mathsf{beh}_{\mathbb{S}}(s)$.

Two states s and s' in two black boxes can then be called *behaviourally equivalent*, notation: $\mathbb{S}, s \simeq \mathbb{S}', s'$, if they display the same behaviour, that is, if $\mathsf{beh}_{\mathbb{S}}(s) = \mathsf{beh}_{\mathbb{S}'}(s')$.

Example 1.2 A *deterministic finite state automaton* or *DFA* over an alphabet C is a triple $\mathbb{A} = (A, \delta, F)$, where A is a finite set of states, $\delta : A \times C \to A$ is the *transition map* of $\mathbb{A}$, and $F \subseteq A$ is the set of *accepting states* of the automaton. In contrast to the usual presentation we do not take the initial state of a DFA to be part of its structure; rather, we will consider *initialised versions* of DFAs, where we explicitly single out an initial state for the automaton.

Let C^* denote the set of finite words over C, then we may extend δ to a map $\widehat{\delta} : A \times C^* \to A$ as follows:

$$\begin{array}{rcl} \widehat{\delta}(a, \varepsilon) & := & a \\ \widehat{\delta}(a, cw) & := & \widehat{\delta}(\delta(a, c), w). \end{array}$$

We define, for a state $a \in A$,

$$L_{\mathbb{A}}(a) := \{w \in C^* \mid \widehat{\delta}(a, w) \in F\},$$

as the *language* accepted by $\mathbb{A}$, initialized at a. Two initialized automata $(\mathbb{A}, a)$ and $(\mathbb{A}', a')$ are *(language) equivalent* if they accept exactly the same words, that is, if $L_{\mathbb{A}}(a) = L_{\mathbb{A}'}(a')$.

Example 1.3 The key structures featuring in the semantics of *modal logic* are Kripke frames and Kripke models. A *Kripke frame* is a pair (S, R) consisting of a set S of objects called *states*, *points* or *worlds*, and an *accessibility relation* $R \subseteq S \times S$. A *Kripke model* is a triple $\mathbb{S} = (S, R, V)$ such that (S, R) is a Kripke frame (the *underlying Kripke frame* of the model), and V is a *valuation*, i.e., a map $\mathsf{Q} \to PS$, where Q is some fixed set of *proposition letters*.

A *bisimulation* between two Kripke models $\mathbb{S}$ and $\mathbb{S}'$ is a binary relation $Z \subseteq S \times S'$ such that, for all $s \in S$ and $s' \in S'$ with Zss' the following conditions hold:

(atom) s and s' satisfy the same proposition letters;

(forth) for all $t \in S$ such that Rst there is a $t' \in S'$ with $R's't'$ and Ztt';

(back) for all $t' \in S'$ such that $R's't'$ there is a $t \in S$ with Rst and Ztt'.

If there is a bisimulation Z linking s and s' we say that s and s' are *bisimilar*, notation $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ (or $Z : \mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ if we want to make the bisimulation explicit).

Given a modal language L, we let $Th^L_{\mathbb{S}}(s)$ denote the set of L-formulas that are true at s in $\mathbb{S}$; then two states are called *(L-)equivalent* (notation: $\equiv_L$) if they satisfy the same L-formulas. A theme in the theory of modal logic is to study the relation between equivalence and bisimilarity. A class of models C is said to have the *Hennessy-Milner* property with respect to a language L if $\equiv_L = \leftrightarroweq$ on C.

Example 1.4 The theory of *non-well-founded sets* provides an alternative to the standard axiomatic set theories by allowing sets to contain themselves, or otherwise violate the rule of well-foundedness. More in detail, in non-well-founded set theories, the Foundation Axiom *FA* is replaced by axioms implying its negation. For instance, working with the *anti-foundation axiom AFA* we may associate, with each so-called *apg* or *accessible pointed graph* (that is, a directed graph such that every node can be reached via a finite path from a specified *root* of the graph) a *hyperset*, that is, a set that is not necessarily well-founded. And, two apgs yield the *same* set iff they are bisimilar.

Example 1.5 As a final example of a state-based evolving system we mention Markov chains: transition systems that evolve probabilistically. Recall that a *(discrete) probability distribution* on a set S is a map $\mu : S \to [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. Formally, a Markov chain can be modelled as a pair (S, σ), where σ assigns a probability distribution σ_s to each state s.

For a concrete example, think of a gambler wagering €1 on a series of fair coin tosses – this series may be indefinite, or end if the gambler loses his money. This experiment can be modelled by the Markov chain (S, σ) where $S = \{s_n \mid n \in \omega\}$, with state s_n representing the state where the gambler owns €n. For $n > 0$ we have that σ_{s_n} assigns a 0.5 probability to both s_{n-1} and s_{n+1} (and a probability 0 to all other states), while σ_{s_0} assigns a 1.0 probability to s_0 (and a probability 0 to all other states).

1.2 Coalgebras and their morphisms

As we will see now, the structures described in the previous section all are specimens of *coalgebras*. Universal Coalgebra is a theory of state-based evolving systems, formulated in the language of category theory.[1]

[1]See the appendix for some background definitions on category theory.

Definition 1.6 Given an endofunctor $T : \mathsf{C} \to \mathsf{C}$ on some category C, a T-coalgebra is a pair (X, ξ) where X is an object in C and $\xi : X \to TX$ is an arrow in C. We will sometimes refer to T as the *type* of (X, ξ). If, for an arrow $f : X' \to X$, the following diagram commutes:

$$\begin{tikzcd} X' \arrow[r, "f"] \arrow[d, "\xi'"'] & X \arrow[d, "\xi"] \\ TX' \arrow[r, "Tf"] & TX \end{tikzcd} \tag{1}$$

we call f a *(coalgebra) morphism* from $\mathbb{X}' = (X', \xi')$ to $\mathbb{X} = (X, \xi)$, and write $f : \mathbb{X}' \to \mathbb{X}$.

We let $\mathsf{Coalg}_{\mathsf{C}}(T)$ denote the category with T-coalgebras as objects and T-coalgebra morphisms as arrows; the category C will be called the *base* category of $\mathsf{Coalg}_{\mathsf{C}}(T)$. $\lhd$

We will usually (but not always) confine our attention to *systems*, that is, coalgebras over the category Set. Intuitively, a set functor T specifies the one-step dynamics that a system can engage in.

Definition 1.7 A *set functor* is an endofunctor $T : \mathsf{Set} \to \mathsf{Set}$ on the category Set of sets and functions; given such a set functor T, we will sometimes refer to T-coalgebras as T-*systems*. Given the prominence of these coalgebras we will often simply write $\mathsf{Coalg}(T)$ rather than $\mathsf{Coalg}_{\mathsf{Set}}(T)$ to denote the class of all T-systems. Where $\mathbb{S} = (S, \sigma)$ is a T-system, we refer to S and σ as, respectively, the *carrier* or *state space* and the *transition map* or *coalgebra map* of $\mathbb{S}$. A *pointed* or *initialized* T-system is a triple (S, σ, s) such that (S, σ) is a T-system and $s \in S$. $\lhd$

The coalgebraic viewpoint on systems combines wide applicability and mathematical simplicity: since every set functor determines its own type of coalgebra, notions, properties and results of state-based systems can be uniformly explained just in terms of properties of their type functors. This applies to systems as diverse as streams, probabilistic transition systems, automata, Kripke structures and neighbourhood frames. In the appendix we give a list of set functors; here we give a few examples of the associated coalgebras.

Example 1.8 (a) The black boxes of Example 1.1 are systems of the functor type $K_C \times Id$, where K_C is the constant functor associated with the set C, and Id is the identity functor on Set.

(b) Deterministic finite automata (Example 1.2) are systems of type $2 \times Id^C$, where 2 is the set $\{0, 1\}$. To see this, consider a coalgebra $\mathbb{X} = (X, \xi)$ of this type; then ξ determines, for each state $x \in X$, two things: an element $\xi_0(x)$ of the set 2, specifying whether x is accepting ($\xi_0(x) = 1$) or not ($\xi_0(x) = 0$), and an element $\xi_1(x) \in X^C$, that is, a map $\xi_1(x) : C \to X$ providing a successor of x in X for each letter $c \in C$. A $2 \times Id^C$-coalgebra (X, ξ) thus corresponds to the DFA $X, \xi_0^{-1}(1), \xi_1)$.

(c) Kripke frames are coalgebras for the powerset functor P, whereas Kripke models are coalgebras for the functor $K_{P\mathsf{Q}} \times P$. We leave it for the reader to verify that, given two Kripke frames $\mathbb{S}$ and $\mathbb{S}'$, a map $f : S \to S'$ is a coalgebra morphism iff it is a bounded morphism, that is, for every $s \in S$ it satisfies:

∘ if $(s,t) \in R$ then $(fs, ft) \in R'$;
∘ if $(fs, t') \in R'$ then $(s,t) \in R$ for some t such that $ft = t'$.

(d) Non-well-founded sets can be represented by certain pointed P-systems.

(e) Markov chains are D-systems, where D is the distribution functor (which assigns, to a set S, a discrete probability distribution on S).

(f) For every set functor T we will allow the empty T-coalgebra $(\varnothing, \varnothing)$.

1.3 Final coalgebras and coinduction

For many coalgebra types T one may associate with an arbitrary state s in an arbitrary T-coalgebra $\mathbb{S}$, a natural notion of *behaviour*. This can often be formalised by defining a *behaviour map* and proving that this map is the unique coalgebra morphism from $\mathbb{S}$ to some *final* or *terminal* coalgebra $\mathbb{Z}$ of type T.

Definition 1.9 Let T be an endofunctor on some category C. A *T-coalgebra* $\mathbb{Z} = (Z, \zeta)$ is *final* or *terminal* if it is a final object in the category $\mathsf{Coalg}_{\mathsf{C}}(T)$; that is, if for every T-coalgebra $\mathbb{X} = (X, \xi)$ there is a *unique* morphism from $\mathbb{X}$ to $\mathbb{Z}$; this morphism will be denoted as $\mathsf{beh}_{\mathbb{X}} : \mathbb{X} \to \mathbb{Z}$. ◁

Note that final coalgebras, when they exist, are unique modulo isomorphism. For this reason we will often speak of *the* final T-coalgebra of a functor T.

Example 1.10 A *stream* over a set C is a map $\alpha : \omega \to C$ (where ω is the set of natural numbers). We may turn the set C^ω of C-streams into a $K_C \times Id$-coalgebra itself by endowing it with a coalgebra map $\gamma := (\mathsf{h}, \mathsf{t}) : C^\omega \to C \times C^\omega$. Here we define the maps $\mathsf{h} : C^\omega \to C$ and $\mathsf{t} : C^\omega \to C^\omega$ by putting, for an arbitrary C-stream $\alpha : \omega \to C$:

$$\begin{array}{lll} \mathsf{h}(\alpha) & := & \alpha(0) \\ \mathsf{t}(\alpha) & := & \lambda n.\alpha(n+1). \end{array}$$

That is, the coalgebra map γ splits an infinite C-stream $c_0 c_1 c_2 \cdots$ into its *head* c_0 and its *tail* $c_1 c_2 c_3 \cdots$.

It is then not very hard to prove that the *stream coalgebra* (C^ω, γ) is a final coalgebra for the functor $K_C \times Id$: this boils down to showing that, for an arbitrary 'black box machine' $\mathbb{S} = (S, h, n)$, the behaviour map $\mathsf{beh} : S \to C^\omega$ is the *unique* coalgebra morphism from $\mathbb{S}$ to (C^ω, γ).

Example 1.11 For a second example, fix an alphabet C and define a $(C\text{-})$*language* to be any set of finite words. Further on we will see that we can endow the collection $P(C^*)$ of C-languages with a very natural coalgebra map for the functor $2 \times Id^C$ of deterministic finite C-automata, and prove that the resulting structure is in fact a final $2 \times Id^C$-coalgebra.

Finality is also the key categorical concept underlying the important coalgebraic principle of *coinduction*. Here is a first example.

Example 1.12 Take the function zip that merges two streams by taking elements from either stream in turn. For a coalgebraic definition of this map, define the transition map $\delta : (C^\omega \times C^\omega) \to C \times (C^\omega \times C^\omega)$ as follows:

$$\delta(\alpha, \beta) := (\mathsf{h}(\alpha), (\beta, \mathsf{t}(\alpha)),$$

where h and t are the maps defined in Example 1.10. This defines a $K_C \times Id$-coalgebra on the set $(C^\omega \times C^\omega)$, so that by finality of the stream coalgebra (C^ω, γ) there is a (unique) map $\mathsf{zip} : C^\omega \times C^\omega \to C^\omega$ which is a coalgebra morphism from $(C^\omega \times C^\omega, \delta)$ to $(C^\omega, \langle \mathsf{h}, \mathsf{t}\rangle)$:

$$\begin{tikzcd}
C^\omega \times C^\omega \arrow[r, "\mathsf{zip}"] \arrow[d, "\delta"] & C^\omega \arrow[d, "\langle \mathsf{h},\mathsf{t}\rangle"] \\
C \times (C^\omega \times C^\omega) \arrow[r, "(\mathsf{id}_C{,}\mathsf{zip})"'] & C \times C^\omega
\end{tikzcd}$$

One may verify that this coalgebra morphism indeed defines the map that zips two streams together.

Unfortunately, as we will see further on, final coalgebras do not exist for every functor.

Example 1.13 In the categories of Kripke frames and Kripke models, final objects do not exist. The *canonical model* comes close, but to turn this structure into a final coalgebra, we have to enrich the base category Set with topological structure. As a result that we will discuss later on, we may see the canonical *general* frame as a final coalgebra for a suitable base category and coalgebra functor.

1.4 Behavioural equivalence and bisimilarity

Probably the most intuitive notion of equivalence between systems is that of *behavioral*, or *observational*, equivalence. The idea here is to consider two states to be similar if we cannot distinguish them by observations, because they display the same behavior. For instance, we call two deterministic automata (pointed $2 \times Id^C$-coalgebras) equivalent if they recognize the same language. In case the functor T admits a final coalgebra $\mathbb{Z}$, this idea is easily formalized by making state s in coalgebra $\mathbb{S}$ equivalent to state s' in coalgebra $\mathbb{S}'$ if $\mathsf{beh}_{\mathbb{S}}(s) = \mathsf{beh}_{\mathbb{S}'}(s')$. In case the functor does not admit a final coalgebra, we generalize this demand as follows.

Definition 1.14 Let $\mathbb{S} = (S, \sigma)$ and $\mathbb{S}' = (S', \sigma')$ be two systems for the set functor T. Then $s \in S$ and $s' \in S'$ are *behaviorally equivalent*, notation: $\mathbb{S}, s \simeq_T \mathbb{S}', s'$ if there is a T-system $\mathbb{X} = (X, \xi)$ and homomorphisms $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$ such that $f(s) = f'(s')$. $\lhd$

Remark 1.15 It is easily checked that in case T admits a final coalgebra $\mathbb{Z}$, then indeed $\mathbb{S}, s \simeq_T \mathbb{S}', s'$ iff $\mathsf{beh}_{\mathbb{S}}(s) = \mathsf{beh}_{\mathbb{S}'}(s')$. The direction from right to left is trivial, so assume that s and s' are behaviorally equivalent because of the existence of $\mathbb{X}$, f and f' as in the formulation of the definition. Observe that the map $\mathsf{beh}_{\mathbb{X}} \circ f$ is a coalgebra morphism from $\mathbb{S}$ to $\mathbb{Z}$, and likewise for $\mathsf{beh}_{\mathbb{X}} \circ f'$ and $\mathbb{S}'$. It then follows from the finality of $\mathbb{Z}$ that $\mathsf{beh}_{\mathbb{S}} = \mathsf{beh}_{\mathbb{X}} \circ f$ and $\mathsf{beh}_{\mathbb{S}'} = \mathsf{beh}_{\mathbb{X}} \circ f'$. Hence, from $f(s) = f'(s')$ it follows that $\mathsf{beh}_{\mathbb{S}}(s) = \mathsf{beh}_{\mathbb{S}'}(s')$, as required.

As we will see further on, in many cases of interest, behavioral equivalence can be characterized via the equally fundamental concept of *bisimilarity*, which involves the notion of a *coalgebraic bisimulation.*

Definition 1.16 A *bisimulation* between two systems $(\mathbb{S}, \sigma)$ and $(\mathbb{S}', \sigma')$ is a relation $B \subseteq S \times S'$ for which, as in the diagram below,

$$\begin{tikzcd} S \arrow[d, "\sigma"'] & B \arrow[l, "\pi"'] \arrow[r, "\pi'"] \arrow[d, dotted, "\beta"] & S' \arrow[d, "\sigma'"] \\ TS & TB \arrow[l, "T\pi"] \arrow[r, "T\pi'"'] & TS' \end{tikzcd} \tag{2}$$

there is a (not necessarily unique) coalgebra map $\beta : B \to TB$ such that the two projection maps, π and π', from B to S and S' respectively, are both coalgebra morphisms.

We write $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ to denote that there is some bisimulation B between $\mathbb{S}$ and $\mathbb{S}'$ that links the states s in $\mathbb{S}$ and s' in $\mathbb{S}'$. $\lhd$

That is, a bisimulation is a relation that, seen as a set, can itself be endowed with a coalgebra structure satisfying some natural conditions.

1.5 Coalgebra and modal logic

Logic comes in when we want to design specification languages for describing the behaviour of state-based evolving systems, and derivation systems for reasoning about this behaviour.

Definition 1.17 An *(abstract) logic* for T-coalgebras is a pair $(L, \Vdash)$ consisting of a set L of *formulas* and, for each T-coalgebra $\mathbb{S} = (S, \sigma)$, a *satisfaction* relation $\Vdash_{\mathbb{S}} \subseteq S \times L$. In case $(s, \varphi) \in \Vdash_{\mathbb{S}}$, we say that φ *is true* or *holds* at $s \in \mathbb{S}$, or that s *satisfies* φ in $\mathbb{S}$; we often write $\mathbb{S}, s \Vdash s$ or even $s \Vdash \varphi$ instead of $s \Vdash_{\mathbb{S}} \varphi$.

Given a state s in a coalgebra $\mathbb{S}$, we define $Th^L_{\mathbb{S}}(s) := \{\varphi \in L \mid \mathbb{S}, s \Vdash \varphi\}$. Conversely, given a formula $\varphi \in L$ and a coalgebra $\mathbb{S}$, we let $[\![\varphi]\!]^{\mathbb{S}}$ denote the set of states in $\mathbb{S}$ where φ holds, that is, $[\![\varphi]\!]^{\mathbb{S}} := \{s \in S \mid \mathbb{S}, s \Vdash \varphi\}$. If $Th^L_{\mathbb{S}}(s) = Th^L_{\mathbb{S}'}(s')$, we say that s and s' are *L-equivalent*, and we write $\mathbb{S}, s \equiv_L \mathbb{S}', s'$ (or simply $s \equiv_L s'$ if $\mathbb{S}$ and $\mathbb{S}'$ are understood).

Finally, we will call a formula *satisfiable* if it is satisfied at some state in some coalgebra, and *valid* if it holds at every state in every coalgebra. $\lhd$

In the same way that universal coalgebra tries to give an account of state-based evolving systems, *uniformly* in the coalgebra type T, research in coalgebraic logic has been directed towards a development of logical languages and derivation systems that are similarly uniform in the parameter T. Apart from uniformity, here are some other desiderata for a coalgebraic logic.

Definition 1.18 Let $(L, \Vdash)$ be a logic for a coalgebra type T. We say that this logic is *invariant (for behavioural equivalence)* if $\mathbb{S}, s \simeq \mathbb{S}', s'$ implies $Th^L_{\mathbb{S}}(s) = Th^L_{\mathbb{S}'}(s')$, and *expressive* if conversely, $Th^L_{\mathbb{S}}(s) = Th^L_{\mathbb{S}'}(s')$ implies that s and s' are behaviourally equivalent. The logic is *decidable* if there is an algorithm that decides, on input $\varphi \in L$, whether there is some pointed coalgebra satisfying φ. $\lhd$

In addition, for practical purposes one generally wants the logic to be *finitary* in the sense that formulas are finite objects. Other desirable properties of a coalgebraic logic include good model-theoretic behaviour, and the existence of a derivation system that is sound and complete for the collection of valid formulas.

With Kripke models as paradigmatic examples of coalgebra, and modal logic providing *the* bisimulation-invariant logic for Kripke models, it should come as no surprise that most coalgebraic logics can be seen as generalisations of basic modal logic in some sense. The literature on coalgebra witnesses different ways to generalise basic modal logic from Kripke structures to arbitrary systems; here we mention two approaches.

First, however, we briefly discuss the role of *proposition letters* in coalgebraic modal logic. Generalising the relation between Kripke models and Kripke frames, we introduce the notion of a T-model.

Definition 1.19 Let T be a set functor, and let Q be an arbitrary but fixed set of proposition letters. A T-*model* is a triple (S, σ, V) such that (S, σ) is a T-coalgebra, and $V : \mathsf{Q} \to PS$ is a *valuation*. A morphism between T-models $\mathbb{S} = (S, \sigma, V)$ and $\mathbb{S}' = (S', \sigma', V')$ is a coalgebra morphism $f : (S, \sigma) \to (S, \sigma')$ such that $s \in V(p)$ iff $fs \in V'(p)$, for all $s \in S$. $\lhd$

There are two natural ways to think about T-models: either as T-coalgebras extended with a Q-valuation, or as coalgebras for the functor $T_{\mathsf{Q}} := K_{P\mathsf{Q}} \times T$. (Clearly, in the latter case it would be more natural to represent the valuation V as its associated *colouring* $V^\flat : S \to P\mathsf{Q}$ given by $V^\flat(s) := \{p \in \mathsf{Q} \mid s \in V(p)\}$, cf. Example A.5.) In these notes we will generally take the first perspective, since it is more compatible with the perspective on proposition letters as *variables*. Nevertheless we will apply various coalgebraic definitions to T-models as if they were indeed $K_{P\mathsf{Q}} \times T$-coalgebras.

Let us now take a quick look at two of the approaches towards coalgebraic modal logic.

Example 1.20 In the first approach towards coalgebraic modal logic, which is completely parametric in the functor T, the set of formulas L is closed under the following clause, which introduces a modal operator ∇:

if $\alpha \in TX$ for some finite set X of formulas, then $\nabla\alpha$ is a formula.

In the case of the powerset functor ($T = P$), we can write, for instance, $\nabla\{\varphi_0, \varphi_1\}$, where φ_0 and φ_1 are formulas. The formula $\nabla\{\varphi_0, \varphi_1\}$ will be equivalent to $(\Diamond\varphi_0 \wedge \Diamond\varphi_1) \wedge \Box(\varphi_0 \vee \varphi_1)$. In general, the semantics of ∇ in a Kripke structure $\mathbb{S} = (S, \sigma)$ will be given as

$$\mathbb{S}, s \Vdash \nabla\alpha \text{ iff } (\sigma(s), \alpha) \in \overline{P}(\Vdash),$$

where $\overline{P}(\Vdash) \subseteq PS \times PL$ is the (Egli-Milner) *lifting*[2] of the binary satisfaction relation $\Vdash$.

As we will see, this approach generalises well to any set functor T that 'preserves weak pullbacks' — the point of this condition being that T preserves weak pullbacks iff its lifting $\overline{T}$ preserves relation composition.

The ∇-logic provided by the relation-lifting approach described in Example 1.20 may provide coalgebraic logics in a completely uniform way, but its unusual syntax makes it not easy to work with.

The second approach to coalgebraic logic provides coalgebraic logics with a more standard modal syntax. Here, the modalities of the language correspond to so-called *predicate liftings*, where an n-ary predicate lifting is a natural transformation $\breve{P}^n \xrightarrow{\cdot} \breve{P}T$. Here we confine ourselves to a few examples of such coalgebraic modalities.

Example 1.21 The standard interpretation of the modalities $\Diamond$ and $\Box$ in Kripke structures can be formulated as follows:

$$\begin{array}{lll} \mathbb{S}, s \Vdash \Box\varphi & \text{iff} & R(s) \subseteq [\![\varphi]\!] \\ \mathbb{S}, s \Vdash \Diamond\varphi & \text{iff} & R(s) \cap [\![\varphi]\!] \neq \varnothing. \end{array}$$

Example 1.22 *Monotone modal logic* is a variant of standard modal logic where formulas are interpreted in so-called monotone neighbourhood models. These are structures of the form $\mathbb{S} = (S, \sigma, V)$ where S is a set of states, V is a valuation, and σ is a map $S \to PPS$ that assigns to each state $s \in S$ a collection $\sigma(s) \subseteq PS$ of neighbourhoods. Here, each collection $\sigma(s)$ is required to be *upwards closed* in the sense that $X \in \sigma(s)$ implies $Y \in \sigma(s)$ for all Y with $X \subseteq Y \subseteq S$.

In these structures we may interpret the modalities $\Diamond$ and $\Box$ as follows:

$$\begin{array}{lll} \mathbb{S}, s \Vdash \Box\varphi & \text{iff} & [\![\varphi]\!] \in \sigma(s) \\ \mathbb{S}, s \Vdash \Diamond\varphi & \text{iff} & (S \setminus [\![\varphi]\!]) \notin \sigma(s). \end{array}$$

Using the upwards-closedness of $\sigma(s) \subseteq P(S)$ it is not hard to show that $\Box\varphi$ holds at s iff s has a neighbourhood $U \in \sigma(s)$ such that $\mathbb{S}, u \Vdash \varphi$, for each $u \in U$, whereas $\Diamond\varphi$ holds at s iff every neighbourhood $U \in \sigma(s)$ contains some point u where φ holds.

To see how monotone modal logic generalises standard modal logic, think of a Kripke model (S, R, V) as the neighbourhood model $(S, \widehat{\sigma}, V)$, where $\widehat{\sigma}(s) := \{X \in PS \mid R(s) \subseteq X\}$.

Example 1.23 Let (S, σ) be a Markov chain, that is, a coalgebra for the distribution functor D. Given a rational number $q \in [0, 1]$, we introduce a modality $\Diamond_q$, with the following intended meaning:

$$\mathbb{S}, s \Vdash \Diamond_q\varphi \text{ iff } \sum_{t \in [\![\varphi]\!]^{\mathbb{S}}} \mu_s(t) > q.$$

That is, the formula $\Diamond_q\varphi$ holds at s iff the probability that φ holds at the next state after s is bigger than q.

[2] The Egli-Milner lifting of a relation $R \subseteq S \times S'$ is the relation $\overline{P}(R) \subseteq PS \times PS'$ given by $(X, X') \in \overline{P}(R)$ iff for all $x \in X$ there is an $x' \in X'$ such that Rxx' and for all $x' \in X'$ there is an $x \in X$ such that Rxx'.

1.6 Literature

Here are some relevant texts on coalgebra and modal logic. First we mention some books:

- J. Barwise and L. Moss, *Vicious Circles*, CSLI Publications, 1996.
- B. Jacobs, *Introduction to Coalgebra: towards mathematics of states and observation*, Cambridge University Press, 2016.
- J. Rutten, *The Method of Coalgebra: exercises in coinduction*, CWI, Amsterdam, The Netherlands, 2019, ISBN 978-90-6196-568-8.

Here is a list of introductory and survey articles:

- B. Jacobs and J. Rutten, *A tutorial on (co)algebras and (co)induction*, Bulletin of the European Association for Theoretical Computer Science, 62 (1997), pp. 222–259..
- J.J.M.M Rutten, *Universal coalgebra: a theory of systems*, Theoretical Computer Science 249 (2000), pp. 3-80.
- H.P. Gumm, *Universelle coalgebra*, appendix to Th. Ihringer: Universelle Algebra, Heldermann Verlag, Berlin, 2003. (available from the web page of H.P. Gumm.)
- A. Kurz, A. Palmigiano and Y. Venema, *Coalgebra and logic: an overview*, Journal of Logic and Computation 20 (2010), pp. 985-988.
- C. Kupke and D. Pattinson, *Coalgebraic Semantics of Modal Logics: an Overview*, Theoretical Computer Science, 412(38) (2011), pp. 5070-5094.
- C. Cîrstea, A. Kurz, D. Pattinson, L. Schröder and Y. Venema, *Modal logics are coalgebraic*, The Computer Journal, 54 (2011), pp. 31-41.

2 Final Coalgebras and Coinduction

In section 1.3 we introduced final coalgebras. In this chapter we study the concept in more detail, and we see how it relates to the fundamental coalgebraic definition and proof principle of *coinduction*.

As a first example of a final coalgebra, it is instructive to look at a base category different from Set.

Example 2.1 Let $\mathbb{C} = (C, \leq)$ be an arbitrary poset, that is, $\leq$ is a reflexive, transitive and antisymmetric relation on the set C. We may think of $\mathbb{C}$ as a category by taking C as the set of objects and providing a unique arrow between any pair of elements $c, d \in C$ for which $c \leq d$. An *endofunctor* on $\mathbb{C}$ is then nothing but a *monotone* or *order-preserving* function $F : C \to C$. Given that arrows between objects are unique if they exist, a *coalgebra* $(c, \gamma : c \to Fc)$ for such a functor can be identified with its carrier c, and conversely, any $c \in C$ for which $c \leq Fc$ is the carrier of an F-coalgebra. In other words, we may identify F-coalgebras with the *prefixpoints* of F.

We leave it for the reader to verify that a final coalgebra for a functor F on $\mathbb{C}$ is a *greatest fixpoint* of the map F.

2.1 The language coalgebra

As a key example of a final coalgebra we will show how to endow the collection of *languages* over some finite alphabet C with coalgebra structure that turns it into the final coalgebra for the set functor $2 \times Id^C$ associated with deterministic automata.

Here we will represent a deterministic automaton over an alphabet C as a triple $\mathbb{S} = (S, \chi, \tau)$, where $\chi : S \to 2$ and $\tau : S \to S^C$ correspond to the acceptance condition and the transition map, respectively. Note that we drop the condition that the carrier of the automaton is finite. We will also use the convention that $s \xrightarrow{a} t$ means $t = \tau(s)(a)$ and $s{\downarrow}$ indicates that s is accepting, i.e., $\chi(s) = 1$.

As we saw in the introduction, we can identify deterministic automata with coalgebras of the functor $2 \times Id^C$. It is easy to see that a map $f : S \to S'$ is a coalgebra morphism between two automata $\mathbb{S} = (S, \chi, \tau)$ and $\mathbb{S}' = (S', \chi', \tau')$ if it satisfies, for all $s \in S$ and $a \in C$, the conditions $\chi(s) = \chi'(fs)$ and $f(\tau(s)(a)) = \tau'(fs)(a)$.

Definition 2.2 Consider the following *language coalgebra* $\mathbb{L} := (\mathcal{L}_C, \omega, \delta)$, where
• $\mathcal{L}_C := P(C^*)$ is the collection of all *languages* over C,
• $\omega : \mathcal{L}_C \to 2$ is given by $\omega(L) := 1$ if $\varepsilon \in L$, and $\omega(L) = 0$ if $\varepsilon \notin L$;
• $\delta : \mathcal{L}_C \to (\mathcal{L}_C)^C$ is the map given by $\delta(L)(a) := L_a$, the so-called *a-derivative* of L:

$$L_a := \{u \in C^* \mid au \in L\}.$$

If no confusion concerning the alphabet is likely, we will usually write $\mathcal{L}$ rather than $\mathcal{L}_C$. $\lhd$

Recall that for an arbitrary automaton $\mathbb{S} = (S, \chi, \tau)$, we defined the language recognized by a state $s \in S$ by putting

$$L_{\mathbb{S}}(s) := \{u \in C^* \mid \chi(\widehat{\tau}(s)(u)) = 1\},$$

where $\widehat{\tau} : S \to S^{C^*}$ is inductively defined by putting $\widehat{\tau}(s)(\varepsilon) := s$ and $\widehat{\tau}(s)(cu) := \widehat{\tau}\big(\tau(s)(c)\big)(u)$.

We claim that, for any alphabet C, $\mathbb{L}$ is the final coalgebra of type $2 \times Id^C$, with the language maps as the witnessing coalgebra morphisms.

Proposition 2.3 (Finality of $\mathbb{L}$) *For any $2 \times Id^C$-coalgebra $\mathbb{S}$, the map $L_{\mathbb{S}}$ is the unique coalgebra morphism $L_{\mathbb{S}} : \mathbb{S} \to \mathbb{L}$.*

Proof. Fix $\mathbb{S} = (S, \chi, \tau)$. We first show that $L_{\mathbb{S}}$ is a coalgebra morphism. For acceptance, we check that $\chi(s) = \omega(L_{\mathbb{S}}(s))$:

$$\begin{aligned} \chi(s) = 1 &\text{ iff } \varepsilon \in L_{\mathbb{S}}(s) && \text{(definition } L_{\mathbb{S}}) \\ &\text{ iff } \omega(L_{\mathbb{S}}(s)) = 1 && \text{(definition } \omega) \end{aligned}$$

With respect to the transition function, we need to show that $L_{\mathbb{S}}(\tau(s)(c)) = \delta(L_{\mathbb{S}}(s))(c)$, for all $s \in S$ and $c \in C$. But this identity holds because of the following chain of equivalences, for an arbitrary word $u \in C^*$:

$$\begin{aligned} u \in L_{\mathbb{S}}(\tau(s)(c)) &\text{ iff } \chi\big(\widehat{\tau}(\tau(s)(c))(u)\big) = 1 && \text{(definition } L_{\mathbb{S}}) \\ &\text{ iff } \chi(\widehat{\tau}(s)(cu))) = 1 && \text{(definition } \widehat{\tau}) \\ &\text{ iff } cu \in L_{\mathbb{S}}(s) && \text{(definition } L_{\mathbb{S}}) \\ &\text{ iff } u \in \delta(L_{\mathbb{S}}(s))(c) && \text{(definition } \delta) \end{aligned}$$

Second, we prove uniqueness. Assuming that $f : \mathbb{S} \to \mathbb{L}$ is a coalgebra morphism, we need to show that $f = L_{\mathbb{S}}$. It suffices to show that any word $u \in C^*$ satisfies the following:

$$\text{for all } s \in S: \quad u \in L_{\mathbb{S}}(s) \text{ iff } u \in f(s). \tag{3}$$

We will prove (3) by induction on u. In the base case, where $u = \varepsilon$, we have

$$\begin{aligned} \varepsilon \in L_{\mathbb{S}}(s) &\text{ iff } \omega(L_{\mathbb{S}}(s)) = 1 && \text{(definition } \omega) \\ &\text{ iff } \chi(s) = 1 && (L_{\mathbb{S}} \text{ is a morphism}) \\ &\text{ iff } \omega(f(s)) = 1 && (f \text{ is a morphism}) \\ &\text{ iff } \varepsilon \in f(s) && \text{(definition } \omega) \end{aligned}$$

Now assume that $u = cv$, for some $c \in C$, then we find

$$\begin{aligned} cv \in L_{\mathbb{S}}(s) &\text{ iff } \chi(\widehat{\tau}(s)(cv)) = 1 && \text{(definition } L_{\mathbb{S}}) \\ &\text{ iff } \chi\big(\widehat{\tau}(\tau(s)(c))(v)\big) = 1 && \text{(definition } \widehat{\tau}) \\ &\text{ iff } v \in L_{\mathbb{S}}(\tau(s)(c)) && \text{(definition } L_{\mathbb{S}}) \\ &\text{ iff } v \in f(\tau(s)(c)) && \text{(induction hypothesis)} \\ &\text{ iff } v \in \delta(f(s))(c) && (f \text{ is a morphism}) \\ &\text{ iff } cv \in f(s) && \text{(definition } \delta) \end{aligned}$$

This suffices to prove the induction step of (3). QED

2.2 Properties of final coalgebras

Final coalgebras have various interesting properties. We first show that, if existing, final coalgebras are unique modulo isomorphism[3]. Because of this fact we will often speak of 'the' final T-coalgebra if T admits final coalgebras.

Proposition 2.4 *Let $\mathbb{Z} = (Z, \zeta)$ and $\mathbb{Z}' = (Z', \zeta')$ be final T-coalgebras for some functor $T : \mathsf{C} \to \mathsf{C}$. Then $\mathbb{Z}$ and $\mathbb{Z}'$ are isomorphic.*

Proof. By finality of $\mathbb{Z}$ there is a coalgebra morphism $g : \mathbb{Z}' \to \mathbb{Z}$, and by finality of $\mathbb{Z}'$ there is a coalgebra morphism $f : \mathbb{Z} \to \mathbb{Z}'$. But then the composition $g \circ f : \mathbb{Z} \to \mathbb{Z}$ is a coalgebra morphism as well, and by unicity it must be identical to the identity arrow $\mathsf{id}_{\mathbb{Z}}$. Similarly, we find that $g \circ f = \mathsf{id}_{\mathbb{Z}'}$. Thus $\mathbb{Z}$ and $\mathbb{Z}'$ are isomorphic indeed. QED

The following proposition states a key fact about final coalgebras.

Proposition 2.5 (Lambek's Lemma) *Let $\mathbb{Z}$ be a final T-coalgebra for some functor $T : \mathsf{C} \to \mathsf{C}$. Then the coalgebra map $\zeta : Z \to TZ$ of $\mathbb{Z}$ is an isomorphism in C.*

Proof. Applying the functor T to the coalgebra map ζ of $\mathbb{Z}$, we obtain the map $T\zeta : TZ \to TTZ$, and hence, a coalgebra $\mathbb{Z}_2 := (TZ, T\zeta)$. By finality of $\mathbb{Z}$ we obtain a coalgebra morphism ! from $\mathbb{Z}_2$ to $\mathbb{Z}$, given by a C-arrow $! : TZ \to Z$. But then the composition $! \circ \zeta$ is a coalgebra morphism from $\mathbb{Z}$ to itself, just like the identity arrow id_Z. In a diagram:

$$
\begin{tikzcd}
Z \arrow[r, "\zeta"] \arrow[rr, bend left, "\mathsf{id}_Z"] \arrow[d, "\zeta"'] & TZ \arrow[r, "!"] \arrow[d, "T\zeta"] & Z \arrow[d, "\zeta"] \\
TZ \arrow[r, "T\zeta"'] & TTZ \arrow[r, "T!"'] & TZ
\end{tikzcd}
\tag{4}
$$

It follows by unicity that $! \circ \zeta = \mathsf{id}_Z$.

For the reverse composition $\zeta \circ !$ we have that $\zeta \circ ! = T! \circ T\zeta$ since ! is a morphism, cf. the right rectangle in the diagram above. But then we easily derive that $\zeta \circ ! = T(! \circ \zeta) = T\mathsf{id}_Z = \mathsf{id}_{TZ}$. In other words, $\zeta \circ !$ is the identity arrow on TZ.

Finally, since $! \circ \zeta = \mathsf{id}_Z$ and $\zeta \circ ! = \mathsf{id}_{TZ}$ we see that ζ is an isomorphism indeed, with ! as its inverse. QED

As an immediate corollary of this, we see that set functors involving the full powerset functor in a nontrivial way, will generally not admit a final coalgebra.

Corollary 2.6 *The categories of Kripke frames and of Kripke models do not admit final coalgebras.*

[3]See the appendix for the categorical definition of an isomorphism. In the case of coalgebras, two coalgebras $\mathbb{S}$ and $\mathbb{S}'$ are isomorphic if there are coalgebra morphisms $f : \mathbb{S} \to \mathbb{S}'$ and $g : \mathbb{S}' \to \mathbb{S}$ such that $g \circ f$ and $f \circ g$ are the identity arrows on $\mathbb{S}$ and $\mathbb{S}'$, respectively.

Proof. Recall that Kripke frames are coalgebras for the powerset functor P. Now suppose for contradiction that P admits a final coalgebra $\mathbb{Z} = (Z, \zeta)$. It would follow by Lambek's Lemma that ζ is a bijection between Z and its powerset; but this is impossible by Cantor's theorem. The case of Kripke models can be proved similarly. QED

The following proposition is one way to formalise the proof principle of *coinduction* — we shall come back to this.

Proposition 2.7 *Let $\mathbb{Z}$ be a final T-coalgebra for some set functor T. Then the relation $\simeq_{\mathbb{Z}}$ of behavioural equivalence on $\mathbb{Z}$ is the identity relation Δ_Z on Z:*

$$\simeq_{\mathbb{Z}} = \Delta_Z. \tag{5}$$

Proof. Suppose that z and z' are two states in $\mathbb{Z}$ that are behaviourally equivalent. In Remark 1.15 we saw that this implies that $\mathsf{beh}_{\mathbb{Z}}(z) = \mathsf{beh}_{\mathbb{Z}}(z')$, and since $\mathsf{beh}_{\mathbb{Z}}$ is the identity map on Z, it follows that $z = \mathsf{beh}_{\mathbb{Z}}(z) = \mathsf{beh}_{\mathbb{Z}}(z') = z'$. QED

2.3 Existence of final coalgebras

If not all set functors admit final coalgebras, which ones do? Some good sufficient conditions are known.

Definition 2.8 Let T be some set functor, and κ some cardinal. Call T *κ-small* if

$$T(S) = \bigcup\{(T\iota_S^A)[T(A)] \mid \iota_S^A : A \hookrightarrow S, |A| < \kappa\},$$

for all sets $S \neq \varnothing$, where for an arbitrary subset A of S, the arrow ι_S^A denotes the inclusion map of A into S. T is *small* if it is small for some cardinal κ. An ω-small functor is usually called *finitary.* $\lhd$

In words, the definition requires every element of $T(S)$ to be in the range of $T\iota$ for an appropriate inclusion map $\iota : A \hookrightarrow S$, where A is of size smaller than κ. In case T *preserves inclusions* (meaning that T maps any inclusion $\iota : A \hookrightarrow B$ to an inclusion $T\iota : TA \hookrightarrow TB$), the definition boils down to the requirement that

$$T(S) = \bigcup\{T(A) \mid A \subseteq S, |A| < \kappa\}.$$

Fact 2.9 *Every small set functor admits a final coalgebra.*

Examples of small functors abound; for instance, whenever we replace, in a Kripke polynomial functor, the power set functor by a bounded variant such as the finite power set functor, the result is a small functor.

In particular, the finite power set functor P_ω itself is ω-small. As an immediate corollary of this fact, the categories of image-finite frames and models, which can be represented as coalgebras for, respectively, the functors P_ω and $P\mathsf{Q} \times P_\omega$, both have final objects. More in general we can prove the following.

Corollary 2.10 *Every finitary Kripke polynomial functor admits a final coalgebra.*

For Set-based functors that do not admit a final coalgebra, there are various 'second-best' ways to proceed. For instance, one may show that T *does* have a final coalgebra in an *extended* or *modified* category.

Example 2.11 If one is willing to allow coalgebras taking a *class* rather than a set as their carrier, one may *create* a final coalgebra, outside the category Set, as follows. Let T be a set functor for which final coalgebras do not exist; for convenience we assume that all functors preserve inclusions.

Let SET be the category that has classes as objects, and class functions as arrows, that is, functions that map sets to sets but may have a class rather than a set as their (co-)domain. Call an endofunctor T on SET *set-based* if for each class C and each $X \in T\mathsf{C}$ there is a set $S \subseteq \mathsf{C}$ such that $X \in TS$. Now Aczel & Mendler proved that every set-based endofunctor on SET admits a final coalgebra – the similarity to Fact 2.9 is no coincidence.

This fact can be used as follows. Given an endofunctor T on Set, there is a *unique* way to extend T to a set-based endofunctor T^+ on SET. (On objects, simply put $T^+(\mathsf{C}) := \bigcup\{T(S) \mid S \subseteq \mathsf{C} \text{ a set}\}$.)

The theorem of Aczel & Mendler then guarantees the existence of a final object $\mathbb{Z}$ in $\mathsf{Coalg}(T^+)$. This coalgebra will be class-based if T does not admit a final coalgebra, but it will be final, not only with respect to the set-based coalgebras in $\mathsf{Coalg}(T^+)$, but also with respect to the class-based ones. As an important manifestation of this idea, Aczel showed that the class of non-well-founded sets provides the final coalgebra for (the SET-based extension of) the power set functor.

Example 2.12 One way to look at Lambek's Lemma is that final T-coalgebras provide solutions to the 'equation' $S \cong TS$. In the case of the powerset functor, Cantor's theorem states that this equation does not have a solution in the category Set. This situation is reminiscent of that in *domain theory*, which provides solutions to the equation $X \cong X^X$ by imposing topological structure on sets.

Something similar can be done here. Define a *Stone space* to be a pair $\mathbb{X} = (X, \tau)$, where τ is a zero-dimensional compact Hausdorff space, and let Stone denote the category of Stone spaces as objects with continuous maps as arrows. As an analog to the powerset functor on Set, we can define the *Vietoris* functor V on the category Stone; on objects, the Vietoris space $V\mathbb{X}$ is based on the collection of *compact* subsets of X. We may then show that, indeed, the final coalgebra for this functor exists.

Further on we will see that in modal logic, whereas the canonical model (over a finite set Q of proposition letters) is not final in the category of coalgebras for the Kripke model functor $K_{P\mathsf{Q}} \times P$, we may identify the canonical *general* model over Q with the final $K_{P\mathsf{Q}} \times V$-coalgebra. Thus we can solve the equation $S \cong TS$ by *modifying* the base category of our coalgebras.

2.4 The terminal sequence

Whether the functor admits a final coalgebra or not, one may always (try to) approximate it by considering the so-called final or terminal *sequence*.

Example 2.13 Let us first consider an example outside the category Set. Suppose that the poset $\mathbb{C} = (C, \leq)$ is in fact a complete lattice, that is, with each subset $X \subseteq C$ we may associate a *meet* or *greatest lower bound* $\bigwedge X$; this means in particular that $\mathbb{C}$ is bounded: it has a largest element $\top := \bigwedge \varnothing$, and a smallest element $\bot := \bigvee \varnothing$.

The Knaster-Tarski theorem states that in this setting, every monotone map $F : C \to C$ has both a least and a greatest fixpoint. That is, every endofunctor F on the *category* $\mathbb{C}$ admits both an initial and a final coalgebra (see Example 2.1).

It is instructive for our purposes to prove this theorem, and in particular, to see how to find the greatest fixpoint by approximating it from above. We define an ordinal-indexed sequence $\langle z_\alpha \rangle$ using transfinite induction:

$$z_\alpha := \begin{cases} \top & \text{if } \alpha = 0 \\ F z_\beta & \text{if } \alpha = \beta + 1 \text{ is a successor ordinal} \\ \bigwedge \{ z_\beta \mid \beta < \alpha \} & \text{if } \alpha \text{ is a limit ordinal} \end{cases}$$

Note that in fact, if we take 0 to be a limit ordinal, we can reduce the first clause of the definition to the third.

It is not hard to prove, for a monotone map $F : C \to C$, the existence of an ordinal α for which $z_\alpha = z_{\alpha+1}$, and to show that the object z_α is in fact the greatest fixpoint of F.

In the case of set functors, we may take inspiration from this example to define the *terminal sequence* associated with T.

Definition 2.14 The *final* or *terminal sequence* associated with a given set functor T, is an ordinal indexed sequence of objects $\langle Z_\alpha \rangle$ with maps $p^\alpha_\beta : Z_\alpha \to Z_\beta$ for $\beta \leq \alpha$, such that (i) $Z_{\alpha+1} = T Z_\alpha$ and $p^{\alpha+1}_{\beta+1} = T p^\alpha_\beta$, (ii) $p^\alpha_\alpha = \mathsf{id}_{Z_\alpha}$ and $p^\beta_\gamma \circ p^\alpha_\beta = p^\alpha_\gamma$, (iii) if λ is a limit ordinal, then Z_λ with $\{ p^\lambda_\alpha \mid \alpha < \lambda \}$ is a limit[4] of the diagram with objects $\{ Z_\alpha \mid \alpha < \lambda \}$ and arrows $\{ p^\alpha_\beta \mid \alpha, \beta < \lambda \}$. (In particular, taking 0 to be a limit ordinal, we find that $Z_0 = 1$ is some final object of the category Set, i.e., a singleton set.) $\lhd$

In the diagram below we depict an initial part of this construction:

$$\begin{tikzcd}
Z_0 = 1 & Z_1 = T1 \arrow[l, "p^1_0"'] & Z_2 = T^2 1 \arrow[l, "p^2_1"'] \arrow[ll, bend left, "p^2_0"] & {} \arrow[l] & \cdots & Z_\omega \arrow[lllll, bend right, "p^\omega_0"'] \arrow[llll, bend right, "p^\omega_1"'] \arrow[lll, bend right, "p^\omega_2"'] & Z_{\omega+1} = T Z_\omega \arrow[l, "p^{\omega+1}_\omega"'] & \cdots \arrow[l]
\end{tikzcd}$$

(6)

It is not hard to prove that, modulo isomorphism, the terminal sequence is uniquely determined by these conditions. Intuitively, it can be seen as an approximation of the final coalgebra for T. That is, where elements of the final coalgebra represent 'complete' behavior, elements of Z_α represent behavior that can be 'performed in α many steps'.

[4]See the appendix for the categorical definition of a limit.

To make this precise and formal, observe that for any T-coalgebra $\mathbb{S}$ there is a *unique* ordinal-indexed class of functions $\mathsf{beh}_\alpha : S \to Z_\alpha$ such that beh_0 is fixed by the finality of Z_0 in Set, $\mathsf{beh}_{\alpha+1} := (T\mathsf{beh}_\alpha) \circ \sigma$:

$$
\begin{tikzcd}
\cdots & Z_\alpha & & Z_{\alpha+1} \arrow[ll, "p^{\alpha+1}_\alpha"'] & \cdots \\
& S \arrow[u, "\mathsf{beh}_\alpha"] \arrow[rr, "\sigma"'] \arrow[urr, dotted, "\mathsf{beh}_{\alpha+1}"'] & & TS \arrow[u, "T\mathsf{beh}_\alpha"'] &
\end{tikzcd}
\tag{7}
$$

while for a limit ordinal λ, beh_λ is given as the unique map $\mathsf{beh}_\lambda : S \to Z_\lambda$ such that $\mathsf{beh}_\alpha = p^\lambda_\alpha \circ \mathsf{beh}_\lambda$ for all $\alpha < \lambda$. It is not hard to prove that, for instance, $\mathbb{S}, s \simeq \mathbb{S}', s'$ implies that $\mathsf{beh}_\alpha(s) = \mathsf{beh}_\alpha(s')$ for all α.

The relation with final coalgebras can be made precise, as follows. On the one hand, if the terminal sequence converges, in the sense that some arrow $p^{\alpha+1}_\alpha$ is a bijection, then the coalgebra $(Z_\alpha, (p^{\alpha+1}_\alpha)^{-1})$ is a final coalgebra for T. And conversely, under some mild constraints on T, Adámek & Koubek proved that if T admits a final coalgebra, then the terminal sequence converges to it.

2.5 Coinduction as a definition principle

Coinduction is an important coalgebraic principle, and just like its algebraic counterpart of *induction*, it can be used as a tool to *define* various operations, but also as a coalgebraic *proof* principle.

To see how coinductive definitions work, suppose that $\mathbb{Z} = (Z, \zeta)$ is the final coalgebra for some set functor T. Coinduction is based on the observation that, in order to define a map from some set S to Z, it suffices to turn S into a coalgebra by endowing it with coalgebra structure: any coalgebra map $\sigma : S \to TS$ canonically induces a map from S to Z, namely the unique coalgebra morphism $!_\sigma : (S, \sigma) \to \mathbb{Z}$. (In fact, we saw this principle already at work in the proof of Lambek's Lemma, where we defined a map from TZ to Z by considering the coalgebra map $T\zeta$ on TZ.)

Example 2.15 Let $\mathbb{Z} = (C^\omega, \mathsf{h}, \mathsf{t})$ be the *stream coalgebra* of Example 1.10 — here we write h and t rather than head and tail. We already saw that $\mathbb{Z}$ is the final $K_C \times Id$-coalgebra, we can now use the finality to define operations on streams.

To start with, consider the coalgebra map $\langle \mathsf{h}, \mathsf{t} \circ \mathsf{t} \rangle : C^\omega \to C \times C^\omega$ defined as $\gamma_e(\alpha) := (\mathsf{h}(\alpha), \mathsf{t}(\mathsf{t}(\alpha)))$. By finality of $\mathbb{Z}$ there is a unique map $\mathsf{e} : C^\omega \to C^\omega$ making the following diagram commute:

$$
\begin{tikzcd}
C^\omega \arrow[r, "\mathsf{e}"] \arrow[d, "\langle \mathsf{h}, \mathsf{t}\circ\mathsf{t} \rangle"'] & C^\omega \arrow[d, "\langle \mathsf{h}, \mathsf{t} \rangle"] \\
C \times C^\omega \arrow[r, "\mathsf{id}_C \times \mathsf{e}"'] & C \times C^\omega
\end{tikzcd}
\tag{8}
$$

Another way of looking at this definition is that, in order to define $\mathsf{e}(\alpha)$, we specfify

$$\begin{array}{rcl} \mathsf{h}(\mathsf{e}(\alpha)) & := & \mathsf{h}(\alpha) \\ \mathsf{t}(\mathsf{e}(\alpha)) & := & \mathsf{t}(\mathsf{t}(\alpha)). \end{array}$$

In fact, the map $\mathsf{e} : C^\omega \to C^\omega$ is the operation on streams that creates a new stream out of all items at an *even* position in the input stream. To prove this, it suffices to show that the map $\lambda\alpha.(\lambda n.\alpha(2n))$ makes the diagram (8) commute.

Similary, we can define a map $\mathsf{q} : C^\omega \to C^\omega$ selecting the *odd* items of an input stream, by means of the following diagram:

$$\begin{tikzcd} C^\omega \arrow[r, "\mathsf{q}"] \arrow[d, "\langle \mathsf{h}\circ\mathsf{t}, \mathsf{t}\circ\mathsf{t}\rangle"'] & C^\omega \arrow[d, "\langle \mathsf{h}, \mathsf{t}\rangle"] \\ C \times C^\omega \arrow[r, "\mathsf{id}_C \times \mathsf{q}"'] & C \times C^\omega \end{tikzcd} \tag{9}$$

Example 2.16 Fix an alphabet C. Recall from Proposition 2.3 that the language coalgebra $\mathbb{L} = (\mathcal{L}, \omega, \delta)$ is the final coalgebra for the 'automaton' functor $2 \times Id^C$. We can use this fact to define operations on languages.

For instance, given a word $u = c_1 \cdots c_k$ (with $k \geq 0$), we let $\otimes u$ denote its converse, $\otimes u := c_k \cdots c_1$, and we set $\otimes L := \{\otimes u \mid u \in L\}$. Coinductively, we can define this language by imposing the following structure on $\mathcal{L}$. As the acceptance condition we simply take the same map ω as for $\mathbb{L}$, while for the transition map τ we put

$$\tau(L)(c) := \{u \in C^* \mid uc \in L\}.$$

We leave it as an exercise for the reader to verify that the following diagram commutes:

$$\begin{tikzcd} \mathcal{L} \arrow[r, "\otimes"] \arrow[d, "\langle \omega, \tau\rangle"'] & \mathcal{L} \arrow[d, "\langle \omega, \delta\rangle"] \\ 2 \times \mathcal{L}^C \arrow[r, "\mathsf{id}_2 \times \otimes^C"'] & 2 \times \mathcal{L}^C \end{tikzcd} \tag{10}$$

so that by finality, $\otimes : \mathcal{L} \to \mathcal{L}$ is the *unique* coalgebra morphism $\otimes : (\mathcal{L}, \tau, \omega) \to \mathbb{L}$. That means that (10) can be seen as a coinductive *definition* of $\otimes$.

Example 2.17 As a slightly different example, we give a coinductive definition of the *shuffle* $K \,||\, L$ of two languages K and L. For an intuitive description of this operation[5] , a word belongs to $K \,||\, L$ if it consists of two interleaved words, from K and L, respectively. In particular, we want that

$$(K \,||\, L)_c = K_c \,||\, L \;\cup\; K \,||\, L_c,$$

[5]For a more precise definition of the language $K \,||\, L$, first define the relation $\lhd$ on finite words by putting $c_1 \cdots c_k \lhd d_1 \cdots d_n$ if there is an order-preserving map on the indices $f : \{1, \ldots, k\} \to \{1, \ldots, n\}$ such that $c_i = d_{fi}$, for all $i \in \{1, \ldots, k\}$. Say that w *merges* u and v if $|w| = |u| + |v|$ and both $u \lhd w$ and $v \lhd w$. We can then define $K \,||\, L$ as the collection of all words that merge words from K and L.

for all $c \in C$, where we recall that, for a language $L \in \mathcal{L}$, we let L_a denote its a-derivative, $L_a := \{u \in C^* \mid au \in L\}$.

For a coinductive definition, we consider the set $\mathcal{E}$ of expressions defined by the following grammar:

$$E \;::=\; \underline{L} \mid E_0 + E_1 \mid E_0 \,||\, E_1$$

where $L \in \mathcal{L}$, i.e., we associate a formal symbol $\underline{L}$ with every language L.

To turn the set $\mathcal{E}$ into an automaton $\mathbb{E} := (\mathcal{E}, \chi, \tau)$, consider the following axioms and rules (where we use the notation introduced in the beginning of section 2.1):

$$\frac{}{\underline{L}{\downarrow}}\,\varepsilon \in L \qquad \frac{E{\downarrow}}{(E+F){\downarrow}} \qquad \frac{F{\downarrow}}{(E+F){\downarrow}} \qquad \frac{E{\downarrow}\quad F{\downarrow}}{(E\,||\,F){\downarrow}}$$

$$\frac{}{\underline{L} \xrightarrow{c} \underline{L_c}} \qquad \frac{E \xrightarrow{c} E' \quad F \xrightarrow{c} F'}{E + F \xrightarrow{c} E' + F'} \qquad \frac{E \xrightarrow{c} E' \quad F \xrightarrow{c} F'}{E \,||\, F \xrightarrow{c} E' \,||\, F + E \,||\, F'}$$

It is not hard to see that this deductive system uniquely determines two operations, $\chi : \mathcal{E} \to 2$ and $\tau : \mathcal{E} \to \mathcal{E}^C$. Hence by finality of $\mathbb{L}$ there is a unique coalgebra morphism $f : \mathbb{E} \to \mathbb{L}$. Think of $f(E)$ as the *interpretation* of the term E. We can then define

$$K \,||\, L := f(\underline{K} \,||\, \underline{L}).$$

To get a feeling for this definition we compute the derivative $(K \,||\, L)_c$:

$$\begin{aligned}
(K \,||\, L)_c &= f(\underline{K} \,||\, \underline{L})_c && \text{(definition } ||\text{)}\\
&= \delta(f(\underline{K} \,||\, \underline{L}))(c) && \text{(definition } \delta\text{)}\\
&= f(\tau(\underline{K} \,||\, \underline{L})(c)) && (f \text{ is a morphism})\\
&= f(\underline{K_c} \,||\, \underline{L} + \underline{K} \,||\, \underline{L_c}) && \text{(definition } \tau\text{)}\\
&= f(\underline{K_c} \,||\, \underline{L}) \cup f(\underline{K} \,||\, \underline{L_c}) && (*)\\
&= K_c \,||\, L \cup K \,||\, L_c && \text{(definition } ||\text{)}
\end{aligned}$$

Here we use in (*) the observation that $f(E + F) = f(E) \cup f(F)$, which is easily proved by coinduction, see Example 2.20.

2.6 Coinduction as a proof principle

As a *proof* principle, coinduction has two manifestations. In its most direct form, proofs by coinduction use the uniqueness of coalgebra morphisms to a final coalgebra.

Example 2.18 Consider the maps e and q of Example 2.15. We claim that

$$\mathsf{q} = \mathsf{e} \circ \mathsf{t}, \tag{11}$$

as should be clear intuitively. To prove (11) coinductively, it suffices to prove that the map $\mathsf{e} \circ \mathsf{t}$, just like q, is a coalgebra morphism $\mathsf{e} \circ \mathsf{t} : (C^\omega, \langle \mathsf{h} \circ \mathsf{t}, \mathsf{t} \circ \mathsf{t}\rangle) \to (C^\omega, \langle \mathsf{h}, \mathsf{t}\rangle)$; that is, the diagram below commutes:

$$\begin{tikzcd} C^\omega \arrow[r, "\mathsf{e}\circ\mathsf{t}"] \arrow[d, "\langle \mathsf{h}\circ\mathsf{t}, \mathsf{t}\circ\mathsf{t}\rangle"'] & C^\omega \arrow[d, "\langle \mathsf{h}, \mathsf{t}\rangle"] \\ C \times C^\omega \arrow[r, "\mathsf{id}_C \times \mathsf{e}\circ\mathsf{t}"'] & C \times C^\omega \end{tikzcd} \tag{12}$$

But for this purpose it suffices to show that the following two equations hold:

$$\mathsf{h} \circ (\mathsf{e} \circ \mathsf{t}) = \mathsf{h} \circ \mathsf{t} \tag{13}$$

$$\mathsf{t} \circ (\mathsf{e} \circ \mathsf{t}) = (\mathsf{e} \circ \mathsf{t}) \circ (\mathsf{t} \circ \mathsf{t}) \tag{14}$$

This is not so hard. For (13), we may calculate

$$\begin{aligned} \mathsf{h} \circ (\mathsf{e} \circ \mathsf{t}) &= (\mathsf{h} \circ \mathsf{e}) \circ \mathsf{t} && \text{(associativity)} \\ &= (\mathsf{id}_C \circ \mathsf{h}) \circ \mathsf{t} && \text{(diagram (8))} \\ &= \mathsf{h} \circ \mathsf{t}, && (\mathsf{id}_C \circ \mathsf{h} = \mathsf{h}) \end{aligned}$$

while we prove (14) as follows:

$$\begin{aligned} \mathsf{t} \circ (\mathsf{e} \circ \mathsf{t}) &= (\mathsf{t} \circ \mathsf{e}) \circ \mathsf{t} && \text{(associativity)} \\ &= (\mathsf{e} \circ (\mathsf{t} \circ \mathsf{t})) \circ \mathsf{t} && \text{(diagram (8))} \\ &= (\mathsf{e} \circ \mathsf{t}) \circ (\mathsf{t} \circ \mathsf{t}) && \text{(associativity)} \end{aligned}$$

Example 2.19 In a similar way we can prove that

$$\mathsf{zip} \circ \langle \mathsf{e}, \mathsf{q} \rangle = \mathsf{id}_{C^\omega}, \tag{15}$$

where zip is the map defined in Example 1.12.

By finality it suffices to show that $\mathsf{zip} \circ \langle \mathsf{e}, \mathsf{q} \rangle$ is a coalgebra morphism on the stream coalgebra, and since we know that zip is a coalgebra morphism (i.e., the right rectangle below commutes), we can confine ourselves to proving that the left rectangle in the diagram below commutes:

$$\begin{tikzcd} C^\omega \arrow[r, "\langle \mathsf{e}, \mathsf{q}\rangle"] \arrow[d, "\langle \mathsf{h}, \mathsf{t}\rangle"'] & C^\omega \times C^\omega \arrow[r, "\mathsf{zip}"] \arrow[d, "\delta"] & C^\omega \arrow[d, "\langle \mathsf{h}, \mathsf{t}\rangle"] \\ C \times C^\omega \arrow[r, "T\langle \mathsf{e}, \mathsf{q}\rangle"'] & C \times (C^\omega \times C^\omega) \arrow[r, "T\mathsf{zip}"'] & C \times C^\omega \end{tikzcd}$$

Here $T\langle \mathsf{e}, \mathsf{q} \rangle = \mathsf{id}_C \times \langle \mathsf{e}, \mathsf{q} \rangle$, $T\mathsf{zip} = \mathsf{id}_C \times \mathsf{zip}$, and δ is as given in Example 1.12: $\delta(\alpha, \beta) := (\mathsf{h}(\alpha), (\beta, \mathsf{t}(\alpha)))$.

To verify that the left rectangle above commutes we need to check that $\delta \circ \langle \mathsf{e}, \mathsf{q} \rangle = T\langle \mathsf{e}, \mathsf{q} \rangle \circ \langle \mathsf{h}, \mathsf{t} \rangle$, which boils down to proving

(a) $\mathsf{h} \circ \mathsf{e} = \mathsf{h}$,

(b) $\mathsf{q} = \mathsf{e} \circ \mathsf{t}$

(c) $\mathsf{t} \circ \mathsf{e} = \mathsf{q} \circ \mathsf{t}$.

But we obtain (a) because $\mathsf{h} \circ \mathsf{e} = \mathsf{id}_C \circ \mathsf{h}$ (definition of e), and (b) was shown in the previous example, cf. (11). Finally, for (c), observe that $\mathsf{t} \circ \mathsf{e} = \mathsf{e} \circ (\mathsf{t} \circ \mathsf{t})$ by definition of e (diagram (8)), and $\mathsf{e} \circ (\mathsf{t} \circ \mathsf{t}) = \mathsf{q} \circ \mathsf{t}$ by associativity and (11).

Often, coinduction is referred to as the proof principle that uses the fact that behavioural equivalence is the identity relation on a final coalgebra (Proposition 2.7). More specifically, given the fact that bisimilarity implies behavioural equivalence (as we will see further on), one may prove two states in a final coalgebra to be identical if we can link them by a bisimulation.

Example 2.20 In the case of C-automata (coalgebras of type $2 \times Id^C$), a bisimulation on a coalgebra (S, χ, τ) is a relation $B \subseteq S \times S$ such that, whenever $(s_0, s_1) \in B$, we have
(acc) $\chi(s_0) = \chi(s_1)$ (that is: $s_0{\downarrow}$ iff $s_1{\downarrow}$), and
(nxt) $(\tau(s_0)(c), \tau(s_1)(c)) \in B$, for all $c \in C$.

Let us now prove the statement that (cf. Example 2.17)

$$f(\underline{L}) = L, \tag{16}$$

for every language $L \in \mathcal{L}$. By coinduction, it suffices to show that the relation

$$B := \{(f(\underline{L}), L) \mid L \in \mathcal{L}\} \tag{17}$$

is a bisimulation. We check the two conditions.

For (acc), we observe that $f(\underline{L}){\downarrow}$ (in $\mathbb{L}$) iff $\underline{L}{\downarrow}$ (in $\mathbb{E}$) since f is a coalgebra morphism. But we have $\underline{L}{\downarrow}$ in $\mathbb{E}$ iff $\varepsilon \in L$ by definition of acceptance in $\mathbb{E}$, and we have $L{\downarrow}$ (in $\mathbb{L}$) iff $\varepsilon \in L$ by definition of acceptance in $\mathbb{L}$. This suffices to prove (acc).

For (nxt) we need to show, for an arbitrary language $L \in \mathcal{L}$ and an arbitrary letter $c \in C$, that the pair $\big(\delta(f(\underline{L}), c), \delta(L, c)\big) \in B$. To that aim, observe that $\delta(f(\underline{L}), c) = f(\tau(\underline{L})(c)) = f(\underline{L_c})$, respectively since f is a morphism and by definition of τ. But since we have $\delta(L)(c) = L_c$ (by definition of δ), it is immediate that $\big(\delta(f(\underline{L}), c), \delta(L, c)\big) = \big(f(\underline{L_c}), L_c\big) \in B$. This finishes the proofs of (17) and (16).

Similarly, we can prove that

$$f(E_0 + E_1) = f(E_0) \cup f(E_1), \tag{18}$$

for all expressions E_0 and E_1, by showing that the relation

$$R := \Big\{\big(f(E_0 + E_1), f(E_0) \cup f(E_1)\big) \mid E_0, E_1 \in \mathcal{E}\Big\}$$

is a bisimulation on $\mathbb{L}$.

Finally, we can use coinduction to establish, in a relatively straightforward way, various useful properties of the operation $||$, such as commutativity, associativity, or distribution with respect to $+/\cup$.

3 Bisimilarity and Behavioural Equivalence

In section 1.4 of the Introduction we defined two coalgebraic notions of equivalence: behavioral equivalence and bisimilarity. In this chapter we discuss these notions in more detail.

3.1 Basic observations

Obviously, the first question is how the notions of behavioral equivalence and bisimilarity relate to each other. One direction is clear: bisimilarity is a sufficient condition for behavioral equivalence.

Proposition 3.1 *Let $T : \mathsf{Set} \to \mathsf{Set}$ be some functor, and let s_0 and s_1 be states of the T-coalgebras $\mathbb{S}_0$ and $\mathbb{S}_1$, respectively. Then $\mathbb{S}_0, s_0 \leftrightarroweq \mathbb{S}_1, s_1$ implies $\mathbb{S}_0, s_0 \simeq \mathbb{S}_1, s_1$.*

Proof. In the special case that T admits a final coalgebra, a very simple proof obtains. Assume that $\mathbb{S}_0, s_0 \leftrightarroweq \mathbb{S}_1, s_1$, and let $B \subseteq S_0 \times S_1$ with $\beta : B \to TB$ be a coalgebra witnessing this. It follows from the definitions that both $\mathsf{beh}_{\mathbb{S}_0} \circ \pi_0$ and $\mathsf{beh}_{\mathbb{S}_1} \circ \pi_1$ are coalgebraic morphisms from (B, β) to the final coalgebra, so from finality it follows that $\mathsf{beh}_{\mathbb{S}_0} \circ \pi_0 = \mathsf{beh}_{\mathbb{S}_1} \circ \pi_1$. From this it is immediate that $B \subseteq \simeq$; and so from $(s_0, s_1) \in B$ it follows that $\mathbb{S}_0, s_0 \simeq \mathbb{S}_1, s_1$.

In the general case the proof of this proposition is similar to the one of Theorem 3.13 below (with an application of *pushouts* instead of pullbacks), so we omit details. QED

The converse statement of Proposition 3.1 does not hold: in general, bisimilarity is a strictly stronger notion than behavioral equivalence.

Example 3.2 Consider the so-called '3-2-functor' $T_2^3 : \mathsf{Set} \to \mathsf{Set}$ given on objects by

$$T_2^3(S) := \big\{(s_0, s_1, s_2) \in S^3 \mid |\{s_0, s_1, s_2\}| \leq 2\big\},$$

while for an arrow $f : S \to S'$ we define $(T_2^3 f)(s_0, s_1, s_2) := (fs_0, fs_1, fs_2)$. We leave it as an exercise to the reader to verify that this indeed defines a set functor.

Now consider the following coalgebra $\mathbb{S} = (S, \sigma)$, where $S = \{0, 1\}$ and σ is given by $\sigma(0) = (0, 0, 1)$ and $\sigma(1) = (1, 0, 0)$. Then it is not hard to see that $\mathbb{S}, 0 \simeq \mathbb{S}, 1$, but at the same time we claim that there is *no* T_2^3-bisimulation on $\mathbb{S}$ linking 0 and 1. To see this, suppose for contradiction that $R \subseteq S \times S$ would be such a bisimulation, witnessed by the coalgebra map $\rho : R \to T_2^3 R$. If the projection maps $\pi_0, \pi_1 : R \to S$ are to be coalgebra morphisms, ρ has to map the pair $(0, 1)$ to some triple $\rho(0, 1) = \big((s_0, s_1), (t_0, t_1), (u_0, u_1)\big)$ such that

$$\begin{array}{lllllllll} (s_0, t_0, u_0) & = & ((T_2^3 \pi_0) \circ \rho)(0, 1) & = & (\sigma \circ \pi_0)(0, 1) & = & \sigma(0) & = & (0, 0, 1) \\ (s_1, t_1, u_1) & = & ((T_2^3 \pi_1) \circ \rho)(0, 1) & = & (\sigma \circ \pi_1)(0, 1) & = & \sigma(1) & = & (1, 0, 0). \end{array}$$

Clearly then we find $\rho(0, 1) = \big((s_0, s_1), (t_0, t_1), (u_0, u_1)\big) = \big((0, 1), (0, 0), (1, 0)\big)$. But this object does not belong to the set $T_2^3 R$, since (s_0, s_1), (t_0, t_1) and (u_0, u_1) are all distinct.

Example 3.3 A more natural example of a set functor for which behavioural equivalence and bisimilarity are properly distinct notions, is the *monotone neighbourhood functor* M. We will come back to this example later.

In section 3.3 below we will discuss an important class of set functors for which we do have $\simeq = \leftrightarroweq$. First, however, we make some basic observations on bisimulations and, in the next section, we give an alternative characterization of bisimulations.

Example 3.4 For an arbitrary set functor T, it is easy to see that for any coalgebra $\mathbb{S}$, the diagonal relation Δ_S is a bisimulation equivalence on $\mathbb{S}$. Furthermore, the converse of a bisimulation is again a bisimulation.

As another general example, coalgebra morphisms can be seen as functional bisimulations. To be more precise, let $f : S_0 \to S_1$ be a function between the carriers of two T-coalgebras $\mathbb{S}_0$ and $\mathbb{S}_1$. Recall that the *graph* of f is the relation $\mathsf{Gr} f := \{(s, f(s)) \mid s \in S_0\}$. Then it holds that

$$f \text{ is a coalgebra morphism iff its graph } \mathsf{Gr} f \text{ is a bisimulation.} \tag{19}$$

In order to see why this is so, first suppose that $\mathsf{Gr} f : \mathbb{S}_0 \leftrightarroweq \mathbb{S}_1$. Since the projection map $\pi_0 : \mathsf{Gr} f \to S_0$ is a bijective morphism, its inverse π_0^{-1} is also a morphism. But then $f = \pi_1 \circ \pi_0^{-1}$, as the composition of two morphisms, is also a morphism. For the other direction, suppose that f is a morphism; then it is straightforward to verify that the map $(T\pi_0)^{-1} \circ \sigma \circ \pi_0$ equips the set $\mathsf{Gr} f$ with the required coalgebraic structure.

However, the collection of bisimulations is not in general closed under taking relational composition, and the relation $\leftrightarroweq$ of bisimilarity on a given coalgebra is generally not an equivalence relation.

3.2 Bisimulations and relation lifting

Bisimulations admit an elegant alternative characterization which involves the notion of *relation lifting*.

Example 3.5 As an example, consider the power set functor P. Recall that a relation $B \subseteq S_0 \times S_1$ is a bisimulation between two P-coalgebras (Kripke frames) $\mathbb{S}_0 = (S_0, R_0[\cdot])$ and $\mathbb{S}_1 = (S_1, R_1[\cdot])$ iff B satisfies the conditions *(back)* and *(forth)* of Example 1.3. Now suppose that we define, for an arbitrary relation $R \subseteq S_0 \times S_1$, the relation $\overline{P}(R) \subseteq P(S_0) \times P(S_1)$ by putting

$$\overline{P}(R) := \{(Q_0, Q_1) \mid \forall q_0 \in Q_0\, \exists q_1 \in Q_1. (q_0, q_1) \in R \text{ and } \forall q_1 \in Q_1\, \exists q_0 \in Q_0. (q_0, q_1) \in R\}. \tag{20}$$

In other words, we *lift* the relation R to the level of the power sets of S_0 and S_1. The definition of a bisimulation between P-coalgebras can now be characterized as follows:

$$B : \mathbb{S}_0 \leftrightarroweq \mathbb{S}_1 \text{ iff } (R_0[s_0], R_1[s_1]) \in \overline{P}(B) \text{ for all } (s_0, s_1) \in B.$$

This nice way of characterizing bisimulation via relation lifting is not limited to the power set functor — it applies in fact to *every* set functor.

Definition 3.6 Let T be some set functor. Given a relation $R \subseteq S_0 \times S_1$, consider R as a *span*

$$S_0 \xleftarrow{\pi_0} R \xrightarrow{\pi_1} S_1 \ ,$$

where $\pi_i : R \to S_i$ and $p_i : TS_0 \times TS_1 \to TS_i$ denote the respective projection maps. We define the *relation lifting* of R as the relation $\overline{T}R \subseteq TS_0 \times TS_1$ given by

$$\overline{T}R := \{((T\pi_0)(u), (T\pi_1)(u)) \mid u \in TR\}, \tag{21}$$

that is, $\overline{T}R$ is the image of TR under the map $\tau_R := \langle T\pi_0, T\pi_1 \rangle$. $\lhd$

In other words, we apply the functor T to the relation R, seen as a span. It follows from the category-theoretic properties of the product $TS_0 \times TS_1$ that there is a unique map $\tau_R := \langle T\pi_0, T\pi_1 \rangle$ from TR to $TS_0 \times TS_1$ such that $p_i \circ \tau_R = T\pi_i$ for $i = 0, 1$. Now we define $\overline{T}R$ as the image of TR under the map τ_R obtained from the lifted projection maps $T\pi_0$ and $T\pi_1$. In a diagram:

$$\begin{tikzcd}
TS_0 & TR \arrow[l, "T\pi_0"'] \arrow[r, "T\pi_1"] \arrow[d, two heads] \arrow[dd, bend right, "\tau_R"'] & TS_1 \\
 & \overline{T}R \arrow[d, hook] & \\
 & TS_0 \times TS_1 \arrow[uul, "p_0"] \arrow[uur, "p_1"'] &
\end{tikzcd}$$

Remark 3.7 Strictly speaking, the definition of the relation lifting of a given relation depends on its type. That is, given sets A, B, A' and B', and a relation R such that $R \subseteq A \times B$ and and $R \subseteq A' \times B'$, for the definition of TR it matters whether we look at R as a relation from A to B or as a relation from A' to B'.

This possible source of ambiguity evaporates if we require the functor T to be standard, see the appendix for more details. In this chapter we will only use the lifting of a relation in a setting where its type is fixed, but readers who are worried about this issue may add the (fairly harmless) condition that T is standard.

The results listed in the following theorem summarize the most important properties of bisimulations.

Theorem 3.8 *Let $\mathbb{S}_0$ and $\mathbb{S}_1$ be two coalgebras for some set functor T.*

1. *For any set $B \subseteq S_0 \times S_1$, we have $B : \mathbb{S}_0 \underline{\leftrightarrow} \mathbb{S}_1$ iff $(\sigma_0(s_0), \sigma_1(s_1)) \in \overline{T}(B)$, all $(s_0, s_1) \in B$.*
2. *The collection of bisimulations between $\mathbb{S}_0$ and $\mathbb{S}_1$ forms a complete lattice under the inclusion order, with joins given by unions.*
3. *The bisimilarity relation $\underline{\leftrightarrow}$ is the largest bisimulation between $\mathbb{S}_0$ and $\mathbb{S}_1$.*

Proof. The first part of the theorem is an almost immediate consequence of the definitions. To see this, recall that $B : \mathbb{S}_0 \underline{\leftrightarrow} \mathbb{S}_1$ iff we can find a coalgebra map $\beta : B \to TB$ such that $(T\pi_i) \circ \beta = \sigma_i \circ \pi_i$ for $i = 0, 1$, and that the latter requirement is equivalent to stating that $(T\pi_i)(\beta(s_0, s_1)) = \sigma s_i$. From this it easily follows that $B : \mathbb{S}_0 \underline{\leftrightarrow} \mathbb{S}_1$ iff for every $(s_0, s_1) \in B$ there is a $u \in TB$ such that $(\sigma_0 s_0, \sigma_1 s_1) = ((T\pi_0)(u), (T\pi_1)(u))$. This suffices by (21).

The crucial observation in the proof of the other two parts is that

$$\overline{T} : P(S_0 \times S_1) \to P(TS_0 \times TS_1) \text{ is a monotone operation.} \tag{22}$$

For a proof, let $R \subseteq R'$ be two relations between S_0 and S_1, with $\iota : R \to R'$ denoting the inclusion map. By definition of $\overline{T}$, we may without loss of generality represent an arbitrary element of $\overline{T}(R)$ as a pair $\tau_R(u) = ((T\pi_0)(u), (T\pi_1)(u))$ for some $u \in TR$. Define $u' := (T\iota)(u)$, then u' belongs to TR', and for each i we find that $(T\pi'_i)(u') = (T\pi'_i \circ T\iota)(u) = T(\pi'_i \circ \iota)(u) = (T\pi_i)(u)$. That is, $\tau_R(u) = \tau_{R'}(u')$, which shows that $\tau_R(u)$ belongs to $\overline{T}R'$. This proves (22).

Now for the proof of part 2, recall that a partial order is a complete lattice if it is closed under arbitrary joins. Hence, it suffices to prove that the union B of a collection $\{B_j \mid j \in J\}$ of bisimulations is again a bisimulation. Take an arbitrary pair $(s_0, s_1) \in B$. Then (s_0, s_1) belongs to B_j for some $j \in J$. Hence, by part 1, we find $(\sigma_0 s_0, \sigma_1 s_1)$ in $\overline{T}(B_j)$, so $(\sigma_0(s_0), \sigma_1(s_1)) \in \overline{T}(B)$ by the monotonicity of $\overline{T}$. But then B is a bisimulation by part 1.

Finally, for part 3, note that it is an immediate consequence of part 2 that $\leftrightarroweq$, being the union of all bisimulations between $\mathbb{S}_0$ and $\mathbb{S}_1$, is a bisimulation itself. Hence, by definition, it is the greatest bisimulation between $\mathbb{S}_0$ and $\mathbb{S}_1$. In fact, it follows by the Knaster-Tarski theorem (on fixed points of monotone operations on complete lattices), that $\leftrightarroweq$ is in fact the greatest fixed point of the map $\Lambda : R \mapsto \{(s_0, s_1) \mid (\sigma_0(s_0), \sigma_1(s_1)) \in \overline{T}(R)\}$. QED

In the case of Kripke polynomial functors, relation lifting can be characterized using *induction* on the construction of the functor.

Proposition 3.9 *Let S and S' be two sets, and let $R \subseteq S \times S'$ be a binary relation between S and S'. Then the following induction defines the relation lifting $\overline{K}(R) \subseteq KS \times KS'$, for each Kripke polynomial functor K:*

$$\begin{array}{rcl}
\overline{Id}(R) & := & R, \\
\overline{K_C}(R) & := & \Delta_C, \\
\overline{K_0 \times K_1}(R) & := & \{((x_0, x_1), (x'_0, x'_1)) \mid (x_0, x'_0) \in \overline{K_0}(R) \textit{ and } (x_1, x'_1) \in \overline{K_1}(R)\}, \\
\overline{K_0 + K_1}(R) & := & \{(\kappa_0 x_0, \kappa_0 x'_0) \mid (x_0, x'_0) \in \overline{K_0}(R)\} \cup \{(\kappa_1 x_1, \kappa_1 x'_1) \mid (x_1, x'_1) \in \overline{K_1}(R)\}, \\
\overline{K^D}(R) & := & \{(f, f') \mid (f(d), f'(d)) \in \overline{K}(R) \textit{ for all } d \in D\}, \\
\overline{PK}(R) & := & \{(Q, Q') \mid \forall q \in Q\, \exists q' \in Q'.\, (q, q') \in \overline{K}(R) \textit{ and } \forall q' \in Q'\, \exists q \in Q.\, (q, q') \in \overline{K}(R)\}.
\end{array}$$

Here κ_0 and κ_1 are the co-projection maps associated with the coproduct, cf. Definition B.14.

3.3 Bisimilarity and behavioural equivalence: smooth functors

In Example 3.2 we saw that bisimilarity is a strictly stronger notion than behavioural equivalence. Here is a constraint on the functor that guarantees the two notions to coincide.

Definition 3.10 A *weak pullback* of two arrows $f_0 : A_0 \to B$, $f_1 : A_1 \to B$ in a category C is a pair of arrows $p_0 : W \to A_0$, $p_1 : W \to A_1$ such that (i) $f_0 \circ p_0 = f_1 \circ p_1$, while (ii) for every pair $p'_0 : W' \to A_0$, $p'_1 : W' \to A_1$ that also satisfies $f_0 \circ p'_0 = f_1 \circ p'_1$, there is a mediating arrow $w' : W' \to W$ such that $p_0 \circ w' = p'_0$ and $p_1 \circ w' = p'_1$. We will call a functor $T : \mathsf{C} \to \mathsf{C}'$ *smooth* if it preserves weak pullbacks; that is, if for any weak pullback (p_0, p_1) of any (f_0, f_1) in C, the pair (Tp_0, Tp_1) is a weak pullback of (Tf_0, Tf_1) in C'. $\lhd$

$$\begin{tikzcd}
 & A_0 \arrow[r, "f_0"] & B \\
 & W \arrow[u, "p_0"] \arrow[r, "p_1"] & A_1 \arrow[u, "f_1"'] \\
W' \arrow[uur, "p'_0"] \arrow[ur, dotted, "w'"] \arrow[urr, "p'_1"'] & &
\end{tikzcd}$$

Note that the mediating arrow w' need not be unique: adding this requirement to the definition would give the more familiar, and stronger, notion of a *pullback*. The category Set has pullbacks: for $f_0 : A_0 \to B$ and $f_1 : A_1 \to B$, we can take the projections to A_0 and A_1 from the set $\mathsf{pb}(f_0, f_1) := \{(a_0, a_1) \in A_0 \times A_1 \mid f_0(a_0) = f_1(a_1)\}$.

Many but not all endofunctors on Set in fact preserve weak pullbacks.

Proposition 3.11 *All polynomial functors preserve pullbacks, and all Kripke polynomial functors preserve weak pullbacks.*

The main reason that this prima facie rather exotic property is in fact of great importance in the theory of universal coalgebra, is the following fact.

Fact 3.12 *For any set functor T the following are equivalent:*
(1) T is smooth;
(2) $\overline{T}(R\,; Q) = \overline{T}R\,; \overline{T}Q$, for all pairs of relations $R \subseteq X \times Y$ and $Q \subseteq Y \times Z$.

Theorem 3.13 *If T is a smooth set functor, the following hold on the class of T-coalgebras:*
(1) the relational composition of two bisimulations is again a bisimulation;
(2) the notions of bisimilarity and behavioral equivalence coincide:

$$\simeq_T = \underline{\leftrightarrow}_T.$$

Proof. For the proof of the first statement, let $\mathbb{S}_0, \mathbb{S}_1$ and $\mathbb{S}_2$ be T-coalgebras, and let $B' : \mathbb{S}_0 \underline{\leftrightarrow} \mathbb{S}_1$ and $B'' : \mathbb{S}_1 \underline{\leftrightarrow} \mathbb{S}_2$ be bisimulations. We will show that $B := B'; B''$ is a bisimulation as well, between $\mathbb{S}_0$ and $\mathbb{S}_2$. For that purpose, take an arbitrary pair $(s_0, s_2) \in B$; then by definition there must be a state $s_1 \in S_1$ such that $(s_0, s_1) \in B'$ and $(s_1, s_2) \in B''$. But since B' and B'' are bisimulations, this means that $(\sigma_0 s_0, \sigma_1 s_1) \in \overline{T}B'$ and $(\sigma_1 s_1, \sigma_2 s_2) \in \overline{T}B''$, so that $(\sigma_0 s_0, \sigma_2 s_2) \in \overline{T}B'; \overline{T}B''$. Hence by smoothness we find that $(\sigma_0 s_0, \sigma_2 s_2) \in \overline{T}(B'; B'') = \overline{T}B$, as required.

Turning to the second statement, let s_0 and s_1 be states of the T-coalgebras $\mathbb{S}_0$ and $\mathbb{S}_1$, respectively. We need to prove that $\mathbb{S}_0, s_0 \underline{\leftrightarrow} \mathbb{S}_1, s_1$ iff $\mathbb{S}_0, s_0 \simeq \mathbb{S}_1, s_1$. Because of Proposition 3.1 it suffices to prove the direction from right to left.

Let $f_0 : \mathbb{S}_0 \to \mathbb{X}$ and $f_1 : \mathbb{S}_1 \to \mathbb{X}$ be two coalgebra morphisms such that $f_0(s_0) = f_1(s_1)$. Then in Set, the set $B := \{(s_0, s_1) \in S_0 \times S_1 \mid f_0(s_0) = f_1(s_1)\}$, together with the projection functions $\pi_0 : B \to S_0$ and $\pi_1 : B \to S_1$ constitutes a pullback of f_0 and f_1, cf. the square in the foreground of the picture. Because T preserves weak pullbacks, the square in the background of the picture is then a weak pullback diagram in Set.

$$
\begin{tikzcd}
 & TS_0 \arrow[rr, "Tf_0"] & & TX \\
S_0 \arrow[ur, "\sigma_0"] \arrow[rr, "f_0"] & & X \arrow[ur, "\xi"'] & \\
 & TB \arrow[uu, "T\pi_0"] \arrow[rr, "T\pi_1"'] & & TS_1 \arrow[uu, "Tf_1"'] \\
B \arrow[uu, "\pi_0"] \arrow[ur, dotted, "\beta"] \arrow[rr, "\pi_1"'] & & S_1 \arrow[uu, "f_1"'] \arrow[ur, "\sigma_1"'] &
\end{tikzcd}
$$

Now consider the two arrows $\sigma_i \circ \pi_i : B \to T(S_i)$. First observe that $Tf_i \circ \sigma_i = \xi \circ f_i$ for each i, because each f_i is a coalgebra morphism. Hence, chasing the diagram we find that

$$
\begin{aligned}
Tf_0 \circ \sigma_0 \circ \pi_0 &= \xi \circ f_0 \circ \pi_0 \\
&= \xi \circ f_1 \circ \pi_1 = Tf_1 \circ \sigma_1 \circ \pi_1.
\end{aligned}
$$

Since $T\pi_0$ and $T\pi_1$ form a weak pullback of Tf_0 and Tf_1, this implies the existence of a mediating function $\beta : B \to TB$ such that $T\pi_i \circ \beta = \sigma_i \circ \pi_1$. In other words, $\mathbb{B} := (B, \beta)$ is a T-coalgebra, and the projection maps π_0 and π_1 are morphisms from $\mathbb{B}$ to $\mathbb{S}_0$ and $\mathbb{S}_1$, respectively.

QED

Corollary 3.14 *Let T be a smooth functor. Then for any T-coalgebra $\mathbb{S}$, the maximal bisimulation $\Leftrightarrow$ on $\mathbb{S}$ is an equivalence relation.*

Proof. It is not hard to see that the maximal bisimulation $\Leftrightarrow$ is always reflexive and symmetric. For transitivity, it follows by smoothness of T that the composition $\Leftrightarrow ; \Leftrightarrow$ is again a bisimulation; but since $\Leftrightarrow$ is maximal we find that $\Leftrightarrow ; \Leftrightarrow \subseteq \Leftrightarrow$, which is just another formulation of the transitivity of $\Leftrightarrow$. QED

4 Covarieties

In universal *algebra* an important part is played by varieties: classes of algebras that are closed under the operations of taking homomorphic images, subalgebras and products of algebras. In this chapter, we introduce the notion of a *covariety* as a natural coalgebraic analog of a variety, and we consider some natural closure operations on classes of coalgebras.

4.1 Homomorphic images

Definition 4.1 Let T be some endofunctor on Set. If $f : \mathbb{S} \to \mathbb{S}'$ is a surjective coalgebra morphism between the T-coalgebras $\mathbb{S}$ and $\mathbb{S}'$, then we say that $\mathbb{S}'$ is a *homomorphic image* of $\mathbb{S}$, and write $\mathbb{S} \twoheadrightarrow \mathbb{S}'$, or $f : \mathbb{S} \twoheadrightarrow \mathbb{S}'$ if we want to make the morphism explicit. $\lhd$

In universal algebra, one finds a one-one correspondence between homomorphic images and congruences. Something similar applies here, but the analogy is perfect only in the case of functors that preserve weak pullbacks.

Proposition 4.2 *Let $\mathbb{S} = (S, \sigma)$ be a T-coalgebra for some set functor T.*

(1) Given a bisimulation equivalence[6] *E on $\mathbb{S}$, there is a unique coalgebra structure $\overline{\sigma}$ on S/E such that the quotient map $q : S \to S/E$ is a coalgebra morphism.*

(2) If T preserves weak pullbacks, then the relation $\ker(f) := \{(s,t) \in S^2 \mid fs = ft\}$ *is a bisimulation equivalence, for any coalgebra morphism $f : \mathbb{S} \to \mathbb{S}'$.*

Proof. For part (1), we leave it as an exercise for the reader to show that the set $\overline{S} = S/E$ of E-cells, together with the quotient map q, is a coequalizer of the projection maps $\pi_0, \pi_1 : E \to S$:

$$\begin{tikzcd} E \arrow[r, shift left, "\pi_0"] \arrow[r, shift right, "\pi_1"'] & S \arrow[r, "q"] & \overline{S} \end{tikzcd}$$

Now assume that, next to being an equivalence relation, E is also a bisimulation on $\mathbb{S}$. Then by definition there is a coalgebra map $\eta : E \to TE$ such that both π_i are coalgebra morphisms $\pi_i : (E, \eta) \to \mathbb{S}$. It follows that

$$\begin{array}{rcll} Tq \circ \sigma \circ \pi_0 & = & Tq \circ T\pi_0 \circ \eta & (\pi_0 \text{ is a morphism}) \\ & = & T(q \circ \pi_0) \circ \eta & (\text{functoriality}) \\ & = & T(q \circ \pi_1) \circ \eta & (q \text{ is a coequalizer}) \\ & = & Tq \circ T\pi_1 \circ \eta & (\text{functoriality}) \\ & = & Tq \circ \sigma \circ \pi_1 & (\pi_1 \text{ is a morphism}) \end{array}$$

In other words, the map $Tq \circ \sigma : S \to T\overline{S}$ is a competitor for the coequalizer map q, and so there is a unique map $\overline{\sigma} : \overline{S} \to T\overline{S}$ such that $\overline{\sigma} \circ q = Tq \circ \sigma$, in a diagram:

$$\begin{tikzcd} E \arrow[r, shift left, "\pi_0"] \arrow[r, shift right, "\pi_1"'] \arrow[d, "\eta"'] & S \arrow[r, "q"] \arrow[d, "\sigma"] & \overline{S} \arrow[d, dotted, "\overline{\sigma}"] \\ TE \arrow[r, shift left, "T\pi_0"] \arrow[r, shift right, "T\pi_1"'] & TS \arrow[r, "Tq"'] & T\overline{S} \end{tikzcd}$$

[6] A *bisimulation equivalence* is a bisimulation that is also an equivalence relation.

Clearly then $\overline{\sigma}$ is the required coalgebra map on $\overline{S}$.

For the second part of the proposition, observe that $\ker(f)$ is the relational composition of the graph of f with its converse. The result then follows from Theorem 3.13. QED

4.2 Subcoalgebras

The next class operation that we consider is that of taking subcoalgebras.

Definition 4.3 Let $\mathbb{X} = (X, \xi)$ and $\mathbb{S} = (S, \sigma)$ be two T-coalgebras, such that S is a subset of X. If the inclusion map $\iota : S \to X$ is a coalgebra morphism from (S, σ) to (X, ξ), then we say that S is *open* with respect to $\mathbb{X}$, and we call the structure (S, σ) a *subcoalgebra* of $\mathbb{X}$, writing $\mathbb{S} \leq \mathbb{X}$. $\lhd$

Interestingly enough, the transition map of a subcoalgebra is completely determined by the underlying open set.

Proposition 4.4 *Let $\mathbb{S}_0 = (S, \sigma_0)$ and $\mathbb{S}_1 = (S, \sigma_1)$ be two subcoalgebras of the coalgebra $\mathbb{X}$. Then $\sigma_0 = \sigma_1$.*

Proof. The case of S being empty is trivial, so suppose otherwise. Then from the assumption that $\mathbb{S}_0$ and $\mathbb{S}_1$ are subcoalgebras of $\mathbb{A}$, we may infer that $(T\iota) \circ \sigma_0 = \xi \circ \iota = (T\iota) \circ \sigma_1$, where ι is the inclusion map of S into X. It follows from the functoriality of T that $T\iota$ is an injection, so that we may conclude that $\sigma_0 = \sigma_1$. QED

Remark 4.5 In case T is a standard functor, we may simplify Definition 4.3: in this case any subcoalgebra of $\mathbb{X} = (X, \xi)$ must be of the form $\mathbb{S} = (S, \xi \restriction_S)$ with $S \subseteq X$ and $\xi(s) \in TS$, for all $s \in S$.

Example 4.6 (a) Given a Kripke frame (S, R) we call a structure (S', R') a *subframe* of $\mathbb{S}$ if $S' \subseteq S$ and $R' = R \cap (S' \times S')$. Such a subframe is *generated* if its universe S' is closed under the relation R, that is, if $s \in S'$ and $(s, t) \in R$ imply $t \in S'$. It is easy to see that the generated subframes of a frame correspond to the subcoalgebras of its coalgebraic representation.

(b) Similarly, a state b in a deterministic finite automaton $\mathbb{A} = (A, \delta, F)$ is *reachable* from a state $a \in A$ if there is some word u such that $\widehat{\delta}(a, u) = B$. Given some automaton $\mathbb{A}$, the states that are reachable from a given state a form an open set.

(c) The empty set is always open: every coalgebra has the empty coalgebra as a subcoalgebra.

Some further observations concerning subcoalgebras are in order. First of all, the topological terminology is justified by the following proposition.

Proposition 4.7 *Given a coalgebra $\mathbb{X}$ for some set functor T, the collection $\tau_{\mathbb{X}}$ of $\mathbb{X}$-open sets forms a topology.*

Proof. Closure of $\tau_{\mathbb{X}}$ under taking (arbitrary) unions follows from Theorem 3.8, together with the observation that

$$S \subseteq X \text{ is open with respect to } \mathbb{X} \text{ iff } \Delta_S \text{ is a bisimulation on } \mathbb{X}, \tag{23}$$

which in its turn is an immediate consequence of (19).

To prove that $\tau_{\mathbb{X}}$ is closed under taking finite intersections, assume that $\mathbb{A} = (A, \alpha)$ and $\mathbb{B} = (B, \beta)$ are two subcoalgebras of $\mathbb{S}$. The case where $A \cap B = \varnothing$ is trivial, so assume otherwise. Define $C := A \cap B$, fix some element $c \in C$, and consider the maps $f_c : A \to C$ and $g_c : S \to B$, given by

$$f_c(a) := \begin{cases} a & \text{if } a \in C \\ c & \text{otherwise} \end{cases} \qquad \text{and} \qquad g_c(s) := \begin{cases} s & \text{if } s \in B \\ c & \text{otherwise.} \end{cases}$$

Now consider the following diagram:

$$\begin{tikzcd}
& TA \arrow[rr, hook, "T\iota_S^A"] \arrow[dd, bend left, "Tf_c"] & & TS \arrow[dd, bend left, "Tg_c"] \\
A \arrow[ur, "\alpha"] \arrow[rr, hook, "\iota_S^A"] \arrow[dd, bend left, "f_c"] & & S \arrow[ur, "\sigma"'] \arrow[dd, bend left, "g_c"] & \\
& TC \arrow[uu, hook, "T\iota_A^C"] \arrow[rr, hook, "T\iota_B^C"'] & & TB \arrow[uu, hook, "T\iota_S^B"] \\
C \arrow[uu, hook, "\iota_A^C"] \arrow[ur, dotted, "\gamma"] \arrow[rr, hook, "\iota_B^C"'] & & B \arrow[uu, hook, "\iota_S^B"] \arrow[ur, "\beta"'] &
\end{tikzcd}$$

We define the map $\gamma : C \to TC$ by putting

$$\gamma := Tf_c \circ \alpha \circ \iota_A^C,$$

and claim that the resulting structure (C, γ) is a subcoalgebra of $\mathbb{S} = (S, \sigma)$. For this purpose we have to show that

$$\sigma \circ \iota_S^C = T\iota_S^C \circ \gamma. \tag{24}$$

We leave it as an exercise for the reader to show (24), using equalities such as $\iota_B^C \circ f_c = g_c \circ \iota_S^A$ and $g_c \circ \iota_S^B = \mathsf{id}_B$. QED

Remark 4.8 In the case where T is standard, the proof that $\tau_{\mathbb{X}}$ is closed under taking finite intersections can again be simplified. In order to prove that $C := A \cap B$ is the carrier of a subcoalgebra of $\mathbb{S}$, it suffices to show that $\sigma(c) \in TC$ for an arbitrary element c of $C := A \cap B$. But since T is standard we have $TC = TA \cap TB$, and we find $\sigma(c) \in TA \cap TB$ by the assumptions on A and B, respectively.

It follows from Proposition 4.7 that the partial order $(\tau_{\mathbb{X}}, \subseteq)$, consisting of the collection of open subsets of X ordered by set inclusion actually forms a *complete lattice*. In this lattice, the *join* of a collection $\mathcal{U}$ of opens is simply given as its union, but its *meet*, that is, the largest open set contained in each open $U \in \mathcal{U}$, is not necessarily its intersection.

In particular, given a subset S of X, there is an open set $V \subseteq X$ which is the *meet* of the collection $\{U \in \tau_{\mathbb{X}} \mid S \subseteq U\}$. However, there is no guarantee that V is the *intersection* of this collection, or, indeed, that S is actually a subset of V, as is witnessed by the example below. This example thus shows that it does not make sense to speak of the smallest subcoalgebra containing a given subset, or of *point-generated* subcoalgbras.

It follows from Proposition 4.7 that, given a subset S of (the carrier of) a coalgebra $\mathbb{X}$, there is a largest subcoalgebra $\langle S \rangle_{\mathbb{X}}$ of $\mathbb{X}$ (of which the carrier is) contained in S: Its carrier is given as the union of all open subsets of S. Note however, that this carrier might be empty, even if S is not.

Example 4.9 Consider the standard Euclidean topology on the real numbers, seen as a coalgebra for the *filter functor* F. This functor is a subfunctor of the (monotone) neighborhood functor which maps a set S to the collection of all *filters* on S[7] and a function $f : S \to S'$ simply to the function $Mf = \breve{P}\breve{P}f$. Prime examples of F-coalgebras are the topological spaces. To see this, represent a topology σ on the set S by the function mapping a point $s \in S$ to the collection $\{U \in \sigma \mid s \in U\}$ of its neighborhoods.

One can show that a set S of reals is open in the topological sense iff it is open in the sense of Definition 4.3 — in fact, this holds for any topology. Now take an arbitrary point r in $\mathbb{R}$. Obviously, the *meet* of all open neighborhoods containing r is the empty set. This example thus shows that in general it does not make sense to speak of point-generated subcoalgebras.

Before we turn to further coalgebraic constructions, consider the following natural link between homomorphic images and subcoalgebras.

Proposition 4.10 *Given a coalgebra morphism $f : \mathbb{S} \to \mathbb{S}'$, there is a (unique) subcoalgebra $f[\mathbb{S}]$ of $\mathbb{S}'$ such that $f : \mathbb{S} \to f[\mathbb{S}]$ is a surjective morphism.*

Proof. For a proof of this proposition, let $X := f[S]$ be the (set-theoretic) image of S under f, and let $g : X \to S$ be a right inverse of f, that is, $f(g(x)) = x$ for all $x \in X$. Now define $\xi : X \to TX$ by $\xi := Tf \circ \sigma \circ g$. It can be shown that the resulting structure $\mathbb{X}$ is always a subcoalgebra of $\mathbb{S}'$, and that $f : \mathbb{S} \to \mathbb{X}$ is a surjective morphism; further details are left for the reader. QED

4.3 Sums and other colimits

Our last example of a coalgebraic construction concerns the straightforward generalization of the disjoint union of Kripke models and frames. The idea is embodied in the following Proposition.

[7] Recall that a filter on S is a collection $\mathcal{F}$ of subsets of S which is not only upward closed (with respect to the inclusion relation), but also closed under taking (finite) intersections, that is, $X \cap Y \in \mathcal{F}$ if $X, Y \in \mathcal{F}$.

Proposition 4.11 *Let $\mathbb{S}_0 = (S_0, \sigma_0)$ and $\mathbb{S}_1 = (S_1, \sigma_1)$ be two T-coalgebras, and let $S := S_0 \uplus S_1$ be the disjoint union of S_0 and S_1. Then there is a unique arrow $\sigma : S \to TS$ making the embeddings κ_i into coalgebra morphisms.*

Proof. Consider the diagram below, where S, together with the embedding maps κ_0 and κ_1, is the coproduct of S_0 and S_1. Since the maps $T\kappa_0 \circ \sigma_0$ and $T\kappa_1 \circ \sigma_1$ provide an alternative co-cone, there must be a (unique) mediating arrow $\sigma : S \to TS$, making the two rectangles in the diagram commute.

$$\begin{tikzcd} S_0 \arrow[r, "\kappa_0"] \arrow[d, "\sigma_0"'] & S \arrow[d, dotted, "\sigma"] & S_1 \arrow[l, "\kappa_1"'] \arrow[d, "\sigma_1"] \\ TS_0 \arrow[r, "T\kappa_0"'] & TS & TS_1 \arrow[l, "T\kappa_1"] \end{tikzcd} \tag{25}$$

Clearly then this σ meets the requirements stated in the Proposition. QED

The sum of an arbitrary collection of coalgebras is defined as follows.

Definition 4.12 The *sum* $\coprod_I \mathbb{S}_i$ of a family $\{\mathbb{S}_i \mid i \in I\}$ of coalgebras for some set functor T, is defined by endowing the disjoint union $S := \biguplus_I S_i$ with the unique map $\sigma : S \to TS$ which turns all embeddings $\kappa_i : S_i \to S$ into coalgebra morphisms. $\lhd$

Sums are coproducts in the category of coalgebras.

Proposition 4.13 *Let $\{\mathbb{S}_i \mid i \in I\}$ be a family of coalgebras for some set functor T. Then $\coprod_I \mathbb{S}_i$ is a coproduct of $\{\mathbb{S}_i \mid i \in I\}$ in the category $\mathsf{Coalg}(T)$.*

In particular, Proposition 4.13 implies that coproducts actually *exist* (in the category of coalgebras for a given set functor T).

Remark 4.14 This result can be in fact be generalised to *arbitrary* colimits: it can be shown that for every set functor T, the category $\mathsf{Coalg}(T)$ has all colimits, and that these colimits are in fact based on the underlying colimits n Set.

4.4 Covarieties

We have now gathered all the basic class operations needed to define the notion of a covariety.

Definition 4.15 Let T be some endofunctor on Set. A class of T-coalgebras is a *covariety* if it is closed under taking homomorphic images, subcoalgebras and sums. The smallest covariety containing a class K of T-coalgebras is called the covariety *generated* by K, notation: $\mathsf{Covar}(\mathsf{K})$. $\lhd$

As in the case of universal algebra, in order to obtain a more succinct characterization of the covariety generated by a class of coalgebras, one may develop a calculus of class operations.

Definition 4.16 Let H, S and Σ denote the class operations of taking (isomorphic copies of) homomorphic images, subcoalgebras, and sums, respectively. $\lhd$

On the basis of these (and other) operations one may investigate the validity of 'inequalities' like $\mathsf{HS} \leq \mathsf{SH}$ (meaning that $\mathsf{HS}(\mathsf{K}) \subseteq \mathsf{SH}(\mathsf{K})$ for all classes K of coalgebras). We first prove that each of the three operations H, S and Σ is a *closure operation*; in each case the proof of this observation is straightforward.

Proposition 4.17 *Let* C *be one of the class operations* H, S *or* Σ, *and let* K *and* K' *be arbitrary classes of* T*-coalgebras. Then we find*

1) $\mathsf{K} \subseteq \mathsf{C}(\mathsf{K})$;
2) $\mathsf{K} \subseteq \mathsf{K}'$ *implies* $\mathsf{C}(\mathsf{K}) \subseteq \mathsf{C}(\mathsf{K}')$;
3) $\mathsf{C}(\mathsf{C}(\mathsf{K})) \subseteq \mathsf{C}(\mathsf{K})$.

In the following proposition we list some relevant inequalities involving class operations.

Proposition 4.18 *Let* K *be an arbitrary class of* T*-coalgebras. Then we find*

1) $\mathsf{HS}(\mathsf{K}) \subseteq \mathsf{SH}(\mathsf{K})$;
2) $\Sigma\mathsf{S}(\mathsf{K}) = \mathsf{S}\Sigma(\mathsf{K})$;
3) $\Sigma\mathsf{H}(\mathsf{K}) \subseteq \mathsf{H}\Sigma(\mathsf{K})$.

Proof. For part (1), let $\mathbb{A}$ be a coalgebra in $\mathsf{HS}(\mathsf{K})$. Then there is a coalgebra $\mathbb{C}$ in K, and a coalgebra $\mathbb{B}$, with morphisms e and f such that $e : \mathbb{B} \rightarrowtail \mathbb{C}$ and $f : \mathbb{B} \twoheadrightarrow \mathbb{A}$. Now consider the following push-out diagram in $\mathsf{Coalg}(T)$ (cf. Example B.20):

$$\begin{tikzcd} \mathbb{B} \arrow[r, two heads, "f"] \arrow[d, "e"'] & \mathbb{A} \arrow[d, "p_A"] \\ \mathbb{C} \arrow[r, "p_C"'] & \mathbb{P} \end{tikzcd}$$

Since (P, p_A, p_C) is also the push-out of (B, e, f) in Set, it follows that the maps p_A and p_C are, respectively, surjective and injective.[8] But then we immediately find that $\mathbb{P} \in \mathsf{H}(\mathsf{K})$ and $\mathbb{A} \in \mathsf{SH}(\mathsf{K})$, as required.

We leave the proof of part (2) as an exercise for the reader. For part (3), assume that we have $\mathbb{B} \in \Sigma\mathsf{H}(\mathsf{K})$. That is, there is a family of algebras $\{\mathbb{A}_i \mid i \in I\}$ in K, and a family of surjective morphisms $\{f_i : \mathbb{A}_i \twoheadrightarrow \mathbb{B}_i \mid i \in I\}$ such that $\mathbb{B} = \Sigma_i \mathbb{A}_i$, with the injections $\kappa_i : \mathbb{B}_i \rightarrowtail \mathbb{B}$.

Let $\mathbb{A}$ be the sum of the coalgebras $\mathbb{A}_i$, with the morphisms $\lambda_i : \mathbb{A}_i \rightarrowtail \mathbb{A}$. Then the coalgebra $\mathbb{B}$, with the morphisms $\{\kappa_i \circ f_i : \mathbb{A}_i \to \mathbb{B}_i\}$, is a competitor cone of $\mathbb{A}$, and so there is a unique morphism $f : \mathbb{A} \to \mathbb{B}$ such that $f \circ \lambda_i = \kappa_i \circ f_i$, for all i:

$$\begin{tikzcd} \mathbb{A}_i \arrow[r, two heads, "f_i"] \arrow[d, "\lambda_i"'] & \mathbb{B} \arrow[d, "\kappa_i"] \\ \mathbb{A} \arrow[r, "f"'] & \mathbb{B} \end{tikzcd}$$

[8] Some justification is needed here.

It remains to show that f is surjective. For that purpose, take an arbitrary state $b \in \mathbb{B}$, then b must be of the form $b = \kappa_i(b_i)$ for some $b_i \in B_i$. By surjectivity of f_i there is a state $a \in A_i$ such that $b_i = f_i a_i$. But then we find that $b = \kappa_i f_i a_i = f \lambda_i a_i$, which obviously implies that b lies in the range of f. So f is surjective indeed. QED

The above results easily lead to the following coalgebraic analog of Tarski's HSP-theorem in universal algebra.

Theorem 4.19 *Let K be a class of T-coalgebras for some set functor T. Then*

$$\mathsf{Covar}(\mathsf{K}) = \mathsf{SH\Sigma}(\mathsf{K}).$$

Proof. It is an immediate consequence of the definitions that $\mathsf{SH\Sigma}(\mathsf{K}) \subseteq \mathsf{Covar}(\mathsf{K})$. For the opposite inclusion it suffices to show that the class $\mathsf{SH\Sigma}(\mathsf{K})$ is closed under the operations S, H and $\Sigma(\mathsf{K})$, but this is rather obvious by the Propositions 4.17 and 4.18. For instance, we easily verify that $\Sigma(\mathsf{SH\Sigma}(\mathsf{K})) = \mathsf{S\Sigma H\Sigma}(\mathsf{K}) \subseteq \mathsf{SH\Sigma\Sigma}(\mathsf{K}) = \mathsf{SH\Sigma}(\mathsf{K})$. QED

5 Coalgebraic modalities via relation lifting

In this chapter we take an approach to coalgebraic logic which is completely uniform in the type functor T. We introduce a coalgebraic modality ∇ of which the 'arity' is the finitary version T_ω of the functor itself. That is, the set L of formulas will be closed under the following clause:

$$\text{if } \alpha \in TX \text{ for some finite set } X \text{ of formulas, then } \nabla\alpha \text{ is a formula.}$$

whereas the semantics of ∇ will be defined by *lifting* the satisfaction relation $\Vdash$ between states and formulas to the relation $\overline{T}(\Vdash)$.

In the special case where T is the powerset functor P, the nabla operator ∇ is known under the name of the *cover modality*; we discuss this case in some detail before moving on to the more general case.

Convention 5.1 Throughout this chapter we will assume that T is a smooth and standard set functor; that is, T preserves both weak pullbacks and inclusions. The first restriction is to ensure optimal behaviour of the relation lifting $\overline{T}$, while the second one is mainly for convenience. In the Facts B.36 and B.39 we list a number of properties of the operation $\overline{T}$ (all of which will be used throughout this chapter).

Furthermore, we will assume that Q is an arbitrary but fixed set of proposition letters.

5.1 The cover modality

As we will see now, there is an interesting coalgebraic alternative for the standard formulation of basic modal logic in terms of boxes and diamonds. This alternative set-up is based on a connective ∇, sometimes referred to as the *cover* modality, which turns a *(finite) set* α of formulas into a formula $\nabla\alpha$.

Definition 5.2 Formulas of the language $\mathrm{ML}_\nabla(\mathsf{Q})$ are given by the following recursive definition:

$$a ::= p \mid \bot \mid \neg a \mid a_0 \vee a_1 \mid \nabla\alpha$$

where $p \in \mathsf{Q}$, and α denotes a finite set of formulas. ◁

Observe that formulas will be denoted by lower case letters $a, b, \ldots$

For the semantics of the cover modality, observe that we may think of the forcing or satisfaction relation $\Vdash$ simply as a binary relation between states and formulas. This relation can thus be lifted to a relation $\overline{P}(\Vdash)$ between *sets* of formulas and *sets* of states.

Definition 5.3 The semantics of this modality in a Kripke model $\mathbb{S} = (S, R, V)$ is given by

$$\mathbb{S}, s \Vdash \nabla\alpha \text{ iff } (R(s), \alpha) \in \overline{P}(\Vdash),$$

where $\overline{P}(\Vdash)$ denotes the Egli-Milner relation lifting of the relation $\Vdash$. ◁

In words: $\nabla\alpha$ holds at s iff every successor of s satisfies some formula in α, and every formula in α holds at some successor of s. The modality ∇ is sometimes called the *cover modality*: it holds at a state s if the set $\{[\![a]\!]^{\mathbb{S}} \mid a \in \alpha\}$ covers the collection $R(s)$ of successors of s, in the sense that $R(s) \subseteq \bigcup\{[\![a]\!]^{\mathbb{S}} \mid a \in \alpha\}$, while at the same time $R(s) \cap [\![a]\!]^{\mathbb{S}} \neq \varnothing$, for every $a \in \alpha$.

Remark 5.4 It is not so hard to see that the cover modality can be defined in the standard modal language:

$$\nabla\alpha \equiv \Box \bigvee \alpha \wedge \bigwedge \Diamond\alpha, \tag{26}$$

where $\Diamond\alpha$ denotes the set $\{\Diamond a \mid a \in \alpha\}$. Things start to get interesting once we realize that both the ordinary diamond $\Diamond$ *and* the ordinary box $\Box$ can be expressed in terms of the cover modality (and the disjunction):

$$\begin{array}{lcl} \Diamond a & \equiv & \nabla\{a, \top\}, \\ \Box a & \equiv & \nabla\varnothing \vee \nabla\{a\}. \end{array} \tag{27}$$

Here, as always, we use the convention that $\bigvee \varnothing = \bot$ and $\bigwedge \varnothing = \top$.

Given that ∇ and $\{\Diamond, \Box\}$ are mutually expressible, we arrive at the following proposition. Here we say that two languages are *effectively equi-expressive* if there are effectively definable truth-preserving translations from one language to the other, and vice versa. Recall that ML is the language of standard modal logic.

Proposition 5.5 *The languages* ML *and* ML_∇ *are effectively equi-expressive.*

A remarkable observation about the cover modality is that we can do far better than this: based on the following *modal distributive law*, we can almost completely eliminate the Boolean connective of conjunction from the language ML_∇.

Proposition 5.6 *Let α and α' be two sets of formulas. Then the following two formulas are equivalent:*

$$\nabla\alpha \wedge \nabla\alpha' \ \equiv \bigvee_{Z \in \alpha \bowtie \alpha'} \nabla\{a \wedge a' \mid (a, a') \in Z\}, \tag{28}$$

where $\alpha \bowtie \alpha'$ is the set of all binary relations $Z \subseteq \alpha \times \alpha'$ such that $(\alpha, \alpha') \in \overline{P}(Z)$.

Proof. For the direction from left to right, suppose that $\mathbb{S}, s \Vdash \nabla\alpha \wedge \nabla\alpha'$. Let $Z \subseteq \alpha \times \alpha'$ consist of those pairs (a, a') such that the conjunction $a \wedge a'$ is true at some successor t of s. It is then straightforward to verify that Z is full on α and α', that is: $(\alpha, \alpha') \in \overline{P}(Z)$, and that $\mathbb{S}, s \Vdash \nabla\{a \wedge a' \mid (a, a') \in Z\}$.

The converse direction is a fairly direct consequence of the definitions. QED

As a corollary of Proposition 5.6 we can restrict the use of conjunction in modal logic to that of a *special conjunction* connective $\bullet$ which may only be applied to a propositional formula and a ∇-formula.

Definition 5.7 We first define the set $\mathrm{CL}(\mathsf{Q})$ of *literal conjunctions* by the following grammar:

$$\pi \ ::= \ p \mid \neg p \mid \bot \mid \top \mid \pi \wedge \pi,$$

and then let the following grammar define the set $\mathrm{DML}_\nabla(\mathsf{Q})$ of *disjunctive modal formulas* in Q:

$$a \ ::= \ p \mid \neg p \mid \bot \mid \top \mid a_0 \vee a_1 \mid \pi \bullet \nabla\alpha.$$

Here $p \in \mathsf{Q}$, $\pi \in \mathrm{CL}(\mathsf{Q})$ and $\alpha \in P_\omega \mathrm{DML}_\nabla(\mathsf{Q})$. $\lhd$

As mentioned, the bullet connective is semantically equivalent to conjunction:

$$\mathbb{S}, s \Vdash \pi \bullet \nabla\alpha \text{ iff } \mathbb{S}, s \Vdash \pi \text{ and } \mathbb{S}, s \Vdash \nabla\alpha.$$

Note however, that this conjunction is special in the sense that it combines 'local' information about s itself with information about the unfolding of s. The point is that this is the *only* form of conjunction that the language DML_∇ allows.

Theorem 5.8 *The languages* ML *and* DML_∇ *are effectively equi-expressive.*

Proof. We will show how to rewrite a formula $a \in \mathrm{ML}$ into an equivalent formula in DML_∇. Start by rewriting a into *negation normal form*:

$$a \ ::= \ p \mid \neg p \mid \bot \mid \top \mid a_0 \vee a_1 \mid a_0 \wedge a_1 \mid \Diamond a \mid \Box a,$$

then by (27) we can find an equivalent formula a' in the language given by

$$a \ ::= \ p \mid \neg p \mid \bot \mid \top \mid a_0 \vee a_1 \mid a_0 \wedge a_1 \mid \nabla\alpha.$$

Finally, the modal distributive law (28) allows us to push down nabla's to the propositional level, and so using the propositional distributive law $(a \wedge (b_0 \vee b_1) \equiv (a \wedge b_0) \vee (a \wedge b_1))$, we can rewrite a' into an equivalent disjunctive modal formula. QED

Theorem 5.8 can be used to prove various interesting results about modal logic, such as the finite model property, or the decidability of the satisfiability problem — in *linear* time, once the formula is in disjunctive normal form. Rather than proving these corollaries here, we will prove these results in the far more general setting of the *coalgebraic* cover modality.

5.2 Moss' coalgebraic cover modality

We will now generalise the cover modality from the case where $T = P$ to the setting where T is an arbitrary smooth and standard functor. We are eager to keep our language *finitary*, in the sense that formulas will be finitary objects, with for instance finitely many subformulas. For this reason we will work with the *finitary version* of the functor T.

Recall from Convention 5.1 that T preserves inclusions; because of this we may define its *finitary* version $T_\omega : \mathsf{Set} \to \mathsf{Set}$ by putting

$$\begin{array}{lcl} T_\omega(S) & := & \bigcup\{TX \mid X \subseteq_\omega S\}, \\ T_\omega(f : S \to S') & := & (Tf)\restriction_{T_\omega S} . \end{array}$$

It is easy to verify that T_ω also preserves inclusions; given the definition of T_ω on functions, we may write Tf instead of $T_\omega f$ without causing confusion. Note, however, that T_ω being finitary does not necessarily mean that $T_\omega S$ is *finite*, even if S is so; a simple counterexample is given by the bag functor.

Definition 5.9 Formulas of the language $\mathrm{ML}_T(\mathsf{Q})$ are given by the following recursive definition:

$$a \ ::= \ p \mid \bot \mid \neg a \mid a_0 \vee a_1 \mid \nabla \alpha$$

where $p \in \mathsf{Q}$, and $\alpha \in T_\omega(\mathrm{ML}_T(\mathsf{Q}))$. We will often write ML_T instead of $\mathrm{ML}_T(\mathsf{Q})$ if the set Q of proposition letters is either understood or irrelevant. $\lhd$

The semantics of ML_T is defined as follows. Recall that a T*-model* is a triple (S, σ, V), where (S, σ) is a T-coalgebra and $V : \mathsf{Q} \to PS$ is a valuation.

Definition 5.10 Let $\mathbb{S} = (S, \sigma, V)$ be a T-model. Then by induction on the complexity of ML_T-formulas we define the satisfaction relation $\Vdash$:

$$\begin{array}{lcl}
\mathbb{S}, s \Vdash p & \text{iff} & s \in V(p) \\
\mathbb{S}, s \Vdash \bot & : & \text{never} \\
\mathbb{S}, s \Vdash \neg a & \text{iff} & \mathbb{S}, s \nVdash a \\
\mathbb{S}, s \Vdash a_0 \vee a_1 & \text{iff} & \mathbb{S}, s \Vdash a_0 \text{ or } \mathbb{S}, s \Vdash a_1 \\
\mathbb{S}, s \Vdash \nabla \alpha & \text{iff} & (\sigma(s), \alpha) \in \overline{T}(\Vdash).
\end{array}$$

A formula $a \in \mathrm{ML}_T$ is *satisfiable* iff a is satisfiable in some state of some T-model, and *valid* if its negation is not satisfiable.

Furthermore, we say that two pointed T-models are ML_T*-equivalent* or *(modally) equivalent* if they satisfy the same ML_T-formulas, notation: $\mathbb{S}, s \equiv_T \mathbb{S}', s'$. $\lhd$

Remark 5.11 Before we consider the instantiations of this logic for some set functor T, we argue that the semantics of ML_T is *well defined.* The reader might have some worries about the inductive clause for the ∇ modality, since the definition refers to the lifting of the *full* satisfaction relation.

The point is that because of our assumptions on T, its associated relation lifting $\overline{T}$ commutes with restrictions, cf. Fact B.39. This means that

$$(\sigma(s), \alpha) \in \overline{T}(\Vdash) \text{ iff } (\sigma(s), \alpha) \in \overline{T}(\Vdash\restriction_{S \times X}), \tag{29}$$

where X is any finite set of formulas such that $\alpha \in T_\omega X$. Thus, in order to determine whether $\nabla\alpha$ holds at s or not, we only have to know the interpretation of the formulas used in the justification that $\nabla\alpha$ is a formula. Below we shall see that in fact there is a unique set $Base(\alpha)$ which is the smallest (finite) set X such that $\alpha \in TX$. In other words, we may replace the 'quasi-inductive' clause for ∇ in Definition 5.10 with the following, properly inductive one:

$$\mathbb{S}, s \Vdash \nabla\alpha \text{ iff } (\sigma(s), \alpha) \in \overline{T}(\Vdash\restriction_{S \times Base(\alpha)}). \tag{30}$$

Example 5.12 In this example we look at the interpretation of the coalgebraic cover modality instantiated for various coalgebra types T. That is, let $\mathbb{S} = (S, \sigma, V)$ be a T-model and consider an element $\alpha \in T_\omega(\mathrm{ML}_T)$; below we will explain what it means for the formula $\nabla\alpha$ to hold at s.

(a) In case $T = K_C$ is a constant functor, $\alpha \in T(\mathrm{ML}_T) = C$ is just a colour $\alpha \in C$. In this case we find $\mathbb{S}, s \Vdash \nabla\alpha$ iff $\sigma(s) = \alpha$.

(b) If $T = Id$ is the identity functor, $\alpha \in T(\mathrm{ML}_T) = \mathrm{ML}_T$ is just a formula. We obtain the *next-time operator* of linear temporal logic: $\mathbb{S}, s \Vdash \nabla\alpha$ iff $\mathbb{S}, \sigma(s) \Vdash \alpha$.

(c) For the binary tree functor $T = Id \times Id$, the semantics of nabla is as follows: given $\alpha = (a_0, a_1) \in \mathrm{ML}_T \times \mathrm{ML}_T = T(\mathrm{ML}_T)$, we have $\mathbb{S}, s \Vdash \nabla(a_0, a_1)$ iff $\mathbb{S}, t_0 \Vdash a_0$ and $\mathbb{S}, t_1 \Vdash a_1$, where t_0 and t_1 are the 'left' and 'right' successor of s, respectively, given by $\sigma(s) = (t_0, t_1)$.

(d) For the automata functor $T = 2 \times Id^C$, an element $\alpha \in T(\mathrm{ML}_T)$ is of the form $(i, \overline{a})$ with $i \in \{0, 1\}$ and $\overline{a} = (a_c)_{c \in C}$, with each $a_c \in \mathrm{ML}_T$. With $\sigma = \langle \chi, \tau \rangle$ we have $\mathbb{S}, s \Vdash \nabla(i, \overline{a})$ iff $\chi(s) = i$ and $\mathbb{S}, \tau(s)(c) \Vdash a_c$.

(e) Where $T = P$ is the power set functor, it is easy to verify that ∇_P is the cover modality discussed in the previous section.

Example 5.13 In this example we look at the ∇-logic of the distribution functor D; the nabla modality for the bag functor is defined similarly.

First of all, observe that defined as in Example B.6, D does not preserve inclusions. We can remedy this by taking a variant D' of D which takes a set S to the collection $D'S$ of *partial maps* from S to $(0, 1]$. We will not pursue this road further, but we may use it to observe that an element $\alpha \in D_\omega(\mathrm{ML}_D)$ can be represented as a finite set $\{(a_1, p_n), \ldots, (a_n, p_n)\}$ such that $p_i > 0$ for all i, and $\Sigma_i p_i = 1$.

For the definition of relation lifting $\overline{D}$, consider a relation $R \subseteq X_0 \times X_1$. We claim that the relation $\overline{D}R \subseteq DX_0 \times DX_1$ consists of those pairs (μ_0, μ_1) for which there is a distribution $\rho : R \to [0, 1]$ satisfying the 'magic square conditions':

$$\begin{array}{rl} \text{for all } x_0 \in X_0. & \mu_0(x_0) = \sum_{y_1 \in X_1} \rho(x_0, y_1), \\ \text{and for all } x_1 \in X_1. & \mu_1(x_1) = \sum_{y_0 \in X_0} \rho(y_0, x_1). \end{array}$$

For a concrete example, consider the table below:

	a	b	c	d	e	
1	$\square$	.17	$\square$	.08	.08	.33
2	.10	.03	.12	$\square$	.08	.33
3	$\square$	.00	.18	.12	.04	.34.
	.10	.20	.30	.20	.20	

Here we have $X_0 = \{a, b, c, d, e\}$, $X_1 = \{1, 2, 3\}$ and $R \subseteq X_0 \times X_1$ consists of those pairs of which the corresponding entry in the table is marked by a number (that is, not by $\square$). The distributions μ_0 and μ_1 are given, respectively, in the bottom row and rightmost column of the table. The entries in the table represent a distribution ρ that meets the magic square conditions.

Now let $\mathbb{S} = (S, \sigma, V)$ be a D-model, and consider a formula $\nabla\alpha$ with $\alpha = \{(a_1, p_n), \ldots, (a_n, p_n)\}$. Then we find that

$$
\begin{aligned}
\mathbb{S}, s \Vdash \nabla\alpha \text{ iff } & \text{ there is a relation } R \subseteq \Vdash \text{ and a map } \rho : R \to [0,1] \text{ such that} \\
& \text{for all } i : \quad \sum_{\{t \mid \mathbb{S}, t \Vdash a_i\}} \rho(t, a_i) = p_i, \\
& \text{and for all } t \in S : \quad \sum_{\{i \mid \mathbb{S}, t \Vdash a_i\}} \rho(t, a_i) = \sigma(s)(t).
\end{aligned}
$$

5.3 Basic properties of ∇

In this section we prove some of the basic properties of the coalgebraic cover modality. We start with showing that ML_T is a finitary logic indeed, i.e., that every formula has only finitely many subformulas. The key property of finitary functors that will make this possible is that for every $\alpha \in T_\omega A$ there is a *smallest* subset $A' \subseteq A$ that *supports* α, i.e., is such that $\alpha \in T_\omega A'$.

Definition 5.14 Given a finitary functor T and an element $\alpha \in TX$, we define

$$Base_X^T(\alpha) := \bigcap \{Y \subseteq_\omega X \mid \alpha \in TY\}.$$

We write $Base^T$ rather than $Base^{T_\omega}$, and in fact omit the superscript whenever possible. $\lhd$

Example 5.15 The following examples are easy to check: $Base_X^{Id} : X \to P_\omega X$ is the singleton map, $Base_X^P : P_\omega X \to P_\omega X$ is the identity map on $P_\omega X$, $Base_X^{Id^2} : X \times X \to P_\omega X$ maps the pair (x_1, x_2) to the set $\{x_1, x_2\}$, and $Base^D$ maps a finitary distribution μ to its support $\{s \in S \mid \mu(s) > 0\}$.

Proposition 5.16 *Let $T : \mathsf{Set} \to \mathsf{Set}$ be smooth and standard. Then the following hold:*
(1) for any $\alpha \in T_\omega X$, $Base_X^T(\alpha)$ is the smallest set supporting α;
(2) $Base^T$ provides a natural transformation $Base : T_\omega \stackrel{\cdot}{\to} P_\omega$.

Recall that $Base^T$ being a natural transformation means that the following diagram commutes, for any map $f : X \to Y$:

$$
\begin{array}{ccccc}
X & \quad & T_\omega X & \xrightarrow{Base_X^T} & P_\omega X \\
{\scriptstyle f}\downarrow & & {\scriptstyle Tf}\downarrow & & \downarrow{\scriptstyle P_\omega f} \\
Y & & T_\omega Y & \xrightarrow[Base_Y^T]{} & P_\omega Y
\end{array}
\tag{31}
$$

Proof. Part (1) of the Proposition is an immediate consequence of Proposition B.33.

For the second part, consider a map $f : X \to X'$. We have to show $P_\omega f \circ Base_X = Base_{X'} \circ T_\omega f$. Fix $\alpha \in T_\omega X$ and write $B = Base_X(\alpha)$ and $B' = Base_{X'}(T_\omega f(\alpha))$. We need to prove $B' = f[B]$. For the inclusion $\subseteq$, first note that the following diagram must commute (recall that hooked arrows denote inclusions):

$$\begin{tikzcd}
T_\omega B \arrow[r, "T_\omega f\restriction_B"] \arrow[d, hook] & T_\omega(f[B]) \arrow[d, hook] \\
T_\omega X \arrow[r, "T_\omega f"'] & T_\omega X'
\end{tikzcd}$$

From this it follows that $T_\omega f(\alpha)$ actually belongs to $T_\omega(f[B])$. In other words, $f[B]$ supports $T_\omega f(\alpha)$ and, as B' is by definition the smallest such set, it follows that $B' \subseteq f[B]$.

For the opposite inclusion $\supseteq$, we consider the diagram below. Here 1 denotes some arbitrary singleton set, and the arrow α denotes any function from 1 to $T_\omega X$ mapping the unique element of 1 to α.

$$\begin{tikzcd}
1 \arrow[dr, dotted] \arrow[drr, bend left, "T_\omega f(\alpha)"] \arrow[ddr, bend right, "\alpha"'] & & \\
& T_\omega(f^{-1}(B')) \arrow[r] \arrow[d, hook] & T_\omega(B') \arrow[d, hook] \\
& T_\omega X \arrow[r, "T_\omega f"] & T_\omega X'
\end{tikzcd}$$

But since T_ω preserves weak pullbacks, the dotted arrow exists and shows that $\alpha \in T_\omega(f^{-1}(B'))$. By minimality of the base, it follows $B \subseteq f^{-1}(B')$, that is, $B' \supseteq f[B]$. QED

By Fact 5.16(1) we may find for any formula $\nabla\alpha$ a *smallest* (and finite) collection X of formulas such that $\alpha \in T_\omega X$, namely, the set $X = \mathit{Base}(\alpha)$. This means that we can define a natural notion of *subformula*.

Definition 5.17 We define the set $\mathtt{Sfor}(a)$ of *subformulas* of a formula $a \in \mathrm{ML}_T$ by the following induction:

$$\begin{array}{lcll}
\mathtt{Sfor}(a) & := & \{a\} & \text{if } a \in \{p, \bot\} \\
\mathtt{Sfor}(\neg a) & := & \{\neg a\} \cup \mathtt{Sfor}(a) & \\
\mathtt{Sfor}(a_0 \vee a_1) & := & \{a_0 \vee a_1\} \cup \mathtt{Sfor}(a_0) \cup \mathtt{Sfor}(a_1) & \\
\mathtt{Sfor}(\nabla\alpha) & := & \{\nabla\alpha\} \cup \bigcup\{\mathtt{Sfor}(a) \mid a \in \mathit{Base}(\alpha)\} &
\end{array}$$

The elements of $\mathit{Base}(\alpha)$ will be called the *immediate* subformulas of $\nabla\alpha$. $\lhd$

The next properties that we consider are invariance and expressivity.

Theorem 5.18 *For any smooth and standard functor T, the language ML_T is invariant: Given any two pointed T-models $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ we have*

$$(\mathbb{S}, s) \simeq_T (\mathbb{S}', s') \text{ implies } (\mathbb{S}, s) \equiv_T (\mathbb{S}', s'). \tag{32}$$

Proof. Given the smoothness of T, it suffices to prove that *bisimilarity* implies modal equivalence. Assuming that $Z : S \leftrightarrows S'$, we will prove by induction on the complexity of ML_T-formulas that, for every ML_T-formula a:

$$\mathbb{S}, s \Vdash a \text{ iff } \mathbb{S}', s' \Vdash a, \tag{33}$$

for every pair of states $(s, s') \in Z$. Clearly this suffices to prove the proposition.

Skipping the routine parts of the proof (i.e., the base step and boolean cases of the inductive step), we focus on the case where $a = \nabla\alpha$. We only prove the direction from right to left of (33).

So, assume that $(s, s') \in Z$ and $\mathbb{S}', s' \Vdash \nabla\alpha$, and let $\Vdash \subseteq S \times \mathrm{ML}_T$ and $\Vdash' \subseteq S' \times \mathrm{ML}_T$ denote the satisfaction relations on $\mathbb{S}$ and $\mathbb{S}'$, respectively. It follows from $(s, s') \in Z$ that $(\sigma(s), \sigma'(s')) \in \overline{T}Z$, and from $\mathbb{S}', s' \Vdash \nabla\alpha$ that $(\sigma'(s'), \alpha) \in \overline{T}(\Vdash')$; but from the latter fact, together with the observation that $\alpha \in TBase(\alpha)$, we may derive that $(\sigma'(s'), \alpha) \in \overline{T}(\Vdash'\restriction_{S \times Base(\alpha)})$ (cf. Fact B.39). Putting these observations together with the fact that $\overline{T}$ preserves relation composition, we find that

$$(\sigma(s), \alpha) \in \overline{T}\big(Z \,;\, \Vdash'\restriction_{S \times Base(\alpha)}\big).$$

But by the inductive hypothesis we obtain that $Z \,;\, \Vdash'\restriction_{S' \times Base(\alpha)} \subseteq \Vdash$. so that it follows by the monotonicity of relation lifting that $(\sigma(s), \alpha) \in \overline{T}(\Vdash)$. From this it is immediate by the semantics of ∇ that $\mathbb{S}, s \Vdash \nabla\alpha$, as required. QED

As could be expected, the converse of this proposition only holds if we restrict attention to *image-finite* coalgebras.

Definition 5.19 A T-coalgebra $\mathbb{S} = (S, \sigma)$ is *image-finite* if $\sigma(s) \in T_\omega S$, for all $s \in S$. $\lhd$

Theorem 5.20 *For any smooth and standard functor T, the language ML_T is expressive on the class of image-finite T-models: Given any two pointed T_ω-models $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ we have*

$$(\mathbb{S}, s) \equiv_T (\mathbb{S}', s') \textit{ implies } (\mathbb{S}, s) \simeq_T (\mathbb{S}', s'). \tag{34}$$

Proof. It suffices to show that the relation of modal equivalence is itself a *bisimulation*, when restricted to the class of image-finite coalgebra models.

Fix two T-models $\mathbb{S} = (S, \sigma, V)$ and $\mathbb{S}' = (S', \sigma', V')$, and let $\equiv \subseteq S \times S'$ denote the relation of modal equivalence between $\mathbb{S}$ and $\mathbb{S}'$. (That is, we avoid notational clutter and write $\equiv$ instead of $\equiv_T$.) We will use Theorem 3.8 in order to prove that $\equiv$ is a bisimulation, and suppose for contradiction that $s \equiv s'$. Our goal will be to show that

$$(\sigma s, \sigma' s') \in \overline{T}(\equiv). \tag{35}$$

First of all, it follows by image-finiteness that we may define the (finite) sets $B := Base(\sigma s)$ and $B' := Base(\sigma' s')$. Since $\overline{T}$ commutes with restrictions it suffices to prove that

$$(\sigma s, \sigma' s') \in \overline{T}(\equiv\restriction_{B \times B'}). \tag{36}$$

Furthermore, since B and B' are finite, for every $t \in B$ we may find a formula c_t such that, for all $t' \in B'$,

$$\mathbb{S}', t' \Vdash c_t \text{ iff } t \equiv t',$$

as the reader can easily verify. But with $H := \{c_t \mid t \in B\}$, we may think of c as a surjection $c : B \twoheadrightarrow H$ satisfying $\mathsf{Gr}(c) \subseteq \Vdash$ and

$$\equiv\restriction_{B \times B'} = \mathsf{Gr}(c) \,;\, (\Vdash_{B' \times H})^{\smile}.$$

From this we may conclude, using various properties of relation lifting, that

$$\overline{T}(\equiv\restriction_{B \times B'}) = \mathsf{Gr}(Tc) \,;\, (\overline{T}\Vdash_{B' \times H})^{\smile}. \tag{37}$$

Now let

$$\gamma := (Tc)(\sigma s),$$

then by definition we have $(\sigma s, \gamma) \in \mathsf{Gr}(Tc)$. Furthermore, from $\mathsf{Gr}(c) \subseteq \Vdash$ we derive $\mathsf{Gr}(Tc) \subseteq \overline{T}\Vdash$, so that we find $(\sigma s, \gamma) \in \overline{T}\Vdash$. This implies that the formula $\nabla\gamma$ is *true* at s, and since $s \equiv s'$, it is also true at s'. By the semantics of the ∇ modality this means that $(\sigma s', \gamma) \in \overline{T}\Vdash$. Using various properties of relation lifting we find that $(\gamma, \sigma s') \in (\overline{T}\Vdash_{B' \times H})^{\smile}$. We may then conclude that $(\sigma s, \sigma s') \in \mathsf{Gr}(Tc) \,;\, (\overline{T}\Vdash_{B' \times H})^{\smile}$, which by (37) suffices to prove (36), and thus (35). In other words, we are done. QED

The last basic property that we mention is that of *satisfiability reduction.*

Proposition 5.21 *Let $\nabla\alpha$ be a formula in ML_T. Then $\nabla\alpha$ is satisfiable iff every $a \in Base(\alpha)$ is satisfiable.*

Proof. For the direction from right to left, assume that every $a \in Base(\alpha)$ is satisfiable. That is, assume that for every $a \in Base(\alpha)$ there is a pointed model $(\mathbb{S}_a, s_a)$, with $\mathbb{S}_a = (S_a, \sigma_a, V_a)$ and such that $\mathbb{S}, s_a \Vdash a$, for each a. We define a new model $\mathbb{S} = (S, \sigma, V)$, where $S := \{r\} \uplus \biguplus\{S_a \mid a \in Base(\alpha)\}$. For the valuation V we simply define $V(p) := \bigcup_a V_a(p)$, while on an element $s \in S_a$ the coalgebra map $\sigma : S \to TS$ is defined by putting $\sigma(s) := \sigma_a(s)$. For the definition of the unfolding $\sigma(r)$ of the 'root' r, consider the map $f : Base(\alpha) \to S$ given by $a \mapsto s_a$, and simply put $\sigma(r) := (Tf)(\alpha)$. It is then immediate by the definition of f that $(\mathsf{Gr} f)^{\smile} \subseteq \Vdash$, so that we find, using various properties of relation lifting (cf. Fact B.36 and B.39):

$$(\sigma r, \alpha) \in (\mathsf{Gr}(Tf))^{\smile} = \overline{T}((\mathsf{Gr} f)^{\smile}) \subseteq \overline{T}(\Vdash),$$

from which it follows that $\mathbb{S}, r \Vdash \nabla\alpha$ indeed.

For the opposite direction, we need the following little fact about the functor T:

$$\text{for any } f : A \to B \text{ and any } \alpha \in TA \text{ we have } (Tf)(\alpha) \in T(f[A]). \tag{38}$$

To see why (38) holds, factorize f as the unique composition $f = \iota \circ f'$ of a surjection $f' : A \twoheadrightarrow f[A]$ and an inclusion $\iota : f[A] \hookrightarrow B$. From this factorization it follows that $Tf = (T\iota) \circ (Tf')$, where $f' : TA \twoheadrightarrow Tf[A]$ is surjective since every set functor preserves surjections, and $T\iota : Tf[A] \hookrightarrow TB$ is an inclusion by assumption on T. From these observations (38) is easy to derive.

Now assume that $\nabla\alpha$ is satisfiable, then there is a T-model $\mathbb{S} = (S, \sigma, V)$ such that $\mathbb{S}, s \Vdash \nabla\alpha$ for some state $s \in S$. Then by definition of the semantics of ∇ we have that $(\sigma s, \alpha) \in \overline{T}(\Vdash)$, and so by definition of $\overline{T}$ there is an object $\rho \in T(\Vdash)$ such that $T\pi_0(\rho) = \sigma(s)$ and $T\pi_1(\rho) = \alpha$, where $\pi_0 : \Vdash \to S$ and $\pi_1 : \Vdash \to \mathrm{ML}_T$ are the projection functions on the relation $\Vdash$. But then it follows by (38) that $\alpha \in T(\pi_1[\Vdash]) = T(\mathsf{Ran}(\Vdash))$, so that $Base(\alpha) \subseteq \mathsf{Ran}(\Vdash)$. In other words, for every $a \in Base(\alpha)$ there is an $s \in S$ where a holds. In particular, this means that every $a \in Base(\alpha)$ is satisfiable. QED

5.4 Coalgebraic modal distributive laws

In this section we will formulate three *coalgebraic modal distributive laws (CMDLs)* describing the interaction between the coalgebraic modality ∇ on the one hand, and the boolean operations on the other. For a concise formulation of these principles it will be convenient to slightly rearrange the coalgebraic modal language, working with the finitary versions $\bigwedge$ and $\bigvee$ of the binary connectives for conjunction and disjunction. That is, in this section we will be working with the following variant of the language.

Definition 5.22 The language L_T is given by the following grammar:

$$a \ ::= \ p \mid \neg a \mid \bigwedge A \mid \bigvee A \mid \nabla\alpha$$

where $p \in \mathsf{Q}$, $A \in P_\omega L_T$ and $\alpha \in T_\omega L_T$. $\lhd$

The semantics of $\bigwedge$ and $\bigvee$ is as expected:

$$\begin{array}{lll} \mathbb{S}, s \Vdash \bigwedge A & \text{iff} & \mathbb{S}, s \Vdash a \text{ for all } a \in A \\ \mathbb{S}, s \Vdash \bigvee A & \text{iff} & \mathbb{S}, s \Vdash a \text{ for some } a \in A \end{array}$$

In particular, this means that instead of taking $\top$ and $\bot$ to be primitive symbols, we may consider them as abbreviations:

$$\begin{array}{lll} \top & := & \bigwedge \varnothing \\ \bot & := & \bigvee \varnothing. \end{array}$$

A key aspect of the formulation of the CMDLs is the observation that we may think of the connectives $\bigwedge, \bigvee$ and $\neg$ as *maps* of the respective types $\bigwedge, \bigvee : P_\omega L_T \to L_T$ and $\neg : L_T \to L_T$. In particular, this perspective allows us to apply the functor T to these connectives, obtaining maps $T\bigwedge, T\bigvee : T_\omega P_\omega L_T \to T_\omega L_T$, and $T\neg : T_\omega L_T \to T_\omega L_T$. Thus, for any object $\Phi \in T_\omega P_\omega L_T$ we find $(T\bigvee)\Phi \in T_\omega L_T$, which means that $\nabla(T\bigvee)\Phi$ is a well-formed formula.

Convention 5.23 Since we will be dealing here with formulas and similar objects in various, closely related sets, incuding $\mathsf{Q}, L_T, T_\omega L_T, P_\omega L_T, P_\omega T_\omega L_T$ and $T_\omega P_\omega L_T$, it will be convenient to use some kind of *naming convention*, see Table 1.

In order to formulate the modal distributive laws we need some auxiliary definitions.

Set	Elements
Q	$p, q, \ldots$
L_T	$a, b, \ldots$
$T_\omega L_T$	$\alpha, \beta, \ldots$
$P_\omega L_T$	$A, B, \ldots$
$P_\omega T_\omega L_T$	$\Gamma, \Delta, \ldots$
$T_\omega P_\omega L_T$	$\Phi, \Psi, \ldots$

Table 1: Naming convention

Definition 5.24 Given a smooth and standard set functor T, we define, for every set X, a function $\lambda^T_X : TPX \to PTX$ by putting

$$\lambda^T_X(\Phi) := \{\alpha \in TX \mid (\alpha, \Phi) \in \overline{T}(\in_X)\}, \tag{39}$$

where $\in_X$ denotes the membership relation $\in$, restricted to $X \times PX$. Elements of $\lambda^T_X(\Phi)$ will be referred to as *lifted members* of Φ. The family $\lambda^T = \{\lambda^T_X\}_{X \in \mathsf{Set}}$ will be called the *T-transformation.*

A set $\Phi \in TPX$ is a *redistribution* of a set $\Gamma \in PTX$ if $\Gamma \subseteq \lambda^T_X(\Phi)$, that is, every element of Γ is a lifted member of Φ. In case $\Gamma \in P_\omega T_\omega X$, we call a redistribution Φ *slim* if $\Phi \in T_\omega P_\omega(\bigcup_{\gamma \in \Gamma} Base(\gamma))$. The set of slim redistributions of Γ is denoted as $SRD(\Gamma)$. $\lhd$

Remark 5.25 Properties of $\overline{T}$ are intimately related to those of λ^T. In order to express the connection, we need to introduce the concept of a (categorical) distributive law.

Let T be a covariant set functor. A *distributive law* of T over a (co- or contravariant) set functor M is a natural transformation $\theta : TM \to MT$; that is, the following diagram commutes, for every map $f : X \to Y$:

$$\begin{array}{ccc} X & & TMX \xrightarrow{\theta_X} MTX \\ {\scriptstyle f}\downarrow & & {\scriptstyle TMf}\downarrow \qquad\quad \downarrow{\scriptstyle MTf} \\ Y & & TMY \xrightarrow{\theta_Y} MTY \end{array}$$

(In case M is a covariant functor the downward arrows have to be reversed.)

Now observe that we can in fact *define* the family of functions $\lambda^T = \{\lambda^T_X\}_{X \in \mathsf{Set}}$ as in (39) for any set function T. If T preserves weak pullbacks, $\lambda^T = \{\lambda^T_X\}_{X \in \mathsf{Set}}$ is a distributive law of T over the power set functor P.

Definition 5.26 Let T be a smooth and standard set functor which restricts to finite sets. Consider the following *coalgebraic modal distributive laws*:

($DL_\bigvee$) $\nabla(T\bigvee)(\Phi) \equiv \bigvee \left\{\nabla\alpha \mid (\alpha, \Phi) \in \overline{T}(\in_{L_T})\right\}$

($DL_\bigwedge$) $\bigwedge \left\{\nabla\gamma \mid \gamma \in \Gamma\right\} \equiv \bigvee \left\{\nabla(T\bigwedge)(\Phi) \mid \Phi \in SRD(\Gamma)\right\}$

($DL_\neg$) $\neg\nabla\alpha \equiv \bigvee \left\{\nabla T(\bigwedge \circ P\neg)\Psi \mid \Psi \in T_\omega P_\omega Base(\alpha) \text{ and } (\alpha, \Psi) \notin \overline{T}(\not\in)\right\}$ $\lhd$

Note that the restriction to finite sets is needed in order to keep the disjunctions on the right hand side of the equivalences finite

Proposition 5.27 *Let T be a smooth and standard set functor which restricts to finite sets. All three coalgebraic modal distributive laws are valid.*

Proof. In order to understand the validity of these laws, fix some T-model $\mathbb{S} = (S, \sigma, V)$.

We first consider $(DL_{\bigvee})$, proving the direction from left to right. First observe that for any $A \subseteq_\omega L_T$ we have $\mathbb{S}, s \Vdash \bigvee A$ iff $\mathbb{S}, s \Vdash a$, for some $a \in A$. Putting it differently, the relations $\Vdash ; \in$ and $\Vdash ; \bigvee^{\smile}$ coincide[9]. From this it follows that

$$\overline{T}(\Vdash ; \in) \;=\; \overline{T}(\Vdash ; \bigvee^{\smile}). \tag{40}$$

Now fix some object $\Phi \in T_\omega P_\omega L$, and suppose that s is a state in $\mathbb{S}$ such that $s \Vdash \nabla(T \bigvee)\Phi$. By the truth definition, the pair $(\sigma(s), (T \bigvee)(\Phi))$ belongs to the relation $\overline{T}(\Vdash)$, and so $(\sigma(s), \Phi)$ belongs to $(\overline{T}\Vdash) ; (T \bigvee)^{\smile} = \overline{T}(\Vdash ; \bigvee^{\smile})$. But then by (40), we find $(\sigma(s), \Phi) \in \overline{T}(\Vdash ; \in) = \overline{T}\Vdash ; \overline{T}\in$. In other words, there is some object β such that $(\sigma(s), \beta) \in \overline{T}(\Vdash)$ and $(\beta, \Phi) \in \overline{T}(\in)$. Clearly then $s \Vdash \nabla\beta$, and so we have $s \Vdash \bigvee\{\nabla\beta \mid \beta \,\overline{T}\!\in \Phi\}$, as required.

For the validity of $(DL_{\bigwedge})$, we also confine attention to the direction from left to right. Assume that $\mathbb{S}, s \Vdash \nabla\gamma$ for all $\gamma \in \Gamma$. We need to come up with some slim redistribution Φ of Γ such that $\mathbb{S}, s \Vdash \nabla(T \bigwedge)\Phi$. For this purpose we associate, with any state $t \in S$, the finite set

$$A_t := \{a \in \bigcup_{\gamma \in \Gamma} Base(\gamma) \mid \mathbb{S}, t \Vdash a\}.$$

Taking A to be the map $A : S \to P_\omega L_T$, we may define $\Phi := (TA)(\sigma(s)) \in T_\omega P_\omega L_T$.

First we show that $\mathbb{S}, s \Vdash \nabla(T \bigwedge)\Phi$. Observe that by definition of the map $A : S \to P_\omega L_T$, the function $\bigwedge \circ A : S \to L_T$ is such that

$$\mathsf{Gr}(\bigwedge \circ A) \subseteq \Vdash.$$

From this we obtain

$$\mathsf{Gr}((T \bigwedge) \circ (TA)) \subseteq \overline{T}(\Vdash)$$

by the properties of the operation $\overline{T}$. But that means that for every element $\tau \in TS$, we have that $\big(\tau, ((T \bigwedge) \circ (TA))(\tau)\big) \in \overline{T}\Vdash$. In particular, we find that $(\sigma s, (T \bigwedge)\Phi) = (\sigma s, (T \bigwedge)(TA)(\sigma(s)) \in \overline{T}\Vdash$, showing that $\mathbb{S}, s \Vdash \nabla(T \bigwedge)\Phi$ as required.

It is left to prove that Φ is a slim redistribution of Γ. Observe that by definition of the map A, we have that

$$\mathsf{Gr}(A) ; \in^{\smile} = \Vdash\restriction_{S \times B},$$

where $B := \bigcup_{\gamma \in \Gamma} Base(\gamma)$. From this it follows by the properties of relation lifting that

$$\mathsf{Gr}(TA) ; (\overline{T}\in)^{\smile} = \overline{T}(\Vdash)\restriction_{TS \times TB}.$$

[9]Here we write $\bigvee$ instead of $\mathsf{Gr}(\bigvee)$

But then for each $\gamma \in \Gamma$ we may derive from the fact that $(\sigma s, \gamma) \in \overline{T}(\Vdash)\restriction_{TS\times TB}$ that there is some object Ψ such that $(\sigma s, \Psi) \in \mathsf{Gr}(TA)$ and $(\Psi, \gamma) \in (\overline{T}\in)^{\smile}$. It then easily follows that $\Psi = (TA)(\sigma s) = \Phi$ and so $(\gamma, \Phi) = (\gamma, \Psi) \in \overline{T}(\in)$. In other words, each $\gamma \in \Gamma$ is a lifted member of Φ, and so Φ is a redistribution of Γ; but then by its definition it is slim.

Finally, the validity of $(DL_{\neg})$ is left as an exercise to the reader. QED

5.5 Coalgebraic Logic

We will now see that the coalgebraic modal distributive laws that we proved in the previous section are in fact quite strong principles, with important applications.

We start with the coalgebraic generalisation of the disjunctive normal form result on the cover modality, Theorem 5.8.

Definition 5.28 We let the following grammar:

$$a \ ::= \ p \mid \neg p \mid \bot \mid \top \mid a_0 \vee a_1 \mid \pi \bullet \nabla\alpha.$$

define the set $\mathrm{DML}_T(\mathsf{Q})$ of *disjunctive T-modal formulas* in Q. $\lhd$

The proof of the following theorem is completely analogous to that of Theorem 5.8.

Theorem 5.29 *Let T be a smooth and standard set functor which restricts to finite sets. The languages ML_T and DML_T are effectively equi-expressive.*

For the following result recall that a modal logic $(L, \Vdash)$ has the *finite model property* if every satisfiable L-formula is satisfiable in a finite model.

Theorem 5.30 *Let T be a smooth and standard set functor which restricts to finite sets. Then ML_T has the finite model property.*

Proof. By Theorem 5.29 it suffices to prove the finite model property for *disjunctive* formulas. We leave it as an exercise for the reader to establish this result — this goes by a straightforward proof by induction on the complexity of DML_T-formulas, of which the inductive case for the ∇ modality uses the observation underlying the proof of our satisfiability reduction result, Proposition 5.21. QED

Remark 5.31 Theorem 5.29 can also be used to obtain *decidability* results for logics ML_T. For instance, it can be proved that the satisfiability problem for the language $\mathrm{ML}_P = \mathrm{ML}_\nabla$ can be solved in *linear time*. However, since these results depend on the functor, or more specifically: on the representation of formulas of the form $\nabla\alpha$, we refrain from going into detail here.

Remark 5.32 As another corollary of Theorem 5.29 we can show that for any smooth and standard set functor T which restricts to finite sets, the logic ML has *uniform interpolation*, a strong version of Craig's interpolation property. $\bullet$

Finally, we briefly mention a sound and complete derivation system for the set of valid ML-formulas.

Definition 5.33 Let T be a smooth and standard set functor which restricts to finite sets. For the *derivation system* **M**, we start with fixing an arbitrary sound and complete set of axioms and rules for classical propositional logic; we extend this with the following derivation rule:

$$\frac{\{a \to b \mid (a,b) \in Z\}}{\nabla\alpha \to \nabla\beta}\ (\alpha,\beta) \in \overline{T}Z,$$

together with the one-sided versions of the coalgebraic modal distributive laws:

$(A_{\bigvee})$ $\nabla(T\bigvee)(\Phi) \to \bigvee\left\{\nabla\alpha \mid (\alpha,\Phi) \in \overline{T}(\in_X)\right\}$

$(A_{\bigwedge})$ $\bigwedge\left\{\nabla\gamma \mid \gamma \in \Gamma\right\} \to \bigvee\left\{\nabla(T\bigwedge)(\Phi) \mid \Phi \in SRD(\Gamma)\right\}$

$(A_{\neg})$ $\neg\nabla\alpha \to \bigvee\left\{\nabla T(\bigwedge \circ P\neg)\Psi \mid \Psi \in T_\omega P_\omega Base(\alpha) \text{ and } (\alpha,\Psi) \notin \overline{T}(\not\in)\right\}$ $\lhd$

▶ `Formulate completeness`

6 Coalgebraic modalities via predicate liftings

In this chapter we take an approach to coalgebraic modal logic where the modalities are in 1-1 correspondence with so-called *predicate liftings* for the functor T. That is, with each set Λ of such predicate liftings we will associate a modal formalism ML_Λ for T-coalgebras. As a result this set-up is not completely uniform in the coalgebra type T, but it has some advantages over the approach based on relation lifting. First of all, the language of ML_Λ is completely standard, with a syntax that adds to the language of propositional logic an n-ary modality $\heartsuit_\lambda$ for each n-ary predicate lifting $\lambda \in \Lambda$. Second, there is no reason to restrict attention to functors that are smooth (preserve weak pullbacks). And finally, predicate liftings provide a uniform framework to many well-known variants of standard modal logic (including monotone and probabilistic modal logic, which were already mentioned in section 1.5).

Before we introduce the approach in full generality, we briefly discuss a few other concrete variants of standard modal logic that are covered by the approach.

6.1 Variants of modal logic

Example 6.1 (1) The *next-time operator* $\bigcirc$ of *linear time logic* is perhaps the most simple example. For its definition, consider models of the form (ω, V), where $V : \mathsf{Q} \to P(\omega)$ is a valuation on the set ω of natural numbers; the modality $\bigcirc$ is interpreted as follows:

$$\omega, V, n \Vdash \bigcirc\varphi \text{ iff } \omega, V, n+1 \Vdash \varphi.$$

Clearly the semantics of this operator can be generalised to arbitrary T-models for the identity functor $T = Id$.

(2) Similarly, on the binary tree 2^ω we can interpet two modalities $\bigcirc_0$ and $\bigcirc_1$, with the following interpretation:

$$2^\omega, V, u \Vdash \bigcirc_i\varphi \text{ iff } 2^\omega, V, u \cdot i \Vdash \varphi,$$

where $i \in \{0, 1\}$ and $V : \mathsf{Q} \to P(2^\omega)$ is a valuation on the set of finite words over the alphabet $2 = \{0, 1\}$.

The semantics of these operators can be generalised to arbitrary models for the binary tree functor $T = Id \times Id$.

(3) Graded modal logic is a version of modal logic that allows statements about the *number* of successors that satisfy a certain formula. Formally, interpreted in Kripke models, the modality $\diamond_{\geq k}$ has the following semantics:

$$\mathbb{S}, n \Vdash \diamond_{\geq k}\varphi \text{ iff } s \text{ has at least } k\ \varphi\text{-successors},$$

where a φ-successor of s is a state $t \in R(s)$ where φ holds. If we restrict attention to image-finite Kripke models, it also makes sense to introduce the following 'majority modality' M:

$$\mathbb{S}, n \Vdash M(\varphi, \psi) \text{ iff } s \text{ has more } \varphi\text{-successors than } \psi\text{-successors}.$$

Note that these modalities are not bisimulation invariant if we consider Kripke frames as coalgebras for the powerset functor. However, as we will see below, we may also see Kripke frames as coalgebras for the *bag* functor B (see the appendix for its definition), and for that functor both modalities will turn out to be invariant.

As we will see in this section, the common semantic pattern in many of these formalisms can be captured rather nicely in a coalgebraic framework by the notion of a *predicate lifting*.

6.2 Modalities via predicate liftings

To introduce the notion of a predicate lifting, we consider the example of probabilistic modal logic. In Example 1.8 we defined the semantics of the modality $\Diamond_q$ (with q a rational number in $[0,1]$) in a model $\mathbb{S} = (S, \sigma, V)$ for the distribution functor D, as follows:

$$\mathbb{S}, s \Vdash \Diamond_q \varphi \text{ iff } \sum_{u \in [\![\varphi]\!]} \sigma(s)(u) > q, \tag{41}$$

where we recall that $[\![\varphi]\!]$ denotes the *extension* of φ, i.e., the set $[\![\varphi]\!] = \{t \in S \mid \mathbb{S}, t \Vdash \varphi\}$ of states in $\mathbb{S}$ where φ is true. The way that we will be thinking of this definition now is as

$$\mathbb{S}, s \Vdash \Diamond_q \varphi \text{ iff } \sigma(s) \in \big\{\mu \in D(S) \mid \sum_{u \in [\![\varphi]\!]} \mu(u) > q\big\}, \tag{42}$$

or, in fact, as

$$\mathbb{S}, s \Vdash \Diamond_q \varphi \text{ iff } \sigma(s) \in \theta^q_S([\![\varphi]\!]), \tag{43}$$

where $\theta^q_S : PS \to PDS$ is defined by

$$\theta^q_S : U \mapsto \{\mu \in D(S) \mid \sum_{u \in U} \mu(u) > q\}.$$

In other words, we may think of the semantics of the modality $\Diamond_q$ as being indexed by a family θ^q of maps $\theta^q_S : PS \to PDS$, where each θ^q_S *lifts* a predicate on S (i.e., a subset of S) to a predicate on DS.

Now in principle we may associate a modality with each such family θ. However, as we will see below, it will make a lot of sense to impose the following uniformity condition on the family of maps: We will require that, for each map $f : S' \to S$, the following diagram commutes:

$$\begin{tikzcd} S' \arrow[d, "f"'] & PS' \arrow[r, "\theta_{S'}"] & PDS' \\ S & PS \arrow[u, "\breve{P}f"] \arrow[r, "\theta_S"'] & PDS \arrow[u, "\breve{P}Df"'] \end{tikzcd}$$

That is, we will require a 'proper' predicate lifting for the distribution functor to be a *natural transformation* $\theta : \breve{P} \stackrel{.}{\to} \breve{P}D$, where $\breve{P}$ is the *contravariant* powerset functor. In general, for an arbitrary set functor T we introduce the concept of a *predicate lifting* of some arbitrary but fixed finite arity, as follows.

Definition 6.2 A *predicate lifting* is a natural transformation of the form $\lambda : \breve{P}^n \stackrel{\cdot}{\to} \breve{P}T$, for some number $n \in \omega$ which we shall refer to as the *arity* of λ, notation: $n = \mathsf{ar}\lambda$. $\lhd$

Recall that the *naturality condition* on predicate liftings means that the following diagram commutes, for every function $f : S' \to S$:

$$\begin{tikzcd} S' \arrow[d, "f"'] & (PS')^n \arrow[r, "\lambda_{S'}"] & PTS' \\ S & (PS)^n \arrow[u, "(\breve{P}f)^n"] \arrow[r, "\lambda_S"'] & PTS \arrow[u, "\breve{P}Tf"'] \end{tikzcd} \tag{44}$$

The idea now is that with each predicate lifting λ we may associate a modality $\heartsuit_\lambda$, of the same arity as λ, and that the semantics of $\heartsuit_\lambda$ in a coalgebra (S, σ) is defined in terms of the predicate lifting λ itself (and the coalgebra map σ). The logics that we about to introduce are thus parametrised by a collection of predicate liftings; such collections we shall refer to as *modal signatures*.

Definition 6.3 A *modal signature* for a set functor T is nothing but a set of predicate liftings for T. Given such a collection Λ, and a set Q of proposition letters, the formulas of the modal logic $\mathrm{ML}_\Lambda(\mathsf{Q})$ are given by the following grammar:

$$\varphi \ ::= \ p \mid \bot \mid \neg\varphi \mid \varphi_0 \vee \varphi_1 \mid \heartsuit_\lambda(\varphi_0, \ldots, \varphi_{n-1}),$$

where $p \in \mathsf{Q}$, and λ is an n-ary predicate lifting for T. $\lhd$

We will also use the connectives $\wedge$, $\to$ and $\leftrightarrow$ as the standard abbreviations.

The semantics of the languages ML_Λ is defined in a uniform way, with the modality $\heartsuit_\lambda$ being interpreted 'by λ itself'.

Definition 6.4 Let $\mathbb{S} = (S, \sigma, V)$ be a T-model for some set functor T. We define the satisfaction relation $\Vdash_{\mathbb{S}} \subseteq S \times \mathrm{ML}_\Lambda(\mathsf{Q})$ by induction on the complexity of $\mathrm{ML}_\Lambda(\mathsf{Q})$-formulas. With all other clauses of this definition being standard, we only mention the clause for the coalgebraic modalities:

$$\mathbb{S}, s \Vdash \heartsuit_\lambda(\varphi_0, \ldots, \varphi_{n-1}) \text{ iff } \sigma(s) \in \lambda([\![\varphi_0]\!]^{\mathbb{S}}, \ldots, [\![\varphi_{n-1}]\!]^{\mathbb{S}}). \tag{45}$$

The notions of satisfiability, validity, and ML_Λ-equivalence are all defined in the obvious way (cf. Definition 5.10 for the ∇-logic); the relation of ML_Λ-equivalence between formulas will usually be denoted as $\equiv_\Lambda$ rather than as $\equiv_{\mathrm{ML}_\Lambda}$. $\lhd$

Remark 6.5 A succinct way of defining the semantics of the modality $\heartsuit_\lambda$ (45) is as follows:

$$[\![\heartsuit_\lambda(\varphi_0, \ldots, \varphi_{n-1})]\!]^{\mathbb{S}} := (\breve{P}\sigma)\big(\lambda([\![\varphi_0]\!]^{\mathbb{S}}, \ldots, [\![\varphi_{n-1}]\!]^{\mathbb{S}})\big). \tag{46}$$

Example 6.6 (1) The box and diamond modalities of standard modal logic can be seen as the coalgebraic modalities associated with the unary predicate liftings $\lambda^{\Box}, \lambda^{\Diamond} : \breve{P} \dot{\to} \breve{P}P$ given by

$$\begin{array}{ll} \lambda^{\Box}_S : & U \mapsto \{X \in PS \mid X \subseteq U\}, \\ \lambda^{\Diamond}_S : & U \mapsto \{X \in PS \mid X \cap U \neq \varnothing\}. \end{array}$$

We quickly verify that $\lambda^{\Box}$ satisfies the naturality condition (44). In order to show that $\lambda^{\Box}_{S'} \circ \breve{P}f = (\breve{P}Pf) \circ \lambda^{\Box}_S$, it suffices to show that the following identies hold, for every $U \in PS$:

$$\begin{array}{rcll} \lambda^{\Box}_{S'}(\breve{P}f(U)) & = & \{X' \in PS' \mid X' \subseteq \breve{P}f(U)\} & \text{(definition } \lambda^{\Box}_{S'}) \\ & = & \{X' \in PS' \mid fx' \in U, \text{ all } x' \in X'\} & \text{(obvious)} \\ & = & \{X' \in PS' \mid Pf(X') \subseteq U\} & \text{(obvious)} \\ & = & \{X' \in PS' \mid Pf(X') \in \lambda^{\Box}_S(U)\} & \text{(definition } \lambda^{\Box}_{S'}) \\ & = & (\breve{P}Pf)(\lambda^{\Box}_S(U)). & \text{(definition } \breve{P}Pf) \end{array}$$

(2) In the case of monotone modal logic, the box and diamond modalities are induced by the following predicate liftings $\mu^{\Box}, \mu^{\Diamond} : \breve{P} \dot{\to} \breve{P}M$:

$$\begin{array}{ll} \mu^{\Box}_S : & U \mapsto \{\sigma \in MS \mid U \in \sigma\}, \\ \mu^{\Diamond}_S : & U \mapsto \{\sigma \in MS \mid (S \setminus U) \notin \sigma\}. \end{array}$$

(3) The next-time operator $\bigcirc$ of linear temporal logic is obtained from the *identity lifting* $\lambda^{\bigcirc} : \breve{P} \to \breve{P}$:

$$\lambda^{\bigcirc}_S : U \mapsto U.$$

Example 6.7 Similarly to the case of probabilistic modal logic, for each $k \in \mathbb{N}$ we can define a predicate lifting $\lambda^{\geq k} : \breve{P} \to \breve{P}B$ for the bag functor B:

$$\lambda^{\geq k}_S : U \mapsto \{\mu : S \to \mathbb{N}^{\infty} \mid \sum_{u \in U} \mu(u) \geq k\}.$$

Now consider a Kripke model $\mathbb{S} = (S, \sigma, V)$, with $\sigma : S \to PS$. We may think of σ as a coalgebra map $\sigma^{\circ} : S \to BS$ for the functor B, by putting

$$\sigma^{\circ}(s)(t) := \begin{cases} 1 & \text{if } t \in \sigma(s), \\ 0 & \text{if } t \notin \sigma(s). \end{cases}$$

It is then straightforward to verify that for any Kripke model $\mathbb{S}$ we have

$$\mathbb{S}, n \Vdash \heartsuit_{\lambda^{\geq k}} \varphi \text{ iff } s \text{ has at least } k \ \varphi\text{-successors},$$

so that we can indeed think of graded modal logic as a coalgebraic logic.

Similarly, the 'majority modality' M of Example 6.1(3) can be seen as the coalgebraic modality that is induced by the *binary* predicate lifting $\lambda^M : \breve{P}^2 \dot{\to} \breve{P}B_{\omega}$ given by

$$\lambda^M_S : (U_0, U_1) \mapsto \{\mu : S \to \mathbb{N} \mid \sum_{u \in U_0} \mu(u) > \sum_{u \in U_1} \mu(u)\}.$$

Remark 6.8 Nullary predicate liftings exist. To unravel their meaning, note that we may think of any set of the form $(PS)^0$ as a *singleton* (more precisely, as the singleton consisting of the unique map $!_S : 0 \to PS$, where $0 = \varnothing$ is the empty set). Hence, we may identify a map $\lambda_S : (\breve{P}S)^0 \to \breve{P}TS$ with a *distinguished element* $\lambda_S(!_S)$ of the set $\breve{P}TS$, i.e., a subset of TS, and the naturality condition states that

$$(PTf)(\lambda_{S'}(!_{S'})) = \lambda_S(!_S),$$

for any map $f : S \to S'$.

Now suppose that λ is such a nullary predicate lifting, then the nullary modality $\heartsuit_\lambda$ associated with λ can be seen as a modal *constant*:

$$\mathbb{S}, \sigma, s \Vdash \heartsuit_\lambda \text{ iff } \sigma(s) \in \lambda_S(!_S).$$

Below we give two natural examples of this phenomenon.

Example 6.9 A natural example of a nullary modality is the constant $\surd$ that is sometimes used to indicate that a state in a finite deterministic automaton is accepting. Recall that these devices are coalgebras for the functor $2 \times Id^C$, and consider the nullary predicate lifting $\lambda^\surd : \breve{P}^0 \dot{\to} \breve{P}(2 \times Id^C)$ given by

$$\lambda^\surd_S(!_{S'}) := \{(i, f) \in 2 \times S^C \mid i = 1\}.$$

We obtain, for any state s in a deterministic automaton $\mathbb{S} = (S, \sigma)$, that s is accepting iff $\sigma(s) \in \lambda^\surd_S(!_S)$, so that we may think of the predicate lifting $\lambda^\surd$ as inducing the modality $\surd$.

Example 6.10 Let T be a functor, and Q a set of proposition letters. Recall that we may see a T-model (S, σ, V) over Q as a coalgebra (S, σ_V) for the functor $T_\mathsf{Q} = K_{P\mathsf{Q}} \times T$, where $\sigma_V : S \to P\mathsf{Q} \times TS$ is defined by putting

$$\sigma_V(s) := (V^\flat(s), \sigma(s)).$$

Now fix a proposition letter $q \in \mathsf{Q}$, and consider the following *nullary* predicate lifting $\lambda^q : \breve{P}^0 \dot{\to} \breve{P}T_\mathsf{Q}$ for this functor:

$$\lambda^q_S(!_S) := \{(c, \tau) \in P\mathsf{Q} \times TS \mid q \in c\}.$$

Furthermore, observe that the *modality* associated with this predicate lifting is also nullary, that is, a constant; its semantics in a T_Q-coalgebra (X, ξ) is given by

$$\mathbb{X}, \xi, x \Vdash \heartsuit_{\lambda^q} \text{ iff } q \in \pi_0(\lambda^q_X(!_X)).$$

In particular, this means that if (X, ξ) is of the form (S, σ_V) for some T-model (S, σ, V), we obtain that

$$(S, \sigma, V), s \Vdash q \text{ iff } (S, \sigma_V), s \Vdash \heartsuit_{\lambda^q}. \tag{47}$$

Based on this equivalence, we may think of proposition letters as modalities associated with nullary predicate liftings.

Remark 6.11 Given a set functor T and a set Q of proposition letters, we can now make the connection explicit between modal languages for T-models over Q on the one hand, and for T_{Q}-coalgebras on the other.

Based on Example 6.10, we see that there is a 1-1 connection between proposition letters in Q and nullary predicate liftings for T_{Q} that disect the '$P\mathsf{Q}$-part' c of an arbitrary object $(c, \tau) \in T_{\mathsf{Q}}S$.

To finish the picture, we now associate with an arbitary n-ary predicate lifting $\lambda : P^n \dot{\to} PT$ for T, an n-ary predicate lifting $\lambda' : P^n \dot{\to} PT_{\mathsf{Q}}$ for T_{Q} as follows:

$$\lambda'_S(X_0, \ldots, X_{n-1}) := \big\{(c, \tau) \in T_{\mathsf{Q}} \mid \tau \in \lambda_S(X_0, \ldots, X_{n-1})\big\}.$$

Then, given a modal signature Λ for T and a set Q of proposition letters, we define the signature $\Lambda + \mathsf{Q}$ for the functor T_{Q} by putting

$$\Lambda + \mathsf{Q} := \{\lambda' \mid \in \Lambda\} \cup \{\lambda^q \mid q \in \mathsf{Q}\},$$

and we leave it for the reader to verify that with this definition, we can see the language $\mathrm{ML}_\Lambda(\mathsf{Q})$ (for T-models over Q) and $\mathrm{ML}_{\Lambda+\mathsf{Q}}(\varnothing)$ (for T_{Q}-coalgebras) as notational variants of one another. In the sequel we will use this observation and use the language $\mathrm{ML}_\Lambda(\mathsf{Q})$ for T_{Q}-coalgebras; in particular, we will always simply write q instead of $\heartsuit_{\lambda^q}$.

6.3 Basic properties of ML_Λ

In this subsection we make some first observations about the modal logic of predicate liftings. First we show how the naturality condition (44) implies invariance under behavioural equivalence.

Theorem 6.12 *Let Λ be a modal signature for the set functor T. Then the language ML_Λ is invariant: Given any two pointed T-models $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ we have*

$$(\mathbb{S}, s) \simeq_T (\mathbb{S}', s') \text{ implies } (\mathbb{S}, s) \equiv_\Lambda (\mathbb{S}', s'). \tag{48}$$

Proof. Given the definition of behavioural equivalence, it suffices to prove that for any coalgebra morphism $f : \mathbb{S} \to \mathbb{S}'$, and any state $s \in S$, we have $(\mathbb{S}, s) \equiv_\Lambda (\mathbb{S}', fs)$. So fix $f : \mathbb{S} \to \mathbb{S}'$; we will show that every formula $\varphi \in \mathrm{ML}_\Lambda(\mathsf{Q})$ satisfies the following:

$$\mathbb{S}, s \Vdash \varphi \text{ iff } \mathbb{S}', fs \Vdash \varphi, \text{ for all states } s \in S,$$

or equivalently,

$$[\![\varphi]\!]^{\mathbb{S}} = (\breve{P}f)[\![\varphi]\!]^{\mathbb{S}'}. \tag{49}$$

We will prove (49) by a straightforward formula induction. Leaving the routine cases as an exercise to the reader, we focus on the case where $\varphi = \heartsuit_\lambda(\psi_0, ..., \psi_{n-1})$. Our proof makes use of the fact that the diagram below commutes. This should be obvious for the left rectangle, witnessing the naturality of λ; observe that the right rectangle is obtained by applying the (contravariant!) functor $\breve{P}$ to the diagram indicating that $f : S \to S'$ is a coalgebra morphism.

$$\begin{tikzcd}
(PS)^n \arrow[r, "\lambda_S"] & PTS \arrow[r, "\breve{P}\sigma"] & PS \\
(PS')^n \arrow[u, "(\breve{P}f)^n"] \arrow[r, "\lambda_{S'}"'] & PTS' \arrow[u, "\breve{P}Tf"'] \arrow[r, "\breve{P}\sigma'"] & PS' \arrow[u, "\breve{P}f"']
\end{tikzcd} \tag{50}$$

Now consider the following calculation:

$$\begin{aligned}
[\![\heartsuit_\lambda(\ldots,\psi_i,\ldots)]\!]^{\mathbb{S}} &= (\breve{P}\sigma)\lambda_S(\ldots,[\![\psi_i]\!]^{\mathbb{S}},\ldots) && \text{(semantics } \heartsuit_\lambda \text{ in } \mathbb{S}\text{, as in (46))} \\
&= (\breve{P}\sigma)\lambda_S(\ldots,(\breve{P}f)[\![\psi_i]\!]^{\mathbb{S}'},\ldots) && \text{(induction hypothesis)} \\
&= (\breve{P}\sigma)(\breve{P}Tf)\lambda_{S'}(\ldots,[\![\psi_i]\!]^{\mathbb{S}'},\ldots) && \text{(naturality of } \lambda\text{, (50) left)} \\
&= (\breve{P}f)(\breve{P}\sigma)\lambda_{S'}(\ldots,[\![\psi_i]\!]^{\mathbb{S}'},\ldots) && \text{(} f \text{ is a morphism, (50) right)} \\
&= (\breve{P}f)[\![\heartsuit_\lambda(\ldots,\psi_i,\ldots)]\!]^{\mathbb{S}'} && \text{(semantics } \heartsuit_\lambda \text{ in } \mathbb{S}'\text{, as in (46))}
\end{aligned}$$

showing that (49) holds for $\varphi = \heartsuit_\lambda(\psi_0, ..., \psi_{n-1})$ indeed. QED

Concerning the property of expressiveness, we find that a general result can be obtained if we put some constraints on the signature Λ.

Definition 6.13 Let Λ be a modal signature for a set functor T. We say that Λ is *separating* for T, if for all sets S and all pairs of distinct objects $\sigma_0, \sigma_1 \in TS$ there is a $\lambda \in \Lambda$ and a tuple $(A_0, \ldots, A_{n-1})$ such that *exactly* one of the two objects σ_i belongs to the set $\lambda_S(A_0, \ldots, A_{n-1})$. $\lhd$

Example 6.14 (1) The box relation lifting $\lambda^\Box$ is separating on its own, for the powerset functor P. To see this, consider two subsets $X, Y \in PS$. If X and Y are distinct, suppose without loss of generality that $Y \not\subseteq X$, so that $Y \notin \lambda^\Box(X) = \{U \in PS \mid U \subseteq X\}$.

(2) The predicate liftings associated with the graded modalities are jointly separating. To see this, consider two bags $\beta_0, \beta_1 : S \to \mathbb{N}^\infty$ over some set S. If β_0 and β_1 are distinct, then we must have $\beta_0(s) \neq \beta_1(s)$, for some $s \in S$. Without loss of generality we may assume that $\beta_0(s) < \beta_1(s)$, so that in particular, $\beta_0(s)$ belongs to $\mathbb{N}$, say, $\beta_0(s) = m$. Recall that $\lambda^{m+1}_S(\{s\}) = \{\beta \in BS \mid \beta(s) \geq m+1\}$, so that we find $\beta_0 \notin \lambda^{m+1}_S(\{s\})$ but $\beta_1 \in \lambda^{m+1}_S(\{s\})$.

Theorem 6.15 *Let Λ be a separating modal signature for the set functor T. Then the language* ML_Λ *is expressive on the class of image-finite T-coalgebras: Given any two pointed T_ω-models $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ we have*

$$(\mathbb{S}, s) \equiv_\Lambda (\mathbb{S}', s') \textit{ implies } (\mathbb{S}, s) \simeq_T (\mathbb{S}', s'). \tag{51}$$

Proof. For notational simplicity we will confine ourselves to a setting where all predicate liftings in Λ are unary, leaving the (routine) generalisation to an arbitrary signature as an exercise. It will also be convenient to assume that T preserves inclusions. Furthermore, we will only treat the special case where the coalgebras $\mathbb{S}$ and $\mathbb{S}'$ coincide (i.e., $\mathbb{S} = \mathbb{S}'$); the general case, where the two coalgebras are distinct, can easily be reduced to this by considering their disjoint union.

So let Λ be a separating set of unary predicate liftings for some set functor T, and let $\mathbb{S} = (S, \sigma, V)$ be an image-finite T-model, that is, $\sigma : S \to T_\omega S$. To avoid notational clutter we will simply write $\equiv$ for the equivalence relation $\equiv_\Lambda$.

Our aim is prove that $s_0 \equiv s_1$ implies $s_0 \simeq s_1$, for all $s_0, s_1 \in \mathbb{S}$. Clearly then it suffices to show that the relation $\equiv$ is contained in the kernel of some coalgebra morphism. We will show that in fact we may define coalgebra structure $\overline{\sigma} : \overline{S} \to T\overline{S}$ on the set $\overline{S}$ of $\equiv$-cells in such a way that the quotient map $q : S \to \overline{S}$ becomes a coalgebra morphism:

$$
\begin{tikzcd}
S \arrow[r, "q"] \arrow[d, "\sigma"'] & \overline{S} \arrow[d, dotted, "\overline{\sigma}"] \\
TS \arrow[r, "Tq"'] & T\overline{S}
\end{tikzcd}
$$

Now suppose that we can show that

$$q(s_0) = q(s_1) \text{ implies } (Tq)(\sigma s_0) = (Tq)(\sigma s_1), \tag{52}$$

then putting

$$\overline{\sigma}(\overline{s}) := (Tq)(\sigma s)$$

would give a correctly defined map, for which the quotient map q is trivially a coalgebra morphism.

That is, we have reduced our problem to finding a proof for (52), and to this aim we reason by contraposition: Assuming that

$$(Tq)(\sigma s_0) \neq (Tq)(\sigma s_1), \tag{53}$$

we will show that $q(s_0) \neq q(s_1)$ (that is, $s_0 \not\equiv_\Lambda s_1$). It follows by separation from (53) that there is some $\lambda \in \Lambda$ and some $A \subseteq \overline{S}$ such that (without loss of generality) we have

$$(Tq)(\sigma s_0) \in \lambda_{\overline{S}}(A), \text{ but } (Tq)(\sigma s_1) \notin \lambda_{\overline{S}}(A). \tag{54}$$

Our purpose is now is find a formula witnessing this, in the sense that this formula holds at s_0 but not at s_1: this would show indeed that $s_0 \not\equiv s_1$, and so $q(s_0) \neq q(s_1)$. Note that $A \subseteq \overline{S}$ simply means that A is a collection of equivalence classes.

Now if we would have $\bigcup A = [\![\varphi]\!]$ we would be done immediately since in this case we could prove that $\mathbb{S}, s_0 \Vdash \heartsuit_\lambda \varphi$ while $\mathbb{S}, s_1 \not\Vdash \heartsuit_\lambda \varphi$. To see this, note that we have $s_i \Vdash \heartsuit_\lambda \varphi$ iff $\sigma(s_i) \in \lambda_S([\![\varphi]\!])$ and that $\bigcup A = [\![\varphi]\!]$ implies $\lambda_S([\![\varphi]\!]) = \lambda_S(\bigcup A) = \lambda_S(\breve{P}q(A)) = (\breve{P}Tq)(\lambda_{\overline{S}}(A))$, where the last identity is by naturality of λ. It is thus immediate by (54) that $\sigma s_0 \in \lambda_S([\![\varphi]\!])$ but $\sigma s_1 \notin \lambda_S([\![\varphi]\!])$, as required.

In the general case we need to work harder. But since $\mathbb{S}$ is a T_ω-coalgebra, and T preserves inclusions, there is a *finite* subset $X \subseteq S$ such that both σs_0 and σs_1 belong to the set TX.

We leave it for the reader to verify that there is a formula $\varphi \in \mathrm{ML}_\Lambda$ that characterizes, *within* X, the union $\bigcup A$, in the sense that

$$\text{for all } x \in X: \quad \mathbb{S}, x \Vdash \varphi \text{ iff } x \in \bigcup A,$$

or equivalently, since $x \in \bigcup A$ is another way of saying that $q(x) \in A$:

$$X \cap [\![\varphi]\!] \;=\; X \cap (\breve{P}q)A. \tag{55}$$

We now claim that, for this formula φ, we have

$$\mathbb{S}, s_0 \Vdash \heartsuit_\lambda\varphi \text{ but } \mathbb{S}, s_1 \not\Vdash \heartsuit_\lambda\varphi. \tag{56}$$

To prove this, we first observe that by the semantics of $\heartsuit_\lambda$ we have that $\mathbb{S}, s_i \Vdash \heartsuit_\lambda\varphi$ iff $\sigma(s_i) \in \lambda_S([\![\varphi]\!])$, while it follows from (54) that $\sigma(s_0) \in (\breve{P}Tq)\lambda_{\overline{S}}(A)$ but $\sigma(s_1) \notin (\breve{P}Tq)\lambda_{\overline{S}}(A)$. Hence, because both $\sigma(s_0)$ and $\sigma(s_1)$ belong to TX, it suffices to show that

$$TX \cap (\breve{P}Tq)\lambda_{\overline{S}}(A) \;=\; TX \cap \lambda_S[\![\varphi]\!]. \tag{57}$$

We will establish this by chasing the diagram below, where we use a trick to interpret the intersection with X in (55) and with TX in (57) using the inclusion map $\iota : X \hookrightarrow S$. That is, we observe that $\breve{P}\iota : PS \to PX$ is given by $U \mapsto X \cap U$; as a consequence another way of formulating (55) is:

$$(\breve{P}\iota)([\![\varphi]\!]) = (\breve{P}\iota)(\breve{P}q)(A). \tag{58}$$

Similarly, since T preserves inclusions we have that $T\iota : TX \to TS$ is the inclusion map witnessing that $TX \subseteq TS$, and so $\breve{P}T\iota : PTS \to PTX$ is given by $\Sigma \mapsto (TX) \cap \Sigma$.

$$\begin{tikzcd}
\overline{S} & P\overline{S} \arrow[r, "\lambda_{\overline{S}}"] \arrow[d, "\breve{P}q"'] & PT\overline{S} \arrow[d, "\breve{P}Tq"] \\
S \arrow[u, "q"] & PS \arrow[r, "\lambda_S"] \arrow[d, "\breve{P}\iota"'] & PTS \arrow[d, "\breve{P}T\iota"] \\
X \arrow[u, "\iota"] & PX \arrow[r, "\lambda_X"] & PTX
\end{tikzcd}$$

$$\begin{aligned}
& TX \cap (\breve{P}Tq)\lambda_{\overline{S}}(A) \\
&= (\breve{P}T\iota)(\breve{P}Tq)\lambda_{\overline{S}}(A) && \text{(just discussed)} \\
&= \lambda_X(\breve{P}\iota)(\breve{P}q)(A) && \text{(naturality of } \lambda) \\
&= \lambda_X(\breve{P}\iota)([\![\varphi]\!]) && \text{(see (58))} \\
&= (\breve{P}T\iota)\lambda_S([\![\varphi]\!]) && \text{(naturality of } \lambda) \\
&= TX \cap \lambda_S[\![\varphi]\!] && \text{(just discussed)}
\end{aligned}$$

This proves (57), and therefore (56). That is, we have shown that $s_0 \not\equiv s_1$, on the assumption that $(Tq)(\sigma s_0) \neq (Tq)(\sigma s_1)$. This means that (52) holds, and as we argued already, this suffices to prove the Theorem. QED

6.4 Finite model property

In this subsection we will show that the coalgebraic modal logic ML_Λ has the (strong) finite model property. That is, we will show that any satisfiable ML_Λ-formula φ can in fact be satisfied in a *finite* coalgebra of which the size (number of states) is exponentially bounded by the size of φ. We will prove this result by adapting the method of filtration, which is well known in the theory of standard modal logic, to the more general coalgebraic setting.

First we need some preliminary definitions.

Definition 6.16 The collection $\mathtt{Sfor}(\varphi)$ of *subformulas* of a ML_Λ-formula φ is defined in the standard way. The *size* $|\varphi|$ of a φ is defined as its number of subformulas: $|\varphi| := |\mathtt{Sfor}(\varphi)|$.

A set of formulas Σ is called *subformula-closed* if it is closed under taking subformulas, that is, if $\mathtt{Sfor}(\varphi) \subseteq \Sigma$ for all $\varphi \in \Sigma$. $\lhd$

The idea behind the method of filtration is fairly simple: given a subformula-closed set Σ and a T-model $\mathbb{S} = (S, \sigma, V)$, define on S a suitable equivalence relation $\equiv_\Sigma$ of finite index, build a new T-model $\overline{\mathbb{S}}$ on the finite set of $\equiv_\Sigma$-cells, and show that any state $s \in S$ satisfies the same formulas in $\mathbb{S}$ as its cell $\overline{s}$ does in the filtrated model $\overline{\mathbb{S}}$.

Definition 6.17 Let Σ be a finite, subformula closed set of formulas in $\mathrm{ML}_\Lambda(\mathsf{Q})$, and let $\mathbb{S} = (S, \sigma, V)$ be a T-model. We define $\equiv_\Sigma \subseteq S \times S$ as the equivalence relation given by

$$s \equiv_\Sigma t \text{ iff for all } \varphi \in \Sigma : \mathbb{S}, s \Vdash \varphi \iff \mathbb{S}, t \Vdash \varphi,$$

and denote the $\equiv_\Sigma$-cell of a state s as $\overline{s}$. We also let $\overline{S} := \{\overline{s} \mid s \in S\}$ denote the set of cells, and let $q : s \mapsto \overline{s}$ denote the quotient map $q : S \to \overline{S}$. $\lhd$

In order to define a coalgebra map $\overline{\sigma} : \overline{S} \to T\overline{S}$, we may pick any *choice function* $c : \overline{S} \to S$, and define $\overline{\sigma} := Tq \circ \sigma \circ c$, cf. the diagram below:

$$\begin{tikzcd}
S \arrow[r, bend right, "q"'] \arrow[d, "\sigma"] & \overline{S} \arrow[l, dotted, bend right, "c"'] \arrow[d, dotted, "\overline{\sigma}"] \\
TS \arrow[r, "Tq"'] & T\overline{S}
\end{tikzcd}$$

Here we call $c : \overline{S} \to S$ a choice function if c picks an element from each $\equiv_\Sigma$-cell; in other words, we require that $q \circ c = \mathsf{id}_{\overline{S}}$ and $\ker(c \circ q) \subseteq \equiv_\Sigma$.

Note that while the 'outer' rectangle of the above diagram commutes by definition, the 'inner' one need not commute: it will generally not be possible to define a coalgebra map on $\overline{S}$ for which the quotient map q is a coalgebra morphism. Fortunately, for our purposes we don't need such a coalgebra morphism.

Definition 6.18 Let Σ be a finite, subformula closed set of formulas in $\mathrm{ML}_\Lambda(\mathsf{Q})$, and let $\mathbb{S} = (S, \sigma, V)$ be a T-model. A Σ-*filtration* of $\mathbb{S}$ is any T-model $\overline{\mathbb{S}} = (\overline{S}, \overline{\sigma}, \overline{V})$ such that
(1) $\overline{S} = S/{\equiv_\Sigma}$ is the class of $\equiv_\Sigma$-cells,
(2) $\overline{\sigma} = Tq \circ \sigma \circ c$ for some choice function $c : \overline{S} \to S$, and
(3) $\overline{V}(q) = \{\overline{s} \mid s \in V(q)\}$ for $q \in \Sigma \cap \mathsf{Q}$. $\lhd$

Observe that filtrations are not unique, they depend in particular on the choice of the choice function c.

We can now prove the following Filtration Lemma.

Theorem 6.19 (Filtration Lemma) *Let Λ be a modal signature for a set functor T, and let $\Sigma \subseteq \mathrm{ML}_\Lambda$ be a finite subformula-closed set of formulas. Furthermore, let $\mathbb{S} = (S, \sigma, V)$ be a T-model, and let $\overline{\mathbb{S}} = (\overline{S}, \overline{\sigma}, \overline{V})$ be a Σ-filtration of $\mathbb{S}$. Then for all formulas $\varphi \in \Sigma$ we have*

$$\mathbb{S}, s \Vdash \varphi \textit{ iff } \overline{\mathbb{S}}, \overline{s} \Vdash \varphi, \textit{ for all states } s \in S. \tag{59}$$

Proof. First note that since $s \equiv c(\overline{s})$ for all $s \in S$, the statement (59) is equivalent to

$$[\![\varphi]\!]^{\overline{\mathbb{S}}} = (\breve{P}c)[\![\varphi]\!]^{\mathbb{S}}, \tag{60}$$

for all formulas $\varphi \in \Sigma$. It therefore suffices to prove (60), and we will do so by a straightforward formula induction.

Second, note as well that the statement (59) is also equivalent to

$$[\![\varphi]\!]^{\mathbb{S}} = (\breve{P}q)[\![\varphi]\!]^{\overline{\mathbb{S}}}, \tag{61}$$

in fact for *all* formulas. This means that we may use (61) as the inductive hypothesis in our inductive proof.

Turning to the actual inductive proof of (60), we only consider the inductive step where $\varphi = \heartsuit_\lambda(\psi_0, \ldots, \psi_{n-1})$, and for notational simplicity we confine ourselves to the case where λ is unary, i.e, $\varphi = \heartsuit_\lambda \psi$ for some formula ψ to which the inductive hypothesis applies. Now consider the following diagram:

$$\begin{tikzcd}
P\overline{S} \arrow[r, "\lambda_{\overline{S}}"] \arrow[d, bend right, "\breve{P}q"'] & PT\overline{S} \arrow[r, "\breve{P}\overline{\sigma}"] \arrow[d, "\breve{P}Tq"'] & P\overline{S} \\
PS \arrow[r, "\lambda_S"'] & PTS \arrow[r, "\breve{P}\sigma"'] & PS \arrow[u, bend right, "\breve{P}c"']
\end{tikzcd} \tag{62}$$

$$\begin{aligned}
[\![\heartsuit_\lambda \psi]\!]^{\overline{\mathbb{S}}} &= (\breve{P}\overline{\sigma})\lambda_{\overline{S}}([\![\psi]\!]^{\overline{\mathbb{S}}}) && \text{(semantics of } \heartsuit_\lambda) \\
&= (\breve{P}c)(\breve{P}\sigma)(\breve{P}Tq)\lambda_{\overline{S}}([\![\psi]\!]^{\overline{\mathbb{S}}}) && \text{(definition } \overline{\sigma}) \\
&= (\breve{P}c)(\breve{P}\sigma)\lambda_S(\breve{P}q)([\![\psi]\!]^{\overline{\mathbb{S}}}) && \text{(naturality of } \lambda) \\
&= (\breve{P}c)(\breve{P}\sigma)\lambda_S([\![\psi]\!]^{\mathbb{S}}) && \text{(induction hypothesis (61))} \\
&= (\breve{P}c)[\![\heartsuit_\lambda \psi]\!]^{\mathbb{S}} && \text{(semantics of } \heartsuit_\lambda)
\end{aligned}$$

In other words, we have established (60) for $\varphi = \heartsuit_\lambda \psi$, as required. QED

Corollary 6.20 (Strong Finite Model Property) *Let φ be a formula in ML_Λ, where Λ is a modal signature for a set functor T. If φ is satisfiable in some T-model, then it is satisfiable in a finite T-model (S, σ, V) such that $|S| \leq 2^{|\varphi|}$.*

Proof. Fix an ML_Λ-formula φ, and let $\Sigma := \mathtt{Sfor}(\varphi)$. It follows by the filtration lemma that φ, if satisfiable in some pointed T-model $(\mathbb{S}, s)$, also holds at the state $\overline{s}$, in any filtration $\overline{\mathbb{S}}$ of $\mathbb{S}$. This proves the theorem, since it easily follows from the definion of the relation $\equiv_\Sigma$ that $|\overline{S}| \leq |P\Sigma| = 2^{|\varphi|}$. QED

6.5 Predicate liftings as coalgebra type changers

This short section presents a slightly different perspective in which predicate liftings provide natural ways to transform T-coalgebras to neighbourhood frames. We first consider a simplified version, in which T-coalgebras are transformed to Kripke frames.

Example 6.21 Define a *natural relation* for T to be a natural transformation $\mu : T \dot{\rightarrow} P$.

Given such a natural relation $\mu : T \dot{\rightarrow} P$, we can transform a T-coalgebra $\mathbb{S} = (S, \sigma)$ into a Kripke frame $\mathbb{S}^\mu := (S, \mu_S \circ \sigma)$. By naturality of μ, any T-homomorphism $f : \mathbb{S} \to \mathbb{S}'$ is also a bounded morphism $f : \mathbb{S}^\mu \to (\mathbb{S}')^\mu$. To check this, one may easily verify that $Pf \circ (\mu_S \circ \sigma) = (\mu_{S'} \circ \sigma') \circ f$ by chasing the diagram below:

$$
\begin{tikzcd}
S \arrow[r, "\sigma"] \arrow[d, "f"'] & TS \arrow[r, "\mu_S"] \arrow[d, "Tf"] & PS \arrow[d, "Pf"] \\
S' \arrow[r, "\sigma'"] & TS' \arrow[r, "\mu_{S'}"] & PS'
\end{tikzcd}
$$

Connecting this to logic, with any natural relation μ we may associate a modality $\langle \mu \rangle$ for T-coalgebras, with the following interpretation:

$$\mathbb{S}, s \Vdash \langle \mu \rangle \varphi \text{ iff } [\![\varphi]\!]^{\mathbb{S}} \cap \mu_S \sigma(s) \neq \varnothing.$$

As an example, recall from Fact 5.16(2) that for smooth and standard functors T, we have a natural transformation $Base^T : T_\omega \dot{\rightarrow} P_\omega$. Hence, we may take the *Base* operation as a way to transform T_ω-coalgebras into P_ω-coalgebras, that is, image-finite Kripke frames. Naturality ensures that every morphism between T-coalgebras is also a bounded morphism between the underlying Kripke frames.

As we will see now, predicate liftings can be seen as generalisations of this phenomenon, where we move from Kripke frames to the more general setting of neighbourhood frames. For this general setting we introduce the *transpose* of a predicate lifting, cf. Definition A.4. This notion is based on the correspondence between maps $A \to PB$ and maps $B \to PA$ — we have seen this correspondence already in the coalgebraic presentation of a valuation $V : \mathsf{Q} \to PS$ as a colouring $V^\flat : S \to P\mathsf{Q}$.

Definition 6.22 Given a map $\alpha : A \to PB$ we define its *transposed map* $\alpha^\flat : B \to PA$ by putting $\alpha^\flat(b) := \{a \in A \mid b \in \alpha(a)\}$.

Extending this definition, given an n-ary predicate lifting for the set functor T, we define its *transpose* $\lambda^\flat$ as the set-indexed family of maps

$$\lambda^\flat_S : TS \to P(P^n(S))$$

given by $\lambda^\flat_S(\sigma) := (\lambda_S)^\flat(\sigma)$. $\lhd$

By the naturality of predicate liftings we obtain the following proposition, which shows that predicate liftings indeed generalise the natural relations of Example 6.21.

Proposition 6.23 *If $\lambda : P^n \dot{\rightarrow} PT$ then $\lambda^\flat$ is a natural transformation*

$$\lambda^\flat : T \dot{\rightarrow} \breve{P} \circ \breve{P}^n.$$

It follows from this proposition that any predicate lifting λ induces a transformation of T-coalgebras to n-ary neighbourhood frames. We confine attention to the unary case.

Definition 6.24 Let $\lambda : \breve{P} \stackrel{\cdot}{\to} \breve{P}T$ be a unary predicate lifting for the set functor T. Given a T-coalgebra $\mathbb{S} = (S, \sigma)$, we let $\mathbb{S}^\lambda$ denote the neighbourhood frame $\mathbb{S}^\lambda := (S, \lambda^\flat \circ \sigma)$; given a function $f : S \to S'$ we define $f^\lambda := f$. ◁

The following proposition is easy to verify.

Proposition 6.25 *Let $\lambda : \breve{P} \stackrel{\cdot}{\to} \breve{P}T$ be a unary predicate lifting for the set functor T. Then the construction $(\cdot)^\lambda$ is a functor from the category of T-coalgebras to the category of neighbourhood frames (N-coalgebras).*

The following proposition, which is easy to verify, provides a slightly different perspective on the concept of separation.

Proposition 6.26 *Let Λ be a modal signature for a set functor T. Then Λ is separating iff, for every set S, the collection $(\lambda^\flat_S)_{\lambda \in \Lambda}$ of transposed functions is jointly injective (i.e., for any pair of distinct objects $\tau_0, \tau_1 \in TS$ there is a $\lambda \in \Lambda$ such that $\lambda^\flat_S(\tau_0) \neq \lambda^\flat_S(\tau_1)$).*

6.6 Predicate liftings and the Yoneda lemma

In the final subsection of this chapter we take a slightly different perspective on the modalities that are given by predicate liftings, seeing them as describing certain *admissible patterns.* This perspective will also reveal *how many* predicate liftings of each different arity there are.

The key observation here is that there is a natural bijection between PS (subsets of S) and 2^S (characteristic functions on S).

Definition 6.27 Given a subset $X \subseteq S$, we define the *characteristic function of X* as the map $\chi^S_X : S \to 2$ given by

$$\chi^S_X(s) := \begin{cases} 1 & \text{if } s \in X \\ 0 & \text{if } s \notin X. \end{cases}$$

In case S is understood, we may drop the superscript 'S'. Conversely, given a map $\chi : S \to 2$, we shall call $\chi^{-1}(1) \in PS$ the *subset determined by* χ. ◁

Remark 6.28 This correspondence reaches far enough for us to think of the contravariant power set functor as the functor 2^- that associates with a set S the collection 2^S of functions from S to $2 = \{0, 1\}$, and with an arrow $f : S' \to S$ the 'precomposition map' that assigns to an arbitrary characteristic function $\chi : S \to 2$ the function $\chi \circ f : S' \to 2$.

From this point of view, a unary predicate lifting is a way of transforming arrows $S \to 2$ into arrows $TS \to 2$. In particular, any arrow $\gamma : T2 \to 2$ (that is, any characteristic function χ_Γ corresponding to a subset $\Gamma \subseteq T2$), induces a unary predicate lifting: Given an arrow $\chi : S \to 2$, simply consider the arrow $\gamma \circ T\chi$, as in the diagram below:

$$S \xrightarrow{\chi} 2$$

$$\begin{tikzcd} TS \arrow[r, "T\chi"] \arrow[rr, bend left, "\gamma\circ T\chi"] & T2 \arrow[r, "\gamma"] & 2 \end{tikzcd}$$

Formulated in terms of subsets rather than characteristic functions, we arrive at the following definition.

Definition 6.29 Given an object $\Gamma \subseteq T2$, let $\widehat{\Gamma}$ be the following set-indexed family of operations. For a set S, we define

$$\widehat{\Gamma}_S : PS \to PTS,$$

by putting, for any $X \subseteq S$:

$$\widehat{\Gamma}_S(X) := \{\sigma \in TS \mid (T\chi^S_X)(\sigma) \in \Gamma\},$$

where $\chi^S_X : S \to 2$ is the characteristics map associated with X. Where $\widehat{\Gamma}$ is a predicate lifting, we shall denote its associated modality as $\heartsuit_\Gamma$ rather than as $\heartsuit_{\widehat{\Gamma}}$. $\lhd$

Remark 6.30 Taking a glance at the *modalities* that are induced by subsets of the set $T2$, we consider the interpretation of the formula $\heartsuit_\Gamma\varphi$ in the T-model $\mathbb{S} = (S, \sigma, V)$. If we represent the set $[\![\varphi]\!]$ with its characteristic function $\chi^S_{[\![\varphi]\!]}$, applying the functor T to this arrow we obtain

$$TS \xrightarrow{T\chi^S_{[\![\varphi]\!]}} T2\ .$$

Now think of the elements of $T2$ as 'T-patterns', then the above arrow associates a T-pattern with each object $\tau \in TS$. We can say that the formula $\heartsuit_\Gamma\varphi$ holds at s if the pattern $\big(T\chi^S_{[\![\varphi]\!]}\big)(\sigma(s))$ associated with $\sigma(s) \in TS$ is *admissible*, i.e., belongs to the set Γ, or, equivalently, that $\big(\chi^{T2}_\Gamma \circ T\chi^S_{[\![\varphi]\!]}\big)(\sigma(s)) = 1$.

Example 6.31 As an example, consider the binary tree functor $Id \times Id$. The set $T2$ consists of four patterns: $T2 = \{(0,0), (0,1), (1,0), (1,1)\}$. As an example, take the set $\Gamma = \{(0,0), (1,1)\}$; it is not hard to see that it induces the predicate lifting $\widehat{\Gamma} : \breve{P} \stackrel{.}{\to} \breve{P}\circ(Id\times Id)$ given by

$$\widehat{\Gamma}_S(X) := X \times X \cup (S \setminus X) \times (S \setminus X).$$

The associated modality $\heartsuit_\Gamma$ has the following semantics:

$$\mathbb{S}, s \Vdash \heartsuit_\Gamma\varphi \text{ iff the two successors of } s \text{ either both satisfy or both falsify } \varphi.$$

Proposition 6.32 *For any $\Gamma \subseteq T2$, the collection $\widehat{\Gamma}$ of maps constitutes a predicate lifting $\widehat{\Gamma} : \breve{P} \stackrel{.}{\to} \breve{P}T$.*

Proof. Given a map $f : S' \to S$, we need to check that the following diagram commutes:

$$\begin{tikzcd}
S' \arrow[d, "f"'] & PS' \arrow[r, "\widehat{\Gamma}_{S'}"] & PTS' \\
S & PS \arrow[u, "\breve{P}f"] \arrow[r, "\widehat{\Gamma}_S"'] & PTS \arrow[u, "\breve{P}Tf"']
\end{tikzcd}$$

This follows by the following chain of identities, for an arbitrary subset $X \subseteq S$:

$$\begin{aligned}
(\widehat{\Gamma}_{S'} \circ \breve{P}f)(X) &= \widehat{\Gamma}_{S'}(\breve{P}f(X)) && \text{(obvious)}\\
&= \{\sigma' \in TS' \mid (T\chi^{S'}_{\breve{P}f(X)})(\sigma') \in \Gamma\} && \text{(definition } \widehat{\Gamma}_{S'})\\
&= \{\sigma' \in TS' \mid (T(\chi^{S}_{X} \circ f))(\sigma') \in \Gamma\} && (*)\\
&= \{\sigma' \in TS' \mid (T\chi^{S}_{X})((Tf)(\sigma')) \in \Gamma\} && \text{(functoriality of } T)\\
&= \{\sigma' \in TS' \mid (Tf)(\sigma') \in \widehat{\Gamma}_{S}(X)\} && \text{(definition } \widehat{\Gamma}_{S})\\
&= (\breve{P}Tf)(\widehat{\Gamma}_{S}(X)) && \text{(definition } \breve{P}Tf)\\
&= (\breve{P}Tf \circ \widehat{\Gamma}_{S})(X) && \text{(obvious)}
\end{aligned}$$

Here the identity (*) is immediate by the observation that

$$\chi^{S}_{X} \circ f = \chi^{S'}_{\breve{P}f(X)}$$

as is revealed by a straightforward verification, for an arbitrary element $s' \in S'$: $\chi^{S}_{X} \circ f(s') = 1$ iff $f(s') \in X$ iff $s' \in f^{-1}(X) = (\breve{P}f)(X)$ iff $\chi^{S'}_{\breve{P}f(X)}(s') = 1$. QED

Interestingly, we may prove that *all* unary predicate liftings are of this form, and this result generalises to predicate liftings of arbitrary arity. This is the main content of the following theorem, which is in fact the instantiation of the well-known *Yoneda Lemma* to the setting of predicate liftings.

Theorem 6.33 *For any set functor T there is a natural bijection between the set of n-ary predicate liftings for T and the power set of $T(Pn)$.*

The key observations underlying the proof of this theorem are the natural correspondences

$$(PS)^n \sim (2^S)^n \sim (2^n)^S \sim (Pn)^S$$

between n-tuples of subsets of S, n-tuples of characteristic functions, maps from S to 2^n, and maps from S to Pn.

Proof. We confine the proof of this result to providing an n-ary predicate lifting for each subset of TPn, and vice versa.

So let $\Gamma \subseteq T(Pn)$ be a set of admissible n-ary T-patterns. The associated predicate lifting $\widehat{\Gamma} : P \dot{\to} PT$ is obtained by a straightforward generalisation of Definition 6.29. For the details, take an arbitrary tuple $X = (X_0, \ldots, X_{n-1}) \in (PS)^n$ of subsets of S, and represent this tuple as the map $\chi_X := (\chi_0, \ldots, \chi_{n-1})$ of associated characteristic functions, then we have $\chi_X : S \to 2^n$, and so $T\chi_X : TS \to T(2^n)$. Now put

$$\widehat{\Gamma}_S(X_0, \ldots, X_{n-1}) := \{\tau \in TS \mid (T\chi_X)(\tau) \in \Gamma\}.$$

We leave it for the reader to verify that this collection of maps indeed provides a natural transformation $\widehat{\Gamma} : \breve{P}^n \to \breve{P}T$.

For the opposite direction, let $\lambda : \breve{P}^n \dot{\to} \breve{P}T$ be an n-ary predicate lifting. Our aim is to find a subset $\Theta_\lambda \subseteq T(2^n)$ such that $\lambda = \widehat{\Theta_\lambda}$.

The easiest way to proceed is by thinking of λ_S as a way to transform arrows $S \to 2^n$ into arrows $TS \to 2$, that is, $\lambda_S : (2^n)^S \to 2^{TS}$. To find the object Θ_λ, we take a *special* set S, viz., the set 2^n itself, and, as input for λ, a *special* arrow, viz., the *identity* arrow $\mathsf{id}_{2^n} : 2^n \to 2^n$. Then we let $\Theta_\lambda \subseteq T(2^n)$ be the subset of $T(2^n)$ that is determined by the image of this arrow id_{2^n} under λ.

If we think of λ as a set-indexed family of maps $\lambda_S : (PS)^n \to PTS$, we may define

$$\Theta_\lambda := \lambda(\mathcal{U}),$$

where $\mathcal{U}$ is the distinguished element of the set $P^n(Pn)$ corresponding to the identity map id_{2^n}, that is, $\mathcal{U} = (\mathcal{U}_0, \ldots, \mathcal{U}_{n-1})$ with

$$\mathcal{U}_i := \{U \subseteq n \mid i \in U\}.$$

As mentioned, we leave it for the reader to verify that the maps just defined are each other's inverse, i.e., that $\lambda = \widehat{\Theta_\lambda}$ for all predicate liftings λ, and that $\Gamma = \Theta_{\widehat{\Gamma}}$ for all $\Gamma \in PTn$, some $n \in \omega$. QED

A Appendix: Basic mathematical definitions

In this second appendix we fix notation and terminology for some basic mathematical concepts. First we consider sets, functions and relations.

A binary relation between two sets X and Y is nothing but a set $R \subseteq X \times Y$ of pairs from X and Y, respectively.

Definition A.1 Let $f : X \to Y$ be a function. We let $\mathsf{Gr} f := \{(x, y) \in X \times Y \mid y = fx\}$ denote the *graph* of f, and define $f[X] := \{fx \mid x \in X\}$. $\lhd$

Definition A.2 Given a relation $R \subseteq X \times Y$, we denote the *domain* $\mathsf{Dom}(R) \subseteq X$ and *range* $\mathsf{Ran}(R) \subseteq Y$ of R by the followings sets:

$$\begin{array}{lcl} \mathsf{Dom}(R) & := & \{x \in X \mid (x, y) \in R \text{ for some } y \in Y\} \\ \mathsf{Ran}(R) & := & \{y \in Y \mid (x, y) \in R \text{ for some } x \in X\}, \end{array}$$

respectively, and we denote by $\pi_0^R : R \to X$ and $\pi_1^R : R \to Y$ the projection maps associated with R. Given subsets $X' \subseteq X$, $Y' \subseteq Y$, the *restriction* of R to X' and Y' is given as

$$R{\restriction}_{X' \times Y'} := R \cap (X' \times Y').$$

The *converse* of R is defined as the relation $R^{\smile} \subseteq Y \times X$ given by

$$R^{\smile} := \{(y, x) \in Y \times X \mid (x, y) \in R\}.$$

The *composition* of two relations $R \subseteq X \times Y$ and $R' \subseteq Y \times Z$ is denoted by $R\,;R'$ and defined as

$$R\,;R' := \{(x, z) \in X \times Z \mid (x, y) \in R \text{ and } (y, z) \in R', \text{ for some } y \in Y\}.$$

Finally, we let

$$\Delta_X := \{(x, x) \in X \times X \mid x \in X\}$$

denote the *diagonal relation* on a set X. $\lhd$

Proposition A.3 *The transformations below constitute a bijection between the set $P(X \times Y)$ of binary relations between X and Y and the collection $(PY)^X$ of maps from X to the power set of Y:*

- *given $R \subseteq X \times Y$ consider $F_R : X \to PY$ given by $F_R(x) := \{y \in Y \mid (x, y) \in R\}$;*
- *given $F : X \to PY$, define $R_F \subseteq X \times Y$ as $R_F := \{(x, y) \in X \times Y \mid y \in F(x)\}$.*

Related to this, and to the operation of taking the converse of a binary relation, is the following.

Definition A.4 Given a map $F : X \to PY$, we define the map $F^\flat : Y \to PX$ by putting,

$$F^\flat(y) := \{x \in X \mid y \in F(x)\},$$

and we call $F^\flat$ the *transpose* of F. $\lhd$

Note that the relations associated with, respectively, a map $F : X \to P$, and its transpose $F^\flat : Y \to P$ are in fact each other's converse. Observe as well that we have $F = (F^\flat)^\flat$, for every map $F : X \to PY$.

Example A.5 In case Q is a set of proposition letters or variables and S is some set of states, we tend to think of a map from Q to PS as a *valuation*, and of its transpose $F^\flat : S \to P\mathsf{Q}$ as its associated *colouring* or *marking*.

Definition A.6 A *relational structure* or *graph* is a pair $\mathbb{S} = (S, R)$ such that R is a binary relation on S, that is, a set $R \subseteq S \times S$ of S-pairs. In the line of Proposition A.3 we will often think of such a relation R as a map $R : S \to PS$ given by

$$R(s) := \{t \in S \mid (s,t) \in R\}.$$

Elements of the set $R(s)$ will be called *R-successors* of s. ◁

Definition A.7 Fix a relational structure $\mathbb{S} = (S, R)$. A *path* in $\mathbb{S}$ is a sequence $\pi = (t_i)_{0 \leq i < \kappa}$, where κ is either finite or equal to ω, and we have $t_{i+1} \in R(t_i)$ for all i such that $0 \leq i < \kappa$. The *length* of $\pi = (t_i)_{0 \leq i < \kappa}$ is defined as $|\pi| := \kappa$, and we call π *proper* if $\kappa > 0$, *finite* if $0 \leq \kappa < \omega$, and *infinite* if $\kappa = \omega$. We define $\mathsf{first}(\pi) := t_0$, and, provided π is finite, $\mathsf{last}(\pi) := t_{\kappa-1}$. If π is finite we say it is a path *from* $\mathsf{first}(\pi)$ *to* $\mathsf{last}(\pi)$. ◁

Definition A.8 A *tree* is a structure $\mathbb{T} = (T, C, r)$ such that C is a binary relation on T; and r is the *root* of the tree, that is, r is an element of T, such that for every node $t \in T$ there is exactly one path from r to t.

Elements of trees are usually referred to as *nodes*. A *leaf* of $\mathbb{T}$ is a node $t \in T$ such that $C(t) = \varnothing$; nodes that are not leaves are called *inner* nodes. If we have $t \in C(s)$ we say that t is a *child* of s, and, vice versa, that s is the (unique) *parent* of t; distinct children of a node are called *siblings*.

A tree is *finitely branching* if $C(t)$ is finite, for all nodes t; and *well founded* if it has no infinite paths. ◁

It is not hard to prove that the root of a tree is completely determined by its relational structure; that is, we have $r = r'$ if both (T, C, r) and (T, C, r') are trees. For this reason we sometimes represent a tree (T, C, r) simply as the pair (T, C).

B Appendix: The Category Set and its Functors

The theory of coalgebra is categorical in nature. In this appendix we summarize the background knowledge on category theory that is required for understanding the notes; we place a special emphasis on the category Set of sets and functions, since this is the base category of most of the coalgebras that we consider.

For a proper introduction to category theory, the reader is referred to standard textbooks such as Mac Lane's *Categories for the Working Mathematician*, or Awodey's *Category Theory*[13] on which we based parts of this appendix.

B.1 Categories, functors and natural transformations

Definition B.1 A *category* C consists of a class $\mathsf{Ob}(\mathsf{C})$ of *objects*, and for each pair of objects A, B, a family $\mathsf{C}(A, B)$ of *arrows*. If f belongs to the latter set, we write $f : A \to B$, and call A the *domain* and B the *codomain* of the arrow. The collection of arrows is endowed with some algebraic structure: for every object A of C there is an *identity* arrow $\mathsf{id}_A : A \to A$, and every pair $f : A \to B$, $g : B \to C$ can be uniquely *composed* to an arrow $g \circ f : A \to C$. These operations are supposed to satisfy the associative law for composition, while the appropriate identity arrows are left- and right neutral elements. $\lhd$

An arrow $f : A \to B$ is an *iso* if it has an *inverse*, that is, an arrow $g : B \to A$ such that $f \circ g = \mathsf{id}_B$ and $g \circ f = \mathsf{id}_A$.

Example B.2 (a) We let Set denote the category with sets as objects and functions as arrows, with identity arrows and the composition of two arrows defined in the familiar way.

(b) The category Rel has the same objects as Set, but for the set of arrows $\mathsf{Rel}(S', S)$ we take the collections of all binary relations between S' and S, with the identity arrows and the composition of two arrows defined in the obvious way.

(c) The *opposite* category C^{op} of a given category C has the same objects as C, while $\mathsf{C}^{op}(A, B) = \mathsf{C}(B, A)$ for all objects A, B from C, and the operations on arrows are defined in the obvious way.

Definition B.3 A *functor* $F : \mathsf{C} \to \mathsf{D}$ from a category C to a category D consists of an operation mapping objects and arrows of C to objects and arrows of D, respectively, in such a way that $Ff : FA \to FB$ if $f : A \to B$, $F(\mathsf{id}_A) = \mathsf{id}_{FA}$ and $F(g \circ f) = (Fg) \circ (Ff)$ for all objects and arrows involved. A functor $F : \mathsf{C} \to \mathsf{D}^{op}$ is sometimes called a *contravariant* functor from C to D. An *endofunctor* on C is a functor $F : \mathsf{C} \to \mathsf{C}$. $\lhd$

Definition B.4 Let $F, G : \mathsf{C} \to \mathsf{D}$ be two functors. A *natural transformation* $\alpha : F \stackrel{\cdot}{\to} G$ consists of a family of maps $\alpha_A : FA \to GA$, indexed by the collection of objects of C, such that $Gf \circ \alpha_A = \alpha_B \circ Ff$, for every arrow $f : A \to B$ in C. In a diagram:

$$\begin{tikzcd} A \arrow[d, "f"'] & FA \arrow[r, "\alpha_A"] \arrow[d, "Ff"'] & GA \arrow[d, "Gf"] \\ B & FB \arrow[r, "\alpha_B"'] & GB \end{tikzcd} \tag{76}$$

[13] S. Awodey, *Category Theory* (2nd edition), Oxford University Press, 2010.

$\lhd$

B.2 Set functors

Definition B.5 A *set functor* is a (covariant) endofunctor T on the category Set. $\lhd$

Below we give some examples of set functors and of operations on set functors.

Example B.6 (a) Given a set C, we let K_C denote the *constant functor* which maps every set S to the set C, and every map $f : S \to S'$ to the identity map on C. The functor K_C is often simply denoted as C.

(b) The *identity functor Id* is the set functor that maps every object to itself, and similarly maps every arrow to itself.

(c) The *powerset functor* P maps any set S to its power set PS, and any function $f : S \to S'$ to the *direct image map* $Pf : PS \to PS'$ given by $Pf : A \mapsto \{fx \mid x \in X\}$. The *finitary power set functor* P_ω is defined similarly, with the difference that $P_\omega S$ only takes the *finite* subsets of S.

(d) The *contravariant powerset functor* $\breve{P}$ also maps a set S to its power set $\breve{P}S = PS$, but it maps a function $f : S \to S'$ to the *inverse image map* $\breve{P}f : PS' \to PS$ given by $\breve{P}f : X' \mapsto \{x \in S \mid fx \in X'\}$.

(e) Define the covariant set functor $N : \mathsf{Set} \to \mathsf{Set}$ as the composition of the contravariant power set with itself, $N := \breve{P} \circ \breve{P}$. Restricting this example somewhat, we may obtain various interesting functors. For instance, take the functor M given by $MS := \{\mathcal{U} \in NS \mid \mathcal{U}$ is upward closed with respect to $\subseteq$ $\}$ and, for $f : S \to S'$, $Mf = (Nf)\restriction_{MS}$ (it requires a short argument to prove that this defines a functor indeed). N and M are called the *neighbourhood* and the *monotone neighbourhood functor*, respectively.

(f) The *distribution functor* D assigns to a set S the collection $D(S)$ of (discrete) probability distributions over S, i.e., the set of all maps $\mu : S \to [0, 1]$ such that $\sum_{s\in S} \mu(s) = 1$. On arrows, D acts as follows: given a map $f : S \to S'$ and a probability distribution $\mu \in D(S)$, we define the map $(Df)\mu$ on S' by putting

$$(Df)(\mu)(s') := \sum_{s\in f^{-1}(s')} \mu(s).$$

We leave it for the reader to verify that D is indeed a set functor. The main points to check are (i) that $(Df)(\mu)$ is indeed a probability distribution on S', for any $\mu \in D(S)$, and that (ii) $D(g \circ f) = (Dg) \circ (Df)$.

The *finitary* distribution functor D_ω is defined as the restriction of D to probability distributions that have finite *support*, that is, $D_\omega(S) := \{\mu \in DS \mid |S \setminus \mu^{-1}(0)| < \omega\}$. On functions, D_ω is defined as D.

(g) The *bag functor* B is defined analogously. Let $\mathbb{N}^\infty$ be the set $\mathbb{N} \cup \{\infty\}$ of natural numbers extended with the 'number' ∞. We extend the standard addition operation on $\mathbb{N}$ to $\mathbb{N}^\infty$ by putting $n + \infty = \infty + n = \infty + \infty = \infty$ and defining the sum of infinitely many non-zero numbers to be ∞ as well.

Then we define $BS := (\mathbb{N}^\infty)^S$ as the set of *weight functions* $\mu : S \to \mathbb{N} \cup \{\infty\}$. On arrows, B acts similarly as D: given a map $f : S \to S'$ and a weight function μ on S, we define $(Bf)(\mu)$ as the weight function on S' defined by putting

$$(Bf)(\mu)(s') := \sum_{s \in f^{-1}(s')} \mu(s).$$

Similarly to the finitary distribution functor, the *finitary bag functor* B_ω is the restriction of B to bags with finite support, i.e., $B_\omega S := \{\mu : S \to \mathbb{N} \mid \Sigma_{s \in S}\mu(s) < \omega\}$.

(h) The *binary tree functor* is the functor $Id^2 := Id \times Id$.

There are various ways to obtain new functors from old.

Example B.7 Let F, F_0 and F_1 be set functors.

(a) The *composition* of F_0 and F_1, denoted as $F_1 \circ F_0$, is defined in the obvious way, e.g. on objects we put $(F_1 \circ F_0)(S) := F_1(F_0(S))$.

(b) The *product* $F_0 \times F_1$ of F_0 and F_1 is given (on objects) by $(F_0 \times F_1)S := F_0 S \times F_1 S$, while for $f : S \to S'$, the map $(F_0 \times F_1)f$ is given as $((F_0 \times F_1)f)(\sigma_0, \sigma_1) := ((F_0 f)(\sigma_0), (F_1 f)(\sigma_1))$.

(c) The *co-product* $F_0 + F_1$ of F_0 and F_1 is defined in a similarly straightforward way (note that on Set we may think of co-product as disjoint union).

(d) Given a set D, we let the D*-exponent functor* F^D be defined as follows. Given a set S, we put $F^D(S) := (F(S))^D$, that is, the set of maps from D to FS. Given an arrow $f : S \to S'$ and a function $h : D \to FS$, we simply define the arrow $F^D f$ as the function $(Ff) \circ h$.

Of specific interest in the context of coalgebra and modal logic is the following operation on set functors, which generalises the relation between Kripke frames and Kripke models to the level of arbitrary coalgebras over Set. Think of Q as an arbitrary but fixed set of *proposition letters*.

Definition B.8 Given a set functor T and a set Q of proposition letters, we define T_Q as the functor $T_\mathsf{Q} := K_{P\mathsf{Q}} \times T$. We may refer to T_Q as the T*-model functor* associated with Q. $\lhd$

Definition B.9 The collection of *Kripke polynomial set functors* or *KPF* is defined by the following 'grammar':

$$K \ ::= \ K_C \mid Id \mid K_0 \times K_1 \mid K_0 + K_1 \mid K^D \mid P \circ K, \tag{77}$$

where C and D are sets. The *polynomial set functors* are the ones obtained by the same grammar without the powerset functor, and the *finitary* KPFs are obtained by the version of (77) where P is replaced with P_ω. $\lhd$

Apart from the operations described in Example B.7, in the context of coalgebras there is (at least) one other way of interest to obtain new set functors from old, namely, to take the *finitary version* of a functor. For this we need to introduce some notation and terminology concerning inclusions.

Definition B.10 Given two sets A, B with $A \subseteq B$, we let $\iota_B^A : A \to B$ denote the associated *inclusion map*, i.e., $\iota_B^A : a \mapsto a$; we will also write $f : A \hookrightarrow B$ to denote that $f = \iota_B^A$ (and so, in particular, this means that $A \subseteq B$). We say that a set functor T *preserves inclusions* if, for every pair of sets A, B with $A \subseteq B$, we have that $TA \subseteq TB$ and $T\iota_B^A = \iota_{TB}^{TA}$. $\lhd$

► `Examples: power set vs mon nbh`

In case $T : \mathsf{Set} \to \mathsf{Set}$ preserves inclusions we define the notion of *support* as follows.

Definition B.11 Let T be an inclusion preserving set functor, and suppose that $\alpha \in TA$. We say that a subset $B \subseteq A$ *supports* α if $\alpha \in TB$. $\lhd$

Definition B.12 Given a set functor T, we define the following operation T_ω on an arbitrary set S and an arbitrary function $f : S \to S'$:

$$\begin{array}{rcl} T_\omega(S) & := & \{(T\iota_S^X)(\xi) \mid \xi \in TX \text{ for some } X \subseteq_\omega S\}, \\ T_\omega(f) & := & (Tf)\restriction_{T_\omega S} . \end{array}$$

We will call T_ω the *finitary version* of T. $\lhd$

Given the definition of T_ω on functions, we may write Tf instead of $T_\omega f$ without causing confusion. Observe that we obtain $T_\omega S \subseteq TS$, and that, in case T preserves inclusions, the definition of $T_\omega S$ simplifies to $T_\omega S = \bigcup\{TX \mid X \subseteq_\omega S\}$. Whether this is the case or not, we always have that T_ω is a functor (which justifies that we call it the finitary *version* of T).

Proposition B.13 *(1) If T is a set functor, then so is T_ω.*

(2) If T preserves inclusions, then so does T_ω.

Proof. We only prove part (1) of the proposition, where the key property to establish is that for any map $f : S \to S'$, the arrow Tf is well-typed; that is, $\sigma \in T_\omega S$ implies $(Tf)\sigma \in T_\omega S'$. To see why this is the case, take an arbitrary object $\sigma \in T_\omega S$, and let $X \subseteq_\omega S$ and $\xi \in TX$ be such that $\sigma = (T\iota)(\xi)$, where we write $\iota = \iota_S^X$ to simplify notation.

Now consider the following two diagrams, where we let ι' denote the inclusion arrow $\iota' : f[X] \hookrightarrow S'$:

$$\begin{tikzcd} X \arrow[r, hook, "\iota"] \arrow[d, "f\restriction_X"'] & S \arrow[d, "f"] \\ f[X] \arrow[r, hook, "\iota'"'] & S' \end{tikzcd} \qquad \begin{tikzcd} TX \arrow[r, tail, "T\iota"] \arrow[d, "T(f\restriction_X)"'] & TS \arrow[d, "Tf"] \\ Tf[X] \arrow[r, tail, "T\iota'"'] & TS' \end{tikzcd}$$

Since the left diagram commutes, so does the right one. From this it follows that $(T_\omega f)(\sigma) = (Tf)(\sigma) = (Tf)(T\iota)(\xi) = (T\iota')(Tf\restriction_X)(\xi)$ and since $f[X]$ is a *finite* subset of S', with inclusion map ι', this suffices to show that $(Tf)(\sigma) \in T_\omega S'$ indeed. QED

B.3 Limits and colimits in Set

Definition B.14 The *product* of two objects A_0 and A_1 in a category C is an object $A_0 \times A_1$, together with two *projection arrows* $\pi_i : A_0 \times A_1 \to A_i$, such that for any pair of arrows $f_i : X \to A_i$ there is a *unique* arrow $: X \to A_0 \to A_1$ such that the following diagram commutes:

$$\begin{tikzcd} A_0 & A_0 \times A_1 \arrow[l, "\pi_0"'] \arrow[r, "\pi_1"] & A_1 \\ & X \arrow[ul, "f_0"] \arrow[u, dotted, "u"] \arrow[ur, "f_1"'] & \end{tikzcd} \tag{78}$$

We will often denote the arrow u as $\langle f_0, f_1 \rangle$.

Dually, the *co-product* of two objects A_0 and A_1 in a category C is an object $A_0 + A_1$, together with two *insertion arrows* $\kappa_i : A_i \to A_0 \times A_1$, such that for any pair of arrows $f_i : A_i \to X$ there is a *unique* arrow $v : A_0 \to A_1 \to X$ such that the following diagram commutes:

$$\begin{tikzcd} A_0 \arrow[r, "\kappa_0"] \arrow[dr, "f_0"'] & A_0 + A_1 \arrow[d, dotted, "v"] & A_1 \arrow[l, "\kappa_1"'] \arrow[dl, "f_1"] \\ & X & \end{tikzcd} \tag{79}$$

The mediating arrow v will usually be denoted as $[f_0, f_1]$. $\lhd$

Example B.15 In the category Set, we take for the product of two sets S_0 and S_1 their *cartesian product* $S_0 \times S_1 := \{(s_0, s_1) \mid s_i \in S_i\}$, with the obvious projection maps $\pi_i : (s_0, s_1) \mapsto s_i$. For a concrete representation of the co-product of S_0 and S_1 we take the *sum* or *disjoint union* $S_0 + S_1 := (\{0\} \times S_0) \cup (\{1\} \times S_1)$, with the insertion maps $\kappa_i : s \to (i, s)$.

Definition B.16 The binary product and co-product of Definition B.14 are easily generalised to (co-)products over an arbitrary index set I; we omit the details, but introduce the notation $\prod_{i \in I} A_i$ and $\coprod_{i \in I} A_i$ for the product and co-product of the family $\{A_i \mid i \in I\}$. Products and co-products of the empty family are called *final* respectively *initial* objects of the category. $\lhd$

Definition B.17 Given two 'parallel' arrows $f_i : A \to B$ in a category C, we define an *equalizer* of f_0 and f_1 as an arrow $g : X \to A$ that satisfies the equality $f_0 \circ g = f_1 \circ g$, and the following condition. For every arrow $g' : X' \to A$ such that $f_0 \circ g' = f_1 \circ g'$, there is a *unique* arrow $u : X' \to X$ such that $g' = g \circ u$, cf. the diagram on the left:

$$\begin{tikzcd} X \arrow[r, "g"] & A \arrow[r, shift left, "f_0"] \arrow[r, shift right, "f_1"'] & B \\ X' \arrow[u, dotted, "u"] \arrow[ur, "g'"'] & & \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, shift left, "f_0"] \arrow[r, shift right, "f_1"'] & B \arrow[r, "h"] \arrow[dr, "h'"'] & Y \arrow[d, dotted, "v"] \\ & & Y' \end{tikzcd}$$

Dually, a *co-equalizer* of f_0 and f_1 is an arrow $h : B \to Y$ satisfying $h \circ f_0 = h \circ f_1$ and the universal property as indicated in the diagram to the right. $\lhd$

Example B.18 In the category Set, as the equalizer of two functions $f_i : S \to S'$, we can take the set $\mathsf{eq}(f_0, f_1) := \{s \in S \mid f_0 s = f_1 s\}$, together with the inclusion map $\iota : \mathsf{eq}(f_0, f_1) \hookrightarrow S$.

As the co-equalizer of $f_0, f_1 : S \to S'$, consider the *quotient* S' under the equivalence relation E_{f_0,f_1} generated by the set $\{(f_0 s, f_1 s) \mid s \in S\}$, together with the quotient map $q : S' \to S'/E_{f_0,f_1}$ mapping any $s' \in S'$ to its own equivalence class under E_{f_0,f_1}.

Definition B.19 Given two arrows $f_i : A_i \to B$ in a category C, a *pullback* of f_0 and f_1 is an object P, together with two arrows $p_i : P \to A_i$ which satisfy $f_0 \circ p_0 = f_1 \circ p_1$, together with the following condition. Given any 'competitor' P', with arrows $p'_i : P' \to A_i$ such that $f_0 \circ p'_0 = f_1 \circ p'_1$, there is a *unique* arrow $u : P' \to P$ such that $p'_i = p_i \circ u$, in a diagram:

$$
\begin{tikzcd}
 & A_0 \arrow[r, "f_0"] & B \\
 & P \arrow[u, "p_0"'] \arrow[r, "p_1"] & A_1 \arrow[u, "f_1"'] \\
P' \arrow[uur, "p'_0"] \arrow[ur, dotted, "u"] \arrow[urr, "p'_1"'] & &
\end{tikzcd}
$$

Dually we define the notion of a *pushout* of two arrows $f_i : B \to A_i$. $\lhd$

Example B.20 In the category Set we can define, given two functions $f_i : S_i \to S$, the set $\mathsf{pb}(f_0, f_1) := \{(s_0, s_1) \in S_0 \times S_1 \mid f_0(s_0) = f_1(s_1)\}$, and show that this set, together with the projection maps $\pi_i : \mathsf{pb}(f_0, f_1) \to S_i$, is the pullback of f_0 and f_1.

For the pushout, consider arrows $f_i : A \to B_i$, for $i = 0, 1$. Let E be the equivalence relation on the disjoint union $B_0 + B_1$ which is generated from the set of pairs $\{(f_0 a_0, f_1 a_1) \mid a \in A\}$. Let C be the quotient of $B_0 + B_1$ under E, that is, C is the set of E-cells, and let $e_i : B_i \to C$, for $i = 0, 1$, be the quotient map, restricted to B_i. That is, $e_i(b_i)$ is the equivalence class of $b_i \in B_i$ under the equivalence relation E. It is not hard to see that C, together with the functions e_0 and e_1, is the pushout of f_0 and f_1. One may also show that e_i is injective (surjective) if f_i is so.

The concepts of product, equalizer and pullback are not independent.

Fact B.21 *The following are equivalent, for any category* C*:*

(1) C *has finite products and equalizers;*

(2) C *has pullbacks and a final object.*

More in general we can define the notion of a limit or colimit of a diagram.

Definition B.22 Let J and C be categories, and assume that J is *small*, that is, its collection of objects forms a set (rather than a proper class). A *diagram* of *type* J in C is a functor $D : \mathsf{J} \to \mathsf{C}$. We refer to J as the *index set* of D, and write D_i rather than $D(i)$, where i is an arbitrary object or *index* in J. $\lhd$

Example B.23 Here are four examples of diagrams (where we do not draw identity arrows):

$$A_0 \qquad A_1 \qquad \Big| \qquad \begin{tikzcd} A \arrow[r, shift left, "f_0"] \arrow[r, shift right, "f_1"'] & B \end{tikzcd} \qquad \Big| \qquad \begin{tikzcd} A_0 \arrow[r, "f_0"] & B \\ & A_1 \arrow[u, "f_1"'] \end{tikzcd} \qquad \Big| \qquad \begin{tikzcd} B_0 & \\ A \arrow[u, "f_0"] \arrow[r, "f_1"'] & B_1 \end{tikzcd}$$

Definition B.24 A *cone* to a diagram $D : \mathsf{J} \to \mathsf{C}$ consists of an object C in C, together with an arrow $c_j : C \to D_i$ for each object i in J, such that for each arrow $e : i \to j$ in J, the following diagram commutes:

$$\begin{tikzcd} C \arrow[d, "c_i"'] \arrow[dr, "c_j"] & \\ D_i \arrow[r, "D(e)"'] & D_j \end{tikzcd} \tag{80}$$

A *morphism of cones* $\gamma : (C, c_i)_{i \in \mathsf{J}} \to (C', c'_i)_{i \in \mathsf{J}}$ is an arrow $\gamma : C \to C'$ in C such that $c_i = c'_i \circ \gamma$ for each index i:

$$\begin{tikzcd} C \arrow[r, "\gamma"] \arrow[dr, "c_i"'] & C' \arrow[d, "c'_i"] \\ & D_i \end{tikzcd}$$

The notion of a cone dualizes to that of a *co-cone* in the obvious way. We let $\mathsf{Cone}(D)$ and $\mathsf{CoCone}(D)$ denote the emerging categories of cones and co-cones, respectively. $\lhd$

Definition B.25 Let $D : \mathsf{J} \to \mathsf{C}$ be a diagram. A *limit* for D is a terminal object in the category $\mathsf{Cone}(D)$, and a *colimit* for D is an initial object in the category $\mathsf{CoCone}(D)$. A (co-)limit is called *finite* if the index category J is finite. $\lhd$

Spelled out, the limit of a diagram $D : \mathsf{J} \to \mathsf{C}$ is a cone to D, that is, an object C in C, together with a family $p_i : C \to D_i$ of arrows in C satisfying the cone condition (80), and such that for any D-cone $(C', c'_i)_{i \in \mathsf{J}}$ there is a unique cone morphism $u : (C', c'_i)_{i \in \mathsf{J}} \to (C, c_i)_{i \in \mathsf{J}}$.

Example B.26 Limits for the first three diagrams in Example B.23 may easily be identified with respectively products, equalizers and pullbacks of the objects and morphisms displayed. Co-products, co-equalizers and pushouts can be seen as the co-limits of, respectively, the first, second, and fourth diagram.

Limits and colimits do not always exist; if limits (for a certain type of diagram) always exist in a category C, we say that C *has limits (of that type)*.

Fact B.27 *For any category* C *the following are equivalent.*

(1) C *has all (finite) limits;*

(2) C *has all equalizers and (finite) products.*

Fact B.28 *The category* Set *has all limits and colimits.*

B.4 Properties of set functors

In this section we discuss some properties of set functors that are of interest in the setting of coalgebra. First we recall some definitions.

Definition B.29 Let T be a set functor.

(1) T *restricts to finite sets* if TS is finite whenever S is finite.

(2) T is *smooth* if its preserves weak pullbacks. $\lhd$

Proposition B.30 *Let T be a set functor. Then T preserves surjections and non-empty injections. That is, if $f : A \to B$ is surjective then so is $Tf : TA \to TB$; and if $f : A \to B$ is injective then so is $Tf : TA \to TB$, provided A is non-empty.*

Proof. First let $f : A \to B$ be surjective, and take any map $g : B \to A$ such that $f \circ g = \mathsf{id}_B$. It follows by functorialiy that $Tf \circ Tg = \mathsf{id}_{TB}$ which immediately implies the surjectivity of Tf.

Similarly, if $A \neq \varnothing$ and $f : A \to B$ is injective then we may consider a map $h : B \to A$ such that $h \circ f = \mathsf{id}_A$. Again by functoriality we find $Th \circ Tf = \mathsf{id}_{TA}$ which implies the injectivity of Tf. QED

Remark B.31 The above proof breaks down for empty injections, and indeed one may come up with functors that do not preserve the injectivity of the empty map. For instance, as a variation of the constant functor K_C, consider a non-injective map $h : D \to C$. We let K_h be the set functor given as follows. On objects we have $K_h(\varnothing) := D$, while $K_h(A) := C$ if $A \neq \varnothing$; and for an arrow $f : A \to B$ we define $K_h(f)$ to be id_D if $A = B = \varnothing$, id_C if $A \neq \varnothing$, and h if $A = \varnothing \neq B$. In other words, for any non-empty set X we find $K_h(!_X) = h$. This problem is avoided by requiring the functor to be *standard*.

It is often convenient to assume that $TA \subseteq TB$ if $A \subseteq B$; the following property requires a bit more.

Definition B.32 Where A and B are sets such that $A \subseteq B$, we let $\iota^A_B : A \to B$ denote the corresponding *inclusion map*. We say that a set functor T is *standard* if it preserves inclusions, that is, if $A \subseteq B$ then $TA \subseteq TB$ and $T\iota^A_B = \iota^{TA}_{TB}$. $\lhd$

One benefit of this property is that working with standard functors may reduce cognitve (and notational) clutter. Note as well that, in the case where $A = \varnothing$, we find that $T\varnothing \subseteq TB$, and T maps the unique function $!_B : \varnothing \to B$ to the inclusion $\iota^{T\varnothing}_{TB}$. In other words, standard functors preserve the injectivity of all maps. The main advantage of standard functors is that they have various other nice properties. For instance, they preserve non-empty *intersections*.

Proposition B.33 *Let T be a standard set functor. Then T preserves non-empty intersections, i.e., for any pair of sets A, B with $A \cap B \neq \varnothing$, we have*

$$T(A \cap B) = TA \cap TB. \tag{81}$$

If, in addition, T is smooth, then (81) *holds for every pair of sets A, B.*

▶ `add proof sketch or reference [Trnková 1969]`

Restricting attention to standard functors is relatively harmless in the setting of coalgebras, because of the following fact which states that set functors are 'almost' standard.

Proposition B.34 *Let T be some set functor. Then there is a* standard *set functor T', and, for every set S, a bijection mapping T-coalgebra arrows $\sigma : S \to TS$ to T'-coalgebra arrows $\sigma' : S \to T'S$. This family of bijections induces an isomorphism between the categories $\mathsf{Coalg}(T)$ and $\mathsf{Coalg}(T')$.*

Proof. Following the proof of [A&T,III.4.5] we may find a standard functor T' such that the restrictions of T and T' to non-empty sets are naturally isomorphic. Let η denote this natural isomorphism; that is, we have a natural family of bijections $\eta_S : TS \to T'S$, for non-empty sets S. We extend η to $\mathsf{Coalg}(T)$ as follows. Given a T-coalgebra $\mathbb{S} = (S, \sigma)$, we define the map $\sigma^\eta : S \to T'S$ by putting

$$\sigma^\eta := \begin{cases} !_{T'\varnothing} & \text{if } S = \varnothing \\ \eta_S \circ \sigma & \text{otherwise,} \end{cases}$$

and defining $\mathbb{S}^\eta := (S, \sigma^\eta)$. It is straightforward to verify that this constitutes the required isomorphism between $\mathsf{Coalg}(T)$ and $\mathsf{Coalg}(T')$. QED

Example B.35 The bag functor B of Example B.6 is not standard, but we may 'standardize' it by representing any map $\mu : X \to \mathbb{N}^\infty$ via its 'positive graph' $\{(x, \mu x) \mid \mu x > 0\}$. A similar approach works for the finitary bag functor B_ω, and for the distribution functors D and D_ω.

B.5 Relation lifting

Recall that the concept of *relation lifting* was introduced in Definition 3.6.

▶ `Recall definition here`

In the propositions below we list various properties of relation lifting — proof details can be found in [KKV]. We start with some properties of $\overline{T}$ that hold for any functor T.

Fact B.36 *Let T be a set functor. Then the relation lifting $\overline{T}$ has the following properties:*

1. *$\overline{T}$ extends T: $\overline{T}(\mathsf{Gr} f) = \mathsf{Gr}(Tf)$ for all functions $f : A_0 \to A_1$,*
2. *$\overline{T}$ preserves diagonals: $\overline{T}\Delta_A = \Delta_{TA}$ for any set A;*
3. *$\overline{T}$ preserves converse: $\overline{T}R^{\smile} = (\overline{T}R)^{\smile}$ for all relations $R \subseteq A_0 \times A_1$;*
4. *$\overline{T}$ is monotone: $R \subseteq Q$ implies $\overline{T}R \subseteq \overline{T}Q$ for all relations $R, Q \subseteq A_0 \times A_1$;*
5. *$\overline{T}$ is semi-functorial: $\overline{T}(R \,;\, Q) \subseteq \overline{T}R \,;\, \overline{T}Q$, for all relations $R \subseteq A_0 \times A_1, Q \subseteq A_1 \times A_2$.*

In case the functor T is *smooth*, Proposition B.36(5) can be strengthented to full functoriality. In fact, we have the following equivalence, which explains the importance of smoothness in the theory of coalgebras.

Fact B.37 *For any set functor T the following are equivalent:*
(1) T is smooth;
(2) $\overline{T}(R\,;Q) = \overline{T}R\,;\overline{T}Q$, for all relations $R \subseteq A_0 \times A_1, Q \subseteq A_1 \times A_2$.
(3) $\overline{T}$ is an endofunctor on the category Rel *of sets and binary relations.*

In case T is standard, some more properties are preserved.

Remark B.38 ► `First, however, we argue why for standard functors the definition of` $\overline{T}R$ `does not depend on the ambient sets of` R.

Fact B.39 *Let T be a smooth and standard set functor. Then the relation lifting $\overline{T}$ has the following properties:*

1. *$\overline{T}$ preserves domain:* $\mathsf{Dom}(\overline{T}R) = T(\mathsf{Dom}R)$, *for all relations $R \subseteq A_0 \times A_1$;*
2. *$\overline{T}$ preserves range:* $\mathsf{Ran}(\overline{T}R) = T(\mathsf{Ran}R)$, *for all relations $R \subseteq A_0 \times A_1$;*
3. *$\overline{T}$ preserves composition: $\overline{T}(R;Q) = \overline{T}R;\overline{T}Q$, for all relations $R \subseteq A_0 \times A_1, Q \subseteq A_1 \times A_2$;*
4. *$\overline{T}$ preserves restrictions:*
$$\overline{T}(R\restriction_{B_0 \times B_1}) = (\overline{T}R)\restriction_{TB_0 \times TB_1}$$
for all relations $R \subseteq A_0 \times A_1$, and all sets $B_0 \subseteq A_0$ and $B_1 \subseteq A_1$;
5. *$\overline{T}_\omega$ aligns with $\overline{T}$: $\overline{T}_\omega R = \overline{T}R \cap (T_\omega A_0 \times T_\omega A_1)$, for all relations $R \subseteq A_0 \times A_1$.*